



Université du Québec
Institut national de la recherche scientifique
Centre-Énergie Matériaux Télécommunications

Integration of Clouds and Cloudlets in Energy-Aware Fiber-Wireless Access Networks

by

Bhaskar Prasad Rimal

Dissertation submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy (Ph.D.) in Telecommunications

April 2017

Thesis Committee:

External Examiner	Prof. Christine Tremblay (École de Technologie Supérieure)
External Examiner	Prof. Zbigniew Dziong (École de Technologie Supérieure)
Internal Examiner	Prof. Jean-Charles Grégoire (INRS-ÉMT)
Research Director	Prof. Martin Maier (INRS-ÉMT)

*In loving memory of my late mother Jayanti Rimal
who is the major source of inspiration
and encouragement throughout my whole life.
No words are sufficient to describe your contributions.
You are still alive in my heart, thank you mom! . . .*

*In loving memory of my late aunt Sabitri Rimal
for being my friend and the best guardian.
You will be in my memory forever. . .*

*To my father Tanka Natha Rimal
for your constant encouragement and tireless devotion. . .*

Abstract

An integrated Fiber-Wireless (FiWi) broadband access network consists of a Wireless Local Area Network (WLAN) in the wireless front-end segment and an Ethernet Passive Optical Network (EPON), a widely deployed variant of the PON technology, in the optical backhaul segment. This thesis first investigates the energy efficiency issue in FiWi access networks in a comprehensive fashion to extend the battery life of wireless devices while improving the energy efficiency of the optical backhaul. A novel energy conservation scheme is proposed that jointly schedules power-saving modes for both wireless devices (wireless station, access point) in the front-end and optical network units (ONUs) in the optical backhaul. The proposed scheme maximizes the overall network performance by leveraging Time Division Multiple Access (TDMA) to synchronize the power-saving modes and incorporate them into the underlying dynamic bandwidth allocation (DBA) process. The obtained results show that, by utilizing the proposed energy efficiency solutions, more than 70% energy savings can be achieved while preserving upstream delay and incurring a low delay for downstream traffic for the considered evaluation scenarios. That not only helps operators reduce their Operational Expenditures (OPEX) but also holds promise to usher in a low-carbon networked society in years to come.

Cloud computing has been widely used in many applications and in new evolving areas. This thesis investigates the workflow scheduling issue in multi-tenant cloud environments. A four-tier workflow management system is proposed and a novel cloud-based workflow scheduling algorithm (CWSA) is developed that helps minimize the overall workflow completion time, tardiness, cost of execution of the workflows, and utilize idle cloud resources effectively. The simulation results show that the proposed solution improves the workflow performance and outperforms other state-of-the-art scheduling policies as well as provides scalability for the considered evaluation scenarios.

Furthermore, this thesis proposes to enhance capacity-centric FiWi broadband access networks based on data-centric Ethernet technologies with computation- and storage-centric cloudlets to pro-

vide reliable cloud services at the edge of FiWi networks and thereby realize the vision of mobile-edge computing/multi-access edge computing (MEC). To reduce offload delay and prolong battery life of edge devices, novel cloudlet-aware resource management algorithms are proposed that incorporate offloading operations into the underlying FiWi DBA process. To thoroughly study the scheme's performance, a comprehensive analytical framework is developed. The obtained results demonstrate the feasibility and effectiveness of cloudlet enhanced FiWi networks for MEC.

A conventional cloud has high storage and processing capabilities, but it may suffer from a large wide area network (WAN) latency. The emerging concept of MEC or cloudlet may offer lower latency, though it has limited computing and storage capabilities. The conventional centralized clouds and decentralized cloudlets (i.e., MEC) may coexist and be complementary to each other in order to support a more diverse set of emerging applications and services (e.g., low-latency, mission-critical, and location-aware) in 5G networks. In light of this, going beyond traditional access systems and given the important role of optical backhauling in 5G networks as well as mobile-cloud convergence, the major part of this Ph.D. thesis explores FiWi enhanced access network architectures and investigates the performance gains of centralized cloud and cloudlet/MEC enabled FiWi access networks. This thesis explores a network architecture that integrates cloud and cloudlets (or MEC) in FiWi access networks. Subsequently, a novel unified resource management scheme incorporating both centralized cloud and MEC offloading capabilities into the underlying FiWi DBA process is proposed. A comprehensive analytical framework is developed to model packet delay, response time efficiency, and gain-offload overhead ratio for both cloud and conventional broadband access traffic. In addition, given the importance of reliability in optical backhaul and MEC, this thesis develops a probabilistic survivability analysis model to assess the impact of both fiber cuts and MEC server failures. The obtained results demonstrate the feasibility of implementing conventional cloud and MEC in FiWi access networks without affecting the network performance of broadband access traffic. Further, the obtained results indicate that the use of cloudlets at the edge of FiWi access networks can achieve a significantly reduced end-to-end latency and an enhanced overall network performance. Moreover, the results demonstrate the effectiveness of the presented survival schemes by providing wireless stations (STAs) and end users with highly fault-tolerant FiWi connectivity. By jointly employing feeder fiber (FF), distribution fiber (DF), interconnection fiber (IF), wireless protection, and MEC redundancy, the FiWi connectivity probability of STAs is increased even in the case of higher PON fiber link failure probabilities.

The last part of this thesis investigates the emerging application scenario of cloud-cloudlet enhanced FiWi access networks in support of the future so-called Tactile Internet. The concept of the Tactile Internet is elaborated and the taxonomy of enabling technologies for the Tactile Internet is proposed. The role of cloudlet/MEC capable FiWi access networks in facilitating the Tactile Internet is then investigated in greater detail. The obtained results show that the ultra-low end-to-end latency requirements of the Tactile Internet cannot be achieved without bringing computation and storage (i.e., cloudlet or MEC) to the edge of FiWi access networks.

Keywords: Backhaul, cloud computing, cloudlet, direct acyclic graph (DAG), dynamic bandwidth allocation (DBA), energy efficiency, fiber-wireless (FiWi), multi-tenancy, mobile-edge computing (MEC), passive optical network (PON), resource management, scientific workflow applications, survivability, Tactile Internet.

Acknowledgments

Working on the Ph.D. has been a wonderful experience not only in terms of learning but also exploring new innovative ideas. I am very grateful and indebted to many people. I could not have completed my Ph.D. studies without all of their continued support and guidance along the way.

First and foremost, I would like to express my sincere thanks and humble gratitude to my advisor Prof. Martin Maier, the founder and creative director of the Optical Zeitgeist Laboratory, for his expert guidance, invaluable feedback, financial support, and encouragements throughout my research work. His extensive wide knowledge, enthusiasm, and commitment to high-quality research was motivational for me throughout this work that extremely helped me not only how to solve problems and to aim high, but also sharpen my research ability and grow as a research professional. I feel very fortunate to have had the chance to work under his guidance.

Beside my advisor, I would like to extend my sincere gratitude to the other dissertation committee members, Prof. Christine Tremblay, Prof. Zbigniew Dziong, and Prof. Jean-Charles Grégoire for their insightful comments and feedback. Further, I am very thankful to the government of Québec for valuing my work and granting me the prestigious FRQNT Merit scholarship during my Ph.D. studies. I would also like to acknowledge all the administrative and technical staff members of INRS. In particular, I thank H el ene Sabourin, Nathalie Aguiar, and Sylvain Fauvel for their kind support throughout my Ph.D. studies.

I am extremely thrilled about the opportunity to work with all my co-authors. I would like to express my gratitude to them. More specifically, I would like to express my appreciation to Dr. Dung Pham Van at KTH Royal Institute of Technology, Sweden, for his technical advice, helpful comments, constructive criticism, and important contributions. Further, I also thank Prof. Lena Wosinska, Assoc. Prof. Jiajia Chen, Assoc. Prof. Paolo Monti at KTH Royal Institute of Technology, Sweden, Assoc. Prof. Luca Valcarengi at Scuola Superiore Sant'Anna, Pisa, Italy, Dr.

Sergey Andreev (Tampere University of Technology, Finland), and Dr. Tuomas Tirronen (Ericsson Research, Finland) for their valuable collaborations and intellectual inputs, for which I am grateful.

Furthermore, I would like to thank Prof. Mahadev Satyanarayanan for hosting me during the research visit in his group at Carnegie Mellon University (CMU), Pittsburgh, PA, USA. Thanks to the current members of the Elijah project at CMU for their help. In particular, I extend special thanks to Zhuo Chen, Ph.D. Candidate, for helping me settle down in the lab in my initial day at CMU and for fruitful discussions on augmented reality. My research visit to CMU was really interesting and definitely helped broaden my horizons.

I thank all my current fellow lab mates of the Optical Zeitgeist Laboratory for their continued support. In addition, I also thank my former lab mates, Dr. Martin Lévesque, Asst. Prof. Hamzeh Beyranvand (Amirkabir University of Technology-Tehran Polytechnic, Iran), Asst. Prof. Wansu Lim (Kumoh National Institute of Technology, South Korea), and Ahmed Belgana for their helpful suggestions. I would also like to thank Oussama Ben Smida and Nesrine Cherif at INRS-ÉMT for helping me prepare the French overview of this dissertation and Dr. Martin Lévesque for reviewing it.

On a more personal note, I would like to thank all of my family members for their unconditional love, faith, endless support, and patience throughout my Ph.D studies. Thank you all for being there for me. Further, special thanks to my wonderful wife Sara, who came into my life at exactly the right time and helped manage a long-distance relationship with grace and generosity. Thank you for your quiet patience, understanding, emotional support, and encouragement. Finally, dozens of people who helped me in some way, thank you, everyone!

Bhaskar Prasad Rimal

Pittsburgh, PA, USA

December 2016

Résumé

Introduction et motivation

Les réseaux optiques passifs (PONs) sont considérés comme la solution d'accès filaire la plus attrayante grâce à leur capacité élevée, leur fiabilité, leur faible coût, leur portée accrue et leur facilité de déploiement, résultant de leur nature de réseaux tous-passifs et de la topologie point-à-multipoint [1, 2]. Les réseaux d'accès PONs sont fondamentaux pour permettre des services à haut débit rapides et ultra-rapides [3]. Puisque les futures technologies de réseau d'accès devront supporter une capacité très élevée pour faire face à la croissance massive des demandes en trafic, la technologie des fibres optiques est la technologie de fait à l'épreuve du futur pour l'accès à large bande.

Au cours de ces dernières années, l'UIT-T et IEEE ont mis tous leurs efforts pour augmenter la capacité des réseaux PON, et ces systèmes sont maintenant matures [4]. Par exemple, récemment, British Telecom a démontré des technologies optiques Terabit à haute vitesse [5] et des essais réussis d'un super canal optique 3Tb/s qui permettra aux opérateurs d'étendre les réseaux de transmission optique de base de façon à répondre à l'augmentation anticipée des demandes d'applications à large bande passante dans les réseaux mobiles de la 5ème génération (5G) et au-delà de cette dernière. Ce pendant, au lieu d'augmenter constamment la capacité des PON de prochaine génération (NG-PON), l'objectif de la recherche dans les réseaux d'accès à large bande devraient se diriger vers l'amélioration de l'efficacité et le soutien des services et d'applications émergents (par exemple, cloud computing et cloudlet, backhauling mobile et fronthauling, virtualisation). Vers cette fin, il se peut qu'il soit impossible de déployer des fibres optiques n'importe où à cause de la situation géographique ou lorsque la mobilité est une nécessité.

Les technologies d'accès sans fil existantes (par exemple, la Fidélité au sans fil (WiFi) [6], l'évolution à long terme de 3GPP (LTE) [7] et LTE avancé (LTE-A) [8]) peuvent, en revanche, fournir la mobilité de l'utilisateur mais elles exigent un backhaul fiable à grande capacité pour répondre aux exigences de la bande passante des applications émergentes, tels que la télévision à haute définition TVHD, la vidéoconférence de haute qualité et les écrans tridimensionnels qui ont tous le potentiel d'exiger de plus en plus de bande passante. En plus, l'imagerie holographique, les expériences immersives, la réalité virtuelle et augmentée, la télémédecine, l'apprentissage à distance et d'autres applications à large bande passante continueront à augmenter la demande de connectivité filaire à haut débit. Cette tendance s'aligne également avec la loi d'Edholm sur la bande passante [9] qui stipule qu'un réseau optique filaire et sans fil unifié (ou convergent) est nécessaire pour fournir des services fixes et mobiles aux utilisateurs. Dans ce contexte, la convergence des réseaux de fibres et des réseaux sans fil, appelés réseaux d'accès à large bande de fibres et sans fil (FiWi) [10, 11] est introduite comme une solution importante pour remédier aux lacunes précitées et répondre à une demande toujours croissante.

Compte tenu de l'importance primordiale de l'efficacité énergétique dans les réseaux d'accès filaire et sans fil, la conception et l'implémentation des mécanismes d'économie d'énergie pour prolonger la durée de vie de la batterie des périphériques à batterie tout en réduisant la consommation d'énergie du réseau backhaul optique qui est un problème important ouvert à l'enquête. Il est à noter que les réseaux backhaul optique et les réseaux cellulaires sans fil ne sont pas efficaces en énergie car leur faible utilisation moyenne entraîne un énorme gaspillage d'énergie [12]. Par conséquent, les mécanismes d'économie d'énergie de bout-en-bout qui considèrent des segments de fibres et sans fil, sont hautement nécessaires pour les réseaux d'accès FiWi. En outre, au-delà des systèmes d'accès traditionnels et compte tenu du rôle important du réseau optique dans les réseaux 5G en ce qui concerne le besoin de convergence fixe/mobile, l'architecture du réseau d'accès FiWi explorée dans cette thèse est renforcée par des capacités de traitement du cloud centralisées et distribuées afin de créer de services plus efficaces et de soutenir une grande variété d'applications. Il est intéressant de noter que le segment frontal de réseaux FiWi peut être basé non seulement sur 3G et 4G LTE, mais aussi sur les technologies cellulaires 5G émergentes.

Le cloud computing est défini comme "un modèle de services et d'accès où des ressources virtualisées et dynamiquement évolutives sont fournies en tant que service sur internet [13]." Le cloud computing permet l'accès aux infrastructures (par exemple, Amazon Elastic Compute Cloud (EC2),

aux plate-formes (Par exemple, Google AppEngine) et aux logiciels (par exemple, Salesforce.com) fournis par des fournisseurs de services cloud (par exemple, Amazon, Google, Microsoft), des fournisseurs de services tiers et des fournisseurs de services et d'applications à faible coût avec un mode à la demande à tout moment et de n'importe où. Certains services/applications largement utilisés et qui sont basés sur le cloud incluent iCloud d'Apple, Apple Siri, YouTube, Dropbox, Gmail, Office 365 à titre d'exemples. À part ces nombreux avantages, le cloud computing pose plusieurs défis, notamment la latence du réseau étendu (WAN), la réponse en temps réel, la location multiple, la confidentialité et la sécurité, entre autres.

La prochaine évolution du cloud computing est l'intégration de ce dernier dans l'environnement mobile qui a donné la naissance à ce que l'on appelle le paradigme du cloud computing mobile (MCC), où les appareils mobiles accèdent sans fil aux services cloud et/ou au traitement et au stockage de données au-delà de leurs propres capacités sont transférés vers une puissante infrastructure de cloud computing centralisée. En fait, le MCC hérite de toutes les caractéristiques du cloud computing, y compris l'évolutivité, La fiabilité, l'approvisionnement dynamique, la réduction des coûts et la facilité d'intégration [14]. Le MCC permet aux périphériques mobiles à ressources limitées d'exécuter des tâches de calcul dans des infrastructures du cloud au moyen d'un déchargement de calcul, ce qui minimise la consommation d'énergie d'un appareil mobile, tout en utilisant l'appareil mobile en tant qu'un client léger. Plus précisément, la facilité d'utilisation des terminaux mobiles peut être élargie au-delà de leurs limites physiques et leurs intervalles de charge de la batterie peuvent être considérablement prolongés au moyen d'un déchargement de calcul. Cependant, toutes les applications (par exemple, les applications d'assistance cognitive, la détection de visage, les analyses vidéo en temps réel) ne sont pas adaptées pour fonctionner dans l'infrastructure centralisée de cloud computing à cause de la latence de la WAN et potentiellement la faible bande passante. Certaines applications nécessitent une réponse en temps réel, une faible latence et une bande passante élevée. Apporter des capacités qui ressemblent aux fonctionnalités du cloud (par exemple, cloudlets [15, 16]) au bord du réseau pour répondre à ces exigences est un concept émergent.

Les cloudlets diffèrent du cloud conventionnel du fait qu'ils soient des entités décentralisées qui sont situées au bord de l'internet et qui lui sont bien connectées (par exemple, via les fibres optiques), à un seul saut sans fil des périphériques mobiles associées, permettant le support de nouvelles applications à la fois pour des calculs intensifs et sensibles à la latence comme par exemple, l'assistance cognitive, la réalité augmentée (AR), ou la reconnaissance de visage et la navigation

pour les robotiques émergents des clouds [17]. Le déploiement de cloudlet n'est pas limité aux points d'accès WiFi (APs). Ce pendant, selon une étude très récente dans [18], les latences de LTE sont légèrement plus élevés comparativement aux résultats du WiFi pour le téléchargement de calcul pour les applications considérées comme par exemple, la reconnaissance de visage, la réalité augmentée et l'infographie basée sur la physique entre autres. Les Cloudlets peuvent s'appuyer sur les réseaux R&F et doivent être déployés au niveau des bords des réseaux FiWi, de sorte que l'unité du réseau optique intégrée (ONU)-AP ou le point portique de l'ONU-mesh (MPP) est appelée bord de réseaux FiWi, pour fournir une faible latence et une bande passante élevée. Ce pendant, la conception des réseaux intégrés cloud/cloudlet FiWi est difficile car il est essentiel de veiller à ce que le trafic actuel de l'homme à l'homme (H2H) (par exemple, la voix, la vidéo, les données) et/ou le trafic du cloud ne soit pas compromis par d'autres trafics coexistants. Il convient de mentionner qu'aucune étude existante traite la coexistence cloud/cloudlet ainsi que de la coexistence du trafic cloudlet et H2H dans les réseaux FiWi.

Objectifs

Les objectifs de cette thèse sont les suivants:

- Compte tenu de l'importance primordiale de l'efficacité énergétique dans les réseaux d'accès filaires et sans fil, la conception et l'implémentation des mécanismes d'économie d'énergie pour prolonger la durée de vie des batteries des périphériques à batterie tout en réduisant la consommation d'énergie du réseau optique backhaul constituent un problème important à étudier. Notez que le backhaul optique et les réseaux sans fil/cellulaires ne sont pas efficaces en terme d'énergie car leur faible utilisation moyenne entraîne un gaspillage important d'énergie [12]. Par conséquent, des mécanismes d'économie d'énergie de bout-en-bout qui prennent en compte les segments fibre et sans fil sont très nécessaires pour les réseaux d'accès FiWi. Plus précisément, le premier objectif de cette thèse est de concevoir un nouveau système de gestion des ressources de manière coordonnée et synchronisée afin de minimiser la consommation totale d'énergie dans les réseaux d'accès à large bande FiWi tout en ne dégradant pas les performances du réseau. L'objectif de cette étude est de trouver des informations sur l'économie d'énergie optimale obtenue à partir de la programmation en mode veille pour une configuration d'un réseau donnée et d'une contrainte de délai.

- Même si les emplacements multiples permettent aux fournisseurs des services cloud de mieux utiliser les ressources de calculs, de soutenir le développement des services plus flexibles basés sur l'économie d'échelle et de réduire les coûts d'infrastructure, la façon de réaliser efficacement ceci est une question fondamentale. Par exemple, le cloud computing à emplacements multiples pose des défis uniques tels que l'évolutivité, le provisionnement de ressources et la personnalisation des services par locataire. Par conséquent, le deuxième objectif de cette thèse est de concevoir un système de gestion de ressources comprenant à la fois une architecture et une nouvelle politique d'ordonnancement en ce qui concerne les problèmes de emplacements multiples, notamment l'évolutivité et les ressources partagées pour planifier des applications de flux de travail à calculs intensifs.
- Le déchargement des données mobiles n'a pas été entièrement exploré du point de vue des réseaux d'accès FiWi, y compris le cloud computing mobile et le partage de backhaul à fibre dans le scénario des réseaux hétérogènes (HetNets). Le troisième objectif de cette thèse consiste à élaborer un système mobile de déchargement des données, concevant ainsi des stratégies économiques optimales (par exemple, en minimisant la consommation d'énergie et en maximisant les gains).
- Le quatrième objectif de cette thèse est de concevoir les clouds et les cloudlets améliorés dans les réseaux d'accès FiWi, et ainsi contribuant à la réalisation du concept émergent du mobile computing à bord (MEC). Il est important de noter qu'aucune étude existante traite la coexistence cloud/MEC ainsi que la coexistence du trafic MEC et H2H (par exemple, le trafic triple-play) dans les réseaux FiWi. D'autre part, il existe des problèmes de fiabilité du backhaul à fibre optique (par exemple, des coupures de fibre) et du serveur MEC (par exemple, défaut au sein de la plate-forme MEC, échecs de liaison vers/depuis un point d'accès). Par conséquent, le réseau envisagé peut souffrir de larges interruptions de service. Le premier but de cette partie du doctorat est de concevoir l'architecture et de développer un programme unifié de gestion des ressources (une solution pour les deux segments de réseau pour simplifier la gestion de l'architecture intégrée) ainsi que d'étudier ses gains performance achevés.
- L'objectif final de ce travail est d'explorer les nouveaux scénarios de déploiement des réseaux d'accès FiWi améliorés cloud/cloudlet. Plus précisément, cet objectif est d'élaborer le concept de l'internet tactile émergent et d'étudier la faisabilité des réseaux FiWi basés à l'internet tactile.

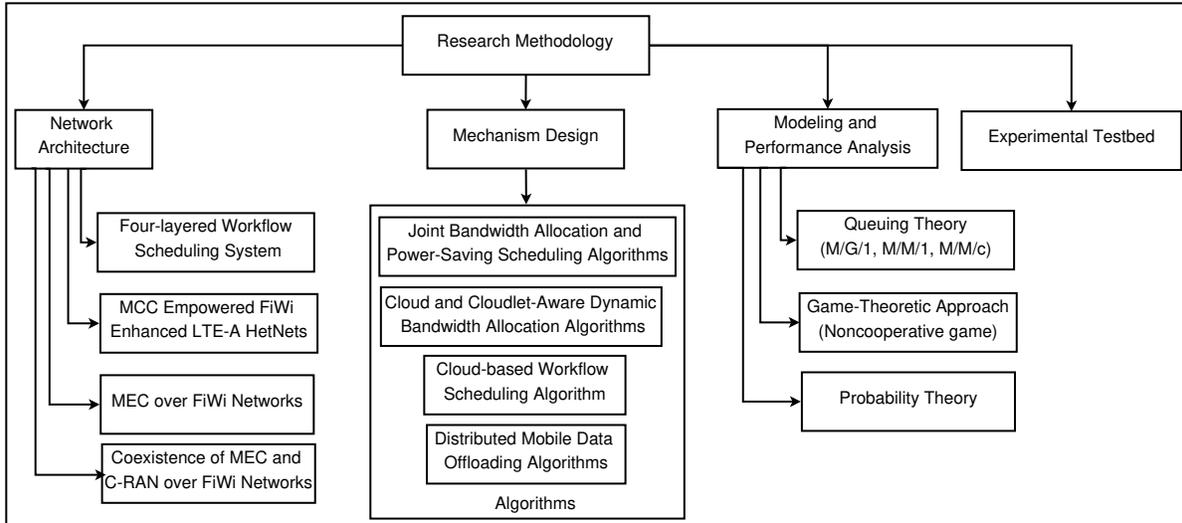


Figure R.1: Méthodologie de recherche.

Méthodologie de recherche

La méthodologie de recherche globale, y compris l'architecture, la conception des mécanismes, la modélisation et l'analyse des performances appliquées dans cette thèse est résumée dans la Fig. R.1 et leur brève description est présentée ci-dessous.

- *Architecture*: Des architectures novatrices sont développées pour le système d'ordonnancement du flux de travail, le déchargement des données mobiles et les réseaux FiWi à haut débit améliorés par le cloud-cloudlet/MEC. Une approche descendante, qui est une stratégie commune pour la conception de réseau, est utilisée pour concevoir les réseaux FiWi améliorés. Dans cette approche, les applications réseaux et les exigences de service sont d'abord identifiées, puis le réseau est conçu pour les prendre en considération. Plus spécifiquement, la fonctionnalité des technologies de réseau et des protocoles existants ainsi que les algorithmes de la gestion des ressources (par exemple, l'allocation dynamique de la bande passante (DBA)) sont exploitées. Les topologies maillées et les topologies d'arbres conventionnelles sont considérées dans la partie frontale des réseaux FiWi.
- *Conception du mécanisme*: De nouveaux algorithmes sont développés pour exploiter les architectures proposées. Ils incluent des algorithmes d'allocation conjointe de la bande passante et d'ordonnancement d'économie d'énergie, l'ordonnancement du flux de travail basé sur le cloud,

les algorithmes de déchargement des données mobiles distribués et finalement les algorithmes DBAs basés sur le cloud et le cloudlet.

- *Modélisation et analyse des performances:* Parmi les divers outils analytiques, la théorie des files d'attente a été largement utilisée pour l'analyse des délais dans les réseaux FiWi et dans le cloud computing. Par conséquent, une analyse complète basée sur la théorie des files d'attente (par exemple M/G/1, M/M/1, M/M/c) a été menée. Grâce à des analyses et à des simulations complètes, la performance des solutions proposées a été étudiée sous différentes configurations de réseaux. Motivé par le fait que de nos jours la théorie des jeux est devenue un outil mathématique commun dans la conception et l'analyse des communications filaires/sans fil, le cadre logiciel mobile du déchargement des données développés dans cette thèse repose sur une approche de Stackelberg qui est à la fois à niveaux multiples et non coopérative.
- *Testbed expérimental:* Un testbed expérimental pour le MEC dans les réseaux d'accès améliorés cloudlet du FiWi est développé pour valider la précision du modèle analytique via des mesures du monde réel.

Les contributions de la thèse

Cette dissertation est une compilation de sept publications scientifiques (six articles de revues et un article de conférence) qui ont été publiées dans de prestigieuses revues de l'IEEE et des une conférence de haute qualité. Les principales contributions de cette thèse qui répondent aux objectifs mentionnés dans la section sont discutées en détail dans ce qui suit.

Systeme de gestion des ressources axé sur l'énergie pour les réseaux d'accès FiWi

La plupart des études antérieures sur les réseaux d'accès FiWi ont mis l'accent sur la conception de l'architecture, la survie et la qualité de service (QoS) garantie, ainsi que leur rôle dans d'autres infrastructures essentielles, comme les réseaux intelligents [19]. Toutefois, l'aspect de l'efficacité énergétique des réseaux d'accès FiWi, qui est d'une importance primordiale non seulement pour prolonger la durée de vie de la batterie des périphériques sans fil mais aussi pour réduire l'OPEX pour les opérateurs de réseau, n'a pas été étudié profondément jusqu'à à présent. Entre-temps, les solutions d'efficacité énergétique pour les réseaux backhaul ont été largement étudiées puisque

les réseaux d'accès sont connus par leur consommation excessive d'énergie afin d'accommoder les charges réelles du trafic. Il a été montré que les réseaux de backhaul pouvaient représenter jusqu'à 50% de la consommation totale des réseaux sans fil [20]. Néanmoins, les dispositifs de réseau d'accès (connu sous le nom ONU) sont fortement sous-utilisés et donc ils sont inactifs la plupart de leur temps [21], résultant une énorme quantité d'énergie gaspillée. De plus, il est souhaitable de consolider et d'harmoniser les stratégies d'efficacité énergétique avec les mécanismes d'allocation des ressources en prenant en compte à la fois le backhaul optique et les segments frontaux sans fil d'un réseau d'accès FiWi afin d'atteindre un rendement énergétique élevé tout en garantissant aux utilisateurs finaux une expérience satisfaisante en terme de QoS.

Ce travail propose un système de conservation d'énergie pour les réseaux FiWi (ECO-FiWi) qui planifie conjointement les modes d'économie d'énergie des stations sans fil et des points d'accès et des unités du réseau optique pour réduire leur consommation d'énergie. ECO-FiWi maximise la performance globale du réseau en exploitant l'accès multiple par répartition dans le temps (TDMA) pour synchroniser les modes d'économie d'énergie et les intégrer dans le processus d'allocation dynamique de la bande passante (DBA). Un modèle complet d'économie d'énergie et une analyse en file d'attente M/G/1 des retards de trame descendant et ascendant sont présentés pour les segments frontal du réseau backhaul.

Les principales contributions de ce travail sont les suivantes. Premièrement, pour la première fois, afin de maximiser l'efficacité énergétique globale sans compromettre la performance en délais des réseaux d'accès FiWi intégrés, les modes d'économie d'énergie de tous les ONU, APs et STAs sont synchronisés et incorporés dans le processus (EPON) DBA. ECO-FiWi prenant en compte l'état du trafic des transmissions ascendantes (US) et descendantes (DS) dans le processus conjoint d'allocation de la bande passante et de l'ordonnancement du mode d'économie d'énergie pour les éléments du réseau FiWi afin de rendre leur temps d'activité proportionnel à leurs charges réelles.

Deuxièmement, un modèle d'économie d'énergie est développé pour fournir aux opérateurs une vue d'ensemble des paramètres et des quantités qui déterminent l'économie d'énergie potentielle pour une configuration de réseau donnée. Troisièmement, pour évaluer la performance du réseau, une analyse complète des délais moyens des trames US et DS basés sur un système de files d'attente M/G/1 est présentée, en tenant compte des segments frontal des réseaux sans fil et backhaul. Outre que les délais moyens, les limites supérieures des délais sont également analysées pour fournir une estimation des délais maximums dans les réseaux ECO-FiWi. Il est important de noter que

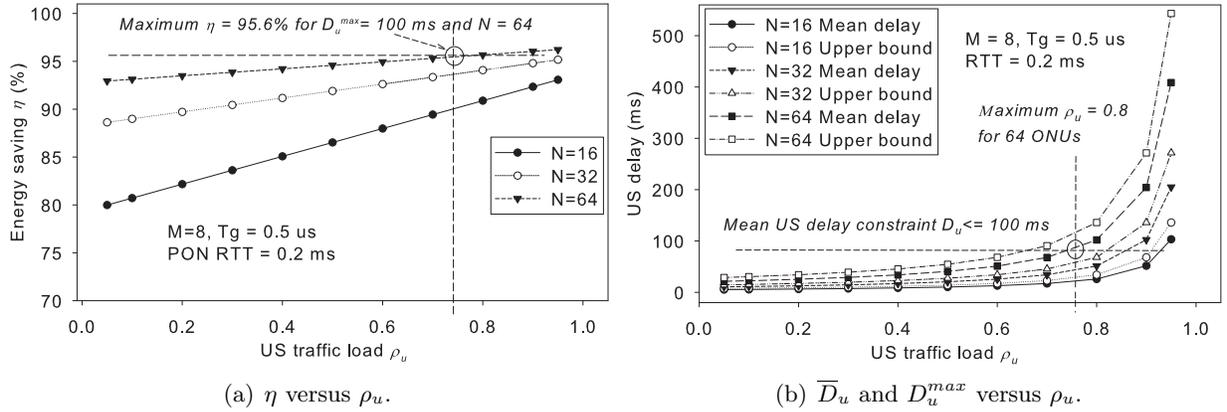


Figure R.2: Économie d'énergie et délais des trames (\bar{D}_u et D_u^{max}) US en fonction de la charge du trafic US ρ_u .

la combinaison du modèle d'économie d'énergie et l'analyse des délais de trame fournissent des informations précieuses sur la quantification de l'économie d'énergie optimale pour une configuration de réseau et une contrainte de délai donnée.

Les résultats obtenus montrent qu'il existe un compromis clair entre l'économie d'énergie et le délai des trames. Une grande valeur du temps de cycle de sondage (T_c) permet d'obtenir des économies d'énergie plus élevées (η), même si elle entraîne un délai de trame plus long. Fig. Fig. R.2(a) révèle que η augmente linéairement avec (ρ_u), ce qui signifie que l'économie d'énergie obtenue à partir du déploiement d'ECO-FiWi croît linéairement avec la charge réelle du trafic. Il est observé que pour toute charge de données US, plus de 80% d'énergie peut être économisée en utilisant ECO-FiWi. De plus, pour une charge de trafic donnée, plus le nombre d'ONUs est élevé, plus l'économie d'énergie devient significative. Cela découle du fait que plus d'ONUs impliquent que chaque ONU a un intervalle de temps plus petit, donc un temps d'inactivité plus long en dehors de son intervalle de temps. De plus, lorsque le nombre des ONUs N augmente, la pente de la fonction linéaire est plus petite. On observe également que lorsque ρ_u est grand, η a tendance à converger. En effet, lorsque ρ_u s'approche de 1, T_c tend vers l'infini.

Fig. R.2(b) montre que les deux délais moyens et maximums des trames US haussent en augmentant ρ_u . Le délai moyen US est d'environ trois quarts du délai maximum US. En plus, le délai des trames US reste relativement faible lorsque la charge du trafic est faible, mais augmente rapidement lorsque la charge de trafic augmente. Par exemple, pour $\rho_u = 0,6$, une trame US a un délai moyen inférieur à 50 ms même pour un système de 64 ONUs, ce qui est acceptable pour de

nombreuses applications en temps réel. Mais pour $\rho_u = 0,8$, le délai moyen US est d'environ 100 ms pour la même configuration du système. En outre, plus le nombre d'ONUs dans le système est élevé, plus les délais US augmentent. Il convient de rappeler que le délai de la trame US provient de la nature TDMA des réseaux ECO-FiWi plutôt que des modes d'économie d'énergie mis en oeuvre. Indépendamment du fait que les STAs et les ONU-APs entrent en mode de veille ou non, une trame US connaîtra le même délai pour une configuration de réseau donnée.

Consolider les économies d'énergie de la Fig. R.2(a) avec le délai US dans la Fig. R.2(b) permet aux opérateurs de trouver un compromis optimal entre le délai et l'énergie. Par exemple, lorsque le délai moyen restreint de trame US est de 100 ms, la Fig. R.2(b) permet d'indiquer la charge maximale du trafic US possible pour garantir des exigences données de QoS. Plus précisément, avec 8 stations sans fil (STAs) par ONU-AP et 10 km de portée de réseau, un système de 64 ONU-APs doit avoir un ρ_u ne dépassant pas 0,8, alors que cette dernière peut atteindre 0,9 pour un système de 32 ONU-APs. En utilisant ces valeurs de ρ_u pour faire la projection sur la courbe d'économie d'énergie de la Fig. R.2(a), les opérateurs peuvent déduire que 95,6% de l'énergie peut être sauvegardé pour un système de 64 ONU-APs sans violer les contraintes données du délai.

Ordonnancement du flux de travail dans le cloud computing à locataires multiples

La multi-localisation est l'une des principales caractéristiques du cloud computing. Même si la multi-localisation permet aux fournisseurs de services cloud de mieux utiliser les ressources de calculs, de soutenir le développement de services plus flexibles basés sur l'économie d'échelle et de réduire les coûts d'infrastructure, la question sur la façon de les réaliser efficacement est fondamentale. Par exemple, le cloud computing à emplacements multiples pose des défis uniques tels que l'évolutivité, le provisionnement de ressources (par exemple, répondre à la demande de grands volumes de locataires par ressource) et la personnalisation (personnalisation du service par locataire) [22, 23]. De plus, les applications multi-locataires doivent être dynamiques ou polymorphes pour répondre aux attentes individuelles des différents locataires et de leurs utilisateurs [24]. Cette thèse examine un cadre de gestion des ressources qui comprend des politiques d'architecture et d'ordonnancement en ce qui concerne les problèmes de multi-localisation, en particulier l'évolutivité et les ressources partagées pour planifier des applications de flux de travail intensives en calcul. De plus, un cadre de gestion des ressources est élaboré dont l'objectif est de séparer les politiques de gestion des ressources des mécanismes de contrôle nécessaires pour les mettre en oeuvre. Il est à noter qu'il y a plusieurs

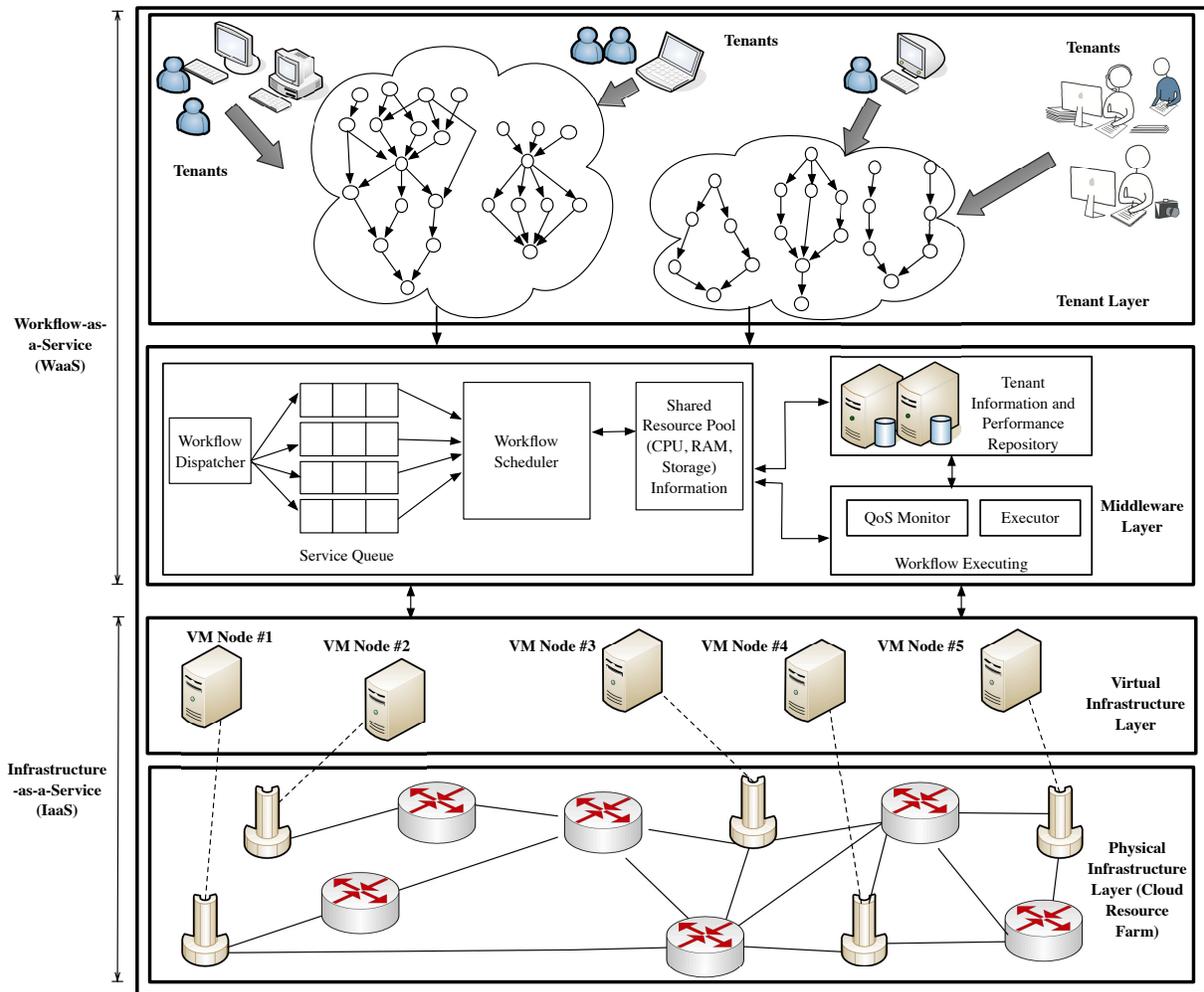


Figure R.3: Architecture de référence proposée pour la planification des flux de travail dans un environnement de cloud multi-locataire.

conséquences nuisibles pour les fournisseurs de services et les locataires comme les performances imprévisibles des applications, l'applicabilité limitée au cloud et les inefficacités dans les centres de données [25]. Cependant, ces questions dépassent le cadre de cette thèse.

Compte tenu de l'importance vitale de la multi-localisation (par exemple, les aspects d'isolement et de personnalisation dans les infrastructures partagées), ce chapitre propose un système de gestion des ressources qui consiste en une nouvelle architecture à quatre couches d'un système d'ordonnancement du flux de travail (voir Fig. R.3) et d'une de d'ordonnancement. L'architecture proposée et l'algorithme de planification de flux de travail basé sur le cloud (CWSA) fournissent la multi-localisation en unissant les abstractions - flux de travail, ressources et mécanisme de contrôle (ordonnancement) - qui permettent des politiques logiquement centralisées. En plus, l'architecture permet une abstraction des ressources qui unifie des ressources arbitraires, telles que le stock-

age, le réseau, le CPU et les groupements, ce qui permet l'agnosticité de ressources. Par ailleurs, l'architecture envisagée fournit un environnement de service et de gestion pour permettre à plusieurs locataires d'exécuter leurs applications intensives en calcul sur une infrastructure de cloud partagée tout en profitant de l'élasticité et la facturation payante du modèle pay-asyou-go du cloud computing.

Les principales contributions de ce travail sont les suivantes. Premièrement, d'un point de vue architectural, nous proposons une nouvelle architecture à quatre couches du système d'ordonnancement du flux de travail dans un environnement de cloud computing à locataires multiples, qui représente une solution rentable pour l'exécution d'applications intensives en calcul du flux de travail. Deuxièmement, d'un point de vue *de gestion des ressources*, nous proposons un algorithme CWSA pour gérer l'ordonnancement de flux de travail structurée et non structurée dans les environnements cloud computing à locataires multiples, ce qui maximise l'utilisation des ressources cloud, minimise le temps d'exécution global (Le temps global d'exécution du flux de travail), et minimise aussi le temps d'exécution de l'ordonnancement, les coûts d'exécution du flux de travail et le délai attendu total (pénalité de délai, si une tâche dans le flux de travail est terminée après son temps dû). Troisièmement, nous effectuons des *études complètes basées sur la simulation* de la politique d'ordonnancement proposée et évaluons différents ensembles de mesures du rendement, y compris l'analyse statistique, pour démontrer la robustesse de la politique d'ordonnancement proposée. Nous comparons ensuite les performances de la politique de la CWSA avec les politiques premier arrivé premier servi (FCFS), EASY (système extensible d'ordonnancement d'Argonne) et le temps d'achèvement minimum (MCT). Il est à noter qu'il n'y a pas de consensus sur les métriques largement acceptées pour mesurer la robustesse de la politique d'ordonnancement adoptée. Ainsi, dans ce travail, nous proposons d'utiliser l'écart type du temps d'exécution, l'asymétrie de ce dernier et la répartition du percentile 99^e du comparateur pour comparer les métriques de robustesse. Quatrièmement, à partir d'une *perspective de démonstration du concept*, la politique d'ordonnancement CWSA est davantage mise en oeuvre dans des applications de flux de travail complexes, scientifiques et en temps réel (par exemple, SIPHT (protocole d'identification de l'sRNA utilisant une technologie à haut débit) et CyberShake - références populaires qui ont été largement utilisées dans les études du flux de travail) pour démontrer l'évolutivité et l'efficacité de la solution proposée.

Réseaux informatiques mobile-edge sans fil - à fibres dans la 5G

Contrairement au cloud centralisé conventionnel, les clouds locaux (cloudlets) poussent la frontière du calcul éloigné des noeuds centraux vers le bord du réseau pour améliorer la disponibilité et l'accessibilité des services cloud, tout en minimisant les latences du réseau étendu (WAN) [16]. À la lumière de cela, le soit-disant MEC [26] a récemment émergé pour offrir des capacités cloud (par exemple, calcul, stockage et mise en cache) au bord des réseaux à proximité des appareils mobiles, enrichissant ainsi l'expérience mobile des utilisateurs à haut débit. Le MEC transforme les stations de base (par exemple 3G, 4G) en des centres de services intelligents capables de fournir des services hautement personnalisés. En outre, le MEC fournit au backhaul des informations en temps réel sur le réseau d'accès radio (RAN) et les besoins en trafic et facilite ainsi la coordination entre les segments backhaul et RAN, ce qui n'a pas encore été pleinement réalisé [27]. Une telle coordination est nécessaire lorsque, par exemple, les réseaux radio ont besoin de moins de bande passante, mais le backhaul n'en est pas au courant et vice versa. D'un point de vue commercial, l'émergence du MEC permet aux opérateurs de réseaux, aux éditeurs indépendants de logiciels et aux fournisseurs de services et de contenus Web de créer de nouvelles chaînes de valeur [26].

En offrant des fonctionnalités attrayantes telles que la latence extrêmement faible et le débit élevé que nécessitent les réseaux 5G, le MEC crée un chemin vers la 5G. Par exemple, une classe d'applications critique 5G appelée internet des objets IoT et l'internet tactile [28], qui exigent une latence extrêmement faible et une fiabilité de classe opérateur (disponibilité de 99,999%), devraient s'appuyer sur le MEC [26]. Par conséquent, le MEC est reconnu par l'organisme européen 5G PPP (5G Infrastructure Public Private Partnership) comme l'une des principales technologies émergentes pour les réseaux 5G [26]. De plus, le MEC devrait fournir une approche standard pour des progrès significatifs vers la 5G, avec le soutien des leaders de l'industrie (par exemple, Intel, Nokia, Huawei et Vodafone) qui participent dans le groupe de spécifications de l'industrie MEC de l'ETSI (ISG). Compte tenu de l'élan croissant sur le MEC, ce chapitre vise à enquêter le MEC dans le contexte de 5G.

Fig. R.4 fournit un aperçu des scénarios de déploiement possibles du MEC avec le cloud computing dans les réseaux 5G, y compris le backhaul optique (par exemple, le réseau optique passif Ethernet (EPON) et le 10G-EPON) et les technologies de backhaul sans fil (par exemple, ondes millimétriques, sous-6 GHz sans licence/avec licence). Comme le montre la Fig. R.4, les serveurs MEC peuvent être déployés dans divers scénarios tels que des points d'accès WiFi, des stations de

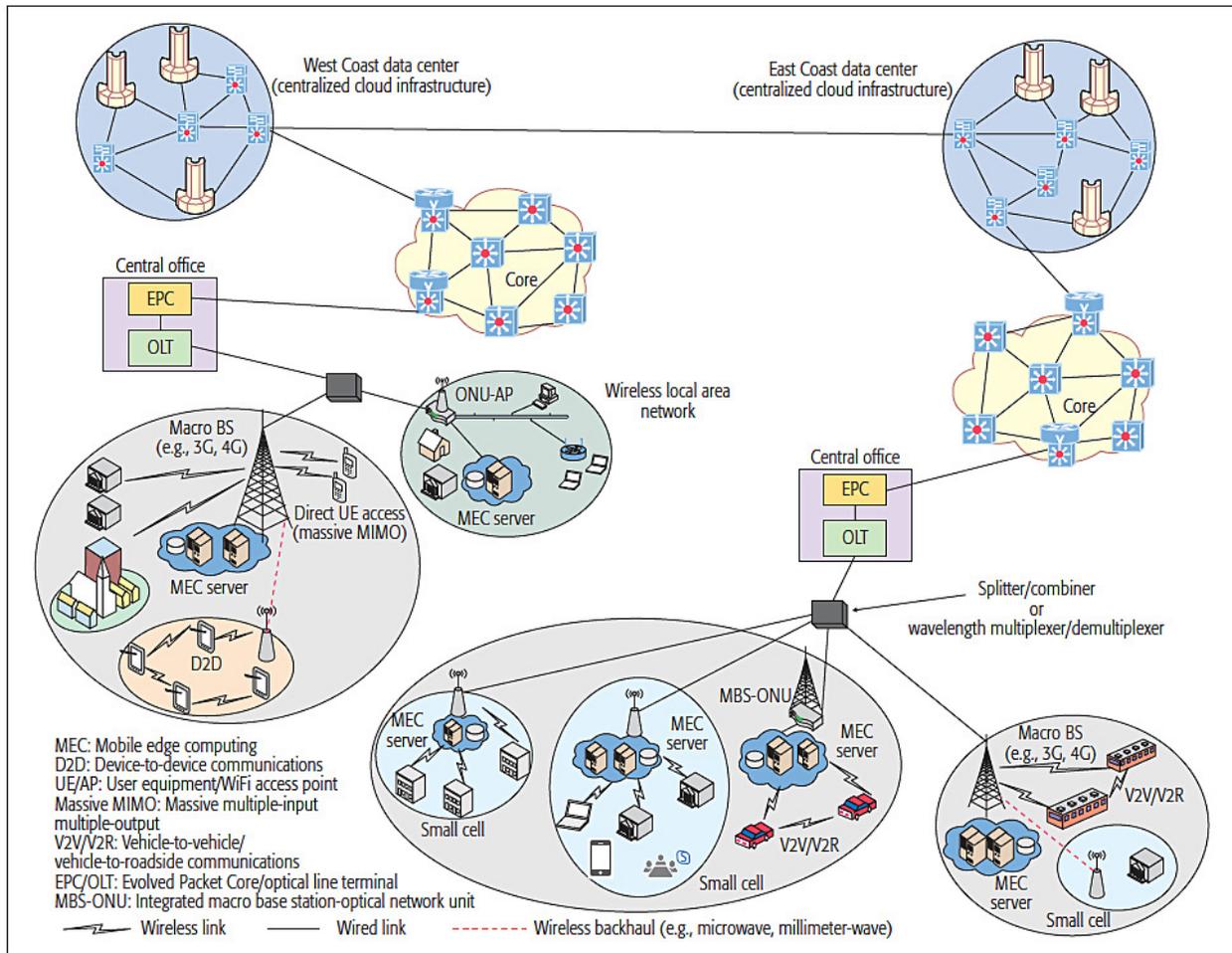


Figure R.4: Illustration de l'évaluation du mobile-edge et de la cloudification des réseaux 5G.

macro base cellulaires (par exemple, 3G, 4G/LTE-A), des petits points d'agrégation de cellules et un bureau central en coexistence avec des clouds conventionnels centralisés.

Récemment, les réseaux FiWi ont été intégrés aux réseaux hétérogènes LTE-Advanced (Het-Nets) [29], pour soutenir davantage le cloud computing conventionnel et les cloudlets [30]. En raison de leurs traits saillants, dans ce travail, les réseaux FiWi sont encore améliorés avec les capacités du MEC. Les serveurs MEC sont intégrés au bord des réseaux FiWi, c'est-à-dire des points d'accès ou des stations de base (BS) arrangés avec des unités de réseau optique (ONUs). Puisque le MEC est considérée comme l'une des principales technologies émergentes pour les réseaux 5G [26], le concept "MEC sur FiWi" introduit doit non seulement aborder les défis de la 5G susmentionnés, mais aussi ses propres et uniques défis. Parmi ces défis, l'intégration du MEC avec les infrastructures existantes du réseau (filaires et sans fil), la coexistence entre les cloudlets et les clouds convention-

nels, et une meilleure gestion des ressources en tenant compte de la coordination backhaul/RAN sont d'une importance primordiale.

La recherche dans le domaine du MEC est encore à ses débuts. Même s'il existe un intérêt croissant pour le MEC, tant du milieu universitaire que de l'industrie, aux meilleures connaissances des auteurs, les scénarios du MEC sur FiWi et le schéma unifié de la gestion des ressources présentés ici sont les premiers qui prennent l'intégration du réseau, la coexistence H2H/MEC et le problème de gestion des ressources en considération. Ce travail vise à fournir une étude complète sur la conception du MEC sur les réseaux FiWi. Il développe d'abord le concept du MEC en étudiant les scénarios typiques de service. Les principaux défis techniques liés à la mise en oeuvre du MEC sont ensuite identifiés et discutés en détails. Étant donné que le WLAN et le 4G LTE/LTE-A figurent parmi les technologies RAN les plus courantes, trois scénarios typiques de conception prévisibles sont envisagés: a) MEC sur FiWi-WLAN, b) MEC sur FiWi-4G LTE, et c) La coexistence de MEC et C-RAN dans le FiWi à base LTE-A HetNets. Afin d'examiner les gains réalisables du réseau, ce chapitre propose pour la première fois un nouveau plan unifié de gestion des ressources qui alloue conjointement la bande passante pour les transmissions à la fois du trafic classique à haut débit et les données MEC en mode TDMA.

Les résultats obtenus montrent les avantages significatifs du MEC sur les réseaux FiWi. Fig. R.5 (a) représente l'efficacité maximale en terme du temps de réponse pouvant être atteinte pour différentes tailles de paquets de déchargement. Pour augmenter la taille des paquets de déchargement, l'efficacité moyenne du temps de réponse s'approche de façon asymptotique de 100%. Par exemple, pour un cas typique de 32 ONU-APs et une taille de paquet de déchargement de 825,60 KB, le temps de réponse moyen est respectivement de 77,10% et 95,36% pour les deux WLANs considérés. Cela se traduit par une réduction du temps de réponse de 77,10% et 95,36% par rapport au temps de réponse obtenu dans un scénario sans déchargement. Fig. R.5(a) révèle également que le MEC sur le FiWi à très haut débit (VHT) basé sur le WLAN contribue à augmenter de manière significative l'efficacité maximale du temps de réponse réalisable. Cet ensemble de résultats valide que le "MEC sur FiWi" basé sur le déchargement de calcul est une solution prometteuse pour améliorer la qualité de l'expérience (QoE) des utilisateurs à l'appui d'une large gamme d'applications futures 5G.

Fig. R.5(b) illustre les performances du délai des paquets du trafic coexistant entre H2H et MEC sous diverses charges de trafic H2H. Les deux courbes ont un comportement similaire. Ce pendant, le délai du paquet de déchargement est plus élevé que le délai de H2H car le trafic déchargé attend

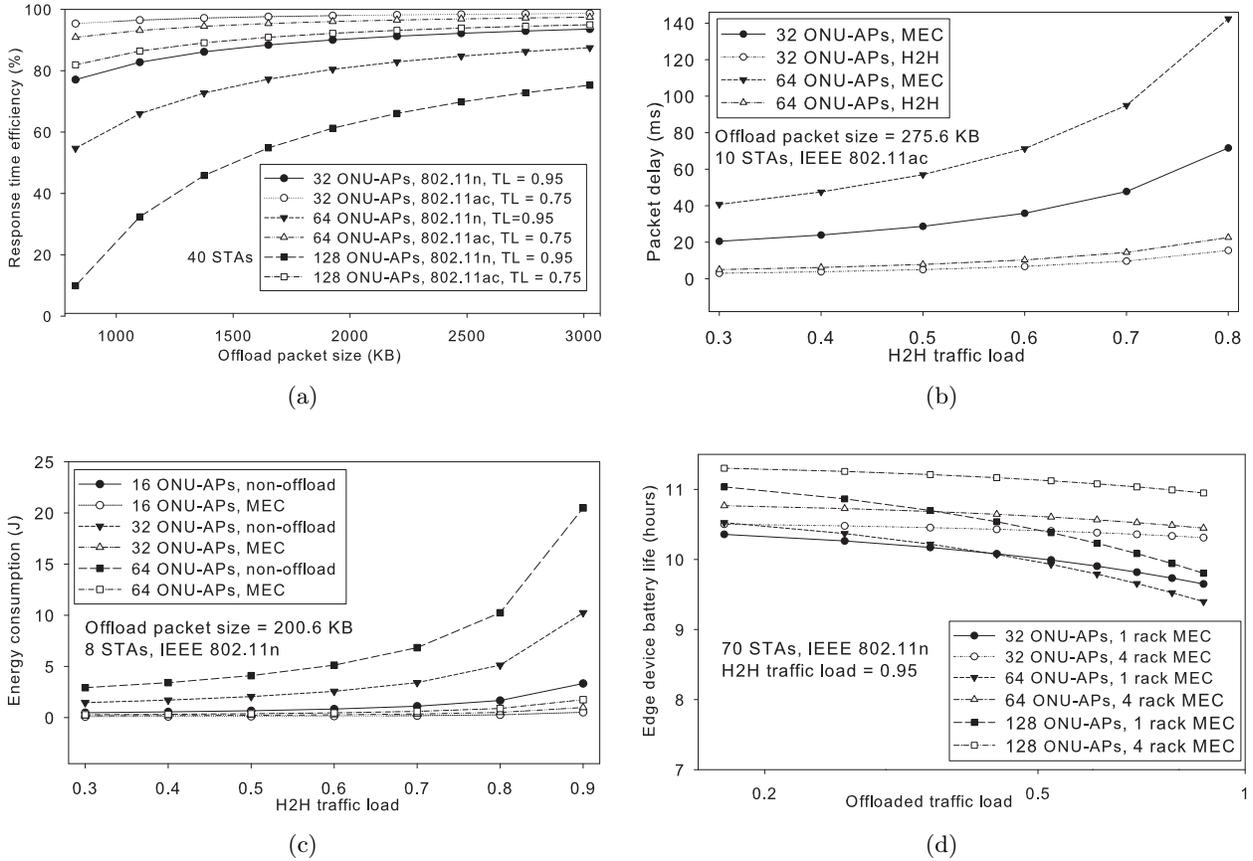


Figure R.5: a) Efficacité du temps de réponse vs. Taille du paquet de déchargement; b) Délai moyen des paquets du trafic coexistant entre H2H et MEC; c) Consommation d'énergie contre la charge du trafic H2H pour les scénarios MEC et non-MEC; d) Durée de vie de la batterie des périphériques vs. Charge du trafic déchargé. (TL: Charge de trafic H2H).

plus longtemps avant d'être transmis en raison d'une plus longue durée de cycle d'interrogation PON. Un retard moyen de paquet de H2H inférieur à 23 ms est atteint pour toutes les valeurs de charges du trafic H2H. Il est important de noter que, pour un scénario typique de 32 APs ONU et une charge de trafic H2H de 0,3, on obtient un délai moyen de paquet de MEC de 20,50 ms. De plus, même avec une charge de trafic H2H de 0,7, le délai MEC reste inférieur à 95 ms, ce qui n'est pas réalisable avec des clouds centralisés typiques. Cela signifie que de nombreuses applications sensibles au délai peuvent être déchargées sur le réseau "MEC sur FiWi" basé sur Ethernet. Cependant, comme le montre la Fig. R.5(b), le déchargement peut ne pas être efficace lorsque la charge du trafic H2H est supérieure à 0,8 en raison d'un délai significativement plus long.

Fig. R.5(c) compare la consommation moyenne de l'énergie entre le scénario MEC et le scénario d'exécution locale (c'est-à-dire non déchargé). La consommation d'énergie est une fonction de la

charge du trafic H2H, le temps de cycle d'interrogation et le nombre d'APs ONU et STAs. Il est démontré sur la Fig. R.5(c) que le MEC au déchargement basé sur FiWi réduit considérablement la consommation d'énergie des dispositifs qui se trouvent bord du réseau. Par exemple, pour 32 APs ONU et une charge de trafic H2H de 0,8, la consommation de l'énergie dans le scénario MEC est 10,13 fois moins importante que dans le scénario de non-déchargement. Finalement, la Fig. R.5(d) montre la durée de vie de la batterie des périphériques à bord en fonction de la charge du trafic déchargé et du nombre de serveurs MEC. C'est remarquable qu'en employant le programme unifié de gestion des ressources proposé et l'ordonnancement en mode en veille, on peut obtenir jusqu'à 11.035 heures et 11.302 heures d'autonomie, respectivement, avec un serveur MEC à un et à quatre créneaux. Ceci vérifie qu'un réseau MEC sur FiWi basé sur Ethernet avec le plan proposé aide à prolonger la durée de vie de la batterie des périphériques de bord de manière significative. Il est à noter qu'en raison de la nature du TDMA du programme proposé, l'utilisation des liaisons backhaul avec une capacité supérieure (par exemple 10G-EPON) dans le réseau MEC sur FiWi proposé n'aurait pas d'incidence sur les résultats obtenus.

Le Mobile-edge computing vs. le cloud computing centralisé sur un réseau d'accès convergé à haut débit sans fil à fibre optique

Avec l'arrivée d'une grande variété de dispositifs intelligents, l'IoT et les applications émergentes 5G, d'énormes volumes de données seront générés au bord du réseau. Ces données peuvent être transmises pour le stockage et le traitement sur des clouds conventionnels centralisés même si la majorité des données peuvent être pré-traitées au bord du réseau [31]. Un cloud conventionnel fait face à plusieurs défis tels que le point de défaillance unique, le manque de sensibilisation à la localisation, l'accessibilité, et les latences du WAN, entre autres [13]. Les cloudlets ont été envisagés pour supporter des applications nécessitant beaucoup de ressources en fournissant un accès à des ressources de calcul et de stockage pour des dispositifs mobiles, par exemple des dispositifs à batterie [15]. Le MEC a pour but de fournir un accès aux capacités de calcul du cloud, de stockage et de réseautage cloud, ainsi que l'environnement de service IT au niveau du réseau dans le RAN, à proximité des abonnés. De plus, le MEC est prometteur d'ouvrir de nouvelles frontières pour les opérateurs, les applications, les fournisseurs de services et de contenu, ce qui leur permet de déployer de façon flexible des services innovants et perturbateurs dans le domaine d'IoT, des environnements immersifs et des grandes analyses de données, entre autres.

Un cloud conventionnel centralisé possède des capacités de stockage et de traitement élevées, mais il présente une latence importante. Au contraire, le MEC peut offrir une latence plus faible, mais il a des capacités de calcul et de stockage limitées par rapport à un cloud conventionnel. Par conséquent, les clouds centralisés et les cloudlets décentralisés (par exemple le MEC) peuvent coexister et être complémentaires l'un de l'autre afin de soutenir un ensemble plus diversifié d'applications et de services émergents de la 5G (par exemple, tolérance aux délais, mission critique et localisateur).

Compte tenu de l'intégration du réseau filaire/sans fil et de la tendance à la décentralisation dans les réseaux 5G, les réseaux FiWi présentent une architecture prometteuse pour supporter à la fois le cloud centralisé et le MEC émergent. Un tel cloud et MEC envisagés dans le réseau FiWi est appelé un *réseau CM-FiWi* dans ce travail. Notez que les études MEC existantes, par exemple, le Groupe de spécifications industrielles du MEC, mettent l'accent sur la mise en œuvre de MEC dans le contexte de 4G LTE, alors que cet article examine les réseaux MEC sur WiFi en raison de leurs faibles coûts, de larges déploiements et de leur haute capacité (100 fois plus élevé que les réseaux cellulaires [32]).

Le réseau CM-FiWi offre plusieurs avantages. D'un point de vue économique, cela permettrait aux opérateurs mobiles de partager non seulement les infrastructures à fibre et sans fil existantes, mais également de fournir une plate-forme de communication unique à des fins multiples telles que les services cloud (MEC, cloud centralisé) et l'accès haut débit aux utilisateurs fixes et sans fil via le partage du backhaul de fibres, réduisant ainsi le CAPEX et l'OPEX. En outre, le CM-FiWi offre la possibilité de réaliser un contrôle de bout-en-bout de l'ensemble du réseau avec prise en compte du back/fronthaul. D'un point de vue technique, un seul mécanisme unifié de gestion de ressources peut être conçu pour exploiter efficacement un tel réseau hautement intégré. Cela contribue à réduire la complexité globale de la gestion du réseau (par exemple, la gestion des ressources, l'allocation de la bande passante). Ce pendant, la conception des réseaux intégrés CM-FiWi est difficile, compte tenu de la coexistence symbiotique du trafic à haut débit et des données cloud. Cette coexistence diversifie encore les caractéristiques et les exigences de la communication. Il est à noter qu'aucune étude existante ne traite la coexistence cloud/MEC ni la coexistence du trafic MEC et H2H (par exemple, le triple play, la vidéo et le trafic de données) dans les réseaux FiWi. D'autre part, il existe des problèmes de fiabilité du backhaul à fibre optique (par exemple, les coupures de fibres d'alimentation et de distribution) et du serveur MEC (défaut au sein de la plate-forme MEC,

défaillance de machine virtuelle, échecs de liaison vers/depuis le point d'accès). La survie devient de plus en plus importante tant pour les fournisseurs de services que pour les clients finaux en raison de la grande quantité de trafic transporté dans les PON TDM à grande vitesse et de leur nombre considérablement augmenté d'ONU (jusqu'à des centaines) [33]. En cas de panne de service, la perte de données serait extrêmement élevée à cause de la très grande capacité des liaisons de fibres [34]. Le réseau envisagé peut souffrir de larges interruptions de service. Beaucoup de ces occurrences donnent des défaillances de noeuds et de liens hautement corrélés (temps et espace) [35]. Ainsi, fournir un niveau acceptable de fiabilité est une question critique dans les réseaux CM-FiWi.

Ce travail vise à concevoir un système unifié de gestion des ressources pour intégrer les activités de déchargement aux opérations sous-jacentes de FiWi. L'architecture du réseau envisagé repose sur l'intégration d'Ethernet PON (EPON), de WLAN (classe Gigabit) à très haut débit (VHT) WLAN (IEEE 802.11ac) et de serveurs Cloud/MEC. Afin de faire face à la coexistence symbiotique du trafic H2H et du trafic cloud et MEC centralisés, ce travail propose d'incorporer des transmissions de déchargement entre les périphériques des dispositifs (par exemple, téléphone cellulaire, PDA, capteurs, textile intelligent) et les serveurs MEC/cloud dans le processus DBA FiWi en utilisant l'ordonnancement TDMA. Il est intéressant de noter que la TDMA est un schéma d'accès largement déployé dans EPON/10G-EPON et le WLAN [36–40] grâce à plusieurs avantages. Nous notons également que quelques études sur les réseaux FiWi, par exemple [29], considèrent la méthode d'accès (CSMA/CA) avec évitement de collision. Cependant, le MEC exige un accès déterministe afin de minimiser le délai. Par conséquent, le réseau CM-FiWi envisagé est conçu à l'aide de TDMA dynamique et du sondage.

À cette fin, les principales contributions de ce travail sont les suivantes. Premièrement, l'architecture cloud-MEC à deux niveaux est introduite dans les réseaux d'accès FiWi à haut débit. Le CM-FiWi crée non seulement une plate-forme d'infrastructure pour les opérateurs de téléphonie mobile ou les tiers fournisseurs de services pour offrir des services MEC à partir du bord des réseaux, mais représente également une solution rentable pour supporter divers types du trafic (H2H, Cloud et MEC). Deuxièmement, un nouveau système unifié de gestion des ressources est proposé pour gérer efficacement et simultanément le H2H et les deux types de trafic en cloud (cloud et MEC) en utilisant le schéma TDMA avec multi-sondage. Les trafics MEC et cloud sont tous les deux ordonnancés en dehors de l'intervalle de transmission du trafic H2H. Troisièmement, les paramètres clés de performance pour évaluer le MEC sont définis, qui sont essentiels à la compréhension de l'interaction

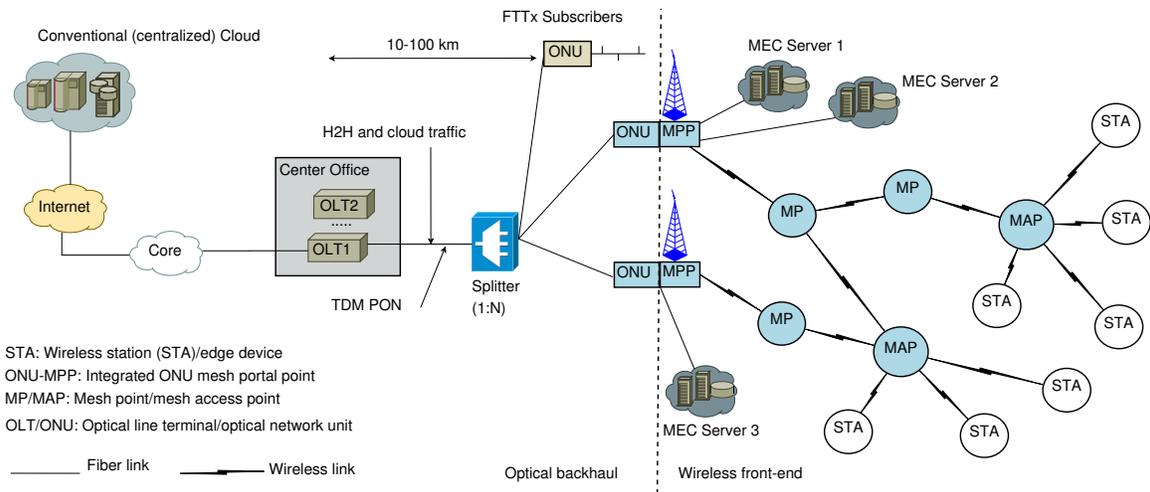


Figure R.6: Architecture des infrastructures de communication CM-FiWi avec partage de backhaul en fibre.

entre le réseau d'accès FiWi à haut débit, cloud et le MEC. Enfin, cette thèse présente un cadre analytique complet pour analyser la performance du projet de gestion unifiée des ressources. De plus, un modèle analytique probabiliste est développé pour estimer la capacité de survie du backhaul optique, du réseau frontal sans fil et du MEC, et d'étudier l'impact avantageux de diverses stratégies avancées de redondance du MEC et du backhaul localisé en fibres.

Fig. R.6 illustre l'architecture des infrastructures de communication de CM-FiWi. Le backhaul à base de fibre optique consiste en un EPON/IEEE P802.3av10G-EPON IEEE 802.3ah largement déployé avec une portée étendue de fibres de 10 à 100 km entre le terminal central de ligne optique (OLT) au bureau central et les ONUs distants. Le backhaul optique fournit aux utilisateurs finaux un accès à haut débit et assure le trafic agrégé du réseau frontal WMN (réseau maillé sans fil). L'OLT se connecte à deux sous-ensembles d'ONUs dans une topologie arborescente. Cependant, des défaillances de liaisons de fibre peuvent encore se produire en raison de coupures de fibre et donc les liens pourraient devenir indisponibles pour acheminer le trafic.

Le premier sous-ensemble d'ONUs est situé dans les locaux des abonnés résidentiels ou commerciaux, fournissant des services FTTx (par exemple, de la fibre à la maison) aux abonnés filaires. Le deuxième sous-ensemble d'ONUs est équipé d'un MPP/passerelle à interface avec le WMN, où les paquets agrégés entrent dans le backhaul optique. Les points maillés (MPs) sont des noeuds de relais prenant en charge des services maillés sans fil, tels que le routage maillé et le transfert de paquets sur plusieurs liens pour le compte d'autres MPs, par lequel les paquets sont finalement relayés au serveur OLT ou MEC par un ONU-MPP. À partir de l'OLT, les paquets sont acheminés

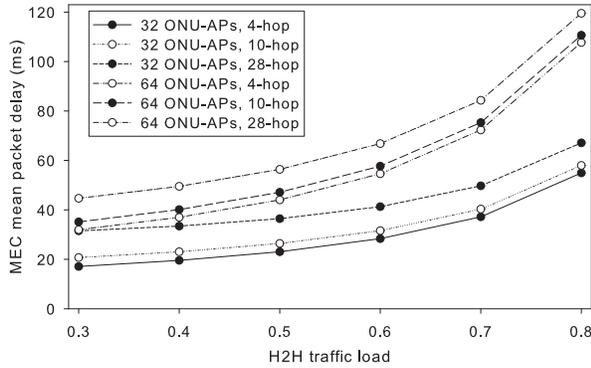


Figure R.7: Impact de la charge de trafic H2H sur les performances de retard de paquets MEC dans un scénario sauts multiples.

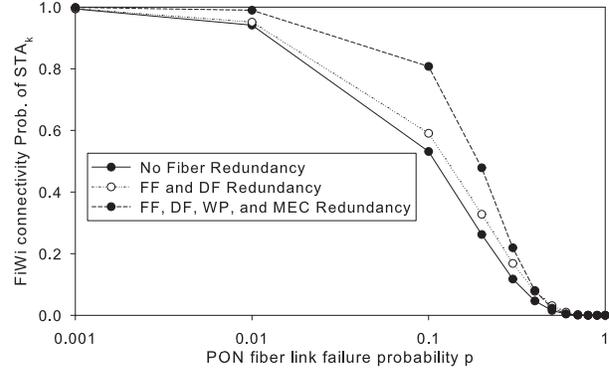


Figure R.8: Probabilité de connectivité FiWi de STA_k contre la probabilité d'échec de backhaul optique p avec différents schémas de survie.

vers l'internet ou vers un cloud conventionnel (centralisé) (voir la Fig. R.6). Les points d'accès maillés (MAP) sont un type spécial de MP qui fournissent un accès sans fil aux clients de maillages conformes à la norme IEEE 802.11 (utilisateurs finaux), c'est-à-dire des périphériques/STA tels qu'un téléphone cellulaire, des PDA, des capteurs et des textile intelligents qui se trouvent dans leur zone de couverture. Les emplacements des ONU-MPPs, des MPs et des MAPs peuvent être planifiés de façon stratégique [41]. Le segment frontal du CM-FiWi peut utiliser des technologies sans fil standards tel que le Wi-Fi. Notez que 4G LTE/LTE-A peut également être utilisé sur le front-end sans fil des réseaux CM-FiWi, ce qui est laissé pour les futurs travaux.

Un ou plusieurs serveurs MEC peuvent être connectés à l'ONU-MPP via des liaisons à fibres optiques dédiées pour fournir des services cloud au bord des réseaux d'accès FiWi (voir la figure R.6). Le serveur MEC (ou le système) se compose d'un MEC hébergeant un système de gestion d'infrastructure (par exemple, serveur, couche de virtualisation et gestionnaire de virtualisation), du système de gestion de plate-forme d'application MEC (par exemple, registre de services, services de communication) et systèmes de gestion d'applications (par exemple, machine virtualisée pour le MEC, application MEC). Pour plus de détails sur chaque composant du serveur MEC, les lecteurs intéressés sont référencés à [42]. Le réseau CM-FiWi envisagé consiste à fournir une infrastructure de communication complète de bout-en-bout. L'infrastructure du réseau unique CM-FiWi peut être utilisée pour le service à haut débit et les deux types de services cloud de manière rentable.

Figure R.7 montre la performance en terme de délai du MEC en fonction de la charge de trafic H2H et du nombre de sauts sans fil pour un nombre croissant d'unités ONU-MPP $N \in \{32, 64\}$. Nous observons que lorsque la charge de trafic H2H augmente, D_{mec} augmente également dans le

scénario sauts multiples. De plus, comme prévu, la Fig. R.7 montre que le délai du MEC augmente lorsque le nombre des sauts sans fil $N \in \{4, 10, 28\}$ augmente. Il est important de noter qu'un MEC avec un maximum de 4 sauts dépasse encore un cloud conventionnel pour un scénario typique de 32 unités APs ONU et une charge de trafic lourd H2H de 0,8. Cela démontre que les applications tolérantes en délai peuvent être déchargées sur 4 sauts MEC au lieu du cloud conventionnel pour une meilleure performance du réseau. En outre, cela indique qu'un puissant serveur MEC multicœur peut être déployé à un point d'agrégation, où plusieurs unités APs ONU sont situées à proximité partageant une seule plate-forme MEC. Cette approche centralise les ressources et contribue ainsi à réduire à la fois le CAPEX et l'OPEX sans subir de délai significatif.

Les avantages des schémas de survie pour la probabilité de connectivité FiWi des STA sont présentés à la Fig. R.8, où une ONU se connecte sans fil à ses ONU voisins via les liens 3-hop WMN dans le schéma de protection sans fil. Figure. R.8 montre qu'en employant conjointement l'alimentateur du fibre (FF), la distribution du fibre (DF), l'interconnexion du fibre (IF), la protection sans fil et systèmes de redondance MEC, la connectivité FiWi est significativement améliorée par rapport au scénario de non-redondance de fibre pour une large gamme de probabilités d'échec de liaison de fibre PON. Par exemple, pour un scénario typique avec une probabilité de défaillance de la liaison de fibre PON de $p = 0,1$, une probabilité de connectivité FiWi des STA de 0,81 est atteinte. Par conséquent, les réseaux CM-FiWi aident à fournir un accès large bande hautement fiable et des services MEC.

Scénario de déploiement émergent des réseaux d'accès FiWi améliorés sur cloud-cloudlet: L'internet tactile

Lors d'une entrevue en 2013, nous avons réfléchi à un futur économique de la convergence technologique dans les années 2020 « âge d'or », où des tâches importantes de notre vie quotidienne pourraient être de plus en plus réalisées par des robots [43]. À titre d'exemple personnel, nous avons envisagé la possibilité souhaitable non seulement de surveiller, mais aussi d'agir à distance à partir du Canada par l'intermédiaire d'internet pour soutenir nos parents âgés vivant en Allemagne. Cette vision de l'internet est maintenant largement connue sous le nom de l'internet tactile, un terme d'abord inventé par G. P. Fettweis au début de l'année 2014 [28, 44]. L'internet tactile devrait avoir le potentiel de créer une pléthore de nouvelles opportunités et applications qui remodeleront notre vie et l'économie. Une analyse préliminaire du marché a révélé que le marché potentiel pourrait

atteindre 20 billions de dollars US dans le monde entier, soit environ 20% du produit intérieur brut (PIB) mondial actuel [45].

L'internet tactile sera centré autour des communications homme-machine (H2M) utilisant des dispositifs tactiles/haptiques. Plus important encore, malgré leurs différences, IoT, 5G et l'internet tactile semblent converger vers un ensemble commun d'objectifs de conception importants: (i) *très faible latence* de l'ordre de 1 ms, (ii) *fiabilité ultra haute* avec une disponibilité presque garantie de 99,999%, (iii) *la coexistence* H2H/M2M, (iv) l'intégration des *technologies centrées sur les données* avec un intérêt particulier sur le WiFi, ainsi que (v) *la sécurité*.

La vision de l'internet tactile et son impact potentiel sur la société devrait ajouter une nouvelle dimension à l'interaction homme-machine dans une variété de domaines d'application différents, y compris les soins de santé, l'éducation et le réseau intelligent. Pour une description détaillée, le lecteur intéressé est référé à [46]. L'infrastructure d'information et de communication qui permet à l'interface tactile envisagée doit répondre à un certain nombre d'exigences de conception. Ces principaux objectifs de conception de l'Internet tactile ne peuvent être obtenus qu'en gardant les applications tactiles locales, proches des utilisateurs, ce qui nécessite une architecture de plateforme de services répartie (c'est-à-dire *décentralisée*) basée sur les cloudlets et le mobile-edge (À discuter plus en détail dans ce qui suit). De plus, des procédures *évolutives* à toutes les couches du protocole sont nécessaires pour réduire la latence de bout-en-bout des capteurs aux actionneurs. Il est important de noter que l'internet tactile établira des demandes exigeantes pour les *réseaux d'accès* futurs en termes de latence, de fiabilité et aussi de capacité (par exemple, des débits de données élevés pour les capteurs vidéo). Les réseaux d'accès filaires répondent en partie à ces exigences, mais les réseaux d'accès sans fil ne sont pas encore conçus pour répondre à ces besoins. Selon le rapport sur l'informatique tactile de l'UIT-T [46], l'intensification de la recherche dans ce domaine sera essentielle, introduisant de nouvelles idées et concepts pour renforcer la redondance et la diversité inhérentes aux réseaux d'accès pour répondre aux exigences strictes de latence et de fiabilité des applications d'internet tactiles. Fig. R.9 montre la taxonomie des technologies et techniques proposées pour l'internet tactile.

Dans [30], nous avons travaillé sur le déploiement des clouds et des cloudlets dans le FiWi amélioré sur LTE-A HetNets pour augmenter le débit, réduire la latence de bout en bout et améliorer l'évolutivité au moyen de déchargement de calcul. Récemment, nous avons étendu ce travail préliminaire en étudiant la coexistence du trafic classique à haut débit et celui du MEC dans un tel

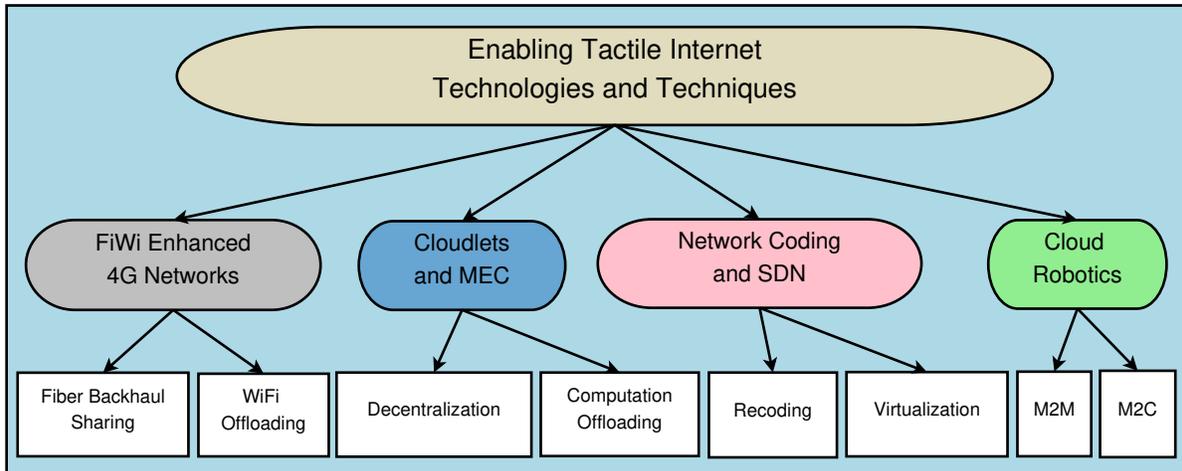


Figure R.9: Taxonomie des technologies et techniques de l'internet tactile.

réseau fortement convergé. Nos résultats obtenus indiquent que l'utilisation de cloudlets au bord des réseaux d'accès FiWi nous permet de rapprocher la vision de l'Internet tactile à la réalité à travers du MEC, ce qui permet d'obtenir une latence de bout-en-bout significativement réduite et une amélioration des performances du réseau global. A titre d'illustration, la Fig. R.10 montre l'efficacité moyenne du temps de réponse de décharge pouvant être atteinte pour le déchargement de calcul sur des cloudlets.

En général, le déchargement des calculs doit être effectué si le temps requis pour exécuter localement une tâche donnée sur le périphérique mobile est beaucoup plus long que le temps de réponse du déchargement de la tâche sur un cloudlet. Cette différence de temps est appelée gain de la décharge. L'efficacité moyenne du temps de réponse du déchargement est définie comme le rapport entre le gain de déchargement et le temps de réponse des tâches exécutées localement sur les appareils mobiles. Dans ce qui suit, nous supposons que la charge de données d'une tâche de calcul est fragmentée en paquets de taille fixe et une application est subdivisée en un certain nombre de tâches fines. La Fig. R.10 représente l'efficacité du temps de réponse du déchargement pouvant être obtenu pour différentes tailles de paquets de déchargement. Nous observons que lorsque la charge du trafic de décharge augmente graduellement, l'efficacité globale du temps de réponse augmente. La figure R.10 montre que pour augmenter les tailles des paquets de déchargement, l'efficacité moyenne globale du temps de déchargement s'approche de façon asymptotique de 100%. Par exemple, pour un cas typique de $N = 16$ et une taille de paquet de déchargement de 1100,60 Ko, l'efficacité moyenne du temps de réponse du déchargement total est de 95,50%. Cela se traduit par une réduction de délai de 95,50% par rapport au délai obtenu dans un scénario sans MEC et sans déchargement.

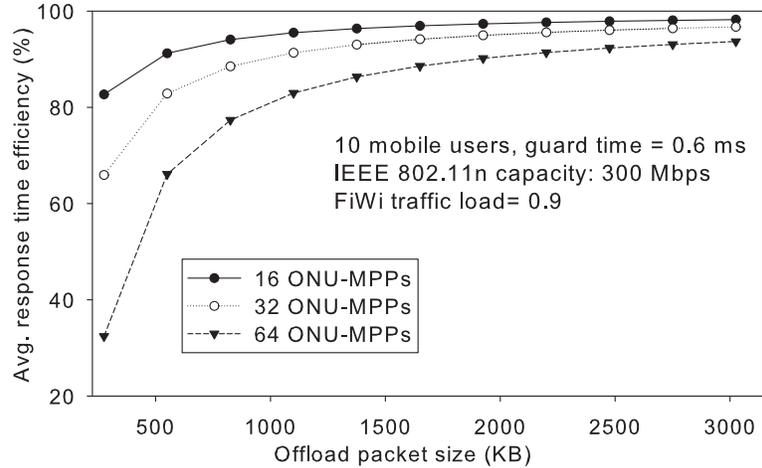


Figure R.10: Rendement moyen du temps de réponse du déchargement vs. La taille du paquet déchargé.

Conclusions

Les agents de l'industrie sont de plus en plus conscients des avantages de la convergence des clouds mobiles en étendant le cloud d'aujourd'hui non modifié à une architecture décentralisée cloud-cloudlet à deux niveaux basée sur les capacités émergentes du MEC. À la lumière de la tendance des réseaux mobiles 5G orientés vers la décentralisation basée sur les clouds, les stations de base intelligentes, MEC, les capacités inhérentes de traitement distribué et la capacité de stockage des réseaux *R&F* peuvent être exploitées pour des nouvelles applications. Cette thèse de doctorat repose sur une vision complète des réseaux d'accès améliorés FiWi à haut débit en se concentrant sur trois grandes tendances, à savoir les communications vertes filaires et/ou sans fil, l'intégration de clouds et de cloudlets dans les réseaux FiWi et l'internet tactile.

Dans les réseaux d'accès FiWi, les problèmes d'efficacité énergétique doivent être abordés de façon complète, en prenant en compte non seulement les terminaux sans fil, mais aussi les backhaul optiques, pour prolonger la durée de vie des appareils sans fil et pour permettre aux opérateurs de réduire leur OPEX sans compromettre la QoS. La première partie de cette thèse a proposé un nouveau système de conservation d'énergie pour le réseau d'accès FiWi basé sur EPON-WLAN (ECO-FiWi) qui planifie conjointement les modes d'économie d'énergie des stations sans fil et des points d'accès et des unités de réseau optique pour réduire leur consommation d'énergie. ECO-FiWi maximise la performance globale du réseau en exploitant le TDMA pour synchroniser les modes d'économie d'énergie et les intégrer dans le processus DBA. Le système ECO-FiWi proposé permet

de réaliser des économies d'énergie importantes (plus de 70% dans les scénarios typiques) tout en préservant le délai ascendant et en engendrant un délai gérable dans le trafic descendant.

Différentes facettes de la recherche sur les clouds et les cloudlets ont été explorées, y compris la planification des flux de travail dans les environnements de cloud computing à locataires multiples et les réseaux d'accès améliorés FiWi par les clouds et les cloudlets. En particulier, la deuxième partie de cette thèse est consacrée à la conception d'une nouvelle architecture à quatre couches du système de planification du flux de travail et d'un nouvel algorithme d'ordonnancement du flux de travail dans un environnement de cloud computing à locataires multiples. Les résultats obtenus montrent que l'ordonnancement CWSA proposé surpasse les autres politiques d'ordonnancement dans la littérature. Plus important encore, le CWSA a démontré utiliser correctement les ressources de calcul en réduisant le temps d'inactivité des noeuds de ressources cloud. En outre, des expériences de preuve de concept ont été réalisées en utilisant des applications de flux de travail scientifiques réelles. L'expérience pour prouver le concept indique que la politique de planification CWSA proposée offre des améliorations significatives pour les applications de flux de travail. Comme fait important, une des principales leçons tirées de cette étude est que la location multiple aide à améliorer l'utilisation des ressources.

De plus, cette thèse a introduit le nouveau concept de MEC sur les réseaux FiWi. Outre plusieurs avantages de MEC, un certain nombre de défis de recherche intéressants en termes d'intégration et de coordination du réseau, la gestion des ressources distribuées, la coexistence du trafic H2H et MEC, la coexistence des clouds et des cloudlets, la fiabilité et la mobilité ont été discutés. Trois scénarios de conception envisagés pour MEC sur les réseaux FiWi ont été étudiés, suivis d'un nouveau programme unifié de gestion des ressources proposé pour les réseaux MEC sur les réseaux FiWi à base Ethernet. Les résultats obtenus montrent les avantages significatifs de MEC sur les réseaux FiWi. Par exemple, pour un scénario typique, une efficacité de temps de réponse de 95,36% peut être obtenue. Il est important de noter que le délai moyen du paquet MEC de 20,50 ms est obtenu pour le scénario considéré, tout en permettant une coexistence efficace H2H/MEC sans dégrader les performances du réseau. De plus, la durée de vie de la batterie des périphériques de bord est prolongée jusqu'à 11h30 en utilisant la solution proposée.

La troisième partie de cette thèse a étudié la faisabilité de la conception de la coexistence des services cloud et de H2H (scénarios centralisés et MEC) dans un réseau d'accès amélioré FiWi. Un nouveau système de gestion unifiée des ressources à base de TDMA à deux couches a été proposé

pour planifier simultanément le trafic de cloud, MEC et H2H conventionnel. Une analyse complète des performances a été développée pour évaluer le temps de réponse des paquets, l'efficacité du temps de réponse et le rapport de surcharge de gain de déchargement. Tandis que le réseau proposé est à l'abri des coupures de fibres, la forme centrale OLT a un point de défaillance unique. Pour assurer la survie de manière efficace pour les segments FiWi et MEC, différents plans de protection ont été proposés et évalués. Les solutions présentées permettent de réduire le délai de transfert des paquets sans affecter les performances du trafic H2H du réseau. Les résultats obtenus montrent que pour un système typique de 32 ONU-APs et de charge du trafic H2H de 0,6, le MEC et le délai de paquet de cloud de 26,73 ms et 126,72 ms peuvent être atteints, respectivement, sans dégrader la performance du réseau du trafic H2H. Sur la base des résultats obtenus, nous suggérons également que le nombre maximal de sauts MEC ne doit pas dépasser 4 dans les réseaux CM-FiWi pour que les performances du réseau soient acceptables.

La dernière partie de cette thèse étudie un scénario d'application émergent des réseaux d'accès FiWi renforcé par les cloud/cloudlet. En particulier, cette thèse étudie la faisabilité du concept de l'internet Tactile et explore sa portée. La taxonomie des technologies et des techniques d'internet tactiles a été présentée. De plus, le rôle des réseaux d'accès FiWi cloudlet/MEC à l'appui de l'internet tactile est ensuite enquêté en profondeur. Les résultats obtenus indiquent que l'utilisation des clouds/MEC au bord des réseaux d'accès FiWi nous permet de rapprocher la vision de l'internet tactile de la réalité, ce qui permet d'obtenir une latence de bout-en-bout significativement réduite et une performance globale améliorée du réseau.

Contents

Abstract	v
Acknowledgments	ix
Résumé	R-1
Contents	i
List of Figures	v
List of Tables	ix
List of Acronyms	xi
List of Symbols	xv
1 Introduction	1
1.1 Background and Motivation	1
1.1.1 Passive Optical Networks	1
1.1.2 Next-Generation PON and Beyond	2
1.1.3 FiWi Broadband Access Networks	5
1.1.4 Recent Progress on FiWi Enhanced Access Networks	7
1.1.5 Cloud Computing, Mobile Cloud Computing, and Cloudlet	10
1.1.6 Mobile Data Offloading vs. Computation Offloading	14
1.2 Objectives	17
1.3 Research Methodology	18
1.4 Contributions of the Thesis	19
1.4.1 Energy-Aware Resource Management Scheme for FiWi Access Networks	20
1.4.2 Workflow Scheduling in Multi-Tenant Cloud Computing	20
1.4.3 Cloud and Edge Computing Enhanced FiWi Access Networks	21
1.4.4 Tactile Internet	23
1.5 List of Publications	24
1.6 Dissertation Outline	27
2 Energy-Aware Resource Management Scheme for Integrated Fiber-Wireless Access Networks	29
2.1 Introduction	29
2.2 Energy Conservation Scheme for FiWi Access Networks	32
2.2.1 Network Architecture	32

2.2.2	ECO-FiWi General Description	34
2.2.3	Joint Bandwidth Allocation and Power-Saving Scheduling Algorithms	36
2.3	Energy Saving Model	40
2.4	Analysis of End-to-End Delay	42
2.4.1	Mean Upstream Frame Delay	43
2.4.2	Mean Downstream Frame Delay	47
2.4.3	Estimate of Upper Bound on End-to-end Frame Delay	50
2.5	Results	52
2.6	Conclusions	58
3	Workflow Scheduling in Multi-Tenant Cloud Computing Environments	61
3.1	Introduction	61
3.2	Workflow	64
3.3	Related Work	66
3.4	Multi-Tenant Cloud Environments for Workflow Scheduling	68
3.4.1	Architecture	69
3.4.2	Problem Formulation	72
3.4.3	Cloud Workflow Scheduling Algorithm	74
3.5	Implementation and Performance Evaluation	81
3.5.1	Experimental Setup	82
3.5.2	Simulation Results	83
3.5.3	Proof-of-Concept Experiment	87
3.6	Conclusions	90
4	Cloudlet Enhanced Fiber-Wireless Access Networks for Mobile-Edge Computing	91
4.1	Introduction	91
4.2	Cloudlet Enhanced FiWi Networks for MEC	94
4.2.1	Network Architecture	94
4.2.2	Unified Resource Management Scheme	96
4.2.3	Cloudlet-Aware Dynamic Bandwidth Allocation Algorithms	100
4.3	Performance Analysis	105
4.3.1	Delay of FiWi and Offloaded Packets	106
4.3.2	Transmission Time of Offload Packet and Result	109
4.3.3	Response Time Efficiency	110
4.3.4	Offload Gain-Overhead Ratio	111
4.3.5	Energy Efficiency Gain and Battery Model	112
4.4	Results	115
4.5	Conclusions	125
5	Mobile-Edge Computing Empowered Fiber-Wireless Access Networks in the 5G Era	127
5.1	Introduction	127
5.2	Mobile-Edge Computing: Typical Service Scenarios and Design Challenges	131
5.2.1	Typical Service Scenarios	131
5.2.2	Design Challenges	133
5.3	MEC over FiWi Network Architecture	134
5.3.1	MEC over FiWi Networks	134
5.3.2	MEC over Ethernet-based FiWi Networks	138

5.3.3	MEC over 4G LTE-based FiWi Networks	138
5.3.4	Coexistence of MEC and C-RAN in FiWi Enhanced LTE-A HetNets	139
5.4	Resource Management and Performance Evaluation	140
5.4.1	Resource Management Scheme for MEC over Ethernet-based FiWi networks	140
5.4.2	Performance Evaluation	141
5.5	Conclusions	144
6	Mobile-Edge Computing vs. Centralized Cloud Computing in Fiber-Wireless Access Networks	145
6.1	Introduction	145
6.2	CM-FiWi Communication Infrastructures	149
6.2.1	Architecture	149
6.2.2	Unified Resource Management Scheme	151
6.3	Performance Analysis	157
6.4	Results	170
6.5	Conclusions and Outlook	175
7	Emerging Deployment Scenario of Cloud-Cloudlet Enhanced FiWi Access Networks: The Tactile Internet	177
7.1	Introduction	177
7.2	Tactile Internet: Vision and Design Guidelines	182
7.3	Tactile Internet: Recent Progress	182
7.4	Open Challenges and Future Research Directions	190
7.5	Conclusions	191
8	Conclusions and Outlook	193
8.1	Conclusions	193
8.2	Outlook	196
	References	199

List of Figures

R.1	Méthodologie de recherche.	R-6
R.2	Économie d'énergie et délais des trames (\bar{D}_u et D_u^{max}) US en fonction de la charge du trafic US ρ_u	R-9
R.3	Architecture de référence proposée pour la planification des flux de travail dans un environnement de cloud multi-locataire.	R-11
R.4	Illustration de l'évaluation du mobile-edge et de la cloudification des réseaux 5G. . .	R-14
R.5	a) Efficacité du temps de réponse vs. Taille du paquet de déchargement; b) Délai moyen des paquets du trafic coexistant entre H2H et MEC; c) Consommation d'énergie contre la charge du trafic H2H pour les scénarios MEC et non-MEC; d) Durée de vie de la batterie des périphériques vs. Charge du trafic déchargé. (TL: Charge de trafic H2H).	R-16
R.6	Architecture des infrastructures de communication CM-FiWi avec partage de backhaul en fibre.	R-20
R.7	Impact de la charge de trafic H2H sur les performances de retard de paquets MEC dans un scénario sauts multiples.	R-21
R.8	Probabilité de connectivité FiWi de STA_k contre la probabilité d'échec de backhaul optique p avec différents schémas de survie.	R-21
R.9	Taxonomie des technologies et techniques de 'internet tactile.	R-24
R.10	Rendement moyen du temps de réponse du déchargement vs. La taille du paquet déchargé.	R-25
1.1	Typical topology of TDM-PON system.	2
1.2	Evolution and capacity trend of PONs (different options of downstream and upstream capacity shown for the IEEE 802.3ca NG-EPON is still under discussion).	3
1.3	Schematic block diagram of RoF system (E-O: Electrical-to-Optical; O-E: Optical-to-Electrical; BPF: Band-pass filter; SMF: Standard single mode fiber).	6
1.4	The evolution of wired and wireless technologies with projections. Source: M. Décina [69], based on data by Bell Labs, G. Fettweis and S. Alamouti [28].	7
1.5	Overview of cloud computing: Characteristics, challenges, service delivery, and deployment models.	11
1.6	Research methodology.	18
2.1	FiWi network architecture with EPON backhaul and WLAN-based front-end.	33
2.2	Illustration of ECO-FiWi for FiWi access networks.	35
2.3	M/G/1 queueing model of US scheduling: (a) Illustration of US scheduling; (b) Illustration of residual delay component.	43

2.4	Frame arrival and its reservation-and-vacation time: (a) During ONU1's data interval; (b) During ONU1's reservation interval; (c) During ONU j 's data interval; (d) During ONU j 's reservation interval; and (e) During ONU-AP's sleep interval.	47
2.5	M/G/1 queueing model of DS scheduling: (a) Illustration of data and reservation intervals; (b) Illustration of reservation time difference between the conventional polling system (upper part) and DS scheduling (lower part).	48
2.6	Illustration of upper bounds on US and DS frame delays in ECO-FiWi networks. . .	51
2.7	Potential energy saving with varying cycle time T_c : Integrated vs. Conventional ONU (η vs. η_{mosta}).	53
2.8	Polling cycle time T_c as function of ρ_u , RTT , and T_g	53
2.9	Energy saving η and US frame delays (\bar{D}_u and D_u^{max}) as function of US traffic load ρ_u	54
2.10	Energy saving (η) and US frame delays (\bar{D}_u and D_u^{max}) as function of PON RTT	56
2.11	Energy saving (η) and US frame delays (\bar{D}_u and D_u^{max}) as function of number of STAs per ONU-AP (M).	56
2.12	DS-based cycle time T_c^d and exhaustive service threshold for DS transmission ρ_d^{th}	57
2.13	DS end-to-end frame delays (\bar{D}_d and D_d^{max}).	58
3.1	SIPHT [140] – an example of scientific workflow.	65
3.2	Proposed reference architecture of workflow scheduling in a multi-tenant cloud environment.	69
3.3	Representation of a cloud workflow by a direct acyclic graph (DAG).	73
3.4	Illustration of scheduling multiple workflows.	80
3.5	Mean makespan vs. total number of workflow.	84
3.6	Cumulative distribution function (CDF) of makespan for different scheduling schemes.	85
3.7	Average execution time of scheduling schemes.	85
3.8	Resource utilization.	86
3.9	Average tardiness vs. number of workflows.	86
3.10	99 th percentile makespan of different scheduling schemes for the SIPHT workflow application.	88
3.11	99 th percentile makespan of different scheduling schemes for the CyberShake workflow application.	88
4.1	Generic network architecture of cloudlet enhanced FiWi access networks for mobile-edge computing.	95
4.2	Illustration of the proposed unified resource management scheme for MEC integrated FiWi access networks.	97
4.3	Format of extended REPORT message.	98
4.4	Packet delay components of both conventional H2H and MEC traffic.	106
4.5	Illustration of the sleep mode at an STA: a) when tasks are executed locally at STA; b) when tasks are offloaded onto a cloudlet.	112
4.6	Polling cycle time T_c vs. conventional FiWi traffic load for different guard time T_g and number of ONU-APs.	117
4.7	Impact of FiWi traffic load on packet delay of both FiWi and MEC traffic.	118
4.8	Impact of FiWi traffic load ρ^{h2h} on cloudlet response time $\bar{R}^{cloudlet}$	118
4.9	Impact of polling cycle time T_c on average offload gain-overhead ratio $\bar{\gamma}$	119
4.10	Impact of offloaded traffic load ρ^{mec} on the average response time of cloudlet $\bar{R}^{cloudlet}$ with different number of ONU-APs.	119

4.11	Impact of offloaded traffic load with different numbers of ONU-APs on the average offload gain-overhead ratio $\bar{\gamma}$	120
4.12	Impact of offload packet size on the average response time efficiency $\bar{\delta}$	120
4.13	Impact of offload packet size on average energy efficiency $\bar{\eta}$	121
4.14	Impact of offloaded traffic load ρ_u^o on average battery life \bar{B}_l with different number of ONU-APs.	121
4.15	Experimental testbed setup for cloudlet enhanced FiWi networks with MEC capabilities: 1) optical fiber loops; 2) passive splitter at remote node of EPON; 3) cable connecting the gateway for core network; 4) cable connecting APs; 5) WLAN access point; 6) Ethernet cable connecting an ONU; 7) Ethernet cable connecting cloudlet; 8) cloudlet server hosting OpenStack++ platform; 9) running VM instance in OpenStack++ on the cloudlet server; 10) STA/edge device running edge application.	122
4.16	Comparison of experimental and analytical results: a) Impact of offload packet size on average response time efficiency, b) impact of offloaded traffic load on average response time of cloudlet.	124
5.1	Illustration of mobile edge computing and cloudification of 5G networks.	129
5.2	Overview of mobile edge computing: benefits, potential use cases, enabling technologies, and challenges (LWA: LTE and WiFi aggregation, LAA: licensed assisted access, LTE-U: LTE Unlicensed, LWI: LTE/WiFi interworking, CDN: content delivery network).	131
5.3	MEC over FiWi network architectures: a) MEC over Ethernet-based FiWi networks and MEC over 4G LTE-based FiWi networks; b) coexistence of MEC and C-RAN over FiWi enhanced 4G LTE HetNets.	137
5.4	Illustration of TDMA-based unified resource management scheme and sleep mode for MEC over WLAN-based FiWi networks.	140
5.5	a) response time efficiency vs. offload packet size; b) mean packet delay of co-existing H2H and MEC traffic; c) energy consumption vs. H2H traffic load for MEC and non-MEC scenarios; d) battery life of edge devices vs. offloaded traffic load. (TL: H2H traffic load).	143
6.1	Architecture of CM-FiWi communications infrastructures with fiber backhaul sharing.	150
6.2	Illustration of the proposed unified resource management scheme for CM-FiWi network.	152
6.3	Components of packet delay of both conventional H2H and MEC traffic.	161
6.4	Components of packet delay of cloud traffic.	163
6.5	Illustration of optical backhaul reliability, MEC reliability, and wireless protection by utilizing redundancy techniques.	168
6.6	Delay performance for single-hop scenario.	171
6.7	Impact of H2H traffic load on MEC packet delay performance in multi-hop scenario.	171
6.8	Delay performance as function of polling cycle time.	172
6.9	Response time efficiency vs. offloaded packet size.	172
6.10	Offload gain-overhead ratio vs. offloaded traffic load.	173
6.11	CCR vs. offloaded traffic load.	173
6.12	FiWi connectivity probability of STA_k vs. optical backhaul failure probability p with different survival schemes.	174
6.13	Mean end-to-end delay vs H2H traffic load with/without IF scenario.	174

7.1	The Tactile Internet: a) revolutionary leap of the Tactile Internet (in compliance with ITU-T Technology Watch Report [46]); b) the three lenses of IoT, 5G, and the Tactile Internet: Commonalities and differences.	179
7.2	Taxonomy of enabling Tactile Internet technologies and techniques.	183
7.3	FiWi enhanced LTE-A HetNets performance: a) Average end-to-end delay vs. aggregate throughput for different WiFi offloading ratio (WOR); b) FiWi connectivity probability of a mobile user vs. EPON fiber link failure probability p	185
7.4	Generic architecture of cloudlet enhanced FiWi access networks for mobile-edge computing.	186
7.5	Average offload response time efficiency vs. offload packet size.	187
7.6	Two-tier cloud robotics systems architecture based on M2M and M2C communications.	190

List of Tables

2.1	Notations and descriptions	36
2.2	Reservation-and-vacation time for US scheduling in ECO-FiWi networks	45
3.1	Resource types and prices used (based on on-demand instances offered by Amazon EC2, US East (N. Virginia))	83
3.2	Workflow parameters and default values	83
3.3	Average makespan, skewness of the makespan and standard deviation of the makespan for different scheduling.	85
3.4	Characteristics and categorization of scientific workflow applications	87
3.5	Average cost (\$) of scientific workflow execution	89
4.1	Notations, definitions, and default values	99
4.2	System configuration of experimental testbed	123
5.1	Comparison of conventional D-RAN, C-RAN, and emerging cloudlet enhanced D-RAN	135
6.1	Summary of different aspects of network and service management in CM-FiWi networks	151
6.2	Notation, description, and default values	154

List of Acronyms

3GPP	Third Generation Partnership Project	DBA	Dynamic Bandwidth Allocation
4G	Fourth Generation	DRX	Discontinuous Reception
5G	Fifth Generation	DS	Downstream
AP	Access Point	DSI	Digital Senses Initiative
AR	Augmented Reality	DSP	Digital Signal Processor
ASIC	Application Specific Integrated Circuit	E-O	Electrical-to-Optical
BBU	Baseband Unit	E2E	End-to-End
BL-HMOEA	Bilevel Hybrid Multiobjective Algorithm	EPC	Evolved Packet Core
BPF	Band-pass Filter	EPON	Ethernet PON
BS	Base Station	ETSI	European Telecommunications Standards Institute
C-RAN	Cloud Radio Access Network	FCFS	First Come First Served
CAPEX	Capital Expenditure	FCP	Fast Critical Path
CARG	Compound Annual Growth Rate	FF	Feeder Fiber
CCR	Communication-to-Computation Ratio	FIFO	First-In-First-Out
CDF	Cumulative Distribution Function	FiWi	Fiber-Wireless
CDN	Content Delivery Network	FOS	Fiber Optic Sensor
CoMP	Coordinated Multi-Point	FPGA	Field Programmable Gate Array
CPRI	Common Public Radio Interface	FSAN	Full Service Access Network
CPS	Cyber-physical System	FTTH/B	Fiber-to-the-Home/Business
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance	GPON	Gigabit Passive Optical Network
D-RAN	Distributed Radio Access Network	GPP	General Purpose Processor
D2D	Device-to-Device	H2H	Human-to-Human
DAG	Directed Acyclic Graph	H2M	Human-to-Machine
DARA	Delay-aware Routing Algorithm	H2R	Human-to-Robot
DAX	Directed Acyclic Graph in XML Format	HAN	Home Area Network
		HbH	Hop-by-Hop
		HD	High-Definition
		HEFT	Heterogeneous Earliest Finish Time
		HetNets	Heterogeneous Networks

HSS	Home Subscriber Server	MTC	Machine Type Communication
IaaS	Infrastructure-as-a-Service	NAV	Network Allocation Vector
IAN	Industrial Area Network	NG-EPON	Next Generation Ethernet PON
IEEE	Institute of Electrical and Electronics Engineers	NG-PON	Next Generation PON
IF	Interconnection Fiber	NGMN	Next Generation Mobile Network
IMT	International Mobile Telecommunications	O-E	Optical-to-Electrical
IoT	Internet of Things	OAN	Optical Access Network
IPACT	Interleaved Polling with Adaptive Cycle Time	OECD	Organization for Economic Cooperation and Development
ISG	Industry Specification Group	OLT	Optical Line Terminal
ISV	Independent Software Vendor	ONU	Optical Network Unit
ITU	International Telecommunication Union	OPEX	Operational Expenditure
ITU-R	ITU Radiocommunication Sector	OTT	Over-the-Top
ITU-T	ITU-Telecommunication	PaaS	Platform-as-a-Service
LAA	Licensed Assisted Access	PDN-GW	Packet Data Networks Gateway
LLID	Logical Link Identifier	PHY	PHYSical
LSA	Link State Advertisement	POL	Passive Optical LAN
LTE	Long Term Evolution	PoLTE	Positioning over LTE
LTE-A	Long Term Evolution-Advanced	PON	Passive Optical Network
LTE-U	Long Term Evolution-Unlicensed	PSM	Power Save Mode
M2M	Machine-to-Machine	PSO	Particle Swarm Optimization
MAC	Medium Access Control	QoE	Quality of Experience
MAP	Mesh Access Point	QoS	Quality of Service
MAUI	Mobile Assistance Using Infrastructure	R&F	Radio-and-Fiber
MCC	Mobile Cloud Computing	RACS	Radio Applications Cloud Server
MCT	Minimum Completion Time	RAN	Radio Access Network
MDC	Micro Datacenter	RAT	Radio Access Technology
MEC	Mobile-Edge Computing or Multi-access Edge Computing	RAU	Remote Antenna Unit
MIMO	Multiple Input Multiple Output	RE	Radio Equipment
MME	Mobility Management Entity	REC	Radio Equipment Control
MP	Mesh Point	RF	Radio Frequency
MPCP	Multi-point Control Protocol	RLNC	Random Linear Network Coding
MPP	Mesh Portal Point	RoF	Radio-over-Fiber
		RRH	Remote Radio Head
		RTT	Round-trip Time
		S-GW	Serving Gateway
		SaaS	Software-as-a-Service
		SDN	Software-Defined Networking

SFiWi	Sensor Enhanced Fiber-Wireless	VHT	Very High Throughput
SIEPON	Service Interoperability in Ethernet Passive Optical Networks	VM	Virtual Machine
SMF	Single Mode Fiber	VNI	Visual Networking Index
STA	Wireless Station	VR	Virtual Reality
TCO	Total Cost of Ownership	WAN	Wide Area Network
TDMA	Time Division Multiple Access	WDM	Wavelength Division Multiplexing
TWDM A	Time and Wavelength Division Multiplexed/Multiple Access	WiFi	Wireless Fidelity
UE	User Equipment	WiMAX	Worldwide Interoperability for Microwave Access
UNR-PF	Ubiquitous Networked Robot Platform	WLAN	Wireless Local Area Network
US	Upstream	WMN	Mesh Network
V2R	Vehicle-to-Roadside	WOBAN	Wireless-Optical Broadband Access Network
V2V	Vehicle-to-Vehicle	WP	Wireless Protection
		WS	Wireless Sensor

List of Symbols

B_u^o	Offload buffer backlog	$P'_{C(is)}$	Fiber cut probability of the back-up distribution fiber
B_d	ONU DS buffer backlog at OLT	$P'_{C(sj)}$	Fiber cut probability of the back-up feeder fiber
B_u	ONU US buffer backlog at ONU	P_{ap}^a	AP power consumption in active state
B_c	Battery capacity	P_{onu}^a	ONU power consumption in active state
C'_{cloud}	Number of CPU cycles needed to execute a task at the cloud	P_{rx}^a	STA Rx power consumption in active state
C'_{local}	Number of CPU cycles needed to execute a task at the edge device	P_{sta}^a	STA total power consumption in active state
C'_{mec}	Number of CPU cycles needed to execute a task at the MEC	P_{tx}^a	STA Tx power consumption in active state
CCR_{cloud}	Communication-to-computation ratio of cloud	P_{ap}^s	AP power consumption in sleep state
CCR_{mec}	Communication-to-computation ratio of MEC	P_{onu}^s	ONU power consumption in sleep state
C_{cloud}	Cloud transmission capacity	P_{sta}^s	STA total power consumption in sleep state
C_{mec}	MEC transmission capacity	P_f^a	Failure probability of active MEC server
E	Total energy consumption of an ONU-AP	P_f^s	Failure probability of standby MEC server
E'_{back}	ONU-AP energy consumption in a cycle	P_f^{MAP}	Failure probability of an MAP
E'_{front1}	Energy consumption of M STA	P_f^{MPP}	Failure probability of an MPP
F_{ofl}^{MEC}	Offloaded packet execution failure rate	P_f^{MP}	Failure probability of an MP
F_{tx}^{MEC}	Transmission failure rate	$P_{C(ij)}$	Fiber cut probability of the primary fiber between ONUs and MEC server
M	Number of wireless stations (STAs)	$P_{C(is)}$	Fiber cut probability of the distribution fiber
M	Number of wireless STAs per ONU-AP	$P_{C(sj)}$	Fiber cut probability of the feeder fiber
N	Number of ONUs in system	Q_u	Service delay component
$P'_{C(ij)}$	Fiber cut probability of the back-up fiber between ONUs and MEC server	R_u	Residual delay component
		R_{act}	Number of active resources

R_{req}	Number of required resources	λ_u	Aggregate arrival rate for US transmission
$S^{cloudlet}$	CPU clock speed cloudlet	ρ^{h2h}	Aggregated H2H traffic load (intensity)
T_{sl}^o	Duration of an offload subslot map	ρ^{mec}	Cloudlet offload traffic load
T_{start}^o	Start of an offload subslot map	ρ_d	Aggregate DS traffic load
T_{sl}^r	Duration of an offload result subslot	ρ_u	Aggregate US traffic load
T_{start}^r	Start of an offload result subslot	σ_m	Sample standard deviation of the makespan
T_{sl}^u	Duration of an unoffloading (conventional data) STA subslot	olt_clk	OLT clock
T_{start}^u	Start of an unoffloading (conventional data) STA subslot	onu_clk	ONU-AP clock
$T_{tx}^{cloudlet}$	Offload transmission time	onu_start	Start of an ONU timeslot
T_{sl}^{max}	Maximum allowable timeslot duration	sta_clk	STA clock
T_{pon}^{msg}	MPCP message time (GATE, REPORT)	sta_start	Start of a STA subslot
T_{wl}^{msg}	WLAN message time (e.g., PS-Poll)	\bar{B}_l	Average battery life of the edge device
T_{ap}^{oh}	AP sleep-to-active overhead time	\bar{D}_{cloud}	Cloud offload mean delay
T_{onu}^{oh}	ONU sleep-to-active overhead time	\bar{D}_{fiwi}	Mean end-to-end packet delay for FiWi
T_{sta}^{oh}	STA sleep-to-active overhead time	\bar{D}_{mec}	Offload packet mean delay of MEC
$T_{slp}^{sta'}$	Sleep time of the STA when the task is offloaded onto a cloudlet	\bar{P}^{sta}	Average power consumption of the STA, when the computation task is offloaded onto a cloudlet
T_{slp}^{sta}	Sleep time of the STA when the task is executed at the STA locally	\bar{R}_{cloud}	Average response time of cloud
T_c	PON polling cycle time	\bar{R}_{local}	Average response time of edge device
T_g	Guard time between two consecutive slots	\bar{R}_{mec}	Average response time of MEC
T_{prop}	Propagation delay from OLT to an ONU	\bar{T}_{cloud}^{exe}	Average time to execute a task at cloud server
T_{sl}	Duration of an ONU timeslot	\bar{T}_{local}^{exe}	Average time to execute a task at edge device
T_{subsl}	Duration of a STA subslot	\bar{T}_{mec}^{exe}	Average time to execute a task at MEC server
X_a	Data set of the makespan	\bar{W}_u	The mean US queuing delay for the gated service discipline
Y_u	Reservation-and-vacation delay component	\bar{X}_d	First moment of frame service times for DS frames
η	Overall energy saving	\bar{V}_d^2	Second moment of reservation times for DS frames
κ	Nominal voltage		
λ_d	Aggregate arrival rate for DS transmission		

\overline{V}_u^2	Second moment of reservation times for US frames	\overline{X}	Average value of the makespan
\overline{V}_d	First moment of reservation times for DS frames	\overline{X}_u	First moment of frame service times for US frames
\overline{V}_u	First moment of reservation times for US frames	$\overline{\delta}_{cloud}$	Average response time efficiency of cloud
\overline{X}_d^2	Second moment of frame service times for DS frames	$\overline{\delta}_{mec}$	Average response time efficiency of MEC
\overline{X}_u^2	Second moment of frame service times for US frames	$\overline{\gamma}_{cloud}$	Average offload gain-overhead ratio of cloud
		$\overline{\gamma}_{mec}$	Average offload gain-overhead ratio of MEC

Chapter 1

Introduction

1.1 Background and Motivation

1.1.1 Passive Optical Networks

Passive optical networks (PONs) are considered the most attractive wired access solution because of their high network capacity, reliability, low-cost, increased reach, and relative ease of deployment, resulting from their nature as all-passive networks and point-to-multipoint topology [1, 2]. PON-based fiber access networks are fundamental for enabling fast and ultra-fast broadband services [3]. Since future access network technologies will need to support very high capacity to cope with the massive growth of traffic demands, optical fiber technology is the de facto future-proof technology for broadband access.

PON system, in general, is a point-to-multipoint optical communication system. For illustration, Fig. 1.1 shows the typical topology of a time division multiplexing (TDM) PON system. The PON consists of an optical line terminal (OLT), which is located at the central office and connects multiple optical network units (ONUs) at the customer premises, typically via a 1:N splitter and N:1 combiner in the downstream (i.e., from the OLT to ONUs) and upstream (i.e., from ONUs to the OLT) direction, respectively. In the downstream direction, Ethernet frames from the OLT are broadcast to all ONUs and all the downstream data are carried on one wavelength, as illustrated in Fig. 1.1. Since the logical link ID (LLID) field defines the destination ONU, an ONU filters the

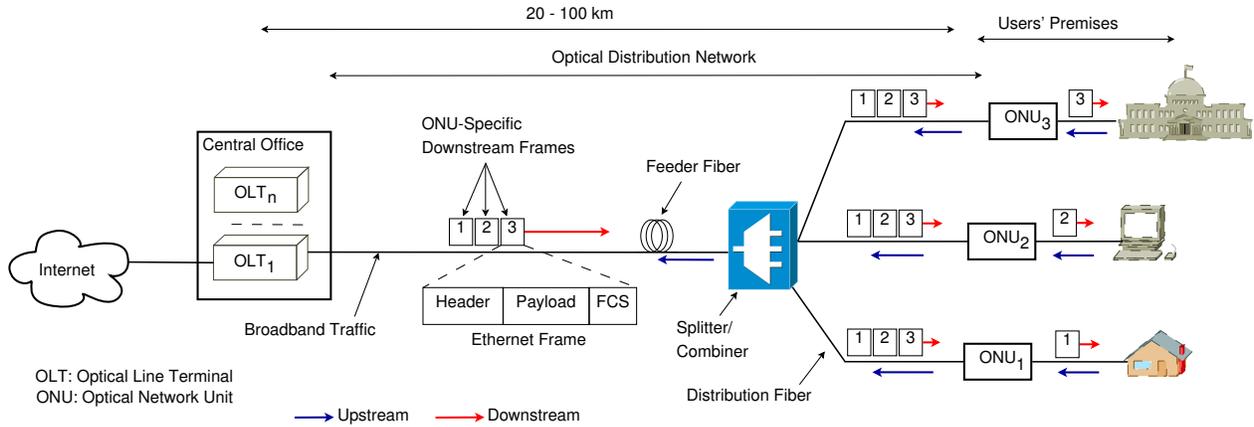


Figure 1.1: Typical topology of TDM-PON system.

received frames based on the LLID in the frame's header (see Fig. 1.1). In the upstream direction, since the medium is shared among ONUs, different channel access schemes are used to avoid data collisions among transmitting ONUs. For example, the most widely deployed ones are time division multiple access (TDMA) and wavelength division multiple access (WDMA).

1.1.2 Next-Generation PON and Beyond

PON technology appears to be the best choice among various wired access technologies for triple-play service delivery (e.g., voice, video, data) to support ever-growing bandwidth-intensive applications and services (e.g., high-definition (HD) TV, 8K ultra-HDTV, high-quality video conferencing, three-dimensional displays, holographic imaging, virtual reality, immersive reality for telemedicine distance learning) due to its reliability, high capacity, and low cost.

Over the past years, different flavors of PON technologies (e.g., Gigabit-PON (G-PON), Ethernet PON (EPON)) have been developed by the ITU-T and IEEE standard bodies to facilitate broadband access. For illustration, the evolution and capacity trend of PONs are shown in Fig. 1.2. The ITU-T/FSAN (Full Service Access Network) G.989 series has defined NG-PON2 [47, 48] as the state-of-the-art PON technology. This architecture relies on a time and wavelength division multiplexing (TWDM) method, which stacks four wavelengths (optionally eight) in a coordinated manner onto a single fiber, with each wavelength delivering 10 Gb/s downstream and 2.5 Gb/s upstream line rates, thereby reaching a total bandwidth of 40 Gb/s downstream and 10 Gb/s upstream. TWDM-PONs are still considered complex (e.g., coordinating wavelengths with time slots together

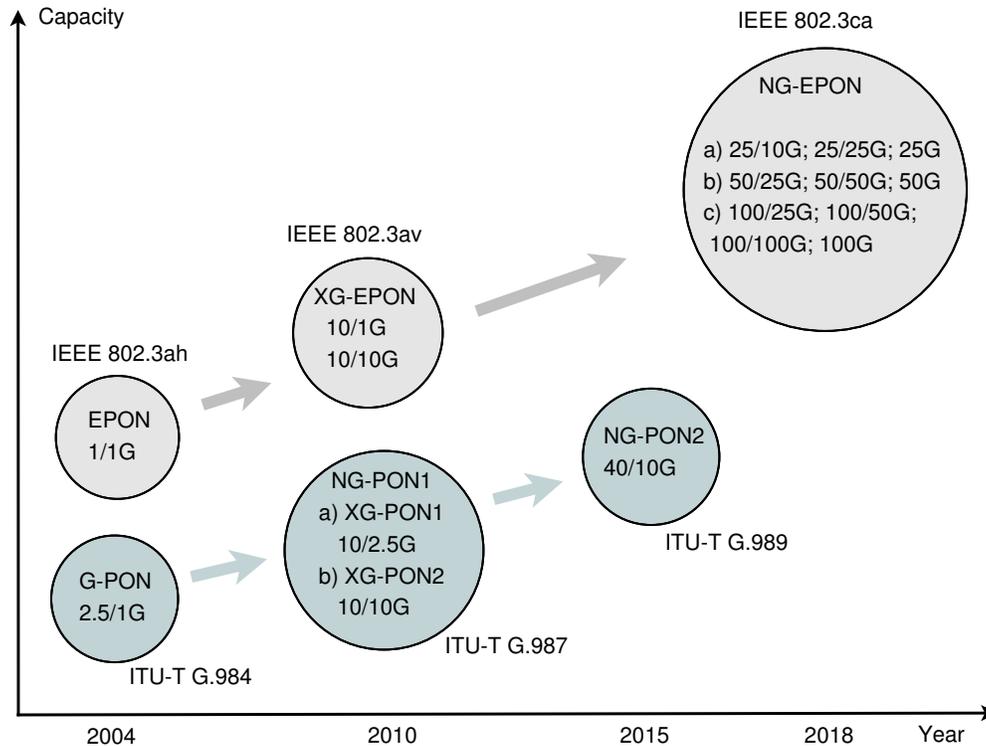


Figure 1.2: Evolution and capacity trend of PONs (different options of downstream and upstream capacity shown for the IEEE 802.3ca NG-EPON is still under discussion).

adds complexity to NG-PON2 dynamic bandwidth allocation (DBA) algorithms) and expensive for widespread deployment at present [49]. On the other hand, similar to the TWDM approach, the IEEE 802.3 working group formed a P802.3ca task force [50] in November 2015 to develop objectives for the next generation of EPON (NG-EPON) that aims at supporting ONUs with one, two, or four wavelengths providing at system capacities of 25/50/100 Gb/s as well as coexistence with the IEEE 802.3av 10G-EPON. The IEEE 802.3ca standard is expected to be released by October 2018.

Given the recent trend in Ethernet-based communication systems, such as Common Public Radio Interface (CPRI)¹ over Ethernet [52] in cloud radio access network (C-RAN) [53], CPRI over Ethernet-based TDM-PONs [54], Ethernet-based transport for mobile fronthaul (IEEE P1914.1), and the passive optical local area network (POL)² technology for enterprise networks [55–57], this thesis considers the most popular and widely deployed PON system to date, that is, TDM EPON

¹CPRI is a specification for the key internal interface of radio base stations between the Radio Equipment Control (REC) and the Radio Equipment (RE). Please refer to [51] for more information about CPRI.

²Passive optical LAN (POL) solutions are based on PON technology (e.g., EPON, G-PON) and use the same principles as PONs in fiber-to-the-home (FTTH) networks. POLs have been optimized for enterprise LAN environments, for example, enterprises, governments, healthcare, education institutions, and hospitality providers. POL is best suited for larger LAN deployments, where scalable and immediate cost savings (up to 50% savings in operating costs) are best realized and long-term operational benefits of the solutions can be gained [55, 56].

(IEEE 802.3ah EPON/IEEE 802.3av 10G-EPON) among many variations of PON technologies, for example, WDM-PON, TWDM-PON [58]. EPON is based on the Ethernet standard, which is simple, easy to manage with capacity growth driven by the enterprise connectivity, and provides economies of scale.

While in recent years both the ITU-T and IEEE standards for PON have put all their effort into achieving higher PON capacity (see Fig. 1.2), those systems are now mature [4]. For example, very recently, British Telecom demonstrated high-speed Terabit optical technologies [5] and successful trials of a 3Tb/s optical superchannel that will allow operators to scale core optical transmission networks gracefully to meet the anticipated increasing demands of bandwidth-intensive applications in fifth generation (5G) mobile networks and beyond. However, instead of just constantly increasing the capacity of NG-PONs, the research focus of broadband access networks should shift towards improving efficiency and supporting emerging services and applications (e.g., cloud computing and cloudlet, mobile backhauling and fronthauling, virtualization). For example, in light of the emerging trends in software-defined networking (SDN) and network function virtualization (NFV), very recently Nippon Telegraph and Telephone Corporation (NTT) has introduced the Flexible Access System Architecture (FASA) concept [59]. FASA aims to provide greater flexibility in optical access equipment through the use of NFV – a virtual OLT for next-generation PONs with the combination of modularized function (software components such as wavelength control, bandwidth control, multicast, OAM (operation, administration, and maintenance)) rather than relying on building equipment for a specific purpose.

Generally, it may not be feasible to deploy optical fiber everywhere due to geographical constraints or when mobility is a necessity. Existing wireless access technologies (e.g., Wireless Fidelity (WiFi) [6], 4G Long Term Evolution (LTE) [7] and LTE-Advanced (LTE-A) [8]), on the other hand, can provide user mobility but require a reliable high-capacity backhaul to meet the bandwidth requirements of emerging bandwidth-hungry applications, such as HDTV, high-quality video conferencing, and three-dimensional displays. In addition, holographic imaging, immersive experiences such as virtual and augmented reality, telemedicine, distance learning, and other high-bandwidth applications will continue to increase the demand for high-speed wireline connectivity. This trend is also in line with the *Edholm's Law of Bandwidth* [9], which states that a unified (or convergent) optical wireline and wireless network is required for providing both fixed and mobile services to users. In light of this, the convergence of fiber and wireless networks, also known as *fiber-wireless*

(*FiWi*) *broadband access networks* [10, 11], is widely viewed as a prominent solution to eliminate the aforementioned shortcomings and to meet the ever growing bandwidth demand, as discussed next.

1.1.3 FiWi Broadband Access Networks

The access network is known as the “last mile” of a telecommunication network that connects the central office with end users. Access technologies can be categorized into two groups: a) wired, for example, Digital Subscriber Line (xDSL) like ITU G.9701 G.Fast, Fiber-to-the-x (FTTx, x: building/home/node) and b) wireless, for example, WiFi (IEEE 802.11b/g/n/ac), WiMax (IEEE 802.16e), 4G LTE/LTE-A. An integrated fiber-wireless (FiWi) broadband access network combines both wired and wireless access technologies. More specifically, FiWi is an architecture that combines the reliability, robustness, and high capacity of optical fiber networks with the ubiquity, flexibility, and cost savings of wireless networks. Further, FiWi networks form a powerful platform for the support and creation of emerging as well as future unforeseen applications and provide broadband services to not only fixed subscribers but also mobile users that help stimulate innovation, generate revenue, and improve the quality of our every-day lives [60, 61].

The IEEE Technical Subcommittee on Fiber-Wireless (TSC-FiWi) Integration [62] defines what FiWi does as follows. “The subcommittee on Fiber-Wireless Integration addresses architectures, techniques, and interfaces for the *integration of fiber and wireless network segments in a unified wired-wireless infrastructure*. It *does not address* architectures or techniques specific to individual optical or wireless networks.” FiWi access networks are also referred to as hybrid optical-wireless networks. Over the past few years, significant progress has been made on the design of hybrid optical-wireless access networks [63]. FiWi networks can be realized in two ways, as discussed in the following.

Radio-over-Fiber: Radio-over-Fiber (RoF) networks use optical fiber as an analog transmission medium between a central station and one or more remote antenna units (RAUs) with the central base station being in charge of controlling access to both optical and wireless media that performs all processing such as modulation/demodulation, routing, medium access control (MAC), among others. For illustration, Fig. 1.3 shows a schematic block diagram of an RoF system for both uplink and downlink. An interesting approach to build low-cost FiWi networks is the use of a single

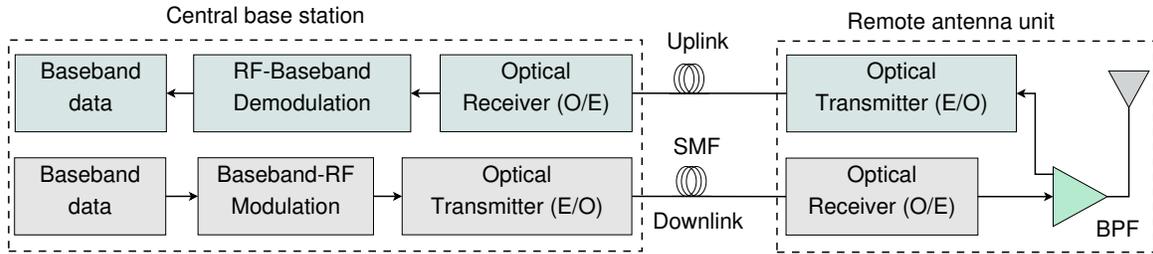


Figure 1.3: Schematic block diagram of RoF system (E-O: Electrical-to-Optical; O-E: Optical-to-Electrical; BPF: Band-pass filter; SMF: Standard single mode fiber).

light source at the central office to generate a downlink optical signal that is reused at RAUs for upstream transmission by means of remote modulation, thereby avoiding the need for an additional light source at each RAU [10, 30]. Traditional RoF networks are suitable to realize networks with centralized control, e.g., traditional cellular networks. RoF networks have been studied for decades and were also used in China Mobile’s cloud radio access network (C-RAN), which relies on a centralized cloud infrastructure and moves baseband units (BBUs) away from remote radio heads (RRHs), rendering the latter ones intentionally as simple as possible without any processing and storage capabilities [53]. The scheduling framework proposed in [64] (see Chapter 2) is a general solution for workflow scheduling in multi-tenant cloud computing environments and applicable for both RoF and R&F based FiWi networks. In general, however, conventional cloud computing typically relies on RoF networks.

Radio-and-Fiber: Radio-and-Fiber (R&F) networks are based on decentralized (optical and wireless) Ethernet technologies and perform protocol translation at the optical-wireless interface in order to cope with the disparate optical and wireless media in a more efficient fashion [10, 30]. Beside MAC protocol translation, the distributed processing and storage capabilities inherently built into optical-wireless interface of R&F networks may be exploited for a number of additional tasks. R&F may become the FiWi network type of choice in light of future 5G mobile networks, benefiting from decentralization based on cloudlets, intelligent base stations, and MEC. The vast majority of R&F based FiWi networks consist of a cascaded TDM IEEE 802.3ah EPON in the backhaul and an IEEE 802.11 a/b/g/n/s WLAN mesh front-end. Apart from next-generation PONs, for example, IEEE 802.3av 10G-EPON or WDM-PON, optical technologies such as tunable lasers play an important role in the design of a flexible and cost-effective optical backhaul for FiWi networks [10, 65].

Beside a WiFi front-end, FiWi networks can consist of a cellular front-end, such as 4G LTE/LTE-A. It is important to note that according to Aptilo (a leading carrier-grade WiFi supplier), WiFi

FiWi Enhanced LTE-A HetNets: Most 4G research has focused on the achievable performance gains in the wireless front-end only without looking into the details of backhaul implementations and backhaul bottlenecks [70]. The importance of high-capacity and low-latency fiber backhaul infrastructures in heterogeneous networks (HetNets) is also emphasized by the introduction of coordinated multi-point (CoMP) in the 3rd Generation Partnership Project (3GPP) LTE-A Release-11 [71]. Motivated by these facts, the authors in [29, 72] proposed the concept of FiWi enhanced LTE-A HetNets, which aims at unifying coverage-centric 4G mobile networks and capacity-centric FiWi broadband access networks based on data-centric Ethernet technologies with fiber backhaul sharing and WiFi offloading capabilities in response to the unprecedented growth of mobile data traffic. It was shown that the proposed solution can provide low-latency and high-reliability fiber backhaul sharing. In addition, it was also reported that a mobile data offloading efficiency of 100% can be achieved for delay-tolerant traffic, provided that mobile users are connected to FiWi with a probability of 0.5 or higher.

Internet of Things over FiWi Access Networks: Communication networks are evolving and transforming to handle massive connectivity of Internet of Things (IoT) devices. IoT is the network of networks where billions of devices/objects (e.g., smartphones, sensors/actuators connected cars, wearables, robots) connect to each other and create new opportunities and challenges. Importantly, the tremendous growth of mobile data traffic together with the increasing integration of radio access technologies (RATs) and the cell densification paving the way to 5G networks gradually shift the bottleneck from the radio interface toward the backhaul segment [70]. With the emerging IoT, the backhaul bottleneck is expected to become even more critical. However, until recently, existing studies on IoT connectivity largely focused on enhancements of RATs without looking into the backhaul segment. Meanwhile, economic considerations may play an important role in the successful rollout of IoT, as experienced in smart grids [73]. To address the backhaul bottleneck, a cost-efficient solution is to share the already deployed high-capacity and reliable optical access network infrastructure. This can be facilitated by integrated FiWi networks.

In [74], the authors aimed to leverage FiWi access networks to design a shared communication infrastructure for supporting both IoT applications and traditional services. Given the paramount importance of energy efficiency in both IoT and access networks, they discussed the possibilities and potential challenges of designing and implementing power-saving mechanisms to prolong the battery life of IoT devices while reducing the energy consumption of the optical backhaul network. Further,

technical guidelines were provided through end-to-end power-saving solutions proposed for typical IoT deployment scenarios, i.e., IoT over EPON-WiFi and IoT over 10G-EPON-LTE. It was shown that in the considered small-scale WiFi-based IoT scenario, more than 95% of energy can be saved by employing TDMA-based scheduling, whereas up to 5 years of battery life can be achieved in the large-scale LTE-based IoT scenario by employing the proposed Discontinuous Reception (DRX) mechanism.

Sensor Enhanced FiWi Access Networks: Recently, the integration of wireless and fiber optic sensors into FiWi access networks has begun to receive increasing attention. For example, in [73], a sensor enhanced fiber-wireless (SFWiWi) network based on EPON, WLAN, wireless sensor (WS), and fiber optic sensor (FOS) technologies was designed as a shared communications infrastructure for broadband access and smart grids. This paper examined the network dimensioning, architecture, and especially smart grid sensors by means of both qualitative and quantitative analyses. A total cost of ownership (TCO) model taking into account sensor lifetime ratio was developed to estimate total sensor-related cost and compare WSs with FOSs, while an energy saving model and a delay analysis of coexisting broadband and smart grid traffic were presented to evaluate the performance of an energy conservation scheme for SFWiWi networks. Results showed that with their extremely long lifetime and ability to sustain in harsh environments, FOSs are superior to WSs when advanced interrogation techniques are deployed to reduce their total cost. Otherwise, WS deployment is suggested. The proposed scheme was able to achieve more than 89% of energy savings, while maintaining low delay for both broadband and smart grid traffic in typical scenarios. Further, this approach was elaborated in [40] with a Field Programmable Gate Array (FPGA) based design of the ECO-SFWiWi network. A pre-synthesis demonstration was performed in addition to hardware emulation results, which were compared with analytical results to verify the correctness and effectiveness of the proposed solution.

Machine-to-Machine Communications over FiWi Access Networks: Another important trend in FiWi research is to enhance FiWi networks to support machine-to-machine (M2M) communications. To cope with the unprecedented acceleration of M2M services over cellular networks, the authors in [75] envisioned an energy-efficient converged PON-4G network that combines the high-capacity and reliability of PON technology with the flexibility and cost savings of LTE to support M2M applications. In particular, this paper proposed a power saving scheme that unifies the cyclic sleep mechanism defined for PON's ONUs and the DRX mechanism defined for LTE's user

equipments (UEs) to reduce the overall power consumption in the envisioned network. Further, this analytical framework was extended in [76]. It was shown that the battery life of battery-constrained M2M devices is significantly prolonged by extending the DRX cycle, while the energy consumption of the backhaul was minimized by incorporating ONU power-saving modes into the DBA process of the optical backhaul. The introduction of a sleep-to-active state in the proposed scheme helps achieve an improved DRX performance. It was shown that the machine-type device (MTD) battery life can be extended to up to several years by configuring the DRX parameters appropriately.

1.1.5 Cloud Computing, Mobile Cloud Computing, and Cloudlet

Cloud Computing: A plethora of definitions and interpretations of cloud computing from both academia and industry exist. Among them, Rimal *et al.* [13] defined cloud computing as “a model of service delivery and access, where dynamically scalable and virtualized resources are provided as a service over the Internet.” Cloud computing allows access to infrastructure (e.g., Amazon Elastic Compute Cloud (EC2)), platforms (e.g., Google AppEngine), and software (e.g., Salesforce.com), which are provided by cloud providers (e.g., Amazon, Google, Microsoft), third party service providers, and service and application providers at low cost in an on-demand fashion at any time, from anywhere. Some widely used cloud-based applications/services include Apple’s iCloud, Apple’s Siri, YouTube, Dropbox, Gmail, Office 365, just to name a few. For illustration, Fig. 1.5 shows an overview of cloud computing.

From a technological point of view, cloud computing has not only become an integral part of the most highly impactful technologies today such as mobile Internet, automation of knowledge, IoT, and big data but also has been identified as the next-generation computing infrastructure. From an economic point of view, clouds offer tremendous benefits. For instance, McKinsey Global Institute reported that the total economic impact of cloud technology could be \$1.7 trillion to \$6.2 trillion annually in 2025 and the proliferation and sophistication of cloud services could become a major driving force in making entrepreneurship more feasible in the coming decade [77]. The key advantages of cloud computing are summarized in the following (see also Fig. 1.5).

- **Dynamic provisioning:** Clouds provide dynamic on-demand provisioning of resources. They create a virtual machine instance whenever end-users need it and terminate it whenever they do not.

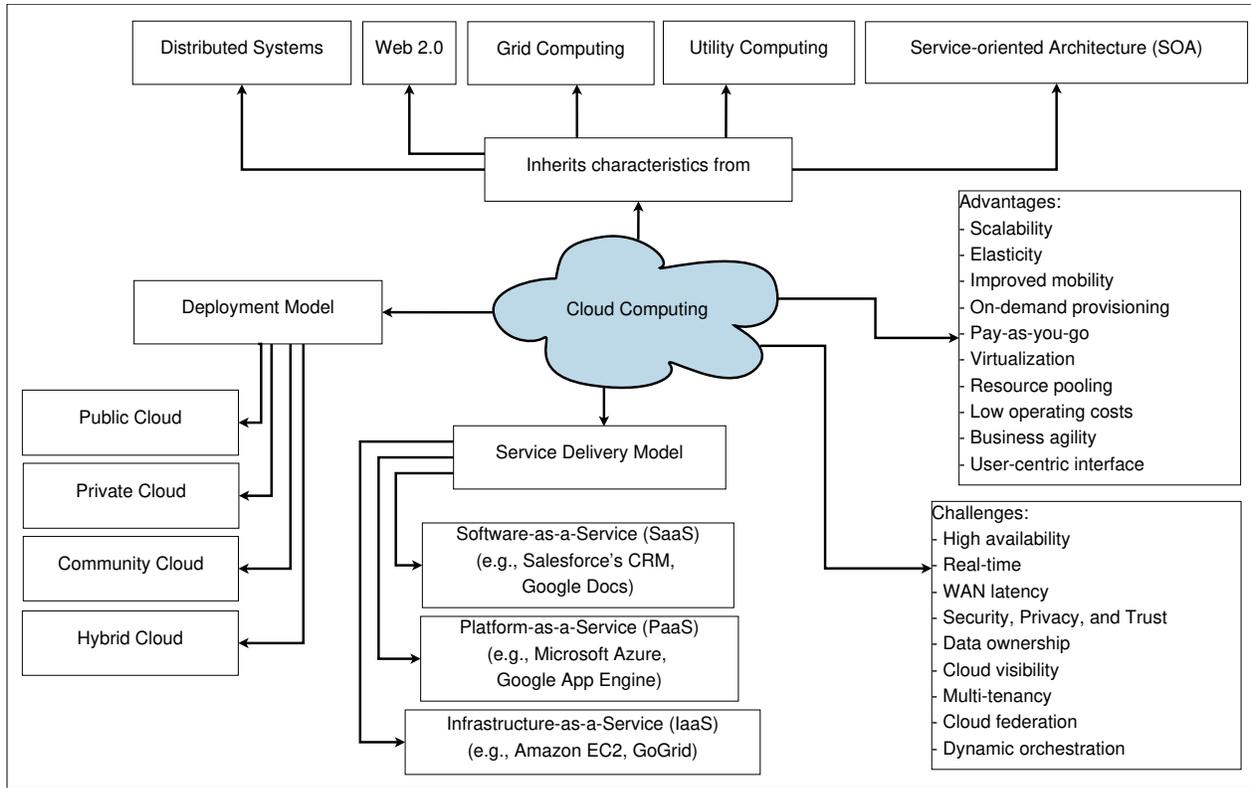


Figure 1.5: Overview of cloud computing: Characteristics, challenges, service delivery, and deployment models.

- **Elasticity:** It allows dynamic scaling, i.e., provisioning and deprovisioning computing resources based on actual demand. Elasticity in cloud infrastructure enables the hypervisor to create virtual machines or containers with the resources to meet the real-time demand.
- **Scalability:** It is a desirable property of any system, network or process, which indicates its ability to either handle an increased workload, or its potential to be enlarged in order to accommodate that growth. There are two types of scalability: vertical and horizontal scalability. Vertical scalability (scaling-up) is accomplished by adding more resources (e.g., more memory, additional CPU, or moving an application to a bigger virtual machine (VM)) to a single node in a system. Horizontal scalability (scaling-out), on the other hand, is the ability to increase capacity by adding multiple nodes (VMs) to a system and divide the workload among them.
- **Multi-tenancy:** It is the ability to share resources and costs to support a variety of applications and a large number of end-users [22, 64]. The challenges and benefits of multi-tenant cloud computing environments are discussed in greater detail in Chapter 3.

- **Ease of Integration:** In cloud computing environments, multiple services from different third-parties can be integrated easily. In addition, open application programming interfaces (APIs) allow telecom operators to create services needed to compete with over-the-top (OTT) players.

Beside the aforementioned advantages, cloud computing poses many challenges, including large and variable wide area network (WAN) latency, real-time response, privacy, and security, among others, as shown in Fig. 1.5. Cloud computing comprises three service models: *Infrastructure-as-a-Service (IaaS)*, *Platform-as-a-Service (PaaS)*, and *Software-as-a-Service (SaaS)* (see Fig. 1.5). IaaS is the delivery of resources (e.g., processing, storage, networks) as a service over the Internet. PaaS provides developers with a platform, including all the systems and environments comprising the end-to-end life cycle of software development. SaaS is also known as application service provider, where all users share common resources and a single instance of the code. Further, as shown in Fig. 1.5, cloud computing is composed of four deployment modes: *private clouds*, *public clouds*, *hybrid clouds*, and *community cloud*. In public clouds, the cloud is owned by one provider (e.g., Amazon EC2, Rackspace) and used by many end-users, where resources are dynamically provisioned on a fine-grained, self-service basis over the Internet, via web services/applications. In contrast, in private clouds, the provider (e.g., Eucalyptus, OpenStack) and end-users are typically associated with the same organization, where data and processes are managed within the organization without restrictions in terms of network bandwidth, security exposures, and legal requirements. Hybrid clouds consist of multiple internal and/or external providers (e.g., Skytap, Elastra). Community clouds are solely owned by a specific community or organization that has shared concerns such as security or mission.

Mobile Cloud Computing (MCC): The next evolution of cloud computing is the integration of cloud computing into the mobile environment, giving rise to the so-called mobile cloud computing (MCC) paradigm, where mobile devices access cloud services wirelessly and/or data processing and storage that are beyond their own capabilities are moved to a powerful centralized cloud computing infrastructure. In fact, MCC inherits all the characteristics of cloud computing, including scalability, improved reliability, dynamic provisioning, reduced cost, and ease of integration [14]. The resource-intensive applications/tasks of mobile devices can be offloaded to MCC, thereby extending the battery life of battery-constrained mobile devices. However, not all applications (e.g., cognitive assistance applications, face detection, real-time video analytics) are suited to run at the centralized cloud computing infrastructure because of WAN latency and potentially low bandwidth.

Some applications require real-time response, low latency, and high bandwidth. Bringing cloud-like capabilities (e.g., cloudlets [15, 16]) to the edge of the network to meet such requirements is an emerging concept, as discussed in the following.

Cloudlet: There is now growing awareness among industry players of reaping the benefits of openness and end-to-end Internet design principles at the early stages of the mobile-cloud convergence by extending today’s unmodified cloud to a second level consisting of self-managed data centers with no hard state called *cloudlets* [15, 16]. Cloudlets may contain cached state from the cloud and may buffer data originating from mobile devices. Cloudlets differ from conventional clouds in that they are decentralized entities that are located at the edge of the Internet and are well connected to it (e.g., via fiber), just one wireless hop away from associated mobile devices, thus enabling the support of new applications that are both compute intensive and latency sensitive, for example, cognitive assistance, augmented reality (AR), or face recognition and navigation for emerging cloud robotics [17]. The resultant two-level cloud-cloudlet architecture leverages both centralized and distributed cloud resources and services, whereby cloudlets are deployed much like WiFi access points (APs) today and mobile devices can gracefully degrade to a fall-back mode that involves a distant cloud, if no cloudlet is available nearby [15]. The deployment of cloudlets is not limited to WiFi APs. However, according to a very recent study in [18], LTE latencies are slightly higher compared to WiFi in computation offloading for the considered applications such as face recognition, AR, and physics-based computer graphics, among others. Cloudlets are best realized by relying on R&F networks and deployed at the edge of FiWi networks, whereby the integrated ONU-AP or ONU-mesh portal point (MPP) is referred to as the *edge of FiWi networks*, to provide low-latency and high-bandwidth, as discussed in greater detail in Chapters 4-6.

Although the commercial deployment of cloudlets is not yet a reality, there is growing industrial interest in cloudlet-like solutions. An example is Nokia’s Radio Applications Cloud Server (RACS) [78], a network element in the RAN that is connected to the 4G LTE base station (eNB). Other similar efforts in this direction include Micro Data Center (MDC)³ [79], ETSI Mobile-Edge Computing (MEC) [42] (see Chapters 5 and 6), and IOx⁴ [80], which is Cisco’s implementation of

³An MDC acts as a stand-alone system, which is a combination of hardware (single or multiple rack/cabinet that houses rack-mounted servers, switches, storage, uninterruptible power supply (UPS), backup systems), software and cabling that serves as an end-to-end computer, store and networking hub [79].

⁴IOx is an application framework (a router operating system) for Fog computing that combines IoT application execution within the fog, secure connectivity with an industry-leading Internetwork Operating System (IOS), and reliable integration with IoT sensors and the cloud [80]. In fact, Fog computing extends the cloud computing closer to the things/devices that produce and act on IoT data. Any device with computing, storage, and network connectivity

Fog computing [81]. The fundamental design goal of cloudlet/Fog computing/MDC is to address similar issues, e.g., reducing WAN latency, providing high bandwidth to mobile users, among others. All these concepts identified that computing, storage, control, configuration, and management should be placed at the edge of communications networks to complement the current centralized cloud computing paradigm. The best way to achieve low latencies and energy savings of battery-constrained mobile devices is to offload compute-intensive tasks onto the edge network, as discussed in the following.

1.1.6 Mobile Data Offloading vs. Computation Offloading

Mobile Data Offloading: Mobile data offloading, also referred to as mobile cellular traffic offloading, is the use of complementary network communication technologies (e.g., WiFi or femtocell) to deliver mobile data traffic originally planned for transmission over cellular networks [82]. The benefits of WiFi offloading, a widely deployed type of mobile data offloading, have been explored in several studies, for example, [83–85].

WiFi offloading can be categorized into two groups: a) *user-initiated offloading*, that is, users decide when and how to offload, i.e., automatic offloading is not possible; b) *network-initiated offloading*, i.e., mobile operators make the offloading decision. It can be a short-term or long-term decision. Short-term policies can be employed for network-initiated offloading such as automatic handover from cellular to WiFi and traffic reroute. Long-term policies include carrier-grade WiFi integration with cellular through new standards and traffic reroute to a cellular operator’s core network. Recently, 3GPP has been involved in developing standards and protocols for data offloading. For instance, 3GPP defined the Access Network Discovery and Selection Function (ANDSF) [86] to trigger the handoff between different access technologies. It also proposed three alternative offloading mechanisms: Local IP Access (LIPA), Selected IP Traffic Offload (SIPTO) [87], and IP Flow Mobility (IFOM) [88]. IFOM is a method that combines access network discovery, interworking WLAN, and mobile IP, where the mobile packet core can be accessed via a WLAN network across a so-called Packet Data Gateway (PDG). Since IFOM uses Dual Stack⁵ Mobile IPv6 (DSMIPv6) [89], the seamless handover from mobile network to WLAN and vice-versa is realized by means of mo-

can be a fog node. For examples, industrial controllers, switches, routers, embedded servers, and video surveillance cameras.

⁵DSMIPv6 requires a dual-stack (IPv4 or IPv6) capable UE.

mobile IP. LIPA offloads traffic entirely from an operator’s RAN, backhaul, and core network without traversing the cellular network, whereas SIPTO extends LIPA to provide selected traffic routing options at different breakpoints (e.g., local gateway (L-GW), which is essentially PGW) along the network edge. A detailed description of IFOM, LIPA, and SIPTO can be found in [90].

Computation Offloading: Computation offloading is a technique for improving the execution efficiency of a distributed system by moving computation from a less capable system to one that is more capable. This capability can take the form of compute power, memory, system load, as well as battery life [91]. It is also known as surrogate computing or cyber foraging [92–94]. Computation offloading is also similar to the concept of SETI@home [95], where computation tasks are sent to surrogates for performing computation. However, note that SETI@home is large-scale public-resource computing, whereas computation offloading is typically small-scale computing.

Offloading decisions are usually made by analyzing parameters, including bandwidth, server speed, available memory, server load, and the amount of data exchanged between servers and mobile clients. The solutions include partitioning programs and predicting parametric variations in application behavior and execution environment. Offloading can be performed at various granularities, e.g., fine-grained, coarse-grained at the process, method/function, class, component, task, service, and application/program levels. Offloading decisions depend on various factors such as *Why* to offload (improve performance or save energy), *How* to decide offloading (i.e., static vs. dynamic), *What* mobile systems use offloading (e.g., laptops, PDAs, wearables, robots), types of applications (multimedia, text editors), and infrastructure for offloading (grid/cloud computing) [96]. There are four major steps in computation offloading, as discussed below.

- *Application Modeling:* Three types of application models, namely, procedure-call, service-invocation, and dataflow can be found in the state-of-the-art literature. Among them procedure-call is widely used. In a procedure-call model, an application is represented by a set of procedures (functions), where each procedure can call other procedures. Recent studies, for example, MAUI [97], CloneCloud [98], and ThinkAir [99] employ this approach. In service-invocation, an application comprises a set of services. In both approaches, a graph or tree can be used to represent the application, where a node represents the procedure/service and an edge represents the call relationship/services.

- **Profiling:** Device and network profiling involves collecting device and network information (e.g., CPU, memory state, network bandwidth), which is used to construct a cost model for the application. For instance, MAUI [97] profiles the energy consumption of each part of the application. CloneCloud uses a combination of static analysis and dynamic profiling to partition applications automatically at a fine granularity [98], whereas Odessa [100] does not profile the application, device, and network independently. Instead, it profiles them in runtime and data transmission time.
- **Optimization:** A mathematical optimizer is used to optimize the objective (e.g., total execution time, energy consumption) for a given application and cost model. The cost model includes completion time, data processing throughput, energy consumption, or any combination thereof. For instance, MAUI [97] optimizes the energy consumption of devices, CloneCloud [98] and ThinkAir [99] optimize the execution time or energy consumption depending on the programmer's choice, whereas Odessa [100] optimizes the makespan for data streaming applications.
- **Implementation:** The implementation applies a suitable approach to determine which tasks of an application are remotely executed on the cloud infrastructure. Three approaches are widely used, namely, client-server, VM migration, and mobile agents. Unlike in the traditional client-server architecture, client applications can be partitioned at different levels of granularity (e.g., class, method, task, thread). More specifically, a mobile client requests the server to execute a particular method with given arguments and the server returns the result of the method execution using Remote Procedure Call (RPC) or Remote Method Invocation (RMI) protocols. Interesting examples of client-server based computation offloading systems include Spectra [101], Chroma [94], [102], Cuckoo [103], MAUI [97], and Odessa [100]. In the VM migration approach, clients prepare an image of their devices and transmit it to the servers for execution. The VM migration approach is used in Slingshot [104], CloneCloud [98], ThinkAir [99], and Cloudlet [105]. In the mobile agent approach, the computation is migrated from mobile devices to servers, whereby mobile agents are autonomous programs that control their movement from machine to machine in a heterogeneous network [96].

1.2 Objectives

The objectives of this dissertation are as follows:

- Given the paramount importance of energy efficiency in both wired and wireless access networks, designing and implementing power-saving mechanisms to prolong battery life of battery-constrained devices while reducing energy consumption of the optical backhaul network is a significant problem to be investigated. Note that optical backhaul and wireless/cellular networks are not energy efficient since their low average utilization results in a significant waste of energy [12]. Therefore, end-to-end power-saving mechanisms considering both fiber and wireless segments are highly needed for FiWi access networks. More specifically, the first aim of this thesis is to design a novel resource management scheme in a coordinated and synchronized manner in order to minimize the total energy consumption in FiWi broadband access networks while not degrading network performance. The focus of this study is to find insights into the optimal energy saving obtained from sleep mode scheduling for a given network configuration and delay constraint.
- Although multi-tenancy allows cloud service providers to better utilize computing resources, supporting the development of more flexible services based on economy of scale and reducing infrastructural costs represent an important open research challenge. For instance, multi-tenant cloud computing poses unique challenges such as scalability, resource provisioning, and per-tenant service customization. Therefore, the second objective of this thesis is to design a resource management framework that consists of both an architecture and a novel scheduling policy with regard to multi-tenancy issues, especially scalability and shared resources for scheduling compute-intensive workflow applications.
- Mobile data offloading has not been fully explored from the perspective of FiWi access networks, in particular mobile cloud computing and fiber backhaul sharing in heterogeneous networks (HetNets). The third aim of this thesis is to develop a mobile data offloading framework with the objective of designing optimal economic strategies (e.g., minimizing energy consumption and maximizing payoffs).
- The fourth aim of this thesis is to design cloud and cloudlet enhanced FiWi access networks, thereby contributing to the realization of the emerging concept of MEC. It is important to note

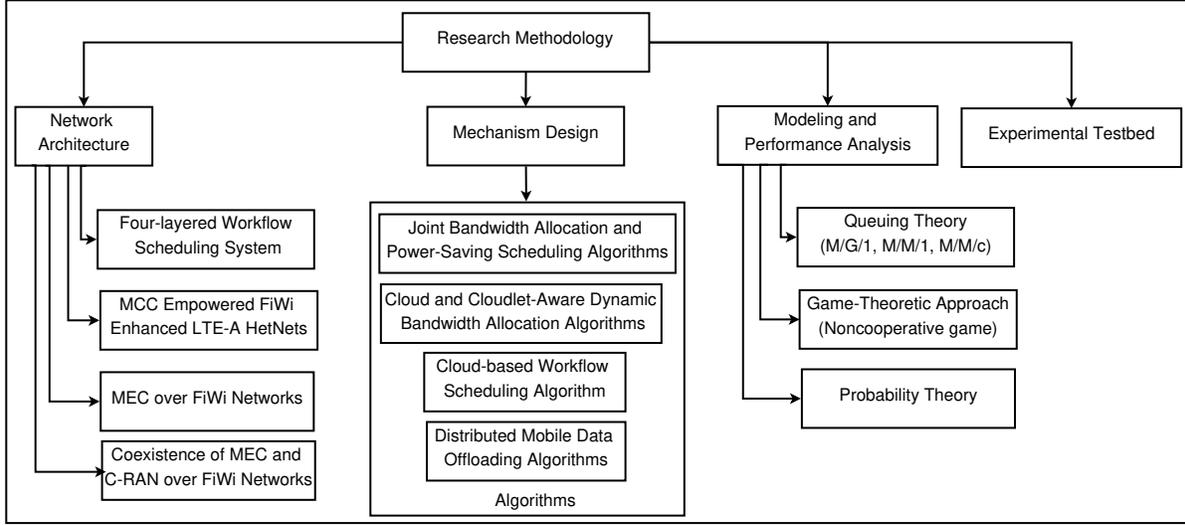


Figure 1.6: Research methodology.

that no existing study deals with the cloud/MEC coexistence as well as the coexistence of MEC and human to-human (H2H) traffic (e.g., triple-play traffic) in FiWi networks. Furthermore, there exist reliability issues of the optical fiber backhaul (e.g., fiber cuts) and MEC server (e.g., fault within the MEC platform, link failures to/from access point). As a result, the envisioned network may suffer from widespread service outages. The foremost goal of this part of the thesis is to design an architecture and develop a novel unified (one solution for both network segments to simplify the management of the integrated architecture) resource management scheme as well as to study their achievable performance gains.

- The final aim of this thesis is to explore emerging deployment scenarios of cloud-cloudlet enhanced FiWi access networks. Specifically, the primary goal is to elaborate on the concept of the emerging Tactile Internet and investigate the feasibility of Tactile Internet capable FiWi networks.

1.3 Research Methodology

The overall research methodology, involving architecture, mechanism design, as well as modeling and performance analysis, applied in this thesis is summarized in Fig. 1.6 and is briefly described in the following.

- *Architecture:* Novel architectures are developed for the workflow scheduling system, mobile data offloading, and cloud-cloudlet/MEC enhanced FiWi broadband access networks. A *top-down approach*, which is a common strategy for network design, is used for designing FiWi enhanced networks. In this approach, the network applications and service requirements are derived first and then the network is designed to support them. More specifically, the functionality of existing network technologies and protocols as well as resource management algorithms (e.g., dynamic bandwidth allocation (DBA) algorithms) are leveraged. Both mesh and conventional tree-based topologies are considered in the front-end of FiWi networks.
- *Mechanism design:* Novel algorithms are developed (see Chapters 2-6) for the operation of the proposed architectures. They include joint bandwidth allocation and power-saving scheduling algorithms, cloud-based workflow scheduling algorithm, distributed mobile data offloading algorithms, as well as cloud- and cloudlet-aware DBA algorithms.
- *Modeling and performance analysis:* Among the diverse set of analytical tools, queuing theory has been widely used for the delay analysis of both FiWi networks and cloud computing. Therefore, a comprehensive analysis based on queuing theory (e.g., M/G/1, M/M/1, M/M/c) is conducted. By means of comprehensive analysis and simulations the performance of the proposed solutions is investigated for a variety of different network configurations. Given that the game-theoretic approach has become a well-known mathematical tool in the design and analysis of wired/wireless communications, the mobile data offloading framework developed in this thesis is based on a noncooperative multi-level Stackelberg game-theoretic approach.
- *Experimental testbed:* An experimental testbed for MEC in cloudlet enhanced FiWi access networks is developed to validate the accuracy of analytical model via real-world measurements.

1.4 Contributions of the Thesis

This dissertation is a compilation of seven publications (six journal articles and one conference paper), which were published in high-calibre IEEE journals and conference proceedings. The key contributions of this thesis that achieve the objectives mentioned in Section 1.2 are discussed in detail in the following.

1.4.1 Energy-Aware Resource Management Scheme for FiWi Access Networks

This work has been published in the following journal and the contributions are summarized in the following:

[106] D. Pham Van, B. P. Rimal, M. Maier, and L. Valcarenghi. ECO-FiWi: An Energy Conservation Scheme for Integrated Fiber-Wireless Access Networks. *IEEE Transactions on Wireless Communications*, vol. 15, no. 6, pp. 3979-3994, June 2016.

- **A Novel Energy Conservation Scheme:** To maximize the overall energy efficiency without compromising delay performance of integrated FiWi access networks, power-saving modes of all ONUs, APs, and wireless stations (STAs) are synchronized and incorporated into the EPON DBA process. An energy conservation scheme for FiWi networks (ECO-FiWi) takes into account traffic status of both upstream (US) and downstream (DS) transmissions in the joint process of allocating bandwidth and scheduling power-saving mode for the FiWi network elements in an attempt to make their time being active proportional to their actual traffic loads.
- **Energy Saving Model:** An energy saving model is developed to provide operators with an overview of parameters and quantities that determine the potential energy saving for a given network configuration.
- **Comprehensive Analysis of Frame Delay:** To evaluate the network performance, a comprehensive analysis of mean US and DS frame delays based on an M/G/1 queuing system is presented, accounting for both optical backhaul and wireless front-end network segments. Apart from mean delays, upper bounds on delays are also analyzed to provide an estimate of maximum delays in ECO-FiWi networks. Importantly, the combination of the energy saving model and analysis of frame delays provides invaluable insights into the quantification of the optimal energy saving for a given network configuration and delay constraint.

1.4.2 Workflow Scheduling in Multi-Tenant Cloud Computing

This work has been published in the following journal and the key contributions are summarized in the following:

[64] B. P. Rimal and M. Maier. Workflow Scheduling in Multi-Tenant Cloud Computing Environments. *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 1, pp. 290-304, Jan. 2017.

- **Four-layered Architecture of Workflow Scheduling System:** This work proposes a novel four-layered architecture for a workflow scheduling system in a multi-tenant cloud computing environment, which represents a cost-effective solution for executing compute-intensive workflow applications. The first layer (tenant) consists of a workflow creator/composer. The second layer (middleware) consists of workflow dispatcher, service queue, workflow scheduler, shared pool resource information, tenant information and performance repository, QoS monitor, and executor. The third layer is the virtual infrastructure layer, whereas the fourth layer consists of the physical infrastructure layer.
- **A Novel Cloud Workflow Scheduling Algorithm:** This work proposes a novel cloud workflow scheduling algorithm to deal with both structured and unstructured workflow scheduling in multi-tenant cloud computing environments, which maximizes cloud resource usage, minimizes the expected makespan, and also minimizes the scheduling execution time, workflow execution costs, and total expected tardiness.
- **Extensive Simulation-based Studies:** The performance of the proposed scheduling policy is evaluated by means of extensive simulations and compared with state-of-the-art algorithms, i.e., First Come First Served (FCFS), EASY (Extensible Argonne Scheduling sYstem) Back-filling, and Minimum Completion Time (MCT) scheduling policies.
- **Proof-of-Concept Experiment:** The proposed scheduling policy is implemented in real-world scientific complex workflow applications (e.g., SIPHT, CyberShake) to benchmark the performance. The proof-of-concept experiment shows the scalability and effectiveness of the proposed solution.

1.4.3 Cloud and Edge Computing Enhanced FiWi Access Networks

This work has been published in the following journals and conference proceedings and their contributions are summarized in the following:

[107] B. P. Rimal, D. Pham Van, and M. Maier. Cloudlet Enhanced Fiber-Wireless Access Networks for Mobile-Edge Computing, *IEEE Transactions on Wireless Communications*, IEEE Xplore Early Access, DOI: 10.1109/TWC.2017.2685578.

- **MEC Enabled FiWi Access Network:** From an architectural perspective, for the first time, this thesis introduces an MEC enabled FiWi access network architecture, which represents a cost-effective and reliable solution for enabling a new class of low-latency and/or resource intensive MEC applications.
- **Cloudlet-Aware Resource Management:** Three novel cloudlet-aware DBA algorithms (C-DBAs) are proposed, which incorporate offloading operations into the underlying EPON/10G-EPON DBA process to effectively handle the coexistence of both H2H and MEC traffic as well as to reduce delay and prolong the battery life of edge devices.
- **Comprehensive Analytical Framework:** This part of the thesis develops a comprehensive analytical framework to evaluate the network performance in terms of packet delay of both H2H and MEC traffic, response time efficiency, offload gain-overhead ratio, energy efficiency, and device battery life.

[108] B. P. Rimal, D. Pham Van, and M. Maier. Mobile-Edge Computing Empowered Fiber-Wireless Access Networks in the 5G Era. *IEEE Communications Magazine*, vol. 55, no. 2, pp. 192-200, Feb. 2017.

- **Design of MEC over FiWi Networks:** This dissertation is the first to formally provide a comprehensive inquiry into the design of MEC over FiWi networks. The concept of MEC over FiWi is elaborated by studying typical service scenarios. Key technical challenges of implementing MEC are then identified and discussed in detail.
- **MEC over FiWi Network Design Scenarios:** Given that WLAN and 4G LTE/LTE-A are among the most common RAN technologies, three foreseeable design scenarios are proposed: (a) MEC over WLAN-based FiWi; (b) MEC over 4G LTE-based FiWi; and (c) coexistence of MEC and C-RAN in FiWi enhanced LTE-A HetNets.
- **A Novel Unified Resource Management Scheme:** A unified resource management scheme for MEC over Ethernet-based FiWi networks is presented that takes the network integration, H2H/MEC coexistence, and resource management issues into consideration. The scheme jointly allocates bandwidth for transmissions of both conventional broadband traffic and MEC data in a TDMA fashion. The obtained results show the significant benefits of MEC over FiWi networks. Importantly, we show that a mean MEC packet delay of 20.50 ms can be obtained for the considered scenario, while allowing for efficient H2H/MEC coexistence

without degrading the network performance. Further, the battery life of edge devices can be prolonged up to 11.30 hours by employing the proposed solution.

[109] B. P. Rimal, D. Pham Van, and M. Maier. Mobile-Edge Computing vs. Centralized Cloud Computing in Fiber-Wireless Access Networks, *Proc., IEEE INFOCOM, Workshop on 5G & Beyond - Enabling Technologies and Applications*, pp. 991-996, San Francisco, CA, USA, Apr. 2016.

[110] B. P. Rimal, D. Pham Van, and M. Maier, Mobile-Edge Computing vs. Centralized Cloud Computing Over a Converged Fiber-Wireless Broadband Access Network, *IEEE Transactions on Network and Service Management*, *in revision*.

- **Two-level Cloud-MEC Architecture:** A two-level cloud-MEC architecture is proposed for FiWi access networks. The architecture not only creates an infrastructure platform for mobile operators or third-party service providers to offer MEC services from the edge of networks, but also represents a cost-effective solution for supporting diverse types of traffic (H2H and both cloud and MEC).
- **Two-layered TDMA-based Resource Management Scheme:** Since no existing study deals with the cloud/MEC coexistence as well as the coexistence of MEC and human-to-human (H2H) traffic (e.g., triple-play voice, video, and data traffic) in FiWi networks, in this work a two-layered TDMA-based resource management scheme is proposed to effectively manage H2H and both cloud traffic types (cloud and MEC) at the same time. Both MEC and cloud traffic are scheduled outside the transmission slot of H2H traffic.
- **Performance Metrics:** Key performance metrics to evaluate MEC are defined, which are instrumental in understanding the interplay between FiWi broadband access network, cloud, and MEC.
- **Comprehensive Analytical Framework:** This part of the thesis presents a comprehensive analytical framework to analyze the performance of the proposed unified resource management scheme. In addition, a probabilistic analytical model is developed to estimate the survivability of the optical backhaul, wireless front-end network, and MEC.

1.4.4 Tactile Internet

This work has been published in the following journal and the main contributions are summarized in the following:

[111] M. Maier, M. Chowdhury, B. P. Rimal, and D. Pham Van. The Tactile Internet: Vision, Recent Progress, and Open Challenges. *IEEE Communications Magazine*, vol. 54, no. 5, pp. 138-145, May 2016.

- **Tactile Internet Vision:** To explore the role of MEC over FiWi in the emerging concept of the Tactile Internet, the vision of the Tactile Internet is elaborated by briefly reviewing its anticipated impact on society and important design guidelines. To facilitate a better understanding of the Tactile Internet, the three lenses of IoT, 5G, and the Tactile Internet are presented, through which a view on the commonalities and differences between IoT, 5G, and the Tactile Internet is provided. Further, a comprehensive survey is provided on the recent progress and enabling technologies proposed for the Tactile Internet. In addition, several open challenges are identified and future research directions are discussed in detail.
- **Bringing the Tactile Internet Vision Closer to Reality:** The use of cloudlets/MEC at the edge of FiWi networks can achieve a significantly reduced end-to-end latency and enhanced overall network performance, thereby bringing the vision of the Tactile Internet closer to reality.

1.5 List of Publications

Publications included in this dissertation

This dissertation includes material extracted from the following publications:

Journals

1. B. P. Rimal, D. Pham Van, and M. Maier. Cloudlet Enhanced Fiber-Wireless Access Networks for Mobile-Edge Computing. *IEEE Transactions on Wireless Communications*, IEEE Xplore Early Access, DOI: 10.1109/TWC.2017.2685578.
2. B. P. Rimal and M. Maier. Workflow Scheduling in Multi-Tenant Cloud Computing Environments. *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 1, pp. 290-304, Jan. 2017.

3. B. P. Rimal, D. Pham Van, and M. Maier. Mobile-Edge Computing Empowered Fiber-Wireless Access Networks in the 5G Era. *IEEE Communications Magazine*, vol. 55, no. 2, pp. 192-200, Feb. 2017.
4. B. P. Rimal, D. Pham Van, and M. Maier. Mobile-Edge Computing vs. Centralized Cloud Computing Over a Converged Fiber-Wireless Broadband Access Network. *IEEE Transactions on Network and Service Management*, in revision, Jan. 2017.
5. D. Pham Van, B. P. Rimal, M. Maier, and L. Valcarenghi. ECO-FiWi: An Energy Conservation Scheme for Integrated Fiber-Wireless Access Networks. *IEEE Transactions on Wireless Communications*, vol. 15, no. 6, pp. 3979-3994, June 2016
6. M. Maier, M. Chowdhury, B. P. Rimal, and D. Pham Van. The Tactile Internet: Vision, Recent Progress, and Open Challenges. *IEEE Communications Magazine*, vol. 54, no. 5, pp. 138-145, May 2016.

Conference Proceedings

1. B. P. Rimal, D. Pham Van, and M. Maier. Mobile-Edge Computing vs. Centralized Cloud Computing in Fiber-Wireless Access Networks. *Proc., IEEE INFOCOM, Workshop on 5G & Beyond - Enabling Technologies and Applications*, pp. 991-996, San Francisco, CA, USA, Apr. 2016.

Publications not included in this dissertation

The following papers have also been published or submitted during the Ph.D. studies. However, their content is not included in this dissertation for compactness.

Journals

1. B. P. Rimal, M. Maier, and M. Satyanarayanan. Experimental Testbed for Edge Computing in Fiber-Wireless Broadband Access Networks. *Submitted to IEEE Communications Magazine*, 2017.

2. B. P. Rimal and M. Maier. Mobile Data Offloading in FiWi Enhanced LTE-A Heterogeneous Networks. *IEEE/OSA Journal of Optical Communications and Networking*, in revision, Mar. 2017.
3. D. Pham Van, B. P. Rimal, J. Chen, P. Monti, L. Wosinska, and M. Maier. Power-Saving Methods for Internet of Things over Converged Fiber-Wireless Access Networks. *IEEE Communications Magazine*, vol. 54, no. 11, pp. 166-175, Nov. 2016.
4. D. Pham Van, B. P. Rimal, M. Maier, and L. Valcarengi. Design, Analysis, and Hardware Emulation of a Novel Energy Conservation Scheme for Sensor Enhanced FiWi Networks (ECO-SFiWi). *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 5, pp. 1645-1662, May 2016.
5. D. Pham Van, B. P. Rimal, S. Andreev, T. Tirronen, and M. Maier. Machine-to-Machine Communications over FiWi Enhanced LTE Networks: A Power-Saving Framework and End-to-End Performance. *IEEE/OSA Journal of Lightwave Technology*, vol. 34, no. 4, pp. 1062-1071, Feb. 2016.
6. M. Maier and B. P. Rimal. The Audacity of Fiber-Wireless (FiWi) Networks: Revisited for Clouds and Cloudlets (**Invited Paper**). *China Communications, Feature Topic on Optical Interconnection Networks for Cloud Data Centers*, vol. 12, no. 8, pp. 33-45, Aug. 2015.
7. A. Belgana, B. P. Rimal, and M. Maier. Open Energy Market Strategies in Microgrids: A Stackelberg Game Approach Based on a Hybrid Multiobjective Evolutionary Algorithm. *IEEE Transactions on Smart Grid*, vol. 6, no. 3, pp. 1243-1252, May 2015.

Book Chapter

1. B. P. Rimal, A. Belgana, and M. Maier. Game-Theoretic Approach for Energy Trading in Smart Grids. *Smart Grid: Networking, Data Management, and Business Models*, CRC, pp. 387-403, Apr. 2016.

Conference Proceedings

1. D. Pham Van, B. P. Rimal, and M. Maier. Fiber Optic vs. Wireless Sensors in Energy-Efficient Integrated FiWi Smart Grid Networks: An Energy-Delay and TCO Comparison. *Proc., IEEE INFOCOM*, pp. 1-9, San Francisco, CA, USA, Apr. 2016.
2. D. Pham Van, B. P. Rimal, and M. Maier. Power-Saving Scheme for PON LTE-A Converged Networks Supporting M2M Communications. *Proc., IEEE International Conference on Ubiquitous Wireless Broadband (ICUWB), Workshop on Fiber-Wireless (FiWi) Access Networks*, pp. 1-5, Montréal, Canada, Oct. 2015.
3. A. Belgana, B. P. Rimal, and M. Maier. Multi-Objective Pricing Game Among Interconnected Smart Microgrids. *Proc., IEEE Power & Energy Society General Meeting*, pp. 1-5, National Harbor, MD, USA, July 2014.

1.6 Dissertation Outline

This dissertation is composed of a number of publications that were realized within the scope of this doctoral thesis. The selected publications provide an integral and consistent overview of the work performed. The remainder of this dissertation is structured as follows.

Chapter 2 presents an energy-aware resource management scheme for FiWi access networks. The achievable energy saving was analytically modeled accounting for both wireless and optical network components. The chapter then presents an M/G/1 queuing-based analysis of the mean end-to-end frame delay for both upstream and downstream traffic as well as their upper bounds. Further, comprehensive numerical results are presented.

Chapter 3 explores a workflow scheduling framework in a multi-tenant cloud computing environment. In particular, the chapter describes the concept of cloud workflow and presents the proposed reference architecture of workflow scheduling. Further, the proposed cloud workflow scheduling algorithm is presented in greater detail. Then, extensive simulations and a proof-of-concept experiment under typical deployment scenarios as well as future research issues are presented in detail in the remainder of this chapter.

Chapter 4 investigates cloudlet enhanced FiWi access networks for MEC. It first presents the envisioned network architecture. After presenting the architecture, the chapter provides an overview of the proposed resource management scheme and then presents the three algorithms in greater detail. The performance of the proposed schemes are analyzed and extensive results and obtained findings are presented.

Chapter 5 explores the possibilities of enabling integrated FiWi access networks to offer MEC capabilities. More specifically, several design scenarios of MEC over FiWi networks for typical RAN technologies are investigated accounting for both network architecture and enhanced resource management. The performance of MEC over Ethernet-based FiWi networks in terms of delay, response time efficiency, and battery life of edge devices is then analyzed. Afterwards, the obtained results and discussions are presented.

Chapter 6 investigates the achievable performance gains of centralized cloud and MEC enabled integrated FiWi access networks. In particular, the chapter presents a novel unified resource management scheme by incorporating both centralized cloud and MEC offloading capabilities into the underlying FiWi dynamic bandwidth allocation process. Further, this chapter presents an analytical framework that models packet delay, response time efficiency, gain-offload overhead ratio, and communication-to-computation ratio for both cloud and broadband access traffic. In addition, this chapter develops a probabilistic survivability analysis to assess the impact of both fiber cuts and MEC server failures. Thereafter the obtained results are presented.

Chapter 7 explores potential application scenarios of cloud/cloudlet enhanced FiWi access networks. In particular, it elaborates on the Tactile Internet vision and presents commonalities and subtle differences between the Tactile Internet, the Internet of Things, and 5G networks. After briefly reviewing its anticipated impact on society and infrastructure requirements, this chapter then provides the recent progress and enabling technologies proposed for the Tactile Internet. In addition, several open challenges and future research directions are identified and discussed in detail.

Finally, chapter 8 concludes the dissertation with a summary of the key findings. The chapter also briefly discusses potential areas for future research that may build upon the work in this dissertation and is worth being explored in the future.

Chapter 2

Energy-Aware Resource Management Scheme for Integrated Fiber-Wireless Access Networks

I have equally contributed to this work. More specifically, my key contributions were as follows: a) equally involved in finalizing Fig. 2.2, b) involved to verify, write, and finalize the proposed algorithms and their descriptions as well as the descriptions of the protocol in Section 2.2.2, c) major contributions to the mathematical modeling of upstream and downstream frame delays, and d) equally contributed in Section 2.5 and 2.6.

2.1 Introduction

In wireless and cellular networks, capacity-limited backhaul links, i.e., network segments from the edges into the core network, are identified as one of the major challenges [112]. The integration of wireless and optical broadband access networks via a common backhaul infrastructure is a promising solution for network operators to reduce their operational and capital expenditures

This chapter is based on the following publication:
[106] D. Pham Van, B. P. Rimal, M. Maier, and L. Valcarengi. ECO-FiWi: An Energy Conservation Scheme for Integrated Fiber-Wireless Access Networks. *IEEE Transactions on Wireless Communications*, vol. 15, no. 6, pp. 3979-3994, June 2016.

(OPEX/CAPEX). Among different options, optical fiber technologies have been widely viewed as the ultimate solution for the backhaul of broadband access networks due to their high capacity, reliability, and low latency characteristics [113]. Integrated Fiber-Wireless (FiWi) access networks composed of an optical fiber backhaul segment and a wireless/cellular front-end segment aim at taking full advantage of wired and wireless technologies to provide broadband services for not only mobile users but also fixed subscribers [114]. By combining the reliability, robustness, and high capacity of the optical backhaul with the flexibility, ubiquity, and cost savings of the wireless front-end, the resultant FiWi networks potentially help remove the traditional barriers between coverage-centric wireless networks and capacity-centric optical fiber networks. Even though many FiWi architectural studies exist on the integration of optical access networks with different wireless technologies, e.g., WiMAX or Long Term Evolution (LTE), the most widely considered FiWi network architecture is based on low-cost simple Ethernet passive optical network (i.e., IEEE 802.3ah EPON or IEEE 802.3av 10G-EPON) fiber backhaul and Wireless Local Area Network (WLAN) front-end networking technologies (i.e., IEEE 802.11b/g/n/s) [114].

Most of previous studies on FiWi access networks focused on architecture design, survivability, and guaranteed quality of service (QoS), as well as their role in other critical infrastructures, e.g., smart grids [19]. However, the energy efficiency aspect of FiWi access networks, which is of paramount importance not only to prolong battery life of wireless devices but also to reduce OPEX for network operators has not been investigated in sufficient detail so far. Meanwhile, energy efficiency solutions for optical backhaul networks have widely been sought since access networks are known to consume much more energy than needed to accommodate actual traffic loads. It was shown that backhaul networks can account for up to 50% of the total power consumption of wireless networks [20]. Nevertheless, access network devices (e.g., ONUs) are highly under-utilized and thus being idle most of their time [21], resulting in a huge amount of wasted energy. Moreover, it is desirable to consolidate and harmonize energy efficiency strategies with resource allocation mechanisms by taking into consideration both optical backhaul and wireless front-end segments of a FiWi access network in order to achieve high energy efficiency, while providing end-users with a satisfactory QoS experience.

A popular energy saving method in communication networks is to turn network components off when they are idle. In WLAN, the power save mode (PSM) was initially defined in IEEE 802.11a/b to save wireless station (STA) energy with sleep (doze) mode operation. The PSM scheme was then

enhanced in follow-up WLAN amendments such as IEEE 802.11e/s/n and extensively studied in the literature [115]. Besides, the standards also support the doze mode operation of the access point (AP)'s wireless interface by using network allocation vector (NAV) blocking to prevent access from wireless devices when it is in doze mode [116]. Similarly, in EPON, the optical network unit (ONU) cyclic sleep, that is, turning off the whole ONU transceiver (i.e., switches off its laser) or only its transmitter in a cyclic fashion, was defined in the ITU-T G.Sup45 and IEEE P1904.1 SIEPON standard and widely investigated in the literature, e.g., [21]. There are several studies addressing energy efficiency in FiWi access networks with power-saving modes [117–121]. Most of them target only ONU sleep mode [117, 118, 120], while leaving wireless front-end network components always active for routing wireless traffic to active ONUs. In these studies, the common goal is to minimize the number of active ONUs as long as there are alternative routes for traffic destined to the sleeping ONUs. The combination of ONU sleep and power-saving modes for wireless devices, e.g., routers or client stations, has been studied in [119, 121, 122]. However, these and other existing solutions do not look into the details of how to incorporate the power-saving modes into the dynamic bandwidth allocation (DBA) process in the optical backhaul, which might result in inefficiency of either or both resource management and energy mechanisms.

This chapter proposes a novel Energy COnservation scheme for EPON-WLAN based FiWi access networks (ECO-FiWi) that aims at jointly scheduling power-saving modes for not only wireless devices (i.e., STAs and APs) in the wireless front-end but also ONUs in the optical backhaul in a coordinated and synchronized manner in order to minimize the total energy consumption. ECO-FiWi is ONU sleep-centric in that power-saving modes of the wireless devices are scheduled based on the ONU operation mode, which in turn strictly follows the cyclic polling-based operation of the EPON DBA process. To improve overall network performance, ECO-FiWi considers both downstream (DS) and upstream (US) traffic conditions when allocating bandwidth and scheduling power-saving modes for FiWi network elements. Instead of minimizing the number of active ONUs, ECO-FiWi exploits the idleness of ONUs outside their TDMA timeslots to put them and their integrated AP modules into sleep mode to save energy. During an EPON timeslot, the ONU-AP schedules transmission subslots for its associated STAs also in a TDMA manner, thus allowing them to enter sleep mode outside their subslots to prolong their battery life. Note that in addition to potential energy efficiency gains, using TDMA to schedule transmissions for STAs also helps avoid collisions and thereby improve network performance compared to contention-based carrier

sense multiple access with collision avoidance (CSMA/CA) protocol [115]. In the remainder of this chapter, the FiWi access network featuring the proposed scheme is henceforth termed ECO-FiWi network.

The main contributions of this chapter are as follows. First, to the best of authors' knowledge, for the first time, to maximize the overall energy efficiency without compromising delay performance of integrated FiWi access networks, power-saving modes of all ONUs, APs, and STAs are synchronized and incorporated into the EPON DBA process. ECO-FiWi takes into account traffic status of both US and DS transmissions in the joint process of allocating bandwidth and scheduling power-saving mode for the FiWi network elements in an attempt to make their time being active proportional to their actual traffic loads. Second, an energy saving model is developed to provide operators with an overview of parameters and quantities that determine the potential energy saving for a given network configuration. Third, to evaluate the network performance, a comprehensive analysis of mean US and DS frame delays based on an M/G/1 queuing system is presented, accounting for both optical backhaul and wireless front-end network segments. Apart from mean delays, upper bounds on delays are also analyzed to provide an estimate of maximum delays in ECO-FiWi networks. Importantly, the combination of the energy saving model and analysis of frame delays provides invaluable insights into the quantification of the optimal energy saving for a given network configuration and delay constraint.

The remainder of the chapter is structured as follows. The proposed ECO-FiWi scheme is described in greater detail in Section 2.2. The energy saving model is presented in Section 2.3. Section 2.4 presents the analysis of mean end-to-end frame delays for both US and DS traffic as well as their upper bounds. Section 2.5 reports on results and obtained findings. Finally, Section 2.6 concludes the chapter.

2.2 Energy Conservation Scheme for FiWi Access Networks

2.2.1 Network Architecture

Fig. 2.1 depicts the considered FiWi network architecture based on EPON and WLAN technologies. The optical backhaul consists of an EPON with its typical tree-and-branch topology. The optical

technologies, depending on deployment scenarios, other wireless/cellular technologies such as 3G and/or 4G (i.e., WiMAX and LTE) can be integrated into the wireless front-end, allowing for heterogeneity of converged access networks.

The power-saving techniques for different FiWi network devices mentioned previously are also illustrated in Fig. 2.1. Note that the WMN in the front-end of FiWi access networks allows the ONUs to be shared by multiple users, thereby significantly reducing the amount of required fiber infrastructure and increasing the coverage. This chapter deals with the energy efficiency of FiWi access networks assuming a conventional topology, where STAs connect directly to APs, each being collocated at an ONU, as widely found in many of today's real-world WLAN deployments.

2.2.2 ECO-FiWi General Description

Fig. 2.2 illustrates the operation of the ECO-FiWi scheme. The overall objective of ECO-FiWi is to design the whole FiWi network to operate in a TDM/TDMA fashion in order to exploit the idleness of network devices outside their transmission slots for implementing power-saving modes. TDMA is widely used in wireless networks due to its collision-free nature, thereby improving network performance [115]. Based on the fact that integrated ONU-APs are the central entities that connect the optical backhaul with the wireless front-end, ECO-FiWi is ONU sleep-centric. On the optical side, the ONU-AP exchanges data traffic with the OLT within its timeslot and then enters sleep mode when it is idle. At the same time, on the wireless side, it allocates bandwidth and schedules power-saving modes for its associated wireless devices accordingly.

In the EPON backhaul network, since only one ONU is scheduled to transmit US traffic using its timeslot at any time given, ECO-FiWi overlaps DS and US transmissions and schedules the ONU to sleep outside the timeslot for saving energy, similar to [124]. Note, however, that while only ONU sleep mode with fixed cycle time is considered in [124], ECO-FiWi schedules power-saving modes for not only ONU, but also AP, and STA in integrated EPON-WLAN networks with a dynamic polling cycle time that depends on the actual traffic loads. In addition, as only one ONU is active at any time given, instead of rerouting wireless traffic during an ONU sleep period as in [117, 118, 120] that might degrade the system performance due to overloading issues while incurring additional rerouting overhead, in ECO-FiWi, the AP module and wireless STAs are put into sleep mode. At the wireless front-end, the ONU-AP employs TDMA scheduling to assign an US subslot for each

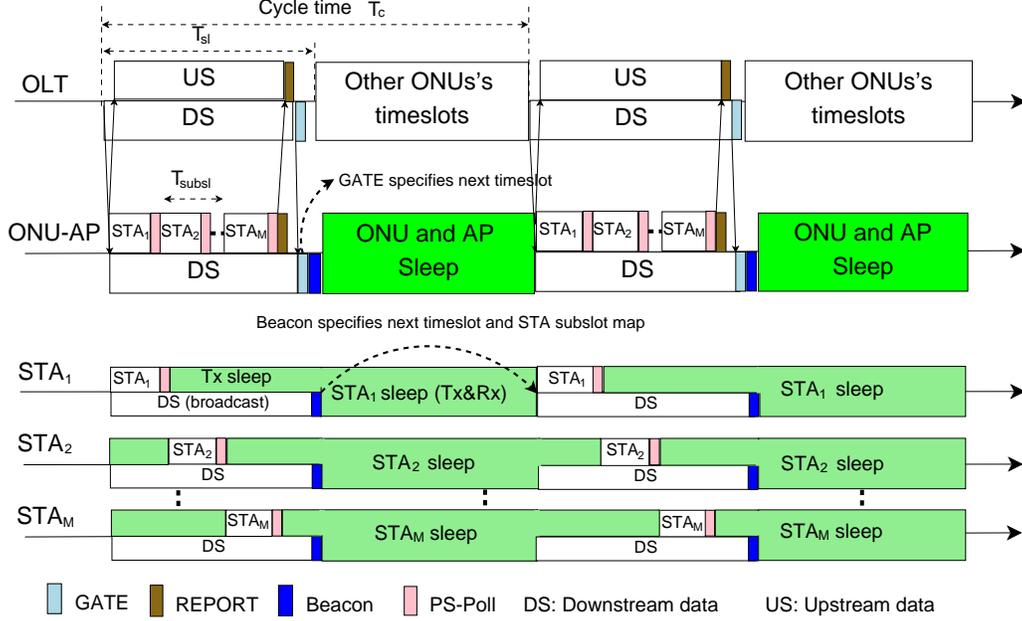


Figure 2.2: Illustration of ECO-FiWi for FiWi access networks.

STA in the next cycle based on its requested bandwidth and broadcasts DS frames to all STAs. In this work, STA transmitter and receiver are assumed to enter sleep mode independently. The transmitter stays active only during its subslot and enters sleep mode for the rest of the time in a cycle. Conversely, the receiver is left active for the whole ONU-AP timeslot to receive broadcast DS traffic and select its destined frames.

Similar to the EPON standard IEEE 802.3ah, the OLT allocates bandwidth to ONUs by means of multi-point control protocol (MPCP) messages. More specifically, the OLT assigns a timeslot and informs the ONU via a **GATE** message once it receives a **REPORT** message from an ONU-AP containing the US bandwidth request in terms of buffer backlog, which is expressed in time units [124]. However, the OLT does not compute the timeslot duration based only on US bandwidth request. Instead, it determines the duration by using a bandwidth algorithm, as described in Algorithm 1 shortly. Meanwhile, the ONU-AP allocates subslots to its associated STAs by means of the **Beacon** and **PS-Poll** frames, both of which are defined in WLAN standards. The **Beacon** frame is extended using its optional bits to include subslot parameters of STAs (e.g, start and duration). This way, ECO-FiWi does not entail any extra control messages in the backhaul, while allowing reduced signaling overhead compared to contention-based protocols in the wireless front-end. For network synchronization, the well-known timestamp mechanism specified in the EPON standard is employed,

Table 2.1: Notations and descriptions

Notation	Description	Value/Unit
N	Number of ONUs in system	16, 32, 64
M	Number of wireless STAs per ONU-AP	0-60
$olt_clk, onu_clk, sta_clk$	OLT, ONU-AP, STA clock	s
T_{prop}	Propagation delay from OLT to an ONU	0.01-1 ms
T_c	PON polling cycle time	s
T_g	Guard time between two consecutive slots	1 μ s
T_{pon}^{msg}	MPCP message time (GATE, REPORT)	0.512 μ s
T_{wl}^{msg}	WLAN message time (e.g., PS-Poll)	0.512 μ s
B_d, B_u	ONU DS/US buffer backlog at OLT/ONU	s
B_u^{sta}	STA US buffer backlog	s
onu_start, T_{sl}	Start and duration of an ONU timeslot	s
sta_start, T_{subsl}	Start and duration of a STA subslot	s
$T_{onu}^{oh}, T_{ap}^{oh}, T_{sta}^{oh}$	ONU, AP, STA sleep-to-active overhead time	2 ms, 2 ms, 250 μ s
P_{onu}^a, P_{onu}^s	ONU power consumption in active, sleep state	5052 mW, 750 mW
P_{ap}^a, P_{ap}^s	AP power consumption in active, sleep state	500 mW, 8 mW
P_{rx}^a, P_{tx}^a	STA Rx, Tx power consumption in active state	1500 mW, 2000 mW
P_{sta}^a, P_{sta}^s	STA total power consumption in active, sleep state	3500 mW, 20 mW

where all ONU-APs and STAs assign their local clocks to the OLT global clock once they receive a time stamped DS control message.

2.2.3 Joint Bandwidth Allocation and Power-Saving Scheduling Algorithms

This section presents algorithms that are executed at OLT, ONU-AP, and STA to realize a joint bandwidth allocation and power-saving scheduling mechanism in ECO-FiWi networks. The system operations are triggered based on the comparison between the local clock value of the device under consideration (i.e., OLT, ONU-AP, STA) and its time variables and the reception of relevant control messages. The OLT, ONU-AP, and STA maintain their own set of variables and parameters. Most of the quantities used in the description hereafter and their default values used in Section 2.5 are summarized in Table 2.1. Other quantities are defined throughout the text.

Algorithm 2.1 (see pseudo-code below) is implemented at an ECO-FiWi OLT for communication with an ECO-FiWi ONU-AP. When the OLT receives a `REPORT` from the ONU-AP (line 1) indicated in the binary variable `rep_rcvd`, it extracts the US bandwidth request and computes the start (onu_start) and duration (T_{sl}) of the timeslot. B_req is the bandwidth needed for both US and

Algorithm 2.1 Algorithm executed at an ECO-FiWi OLT

```
1: if ( $rep\_rcvd = TRUE$ ) then
2:    $B_u \leftarrow$  extracted US buffer backlog from the REPORT message
3:    $B_d \leftarrow$  Current DS buffer backlog
4:    $B\_req = \max\{B_d, B_u + RTT\} + 2T_{pon}^{msg}$ 
5:    $T_{sl} = \min\{B\_req, T_{sl}^{max}\}$ 
6:    $onu\_start = T\_sched + T_g$ 
7:    $T\_sched = T\_sched + T_{sl}$ 
8:   Generate a GATE message with  $RTT$ ,  $onu\_start$  and  $T_{sl}$ 
9:   Timestamp the GATE with  $olt\_clk$  and send it to the ONU-AP
10: else if ( $onu\_start \leq olt\_clk < onu\_start + T_{sl} - 2T_{pon}^{msg}$ ) then
11:   Transmit/receive DS/US data frames to/from the ONU
12: end if
```

DS transmissions. As shown in Fig. 2.2, in a timeslot, beside data bandwidth, for exchanging a GATE and a REPORT, the OLT needs only $2T_{pon}^{msg}$ time units, whereas the ONU needs $RTT + 2T_{pon}^{msg}$ time units ($RTT = 2T_{prop}$ is the round-trip time between the OLT and ONU-AP). Therefore, the OLT determines B_req as $\max\{B_d, B_u + RTT\} + 2T_{pon}^{msg}$ (line 4) [125]. However, the timeslot duration is upper bounded by a predefined maximum allowable duration T_{sl}^{max} in order to avoid channel overload situations (line 5). The start of the timeslot onu_start is the earliest time at which the channel is free, similar to [126] (line 6). After specifying T_{sl} , the OLT updates its time pointer T_sched , which indicates the latest point of time up to which the channel has been allocated (line 7). The OLT then generates and sends the GATE to the ONU-AP (lines 8-9). Beside the timeslot parameters, the GATE also contains RTT , which is necessary for the ONU-AP to compute the time it needs to generate a REPORT. While its clock is in the data interval of the current timeslot, the OLT transmits its buffered DS data frames and at the same time receives US data frames from the ONU-AP (lines 10-11). The reservation interval is triggered by the ONU-AP with a REPORT. Thus, after the data interval of a timeslot, the OLT expects to receive a new REPORT message and its operations are repeated. It should be noted that the differences between the ECO-FiWi OLT and a conventional OLT lie in the bandwidth allocation policy, i.e., lines 4-5, and the scheduling of DS transmission, which is overlapped with the US timeslot, i.e., line 11.

The algorithm executed at an ECO-FiWi ONU-AP is Algorithm 2.2. Once the ONU-AP receives a GATE from the OLT ($gate_rcvd = TRUE$ in line 1), it extracts olt_clk , RTT , onu_start , and T_{sl} (line 2). The ONU-AP first updates its clock to the olt_clk for synchronization (line 3). It then specifies the time it needs to send a REPORT in the next cycle rep_gen and its idle time until the timeslot T_{onu}^{idle} (lines 4-5). Based on T_{sl} and each STA uplink bandwidth re-

Algorithm 2.2 Algorithm executed at an ECO-FiWi ONU-AP

```
1: if ( $gate\_rcvd = TRUE$ ) then
2:   Extract  $olt\_clk$ ,  $RTT$ ,  $onu\_start$ , and  $T_{sl}$  from the GATE message
3:    $onu\_clk = olt\_clk$ 
4:    $rep\_gen = onu\_start + T_{sl} - RTT - 2T_{pon}^{msg}$ 
5:    $T_{onu}^{idle} = onu\_start - onu\_clk$ 
6:   Allocate subslot ( $sta\_start$ ,  $T_{subsl}$ ) for each STA based on  $T_{sl}$  and  $B_u^{sta}$ 
7:   Generate a Beacon frame with  $onu\_start$ ,  $T_{sl}$  and subslot map
    $\{(sta\_start_1, T_{subsl\_1}), \dots, (sta\_start_M, T_{subsl\_M})\}$ 
8:   Timestamp the Beacon with  $onu\_clk$  and broadcast it to all the STAs
9:   if ( $T_{onu}^{idle} > T_{onu}^{oh}$ ) then
10:     $T_{onu}^{wk} = onu\_clk + T_{onu}^{idle} - T_{onu}^{oh}$ 
11:     $T_{ap}^{wk} = onu\_clk + T_{onu}^{idle} - T_{ap}^{oh}$ 
12:    Switch off both ONU and AP transceiver
13:  end if
14: else if ( $onu\_clk = T_{onu}^{wk}$ ) then
15:   Perform ONU wake-up operation
16: else if ( $onu\_clk = T_{ap}^{wk}$ ) then
17:   Perform AP wake-up operation
18: else if ( $onu\_start \leq onu\_clk < rep\_gen$ ) then
19:   Receive DS data frames from the OLT and broadcast to all STAs
20:   if ( $PSPoll\_rcvd = TRUE$ ) then
21:     $B_u = B_u + B_u^{sta} + T_{wl}^{msg}$ 
22:   else
23:    Receive US data frames from each STA in subslot and send to OLT
24:   end if
25: else if ( $onu\_clk = rep\_gen$ ) then
26:   Generate and send a REPORT message with  $B_u$  to the OLT
27: end if
```

quest B_u^{sta} , the ONU-AP computes an uplink subslot T_{subsl} for each STA (line 6). The ONU-AP generates and broadcasts a Beacon frame containing onu_start , T_{sl} , an uplink subslot map $\{(sta_start_1, T_{subsl_1}), \dots, (sta_start_M, T_{subsl_M})\}$, and onu_clk (lines 7-8). The ONU-AP enters sleep mode, if T_{onu}^{idle} is longer than the wake-up overhead time (line 9). In case of sleeping, the ONU-AP computes the waking-up times for ONU module (T_{onu}^{wk}) and AP module (T_{ap}^{wk}) and switches the two modules off (lines 10-12). When onu_clk reaches the waking-up times, the ONU and AP modules wake up, respectively (lines 14-17). During the data interval (line 18), the ONU-AP receives DS data frames from the OLT and broadcasts them immediately to the STAs (line 19). Simultaneously, in the uplink direction, when a PS-Poll is received from an STA, the ONU-AP updates the US total bandwidth request, B_u , for the next cycle by adding the extracted STA bandwidth request, B_u^{sta} , and the time for a PS-Poll, T_{wl}^{msg} , (lines 20-21). Otherwise, it receives uplink data

Algorithm 2.3 Algorithm executed at an ECO-FiWi STA

```
1: if (bec_rcvd = TRUE) then
2:   Extract onu_clk, onu_start,  $T_{sl}$ , sta_start,  $T_{subsl}$ 
3:   sta_clk = onu_clk
4:    $T_{sta\_rx}^{idle}$  = onu_start - sta_clk
5:   if ( $T_{sta\_rx}^{idle}$  >  $T_{sta}^{oh}$ ) then
6:      $T_{sta\_rx}^{wk}$  = onu_start -  $T_{sta}^{oh}$ 
7:     Switch off STA receiver
8:   end if
9: else if (sta_clk =  $T_{sta\_rx}^{wk}$ ) then
10:  Perform wake-up operation for STA receiver
11: else if (onu_start ≤ sta_clk < onu_start +  $T_{sl}$ ) then
12:  Continuously receive broadcast DS data frames and select its own frames
13:  if (sta_clk = sta_start -  $T_{sta}^{oh}$ ) then
14:    Perform wake-up operation for STA transmitter
15:  else if (sta_start ≤ sta_clk < sta_start +  $T_{subsl}$  -  $T_{wl}^{msg}$ ) then
16:    Transmit US data frames to ONU
17:  else if (sta_clk = sta_start +  $T_{subsl}$  -  $T_{wl}^{msg}$ ) then
18:     $B_u^{sta}$  is assigned to US buffer backlog of STA
19:    Send PS-Poll message with  $B_u^{sta}$  to ONU for reporting US bandwidth request
20:    Switch off STA transmitter
21:  end if
22: end if
```

frames from its STAs in the TDMA subslots (lines 22-23). At the end of the data interval, i.e., when *onu_clk* reaches *rep_gen* (line 25), the ONU-AP generates and sends a REPORT with B_u to the OLT for the next cycle (line 26).

The algorithm executed at an ECO-FiWi STA is explained below in Algorithm 2.3. Once the STA receives a Beacon from the ONU-AP (*bec_rcvd* = *TRUE* in line 1), it extracts *onu_clk*, *onu_start*, T_s , *sta_start*, T_{subsl} (line 2). The STA first updates its clock for synchronization with the system (line 3). It then specifies the idle time of its receiver $T_{sta_rx}^{idle}$, which is the same as the ONU-AP idle time (line 4). The STA decides to turn its receiver off, if $T_{sta_rx}^{idle}$ is longer than the receiver waking-up time (line 5). In case of sleeping, the STA computes the time to wake up its receiver $T_{sta_rx}^{wk}$ and switches the receiver off (line 7). When *sta_clk* reaches $T_{sta_rx}^{wk}$, the receiver takes T_{sta}^{oh} to power up (lines 10-11). During the ONU-AP timeslot, the STA receives broadcast downlink traffic from the ONU-AP and selects its destined frames (lines 11-12). The STA transmitter sleeps and wakes up independently from the STA receiver. The transmitter quits the sleep period T_{sta}^{oh} time units before *sta_start*, i.e., just-in-time for its uplink transmission (lines 13-14). During the uplink subslot, the transmitter sends uplink frames to the ONU-AP (lines 15-

16). At the end of its subslot (line 17), the STA sends the ONU-AP a PS-Po11 frame containing B_u^{sta} as the next bandwidth request (line 19). Immediately after sending the PS-Po11 frame, the STA switches its transmitter off (line 20).

2.3 Energy Saving Model

This section presents an analysis of the potential energy saving gains arising from implementing the proposed ECO-FiWi scheme over an always-active FiWi system. Energy saving is defined as the relative energy consumption decrease with respect to the energy consumption without power-saving modes enabled. More specifically, energy saving is calculated for an ONU-AP (or conventional ONU) and its associated STAs within a EPON polling cycle T_c as follows:

$$\eta = \frac{E - E'}{E} = \frac{E - (E'_{back} + E'_{front1} + E'_{front2})}{E}, \quad (2.1)$$

where η is the overall energy saving and E is total energy consumption of an ONU-AP and its M STAs, when the system is always active. $E' = E'_{back} + E'_{front1} + E'_{front2}$ denotes the total energy consumption of the system, when ECO-FiWi is enabled. E'_{back} is the ONU-AP energy consumption in a cycle in the ECO-FiWi scenario. E'_{front1} and E'_{front2} denote the amounts of energy the M STAs consume during an ONU-AP timeslot and the rest of the time in a EPON cycle, respectively. E is given by:

$$E = T_c(P_{onu}^a + P_{ap}^a + MP_{sta}^a), \quad (2.2)$$

where description and considered values of P_{onu}^a , P_{ap}^a , and P_{sta}^a are provided in Table 2.1.

E'_{back} is computed as the total energy consumption of the ONU module and AP module in a polling cycle. In ECO-FiWi, the cycle time T_c is defined as the sum of all timeslot durations, i.e., $T_c = N(T_{sl} + T_g) = NT_{sl}$, assuming that guard time T_g is negligible compared to T_c . During the wake-up process, all network devices are assumed to consume the same amount of power as in active mode. In addition, T_c is assumed to be large enough such that the idle times of ONUs, APs, and STAs outside their slots are always greater than their wake-up overhead times, which are no greater than 2 ms (see Table 2.1). This ensures that the network components enter sleep mode every cycle. Since ONU and AP modules have the same idle time $T_c - T_{sl}$, by taking into account their wake-up

overhead times, E'_{back} is expressed as follows:

$$E'_{back} = P_{onu}^a(T_{sl} + T_{onu}^{oh}) + P_{ap}^a(T_{sl} + T_{ap}^{oh}) + P_{onu}^s(T_c - T_{sl} - T_{onu}^{oh}) + P_{ap}^s(T_c - T_{sl} - T_{ap}^{oh}). \quad (2.3)$$

E'_{front1} is the energy consumption of M STA receivers and M STA transmitters during T_{sl} . For simplicity, this chapter assumes that the wake-up overhead times of STA transmitter and receiver are the same (i.e., both are T_{sta}^{oh}). Recall that during an ONU-AP timeslot, STA receivers are always active, whereas their transmitters are in active mode in a round-robin manner. Note that apart from DS data frames, the receivers must be active to receive a **Beacon** frame at the end of an ONU-AP timeslot (see Fig. 2.2). The **Beacon** frame transmission time is assumed to be negligible with respect to the timeslot duration. Each of the M transmitters wakes up once immediately before its US subslot within a cycle. Thus, E'_{front1} is derived as follows:

$$E'_{front1} = MP_{rx}^a T_{sl} + P_{tx}^a T_{sl} + MP_{tx}^a T_{sta}^{oh} = MP_{sta}^a T_{sl} + MP_{tx}^a T_{sta}^{oh} - (M-1)P_{tx}^a T_{sl}, \quad (2.4)$$

where $P_{sta}^a = P_{rx}^a + P_{tx}^a$ denotes the power consumption of the whole STA transceiver in active mode operation.

E'_{front2} is computed as follows. Outside the timeslot, all STA receivers and transmitters are put into sleep mode and each of the M STA receivers wakes up once immediately before the timeslot for receiving DS broadcast frames (see Algorithm 2.3). Thus, it follows that:

$$E'_{front2} = MP_{sta}^s(T_c - T_{sl} - T_{sta}^{oh}) + MP_{rx}^a T_{sta}^{oh}. \quad (2.5)$$

From Eqs. (2.2–2.5), $E - E'$ in Eq. (2.1) is rewritten as follows:

$$\begin{aligned} E - E' &= P_{onu}^{dif}(T_c - T_{sl} - T_{onu}^{oh}) + P_{ap}^{dif}(T_c - T_{sl} - T_{ap}^{oh}) \\ &\quad + MP_{sta}^{dif}(T_c - T_{sl} - T_{sta}^{oh}) + (M-1)P_{tx}^a T_{sl}, \end{aligned} \quad (2.6)$$

where $P_{onu}^{dif} = P_{onu}^a - P_{onu}^s$, $P_{ap}^{dif} = P_{ap}^a - P_{ap}^s$, and $P_{sta}^{dif} = P_{sta}^a - P_{sta}^s$ denote the difference between power consumption of ONU, AP, and STA in active state and in sleep state, respectively.

By substituting $E - E'$ in Eq. (2.1) with Eq. (2.6) and T_{sl} with T_c/N , the energy saving of an ONU-AP and its wireless components η is expressed as follows:

$$\eta = \frac{(N-1)(P_{onu}^{dif} + P_{ap}^{dif} + MP_{sta}^{dif}) + (M-1)P_{tx}^a}{N(P_{onu}^a + P_{ap}^a + MP_{sta}^a)} - \frac{P_{onu}^{dif}T_{onu}^{oh} + P_{ap}^{dif}T_{ap}^{oh} + MP_{sta}^{dif}T_{sta}^{oh}}{T_c(P_{onu}^a + P_{ap}^a + MP_{sta}^a)}. \quad (2.7)$$

It is observed from Eq. (2.7) that the energy saving depends on many variables, including cycle time T_c , which is a function of the network traffic load, number of ONUs in the system, number of STAs per ONU-AP, power profile of the network devices, and their waking-up overhead times, as discussed in more detail in the next section. It is worth mentioning that a conventional ONU with only fixed subscribers (i.e., without the integration with AP) implements only the ONU sleep mode, similar to [124]. Therefore, its energy saving, η_{nonsta} , can be derived by eliminating the parameters related to wireless components, i.e., AP and STAs, in Eq. (2.7) as follows:

$$\eta_{nonsta} = \frac{(N-1)P_{onu}^{dif}}{NP_{onu}^a} - \frac{P_{onu}^{dif}T_{onu}^{oh}}{T_cP_{onu}^a}. \quad (2.8)$$

2.4 Analysis of End-to-End Delay

This section presents an analysis of end-to-end delay of ECO-FiWi networks for both US and DS traffic, i.e., the delay of a frame sent from an STA to the OLT and vice versa. It is worth noting that the delay performance has been studied and modeled for EPON systems with or without ONU sleep [127, 128] as well as for WLANs with STA power-saving mode [129–131]. Moreover, a delay analysis of PON-WLAN based FiWi networks without energy efficiency (i.e., no power-saving modes) has been presented in [114]. Note, however, that the delay performance of an ECO-FiWi network featuring power-saving modes of ONUs, APs, and STAs at the same time has not been carried out yet. The ECO-FiWi scheduling system is modeled as two polling systems (i.e., US and DS scheduling), whose mean frame delay can be analyzed by using an M/G/1 queuing model with modifications. Upper bounds on US and DS frame delays are then approximated based on a graphical argument. The maximum allowable timeslot duration T_{sl}^{max} is assumed to be $+\infty$ (see Algorithm 2.1). Hence, both US and DS transmissions are granted at least the bandwidth STAs/ONUs request for transmitting their traffic.

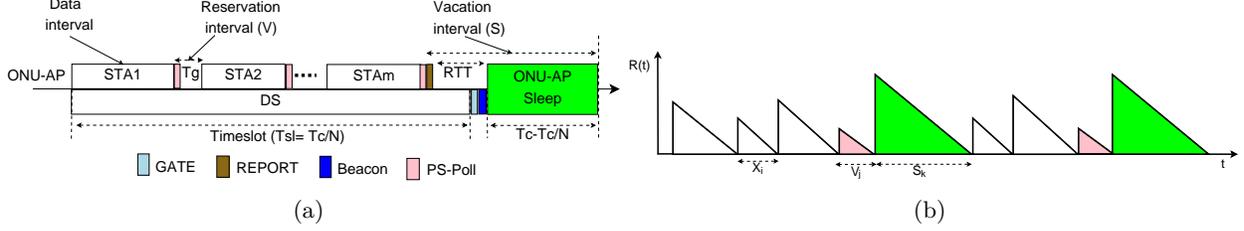


Figure 2.3: M/G/1 queueing model of US scheduling: (a) Illustration of US scheduling; (b) Illustration of residual delay component.

Consider an ECO-FiWi network with N ONU-APs, each associated with M STAs. All ONU-APs are identical in terms of frame arrivals and service times in both US and DS directions. This is also applied to the STAs in the system. All data buffers use first-in-first-out (FIFO) policy to select frames for transmission during their timeslots. The data buffers are assumed to have capacity high enough to avoid frame losses in the system. The OLT polls/serves the ONU-APs in a fixed order, say, 1, 2, ... N . Similarly, an ONU-AP serves its STAs in a fixed order from 1 to M . US and DS frames are assumed to arrive at data buffers according to a Poisson process with aggregate arrival rate λ_u and λ_d for US and DS transmissions, respectively. Thus, each ONU-AP and its STAs have an US arrival rate of λ_u/N . Frame service times are random with the first and second moments denoted by \bar{X}_u , \bar{X}_u^2 for US frames and \bar{X}_d , \bar{X}_d^2 for DS frames, respectively. \bar{V}_u , \bar{V}_u^2 , \bar{V}_d , and \bar{V}_d^2 represent the first and second moments of US and DS reservation times, respectively. Aggregate US and DS traffic loads are denoted by $\rho_u = \lambda_u \bar{X}_u$ and $\rho_d = \lambda_d \bar{X}_d$, respectively. The propagation delays from the ONU-AP to STAs are assumed to be negligible.

2.4.1 Mean Upstream Frame Delay

This section analyzes the mean US end-to-end frame delay for the US scheduling with gate service discipline. Mean US delay for the limited service discipline can be derived following a similar approach as described in [132]. It is important to note that US frame queuing delay is only due to the time-division access to the shared channel in the considered FiWi access network. That is, ONU-APs/STAs have to buffer incoming/generated US frames and wait until their allocated timeslots/subslots for transmitting the traffic, irrespective of the sleep mode operation outside their timeslots/subslots. In other words, ECO-FiWi does not incur any additional delay to US traffic. In addition, in ECO-FiWi, although DS and US transmissions are overlapped, the US scheduling is not affected by DS transmissions, as discussed in Section 2.4.2, Proposition 1.

The US scheduling of an ONU-AP for its associated wireless STAs is modeled as an M -user M/G/1 queue with reservations and vacations. The ONU-AP polls M STAs for their uplink transmissions during its timeslot assigned by the OLT. After that busy period, the ONU-AP goes on “vacation”, i.e., it does not serve STAs. The timeline of the server (ONU-AP) is decomposed into multiple subslots, each consisting of a data interval and a reservation interval, followed by a vacation interval, as illustrated in Fig. 2.3(a). In ECO-FiWi, the US reservation interval is the duration for a $PS - Poll$ frame T_{wl}^{msg} plus guard time T_g (see Table I), which avoids collision of different uplink transmissions. Thus, $V_u = T_{wl}^{msg} + T_g$ is a constant.

The US queuing delay consists of three delay components: the residual delay component R_u , the service delay component Q_u , and the reservation-and-vacation delay component Y_u [127, 132]. R_u is the remaining time until the on-going activity (frame transmission, reservation, or ONU-AP sleep) is completed. Q_u is the time for transmitting all US frames, which arrived at STA buffers ahead of the frame. Y_u is the total time of reservation intervals and ONU-AP sleep intervals seen by the frame of interest. Let \bar{K} denote the mean value of the random variable K . The mean US queuing delay for the gated service discipline \bar{W}_u is given by:

$$\bar{W}_u = \bar{R}_u + \bar{Q}_u + \bar{Y}_u. \quad (2.9)$$

\bar{Q}_u is computed by using Little’s Theorem as follows:

$$\bar{Q}_u = \frac{\rho_u}{N} \bar{W}_u. \quad (2.10)$$

Hence, \bar{W}_u is rewritten as:

$$\bar{W}_u = \frac{N}{N - \rho_u} (\bar{R}_u + \bar{Y}_u). \quad (2.11)$$

Lemma 1: The mean residual delay component of US queuing delay in ECO-FiWi networks is given by:

$$\bar{R}_u = \frac{\rho_u \bar{X}_u^2}{2N \bar{X}_u} + \frac{(1 - \rho_u)V_u}{2N} + \frac{(N - 1)^2(MV_u + RTT)}{2N(1 - \rho_u)}. \quad (2.12)$$

Proof: \bar{R}_u is derived based on the residual time to complete the on-going frame transmission, reservation interval, and ONU-AP sleep interval by using a graphical argument, as illustrated in Fig. 2.3(b), similar to the approach in [127]. \bar{R}_u is expressed as:

$$\bar{R}_u = \lim_{t \rightarrow \infty} \frac{1}{t} \int_0^t R(\tau) d\tau = R_1 + R_2 + R_3 = \lim_{t \rightarrow \infty} \frac{1}{t} \sum_{i=1}^{X_u(t)} \frac{X_i^2}{2} + \lim_{t \rightarrow \infty} \frac{1}{t} \sum_{i=1}^{V_u(t)} \frac{V_i^2}{2} + \lim_{t \rightarrow \infty} \frac{1}{t} \sum_{i=1}^{S(t)} \frac{S_i^2}{2}, \quad (2.13)$$

Table 2.2: Reservation-and-vacation time for US scheduling in ECO-FiWi networks

Frame arrival during	Probability	Waiting time (Y_u)
Case a: STA 1's data	ρ_u/MN	$MV_u + S$
Case b: STA 1's reservation	$(1 - \rho_u)/MN$	$2MV_u + 2S$
Case c: STA j 's data	ρ_u/MN	$(2M - j)V_u + 2S$
Case d: STA j 's reservation	$(1 - \rho_u)/MN$	$(2M - j)V_u + 2S$
Case e: ONU-AP sleep/idle	$(N - 1)/N$	$MV_u + S$

where $X_u(t)$, $V_u(t)$, and $S(t)$ are the number of US data frames, reservation intervals, and sleep intervals in the network zone (the ONU-AP and its STAs) up to time t , respectively. Whereas X_i , V_i , and S_i represent the duration of the i th US data frame, reservation interval, and sleep interval, respectively. As $t \rightarrow +\infty$, $X_u(t)/t$ is the time average of the departure rate, which equals the arrival rate λ_u/N . Thus, the first term on the right side of Eq. (2.13) R_1 becomes:

$$R_1 = \lim_{t \rightarrow \infty} \frac{X_u(t)}{t} \frac{\sum_{i=1}^{X_u(t)} X_i^2}{2X_u(t)} = \frac{\lambda_u \overline{X_u^2}}{2N} = \frac{\rho_u \overline{X_u^2}}{2N \overline{X_u}}. \quad (2.14)$$

Similarly, $V_u(t)/t$ is the time average of the reservation rate, which is equal to $(1 - \rho_u)/(N\overline{V_u})$.

The second term on the right side of Eq. (2.13) R_2 therefore becomes:

$$R_2 = \lim_{t \rightarrow \infty} \frac{V_u(t)}{t} \frac{\sum_{i=1}^{V(t)} V_i^2}{2V_u(t)} = \frac{1 - \rho_u}{2N\overline{V_u}} \overline{V_u^2} = \frac{(1 - \rho_u)V_u}{2N}. \quad (2.15)$$

$S(t)/t$ is the sleep rate, i.e., how many times the ONU-AP enters sleep mode in a time unit. In ECO-FiWi, since there is always one sleep interval of the ONU-AP in a polling cycle, as $t \rightarrow +\infty$, $S(t)/t$ is equal to $1/\overline{T_c}$. Hence, the last term R_3 in Eq. (2.13) is rewritten as:

$$R_3 = \lim_{t \rightarrow \infty} \frac{S(t)}{t} \frac{\sum_{i=1}^{S(t)} S_i^2}{2S(t)} = \frac{\overline{S^2}}{2\overline{T_c}}. \quad (2.16)$$

As depicted in Fig. 2.3(a), in ECO-FiWi, each ONU-AP goes on vacation for $N - 1$ timeslots in addition to $R_{TT} = 2T_{prop}$ in a cycle (see Table I). Thus, the portion of non-data time of the US transmission is $1 - \rho_u$, which corresponds to $N(MV_u + R_{TT})$ in a cycle, where $MV_u + R_{TT}$ represents the total reservation time of all M STAs and the time for a REPORT and a GATE to be exchanged in a cycle. Hence, T_c and S can be expressed as:

$$T_c = \frac{N(MV_u + R_{TT})}{1 - \rho_u} \quad \text{and} \quad (2.17)$$

$$S = \frac{(N-1)T_c}{N} = \frac{(N-1)(MV_u + RTT)}{1 - \rho_u}. \quad (2.18)$$

When the network reaches steady state, \bar{T}_c equals T_c and \bar{S} equals S . By assuming negligible variance of T_c compared to T_c^2 , \bar{S}^2 can be approximated to S^2 . Thus, Eq. (2.16) becomes:

$$R_3 = \frac{1}{2} \left(\frac{N-1}{N} \right)^2 \frac{T_c^2}{T_c} = \frac{(N-1)^2 (MV_u + RTT)}{2N(1 - \rho_u)}. \quad (2.19)$$

By substituting R_1, R_2 , and R_3 from Eqs. (2.14), (2.15), and (2.19) in Eq. (2.13), Lemma 1 is proved. ■

Lemma 2: The mean reservation-and-vacation delay component of US queuing delay in ECO-FiWi networks is given as follows:

$$\bar{Y}_u = \frac{[MV_u(N - \rho_u) + (N - 1)RTT](MN + M - \rho_u)}{MN(1 - \rho_u)} - \frac{(M^2 + M - 2)V_u}{2MN}. \quad (2.20)$$

Proof: Recall that \bar{Y}_u is the expected reservation and vacation time an STA US data frame experiences except the ongoing intervals. As illustrated in Fig. 2.4, \bar{Y}_u is determined based on the probability for the frame to arrive during a specific interval and the respective reservation-and-vacation time seen by the frame. Without loss of generality, the US frame of interest is assumed to belong to STA 1. Table 2.2 summarizes five possible cases of frame arrival, which are illustrated in Fig. 2.4 with their corresponding probability and reservation-and-vacation time. Note that as the ONU-AP enters sleep mode for S time units in a cycle, the probability for an US frame to arrive during sleeping/idle time is $(N-1)/N$. Therefore, by substituting S with the expression in Eq. (2.18), it follows that:

$$\begin{aligned} \bar{Y}_u = & \frac{\rho_u(MV_u + S)}{MN} s + \frac{(1 - \rho_u)(2MV_u + 2S)}{MN} + \frac{(N-1)(MV_u + S)}{N} \\ & + \sum_{j=2}^M \frac{\rho_u \{ (2M - j)V_u + 2S \}}{MN} + \sum_{j=2}^M \frac{(1 - \rho_u) \{ (2M - j)V_u + 2S \}}{MN}. \end{aligned} \quad (2.21)$$

After a few steps of derivation, Eq. (2.21) yields Eq. (2.20) from which Lemma 2 is proved. ■

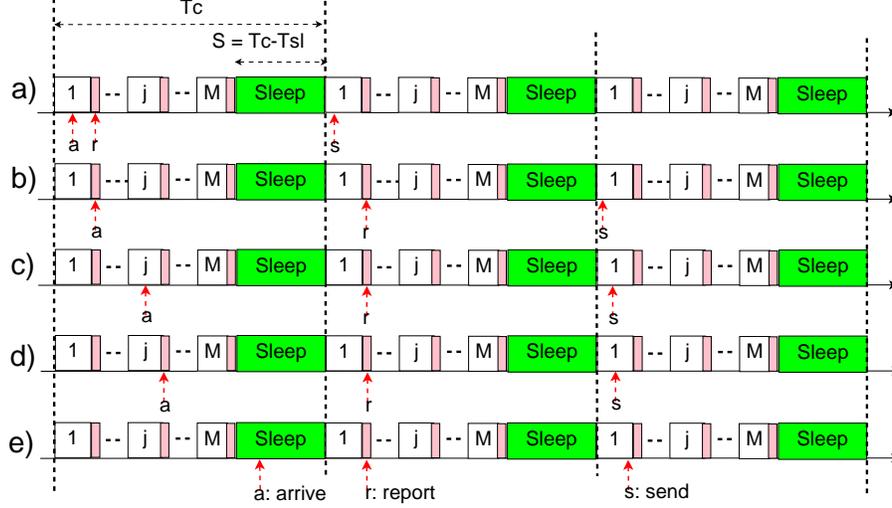


Figure 2.4: Frame arrival and its reservation-and-vacation time: (a) During ONU1's data interval; (b) During ONU1's reservation interval; (c) During ONU j 's data interval; (d) During ONU j 's reservation interval; and (e) During ONU-AP's sleep interval.

Theorem 1: Mean US end-to-end frame delay in ECO-FiWi networks with gated service discipline is given by:

$$\begin{aligned} \bar{D}_u = & \frac{\rho_u \bar{X}_u^2}{2(N - \rho_u) \bar{X}_u} + \frac{(1 - \rho_u)V_u}{2(N - \rho_u)} + \frac{(N - 1)^2(MV_u + RTT)}{2(N - \rho_u)(1 - \rho_u)} \\ & + \frac{[MV_u(N - \rho_u) + (N - 1)RTT](MN + M - \rho_u)}{M(N - \rho_u)(1 - \rho_u)} - \frac{(M^2 + M - 2)V_u}{2M(N - \rho_u)} + \bar{X}_u + \bar{T}_{prop}. \end{aligned} \quad (2.22)$$

Proof: In ECO-FiWi, the ONU-AP sends US frames received from STAs to the OLT immediately without incurring any additional queuing delay. Thus, the mean US end-to-end delay is obtained by adding the average propagation delay from the OLT to an ONU-AP, \bar{T}_{prop} (negligible propagation delay from the ONU-AP to STAs is assumed), and average frame time \bar{X}_u to the mean US queuing delay:

$$\bar{D}_u = \bar{W}_u + \bar{X}_u + \bar{T}_{prop}. \quad (2.23)$$

By substituting \bar{R}_u from Eq. (2.12) and \bar{Y}_u from Eq. (2.20) in Eq. (2.11), \bar{W}_{gated}^u is obtained. Then, replacing \bar{W}_{gated}^u in Eq. (2.23), Theorem 2.4.1 is proved. \blacksquare

2.4.2 Mean Downstream Frame Delay

Different from the EPON standard, where DS frames are broadcast by the OLT, in ECO-FiWi, the OLT buffers DS data frames for each ONU-AP in a dedicated buffer and transmits those in

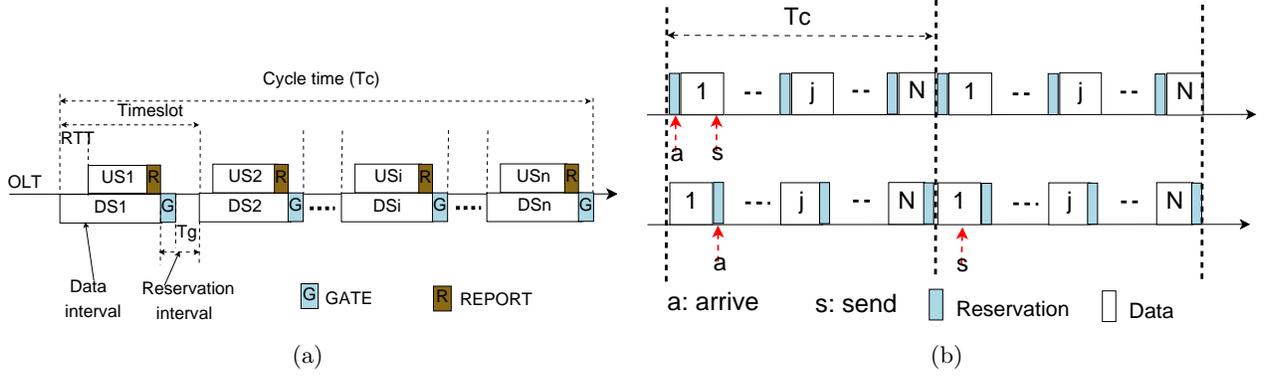


Figure 2.5: M/G/1 queuing model of DS scheduling: (a) Illustration of data and reservation intervals; (b) Illustration of reservation time difference between the conventional polling system (upper part) and DS scheduling (lower part).

synchronization with US transmissions. This results in some delay incurred to DS traffic. This section analyzes the mean DS end-to-end frame delay.

In the DS scheduling, the OLT polls N DS queues (customers) by means of GATE messages. This can be considered an N -user M/G/1 queue with reservation, which is similar to US scheduling in an EPON system implementing the IPACT algorithm [126]. More specifically, each DS timeslot is composed of a DS data interval and a reservation interval, which in turn consists of the transmission time of a GATE message and a guard time between two consecutive timeslots, as shown in Fig. 2.5(a). The guard time is the switchover time between two DS data queues at the OLT and is assumed to be the same as T_g in the US scheduling. Thus, the reservation interval V_d is a constant and equal to $T_{msg}^{pon} + T_g$.

Proposition 1: In ECO-FiWi networks, given a network reach between OLT and ONUs of at least 10 km, the DS transmission is considered a polling system with exhaustive service discipline.

Proof: Recall that, as specified in Algorithm 1, the OLT determines the timeslot duration for both US and DS transmissions in the next cycle as follows:

$$T_{sl} = \max \{B_d, B_u + RTT\} + 2T_{pon}^{msg}. \quad (2.24)$$

The purpose is to prove that in ECO-FiWi networks, T_{sl} is always set to $B_u + RTT + 2T_{pon}^{msg}$ in Eq. (2.24). According to IPACT [126], the adaptive cycle time based only on DS transmissions in

conventional EPON systems T_c^d can be expressed as follows:

$$T_c^d = \frac{NV_d}{1 - \rho_d}. \quad (2.25)$$

In Eq. (2.25), even for $N = 128$ ONUs, with $1 \mu\text{s}$ guard time and 64 Bytes of GATE message ($V_d = 1.512 \mu\text{s}$), a maximum value of T_c^d of 3.8 ms is achieved when ρ_d is equal to 0.95. This implies that on average each ONU would have a maximum timeslot of 0.03 ms. However, in the EPON-based ECO-FiWi system of at least 10 km network reach, the propagation delay is about 0.05 ms, i.e., $RTT = 0.1$ ms. This indicates that $B_u + RTT + 2T_{pon}^{msg} > B_d$ is always true and T_{sl} is always set to $B_u + RTT + 2T_{pon}^{msg}$ in Eq. (2.24), irrespective of DS and US traffic conditions. That is, the DS transmission always has much more data bandwidth than its requested bandwidth B_d in a cycle. In other words, all available DS frames of an ONU at the OLT are transmitted during a data interval, including those that arrived during the data interval, proving that the DS scheduling system operates in the exhaustive service discipline. ■

Note that in general, for a given US traffic load and network setup (i.e., RTT , M , and N), it is possible to specify a threshold of DS traffic load, above which the exhaustive service discipline does not apply to the DS scheduling. From Eqs. (2.17) and (2.25), by solving the equation $T_c^d = T_c$ for DS traffic load, the DS load threshold ρ_d^{th} is obtained as follows:

$$\rho_d^{th} = 1 - \frac{(1 - \rho_u)V_d}{MV_u + RTT}. \quad (2.26)$$

Lemma 3: The mean DS queuing delay has an additional mean reservation delay Δ_Y compared to the mean delay of a conventional polling system, where Δ_Y is given by:

$$\Delta_Y = \frac{(1 - \rho_d)(N - 1)\bar{V}_d}{N}. \quad (2.27)$$

Proof: Let R_d , Q_d , Y_d , and W_d denote the residual, service, and reservation delay components, and total queuing delay in the cyclic polling system, respectively (see Fig. 2.5(a)). According to [132], the delays for an exhaustive system with constant reservation interval are given by:

$$\bar{Q}_d = \rho_d \bar{W}_d, \quad \bar{R}_d = \frac{\lambda_d \bar{X}_d^2}{2} + \frac{(1 - \rho_d)\bar{V}_d}{2}, \quad \text{and} \quad \bar{Y}_d = \frac{(N - 1)\bar{V}_d}{2}. \quad (2.28)$$

Following a similar approach as in [128], the mean DS queuing delay in ECO-FiWi networks \bar{W}_d has an additional mean reservation delay Δ_Y , compared to the mean delay of a conventional polling system. This is because the order of data and reservation intervals of a timeslot in a TDM-PON

system is opposite to that in the conventional polling system [132]. Without loss of generality, the DS frame of interest is assumed to be for ONU1. There are four possible cases for the DS frame arrival (see Fig. 2.5(b)). It may arrive during ONU1's data interval, ONU1's reservation interval, ONU j 's data interval, and ONU j 's reservation interval, where j is the index of another ONU. Their corresponding probabilities are ρ_d/N , $(1 - \rho_d)/N$, ρ_d/N , and $(1 - \rho_d)/N$, respectively. Among those cases, only when the DS frame arrives during ONU1's reservation interval, the reservation components of the two systems are different, as illustrated in Fig. 2.5(b). More precisely, the frame is transmitted in the data interval followed by the reservation interval in the polling system and thus experiences no reservation interval. However, in the case of ECO-FiWi DS scheduling, the frame needs to wait until ONU1's next timeslot and thus experiences $N - 1$ reservation intervals before it is transmitted. Thus, Δ_Y is expressed as:

$$\Delta_Y = \frac{(1 - \rho_d)(N - 1)\bar{V}_d}{N}. \quad (2.29) \quad \blacksquare$$

Theorem 2: The mean DS end-to-end frame delay for ECO-FiWi networks, \bar{D}_d , is given by:

$$\bar{D}_d = \frac{\rho_d \bar{X}_d^2}{2(1 - \rho_d)\bar{X}_d} + \frac{(N - 1)(N + 2 - 2\rho_d)\bar{V}_d}{2N(1 - \rho_d)} + \bar{X}_d + \bar{T}_{prop}. \quad (2.30)$$

Proof: From Eqs. (2.28–2.29), the mean DS queuing delay \bar{W}^d is given by:

$$\bar{W}_d = \bar{R}_d + \bar{Q}_d + \bar{Y}_d + \Delta_Y = \frac{\rho_d \bar{X}_d^2}{2(1 - \rho_d)\bar{X}_d} + \frac{(N - 1)(N + 2 - 2\rho_d)\bar{V}_d}{2N(1 - \rho_d)}. \quad (2.31)$$

As the ONU-AP broadcasts DS data frames received from the OLT to its STAs without incurring any delay to the DS data frame, by adding the average frame time \bar{X}_d and average propagation delay \bar{T}_{prop} , the mean end-to-end DS frame delay \bar{D}_d is obtained as follows:

$$\bar{D}_d = \bar{W}_d + \bar{X}_d + \bar{T}_{prop} = \frac{\rho_d \bar{X}_d^2}{2(1 - \rho_d)\bar{X}_d} + \frac{(N - 1)(N + 2 - 2\rho_d)\bar{V}_d}{2N(1 - \rho_d)} + \bar{X}_d + \bar{T}_{prop}. \quad (2.32) \quad \blacksquare$$

2.4.3 Estimate of Upper Bound on End-to-end Frame Delay

Apart from the analysis of the mean frame delay, estimates of upper bound on US and DS delay can be approximated based on the graphical argument depicted in Fig. 2.6 in order to provide FiWi

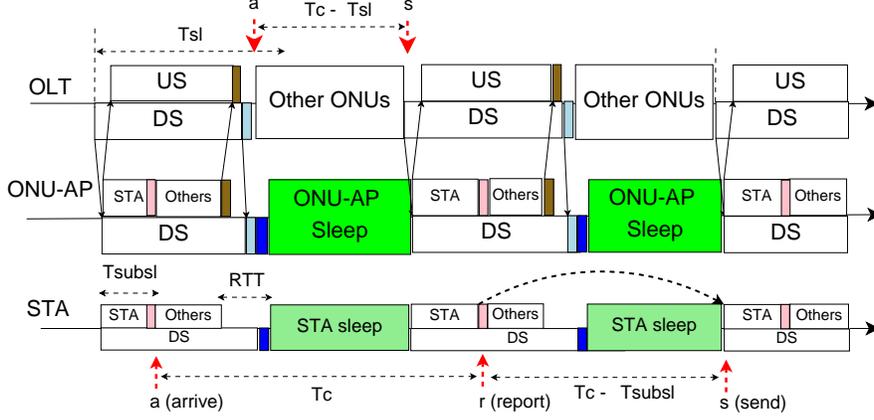


Figure 2.6: Illustration of upper bounds on US and DS frame delays in ECO-FiWi networks.

network operators with an estimate of the maximum traffic delay. Let X_{max} denote the maximum frame transmission time of both DS and US transmissions in an ECO-FiWi network.

Lemma 4: In ECO-FiWi networks with gated service being employed in the US scheduling, the US end-to-end delay is upper bounded by $2T_c - T_{wl}^{msg} + X_{max} + T_{prop}$.

Proof: With the gated service employed in the US scheduling, an US frame experiences the maximum delay, if it arrives at the STA immediately after a PS-Poll message. The frame therefore has to wait until the next PS-Poll to be accounted in the computation of the subslot in the next cycle. This waiting time is equal to T_c , as shown in Fig. 2.6. After being reported, the frame must wait for the previously scheduled frames (of other STAs under the same coverage of the ONU-AP) to be transmitted before its subslot. The frame will be the first to be transmitted in the subslot, and thus the latter waiting time is equal to $T_c - T_{subsl}$. Hence, the total queuing time of the frame is $2T_c - T_{subsl}$, where T_{subsl} is at least the reservation interval, i.e., $T_{subsl} \geq T_{wl}^{msg}$. The frame experiences a delay at the ONU-AP as its frame transmission time $X_u \leq X_{max}$. By adding T_{prop} , the maximum end-to-end delay of an US frame D_u^{max} is given by:

$$D_u^{max} = 2T_c - T_{subsl} + T_{prop} + X_u \leq 2T_c - T_{wl}^{msg} + T_{prop} + X_{max}. \quad (2.33)$$

Lemma 5: In ECO-FiWi networks with gated service being employed in the US scheduling, the DS end-to-end delay is upper bounded by $T_c - T_{prop} - 2T_{pon}^{msg} + X_{max}$.

Proof: Similar to the proof of Lemma 4, the maximum DS queuing delay occurs when a frame arrives at the OLT immediately after a GATE message. Because of the exhaustive service in the DS

transmission, the frame has to wait in the queue for a duration of $T_c - T_{sl}$, as illustrated in Fig. 2.6. Given that $T_{sl} \geq RTT + 2T_{pon}^{msg}$ (see Algorithm 2.1), D_d^{max} is given by:

$$D_d^{max} = T_c - T_{sl} + X_d + T_{prop} \leq T_c - T_{prop} - 2T_{pon}^{msg} + X_{max}. \quad (2.34)$$

2.5 Results

This section presents the performance evaluation of the proposed ECO-FiWi scheme and discusses the findings. An ECO-FiWi network with both DS and US transmission capacity of 1 Gb/s is considered in the evaluation.

Constants and Parameters

The constants and parameter settings used in the evaluation are listed in Table 2.1. The number of ONUs N , number of STAs per ONU-AP M , and RTT are varied to investigate their impact on the performance of ECO-FiWi. T_g is also varied to adjust the cycle time T_c , as will be explained in detail later on. The traffic load of US and DS transmissions, ρ_u and ρ_d , are varied from 0.05 to 0.95. The mean frame transmission times \bar{X}_u and \bar{X}_d are both set to $5.09 \mu s$, whereas their second moment \bar{X}_u^2 and \bar{X}_d^2 are both set to $21.44 (\mu s)^2$, as in [128]. The maximum frame transmission time \bar{X}_{max} is set to $12.14 \mu s$, which corresponds to the maximum Ethernet frame size of 1518 Bytes. MPCP GATE and REPORT messages are of size 64 Bytes [126], corresponding to a transmission time T_{pon}^{msg} of $0.512 \mu s$. T_{wl}^{msg} is assumed to be $0.512 \mu s$. By assuming the same value of T_g , V_u and V_d are the same and constant, i.e., $V_u = V_d = 1.512 \mu s$.

Polling Cycle Time and Energy Saving

Fig. 2.7 shows the dependence of potential energy saving (in percents) on the polling cycle time given in Eqs. (2.7–2.8). For any given configuration, the integrated ONU-AP saves significantly more energy compared to a conventional ONU (without serving wireless subscribers) due to the savings from STAs and APs (i.e., $\eta > \eta_{nonsta}$ is always true). The longer the cycle time, the higher the energy saving obtained from utilizing ECO-FiWi. This is because a longer cycle time allows ONU, AP, and their STAs to stay longer in sleep mode with the same overhead time to wake up

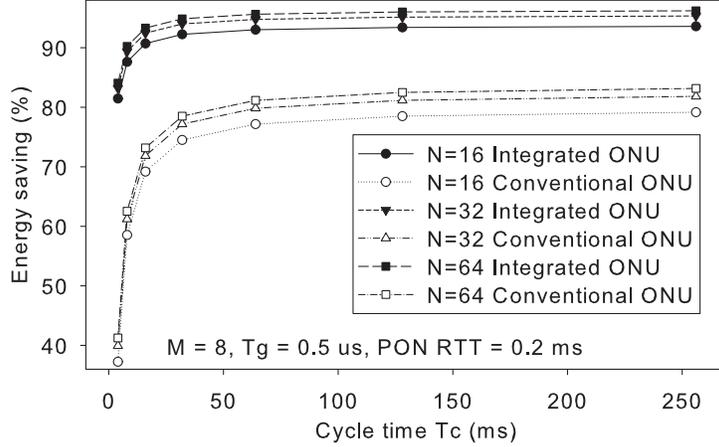


Figure 2.7: Potential energy saving with varying cycle time T_c : Integrated vs. Conventional ONU (η vs. η_{mosta}).

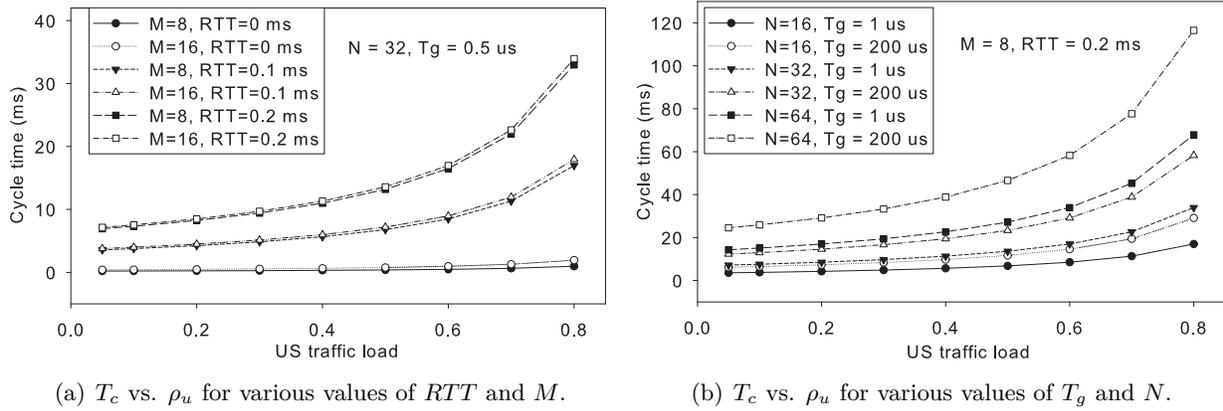


Figure 2.8: Polling cycle time T_c as function of ρ_u , RTT , and T_g .

(just once in a cycle). However, as detailed in Algorithms 2.2 and 2.3, when T_c is not larger than the sleep-to-active overhead times of the network devices, no energy-efficiency would be gained. This happens in the first part of the η_{mosta} curve in Fig. 2.7. On the contrary, with 8 STAs per ONU-AP, η is high even for very small values of T_c because sleep mode is still possible for STAs, provided that they need only 250 μ s to wake up (see Table 2.1).

T_c itself depends on ρ_u , RTT , N , M , and V_u , as given in Eq. (2.17). Fig. 2.8 shows the dependence of T_c on the parameters for various configurations. For any given configuration, when ρ_u increases T_c increases as well. In addition, a larger number of ONUs, or a larger number of STAs per ONU-AP, or a longer network reach (i.e., a larger RTT) considerably increases T_c . It is observed from Fig. 2.8 that RTT and N have a more noticeable effect on T_c than M , which slightly changes the value of T_c . As shown in Fig. 2.8(a), when RTT is small, T_c could be very low (less

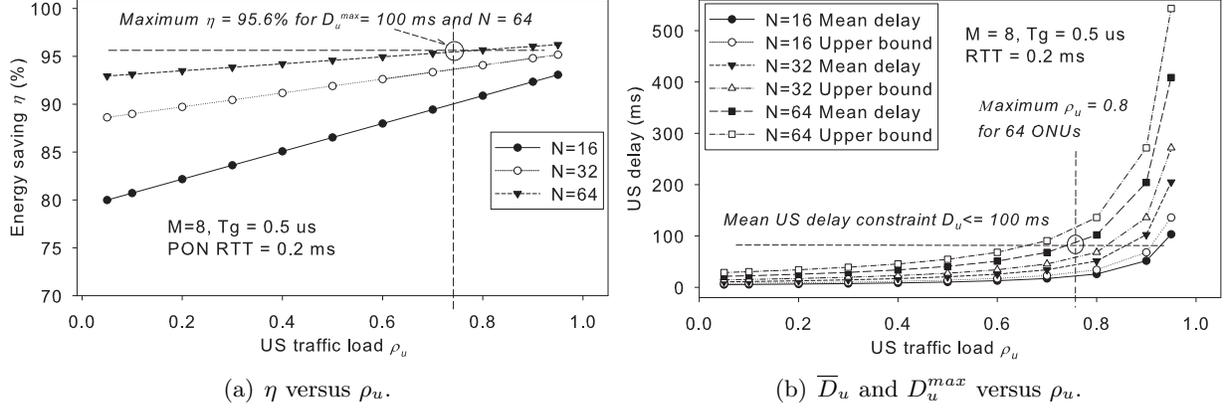


Figure 2.9: Energy saving η and US frame delays (\bar{D}_u and D_u^{max}) as function of US traffic load ρ_u .

than 1 ms for $M = 8$), even for a system of 32 ONUs operating at a traffic load of 0.9. In fact, it is not necessary to have such a low value of T_c , for example, when the user constrained frame delay is relaxed. T_c can be enlarged by configuring a long reservation interval V_u , i.e., extending T_g , as proposed in [133]. As shown in Fig. 2.8(b), increasing T_g from 0.512 μ s to 200 μ s almost doubles T_c . This could be exploited for energy efficiency improvements through power-saving modes as long as frame delays are acceptable.

Energy Saving and Frame Delay

There is a clear trade-off between energy saving and frame delay. A large value of T_c helps achieve higher energy saving, though it incurs a longer frame delay. In this section, the impact of traffic load on the energy saving and delay performance is thoroughly investigated.

Fig. 2.9(a) reveals that η increases linearly with ρ_u , which means that the energy saving achieved from deploying ECO-FiWi grows linearly with the actual traffic load. It is observed that for any US data load, more than 80% of energy can be saved by using ECO-FiWi. In addition, for a given traffic load, the higher the number of ONUs is, the higher the energy saving becomes. This stems from the fact that more ONUs imply that each ONU has a smaller timeslot, thereby a longer idle time outside its timeslot. Furthermore, when N increases, the slope of the linear function is smaller. This can be understood by linking Eq. (2.7) with Eq. (2.17). It is also observed that when ρ_u is large, η tends to converge. This is because when ρ_u approaches 1, T_c goes to infinity and the second term in Eq. (2.7) becomes negligible.

Fig. 2.9(b) shows that both mean and upper bound US frame delays increase for increasing ρ_u . The mean US delay is roughly three fourth of the US upper bound delay. Moreover, the US frame delay remains rather low when the traffic load is low, but rapidly increases when the traffic load increases. For example, for $\rho_u = 0.6$, a US frame experiences a mean delay of less than 50 ms even for a system of 64 ONUs, which is acceptable for many real-time applications. But for $\rho_u = 0.8$, mean US delay is about 100 ms for the same system configuration. In addition, the larger the number of ONUs in the system is, the higher the US delays become. It is worth recalling that the US frame delay comes from the TDMA nature of ECO-FiWi networks rather than the implemented power-saving modes. Irrespective of whether STAs and ONU-APs enter sleep mode or not, a US frame would experience the same delay for a given network configuration.

Consolidating the energy saving in Fig. 2.9(a) with US delay in Fig. 2.9(b) allows operators to find an optimal energy-delay trade-off. For example, when the constrained mean US frame delay is 100 ms, Fig. 2.9(b) helps indicate the maximum possible US traffic load to guarantee given QoS requirements. Specifically, with 8 STAs per ONU-AP and 10 km network reach, a system of 64 ONU-APs must have ρ_u no higher than 0.8, whereas ρ_u can be as high as 0.9 for a system of 32 ONU-APs. Using these values of ρ_u to project on the energy saving curve in Fig. 2.9(a), operators could infer that 95.6% of energy can be saved for a system of 64 ONU-APs without violating given delay constraints. A similar procedure can be applied to find the optimal energy-delay performance for the case of upper bound delay constraint as well as for other evaluation scenarios described in the following (Figs. 2.10 and 2.11).

Impact of Network Reach and Number of STAs

The impact of RTT on energy saving and US delay is shown in Fig. 2.10. As shown in Fig. 2.10(a), for a given value of ρ_u , when RTT increases, η also increases as a result of a larger cycle time, as explained in the previous section (see also Eqs. (2.7) and (2.17)). For small values of RTT , as T_c is not large enough for ONUs and APs (both have a wake-up overhead time of 2 ms) to enter sleep mode, η is rather low. On the one hand, for large values of RTT such as in long-reach FiWi systems [118], η tends to be saturated. On the other hand, Fig. 2.10(b) shows that the US delay is very sensitive to varying RTT . In particular, both US upper bound and mean frame delay increase linearly with RTT and the US delay becomes critical in long-reach FiWi networks. For example,

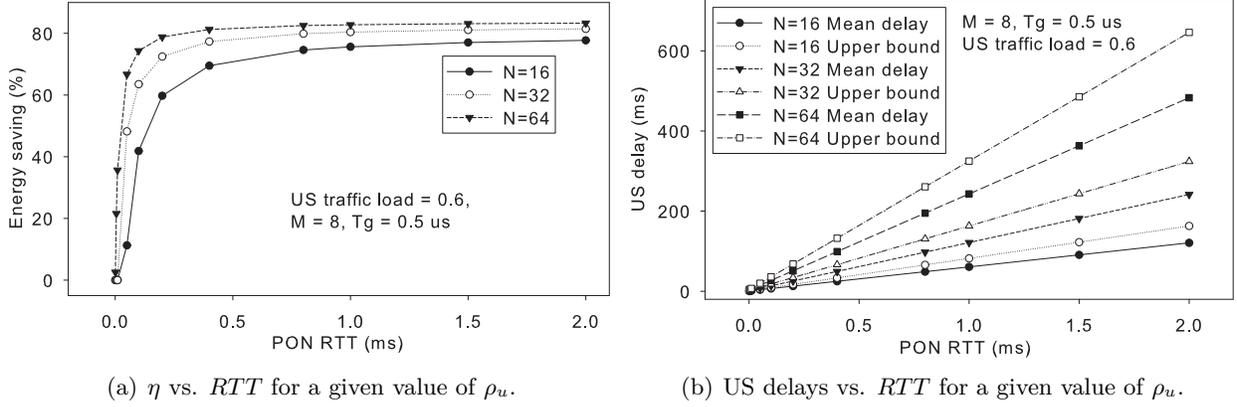


Figure 2.10: Energy saving (η) and US frame delays (\bar{D}_u and D_u^{max}) as function of PON RTT .

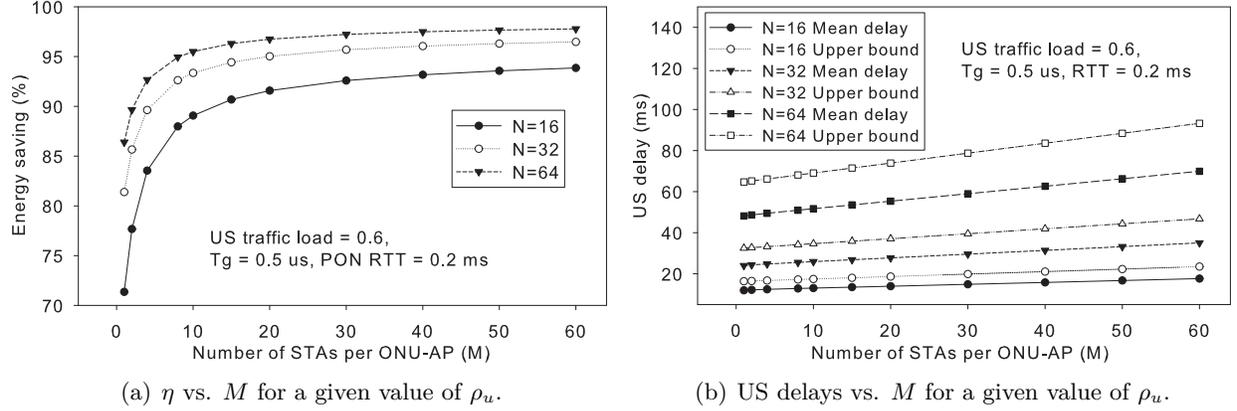


Figure 2.11: Energy saving (η) and US frame delays (\bar{D}_u and D_u^{max}) as function of number of STAs per ONU-AP (M).

for $RTT = 1$ ms, corresponding to a network reach of 100 km from OLT to ONU-AP, each US frame experiences an end-to-end delay of about 242 ms for a system of only 64 ONUs. However, the long frame delay for large RTT is an issue stemming from the long propagation delay in long-reach PONs rather than from the utilization of power-saving modes. The dependence of US delay on RTT can be further explained by referring to Eqs. (2.22) and (2.33).

Fig. 2.11 shows the impact of the number of STAs per ONU-AP M on the energy saving and US delay performance for a given value of ρ_u . Fig. 2.11(a) indicates that STAs significantly contribute to the overall energy saving η as they increase considerably for increasing M . This is because an increasing M results in an increase of T_c , as explained previously (see Eq. (2.17)). Note that even when there is only one STA associated with an ONU-AP, more than 70% of energy could be saved by using the proposed ECO-FiWi scheme. Interestingly, an increasing M leads to only a slight and

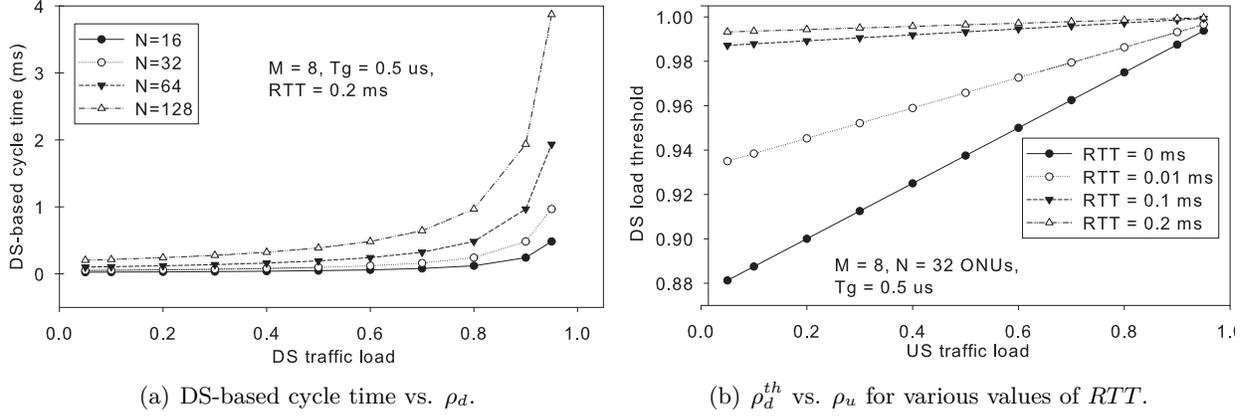


Figure 2.12: DS-based cycle time T_c^d and exhaustive service threshold for DS transmission ρ_d^{th} .

linear increase of the US delay (both \bar{D}_u and D_u^{max}), as shown in Fig. 2.11(b). This can be explained by referring to Eq. (2.22). Overall, Fig. 2.11 reveals that by designing an ECO-FiWi network with many STAs be associated with an ONU-AP, the energy efficiency is considerably improved, while the US frame delay is only slightly increased.

Impact on Downstream Performance

As mentioned earlier, the downside of ECO-FiWi may be the delay incurred to DS traffic. However, the DS delay reported in this section helps confirm that the impact on DS transmissions is marginal. Fig. 2.12(a) shows the cycle time based only on DS traffic, i.e., T_c^d in Eq. (2.25). It is shown that even under high DS traffic loads, the cycle time is rather small for any value of N . This further supports the argument that in ECO-FiWi networks the OLT always determines the timeslot duration as $B_u + RTT + T_{pon}^{msg}$ and T_c is always greater than T_c^d , irrespective of DS traffic, as stated in Section 2.4.2. Fig. 2.12(b) shows the DS load threshold ρ_d^{th} versus US traffic load given by Eq. (2.26). From Fig. 2.12(b), it is observed that even with negligible RTT and no US traffic, ρ_d^{th} is very high, for instance, 0.88 in the case of 16 ONUs. For $RTT=0.2$ ms, ρ_d^{th} is always greater than 0.9, implying that the DS transmission operates in exhaustive service, thus verifying the correctness of the mean DS delay analysis presented in Section 2.4.2.

The mean and upper bound end-to-end DS delays are shown in Fig. 2.13. While \bar{D}_d depends on the actual DS traffic ρ_d , as expressed in Eq. (2.32), D_d^{max} is a function of T_c , as expressed in Eq. (2.34). As proved in Lemma 5, T_c is a function of ρ_u , implying that D_d^{max} is dependent on ρ_u

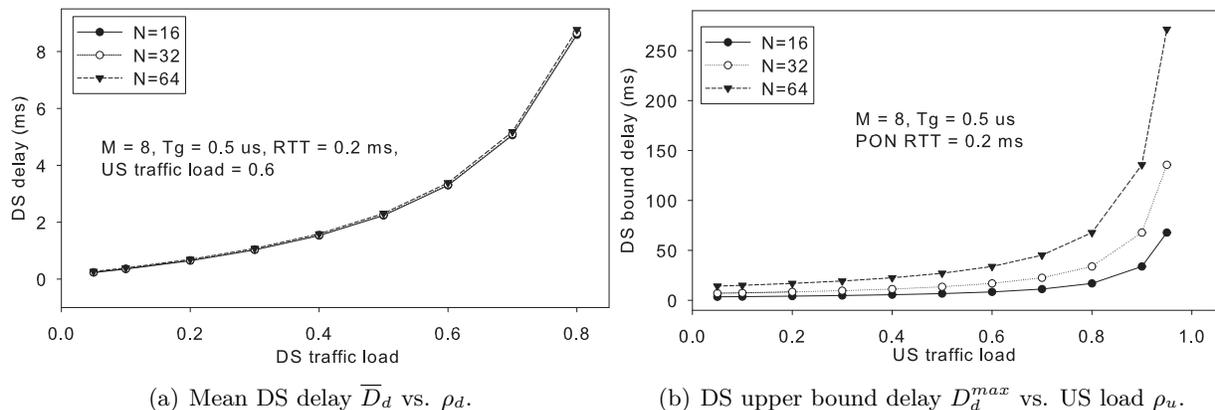


Figure 2.13: DS end-to-end frame delays (\bar{D}_d and D_d^{max}).

rather than ρ_d . Comparing Fig. 2.13(a) with Fig. 2.9(b), it is observed that the mean DS delay is much lower than the mean US delay for a given traffic load (i.e., $\rho_u = \rho_d$). This is because of the exhaustive service discipline in DS transmissions. For example, at a DS load of 0.8, a DS frame experiences a mean delay of less than 9 ms compared to 51 ms mean US delay (in the case of 32 ONUs). This means that ECO-FiWi incurs a very low mean frame delay to DS traffic. Similarly, the DS upper bound delay is approximately half of the US upper bound delay for a given value of ρ_u . In addition, as shown by Fig. 2.13(a), the number of ONUs in the system has a negligible effect on the mean DS delay. Fig. 2.13(b) shows that at a given US traffic load, the upper DS delay bound is much lower than that of US delay. This is also due to the exhaustive service in the DS transmissions, as discussed in Section 2.4.3.

In short, ECO-FiWi saves significant amounts of energy for FiWi access networks, especially when the number of STAs per ONU-AP is large. Energy saving can be further improved by extending the cycle time with an extended guard time between timeslots as long as frame delay allows. Interestingly, the achieved energy saving linearly grows with increasing US traffic load. While ECO-FiWi does not incur any additional delay to US traffic with respect to TDMA scheduling, the high energy efficiency is achieved at the expenses of a very low and manageable incurred DS frame delay.

2.6 Conclusions

This chapter proposed an ECO-FiWi scheme to save energy for EPON-WLAN based FiWi access networks with power-saving modes. Unlike existing solutions, bandwidth allocation and power-

saving mode scheduling of ONUs, APs, and STAs are jointly performed and incorporated into the TDMA-based operation of the EPON DBA process in order to improve overall energy efficiency and network performance. Wireless STAs are scheduled to exchange traffic with their associated ONU-AP in a TDMA manner and put into sleep mode outside their transmission slots for maximizing their energy efficiency and thus battery life. The achievable energy saving was analytically modeled accounting for both wireless and optical network components. A comprehensive analysis of both DS and US end-to-end frame delays was presented based on M/G/1 queueing models covering both optical backhaul and wireless front-end network segments of FiWi access networks. The obtained results confirmed that it is beneficial to apply TDM/TDMA to FiWi access networks for improved energy efficiency and delay performance. By exploiting the idleness of network components outside their transmission slots for power-saving modes, the proposed ECO-FiWi scheme achieves significant amounts of energy saving (more than 70% in typical scenarios), while preserving upstream delay and incurring manageable delay to downstream traffic. A unique feature of ECO-FiWi is that its achievable energy savings linearly increase with respect to traffic load. The proposed ECO-FiWi scheme represents a promising solution for future sustainable FiWi access networks.

Chapter 3

Workflow Scheduling in Multi-Tenant Cloud Computing Environments

3.1 Introduction

Cloud computing is one of the most promising contemporary technologies, on which the research community has recently embarked [13, 134]. It has been emerging as a powerful way to transform the IT and Telcos industries in order to build and deploy custom services and applications, e.g., healthcare, and scientific computations. “Cloud computing is a large-scale distributed computing paradigm that is driven by economies of scale, in which a pool of abstracted, virtualized, dynamically-scalable, managed computing power, storage, platforms, and services are delivered on demand to external customers over the Internet [135].”

Multi-tenancy is one of the key features of cloud computing. In a conventional single-tenant cloud architecture, providers offer a dedicated cloud service (instance of application and underlying infrastructure) to the tenants (customers), where no data is intermingled with other tenants. From the service provider’s point of view, this model does not provide scalable cloud services and economics of scale. On the other hand, in multi-tenant cloud computing, infrastructures, applications, and database are shared among all tenants. At the downside, tenants may not be able to customize

This chapter is based on the following publication:
[64]B. P. Rimal and M. Maier. Workflow Scheduling in Multi-Tenant Cloud Computing Environments. *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 1, pp. 290-304, Jan. 2017.

their use of cloud services in order to fit their specific needs. Moreover, multi-tenancy may be seen differently from a cloud service model perspective. For instance, as for Infrastructure-as-a-Service (IaaS), tenants have the ability to provision computing, storing, and networking resources. Thus, an IaaS provider must allow tenants for virtualization and resource sharing to achieve multi-tenancy. The virtual machine (VM) level multi-tenancy provides the following benefits: a) increased utilization of hardware resources and ease of maintenance (e.g., avoiding dedicated installation per tenant); b) scalability/elasticity, i.e., dynamic workload of large numbers of tenants is handled by scaling up/down resources; c) economics of scale, e.g., reduced service delivery cost for end-users and service providers by sharing the same instance of infrastructure based on an abstraction of isolation between tenants' data and VMs. In Software-as-a-Service (SaaS), a single instance of hosted applications is used by multiple tenants simultaneously, e.g., Force.com [24].

Even though multi-tenancy allows cloud service providers to better utilize computing resources, supporting the development of more flexible services based on economy of scale, and reducing infrastructural costs, how to effectively realize this is a fundamental question. For instance, multi-tenant cloud computing poses unique challenges such as scalability, resource provisioning (e.g., meeting the demand of large volumes of tenants per resource) and customization (per-tenant service customization) [22, 23]. In addition, multi-tenant applications need to be dynamic in nature, or polymorphic, to fulfill the individual expectations of various tenants and their users [24]. This chapter investigates a resource management framework that consists of both architecture and scheduling algorithm with regards to multi-tenancy issues, especially scalability and shared resources for scheduling compute-intensive workflow applications. Further, a resource management framework is developed whose objective is to separate resource management policies from the control mechanisms required to implement them. Note that there are several detrimental consequences for both service providers and tenants such as unpredictable application performance, limited cloud applicability, and inefficiencies in datacenters and revenue [25]. However, these issues are beyond the scope of this chapter.

Workflow scheduling is a process of mapping and managing the execution of inter-dependent tasks on distributed resources. It allocates suitable resources to workflow tasks such that the execution can be completed to satisfy objective functions imposed by users. The proper and efficient scheduling can have significant impact on the performance of the workflow system. In general, scheduling tasks for distributed services was shown to be an NP-hard problem [136]. Thus, there is no optimal solution within polynomial time. Heuristic algorithms were widely developed in

order to achieve near optimal solutions. However, there is not any particular resource management framework for scheduling workflow in multi-tenant cloud computing environments.

This chapter explores the idea of workflow scheduling in the context of multi-tenant cloud computing environments. In a multi-tenant cloud computing environment, designing a separate service layer for workflows on the top of IaaS or SaaS, giving rise to *Workflow-as-a-Service (WaaS)*. That provides fast turn-around times of the submitted tasks by means of automaticity of the centralized submission interface and scheduler. Given the vital importance of multi-tenancy (e.g., isolation and customization aspects in shared infrastructures), this chapter proposes a resource management framework that consists of a novel four-layered architecture of workflow scheduling system and scheduling algorithm. The introduced architecture and cloud-based workflow scheduling algorithm (CWSA) provide multi-tenancy by unifying abstractions – workflows, resource, and control mechanism (scheduling) – that enable logically centralized policies. Further, the architecture enables a resource abstraction that unifies arbitrary resources, such as storage, network, CPU, and pools, enabling resource-agnostic policies. Moreover, the envisioned architecture provides a service and management environment to enable multiple tenants to run their compute-intensive workflow applications on a shared cloud infrastructure while taking advantage of the elasticity and pay-as-you-go billing model of cloud computing.

To deal with the resource management in the envisioned architecture, this chapter proposes a novel CWSA algorithm for compute-intensive workflow applications. CWSA enforces the decisions of resource scheduling centrally without requiring explicit coordination. Indeed, CWSA takes advantage of the gaps between scheduled tasks. These idle periods can be used to schedule other tasks, thereby reducing the overall makespan. Further, CWSA helps minimize the overall workflow completion time, cost of execution of workflows, tardiness, and utilize idle resources of cloud effectively. Importantly, CWSA utilizes computational resources properly by reducing idle time of cloud resource nodes. In addition, CWSA exhibits less context switching and thus outperforms the state-of-the-art scheduling schemes such as First Come First Served (FCFS), EASY (Extensible Argonne Scheduling system) Backfilling, and Minimum Completion Time (MCT). Furthermore, to evaluate the scalability performance of the proposed algorithm, a proof-of-concept experiment with real-world scientific workflow applications such as biology application (e.g., SIPHT) and earthquake science application (e.g., CyberShake) is performed, which verifies the effectiveness of the proposed solution.

The main contributions of this chapter are as follows. First, from *an architectural perspective*, we propose a novel four-layered architecture of workflow scheduling system in a multi-tenant cloud computing environment, which represents a cost-effective solution for executing compute-intensive workflow applications. Second, from *a resource management perspective*, we propose a novel CWSA algorithm to deal with both structured and unstructured workflow scheduling in multi-tenant cloud computing environments, which maximizes cloud resource usage, minimizes the expected makespan (the overall completion time of the workflow), and also minimizes the scheduling execution time, workflow execution costs, and total expected tardiness (delay penalty, if a task in workflow is completed after its due-time). Third, we perform comprehensive *simulation-based studies* of the proposed scheduling algorithm and evaluate different set of performance metrics, including statistical analysis, to demonstrate the robustness of the proposed scheduling algorithm. We then compare the performance of the CWSA with the FCFS, EASY Backfilling, and MCT scheduling schemes. Note that there is no consensus on widely accepted metrics to measure robustness. Thus, in this work, we propose to use the makespan standard deviation, skewness of the makespan, and the 99th percentile distribution of the makespan to compare robustness metrics. Fourth, from *a proof-of-concept demonstration perspective*, CWSA is further implemented in real world scientific complex workflow applications (e.g., SIPHT and CyberShake – popular benchmark that have been widely used in workflow studies) to demonstrate the scalability and effectiveness of the proposed solution.

The remainder of the chapter is structured as follows. Section 3.2 describes cloud workflow in a nutshell. In Section 3.3, we briefly review related work. Section 3.4 presents the proposed reference architecture of workflow scheduling and describes the proposed algorithm in greater detail. Section 3.5 describes the salient features of our implementation, followed by experimental results obtained by means of simulations, and provides a comparison of its performance with that of different alternative scheduling schemes. Further, a proof-of-concept demonstration is presented and finally, we conclude and discuss future work in Section 3.6.

3.2 Workflow

Cloud workflow applications (e.g., high-end scalable media processing, scientific workflow applications, data analytics) can be defined as collections of resource intensive activities processed in a well defined order to achieve a certain goal, which could be executed in geographically distributed

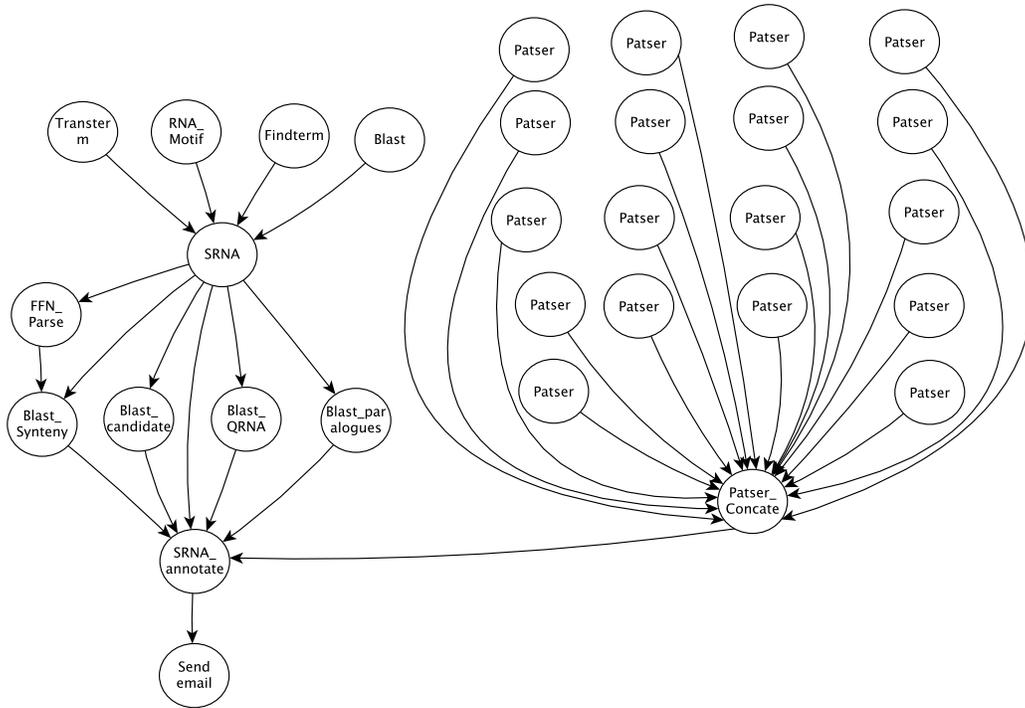


Figure 3.1: SIPHT [140] – an example of scientific workflow.

heterogeneous resources. Based on the structural complexity, cloud workflow applications can be sub-divided into two groups: balance structured, e.g., Electron Micrograph ANalysis (EMAN) – a Bio-imaging workflow and unbalanced structured, e.g., SIPHT and Montage workflows. The balanced structured workflow contains several parallel pipelines that require the same types of service to process different data sets. Conversely, unbalanced structured workflows are complex and require different services [137].

Scientific workflow applications (see Fig. 3.1) have a complex structure and require heterogeneous services. The execution of such workflow applications faces several challenges such as scalability, quality of service (QoS), ensured reproducibility, computing resources, data storage, as well as heterogeneous and distributed data management [138, 139]. Such workflow applications require a distributed high-end computing environment, leading to the recently emerged cloud computing.

Cloud computing offers several distinct features for workflow applications compared to other computing environments such as: a) dynamic resources allocated/de-allocated, which enables the workflow to elastically scale up/down the cloud resources; b) a large shared pool of resources with scalable processing capability, storage, and network resources to execute compute-intensive workflow

applications; c) cloud workflow applications can share application instances and their underlying cloud resources between multiple tenants, which allows a cloud service provider to maximize cloud resource utilization and thereby reduce service costs per tenant; d) manages secure isolation of such resources between multiple tenants through its multi-tenancy feature.

3.3 Related Work

The problem of scheduling tasks on multiple resources has been extensively studied in parallel and distributed systems, cluster and grid computing, and in recent years to a lesser extent in cloud computing. The methodology adopted varies according to the characteristics of the workload (e.g., batch workload or online workload, large/medium/small size, and frequency), characteristics of resources (e.g., physical/virtual resources, number of nodes, and networks), performance metrics of interest, and scheduling based on multi-agent systems, e.g., [141]. Further, most of these algorithms considered a stable infrastructure. We briefly review prior work on various aspects in the following.

Heuristic algorithm: Several studies considered heuristics, e.g., list scheduling, clustering, and task duplication. Examples of list scheduling include Heterogeneous Earliest-Finish-Time (HEFT) [142] and Fast Critical Path (FCP) [143] scheduling for a single workflow. A list scheduling heuristic combined with multi-objective optimization was proposed in [144] for scheduling workflow in grids and clouds. Well-known examples of task duplication based algorithms include [145] and task duplication-based scheduling algorithm for network of heterogeneous systems (TANH) [146]. Clustering heuristics, e.g., [147] for task clustering and CASS-II for task clustering with no duplication [148] were studied in heterogeneous systems.

A cost- and time-based heuristic algorithm was proposed in [149] to minimize the execution, communication costs, and overall completion time for scheduling workflow tasks in cloud environments. The Whittle's index-based heuristic scheduling was proposed in [150] for executing parallel tasks on opportunistically available cloud resources. The opportunistic scheduling allocates low-priority tasks to intermittently available servers to minimize the cost of waiting and migration. A dynamic resource allocation heuristic was proposed in [151] to minimize skewness and improve the utilization of server resources. Notably, in [152], authors have shown that widely-used Best-Fit scheduling algorithm is not throughput-optimal. The work in [153] addressed the static resource-

constrained multi-project scheduling problem (RCMPSP) by considering project and portfolio lateness. However, RCMPSP may not be suitable in the context of multi-tenant cloud environments due to multi-tenant interference, unfairness, as well as variable and unpredictable performance (e.g., throughput). Some tenants may pay for performance isolation and predictability, while other tenants may choose best-effort behavior [154].

Meta-heuristic algorithm: A particle swarm optimization (PSO) based scheduling algorithm was proposed in [155, 156] to minimize the execution cost of workflow applications in cloud computing. The market-oriented cloud workflow systems based on genetic algorithm, ant colony optimization, and PSO has proposed in [157]. In [158], authors proposed a pricing model and dynamic scheduling of single tasks in commercial multi-cloud environments and compared their approach with pareto-optimal solutions based on two classical multi-objective evolutionary algorithms, i.e., SPEA2 and NSGA-II.

Note that heuristic based scheduling algorithms fit only a particular type of problem (e.g., a workflow with a simple structure), while the meta-heuristic algorithm provides a general solution method for developing a specific heuristic to fit a particular kind of problem [137]. Most of the heuristic and meta-heuristic based scheduling algorithms were designed and optimized in the context of grid computing environments. Further, meta-heuristic algorithms such as PSO, SPEA2, and NSGA-II are very time consuming and thus not very effective for large workflow applications.

Scientific workflows execution: The performance and the cost of execution of scientific workflows in a cloud environment was studied in [159]. This study has shown that most of the resources provided by Amazon EC2 are less powerful for I/O-intensive applications like Montage than Abe (NASA HPC cluster) due to the lack of high-performance parallel file systems. A data locality driven task scheduling algorithm was proposed to improve the system performance on cloud computing environments [160]. However, the algorithm did not implement real-world cloud applications to verify its effectiveness (e.g., scalability, dynamic workload). Similarly, a matrix based k-means clustering strategy for data placement in scientific cloud workflows was presented in [161].

There exists a plethora of workflow management systems, e.g., Pegasus [162]. However, many of their features are optimized for conventional grid and cluster computing to execute single/multiple job(s) or workflow and thus may not be able to obtain most of the key aspects of cloud computing, while such systems suffer from limited resource provisioning. Although there are few works

addressing workflow scheduling on clouds, e.g., [161], they were not designed in the context of multi-tenancy.

Deadline-aware scheduling: Dynamic resource provisioning of adaptive applications in cloud environments was studied in [163] based on Q-learning reinforcement guided control theory to maximize the application-specific benefit function within given time and budget constraints for a particular task. VGrADs [164], a virtual grid execution system– was studied for scheduling of a deadline sensitive weather forecast workflow supports. A resubmission heuristic based on the HEFT algorithm [142] was proposed in [165] to meet soft deadlines of scientific workflows in computational grids. A deadline constraint algorithm based on the HEFT was proposed in [166] for scheduling a single workflow instance on an IaaS cloud environment. However, none of these algorithms were designed in the context of multi-tenant cloud environments.

Multi-tenant SaaS applications: There exist only few studies on multi-tenant SaaS applications, e.g., a cluster-based resource allocation algorithm [167], which includes lazy and pro-active duplications to achieve an improved system performance in a two-tier multi-tenant SaaS schedule architecture. Further, the capacity planning of multi-tenant applications using a method for determining the optimal allocation of application threads to physical nodes was studied in [168]. Recently, a resource allocation model for SaaS applications was proposed in [169]. However, none of these approaches dealt with the compute-intensive workflow scheduling in multi-tenant cloud.

Overall, most of the aforementioned studies focused on the scheduling performance of a single workflow. Moreover, these approaches did not consider compute-intensive workflow applications in a multi-tenant cloud computing environment. In fact, there exist no particular scheduling schemes to execute compute-intensive workflow applications in multi-tenant cloud environments. Unlike those approaches, in this chapter we focus on the minimization of the makespan, cost of execution of the workflows, and tardiness, while maximizing the resource utilization within a given deadline in multi-tenant cloud computing environments.

3.4 Multi-Tenant Cloud Environments for Workflow Scheduling

This section presents the proposed reference architecture of workflow scheduling in a multi-tenant cloud environments and describes the proposed CWSA algorithm in greater detail.

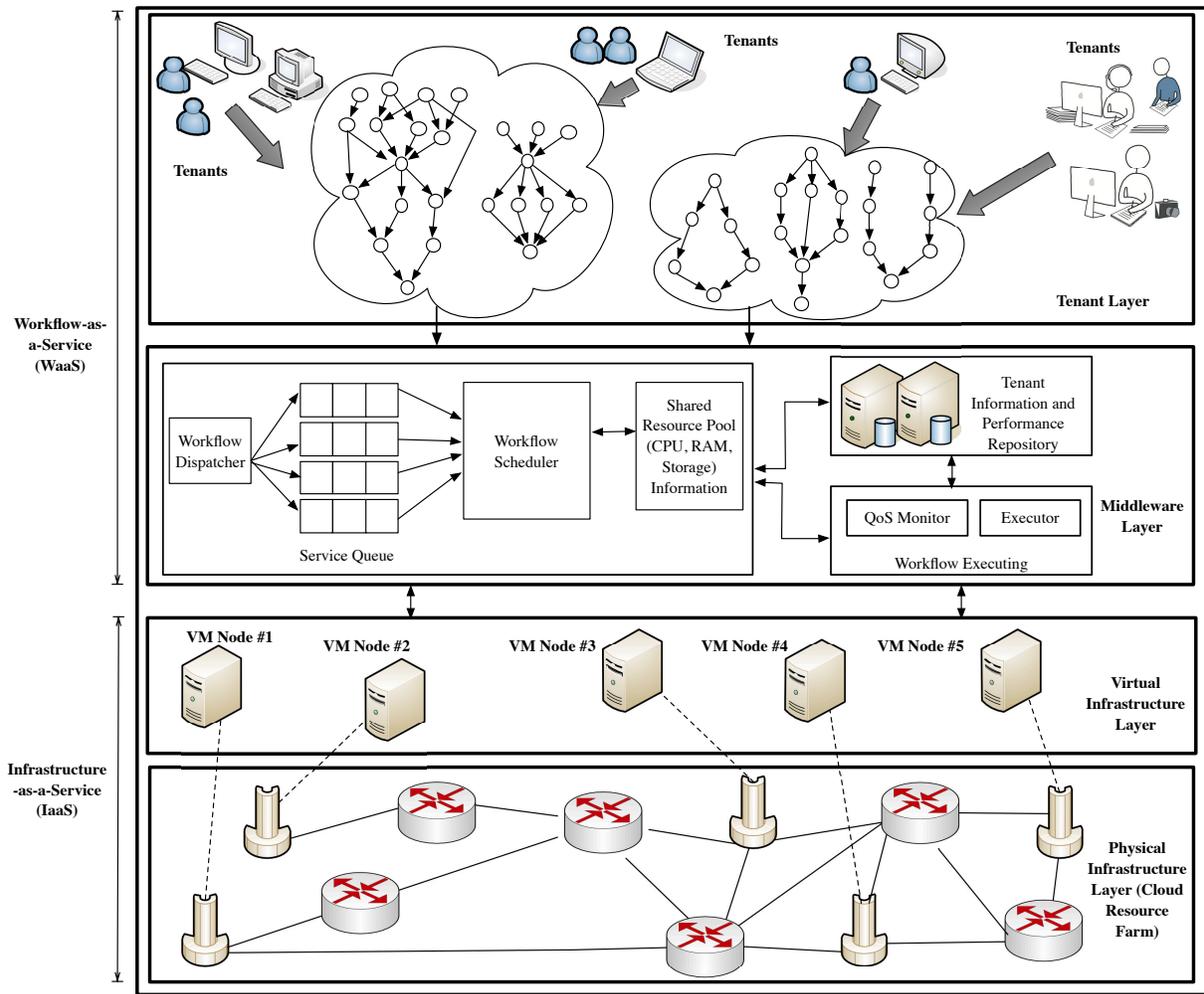


Figure 3.2: Proposed reference architecture of workflow scheduling in a multi-tenant cloud environment.

3.4.1 Architecture

Given the emergence of diverse sets of scientific workflow applications each belonging to different domains, a multi-tenant aware and flexible workflow platform is needed to cost-effectively execute/deploy the workflow applications of multiple tenants. The envisioned architecture enables such workflow applications to share a single infrastructure while taking advantage of the elasticity and pay-as-you-go billing model of cloud computing.

Fig. 3.2 depicts a four-layered architecture of the proposed workflow scheduling system. The first layer (tenant) consists of workflow creator/composer. The second layer (middleware) consists of workflow dispatcher, service queue, workflow scheduler, shared pool resource information, tenant information and performance repository, QoS monitor, and executor. The third layer is the virtual

infrastructure layer, whereas the fourth layer consists of the physical infrastructure layer. Each layer is briefly described in the following.

Tenant layer: Each tenant-specific cloud workflow is configured by acquiring the tenant preferences and QoS. Each tenant submits an individual workflow application to the workflow scheduling system. The workflow tasks are scheduled according to the available resources (virtual machines, datacenter, etc.) at the given deadline. The tenants submit the workflow applications according to a uniform or random distribution. According to the users' QoS requirements, the workflow scheduler checks the availability of services and resources and then applies the given scheduling schemes to execute these workflow tasks.

Middleware layer: In order to realize multi-tenancy, a suitable middleware is required to minimize the underlying complexity (e.g., configure, manage, and identify multiple tenants as well as tenant-specific customization of workflow applications). In fact, it decouples the tenant and infrastructure layers. The middleware layer comprises a number of components such as workflow scheduler, QoS monitoring component, and performance repository component. When the tenant workload increases, a pool of identical application instances is created in order to ensure scalability. These components are briefly described in the following.

Workflow dispatcher: The main functionality of the workflow dispatcher is to aggregate the workload (workflow applications) and dispatch it to the service queue.

Service queue: The service queue maintains a priority queue for all incoming workflow tasks and distributes them to the workflow scheduler.

Workflow scheduler: The workflow scheduler is the core component of the middleware layer, which provides several features for storing task information, maintaining up-to-date cloud resource information, resource selection information (matching task requirements to resource space available on the cloud), QoS monitoring information, and performance information. The major functions of this component help select prioritized workflow tasks from the service queue, execute scheduling schemes, send provisioning instructions to the virtual infrastructure layer for creating virtual resources and subsequently mapping the workflow tasks to virtual machines (VMs). Further, the workflow scheduler communicates with the performance repository and tenant information com-

ponent in order to get information about the current status of the cloud resources and tenants, including which VMs are running on which physical machines in the cloud resource farm.

Shared resource pool: A resource pool is a logical abstraction that is needed for the flexible management of resources. It provides information about used, limit, available, and shared resources (e.g., CPU, RAM, storage, network, and software licenses).

Tenant information and performance repository: This component stores tenant configuration files. More specifically, it saves configuration and customization metadata of all tenants. It also accumulates all the data related to QoS, e.g., deadline, availability, scalability, etc. Therefore, any changes of the configuration would affect the scope of a particular tenant. Further, the services that are to be composed should be selected based on such a configuration (i.e., tenant information).

QoS monitoring: This component monitors the QoS data. Further, it oversees the performance of the service/workflow application instance. It reports on whether the service instance on its worker node is underloaded, overloaded, or in normal conditions based on the threshold. Such information is retrieved from the tenant information and performance repository component.

Executor: The executor notifies a task's completion status after finishing it successfully. During the workflow execution, the scheduler is responsible for handling task dependencies such as transferring dependent files. In addition, the middleware layer coordinates the virtual infrastructure layer by distributing workflow tasks to available resources.

Virtual infrastructure layer: This layer consists of a number of VM instances running on top of the cloud server farm. It allocates on-demand resources and maps the virtual resources to the physical resources.

Physical infrastructure layer: It contains a number of physical resources, i.e., cloud server farm that consists of physical servers for computing, storage, and network. It also provides provisioning and deprovisioning of resources to the virtual infrastructure layer.

3.4.2 Problem Formulation

Application and Resource Models

A cloud workflow can be formally modeled as a directed acyclic graph (DAG). It consists of computational tasks and transmission tasks. A DAG is a tuple $G = (V, E)$, where $V = \{\tau_i \mid i = 1, \dots, v\}$ denotes the set of vertices representing tasks with $|V| = v$ and $E = \{e_{ij} \mid (i, j) \in \{1, \dots, v\} \times \{1, \dots, v\}, |E| = e$ is the set of communication edges representing precedence relation between two computational tasks. The labels on nodes denote computation costs and the labels on edges represent communication costs. The communication time between tasks in the workflow is determined by various factors, including bandwidth, number of tasks, and volume of data transferred. A critical path (CP) is the longest path in the workflow.

For illustration, an example workflow is shown in Fig. 3.3, where edge $e(i, j)$ from $(\tau_i - \tau_j)$ means that τ_j needs an intermediate result from τ_i such that $\tau_j \in \text{succ}(\tau_i)$, where $\text{succ}(\tau_i)$ is the set of all intermediate successor tasks of τ_i . A task without immediate predecessors is known as an entry-task and a task without immediate successors as an exit-task. The average communication cost between two tasks τ_i and τ_j is equal to $C(i, j) = \frac{\text{data}(i, j)}{B}$, where B denotes the average bandwidth and $\text{data}(i, j)$ represents the amount of data required to be transmitted between the τ_i and τ_j . If τ_i and τ_j are assigned to the same resource (i.e., VM), the communication cost is negligible and assumed to be zero. The computation cost is the execution costs of tasks on the resources. It is computed by dividing the total number of instructions required to execute that tasks by the processing capacity of resources (in instructions per second). The service queue is considered large enough to buffer all unprocessed tasks. The IaaS cloud may offer a set of VM instance types with different processing power (CPU, RAM, storage) and prices. When a VM is leased, it needs a boot-up time to be properly initialized and be made available to the tenant (s). Thus, this time is considered in the scheduling and cost evaluation. Since we assume a single cloud datacenter model to execute workflow applications, where physical machines are interconnected by high-bandwidth links. Thus, we do not consider the cost incurred by data transfer. Further, the so-called *pay-per-use* billing model is considered, where partial utilization of the leased VM instance-hour is counted as a full time period.

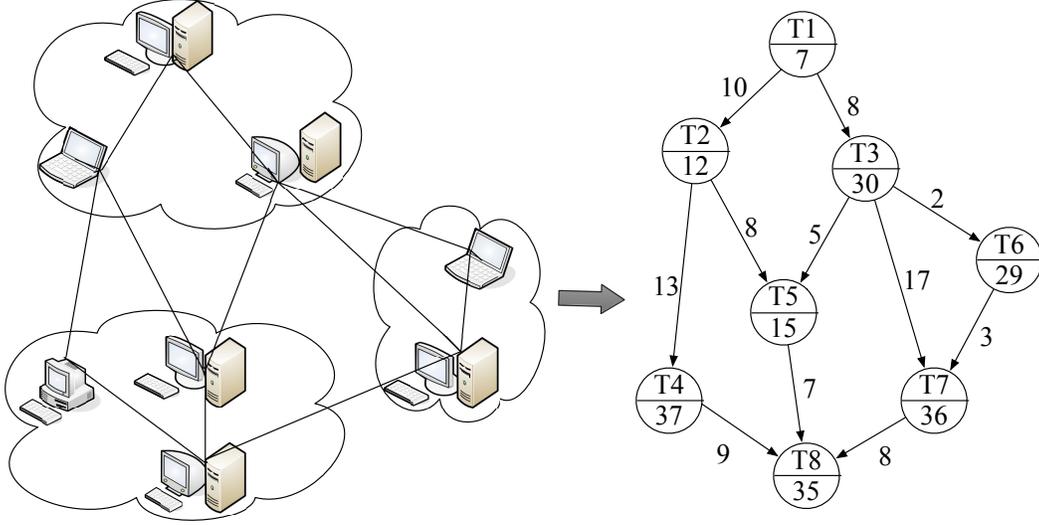


Figure 3.3: Representation of a cloud workflow by a direct acyclic graph (DAG).

Problem Definition

In our model, a given set of cloud workflows $\{\omega_l \mid l = 1, \dots, n\}, \omega_l \in W$, consists of computing-intensive tasks $(\tau_1, \tau_2, \dots, \tau_v)$. The workflows need to be scheduled in the cloud resource farm R_k ($1 \leq k \leq m$), provided that resource $R_k \in R_\phi$ is available for processing at time $P(t)$ such that $\sum_{R_k \in R_\phi} Q_{\tau_i \omega_l R_k} = 1$, where $Q_{\tau_i \omega_l R_k}$ determines to which resource a certain workflow is scheduled. Recall from Fig. 3.2 that each tenant's information is stored in the tenant information and performance repository.

$$Q_{\tau_i \omega_l R_k} = \begin{cases} 1 & \text{if task } \tau_i \text{ of workflow } \omega_l \text{ is allocated to resource } R_k \\ 0 & \text{otherwise.} \end{cases}$$

During the scheduling process, each tenant submits a service request, i.e., a workflow to the workflow scheduler with the resource requirements given by (ID, t_l, VM_l, D_l) , where ID, t_l, VM_l represent the tenant ID, reservation time slot, number of VMs required for ω_l , and associated deadline of each workflow, respectively. Note that tenant ID is very important to create an isolation environment for tenants that separates one tenant context (tenant's information) from another. The objective of the workflow scheduler is to schedule a workflow ω_l to a cloud resource R_k ($1 \leq k \leq m$) for a given time constraint (task τ_i of the workflow ω_l has to complete its execution before the deadline). No task is scheduled for any resources that are not available yet, i.e., $t_{\tau_i \omega_l} \geq p_k Q_{\tau_i \omega_l R_k}$, whereby $t_{\tau_i \omega_l}$

is the starting time of task τ_i of workflow ω_l and p_k represents the time when the resource becomes available, $t_{\tau_i\omega_l} \geq 0$.

We assume that a child task cannot be executed until all of its parent tasks are completed. In a multi-tenant environment, resources are virtualized and shared among many tenants. Minimizing the completion time and tardiness of the workflow application in such a shared environment poses a challenging problem while taking the resource utilization into account. The CWSA algorithm aims at improving the performance of the metrics described below in Section 3.4.3 in more detail.

3.4.3 Cloud Workflow Scheduling Algorithm

Performance Metrics: We define the following performance metrics that are used to study the efficiency and robustness of our proposed workflow scheduling algorithm.

Makespan: The makespan or completion time is a measure of the throughput of the system. Clearly, the main design objective of an efficient scheduling is to minimize the average makespan. The minimum makespan implies a high utilization of computing machines, leading to a higher throughput. The makespan is calculated for each scheduled workflow as the time from submission to completion of task, i.e., denoted by $makespan = SCT$. The workflow makespan C_W as the maximum completion time of all its tasks, is expressed as follows:

$$C_W = \max_{\tau_i \in \omega_l} \{SCT(\tau_i)\}. \quad (3.1)$$

Further, the optimization rate of the makespan (OMS) is given by:

$$OMS = (MMS - MS_{min})/MMS, \quad (3.2)$$

where MMS denotes the mean makespan and MS_{min} denotes the minimum makespan.

Tardiness: A delay penalty with weight is charged, if the task τ_i is completed after its due time. The delay penalty of a given task is defined as its tardiness. The tardiness of a workflow task indicates the time span of the completion time exceeding the projected due time in a fine-grained manner. The minimization of mean tardiness has been used as the primary objective in scheduling

tasks in order to meet a given task's due time, which can be computed as follows:

$$\overline{td}_i = \sum_{i=1}^N [\max(C_i - d_i, 0)]/v, \quad (3.3)$$

where $C_i - d_i \geq 0$ and the maximum tardiness is given by:

$$td_i^{max} = \max_{i=1 \dots v} [\max(C_i - d_i, 0)], \quad (3.4)$$

where C_i , d_i , and v denote the completion time of task τ_i , due time (i.e., sum of arrival time and total processing time of task), and number of tasks in a workflow, respectively.

Laxity: The laxity of a task is the measurement of its urgency. At time t the laxity of task τ_i is $(d_i - t - p)$, where d_i is the task's deadline and p is its remaining computing time requirement. When the laxity of a task is negative, the execution of the task can not meet its deadline. If the laxity of a given task is zero then the execution of the task should start immediately. In case of a positive laxity, the execution of the task can be delayed and thus placed in the queue.

Mean scheduling execution time: This metric accounts for the total time taken by the workflow scheduler to execute workflow tasks using a particular scheduling algorithm.

Resource utilization rate: It is defined as the percentage of time that a resource is busy. Therefore, a better performance can be achieved only through a higher utilization of resources, while meeting given deadlines and QoS requirements of workflow applications. The average resource usage is computed as follows:

$$\overline{RU} = \frac{\sum_{i=0}^{n-1} (RU_{t_i} * (t_{i+1} - t_i))}{t_n}, \text{ where} \quad (3.5)$$

$$RU_i = \frac{R_{act}}{\min(R_{ava}, R_{req})}, \quad (3.6)$$

where n , R_{act} , R_{ava} , and R_{req} denote the number of observations, number of active resources, number of available resources, and required resources for the workflows, respectively.

Makespan standard deviation (MSD): It is used to measure the robustness of scheduling schemes.

The MSD is calculated as follows:

$$\sigma_m = \sqrt{\frac{1}{\mathcal{N}-1} \sum_{a=1}^{\mathcal{N}} (X_a - \bar{X})^2}, \quad (3.7)$$

where σ_m , X_a , \mathcal{N} , and \bar{X} denote the sample standard deviation of the makespan, data set of the makespan, size of the sample, and average value of the makespan, respectively. Note that smaller values of MSD are more likely to result in a stable system performance.

Skewness of makespan: It measures the symmetry of the makespan distribution and can be calculated by using the adjusted Fisher-Pearson standardized moment coefficient, as follows:

$$\gamma(X) = \frac{\mathcal{N}}{(\mathcal{N}-1)(\mathcal{N}-2)} \sum_{a=1}^{\mathcal{N}} \left(\frac{X_a - \bar{X}}{\sigma_m} \right)^3. \quad (3.8)$$

A negative value of skewness indicates that a tail can be found on the left-hand side of the makespan distribution, whereas a positive value of skewness means the opposite.

Proposed Algorithm: The pseudocode of our proposed CWSA algorithm is shown in Algorithm 3.1. The algorithm works as follows. Initially, the resource nodes are sorted in descending order based on their computational speeds. The objective to do so is to select the lowest execution cost for ready workflow tasks (lines 3-6). The scheduler checks the task dependency at scheduling time to verify which tasks can be scheduled one after another. This is done through a depth-first search. When the workflow is submitted to the scheduler, the workflow tasks will be inserted into a service queue (lines 6-7). The ready workflow tasks are sorted according to a deadline priority. Next, an appropriate *schedulegap* is calculated (see also Eq. 3.9), i.e., a suitable time-slot of resource nodes for every ready workflow task (lines 8-11).

The *schedulegap* is the time period of an idle CPU. It occurs when the currently available CPUs is more powerful with respect to the existing schedule than the number of CPUs requested by the given workflow applications in the given time period. When a new task arrives, a better schedule position is selected according to the current schedule position. For the set of workflows $\omega_l \in W$ with deadlines D_l , $l = 1, 2, \dots, n$, the necessary and sufficient condition for the feasibility of a workflow schedule with schedule utilization $SU(t)$ is defined as the utilization by the current task invocations

at time t , which can be expressed as follows:

$$SU(t) = \sum_{TI_i \in CI(t)} (C_i/T_i) \leq 1, \quad (3.9)$$

where TI_i is the task invocation, $CI(t)$ denotes the set of current invocations, and C_i represents the worst-case computing time (i.e., largest time between release and termination). Further, T_i denotes the period of workflow task and (C_i/T_i) denotes the fraction of processor time spent on the execution of the workflow tasks. The counter keeps track of the schedulable utilization at run-time. The counter is set to zero ($SU = 0$) at the initial stage of each schedule gap. When a new workflow task is invoked, it is incremented by (C_i/T_i) , i.e., if $SU + C_i/T_i \leq 1$ then the request for schedule is admitted at current time plus its deadline and is decremented by (C_i/T_i) , if the deadline of the current workflow task's invocation is reached. The algorithm searches for *schedulegaps* on every resource.

The workflow scheduler is able to execute different scheduling schemes. If the *schedulegap* is not found, then the scheduler can schedule the workflows using other schemes like FIFO, EASY Backfilling, and MCT (lines 12-20). For Instance, if *SchedulingPolicy* == 1 then schedule of the workflow is determined by using the FCFS. With FCFS, incoming workflows are sorted in the scheduler queue in their arriving order. Based on this order, if the required resources are available to execute the first workflow it is immediately scheduled. Otherwise, the workflows wait until the resources become available for them.

If *SchedulingPolicy* == 2 the workflow is scheduled by using the EASY Backfilling. In the EASY Backfilling, workflows are sorted based on their deadline and are placed in the queue accordingly. The EASY Backfilling works similarly to FCFS. However, if the first workflow can not be scheduled due to the lack of required resources, the scheduler calculates the earliest start time based on the running workflow. Subsequently, it makes a reservation for the first workflow. Once the resources become available the workflow is scheduled. If a workflow in the backlog can be executed on time without delaying others, it can be moved forward in the queue and be executed earlier. Thus, ideally resources are backfilled with suitable workflows in order to achieve a higher resource utilization.

In the case of *SchedulingPolicy* == 3, the workflow is scheduled by using the MCT. With MCT, the workflow scheduler assigns the workflows in an arbitrary order to the available cloud

Algorithm 3.1 Cloud Workflow Scheduling Algorithm (CWSA)

Require: A Workflow W is defined in a workflow repository. $NewSchedule = 0$.

Ensure: Workflow schedule within the deadline, if the deadline exists.

```
1: repeat
2:    $W \leftarrow$  unscheduled workflow in the list
3:   for all  $r_k \in R$  do
4:     insert  $r_k$  into  $InitialSchedule(IS)$  in descending order based on their computational
       speed.
5:   end for
6:   for all  $\omega_l \in W, r_k \in R$  do
7:     Insert  $\forall$  ready workflow  $\omega_l \in W$  into service queue then traverse the workflow by using a
       depth-first search.
8:     for all  $r_k \in R$  do
9:        $schedulegap \leftarrow$  find  $schedulegap(IS)$ 
10:      Calculate a  $schedulegap$  as described in Eq.(9).
11:    end for
12:    if There does not exist a  $schedulegap$  on the resource nodes  $r_k \in R$  and  $\exists$ 
        $SchedulingPolicy, \forall$  available scheduling, Select  $SchedulingPolicy$  then
13:      switch ( $SchedulingPolicy$ )
14:        case 1:
15:           $NewSchedule \leftarrow$  Schedule workflow task  $\tau_i \in T$  on the resource node  $r_k \in R$  using
            the FCFS scheduling.
16:        case 2:
17:           $NewSchedule \leftarrow$  Schedule workflow task  $\tau_i \in T$  on the resource node  $r_k \in R$  using the
            EASY Backfilling scheduling.
18:        case 3:
19:           $NewSchedule \leftarrow$  Schedule workflow task  $\tau_i \in T$  on the resource node  $r_k \in R$  using the
            MCT scheduling.
20:        end switch
21:      else if ( $\exists schedulegap$  on the resource nodes  $r_k \in R$  & & case = = 4) then
22:        Call function CWSA()
23:      end if
24:    end for
25: until all the workflows have been scheduled.
```

resources such that the workflow will have the minimum completion time. Toward this end, the scheduler firstly chooses a workflow arbitrarily from the service queue. Secondly, it finds the cloud resource that gives the minimum completion time for the chosen workflow. Thirdly, the schedule maps the workflow to the chosen cloud resource and removes the workflow from the service queue. Then, the available time of the resource is updated. The scheduler repeats this process until all workflows have been scheduled and assigned to the resources.

Afterwards, in the case that a $schedulegap$ is found, the workflow scheduler executes the CWSA (see lines 21-22 and Function 3.1). All suitable resource nodes are tested whether a suitable

Function 3.1 Function CWSA ()

```
1: function CWSA ()
2: NewSchedule  $\leftarrow$  InitialSchedule
3: if totalweight > 0 (as described in Eq.(10)) then
4:   for all  $\omega_l \in W, r_k \in R$  do
5:     NewSchedule  $\leftarrow$  Move workflow tasks into found schedulegap. That means in the appropriate scheduling position, i.e., a virtual machine already started.
6:   end for
7: else
8:   remove the workflow from schedule
9: end if
10: end function
```

schedulegap for new workflow tasks exists in their schedules. At regular intervals, if the task queue is not empty and idle resource(s) exist(s) in the cloud resource pool, the workflow scheduler tries to find a suitable resource for the workflow tasks in the resource pool. In Function 3.1, line 4, the value of the total weight is checked. Note that the *totalweight* denotes the highest value of the weight for each assignment of a new workflow task, which is set to the ideal CPU gap according to Eqs. (3.10 – 3.12). Thus, the *totalweight* can be defined as follows [170]:

$$totalweight = weight_{makespan} + weight_{deadline}, \quad (3.10)$$

where,

$$weight_{makespan} = \frac{(makespan_{old} - makespan_{new})}{makespan_{old}}, \quad (3.11)$$

$$weight_{deadline} = \frac{(nondelayed_{new} - nondelayed_{old})}{nondelayed_{old}}, \quad (3.12)$$

whereby *makespan_{old}* and *makespan_{new}* represent the expected makespan of the current schedule of workflow tasks and the makespan of the new schedule of workflow tasks, respectively. Further, *nondelayed_{old}* and *nondelayed_{new}* denote the number of workflow tasks executed within the deadline before and after the workflow task assignment, respectively. The proposed algorithm searches for all gaps on every available resource and selects the best gap. If *totalweight* > 0 (see Function 1, line 3), the current mapping is taken as the best schedule. If there is no proper schedule in the beginning of the schedule, the algorithm searches for the gap between the task on the current position, τ_i , and the next task τ_{i+1} . If all the schedule positions are tested and there no better performance has been

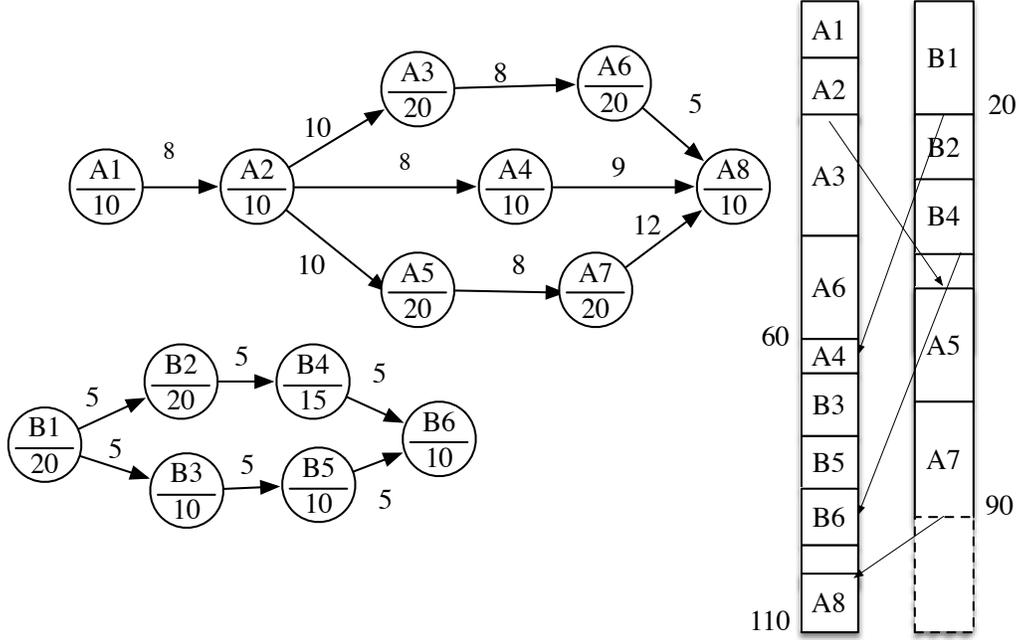


Figure 3.4: Illustration of scheduling multiple workflows.

found, the workflow returns to the initial position. Note that *NewSchedule* has a lower number of tardy tasks, if the value of $weight_{deadline}$ is positive.

Fig. 3.4 depicts an illustrative example of scheduling multiple workflows. Two workflows, W_A and W_B , are to be scheduled on two resources, R_1 and R_2 . For workflow W_A , the scheduler first assigns $\{\tau_{A1}, \tau_{A2}, \tau_{A3}, \tau_{A6}\}$, $\{\tau_{A5}, \tau_{A7}\}$, and $\{\tau_{A4}, \tau_{A8}\}$, respectively. The scheduler schedules $\{\tau_{B1}, \tau_{B2}, \tau_{B4}\}$ on the gap found before τ_{A5} . Similarly, tasks $\{\tau_{B3}, \tau_{B5}, \tau_{B6}\}$ are scheduled on the gap between τ_{A4} and τ_{A8} . Task τ_{A5} starts at time 50, when its predecessor has already finished its execution. The time complexity to find the resource, which has minimum execution time, is $O(mW)$, where W is the number of workflows and m denotes the number of allocated resources.

Advantages of the CWSA Algorithm: In the following, we discuss the superiority of CWSA over the alternative scheduling schemes under consideration. FCFS uses the time instance when a workflow arrives at the cloud scheduler to define the priority for all requests associated with the workflow. This algorithm is inefficient as for increasing workload many workflows waiting for execution may experience unnecessary idle time of some resources. The EASY Backfilling is an example of production batch schedulers, which is similar to FCFS but enables backfilling in order to reduce resource fragmentation. It processes the first task in the queue and reserves the earliest

possible time slot. In principle, it is able to run under the FCFS, but other tasks in the queue are scheduled opportunistically as nodes become available as long as they do not interfere with the reservation of the first task. However, re-computing the tasks' priorities and sorting them at each scheduling event may cause tasks to be delayed. On the other hand, MCT assigns tasks in an arbitrary order to the worker node based on their minimum completion time. This may cause some tasks to be assigned to resources that do not provide the minimum execution time.

In contrast to these schemes, our proposed CWSA algorithm exhibits less context switching and thus outperforms the former ones. Context switching corresponds to the time period needed for switching between two tasks, i.e., bringing a waiting task into execution and sending an execution task into terminate/waiting state. If the total time of execution of all tasks is assumed to be M , then the context switching time equals $M - [\text{sum of all tasks (waiting time + execution time)}]$. We executed each scheduling algorithm and recorded the context switching information with our script using Linux system call *schedule()*; *schedule()* calls *context-switch*, which is responsible for switching from one task to another one when the new task has been selected by *schedule().context-switch()*. Alternatively, *lmbench* and *Perf* could be used to trace context switching information.

The main advantage of the CWSA is its ability to allocate resources at a higher speed for increased schedule lengths of workflows. If shorter schedule lengths are required, it assigns the workflows to the resources at lower speed. CWSA takes advantage of the gaps between scheduled tasks. These idle periods can be used to schedule other tasks, thereby reducing the overall makespan. Further, the execution time of CWSA is shorter than that of the other considered schemes, as discussed next.

3.5 Implementation and Performance Evaluation

This section describes the experimental setup and simulation results. Further, a proof-of-concept experiment is presented to validate our solution using real-world scientific workflow applications, e.g., SIPHT and CyberShake.

3.5.1 Experimental Setup

The performance of the proposed scheduling algorithm is thoroughly evaluated using a discrete event cloud simulator based on the CloudSim framework [171]. CloudSim supports the modeling and simulations of large scale cloud computing environments on a single computing node, including service brokers, resource provisioning, datacenters, and allocation policies. Interested readers may refer to [171] for further information about CloudSim. We extended CloudSim to support the components shown in the proposed architecture in Fig. 3.2 and scheduling algorithms. Since workloads have different characteristics, no single elasticity (auto-scaling, i.e., resource over-provisioning and under-provisioning) algorithm is suitable for all workloads. There are mainly reactive (rule-based) and predictive (proactive) auto-scaling techniques classified in the literature. In this work, we implemented rule-based auto-scaling in the QoS monitor component of Fig. 3.2. Note that most of the cloud service providers (e.g., Amazon EC2 and Rightscale) also apply rule-based mechanisms to scale up/down VMs. For instance, rules like: monitor CPU utilization (U) every 2 min, [**scale-up**] If $U \geq 70\%$ for 10 min, then add 1 VM of small size, wait 4 consecutive 1 min intervals; [**scale-down**] If $U \leq 30\%$ for 12 min, remove 1 VM of small size, wait 5 min consecutive 1 min intervals. For VM reconfiguration, we used an approach similar to the rule-based approach described in [172], we designed rules for auto-scaling VMs. The behavior of auto-scaling can be controlled by changing the configuration file, where rules are defined. The rules specify the upper and lower bounds of the number of VMs and the conditions to triggers scaling. The infrastructure level (physical and virtual) of Fig. 3.2 is modeled by the core layer representing the original CloudSim datacenter, which encapsulates sets of computing hosts that can be homogeneous/heterogeneous with respect to the configuration of their hardware, i.e., CPU cores, storage, memory, and bandwidth. A single datacenter is considered in our cloud model and the datacenter is assumed to have sufficient resources. The VMs/cloud resources are modeled Amazon AWS EC2⁶ standard instance types and the parameters relevant for the experiment are shown in Table 3.1. Each newly provisioned VM needs several minutes to be booted-up. Therefore, a boot-up time of 97 seconds is considered for each instance as in [173].

⁶Amazon Elastic Compute Cloud (Amazon EC2), <https://aws.amazon.com/ec2>.

⁷One EC2 Compute Unit (ECU) is defined as the CPU power of a 1.0-1.2 GHz of a 2007 Opteron or 2007 Xeon processor, as specified in Amazon EC2 documentation. At the peak performance, one ECU equals 4.4 gigaflops per second (GFLOPS) (see [156] and references herein).

Table 3.1: Resource types and prices used (based on on-demand instances offered by Amazon EC2, US East (N. Virginia))

Type	Core	ECU ⁷	RAM (GB)	Storage (GB)	Price (\$/hr)
m1.small	1	1	1.7	1×160	0.044
m1.medium	1	2	3.75	1×410	0.087
m1.large	2	4	7.5	2×420	0.175
m1.xlarge	4	8	15	2×840	0.35

Table 3.2: Workflow parameters and default values

Parameters	Value
Total number of workflow	1-20
Number of node per workflow	3,000
Bandwidth	100-1000 Mbit/s
Baud rate	10,000 Kbit/s

Similar to the deadline assignment in [174], the deadlines are calculated by identifying the smallest amount of time required to execute a single workflow [$D_{min} = \min CP(\omega_i)$], which is the length of the critical path (CP) for the workflow with the shortest critical path and the time required to execute all the workflows equals [$D_{max} = \sum CP(\omega_i)$]. The range [D_{min}, D_{max}] is then divided into equal intervals. We assume that each workflow has a D_{max} , in $2CP$, $3CP$ and $4CP$, where CP is the execution time of tasks in the critical path of the DAG at the best available resource.

We used a VM billing interval of 1 hour (i.e., cost per instance per hour). Thus, the usage is rounded up to the nearest hour and any partial hours are counted as full hours (e.g., 1.1 hours is rounded up to 2 hours). For illustration, the pricing is designed based on Amazon’s `m1.small` instance type of EC2 US East region (i.e, 1.7 GB of memory, 1 virtual core, 160 Gb of instance storage, and instance price of \$0.044 per hour). Other simulation parameters and their default values are listed in Table 3.2.

3.5.2 Simulation Results

This section presents the simulation results and discusses the obtained findings. We focus on finding the effective makespan, execution time, resource utilization, and total tardiness.

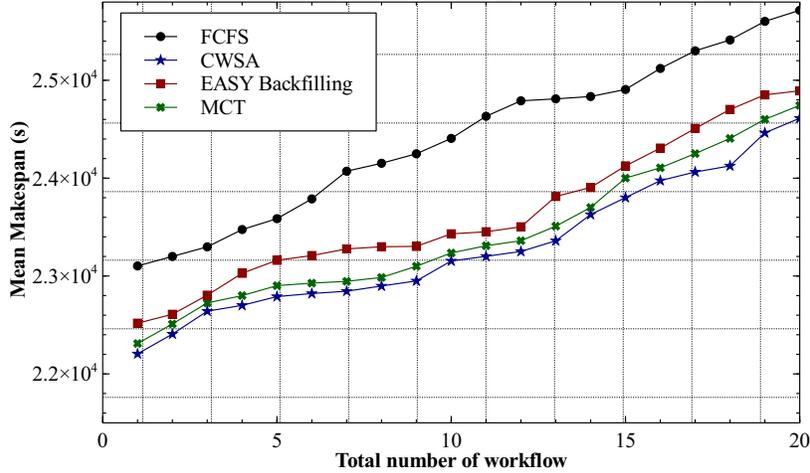


Figure 3.5: Mean makespan vs. total number of workflow.

Makespan: This experiment compares the makespan of different scheduling schemes. The mean makespan is depicted in Fig. 3.5. They are admitted to the workflow scheduler, which utilizes the CPU gap to minimize idle time and thus improve throughput. The highest makespan improvement was obtained for CWSA. On average, CWSA is 57% better than FCFS, 30% better than EASY Backfilling, and 16% better than MCT. Our proposed CWSA algorithm is able to efficiently utilize the resources, thereby enabling that more tasks are completed in a shorter time. The trend of the figure indicates that the makespan is slightly increasing for a growing number of workflows. In fact, with FCFS the makespan increases significantly, while with CWSA the increase is less pronounced compared with the FCFS, EASY Backfilling, and MCT.

Skewness of makespan: The skewness of makespan measures the degree of asymmetry in the makespan data set. We calculated the skewness of makespan and list the obtained results in Table 3.3. We observe that the makespan has an asymmetric distribution. The workloads are skewed, i.e., they experience abnormally high and abnormally low values as they always change, which in turn may degrade the performance.

Network Impact: Next, let us evaluate the network impact of the scheduling schemes. To do so, we calculate the average makespan, skewness of the makespan, and standard deviation of the makespan for different scheduling schemes. These values are listed in Table 3.2. We observe that the average makespan time is shorter with CWSA and longer with FCFS. The higher value of the standard deviation stipulates that the communication costs vary significantly among the tasks of the workflow.

Table 3.3: Average makespan, skewness of the makespan and standard deviation of the makespan for different scheduling.

Scheduling schemes	Average makespan (s)	Skewness of the makespan	Makespan standard deviation
FCFS	24422.62	-0.1297	819.50
CWSA	23294.57	0.4475	686.67
EASY Backfilling	23634.90	0.3884	724.20
MCT	23421.51	0.4509	714.68

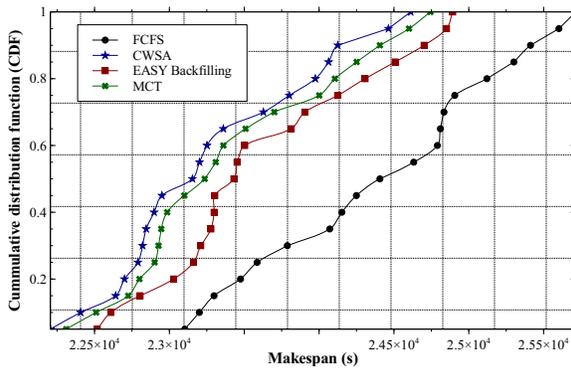


Figure 3.6: Cumulative distribution function (CDF) of makespan for different scheduling schemes.

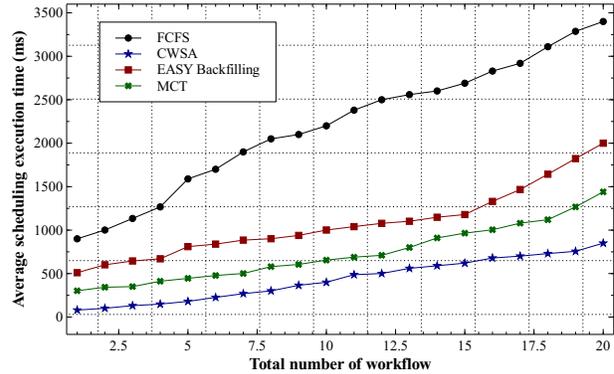


Figure 3.7: Average execution time of scheduling schemes.

Performance distribution using cumulative distribution function (CDF): Next, we study the distribution of the performance variation for different scheduling schemes using the CDF of makespan. The CDF captures the entire performance distribution variation. Fig. 3.6 depicts the CDF of the makespan for the FCFS, CWSA, EASY Backfilling, and MCT. As we can see, CWSA outperforms the other schemes in terms of makespan and it represents the most reliable scheduling algorithm, i.e., the distribution of makespan time of the CWSA tends to have a shorter makespan than the others, as shown in Fig. 3.6. The FCFS and EASY Backfilling scheduling, however, have a steep tail. Conversely, the FCFS has 25% of all makespans being over 24900 seconds.

Execution Time: This experiment evaluates the scalability of scheduling schemes. The average execution time to execute the scheduling schemes averaged over 100 runs with number of workflows. The performance result of each scheduling algorithm with 20 workflows is depicted in Fig. 3.7. The execution time of the CWSA is smaller than that of the FCFS, EASY Backfilling, and MCT. On average, the CWSA is 91% faster than FCFS, 82% faster than EASY Backfilling, and 70% faster than MCT. The execution time appears to grow linearly with the number of workflows. We

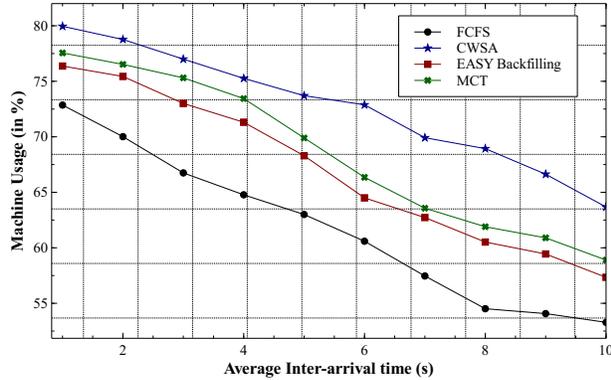


Figure 3.8: Resource utilization.

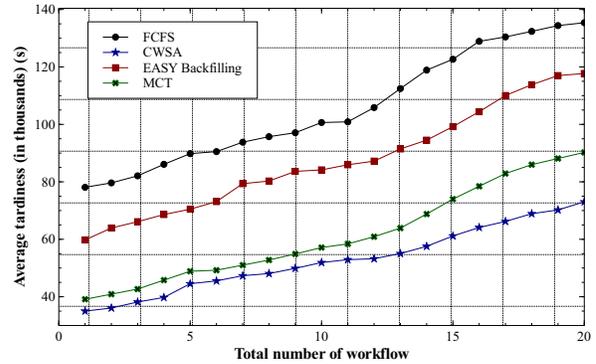


Figure 3.9: Average tardiness vs. number of workflows.

observe that in the case of CWSA, the execution time is lesser and the scalability is better with the increasing number of workflows among others as it efficiently schedules the workflow. Note that the time complexity measures the amount of time required to execute an algorithm, which is given by the upper bound of the amount of workflow tasks performed.

Resource Usage: In this experiment, we evaluate the resource usage. Fig. 3.8 depicts the percentage of resource utilization of each scheduling algorithm. A resource node can be idle when it does not support all the hardware/software requirements required by a queued task. The CWSA exhibits a better resource utilization. On average, in the case of the CWSA, the resource utilization is 19% better than that of FCFS, 9% better than that of EASY Backfilling, and 7% better than that of MCT. This is due to the fact that the CWSA is able to schedule the task in such a way that it keeps the system resources always busy and efficiently utilizes the system’s computational resources. Further, the CWSA utilizes free resources by executing unscheduled tasks in advance in order to minimize costs. This indicates that maximizing the resource utilization is an effective solution to minimize computing units via an imposed deadline.

Tardiness: This experiment evaluates the tardiness of the workflows. Fig. 3.9 shows the average tardiness for different scheduling schemes. In comparison with FCFS, EASY Backfilling, and MCT, the average tardiness values are smaller in the case of CWSA. On average, the CWSA improves the tardiness by 48% compared to FCFS, 37% compared to EASY Backfilling, and 10% compared to MCT. The average tardiness depends on the number of workflows to be executed on the available machines. The trend of the graph shows that tardiness slightly increases for an increasing number of workloads. This clearly indicates that the total tardiness strongly depends on the number of

Table 3.4: Characteristics and categorization of scientific workflow applications

Tenant workload type	Workflow name	Number of tasks	Type (I/O Read/Write (GB), Peak memory (MB), CPU (hours))
Small	CyberShake	30, 50	High, High, High
	SIPHT	30, 60	Low, Medium, Low
Medium	CyberShake	100	High, High, High
	SIPHT	100	Low, Medium, Low
Large	CyberShake	1000	High, High, High
	SIPHT	1000	Low, Medium, Low

available instances of virtual machines and workload. Intuitively, as the load increases, more workflows are subjected to become tardy and may also experience a longer period of tardiness. Such a situation may cause a time-critical workflow application to have an inopportune behavior. However, notice that since the CWSA has a better resource utilization (see Fig. 3.8), more tasks are able to be completed in a shorter time (as execution time is faster than that of others, see Fig. 3.7). Moreover, CWSA is able to utilize the schedule gap more effectively and therefore exhibits the smallest tardiness.

In overall, the proposed scheduling algorithm has shown its effectiveness to improve the makespan and tardiness. On the other hand, the execution time of CWSA algorithm is notably smaller than that of other scheduling schemes. The overall cloud resource utilization is important from a resource owner’s point of view to optimize his or her profit. As illustrated in Fig. 3.8, a better resource utilization is achieved with the proposed CWSA algorithm.

3.5.3 Proof-of-Concept Experiment

This section presents the proof-of-concept experiment. We evaluate the performance of the proposed CWSA algorithm with real-world scientific workflow applications, such as SIPHT (a typical structure of SIPHT workflow is illustrated in Fig. 1) and CyberShake workflows. Note that these workflows are widely considered as a benchmark in the literature. Importantly, they represent a wide range of application domains and have a diverse resource requirements. For instance, SIPHT workflow, from the bioinformatics project at Harvard, is used to automate the process of searching for sRNA encoding-genes for all bacterial replicons in the National Center for Biotechnology Information (NCBI) database [140]. CyberShake workflow is used the Southern California Earthquake Center

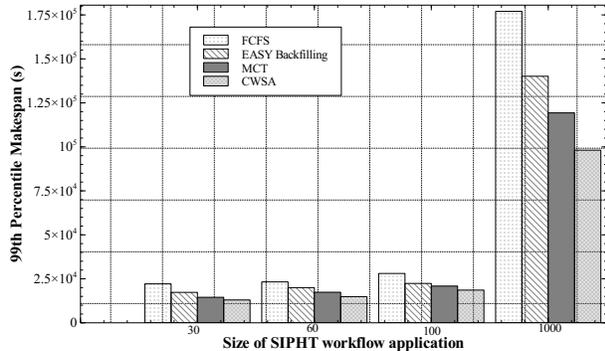


Figure 3.10: 99th percentile makespan of different scheduling schemes for the SIPHT workflow application.

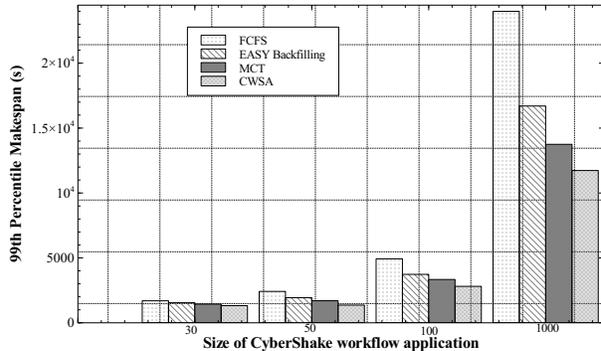


Figure 3.11: 99th percentile makespan of different scheduling schemes for the CyberShake workflow application.

to characterize earthquake hazards by generating synthetic seismograms. CyberShake can be also classified as a data-intensive workflow with large memory and CPU requirements (see Table 3.4). Interested readers may refer to [175] for further details on these workflows.

Experimental Setup: In this experiment, we used the experimental setup, as described in Section 3.5.1. The additional parameters used in the experiment such as the characteristics and categorization of the considered scientific workflow applications are shown in Table 3.4. The medium and large size of workflow applications are more compute-intensive. The structure of these workflows comprise several components such as pipeline, data distribution, data aggregation, and data redistribution. The analysis and resource management for such applications are complex due to the nature of the unstructured (asymmetric) workflow hierarchy and the fact that they cover a wide range of application domains and have intensive resource requirements. We have generated these workflows with the Pegaus workflow system [175]. It generates the DAX (Directed Acyclic Graph in XML format) of these workflow applications for a given number of tasks. The DAX file for workflow contains a list of tasks and the dependencies between them as well as the computation time, and input/output data size of each task.

Results: We have run each scheduling algorithm 100 times for each workflow application. We evaluated the performance using the 99th percentile distribution of the makespan to represent statistical measures of the scheduling schemes for the SIPHT and CyberShake workflow applications. The 99th percentile distribution better characterizes the makespan distribution. Figs. 3.10 and 3.11 compare the 99th percentile makespan of the different scheduling schemes for the SIPHT and CyberShake workflow across a range of application sizes, respectively. Both figures show that at

Table 3.5: Average cost (\$) of scientific workflow execution

Workflow		Scheduling schemes			
		FCFS	EASY	MTC	CWSA
CyberShake	small	22.73	22.74	11.37	11.36
	medium	45.46	45.46	22.73	22.73
	large	136.36	113.64	91.94	90.91
SIPHT	small	159.09	113.64	91.91	90.90
	medium	181.82	159.09	136.40	136.36
	large	1136.36	886.36	772.73	636.36

small workloads, i.e., the size of tasks is small, there exists no significant difference of the makespan between the scheduling schemes. This is because when the number of small tasks of CyberShake and SIPHT decreases, the number of allocated computation instances also decreases. As the size of workflow application increases, the makespan quickly approaches the 99th percentile value. In both cases, CWSA performs much better than the other schemes for large sizes of workflow. Overall, the CWSA scheduling algorithm outperforms the other schemes for both workflow applications.

Cost Evaluation: The total cost of the execution of a workflow is defined as the product of the total execution time (also includes VM overhead/boot-up time) of a workflow and per hour cost of an instance type. The fractional consumption hours are rounded up. Note that the cost of the execution of workflows not only depends on the scheduling algorithm but also on the choice of VM instance types (see also Table 3.1) and the financial budget of a tenant. That means cheaper resources may be attractive, even though they might degrade the performance. As a result, the overall costs may be higher. Note also that the charge of instances is not necessarily proportional to its computing power⁸.

The average execution cost obtained for each scientific workflow is shown in Table 3.5. From the results, we observe that the proposed CWSA performs better than others in terms of cost by generating much cheaper schedules. The CWSA shows a considerably lower cost for CyberShake application while having a lower makespan. On the other hand, as expected, SIPHT is a little costly than CyberShake in every workload type. However, SIPHT large workflow experienced significant performance gains compared to other scheduling schemes. This shows that the cost of the execution of workflow varies depending on the application and on the size of the workflow as

⁸Amazon EC2 Pricing: <https://aws.amazon.com/ec2/pricing/>

well as the structure of the workflow. Overall, we noticed that the proposed solution is not only a cost-effective but also a faster solution for the considered configurations. It is worthwhile noting that even though we considered an on-demand billing model, for the long-term usage, reserved VM instances are much cheaper than on-demand model (see also Amazon EC2 pricing).

3.6 Conclusions

Cloud computing has been widely recognized as an essential computing paradigm to execute compute- and data-intensive business process workflow (e.g., media processing, analytics pipelines, orchestration of services, coordinating resources) and scientific workflow applications for processing of large sets of scientific data, as witnessed by the recent work on Amazon SWF (Simple Workflow Service).

In this chapter, we introduced a four-layered workflow scheduling system. A novel CWSA algorithm was proposed for scheduling workflow applications in a multi-tenant cloud computing environment. An analysis of different performance metrics was carried out. An extensive simulations was performed to evaluate the performance of the proposed scheduling algorithm. The performance of the CWSA was then compared with different scheduling schemes to highlight the performance and robustness of the proposed solution. The obtained results show that our CWSA outperforms other scheduling schemes. Importantly, CWSA was shown to utilize computational resources properly by reducing idle time of cloud resource nodes. Further, we conducted proof-of-concept experiments by employing real-world scientific workflow applications. The proof-of-concept experiment indicates that the proposed CWSA offers significant improvements for larger workflow applications. Importantly, a key lesson learned from this study is that multi-tenancy helps improve the utilization of resources.

Although we have demonstrated the advantages of multi-tenant cloud environments for scheduling workflow applications, there are several potential directions for future work, including the development of a complex model of scheduling schemes by considering resource failures and complex reservation scenarios for multi-tier application scaling, where scaling may affect different applications. For the future, we intend to further investigate the optimization of the CWSA algorithm and apply it in the context of mobile cloud computing.

Chapter 4

Cloudlet Enhanced Fiber-Wireless Access Networks for Mobile-Edge Computing

4.1 Introduction

Cloudlets were envisioned to support emerging resource-intensive and latency-sensitive applications by providing access to computing resources for nearby mobile devices, especially for those with limited resources and capabilities such as battery-constrained devices. The cloudlet is a new architectural element that deploys the three-layer (mobile device-cloudlet-cloud computing) service delivery model [15]. Cloudlets, micro-datacenter⁹, or the more recently introduced concept of fog computing [81] can be further enhanced to give rise to *Mobile-Edge Computing (MEC)* [42]. MEC may be viewed as a decentralized proxy cloud server or a local datacenter that offers cloud services (e.g., processing, storage, and caching) at the network edge in close proximity to mobile and emerging smart devices to enrich today's mobile broadband experience. The best way to achieve this is to offload compute-intensive tasks onto a cloud-like infrastructure in the vicinity of end-users. Note

This chapter is based on the following publication:

[107] B. P. Rimal, D. Pham Van, and M. Maier. Cloudlet Enhanced Fiber-Wireless Access Networks for Mobile-Edge Computing. *IEEE Transactions on Wireless Communications*, IEEE Xplore Early Access, DOI: 10.1109/TWC.2017.2685578.

⁹A small modular datacentre system includes less than 10 servers and 100 VMs in a single box.

that conventional cloud-based offloading techniques, e.g., [97–99, 176], may not be suitable for this since they incur high latency between mobile devices and the remote public cloud server.

While MEC holds promise to enhance users’ mobile experience, building a network infrastructure to support MEC capabilities is challenging. Towards this end, integrated fiber-wireless (FiWi) access networks combine the high capacity, scalability, and reliability of optical fiber networks with the flexibility and ubiquity of wireless networks to provide broadband services for not only mobile users but also fixed subscribers [63, 114]. A promising approach to realize MEC in a cost-effective manner is to integrate cloudlets at the edge of FiWi networks, giving rise to MEC integrated FiWi networks. These highly integrated networks help reduce deployment and operation costs by sharing existing and widely deployed optical fiber infrastructures, e.g., passive optical networks (PONs), for fixed broadband access, mobile backhaul, and cloud services [30, 177].

MEC integrated FiWi networks impose unique challenges. For instance, MEC servers have to comply with the underlying network architecture, interfaces, and functionalities without degrading the FiWi network performance. It is also important to ensure that different involved stakeholders (e.g., operators) and diverse devices have access to shared but limited resources at the network edge [31, 178]. Furthermore, MEC integrated FiWi networks will need to cope with the coexistence of broadband access and MEC traffic. While both FiWi and MEC support triple-play traffic (i.e., voice, video, and data), MEC applications have their own and deterministic performance requirements such as real-time response, location and mobility awareness. It is challenging to ensure that current triple-play and/or critical MEC traffic is not jeopardized by other coexistent traffic.

Research in the area of MEC is still in its infancy. Recently, many industrial players (e.g., Akamai, Saguna) are aiming to propose proprietary MEC architectures and solutions. In addition, there exist a few preliminary studies on MEC, e.g., [26, 42, 179, 180]. The concept of MEC and some of its potential use cases were described in [26, 42, 179]. A brief survey on MEC was provided in [180]. However, to the best of the authors’ knowledge, no technical solution has been proposed to realize MEC, especially from a resource management perspective. This chapter studies the feasibility of MEC in FiWi broadband access networks. While most existing studies, e.g., MEC Industry Specification Group [42], consider only MEC implementation at 4G LTE macro-cell base stations, this chapter examines MEC over wireless local area networks (WLANs), given their low costs, wide deployments, and high capacity (e.g., 100 times higher than cellular networks [28]).

The envisioned MEC integrated FiWi network is built on the integration of cost-effective, simple, and capacity-centric Ethernet PON (EPON)/10G-EPON, Gigabit-class WLAN, and computation- and storage-centric cloudlets. To deal with the coexistence of FiWi traffic, i.e., human-to-human (H2H) and MEC traffic, this chapter proposes to incorporate offloaded traffic transmissions between edge devices (e.g., sensors, wearables) and cloudlets into the EPON/10G-EPON dynamic bandwidth allocation (DBA) procedure. To facilitate the incorporation process, the whole network is designed in two layers of time division multiple access (TDMA). In the first TDMA layer, the central optical line terminal (OLT) schedules timeslots and allocates aggregated bandwidth to remote optical network units (ONUs). In the second TDMA layer, a wireless access point (AP) collocated with an ONU, i.e., ONU-AP, assigns bandwidth in subslots and schedules transmissions of both conventional H2H and MEC traffic to its associated wireless stations (STAs). In this work, the integrated ONU-AP is referred to as the *edge of FiWi networks*. Note that even though other access mechanisms could be employed in the WLAN front-end, a vast majority of previous studies on FiWi networks used carrier sense multiple access with collision avoidance (CSMA/CA) as random MAC protocol in the WLAN front-end segment, e.g., [114]. However, this work considers MEC applications (e.g., face detection) that require deterministic access in order to minimize delay and energy consumption. Therefore, the layered TDM scheduling is chosen in the proposed solution.

The main contributions of this chapter are threefold. From an *architectural perspective*, for the first time, an MEC enabled FiWi access network architecture is introduced, which represents a cost-effective and reliable solution for enabling a new class of low-latency and/or resource-intensive MEC applications. Second, from a *resource management perspective*, a novel unified cloudlet-aware resource management scheme is proposed that enhances DBA in EPONs/10G-EPONs by effectively handling both H2H and MEC traffic without negatively affecting FiWi operations in the envisioned network. Third, a *comprehensive analytical framework* for evaluating the network performance in terms of packet delay for both H2H and MEC traffic, response time efficiency, offload gain-overhead ratio, energy efficiency, and device battery life is developed. Note that the combination of the response time efficiency, energy saving model, and analysis of packet delays provides invaluable insights into the feasibility and effectiveness of MEC integrated FiWi broadband access networks. Furthermore, the analytical results are experimentally validated through experiments. To the best of the authors' knowledge, this is the first experimental testbed for MEC in cloudlet enhanced FiWi access networks.

The remainder of the chapter is structured as follows. Section 4.2 describes the envisioned network architecture and proposed resource management scheme, whose performance is analyzed in Section 4.3. Section 4.4 presents numerical results and obtained findings. In addition, we describe our experimental testbed in Section 4.4 along with experimental results. Finally, Section 4.5 concludes the chapter.

4.2 Cloudlet Enhanced FiWi Networks for MEC

In this section, after describing the network architecture of our envisioned MEC integrated FiWi network, a novel unified resource management scheme for handling the coexistence of H2H and MEC traffic is described in great detail.

4.2.1 Network Architecture

Fig. 4.1 depicts the network architecture of cloudlet enhanced FiWi access networks for MEC. Depending on given deployment scenarios, different optical access network (OAN) technologies (e.g., EPON/10G-EPON and next-generation PON 1&2) with various network reaches could be deployed in the backhaul segment. An EPON/10G-EPON is considered a favorable backhaul technology of choice in this work. The use of EPON/10G-EPON has several benefits (e.g., cost-effective, reliable, simple, Ethernet-based, and widely deployed) over other OAN technologies. For instance, recently Starman and Nokia deployed the first nationwide 10G-EPON residential network in Europe as the most economical and technologically flexible solution considering future industry trends¹⁰. Moreover, in converged fiber-wireless networks, there has been a recently emerging trend towards Ethernet backhaul/fronthaul deployment, e.g., in cloud-radio access networks (C-RAN), where common public radio interface (CPRI) is replaced by Ethernet [52].

The optical fiber backhaul consists of an EPON/10G-EPON with a network reach of 10-20 km between the central OLT and remote ONUs. Recall that different OANs could be deployed in the backhaul segment. For instance, long-reach WDM (wavelength division multiplexing) PONs with an extended optical range of up to 100 km may be employed to enable long-reach PON solutions in

¹⁰Starman and Nokia press release: <http://company.nokia.com/en/news/press-releases/2016/06/06/starman-and-nokia-to-deploy-first-nationwide-10-gigabit-residential-network-in-europe>, June, 2016.

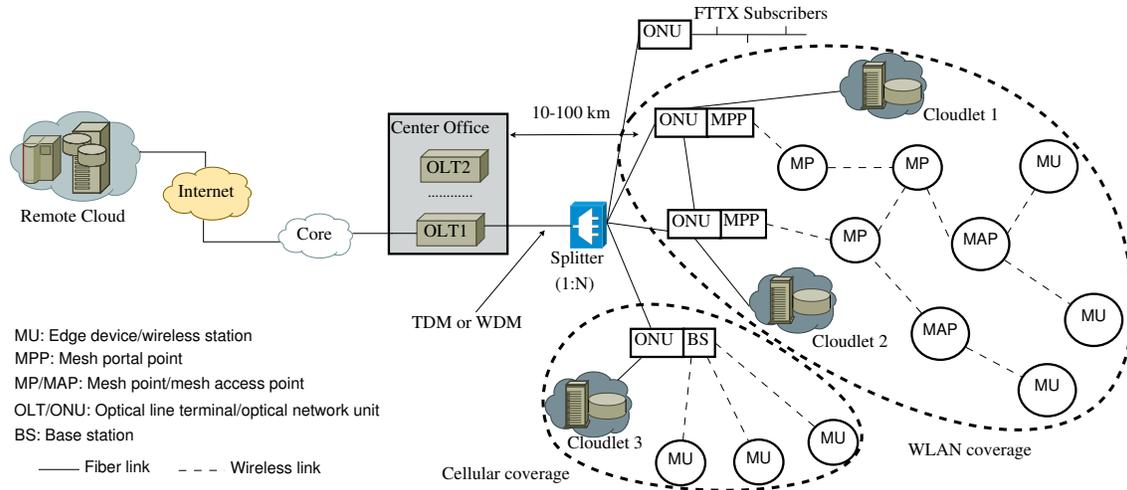


Figure 4.1: Generic network architecture of cloudlet enhanced FiWi access networks for mobile-edge computing.

the future. The OLT connects to three subsets of ONUs in a tree-and-branch topology. The first subset of ONUs are located at the premises of residential or business subscribers, providing FTTx services (e.g., fiber-to-the-home) to wired subscribers. The second subset of ONUs are equipped with a mesh portal point (MPP), giving rise to ONU-MPPs [113], to interface with the WiFi mesh network (WMN) at the wireless front-end, whereby mesh points (MPs) act as intermediate relay nodes and mesh access points (MAPs) serve wireless clients (e.g., PDAs and smart devices) within their coverage area. The integration of ONU and MPP is realized by using so-called radio-and-fiber (R&F) technologies such as cloudlet enhanced distributed radio access network (D-RAN) [181]. R&F networks are based on decentralized (optical and wireless) Ethernet technologies and perform medium access control (MAC) protocol translation at the optical-wireless interface [30].

Further, MEC servers are connected to the ONU-MPPs via dedicated reliable fiber links to provide cloudlet services at the edge of FiWi networks (see Fig. 4.1). As a result, the integrated network carries both conventional FiWi traffic and MEC traffic. In this work, one cloudlet is connected to a single ONU-MPP (see Fig. 4.1). In general, however, multiple cloudlets can be connected to a single ONU-MPP, and vice versa, depending on the capacity of cloudlets, propagation delay between STAs and cloudlets as well as the placement of ONU-APs. Interested readers may refer to [182] for cloudlet network planning and [36] for the optimal placement of ONU-MPPs. An ONU of the third subset connects to a cellular network base station (BS), which may be a conventional macrocell base station (BS) or a small cell (i.e., micro-, pico-, or femtocell) BS. The collocated ONU-BS relies on conventional centralized Radio-over-Fiber (RoF) technologies such as

cloud RAN (C-RAN) [183]. Similar to the second set of ONUs, ONU-BSs may also be connected to cloudlets (see Fig. 4.1). Given the growing interest in Ethernet over C-RAN [52] and decentralization in future 5G networks [112], R&F-based networks (i.e., cloudlet enhanced D-RAN) are likely to become the FiWi network type of choice [30].

Towards this end, from an architectural viewpoint, as shown in Fig. 4.1, there is a possibility to have not only local cloudlets but also a remote public cloud(s) in the considered network architecture. Similarly, even though this chapter does not deal with LTE/LTE-Advanced in the front-end segment, there might be a possibility of coexistence of multiple RAN technologies (e.g., WiFi and LTE/LTE-Advanced). How to handle such a multi-RAN scenario in the envisioned network architecture is a challenging and open issue. From a resource management perspective, it is challenging to design a unified resource management scheme to handle different types of coexistence such as H2H and cloud coexistence or coexistence of cloudlet, H2H, and cloud. Further, when a user moves, his/her virtual machine (VM) should be seamlessly transferred between cloudlets to preserve low round-trip latency. Since the VM mobility is sensitive to various factors such as data volume, processing speed, compression ratio, and bandwidth, it is very challenging to perform service migration in a seamless manner without degrading quality of service (QoS) [108]. Further, it was reported in [184] that classic live migration, e.g., [185], is a good choice for data centers, though it is inappropriate in the context of cloudlets because of long migration times. Further, such type of migration requires rather dynamic and efficient management of virtual networks to avoid application breakdown. All of the aforementioned challenges would require further investigation, especially in the context of MEC.

4.2.2 Unified Resource Management Scheme

General Description: Fig. 4.2 illustrates the operation of the proposed unified resource management scheme. For illustration, communications among OLT, ONU-AP, its associated STAs, and cloudlet are shown. The proposed scheme organizes the whole FiWi network operation in two layers of TDMA. The first TDMA layer is designed for the optical backhaul based on a conventional EPON/10G-EPON, where the OLT allocates bandwidth and schedules an upstream (US) timeslot for each ONU-AP, while broadcasting downstream (DS) frames to all ONU-APs. Each ONU-AP filters out received DS frames that are not destined to it based on its assigned logical link identifier

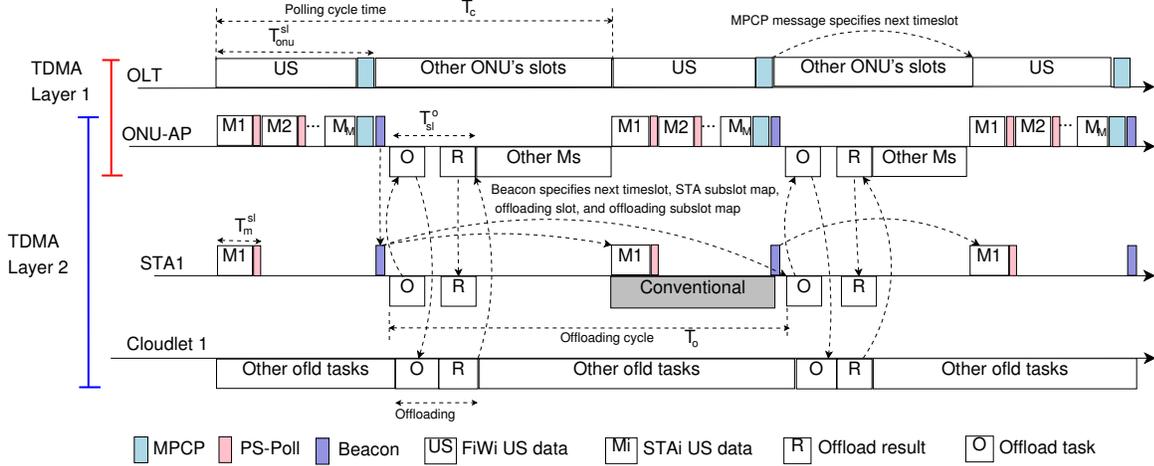


Figure 4.2: Illustration of the proposed unified resource management scheme for MEC integrated FiWi access networks.

(LLID). In the second TDMA layer, the ONU-AP allocates bandwidth in subslots and schedules transmissions for both broadband H2H and offloaded MEC traffic of its associated STAs. During its assigned timeslots, the ONU-AP receives DS/US data frames from the OLT/STAs and broadcasts/sends them to the STAs/OLT. The proposed resource management scheme synchronizes both DS and US transmissions in the same transmission slot, thus allocates bandwidth for the two transmissions rather than only for US one as in the conventional DBA of TDM PONs. In this way, the ONU does not need to listen to DS data streams outside its US timeslot. In doing so, the idleness outside timeslots can be exploited for scheduling the offloading transmissions, similar to sleep mode implementation, as proposed in [124].

The transmissions of MEC traffic of STAs are scheduled outside the FiWi timeslot of their associated ONU-AP within a polling cycle time to allow for efficient coexistence (see Fig. 4.2). The STA sends its US FiWi and offloaded traffic to the ONU-AP within its assigned subslots. Once the ONU-AP receives offloaded traffic from its associated STAs, it transmits the traffic to the cloudlet using a dedicated point-to-point fiber communication link (see Fig. 4.1). In the opposite direction, once it receives the offloading results from the cloudlet, the ONU-AP broadcasts them to all its associated STAs. Note that the layered TDM scheduling employed in the proposed resource management scheme limits the data rate per edge device, i.e., for traditional H2H traffic. In fact, with timeslot allocation, when a high data rate of STAs is required or the number of involved devices is large, the devices need large buffers to store their generated traffic. Importantly, note, however, that delay-sensitive and compute-intensive applications (e.g., face detection) are offloaded onto a

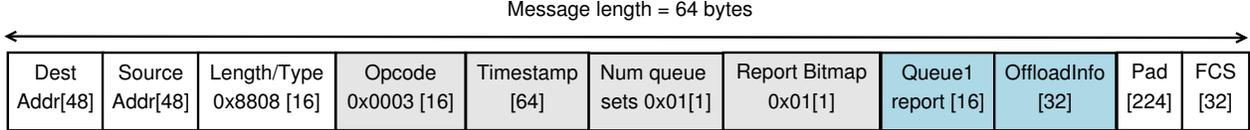


Figure 4.3: Format of extended REPORT message.

cloudlet during the non-H2H timeslot of the edge devices. Thus, the buffering burden of edge devices can be significantly alleviated. It is also worth noting that synchronizing transmissions of both optical backhaul and wireless front-end in the system can be facilitated by the two-layered TDMA resource management. If ONU-APs are not synchronized, wireless traffic could arrive at APs and keeps waiting there until the collocated ONUs have uplink timeslot assigned by the OLT, resulting in an increased end-to-end delay performance. More importantly, the scheduling of offloaded traffic transmissions become more complicated without synchronization.

As for the signaling mechanism used to facilitate our proposed scheme, the OLT allocates and schedules timeslots for ONU-APs via IEEE 802.3ah multipoint control protocol (MPCP) messages. In compliance with the standard, the OLT assigns a timeslot and notifies the ONU-AP via a conventional GATE message containing the transmission start time and duration (window size) of the timeslot in the next cycle. In this work, the REPORT message is extended by using its pad/reserved bits to include additional information needed for the OLT to schedule the next timeslot of an ONU-AP (see Fig. 4.3). The extended REPORT message contains a 32-bit field called *OffloadInfo* that indicates the most recent time instant, up to which the ONU-AP has scheduled offloaded traffic (T_{scd}^o). T_{scd}^o is used by the OLT to ensure no overlap between conventional data transmissions and offload transmissions (transmissions of offloaded traffic). Each REPORT message contains a US bandwidth request in terms of buffer backlog expressed in time units and T_{scd}^o . The OLT implements the DBA algorithm, as described in greater detail in *Algorithm 4.1* shortly.

An ONU-AP allocates US subslots to its associated STAs/edge devices by means of Beacon and PS-Po11 frames, as specified in IEEE 802.11 WLAN standards. The Beacon frame is extended by using its optional bits to include a number of slot-related parameters, i.e., start and duration. Specifically, a broadcast Beacon frame contains the ONU-AP's timeslot and an US subslot map, i.e., subslot start and duration of all STAs. Each STA sends its required bandwidth for conventional FiWi traffic to the ONU-AP by using an extended PS-Po11 frame. The extended PS-Po11 frame contains an offload flag (*ofldFlag*) to notify the ONU-AP about its offload request. The operations

Table 4.1: Notations, definitions, and default values

Notation	Description	Default value/Unit
$C_{pon}, C_{cloudlet}$	Uplink and downlink data rate in PON, cloudlet	1 Gbps, 1 Gbps
C_{wl}	WLAN (IEEE 802.11ac) data rate	6900 Mbit/s
N	Number of ONUs in system	8 – 512
M	Number of wireless STAs	15 – 250
$olt_clk,$ $onu_clk,$ sta_clk	OLT, ONU-AP, STA clock	s
$T_{prop1}, T_{prop2}, T_{prop3}$	Fiber propagation delay between ONU-AP and OLT, Air propagation delay between STA and ONU-AP, Fiber propagation delay between ONU-AP and cloudlet	0.1ms, 0.33 μ s, 50 μ s
T_c	Polling cycle time	ms (variable)
T_g	Guard time between two consecutive transmission slots	46 μ s
$T_{pon}^{msg}, T_{wl}^{msg}$	MPCP message time (GATE, REPORT), WLAN message time (e.g., PS-Poll)	0.512 μ s, 0.512 μ s
B_u, B_u^o	ONU buffer backlog, offload buffer backlog	s
onu_start, T_{sl}	Start and duration of an ONU timeslot	s
T_{start}^u, T_{sl}^u	Start and duration of an unoffloading (conventional data) STA subslot	s
T_{start}^o, T_{sl}^o	Start and duration of an offload subslot map	s
T_{start}^r, T_{sl}^r	Start and duration of an offload result subslot	s
T_{oh}^{sta}	STA sleep-to-active overhead time	250 μ s
P_a^{tx}, P_a^{rx}	STA Tx, Rx power consumption in active state	1.3 W [186]
P_a^{sta}, P_s^{sta}	STA total power consumption in active, sleep state	0.9 W, 0.3 W [186]
$S^{local}, S^{cloudlet}$	CPU clock speed (cycles/second) of STA, cloudlet	$4 \times 10^8, 3.2 \times 10^9$
μ_o	Mean service rate per server	5
λ_o	Arrival rate	2
c	Number of servers in each cloudlet	10

of ONU-AP and STA will be further detailed in *Algorithm 4.2* and *Algorithm 4.3* in Subsection 4.2.3, respectively.

For network synchronization, ONU-APs and STAs assign their local clocks to the OLT global clock, as specified in the timestamp mechanism of the EPON/10G-EPON standard. More specifically, when an ONU-AP receives an MPCP message from the OLT, it resets its local counter by using the timestamp value in the received message. A similar operation applies when the STA receives a Beacon that has the ONU-AP clock time embedded. On the other hand, when the OLT receives a REPORT message from the ONU-AP, it uses the received timestamp value to calculate

the round-trip time between OLT and ONU-AP. Interested readers may refer to [187] for further details on these subjects.

To prolong the battery life of STAs, a sleep mode is employed using a similar approach as in [76]. For simplicity, the sleep mode operation is not shown in Fig. 4.2. However, unlike [76], in this work, an STA uses its idle time in a cycle (outside its subslot) for computation offloading as the primary purpose and then enters sleep mode only if it is still idle after offloading. The STA sleep mode operation will be further detailed in Subsection 4.3.5. Table 4.1 summarizes the parameters and default values used henceforth.

In addition to protocol and message format, the integrated ONU-AP also implies extra computational complexity for the ONU. Interested readers are referred to [40] for the functional block diagram of a similar ONU-AP architecture as well as details about the implementation complexity. However, it should be noted that different from [40], which focused on energy efficiency with sleep mode scheduling, the proposed ONU-AP architecture has different functionalities. It deals with the coexistence of H2H and MEC traffic.

4.2.3 Cloudlet-Aware Dynamic Bandwidth Allocation Algorithms

There exists a plethora of DBA schemes proposed for EPON and EPON-based converged optical-wireless networks, e.g., [188–190]. However, existing DBA algorithms can not be applied directly to schedule both conventional FiWi broadband and offload transmissions in cloudlet enhanced FiWi access networks at the same time. This subsection presents our cloudlet-aware DBA (C-DBA) algorithm, a novel DBA algorithm that unifies EPON DBA and cloudlet resource scheduling to effectively handle the coexistence of both H2H and MEC traffic. C-DBA consists of three algorithms executed by the OLT, ONU-APs, and STAs, as explained in the following:

C-DBA Algorithm Executed by the OLT

The C-DBA algorithm executed by the OLT is shown in Algorithm 1. The OLT assigns a transmission timeslot to a given ONU-AP and generates a `GATE` to notify it. The timeslot duration is determined based on the request reported by the ONU-AP through a `REPORT`. The operation of Algorithm 4.1 is detailed in the following.

Algorithm 4.1 C-DBA Algorithm Executed by the OLT

```
1: if (REPORT_RCVD = TRUE) then  
2:    $B_u \leftarrow$  extracted US buffer backlog and most recently scheduled offload slot  $T_{scd}^o$  from the  
   REPORT  
3:    $B_{req} = B_u + T_{pon}^{msg}$   
4:    $T_{sl} = \min \{B_{req}, T_{sl}^{max}\}$   
5:    $onu\_start = \max \{(T_{scd}^u + T_g + 1), T_{scd}^o\}$   
6:    $T_{scd}^u = onu\_start + T_{sl}$   
7:   Generate a GATE message with  $onu\_start$  and  $T_{sl}$   
8:   Timestamp the GATE with  $olt\_clk$  and send it to the ONU-AP  
9: else if ( $onu\_start \leq olt\_clk \leq onu\_start + T_{sl} - T_{pon}^{msg}$ ) then  
10:  Receive US data frames from the ONU-AP  
11: end if
```

When the OLT receives an extended REPORT message from an ONU-AP, i.e., *REPORT_RCVD* = *TRUE* (line 1), it extracts the US buffer backlog and most recently scheduled time instant for offloading T_{scd}^o . It computes the transmission start time (onu_start) and transmission duration (T_{sl}) of its next timeslot. The bandwidth request (B_{req}) is needed for US data transmission, i.e., B_u , and signaling message exchange, i.e., T_{pon}^{msg} (see Fig. 4.2). Thus, the OLT calculates $B_{req} = B_u + T_{pon}^{msg}$, as shown in line 3. To avoid channel overload, the timeslot T_{sl} is upper bounded by a predefined maximum allowable T_{sl}^{max} (line 4). The start of the timeslot (onu_start) is the earliest time instant at which the channel is free and must be greater than the most recently offload time scheduled by STAs T_{scd}^o (line 5). The OLT updates its timeslot pointer for conventional data (line 6). In lines 7-8, the OLT then generates a conventional GATE message with onu_start and T_{sl} and timestamps the GATE with olt_clk . Next, the OLT sends a GATE message to the ONU-AP. If its clock is in the data interval of the current timeslot, the OLT receives US data frames from the ONU-AP (lines 9-10). After the data interval, the OLT expects to receive a new REPORT message and the above steps are repeated. Note that the OLT operation of the proposed scheme is different from conventional OLT operations in that it is aware of the offloading activity (lines 4-5).

C-DBA Algorithm Executed by an ONU-AP

Different from RoF-based C-RAN, in the envisioned MEC integrated FiWi network, an ONU-AP is an intelligent unit and is fully responsible for making offloading decisions and communicating with cloudlets and STAs to exchange offloaded traffic and computation results. Note that the IEEE 802.3ah standard for EPON does not specify any DBA algorithm or how it must be implemented.

Algorithm 4.2 C-DBA Algorithm Executed by an ONU-AP

```
1: if ( $GATE\_RCVD = TRUE$ ) then
2:   Extract  $olt\_clk$ ,  $onu\_start$ ,  $T_{sl}$  from the GATE message
3:    $onu\_clk = olt\_clk$ 
4:    $report\_gen = onu\_start + T_{sl} - T_{pon}^{msg}$ 
5:    $T_{start}^u[1] = onu\_start$ 
6:    $T_{sl}^u[1] = B_u^u[1]$ 
7:    $T_{start}^o[1] = onu\_start + T_{sl} + 1$ 
8:    $T_{sl}^o[1] = B_u^o[1]$ 
9:   for ( $j = 2$  to  $M$ ) do
10:    if ( $ofld\_req[j] = TRUE$ ) then
11:       $T_{start}^u[j] = T_{start}^u[j - 1] + T_{sl}^u[j - 1] + 1$ 
12:       $T_{sl}^u[j] = B_u^u[j]$ 
13:       $T_{start}^o[j] = T_{start}^o[j - 1] + T_{sl}^o[j - 1] + 1$ 
14:       $T_{sl}^o[j] = B_u^o[j]$ 
15:       $T_{scd}^o[j] = T_{start}^o[j] + T_{sl}^o[j]$ 
16:    end if
17:  end for
18:  Generate a Beacon frame with  $onu\_clk$ , conventional data subslot map  $onu\_start, T_{sl}, T_{start}^r[j], T_{sl}^r[j], \{(T_{start}^u[j], T_{sl}^u[j])\}$  and offload subslot map  $\{(T_{start}^o[j], T_{sl}^o[j])\}$ 
19:  Timestamp the Beacon with  $onu\_clk$  and broadcast to all the STAs
20: else if ( $onu\_start \leq onu\_clk < report\_gen$ ) then
21:    $B_u = 0$ 
22:   if ( $PSPoll\_RCVD[j] = TRUE$ ) then
23:    Extract  $B_u^u[j]$ 
24:     $B_u = B_u + B_u^u[j] + T_{wl}^{msg}$ 
25:    Extract  $ofldFlag$  from the PSPoll of  $STA_j$ 
26:     $ofld\_req[j] \leftarrow ofldFlag$ 
27:    if ( $ofld\_req[j] = 1$ ) then
28:     Extract  $B_u^o[j]$ 
29:    end if
30:   else
31:    Receive US data frames from each STA in subslot and relay to the OLT
32:   end if
33: else if ( $T_{start}^o[j] \leq onu\_clk \leq T_{start}^o[j] + T_{sl}^o$ ) then
34:   Send offload task of  $STA_j$  to cloudlet
35: else if ( $C - MSG\_RCVD[j] = TRUE$ ) then
36:   Extract  $T_{start}^r[j], T_{sl}^r[j]$  from the  $C - MSG$ 
37: else if ( $T_{start}^r[j] \leq onu\_clk \leq T_{start}^r[j] + T_{sl}^r[j]$ ) then
38:   Receive offload results from cloudlet and broadcast immediately
39: else if ( $onu\_clk = report\_gen$ ) then
40:   Generate and send a REPORT message with  $B_u, T_{scd}^o$  to OLT
41: end if
```

Scheduling may be implemented centrally at the OLT only or may be decentralized involving the ONUs. More specifically, while inter-ONU scheduling is implemented at the OLT, intra-ONU scheduling may be implemented either at the OLT or ONU. Thus, the proposed resource management scheme does not violate the EPON standard but rather fully exploits its flexibility. A given ONU-AP is notified of its next timeslot and cycle time via a **GATE**. It computes both FiWi and offload subslots for its STAs after receiving **PS-Poll** from them. Afterwards, the ONU-AP generates a **Beacon** frame and broadcasts it to all its associated STAs to inform their data and offload subslot map. During a data time interval, it receives US/DS FiWi data from the STA/OLT and relays it to the OLT/STA. The ONU-AP receives offload data from the STA and relays it to the cloudlet during its offloading data interval. At the end of its timeslot, the ONU-AP generates and sends a **REPORT** to request bandwidth for the next cycle. The C-DBA algorithm executed by the ONU-AP is shown in Algorithm 4.2, which is further detailed in the following.

When the ONU-AP receives a **GATE** message from the OLT (*GATE_RCVD*) (line 1), it extracts *olt_clk*, *onu_start*, and T_{sl} (see line 2). The ONU-AP synchronizes its clock to the OLT clock (line 3) and then determines the time when it will need to send a **REPORT** message in the next polling cycle time, as shown in line 4. For STA₁, the start of conventional/offload subslot T_{start}^u/T_{start}^o and bandwidth $B_u^u[1]/B_u^o[1]$ are calculated, as specified in lines 5-8. For the remaining STAs, i.e., STA 2 to M , when an offload request is received from each STA (*ofld_req = TRUE*), the ONU-AP calculates the transmission start time of conventional subslot (T_{start}^u) for FiWi traffic based on the earlier start time $T_{start}^u[j - 1]$ and the scheduled transmission duration of the conventional subslot $T_{sl}^u[j - 1]$ of each STA. Next, the ONU-AP allocates bandwidth B_u^u to each STA (lines 11-12). Similarly, the ONU-AP calculates the transmission start time of an offload subslot T_{start}^o and offload bandwidth B_u^o for offloaded traffic based on the earlier transmission start time and scheduled transmission duration, as specified in lines 13-14. The ONU-AP then updates the most recent scheduled offload time based on T_{start}^o and T_{sl}^o (line 15).

The ONU-AP generates a **Beacon** frame, which contains *onu_clk*, *onu_start*, T_{sl} , conventional data subslot map $\{(T_{start}^u[j], T_{sl}^u[j])\}$, offload subslot map $\{(T_{start}^o[j], T_{sl}^o[j])\}$, and offload result subslot map $\{(T_{start}^r[j], T_{sl}^r[j])\}$. Note that T_{start}^r and T_{sl}^r are computed by the cloudlet and sent to the ONU-AP by means of control messages, as described shortly. It then timestamps the **Beacon** frame with *onu_clk* and broadcasts it to all STAs (see lines 18-19). During the conventional data time interval (conventional data), the ONU-AP receives US data frames from each STA in the subslot

Algorithm 4.3 C-DBA Algorithm Executed by the STA

```
1: if (BEACON_RCVD = TRUE) then
2:   Extract onu_clk, onu_start,  $T_{sl}$ ,  $T_{start}^u$ ,  $T_{sl}^u$ ,
    $T_{start}^o$ ,  $T_{sl}^o$ ,  $T_{start}^r$ ,  $T_{sl}^r$ 
3:   sta_clk = onu_clk
4:   pspoll_gen =  $T_{start}^u + T_{sl}^u - T_{wl}^{msg}$ 
5: else if (onu_start ≤ sta_clk < onu_start +  $T_{sl}^u - T_{wl}^{msg}$ ) then
6:   Receive broadcast DS conventional data
   frames and filter own frame
7:   if ( $T_{start}^u ≤ sta\_clk < T_{start}^u + T_{sl}^u - T_{wl}^{msg}$ ) then
8:     Transmit US data frames to ONU-AP
9:   end if
10: else if ( $T_{start}^o ≤ sta\_clk < T_{start}^o + T_{sl}^o$ ) then
11:   Transmit offload task to ONU-AP
12: else if ( $T_{start}^r ≤ sta\_clk < T_{start}^r + T_{sl}^r$ ) then
13:   Receive offload result and filter own frame
14: else if (sta_clk = pspoll_gen) then
15:    $B_u^u$  is assigned to US buffer backlog of STA
16:   ofldFlag = (OBB > 0)? 1 : 0
17:    $B_u^o ← OBB$ 
18:   Send PS-Poll message with  $B_u^u$  and  $B_u^o$  to
   ONU-AP for reporting US bandwidth request
19: end if
```

and relays them to the OLT (line 20). If the ONU-AP receives a PS-Poll frame from the STA, i.e., *PSPoll_RCVD* (line 21), it extracts the STA bandwidth request B_u^u and updates the total bandwidth request, including T_{wl}^{msg} (lines 22-23). Meanwhile, the ONU-AP extracts the offload flag (*ofldFlag*) from the PS-Poll frame of the STA_{*j*} (line 24). If *ofldFlag* > 0, the ONU-AP extracts the bandwidth request for offloading $B_u^o[j]$ of STA_{*j*} (see lines 25-27).

During the offload subslot window, the ONU-AP transmits the offloaded task of STA_{*j*} to the cloudlet for execution and returning results. Thereafter, the cloudlet executes the offloaded tasks and the ONU-AP waits for receiving the results from the cloudlet (see Fig. 4.2). When the ONU-AP receives the result message from the cloudlet, i.e., $C - MSG_RCVD[j] = TRUE$, it extracts $T_{start}^r[j]$ and $T_{sl}^r[j]$ (see lines 34-35). If the ONU-AP clock falls into the offload result subslot, it receives the result from the cloudlet and sends it to STA_{*j*} immediately (see lines 36-37). At the end of the conventional data interval, i.e., *onu_clk* reaches *report_gen* (lines 38-39), the ONU-AP sends a REPORT message, containing B_u and T_{scd}^o , to the OLT for the next polling cycle time.

C-DBA Algorithm Executed by the STA

The C-DBA algorithm executed by the STA is shown in Algorithm 4.3. When the STA receives a `Beacon` frame from the ONU-AP, i.e., $BEACON_RCVD = TRUE$ (line 1), it extracts the information embedded in the `Beacon` frame (line 2). The STA then synchronizes its clock to the ONU clock (line 3). The time to generate a `PS-Po11` frame ($pspoll_gen$) is specified in line 4. During the ONU-AP timeslot, the STA receives conventional broadcast DL traffic from the ONU-AP and filters its destined frames (lines 5-6). During the conventional subslot, the STA transmits data frames to the ONU-AP (lines 5-9). Further, during its assigned offload subslot, the STA transmits offloaded traffic to the ONU-AP and receives the results from the ONU-AP (lines 12-13). At the end of a conventional data subslot (i.e., STA clock reaches $pspoll_gen$), the STA sends a `PS-Po11` frame to the ONU-AP containing B_u^u as the next bandwidth request (see lines 14-15). Next, the STA checks whether an offload request is buffered in the backlog or not. If $OBB > 0$, i.e., there exists an offload request in the backlog buffer, the STA resets $ofldFlag$ (lines 16-17). Afterwards, the STA sends the ONU-AP a `PS-Po11` frame at the end of its subslot to request bandwidth (B_u^u and B_u^o) for the next cycle.

4.3 Performance Analysis

This section presents the performance analysis of the proposed unified resource management scheme. It is assumed that an application is partitioned into fine-grained tasks, similar to [97], and the data load (operations and input data) of a computation task is fragmented into packets of length L (bits). In this work, tasks are classified into two execution strategies: tasks executed locally on the edge device/STA are termed local execution, i.e., non-offload scenario, whereas tasks offloaded onto the cloudlet for execution are called remote execution, i.e., offload scenario.

The performance metrics of interest include average packet delay of both H2H and MEC traffic, average response time efficiency, offload gain-overhead ratio, energy efficiency gain, and device battery life. The response time efficiency $\bar{\delta}$ is defined as the ratio of offload gain and response time of tasks that are executed by an STA locally. The offload gain-overhead ratio $\bar{\gamma}$ is the ratio of offload gain and offload overhead incurred by the communication protocols. The energy saving gain of an

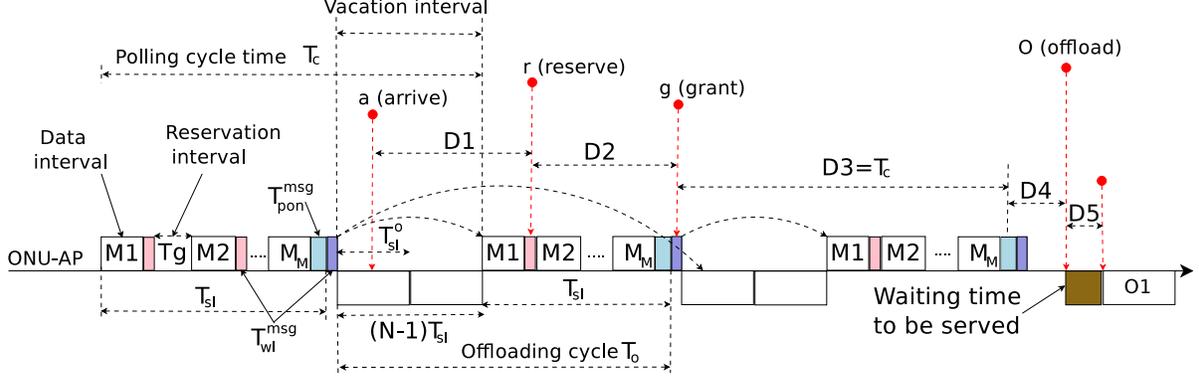


Figure 4.4: Packet delay components of both conventional H2H and MEC traffic.

STA, $\bar{\eta}$, is defined as the relative energy consumption decrease of offloading tasks onto cloudlets with respect to the local execution scenario.

4.3.1 Delay of FiWi and Offloaded Packets

Given that the proposed unified resource management scheme is upstream-oriented, this subsection analyzes the upstream delays of both FiWi and offloaded packets.

1) *FiWi Polling Cycle Time and Offload Subslot*: Let us consider an MEC integrated FiWi network with N ONU-APs, where each ONU-AP is associated with M STAs. Further, all ONU-APs are identical in terms of packet arrivals and service times. The EPON is modelled as a polling system with reservations. An ONU-AP and its associated STAs are modelled as another polling system with an M -user M/G/1 queue with reservations and vacations, where the ONU-AP goes on vacation, i.e., it does not serve FiWi traffic of STAs after its assigned timeslot (see Fig. 4.4). FiWi packets are assumed to arrive at data buffers according to a Poisson process with aggregate arrival rate λ . It is assumed that data buffers have a capacity high enough to avoid packet losses in the system. Let $\rho^{h2h} = \lambda\bar{X}$ be the aggregated H2H traffic load (intensity), $0 \leq \rho^{h2h} < 1$, where the packet service time is random with first moment \bar{X} .

As depicted in Fig. 4.4, from a FiWi transmission perspective, the ONU-AP (i.e., server) time line is decomposed into multiple cycles, each consisting of a data interval, reservation interval, and vacation interval. The vacation interval is the duration outside the timeslot plus MPCP messaging times, i.e., $(N-1)T_{sl} + T_{pon}^{msg}$. The reservation interval equals $T_{wl}^{msg} + T_g$, where T_g is the guard time that is used to avoid collision of UL transmissions. For the vacation-reservation system, within

a polling cycle time T_c , the portion of non-data H2H traffic is equal to $NM(T_{wl}^{msg} + T_g)$, which corresponds to $1 - \rho^{h2h}$. Therefore, T_c is given by:

$$T_c = \frac{NM(T_{wl}^{msg} + T_g)}{1 - \rho^{h2h}}. \quad (4.1)$$

In a polling cycle, for each edge device, there is an offload slot T_{sl}^o for both offloaded traffic and computation results. Hence, T_{sl}^o is the sum of offload packet transmission and result reception time, which is derived based on T_c and cloudlet offload traffic load ρ^{mec} as follows:

$$\rho^{mec} = \frac{M \cdot T_{sl}^o}{T_c} \Rightarrow T_{sl}^o = \frac{\rho^{mec} \cdot T_c}{M}. \quad (4.2)$$

By substituting the expression of T_c from Eq. (4.1) in Eq. (4.2), T_{sl}^o can be rewritten as follows:

$$T_{sl}^o = \left(\frac{\rho^{mec}}{1 - \rho^{h2h}} \right) [N(T_{wl}^{msg} + T_g)]. \quad (4.3)$$

Eq. (4.3) clearly shows that T_{sl}^o depends not only on offloaded traffic but also on FiWi traffic. Further, since offload transmissions are scheduled outside FiWi timeslots, ρ^{mec} must satisfy the following condition to ensure FiWi traffic is not affected by offload traffic:

$$T_{sl}^o \leq \frac{(N-1)T_{sl}}{M} \Rightarrow \rho^{mec} \leq \frac{(N-1)T_{sl}}{T_c}. \quad (4.4)$$

2) *FiWi Packet Delay Estimation*: The FiWi packet delay is analyzed under the assumption of gated service discipline. In a gated system, a reservation is made for packets that arrive before the beginning of the reservation interval [128]. It is worth mentioning that the packet queuing delay of H2H traffic at STAs is due to the time-division access to the shared channel in the network, regardless of the offloading operation outside their timeslots. The end-to-end mean packet delay (i.e., from the STA to the OLT) is obtained by adding propagation delay to the mean end-to-end packet delay for FiWi, which can be derived as in [128]:

$$\bar{D}_{fiwi} = \frac{\lambda \bar{X}^2}{2(1 - \rho^{h2h})} + \frac{(3N - \rho^{h2h})\bar{V}}{2(1 - \rho^{h2h})} + \frac{\sigma_v^2}{2\bar{V}} + T_{prop1}, \quad (4.5)$$

where σ_v^2 , \bar{X}^2 , \bar{V} denote the variance of each reservation time, second moment of each packet service time, and first moment of each reservation time, respectively, and $\rho^{h2h} < 1$. T_{prop1} denotes the fiber

propagation delay between the ONU-AP and OLT. The mean reservation time is the sum of the guard time and the time to transmit a REPORT message as follows:

$$\bar{V} = T_g + \frac{8 \times L_{pon}^{msg}}{C_{pon}}, \quad (4.6)$$

where the REPORT message size is 64 Bytes [191] and C_{pon} denotes the PON data rate (in bit/s). In Eq. (4.6), for example, with 1 Gb/s capacity of EPON, 46 μ s of T_g and 64 Bytes of REPORT, the value of \bar{V} is 46.512 μ s.

3) *Offload Delay Estimation:* As shown in Fig. 4.4, an upstream offload packet experiences the following four delay components: \bar{D}_1 is the *Arrive-to-reserve delay* defined as the time interval between an offload packet arrival and a PS-POLL request sent by a given STA for reserving its subslot; \bar{D}_2 is the time interval between the PS-POLL and a received Beacon frame, which is termed as *Reserve-to-grant delay*; \bar{D}_3 is the *Grant-to-grant delay*, which is defined as the time interval between the received Beacon frame and completion of the most recent scheduled ONU-AP timeslot, i.e., when a new Beacon frame is received (see Fig. 4.4); \bar{D}_4 is the time interval between the second Beacon frame and the actual transmission time of the packet, defined as *Grant-to-offload delay*; and \bar{D}_5 is the time interval between offload packet arrival at the cloudlet until it is served.

Without loss of generality, let us consider the delay of an offload packet arriving at ONU-AP1. First-in-first-out (FIFO) buffers are assumed to have capacity large enough to avoid packet loss. When a packet arrives at an edge device after a PS-POLL message, it must wait until the next PS-POLL to be reported. This duration is the cycle time T_c . Hence, the average waiting time of D_1 is half of T_c . D_2 is equal to the total duration of $(M - 1)$ STA subslots. Given that there are M STAs associated with an ONU-AP and $T_{sl}^u = T_{sl}/M$ is the STA subslot duration, for STA₁ D_2 is $(M - 1)T_{sl}^u$, for STA₂ D_2 is $(M - 2)T_{sl}^u$, ... and for STA _{m} D_2 is 0. Thus, on average D_2 is $(0 + 1 + 2 + M - 1) \times T_{sl}^u/M = (M - 1) \times T_{sl}^u/2 = (M - 1) \times T_{sl}/(2M)$. When M is large, it is possible to assume that $(M - 1)/M = 1$, and thus D_2 is approximately equal to $T_{sl}/2 = T_c/(2N)$. Note that T_{sl} is the duration of an ONU-AP's timeslot, which is calculated as T_c/N , whereby N denotes the number of ONU-APs in the system. After receiving a grant from the OLT, the offload packet waits until the completion of the next FiWi timeslot, which was scheduled earlier. Thus, D_3 is equal to T_c . D_4 is calculated based on the average offloading subslots of M STAs. Thus,

$\bar{D}_1, \bar{D}_2, \bar{D}_3$, and \bar{D}_4 are expressed as follows:

$$\begin{aligned} \bar{D}_1 &= \frac{T_c}{2}, & \bar{D}_2 &= \frac{T_c}{2N}, & \bar{D}_3 &= T_c, \\ \bar{D}_4 &= \frac{M \cdot T_{sl}^o}{2} = \frac{M \rho^{mec} T_c}{2 MN} = \frac{1 \rho^{mec} T_c}{2 N}. \end{aligned} \quad (4.7)$$

The cloudlet is modeled as an M/M/c queueing system, where c is the number of homogeneous servers that share a common pool of arriving tasks. The pool is simply an FCFS queue with unbounded capacity. Each server has a service rate of μ_o . The service time of a task in a server is assumed to be exponentially distributed with a mean of $1/\mu_o$. The waiting time of a packet at the cloudlet consists of a queuing time and a service time. Note that much of the multi-server literature has focused on the M/M/c queue to analyze the performance for cloud computing and cloudlet systems, e.g., [192–194]. The average waiting time can be calculated as:

$$D_5 = \frac{1}{\mu_o} + \frac{C(c, \frac{\lambda_o}{\mu_o})}{c\mu_o - \lambda_o}, \quad (4.8)$$

where the term $C(c, \frac{\lambda_o}{\mu_o})$ represents the well-known Erlang-C formula [195]:

$$C(c, \rho^{mec}) = \frac{\left(\frac{(c\rho^{mec})^c}{c!}\right)\left(\frac{1}{1-\rho^{mec}}\right)}{\sum_{k=0}^{c-1} \frac{(c\rho^{mec})^k}{k!} + \left(\frac{(c\rho^{mec})^c}{c!}\right)\left(\frac{1}{1-\rho^{mec}}\right)}. \quad (4.9)$$

By adding Eqs. (4.7) and (4.8), the overall mean offload packet delay is obtained as:

$$\bar{D}_o = \frac{T_c}{2N}(3N + 1 + \rho^{mec}) + \frac{1}{\mu_o} + \frac{C(c, \frac{\lambda_o}{\mu_o})}{c\mu_o - \lambda_o}. \quad (4.10)$$

4.3.2 Transmission Time of Offload Packet and Result

In addition to the buffering delay an offload packet experiences, as analyzed in the previous subsection, we also need to account for the transmission time of the packet itself and its offloading result. The offload transmission time $T_{tx}^{cloudlet}$ consists of the time that the offload packet L_o^{tx} (in bits) is transmitted from the STA to the ONU-AP and the time it is sent from the ONU-AP to the cloudlet via a dedicated optical fiber link. The computation result receiving time $T_{r2r}^{cloudlet}$ is defined as the sum of time intervals that the result packet L_o^{rx} (in bits) is transmitted from a cloudlet to

the ONU-AP and the packet is relayed by the ONU-AP to all STAs. The overall transmission and result receiving time are computed as follows:

$$T_{tx}^{cloudlet} = \frac{L_o^{tx}}{C_{wl}} + \frac{L_o^{tx}}{C_{cloudlet}}, \quad (4.11) \quad T_{r2r}^{cloudlet} = \frac{L_o^{rx}}{C_{cloudlet}} + \frac{L_o^{rx}}{C_{wl}}, \quad (4.12)$$

where $C_{cloudlet}$ denotes the transmission capacity of the link between the ONU-AP and cloudlet. C_{wl} is the Shannon capacity, which defines the maximum bandwidth that a wireless communication link can provide. It is given by:

$$C_{wl} = B \log_2 \left(1 + \frac{\mathcal{P}}{J} d^{-\beta} \right), \quad (4.13)$$

where the attenuation exponent β is greater than or equal to 2. \mathcal{P} , d , B , and J represent the transmitted power, Euclidean distance between sender and receiver, channel bandwidth shared by all nodes, and interference, respectively. J is assumed to be constant throughout the FiWi network.

4.3.3 Response Time Efficiency

The buffering delay \bar{D}_o , packet transmission time $T_{tx}^{cloudlet}$, and offload result receiving time $T_{r2r}^{cloudlet}$ analyzed previously are used in the following to study the performance gain of the cloudlet scenario compared with the local execution scenario as well as its efficiency.

The response time efficiency $\bar{\delta}$ is used as the metric to compare the offload and non-offload scenarios in terms of response time. The efficiency of computation offloading depends on both transmission and computation aspects of the network. The computation offloading is significant only if the time to execute a computation task at the STA locally (non-offload scenario) is much longer than the response time of offloading that task onto an MEC server (cloudlet). Their difference is defined as the offload gain. The response time $\bar{R}^{cloudlet}$ (or $\bar{R}^{nocloudlet}$) includes four components, i.e., D_o , $T_{tx}^{cloudlet}$ (or execution time of tasks onto a cloudlet $T_{comp}^{cloudlet}$), and $T_{r2r}^{cloudlet}$. The average response time efficiency $\bar{\delta}$ is given by:

$$\bar{\delta} = \frac{\bar{R}^{nocloudlet} - \bar{R}^{cloudlet}}{\bar{R}^{nocloudlet}} \times 100 \text{ [\%]}. \quad (4.14)$$

The value of $\bar{\delta}$ can be either greater than or smaller than zero. The higher the ratio, the smaller the response time compared to the local execution of the task, thus the better the offload scenario.

$\bar{R}^{cloudlet}$ and $\bar{R}^{nocloudlet}$ are computed as follows:

$$\bar{R}^{cloudlet} = \bar{D}_o + T_{tx}^{cloudlet} + T_{comp}^{cloudlet} + T_{r2r}^{cloudlet} + 2(T_{prop2} + T_{prop3}), \quad (4.15)$$

$$\bar{R}^{nocloudlet} = T_{comp}^{nocloudlet}, \quad (4.16)$$

where T_{prop2} and T_{prop3} denote the air propagation delay between STA and ONU-AP and fiber propagation delay between ONU-AP and cloudlet, respectively. The average time to execute the tasks at an STA locally $\bar{T}_{comp}^{nocloudlet}$ and the average time to execute offload the tasks at a cloudlet $\bar{T}_{comp}^{cloudlet}$ are computed as follows:

$$T_{comp}^{nocloudlet} = \frac{\phi^{local}}{S^{local}}, \quad (4.17)$$

$$T_{comp}^{cloudlet} = \frac{\phi^{cloudlet}}{S^{cloudlet}}. \quad (4.18)$$

In Eqs. (4.17-4.18), ϕ^{local} and $\phi^{cloudlet}$ denote the number of CPU cycles necessary to execute tasks locally at the STA and remotely at a cloudlet, respectively. Further, S^{local} and $S^{cloudlet}$ represent the CPU clock speed (cycles/second) of STA and cloudlet, respectively. Note that the number of CPU cycles depends on the offload packet size and the type of task to be executed.

By substituting Eqs. (4.10), (4.11), (4.12), and (4.18) into Eq. (4.15), the average offload response time $\bar{R}^{cloudlet}$ is obtained as:

$$\begin{aligned} \bar{R}^{cloudlet} = \frac{T_c}{2N} (3N + 1 + \rho^{mec}) + \frac{1}{\mu_o} + \frac{C(c, \frac{\lambda_o}{\mu_o})}{c\mu_o - \lambda_o} + \left(\frac{L_o^{tx} + L_o^{rx}}{C_{wl}} \right) + \left(\frac{L_o^{tx} + L_o^{rx}}{C_{cloudlet}} \right) + \\ 2(T_{prop2} + T_{prop3}) + \left(\frac{\phi^{cloudlet}}{S^{cloudlet}} \right). \end{aligned} \quad (4.19)$$

By substituting Eqs. (4.17) and (4.19) into Eq. (4.14), the average response time efficiency $\bar{\delta}$ is obtained.

4.3.4 Offload Gain-Overhead Ratio

This subsection analyzes the computation offload gain-overhead ratio to provide insights into how efficient the proposed offloading scheme is. Computation offloading offers offload gains at the expense of incurred communication overhead, which is composed of delay and computation time. The

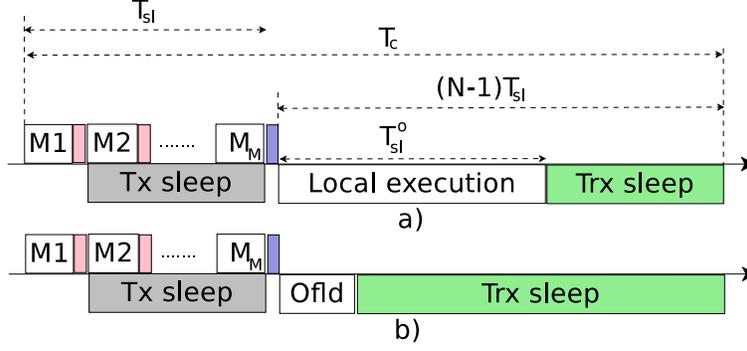


Figure 4.5: Illustration of the sleep mode at an STA: a) when tasks are executed locally at STA; b) when tasks are offloaded onto a cloudlet.

average offload gain-overhead ratio $\bar{\gamma}$ is given by:

$$\bar{\gamma} = \frac{\bar{\mathcal{R}}^{nocloudlet} - \bar{\mathcal{R}}^{cloudlet}}{\bar{D}_o + T_{tx}^{cloudlet} + T_{r2r}^{cloudlet} + 2(T_{prop2} + T_{prop3})}. \quad (4.20)$$

After substituting the values of $\bar{\mathcal{R}}^{nocloudlet}$, $\bar{\mathcal{R}}^{cloudlet}$, \bar{D}_o , $T^{cloudlet}$, and $T_{r2r}^{cloudlet}$ into Eq. (4.20), $\bar{\gamma}$ is obtained. We observe from Eq. (4.20) that $\bar{\gamma}$ is determined by not only the capability of a device, but also the scheduling solution employed in the system.

4.3.5 Energy Efficiency Gain and Battery Model

There exist several studies on energy-savings in FiWi networks, e.g., [76, 117, 118]. However, they are not applicable in cloudlet enhanced FiWi networks because ONU-APs are connected to cloudlet(s) and it is not feasible to turn off an ONU-AP. As explained earlier, STAs can be put into sleep mode when they are idle (see Fig. 4.5). The STA sleep method is employed in both local execution and cloudlet offloading scenarios, as illustrated in Fig. 4.5. The average energy saving $\bar{\eta}$ is derived for the STA within polling cycle time T_c as follows:

$$\bar{\eta} = \frac{\bar{E}^{nocloudlet} - \bar{E}^{cloudlet}}{\bar{E}^{nocloudlet}} \times 100 [\%], \quad (4.21)$$

where $\bar{\eta}$, $\bar{E}^{nocloudlet}$, and $\bar{E}^{cloudlet}$ denote the average energy efficiency gain, average energy consumption of the STA within T_c without and with cloudlet(s), respectively, as derived shortly. The comparison of $\bar{E}^{nocloudlet}$ and $\bar{E}^{cloudlet}$ helps decide whether the task should be locally executed

or offloaded onto cloudlet(s) from a battery life viewpoint. Specifically, if $\bar{E}^{nocloudlet} > \bar{E}^{cloudlet}$ offloading the task onto a cloudlet is more effective. Otherwise, it is better to execute the task at the STA locally.

It is worth noting that most previous studies on computation offloading consider energy consumption during offload data transmissions without taking any power saving mode scheduling strategies into account. In this work, a power saving mode scheduling approach is adopted, similarly to the method presented in [76]. The criterion to make sleep decisions is whether the idle time of the STA is greater than the sleep-to-active time (i.e., overhead time). Recall that the STA uses its idle time in a cycle (outside its subslot) for computation offloading as the primary purpose and enters sleep mode if it is idle after offloading activities (see Fig. 4.5). It follows:

$$T_{slp}^{sta} = \begin{cases} 0 & \text{if } [(N-1)T_{sl} - T_{comp}^{nocloudlet} + (M-1)T_{sl}^{sta} - T_{oh}^{sta}] \leq 0 \\ \text{Eq. (4.23)} & \text{otherwise} \end{cases} \quad (4.22)$$

$$T_{slp}^{sta'} = (N-1)T_{sl} - T_{comp}^{cloudlet} + (M-1)T_{sl}^{sta} - T_{oh}^{sta}, \quad (4.23)$$

where T_{slp}^{sta} and $T_{slp}^{sta'}$ represent the sleep time of the STA when the task is executed at the STA locally and when the task is offloaded onto a cloudlet, respectively.

STA Energy Consumption in Non-Offload Scenario

The total energy consumption of the STA, when tasks are executed locally at the STA (see Fig. 4.5(a)), can be divided into the following three components:

$$\bar{E}^{nocloudlet} = \bar{E}_1^{nocloudlet} + \bar{E}_2^{nocloudlet} + \bar{E}_3^{nocloudlet}, \quad (4.24)$$

where $\bar{E}_1^{nocloudlet}$, $\bar{E}_2^{nocloudlet}$, and $\bar{E}_3^{nocloudlet}$ represent the energy consumption of the STA transceiver in active, sleep, and doze (either transmitter or receiver is “OFF”) mode, respectively. For simplicity, we assume that the STA power consumption for task computation is the same as when its

transceiver is active. $\bar{E}_1^{ncloudlet}$ is computed as follows:

$$\bar{E}_1^{ncloudlet} = \begin{cases} T_c(P_a^{tx} + P_a^{rx}), & \text{if } T_{slp}^{sta} = 0 \\ (T_{comp}^{ncloudlet} + T_{sl}^{sta} + T_{oh}^{sta})(P_a^{tx} + P_a^{rx}), & \text{otherwise,} \end{cases} \quad (4.25)$$

where P_a^{tx} and P_a^{rx} denote the power consumption of the STA transmitter and receiver in active mode, respectively, whereby their default values are listed in Table 4.1. For simplicity, the wake-up overhead times of the STA transmitter and receiver are assumed to be the same.

The STA receivers and transmitters are put into sleep mode outside the subslot (see Fig. 4.5(a)). The energy consumption of the STA in sleep mode is calculated as follows:

$$\bar{E}_2^{ncloudlet} = T_{slp}^{sta} P_s^{sta} = [(N - 1)T_{sl} - T_{comp}^{ncloudlet} + (M - 1)T_{sl}^{sta} - T_{oh}^{sta}] P_s^{sta}, \quad (4.26)$$

where P_s^{sta} is the power consumption of the STA in sleep mode. We assume that the STA transmitter consumes zero power when sleeping. $E_3^{ncloudlet}$ is the power consumption of the STA receiver during $T_{sl} - T_{sl}^{sta}$ (see Fig. 4.5(a)), which is computed as follows:

$$\bar{E}_3^{ncloudlet} = \begin{cases} 0, & \text{if } T_{slp}^{sta} = 0 \\ (T_{sl} - T_{sl}^{sta})P_a^{rx}, & \text{otherwise.} \end{cases} \quad (4.27)$$

STA Energy Consumption in Offload Scenario

The energy consumption of the STA, when tasks are offloaded onto a cloudlet (see Fig. 4.5(b)), can be divided into the following three components:

$$\bar{E}^{cloudlet} = \bar{E}_1^{cloudlet} + \bar{E}_2^{cloudlet} + \bar{E}_3^{cloudlet}, \quad (4.28)$$

where $\bar{E}_1^{cloudlet}$, $\bar{E}_2^{cloudlet}$, and $\bar{E}_3^{cloudlet}$ represent the energy consumption of the STA transceiver in active, sleep, and doze (either transmitter or receiver is ‘‘OFF’’) mode, respectively. $\bar{E}_1^{cloudlet}$, $\bar{E}_2^{cloudlet}$, and $\bar{E}_3^{cloudlet}$ are computed as follows:

$$\bar{E}_1^{cloudlet} = (T_{comp}^{cloudlet} + T_{sl}^{sta} + T_{oh}^{sta})(P_a^{tx} + P_a^{rx}), \quad (4.29)$$

$$\overline{E}_2^{cloudlet} = [(N - 1)T_{sl} - T_{comp}^{cloudlet} + (M - 1)T_{sl}^{sta} - T_{oh}^{sta}]P_s^{sta}, \quad (4.30)$$

$$\overline{E}_3^{cloudlet} = (T_{sl} - T_{sl}^{sta})P_a^{rx}. \quad (4.31)$$

By substituting $E^{nocloudlet}$ from Eqs. (4.24–4.27) and $E^{cloudlet}$ from Eqs. (4.28–4.31) into Eq. (4.21), $\overline{\eta}$ is obtained.

Battery Life

The energy efficiency gain obtained from offloading computation tasks onto cloudlets translates into an extended battery life. To calculate the average battery life \overline{B}_l of the edge device (STA), a battery capacity B_c [mAh] and nominal voltage of κ V are considered. The battery life (in hours) of the STA is defined as follows:

$$\overline{B}_l = \frac{\text{Battery capacity}}{\text{Load current}} = \frac{\kappa \cdot B_c [mAh]}{\overline{P}^{sta} [mW]}, \quad (4.32)$$

where \overline{P}^{sta} is the average power consumption of the STA, when the computation task is offloaded onto a cloudlet, which can be computed based on the total energy consumption in a cycle as follows:

$$\overline{P}^{sta} = \frac{\overline{E}_1^{cloudlet} + \overline{E}_2^{cloudlet} + \overline{E}_3^{cloudlet}}{T_c}. \quad (4.33)$$

By substituting Eqs. (4.1), (4.29), (4.30), and (4.31) into Eq. (4.33), \overline{P}^{sta} is obtained. The battery life of the STA \overline{B}_l is then estimated by substituting \overline{P}^{sta} into Eq. (4.32).

4.4 Results

This section presents the obtained results and findings of the proposed unified resource management scheme. The performance evaluation is based on analytical results for a wide variety of different network configurations. In addition, analytical results are experimentally validated through an experimental testbed.

Configuration

The considered parameters and default values used in the evaluation are listed in Table 4.1, unless specified otherwise. The PON transmission capacity C_{pon} and the transmission capacity of the link between an ONU-AP and cloudlet, $C_{cloudlet}$, are set to 1 Gb/s (same for both upstream and downstream link). MPCP messages (GATE and REPORT) are of size 64 Bytes, corresponding to a transmission time T_{pon}^{msg} of 0.512 μs . The conventional FiWi traffic is generated by mobile users and is assumed to follow a Poisson process, similarly to [128]. The FiWi traffic load (intensity) ρ^{h2h} is varied in the range from 0.05 to 0.95 to investigate its impact on the overall performance. The PS-Poll frame size is set to 20 Bytes. T_{wl}^{msg} is set to 0.512 μs . \bar{X} is assumed to be equal to 5.09 μs [128]. T_c is varied, as explained in more detail shortly and \bar{V} is computed based on Eq. (4.6), as described in Section III. A. 2. Cloudlets may be assembled from the same type of off-the-self servers. Similar to the assumption made in [182], this chapter considers a cloudlet as one rack cloudlet with 3.2 GHz of CPU clock speed ($S^{cloudlet}$). The PHY line rate of WLAN is 6900 Mbit/s. The considered edge devices are HP iPAQ PDAs with a 400-MHz (S^{local}) Intel XScale PXA255 application processor, B_c of 1000 mAh Lithium-Ion and κ of 3.6 V. For computation offloading, a face detection application is considered using the Open Source Computer Vision library (OpenCV) Haar feature-based cascade classifiers [196]. OpenCV provides general-purpose object detection functionalities. Note that face detection is a compute-intensive application, one of the most widely used applications in computation offloading. The application is modeled as a Directed Acyclic Graph (DAG) with computational components, each being characterized by a method, size of method's state, energy consumption, and number of instructions to perform the computations, similar to [97]. The offloaded traffic load ρ^{mec} is varied and calculated according to Eq. (4.2). An image size of 500×426 pixels greyscale mode is converted into KB and used as input data for face detection.

Impact of FiWi Traffic on Network Performance

In the following, various evaluation scenarios are thoroughly investigated to examine the impact of H2H traffic on the polling cycle time, H2H and MEC packet delay, and cloudlet response time. Further, the impact of the polling cycle time on the offload gain-overhead ratio is discussed as well.

The polling cycle time, T_c , plays an important role and has an impact on the network performance (see Eqs. (4.1, 4.19, 4.32)). Among the parameters that determine T_c , the impact of the

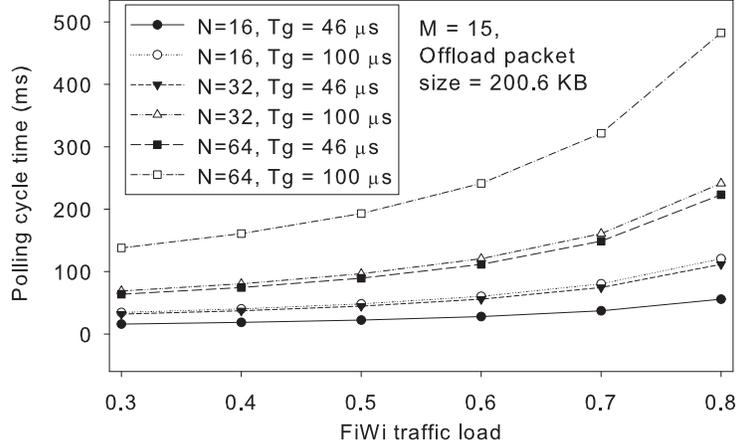


Figure 4.6: Polling cycle time T_c vs. conventional FiWi traffic load for different guard time T_g and number of ONU-APs.

conventional FiWi traffic load ρ^{h2h} on T_c is shown in Fig. 4.6. Fig. 4.6 shows that T_c increases when either ρ^{h2h} or the number of ONU-APs (N) increases. For example, when ρ^{h2h} equals 0.6, T_c is smaller than 242 ms for all values of N . However, T_c increases rapidly when ρ^{h2h} approaches 1. Importantly, as shown in Fig. 4.6, T_c can also be extended by configuring a larger value of guard time T_g between consecutive subslots.

Fig. 4.7 depicts both offload buffering delay \bar{D}_o and FiWi mean packet delay \bar{D}_{fiwi} as a function of ρ^{h2h} for different N . The delay remains rather low when ρ^{h2h} is low, but rapidly increases when ρ^{h2h} increases. Since the number of reservation intervals in a cycle increases with increasing N , the mean packet delay increases when N increases. For a given ρ^{h2h} , both \bar{D}_o and \bar{D}_{fiwi} increase significantly with N . Fig. 4.7 shows that as ρ^{h2h} increases, STAs need to wait longer for offloading activities due to a longer resultant cycle time (see also Eq. (4.10)). However, \bar{D}_o is always higher than \bar{D}_{fiwi} since offload transmissions are scheduled outside the primary FiWi timeslots. For example, for a typical scenario of 32 ONU-APs and ρ^{h2h} of 0.5, the values of \bar{D}_o and \bar{D}_{fiwi} equal 67.74 ms and 6.60 ms, respectively.

Fig. 4.7 provides insights into where and when to offload a computation task. For instance, for a typical scenario of 32 ONU-APs, and ρ^{h2h} less than or equal to 0.6, the cloudlet experiences a delay of less than 85 ms. Thus, delay-sensitive applications, whose delay tolerance is below 85 ms, can be offloaded onto MEC integrated FiWi networks. However, it may not be practical to offload the computation of tasks onto the cloudlet when ρ^{h2h} is greater than 0.8 because of longer delays. Therefore, in such a case, tasks should be locally executed at the STA.

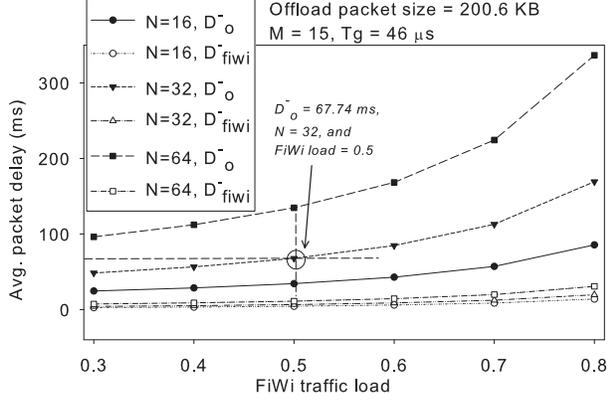


Figure 4.7: Impact of FiWi traffic load on packet delay of both FiWi and MEC traffic.

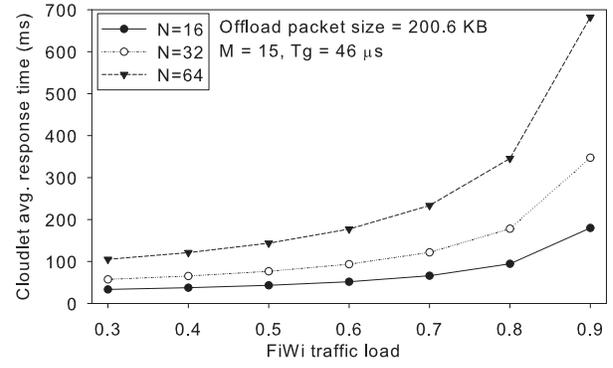


Figure 4.8: Impact of FiWi traffic load ρ^{h2h} on cloudlet response time $\bar{R}^{cloudlet}$.

Fig. 4.8 shows the cloudlet response time $\bar{R}^{cloudlet}$ as a function of ρ^{h2h} for different values of N . In addition, other parameters such as L_o^{tx} , L_o^{rx} , C_{wl} , $C_{cloudlet}$, $S^{cloudlet}$, T_{prop2} , and T_{prop3} directly impact the performance of $\bar{R}^{cloudlet}$ (see Eq. (4.19)). For instance, in the case of 32 ONU-APs, $\rho^{h2h} = 0.7$ (high FiWi load), and L_o^{tx} of 200.6 KB, $\bar{R}^{cloudlet}$ equals 121.80 ms. However, for increasing values of ρ^{h2h} , $\bar{R}^{cloudlet}$ also increases slightly. This is because STAs have to wait until the end of FiWi timeslots in the coexistent H2H/MEC system. Further, Fig. 4.8 helps identify the suitable ρ^{h2h} load for an optimal value of $\bar{R}^{cloudlet}$. For example, in a system of 32 ONU-APs, to achieve a value of $\bar{R}^{cloudlet}$ equal to 93.60 ms, ρ^{h2h} must not exceed 0.6.

Fig. 4.9 depicts the impact of the polling cycle time T_c on the average offload gain-overhead ratio $\bar{\gamma}$. As the cycle time increases, ONU-APs can serve more traffic, translating into more offload packets to be transmitted to the cloudlet. As a result, the average cloudlet response time also increases slightly (see Eq. (4.19)). Note, however, that the value of $\bar{\gamma}$ also increases gradually. For instance, for 32 ONU-APs and $T_c = 42$ ms and 182 ms, $\bar{\gamma}$ equals 263.18 and 273.60, respectively. This means that the higher the value of $\bar{\gamma}$ the more beneficial computation offloading becomes.

Impact of Offloaded Traffic on Network Performance

In the following, various evaluation scenarios are investigated to examine the impact of offloaded traffic on cloudlet response time, offload gain-overhead ratio, response time efficiency, energy efficiency, and device battery life. Further, the trade-off between response time efficiency and achievable energy efficiency is studied.

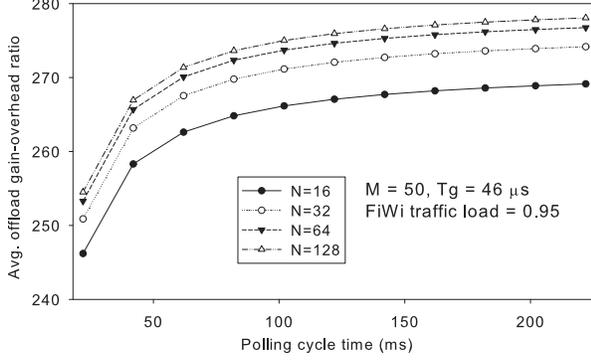


Figure 4.9: Impact of polling cycle time T_c on average offload gain-overhead ratio $\bar{\gamma}$.

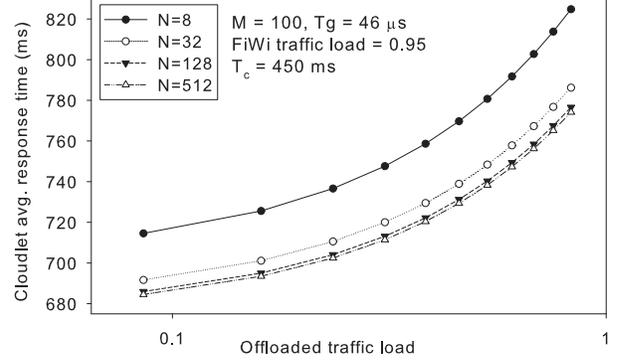


Figure 4.10: Impact of offloaded traffic load ρ^{mec} on the average response time of cloudlet $\bar{R}^{cloudlet}$ with different number of ONU-APs.

The average response time of cloudlet $\bar{R}^{cloudlet}$ at different offloaded traffic loads for different N is shown in Fig. 4.10. $\bar{R}^{cloudlet}$ depends not only on the speed of the CPU clock but also on the offloaded traffic and incurred communications delay (see Eq. (4.19)). $\bar{R}^{cloudlet}$ increases slightly under higher offloaded traffic loads. However, increasing N helps reduce the average amount of time a task spends in the cloudlet queue, thereby decreasing response time. For instance, for a typical scenario of 32 ONU-APs, ρ^{h2h} of 0.95 (heavy) and ρ^{mec} of 0.15, $\bar{R}^{cloudlet}$ and STA response time equal 7774.48 ms and 17187.50 ms, respectively. Overall, Figs. 4.8 and 4.10 show a similar trend of $\bar{R}^{cloudlet}$ for both ρ^{h2h} and ρ^{mec} .

On the other hand, the gain obtained from computation offloading comes at the expense of communication overhead. It also depends on the type of application and amount of offload packets transmitted to the cloudlet. Similar to Fig. 4.9, Fig. 4.11 shows the impact of ρ^{mec} for different N on the offload gain-overhead ratio $\bar{\gamma}$. Fig. 4.11 provides important insights into the computation offloading strategy. More specifically, the higher the value of $\bar{\gamma}$ the more beneficial offloading becomes. Offloading is more beneficial, when the system either has high-computing capabilities (powerful cloudlets) or implies minimum communication overhead, or both criteria are met. The communication overhead can be minimized by designing an efficient resource scheduling scheme. In the proposed scheme, due to the coexistence of FiWi and MEC traffic, a TDMA approach is employed for scheduling. To reduce the communication overhead, offloading activities are performed outside the FiWi timeslots. Figs. 4.9 and 4.11 show a similar trend of $\bar{\gamma}$. This proves that the offload performance does not degrade for ρ^{h2h} .

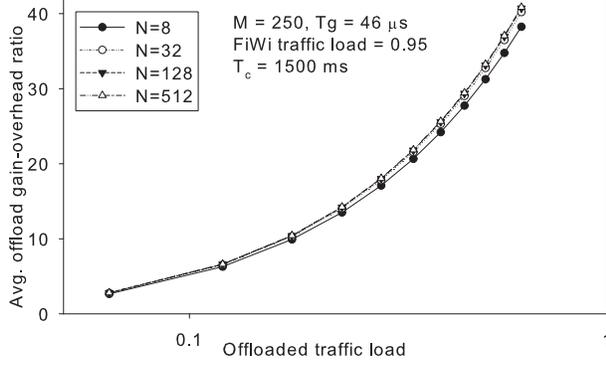


Figure 4.11: Impact of offloaded traffic load with different numbers of ONU-APs on the average offload gain-overhead ratio $\bar{\gamma}$.

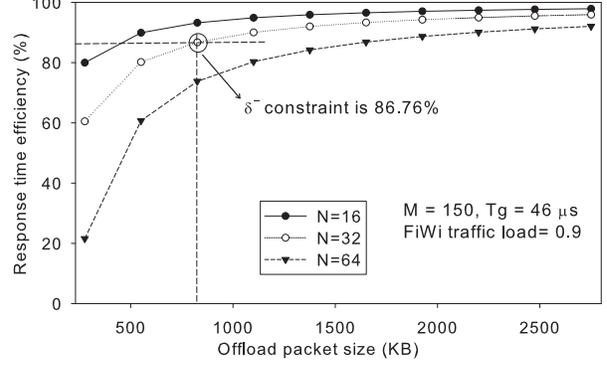


Figure 4.12: Impact of offload packet size on the average response time efficiency $\bar{\delta}$.

Fig. 4.12 illustrates the achievable response time efficiency $\bar{\delta}$. We observe that for an increasing offload packet size, $\bar{\delta}$ also rapidly increases, asymptotically approaching 100%. For instance, for a typical case of 32 ONU-APs and an offload packet size of 825.60 KB, $\bar{\delta}$ equals 86.76%. This translates into a delay reduction of 86.76% with respect to the delay obtained in a non-offload scenario (tasks are locally executed at STAs). Thus, this result indicates that MEC-based computation offloading is a promising solution for a wide range of applications (e.g., delay-sensitive and mission-critical applications) to obtain very low delay performance.

Fig. 4.13 shows the impact of offloaded traffic on the average energy efficiency $\bar{\eta}$ for a different number of ONU-APs. Remarkably, using the proposed solution, more than 78% of $\bar{\eta}$ of the STA can be achieved by offloading computation tasks onto cloudlets under light traffic loads. Fig. 4.13 also indicates that when the offload packet size increases, $\bar{\eta}$ decreases very slightly. This is because high values of ρ^{mec} require STAs to remain longer in the active state (see Fig. 4.5), thus reducing their sleeping time. Although not shown in Fig. 4.13, the energy efficiency in a non-offload scenario is much smaller than that in the MEC-based offload scenario.

It is worth highlighting the trade-off between $\bar{\delta}$ and $\bar{\eta}$ in Figs. 4.12 and 4.13. For example, for a typical system of 32 ONU-APs, ρ^{h2h} of 0.90, and offload packet size of 825.6 KB, Fig. 4.12 indicates that by using the proposed solution, 86.76% of $\bar{\delta}$ can be obtained. Meanwhile, using these values of offload packet size for the energy efficiency curve in Fig. 4.13, 89.85% of $\bar{\eta}$ of the STA is achieved for a system of 32 ONU-APs without violating given response time efficiency constraints. Note, however, that for an increasing offload packet size, $\bar{\eta}$ slightly decreases. Thus, this trade-off provides

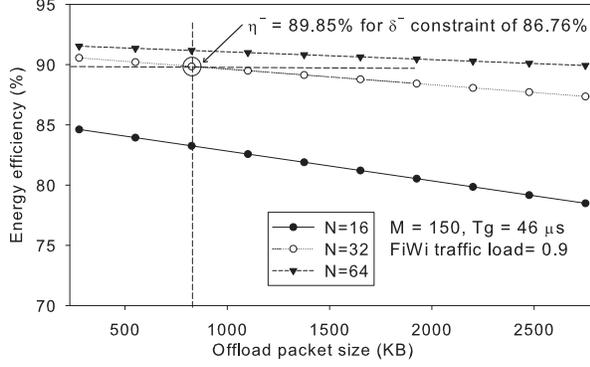


Figure 4.13: Impact of offload packet size on average energy efficiency $\bar{\eta}$.

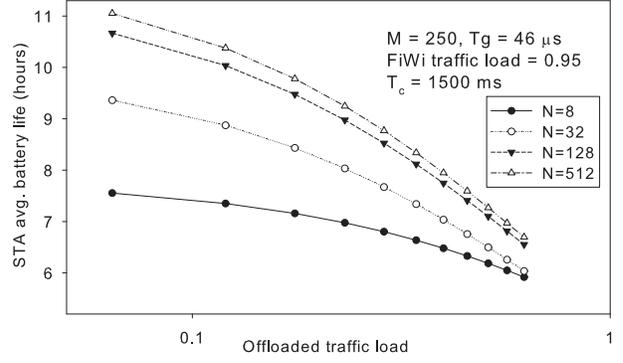


Figure 4.14: Impact of offloaded traffic load ρ_u^o on average battery life \bar{B}_l with different number of ONU-APs.

insights into how much traffic should be offloaded in order to achieve the optimal energy efficiency without violating given delay constraints.

Fig. 4.14 depicts the impact of ρ^{mec} on the average battery life \bar{B}_l for different N . The battery life of edge devices is significantly prolonged with sleep mode scheduling for small values of ρ^{mec} . For example, for a typical scenario of 32 ONU-APs and ρ^{h2h} of 0.95, \bar{B}_l is extended by up to 7.03 hours for $\rho^{mec} = 0.4$. We also observe that for large values of ρ^{mec} , \bar{B}_l tends to converge. This is because when ρ^{mec} approaches 1, \bar{D}_o becomes larger. As a result, STAs should stay longer in active mode (see also Eq. (4.10) and Fig. 4.4). These results show that our proposed scheme in the considered MEC network is a promising solution for obtaining high energy saving gains and long battery lives of edge devices. Overall, the obtained results show that the envisioned MEC enabled FiWi network is not only feasible, but also desirable for many compute-intensive and delay-critical applications.

Experimental Testbed

This section presents an experimental testbed and results obtained through measurement. The goal is to validate the accuracy of our analytical model via experimental results, as explained in greater in the following.

1) *Testbed Description:* In our experimental testbed, an EPON from Sun Telecom with four ONUs (SUN-GE8200) located at a distance of 20 km from the OLT (SUN-GE8100) in conjunction with a WLAN (with a capacity of 1 Gb/s) are used to connect the cloudlet server (Dell OptiPlex 9020

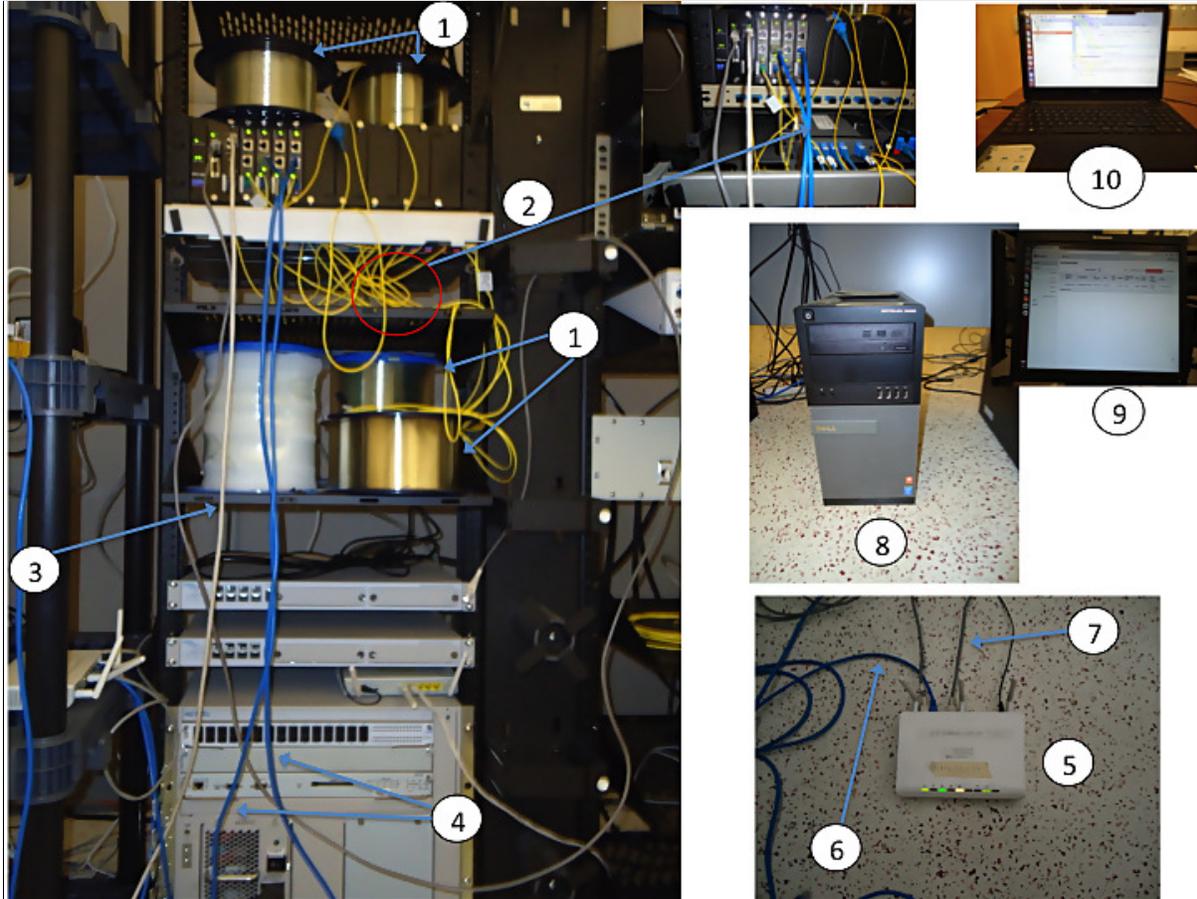


Figure 4.15: Experimental testbed setup for cloudlet enhanced FiWi networks with MEC capabilities: 1) optical fiber loops; 2) passive splitter at remote node of EPON; 3) cable connecting the gateway for core network; 4) cable connecting APs; 5) WLAN access point; 6) Ethernet cable connecting an ONU; 7) Ethernet cable connecting cloudlet; 8) cloudlet server hosting OpenStack++ platform; 9) running VM instance in OpenStack++ on the cloudlet server; 10) STA/edge device running edge application.

desktop machine) with the different client nodes. In our experiment, a FiWi network comprising an integrated ONU-AP associated with one STA (edge device) is considered.

We have used a VM running on a cloudlet server with 4 cores (VCPUs) with a clock speed of 3.6 GHz. The VM instance is allocated a 50 GB disk and 10 GB of RAM. OpenStack++ (i.e., cloudlet-enabled OpenStack)¹¹ developed at Carnegie Mellon University is deployed on the cloudlet server at the edge of the FiWi network due to the openness of this platform. Note that the OpenStack++ extends OpenStack Kilo¹², a free and open-source software platform for creating private and public clouds. In all experiments, we use a Dell Inspiron 3521 notebook as edge/mobile device, whose computing capability is comparable to that of the latest smartphones. A face detection application is

¹¹For details visit <http://elijah.cs.cmu.edu/development.html> for more information about OpenStack++.

¹²For details visit <http://docs.openstack.org/kilo/> for further information about OpenStack Kilo.

Table 4.2: System configuration of experimental testbed

	STA (edge device)	Cloudlet
Model	Dell™ Inspiron 3521	Dell™ OptiPlex 9020 Mini Tower (210-AATM)
CPU	Intel® Pentium® CPU 2127U @ 1.9 GHz, Dual Core	Intel® Core™ i7-4790 CPU @ 3.6 GHz Turbo, Quad Core (4 VCPU for VM), VT-x
RAM	4 GB DDR3L SDRAM	32 GB 1600 MHz DDR3 (10 GB VM RAM)
Disk	500 GB HDD	1 TB HDD (50 GB VM disk)
Network	802.11/b/g/n WiFi	Broadcom NetXtreme 10/100/1000 PCIe Gigabit Ethernet
OS	Ubuntu 14.04 Desktop 64 bit	Host: Ubuntu 14.04 64 bit, Guest: Ubuntu 12.04 Desktop LTS
Virtual machine manager (VMM)	–	QEMU/KVM-2.0.0
Cloud platform	–	OpenStack++
OLT	1×Sun Telecom’s SUN-GE8100 series: - one 10/100/1000Base-T RJ45 and one 1000Base-X (PON-OLT) interfaces - one network management card (one console port (RS232) and one management port (10/100Base-TX-RJ45))	
ONU	4×Sun Telecom’s SUN-GE8200 series: - one 10/100/1000Base-T, one 10/100Base-Tx, and one 1000Base-X (PON-ONU) interfaces	
AP	ZyXEL NWA570N 802.11b/g/n Ethernet wireless access point with 4 10/100 Base-TX (RJ-45) port	
Splitter	1×4 planar lightwave circuit (PLC) splitter (P/N: PLCSB-0104-X-SC)	
Fiber	20 km fiber length from the OLT to an ONU-AP	

run in an example scenario, which is widely considered an example of compute intensive applications. Note, however, that other applications such as wearable cognitive-assistance (e.g., Google Glass, Microsoft HoloLens), homecare assistance, and smart-home applications may also be delivered over this testbed. Fig. 4.15 illustrates the experimental testbed of cloudlet enhanced FiWi networks for MEC along with the deployed networking equipment. As shown in Fig. 4.15, a cloudlet server is deployed at the edge of the FiWi network. Table 4.2 summarizes the detailed specifications of the hardware used in our experimental testbed.

2) *Experimental Results:* Our conducted measurements show that the analytical results match the experimental ones reasonably well in the considered scenarios, as illustrated in Fig. 4.16. We observe from Fig. 4.16(a) that with only one edge/mobile device and cloudlet per ONU-AP, the response time efficiency is consistently higher than 87% for the various traffic loads under consideration. This is because of the small cycle stemming from the configuration parameter values. Simi-

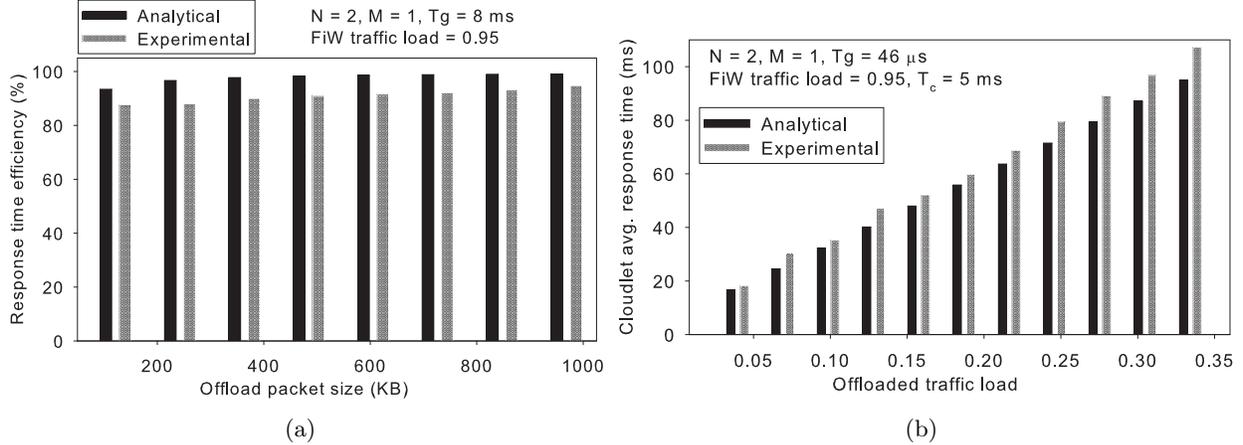


Figure 4.16: Comparison of experimental and analytical results: a) Impact of offload packet size on average response time efficiency, b) impact of offloaded traffic load on average response time of cloudlet.

larly, the experimental measurements shown in Fig. 4.16(b) validate the analytical results depicted in Fig. 4.10. More specifically, Fig. 4.16(b) shows that the cloudlet average response times obtained from our experiments are slightly higher than the respective analytical values. For instance, at an offloaded traffic load of 0.22, the average cloudlet response time equals 68.7 ms, compared to the corresponding analytical value of 63.6 ms. Further, to measure the linear association between a set of experimental and analytical values, x_i and y_i , we use the well-known Pearson product-moment correlation coefficient (PPMCC) metric, r , which is defined as follows:

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2}}, \quad (4.34)$$

where \bar{x} and \bar{y} denote the sample means of the two arrays of values. We obtain a PPMCC value of 0.998 in the results shown Fig. 4.16(b). Note that this value indicates an almost perfect linear association between the analytical and experimental results.

Next, let us validate the accuracy of the obtained analytical results by using the average error metric known as *data variation*, v , which is defined as follows:

$$v = \frac{\text{experimental value} - \text{analytical value}}{\text{analytical value}}. \quad (4.35)$$

On average, we observed approximately 10% data variation between our obtained analytical and experimental results. This is mainly due to the presence of interference and overhead of the hypervisor. We note that our measurements were taken over one hundred instances in our reported

average experimental results, whereby the relative variation between our obtained analytical and experimental results may be further reduced by averaging over a larger number of observations.

4.5 Conclusions

This chapter examined the performance of cloudlet enhanced FiWi access networks for mobile-edge computing based on the integration of EPON, WLAN, and cloudlets. A novel two-layer TDMA-based unified resource management scheme was proposed in order to handle both conventional FiWi and MEC traffic at the same time. A comprehensive analytical framework was developed to evaluate the packet delay performance of both traffic types as well as response time efficiency, offload gain-overhead ratio, energy efficiency, and battery life. An experimental testbed for cloudlet enhanced FiWi networks with MEC capabilities was developed and experiments were performed to validate the accuracy of our analytical model. The obtained results demonstrate the feasibility and effectiveness of implementing MEC in FiWi broadband access networks. More specifically, the presented solution helps reduce offload packet delay, while extending the battery life of edge devices without affecting the network performance of FiWi broadband traffic. The results provide insights into finding the optimal response time efficiency and energy efficiency for a given offload traffic load. For a typical scenario of 32 ONU-APs and an offload packet size of 825.60 KB, the delay is reduced by 86.76% compared to the delay obtained in a non-offload scenario. Further, the envisioned network yields more than 78% of energy savings for edge devices compared to non-offloading scenario and the battery life of edge devices is extended by up to 7.03 hours in typical scenarios. The introduced architecture and proposed unified resource management scheme represent a promising solution for MEC for future applications such as the so-called 5G-enabled *Tactile Internet* [44], [197].

Although EPON was considered in this work as a promising example of OAN backhaul technology, other OAN technologies may be employed in the envisioned network as well. For example, the system and solution would provide a smooth migration from IEEE 802.3ah 1G-EPON to IEEE 802.3av 10G-EPON since they share a common MAC layer. Note that the IEEE 802.3av task force focused only on the physical layer for symmetric 10 Gb/s data rates and asymmetric 10 Gb/s downstream and 1 Gb/s upstream data rates, while maintaining complete backward compatibility with 1 Gb/s EPON equipment. Therefore, the MAC protocol of 1G-EPON remains unchanged [198].

A dynamic mix of heterogeneous workloads on cloudlet such as long-running computationally intensive tasks may have an impact on the FiWi network performance, in particular their packet delay. A potential avenue for future work is to explore how the variability/heterogeneity of workload on cloudlets and different configurations of the wireless front-end would affect the performance of the proposed network architecture.

Chapter 5

Mobile-Edge Computing Empowered Fiber-Wireless Access Networks in the 5G Era

5.1 Introduction

Future fifth generation (5G) networks are expected to be characterized by massive capacity and connectivity, seamless heterogeneity, high flexibility, and adaptability. 5G will be highly integrative and convergent with a focus on increasing integration of cellular and wireless local area network (WLAN) technologies. Typical requirements of 5G networks include diverse quality of service (QoS) levels such as ultra-low latency and ultra-high reliability, reduced costs, low energy consumption, and support of different types of devices and applications [112]. However, to render the 5G vision a reality, huge challenges need to be addressed. Among those, capacity-limited backhaul links are identified as one of the key challenges, especially when considering the extreme densification and diversity of small cells. Network integration is another critical challenge that requires efficient merging and coordination of various types of networks (e.g., wired and wireless access networks). In 5G, the coexistence of different types of traffic will further diversify communication characteristics

This chapter is based on the following publication:
[108] B. P. Rimal, D. Pham Van, and M. Maier. Mobile-Edge Computing Empowered Fiber-Wireless Access Networks in the 5G Era. *IEEE Communications Magazine*, vol. 55, no. 2, pp. 192-200, Feb. 2017.

and requirements. Since both conventional human-to-human (H2H) traffic (e.g., voice, data, and video) and emerging types of traffic (e.g., machine-centric communications) coexist, it is challenging to design unified resource management schemes to support such coexistence to ensure that critical traffic is not affected by other coexistent traffic.

Unlike conventional centralized clouds, local clouds (cloudlets) are pushing the frontier of computing away from central nodes towards the network edge to enhance the availability and reachability of cloud services, while minimizing wide area network (WAN) latencies [16]. In light of this, mobile edge computing (MEC) [26] has recently emerged, which offers cloud capabilities (e.g., computing, storage, and caching) at the edge of networks in close proximity to mobile devices, thereby enriching users' broadband mobile experience. MEC transforms base stations (BSs, e.g., 3G, 4G) into intelligent service hubs that are capable of delivering highly personalized services. Further, MEC helps provide the backhaul with real-time information about radio access network (RAN) and traffic requirements, and thus facilitates coordination between the backhaul and RAN segments, which has not been fully realized so far [27]. Such coordination is required when, for example, radio networks need less bandwidth but the backhaul is not aware of it, and vice versa. From a business viewpoint, the emergence of MEC allows network operators, independent software vendors, and web service and content providers to create new value chains [26].

By offering attractive features including extremely low latency and high throughput, which 5G networks necessitate, MEC creates a pathway to 5G. For instance, a class of 5G applications called mission-critical Internet of Thing (IoT) and Tactile Internet [28], which require extremely low latency and carrier-grade reliability (99.999 percent availability), are expected to rely on MEC [26]. Therefore, MEC is recognized by the European 5G Infrastructure Public Private Partnership (5G PPP) research body as one of the key emerging technologies for 5G networks [26]. In addition, backed by industry leaders (e.g., Intel, Nokia, Huawei, and Vodafone) who participate in the European Telecommunication Standards Institute (ETSI) MEC Industry Specification Group (ISG), MEC is expected to provide a standard-based approach for significant progress towards 5G. Given the growing momentum in MEC, this chapter aims to investigate MEC in the context of 5G.

Figure 5.1 provides an overview of potential deployment scenarios of MEC together with cloudification in 5G networks, including optical backhaul (e.g., Ethernet passive optical network [EPON] and 10G-EPON) and wireless backhaul technologies (e.g., millimeter-wave, microwave, and Sub-6

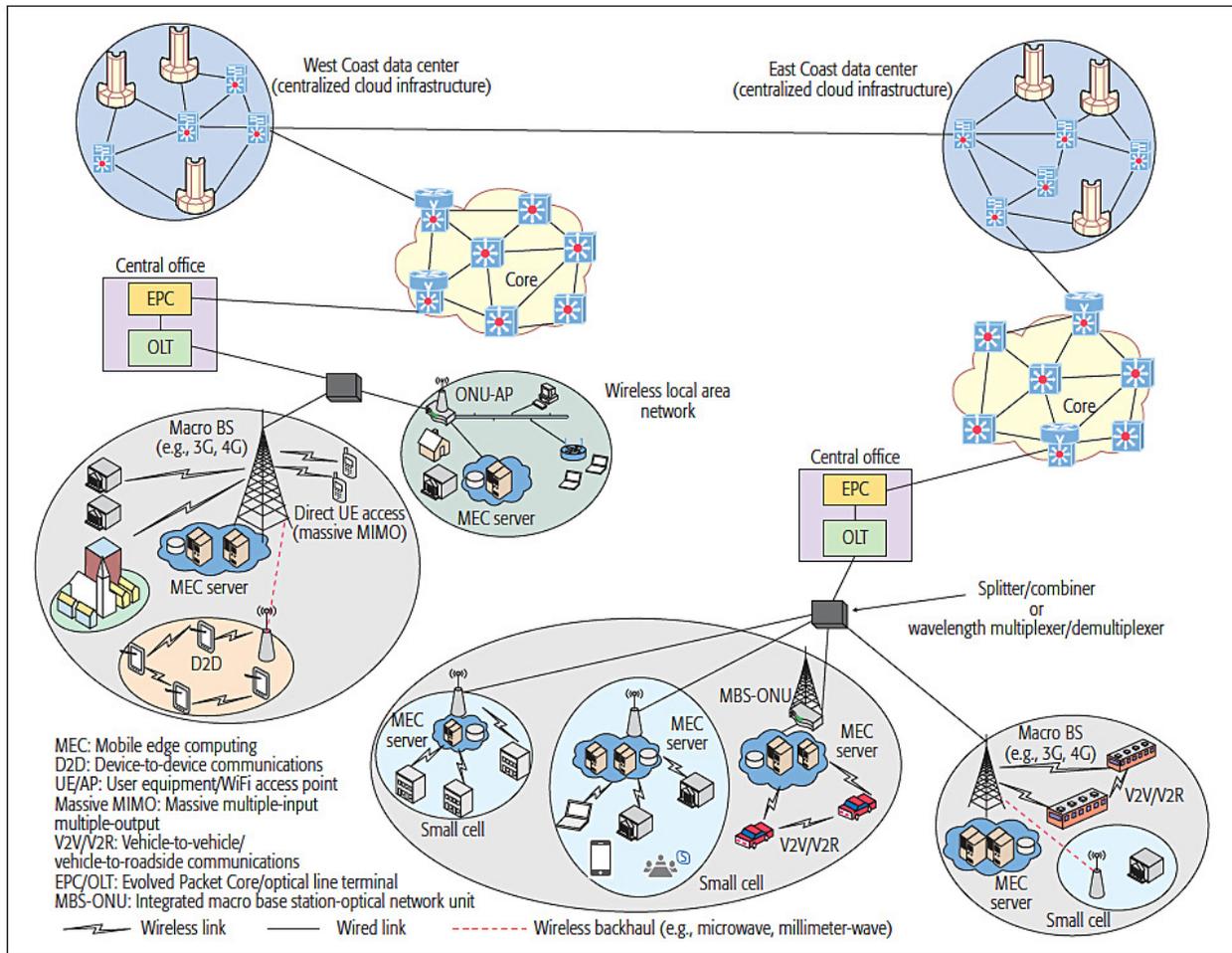


Figure 5.1: Illustration of mobile edge computing and cloudification of 5G networks.

GHz unlicensed/licensed). As shown in Fig. 5.1, MEC servers can be deployed in various scenarios such as at WiFi access points, cellular macro base stations (e.g., 3G, 4G/LTE-A), small cell aggregation points, and central offices in coexistence with conventional centralized clouds.

Ethernet technology is ubiquitously deployed due to its cost effectiveness, interoperability, and backward compatibility. For example, recently, it was advocated that Ethernet should be used to transport common public radio interface (CPRI) frames, the most commonly used standard in Cloud-RAN (C-RAN) [52]. In this regard, Ethernet-based integrated fiber-wireless (FiWi) access networks present a compelling solution for not only broadband access but also mobile backhaul by combining the reliability and capacity of Ethernet-based optical backhaul (e.g., IEEE 802.3ah EPONs) and the extended coverage and flexibility of Ethernet-based wireless front-end (e.g., IEEE 802.11 WLANs). Recently, FiWi networks were integrated with LTE-Advanced heterogeneous networks (HetNets) [29], to further support conventional cloud computing and cloudlets [30]. Because

of to their salient features, in this work, FiWi networks are further enhanced with MEC capabilities. MEC servers are integrated at the edge of FiWi networks, that is, access points or BSs collocated with optical network units (ONUs), giving rise to *MEC over FiWi networks*. Since MEC is considered one of the key emerging technologies for 5G networks [26], the introduced MEC over FiWi concept has to tackle not only the aforementioned 5G challenges but also its own and unique challenges. Among those, the integration of MEC with existing network infrastructures (both wired and wireless), coexistence between cloudlets and conventional clouds, and enhanced resource management in consideration of backhaul/RAN coordination are of primary importance.

Research in the area of MEC is still in its infancy. Even though there is growing interest in MEC from both academia and industry, to the authors' best knowledge, the MEC over FiWi design scenarios and unified resource management scheme presented in this chapter are the first that take network integration, H2H/MEC coexistence, and resource management issues into consideration. This chapter aims to provide a comprehensive inquiry into the design of MEC over FiWi networks. It first elaborates on the concept of MEC by studying typical service scenarios. Key technical challenges of implementing MEC are then identified and discussed in detail. Given that WLAN and 4G LTE/LTE-A are among the most common RAN technologies, three typical foreseeable design scenarios are considered:

1. MEC over WLAN-based FiWi
2. MEC over 4G LTE-based FiWi
3. Coexistence of MEC and C-RAN in FiWi enhanced LTE-A HetNets

To examine achievable network performance gains, for the first time, this chapter proposes a novel unified resource management scheme that jointly allocates bandwidth for transmissions of both conventional broadband traffic and MEC data in a time-division multiple access (TDMA) fashion.

The rest of the chapter is structured as follows. Section 5.2 discusses potential service scenarios of MEC and its design challenges. Section 5.3 proposes the network architecture for MEC over FiWi. Section 5.4 presents the resource management scheme and its performance evaluation for the MEC over WLAN-based FiWi network. Finally, Section 5.5 concludes the chapter.

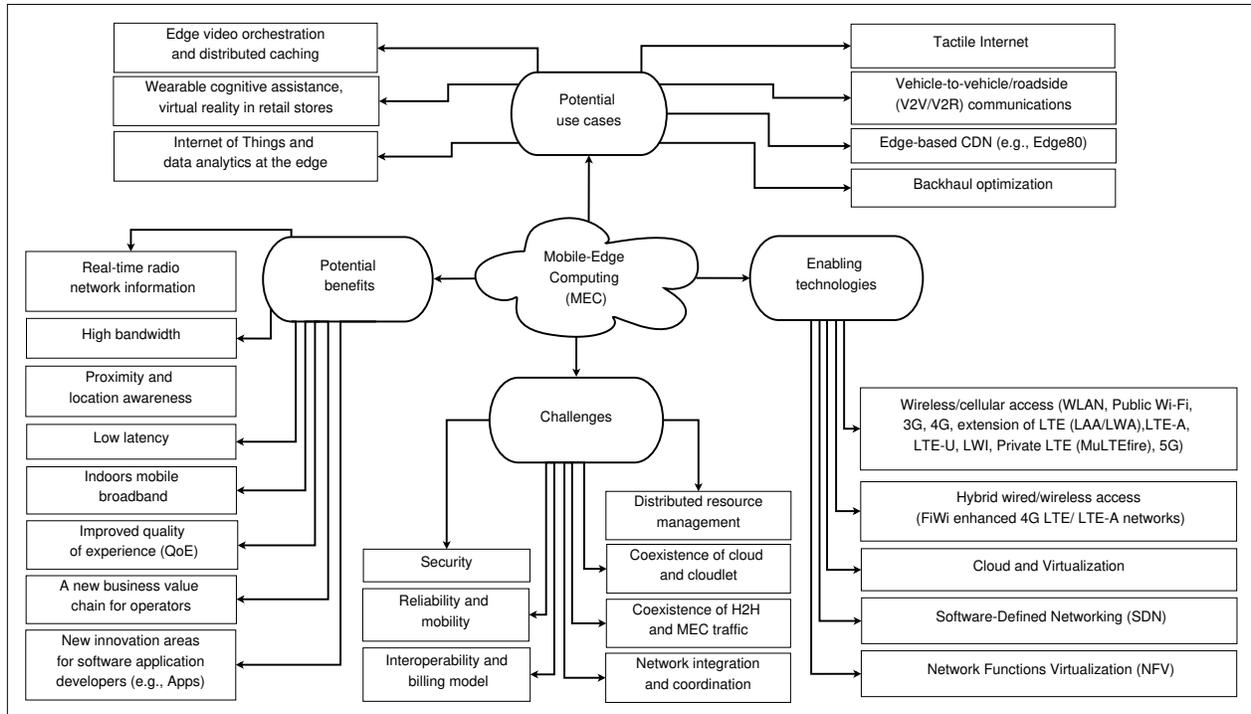


Figure 5.2: Overview of mobile edge computing: benefits, potential use cases, enabling technologies, and challenges (LWA: LTE and WiFi aggregation, LAA: licensed assisted access, LTE-U: LTE Unlicensed, LWI: LTE/WiFi interworking, CDN: content delivery network).

5.2 Mobile-Edge Computing: Typical Service Scenarios and Design Challenges

An overview of MEC, including its benefits, potential use cases, enabling technologies, and challenges, is illustrated in Fig. 5.2. Further, this section presents potential service scenarios and challenges in designing MEC-enabled networks for 5G in great detail.

5.2.1 Typical Service Scenarios

1) *Edge video orchestration and distributed caching*: Mobile edge platforms may provide edge video orchestration services, whereby visual content may be produced and consumed at a location close to subscribers in densely populated areas. Typical examples include ultra-high-definition (4K/8K) video streaming, multiple viewpoints, and live streaming events, where a large number of edge devices access the content with high required minimum bit rates. In addition, the most popular content consumed in the geographical area may be cached at an MEC server. Caching content or

delivering services directly from the network edge helps backhaul and core networks reduce traffic loads. MEC therefore reduces network delay and increase throughput, thereby enriching users' broadband experience.

2) *Backhaul optimization*: Due to the dense deployments of small cells and the increased reliance on unlicensed spectrum resources, network optimization should account for multiple RAN technologies and localized service information. MEC services such as traffic and performance monitoring help provide the backhaul segment with real-time information about RAN and traffic requirements [27]. Such information may be used for advanced backhaul optimization methods. This type of real-time optimization (e.g., rerouting traffic) allows for more efficient use of network resources, better user experience, and advanced traffic congestion management.

3) *Vehicle-to-vehicle/roadside (V2V/V2R) communications*: Connected vehicles, one typical 5G use cases, are a set of networked systems with hundreds of sensors (e.g., Google self-driving cars and high-speed trains). While processing and storing V2V/V2R data centrally may be effective in some cases, it is not practical in many other cases, where real-time information (e.g., about traffic accidents) with low latency and low jitter is required. For instance, V2R services usually have a latency requirement of 10 ms or below. MEC helps provide low-latency and location-based information in real time. This will enable a nearby car to receive data on the order of milliseconds, thus allowing the driver to immediately switch lanes, slow down, or change his/her route. For example, recently, Deutsche Telekom¹³ has deployed MEC to enable V2V communications with latencies under 20 ms, compared to the traditional 100 ms.

4) *Internet of Things (IoT) services*: The IoT ecosystem, a network of billions of physical objects or things, provides very diverse types of applications and services with a wide range of QoS requirements. It is expected that most of the network intelligence will reside closer to IoT sensors, and many IoT applications will have stringent latency, reliability, and security requirements [199]. For instance, telesurgery applications require latency below 10 ms. Further, resource constrained IoT devices may offload their data onto MEC servers, thereby extending battery life, reducing latency, and enabling applications to respond in real time. Also, MEC is expected to provide reliable inter-city communications by moving computing and processing resources closer to sensor nodes in support of capillary networks in large-scale IoT deployments (e.g., smart cities).

¹³<http://rethinkresearch.biz/articles/deutsche-telekom-demos-mec-in-the-connected-car/>, Accessed on March 2016.

5.2.2 Design Challenges

1) *Network integration and coordination:* The key challenge is to seamlessly integrate multiple network technologies (i.e., wired, wireless, cellular) in order to support MEC capabilities. Indeed, MEC servers should be compatible with underlying network architectures, interfaces, and functionalities for improved network performance [26]. Given the diverse potential deployment scenarios of MEC over multiple RANs (e.g., WLAN, LTE), network integration should be considered in the first place at both the architectural and protocol levels. Further, the coordination between backhaul and front-end segments of converged networks in 5G is another important issue. Although MEC holds promise to realize such coordination, it is challenging to incorporate the coordination and synchronization into the MEC design.

2) *Distributed resource management:* It is important to ensure that different and diverse edge devices (e.g., moving users, mobile devices, and connected vehicles) have access to network resources (e.g., bandwidth, storage) at the edge [178]. Since the design complexity increases when a shared but limited amount of resources must be allocated to accommodate dynamic needs of such devices at the edge, designing a resource management scheme with QoS guarantees for MEC networks is more challenging than in conventional networks.

3) *Coexistence of H2H and MEC traffic:* The coexistence of H2H and MEC traffic further diversifies communication characteristics and requirements. The integration of MEC and existing access networks will need to cope with the coexistence of broadband access and MEC traffic. While both H2H and MEC support voice calls, video streaming, web surfing, and social networking, MEC applications have their own and diverse QoS requirements such as real-time response, location and mobility awareness. Importantly, MEC also includes control communications (e.g., emerging Tactile Internet applications). Such applications have very stringent end-to-end latency requirements of about 1 ms to avoid perceivable lags to remotely control actuators (e.g., robots) [28].

4) *Cloud and cloudlet coexistence:* Centralized clouds and distributed cloudlets may coexist and be complementary to each other, and thus support a more diverse set of emerging applications and services in 5G networks. However, determining where an application is executed, at either a cloudlet or a conventional cloud, is a nontrivial task. It depends on the available infrastructure and application requirements, as well as willingness of users to pay. Some applications or parts of an application may be executed at the edge device itself, cloudlets, or centralized clouds. A fundamental

question is how to identify which part of the application to offload onto clouds/cloudlets and which not, given that partitioning applications for offloading increases the complexity and overhead of the MEC design. Further investigation is required to find smart strategies for coexistent cloud and cloudlet systems under realistic network conditions. On the other hand, from a business model point of view, given that clouds and cloudlets may be owned by different operators, interoperability and billing issues may arise. It is challenging to coordinate with individual cloud service providers, each having their own interfaces. To this end, a common deployment and management platform for a multi-cloud environment is desirable to optimize network performance and minimize costs.

5) Reliability and mobility: Given that mission-critical applications (e.g., telecontrol of heavy machinery) may rely on MEC, meeting the required level of reliability and resilience of such applications is challenging. For instance, providing carrier-grade reliability (99.999 percent availability) at the network edge will create a new dimension of complexity for designing robust and highly optimized protocols. Further, MEC should provide service continuity, application and virtual machines (VMs) mobility, and application-specific user-related information. In general, mobility management (e.g., behavior of mobile users' mobility and its predictability [178] and handover optimization) is nontrivial. When a user moves, his/her VM should be seamlessly transferred between MEC servers. Since VM mobility is sensitive to various factors such as data volume, processing speed, compression ratio, and bandwidth, it is challenging to do such migration of VMs in a seamless manner, without degrading the quality of experience (QoE).

5.3 MEC over FiWi Network Architecture

This section presents different design scenarios of MEC over FiWi networks from the architectural perspective.

5.3.1 MEC over FiWi Networks

FiWi networks are realized by integrating optical and wireless technologies, thereby forming a powerful platform to provide future-proof connectivity for existing and emerging applications and services [30]. FiWi networks can be built using any optical access networks, for example, EPON and

Table 5.1: Comparison of conventional D-RAN, C-RAN, and emerging cloudlet enhanced D-RAN

	Conventional D-RAN	Cloud-RAN	Emerging cloudlet enhanced D-RAN
Base stations	Standard complexity and high cost	Lower complexity, conceivably cheaper	Lower complexity and cheaper
Diversity gains	Per-BS diversity gains	Multiplexing and computational (multi-user) diversity gains	Both multiplexing and computational diversity gains at the network edge
Hardware resources	Dedicated digital signal processors (DSP) or application specific integrated circuits implementations	High-volume commodity hardware (e.g., general purpose processors)	High-end rack servers, cloudlets, Nokia Siemens Networks' radio applications cloud server
Backhauling	Backhaul links of up to several tens of kilometers, leading to high latencies in the range of several tens of milliseconds	1) Backhaul links with critically higher throughput required and latency on the order of a few milliseconds; 2) CPRI is widely used for fronthaul interface	1) Significantly lower round-trip latency and real-time information offered within RAN; 2) Ethernet-based (e.g., IEEE 1904.3) and virtual networking techniques in the fronthaul in support of variable bit rates
Flexibility	Hardware driven	Software driven	1) Driven by both hardware and software (e.g., Nokia Liquid Application, RACS, OpenStack++); 2) multipoint-to-multipoint communication
Programmability	Based on DSP	Based on general-purpose processor	Supports both DSP and general-purpose processor

next-generation PON 1&2 in the backhaul segment, and RAN technologies in the front-end segment, such as Gigabit-class IEEE 802.11ac very high throughput (VHT) WLAN and 4G LTE/LTE-A.

An Ethernet-based FiWi network may rely on the emerging cloudlet enhanced distributed-RAN (D-RAN) [181] based on so-called radio-and-fiber (R&F) technologies [30]. In cloudlet enhanced D-RAN, the functionalities of remote radio heads (RRHs) and baseband units (BBUs) are split, whereby RRHs and BBUs are linked via an Ethernet interface and the baseband processing is done at a MEC server. Alternatively, a FiWi network can be realized via radio over fiber (RoF) technologies such as C-RAN. In C-RAN, BBUs that connect a number of macro BSs or small cells (i.e., femto-, picocell) are centralized with pool baseband processing (i.e., BBU pool), while

RF signal is digitized and transmitted over optical fiber for fronthauling (i.e., between RRHs and BBUs). Further, the digitalized RF signal received at the RRH is then converted to analog signal before being transmitted to its associated edge devices in the downlink transmission. CPRI is the currently used standard transmission technology in the fronthaul. However, as fronthaul and backhaul in future mobile networks will converge, CPRI may be mapped into Ethernet frames, as specified in the emerging IEEE 1904.3 standard.

Table 5.1 summarizes the architectural evolution of cloud enabled RAN technologies. To better understand the evolution, typical features of cloudlet enhanced D-RAN, C-RAN, and conventional DRAN [200] are summarized in Table 5.1. Note that, unlike the other two concepts, conventional DRAN separates the processing of control and user plane processes. The RAN server is responsible for the management of RAN-specific functions (e.g., mobility, bearer services, and paging), whereas radio and cell-specific functions (e.g., data bearers, scheduling) are decentralized to the base station.

MEC over FiWi networks can be realized by enabling cloud computing capabilities using, for example, a powerful rack server or cloudlets directly connected to the integrated ONU-mesh portal point/access point (AP) (i.e., at the edge of FiWi networks). Even though MEC servers could be deployed at different locations in FiWi networks such as at central office or anywhere along the backhaul segment, the most important design principle is that MEC servers should be in close proximity to subscribers.

MEC over FiWi networks support the coexistence of conventional clouds and cloudlets (Fig. 5.3) in order to provide both centralized and distributed cloud services. Centralized clouds are suitable for stateless services with limited data transmissions, such as web and batch processing. Indeed, transmitting large volumes of data streams to a remote cloud is not only costly but also incurs higher latencies. Conversely, MEC is suitable for real-time applications (e.g., interactive collaboration, augmented reality, and gaming) that require low latency and location-aware data processing. The coexistence of conventional clouds and MEC in FiWi networks provides a powerful hybrid cloud platform for a wide range of applications and helps meet the QoS requirements of both human and machine generated traffic in 5G networks. Further, from an *economical perspective*, MEC over FiWi helps reduce the capital and operational expenditures (CAPEX/OPEX) due to sharing of existing fiber and wireless infrastructures. From a *technical viewpoint*, a unified resource management mechanism can be designed to efficiently operate such a highly integrated network.

5.3.2 MEC over Ethernet-based FiWi Networks

This design scenario is based on the integration of EPON or 10G-EPON in the optical backhaul and a wireless Ethernet LAN (WLAN) in the front-end. Fig. 5.3(a) illustrates MEC over Ethernet-based FiWi network architecture as a shared communication platform for both broadband H2H and MEC services. The backhaul consists of an OLT located at the central office that serves a single or multiple ONUs at the customer premises. A subset of ONUs is located at the premises of residential or business subscribers, providing fiber to the X (FTTx) services (e.g., fiber-to-the-home) to a single or multiple wired subscribers. The second subset of ONUs is equipped with a mesh portal point (MPP) to interface with the WiFi mesh network, consisting of mesh points (MPs) and mesh APs (MAPs), each serving mobile users within their coverage area. The collocated ONU-APs/MPPs are realized by using R&F technologies [30]. MEC servers are connected to ONU-APs/MPPs through optical fiber point-to-point links. The resource management scheme and its performance evaluation for this design scenario are explained in greater detail in Section 5.4.

This design scenario is considered for medium- and small-scale MEC applications, where each ONU-AP/MPP zone covers a small number of edge devices. It may be suitable for Tactile Internet applications and 5G-enabled robots.

5.3.3 MEC over 4G LTE-based FiWi Networks

In this scenario, MEC servers are deployed at 4G LTE macro BSs (MeNBs), as shown in Fig. 5.3(a). Bandwidth allocation is based on a pool/request/grant mechanism. The OLT schedules transmissions and allocates bandwidth to each ONU-MeNB in a centralized fashion. Upon a granted bandwidth, each ONU-MeNB makes local decisions to schedule transmissions and allocate the bandwidth to its associated edge devices and MEC servers in a fair and distributed manner. Depending on the QoS requirements, an ONU-MeNB forwards the packet to either the OLT or the MEC server.

MEC over 4G LTE-based FiWi networks are especially suitable for edge video orchestration, fast moving users (e.g., train passengers), and wide-area applications (e.g., smart cities). Further, it will significantly accelerate the deployment of V2R communications, where mobility and low latency are highly desirable.

5.3.4 Coexistence of MEC and C-RAN in FiWi Enhanced LTE-A HetNets

C-RAN and cloudlet enhanced D-RAN may coexist in HetNets. Fig. 5.3(b) depicts a way to realize such coexistence. Given the ability to overlay multiple channels, wavelength-division multiplexing (WDM)-PON technologies may be deployed in this scenario without upgrading the optical infrastructure, where C-RAN and cloudlet enhanced D-RAN may use different wavelength channels for baseband and RF transmission. This helps reduce CAPEX, since WDM-PON provides a substantial reduction in the number of fibers used. The collocated ONU-FeNB (femtocell BS) and ONU-PeNB (picocell BS) may rely on WDM-based C-RAN, while an ONU-MeNB may rely on cloudlet enhanced D-RAN, as shown in Fig. 5.3(b).

Since MEC servers are connected to the ONU-MeNB, and the ONU-MeNB may rely on cloudlet enhanced D-RAN, the scheduling and bandwidth allocation should be handled by the ONU-MeNB. In the coordination of BBUs, the OLT is fully responsible for scheduling transmissions and allocating bandwidth to each ONU-FeNB and ONU-PeNB in a centralized fashion. However, due to not only the heterogeneity of constituting network components but also the presence of different radio access technologies in the same network, designing a unified resource management scheme is more complex than the previous two scenarios.

Another possibility of deploying MEC over LTE-A HetNet is at the aggregation level. When multiple BSs are located close to each other, it is effective to deploy an MEC server there. Serving several BSs with a single MEC server not only centralizes computing resources but also reduces CAPEX/OPEX of network operators. Further, device-to-device (D2D) links may be employed between edge devices to improve spectrum efficiency and reduce backhaul traffic loads. Deploying MEC servers in LTE-A HetNets and at the aggregation levels pose several challenges. Among others, advanced resource management schemes are needed to jointly coordinate and synchronize a large number of BSs.

This design scenario is suitable for networks with ultra-high densification, fast moving users (e.g., V2V and drones), and mission-critical IoT applications.

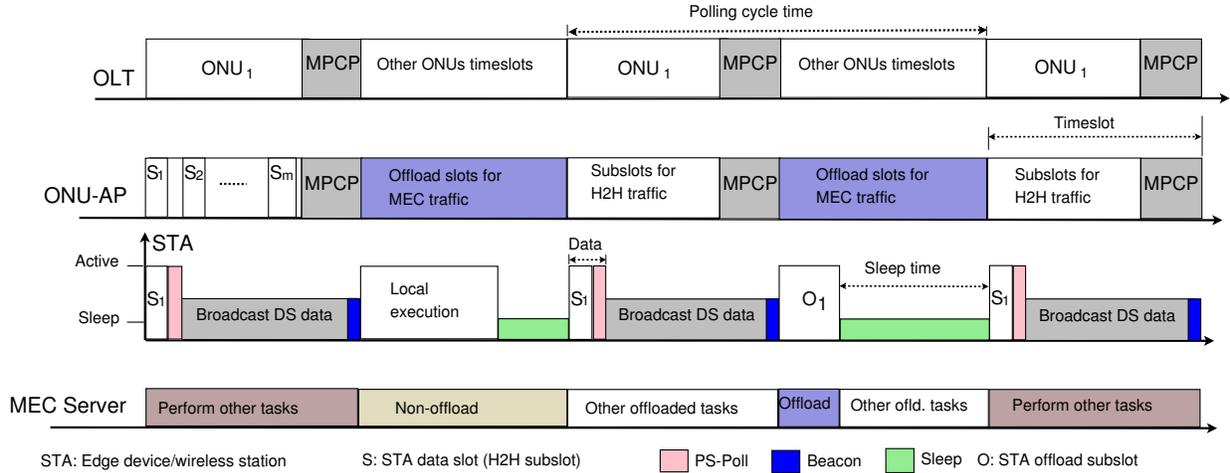


Figure 5.4: Illustration of TDMA-based unified resource management scheme and sleep mode for MEC over WLAN-based FiWi networks.

5.4 Resource Management and Performance Evaluation

5.4.1 Resource Management Scheme for MEC over Ethernet-based FiWi networks

This subsection proposes a TDMA-based unified resource management scheme for MEC over Ethernet-based FiWi networks. TDMA scheduling offers advantages such as reduced energy consumption and collision-free transmission compared to contention-based protocols. In this work, a conventional WLAN topology is considered, where wireless stations (STAs) with no mobility connect directly to APs that are collocated at ONUs.

Figure 5.4 illustrates the unified resource management scheme. The system is based on a two-layer TDM design. The first layer is designed for the optical backhaul, where the OLT schedules timeslots and allocates bandwidth to ONU-APs via multi-point control protocol (MPCP) messages (*GATE* and *REPORT*). In the second layer, the ONU-AP assigns bandwidth in subslots and schedules transmissions of both H2H and MEC traffic for its associated STAs. The transmissions of MEC offloaded traffic and computation results are scheduled outside the H2H subslots within an EPON polling cycle time (Fig. 5.4) to allow for H2H/MEC coexistence without degrading H2H network performance. The ONU-AP receives offloaded traffic from its associated STAs. It then immediately relays the traffic to the MEC server using a dedicated point-to-point fiber communication link.

When the ONU-AP receives the computation results from the MEC server, it broadcasts them to its STAs. A given STA sends its H2H traffic to the ONU-AP within its assigned H2H subslots.

The ONU-AP allocates subslots to its STAs by means of WLAN Beacon and PS-Po11 frames. These frames are extended by using their optional bits to include subslot parameters and bandwidth requests, respectively. The ONU-AP broadcasts a Beacon to its STAs containing an uplink H2H subslot map, whereby each STA sends a PS-Po11 at the end of its own H2H subslot. The ONU-AP aggregates the requested bandwidth and reports it to the OLT via a REPORT at the end of its timeslot. The STA transmits offloaded traffic to and receives computation results from the MEC server in the offload subslot. Further, a power-saving method is employed using a similar approach as in [76] to extend the battery life of edge devices. The general idea is to schedule sleep mode for the STA in a PON cycle, if it is idle after the completion of both its H2H transmission and MEC offloading subslots. For network synchronization, the timestamp mechanism specified in the EPON standard is adopted, where all network devices assign their local clocks to the OLT global clock.

5.4.2 Performance Evaluation

This subsection discusses results and findings obtained from an analytical evaluation of the considered scenario. Computation offloading, that is, offloading compute-intensive tasks to a MEC server connected to an ONU-AP, should be performed if the time to execute a task on the edge device locally is longer than the response time of offloading that task onto an MEC server. This response time difference is called offload gain. The response time efficiency is defined as the ratio of the offload gain and the response time of a task that is locally executed on edge devices. Packet delay is the time a packet waits in a data buffer. The battery life of an edge device is computed based on its battery capacity and the average power consumption.

The TDMA-based resource management scheme is analyzed assuming a polling system with M/G/1 queues [109]. For computation offloading, a face detection application is considered using OpenCV¹⁴. An image size of 500×426 pixels greyscale mode is converted into kilobytes and used as input data for face detection. The data load of a computation task is assumed to be fragmented into packets of fixed size and the application is divided into a number of fine-grained tasks, similar to [97]. The application is modeled as a call graph (directed acyclic graph) with computational

¹⁴Open source computer vision library (OpenCV), <https://www.willowgarage.com/pages/software/opencv>

components, each being characterized by size of methods, energy consumption, and number of instructions to perform the computations.

In the evaluation, an IEEE 802.3ah EPON is considered, and the H2H traffic load (intensity) is varied from 0.05 to 0.9 with Poisson distribution and average packet transmission time of $5.09 \mu\text{s}$ as in [109]. MEC servers are assumed to be 1 rack server (100 CPUs) or 4 rack servers with a clock speed of 3.2 GHz. The considered edge devices are HP iPAQ PDAs with a 400 MHz clock speed and a battery capacity of 1000 mAh Lithium-Ion with power levels in active and sleep states of 0.9 W and 0.3 W, respectively. Maximum data rates of 300 Mb/s and 6900 Mb/s are considered for the wireless front-end based on IEEE 802.11n and IEEE 802.11ac VHT WLANs, respectively.

Figure 5.5(a) depicts the maximum achievable response time efficiency for different offload packet sizes. For increasing offload packet sizes, the average response time efficiency asymptotically approaches 100 percent. For instance, for a typical case of 32 ONU-APs and an offload packet size of 825.60 kB, the average response time efficiency equals 77.10 and 95.36 percent for the two considered WLANs, respectively. This translates into a response time reduction of 77.10 and 95.36 percent with respect to the response time obtained in a non-offload scenario. Figure 5.5(a) also reveals that MEC over VHT WLAN-based FiWi helps increase the maximum achievable response time efficiency significantly. This set of results verifies that MEC over FiWi-based computation offloading is a promising solution for improving users' QoE in support of a wide range of future 5G applications.

Figure 5.5(b) shows the packet delay performance of co-existing H2H and MEC traffic under varying H2H traffic loads. Both curves have a similar behavior. However, the offload packet delay is higher than H2H delay because offloaded traffic waits longer before being transmitted due to a longer resultant PON polling cycle time. An H2H mean packet delay of below 23 ms is achieved for all values of H2H traffic loads. Importantly, for a typical scenario of 32 ONU-APs and a H2H traffic load of 0.3, an MEC mean packet delay of 20.50 ms is obtained. Further, even with a H2H traffic load of 0.7, the MEC delay remains below 95 ms, which may not be achieved with typical centralized clouds. This means that many delay-sensitive applications can be offloaded on the MEC over Ethernet-based FiWi network. However, as shown in Fig. 5.5(b), the offloading may not be efficient when H2H traffic load is greater than 0.8 because of significantly longer delay.

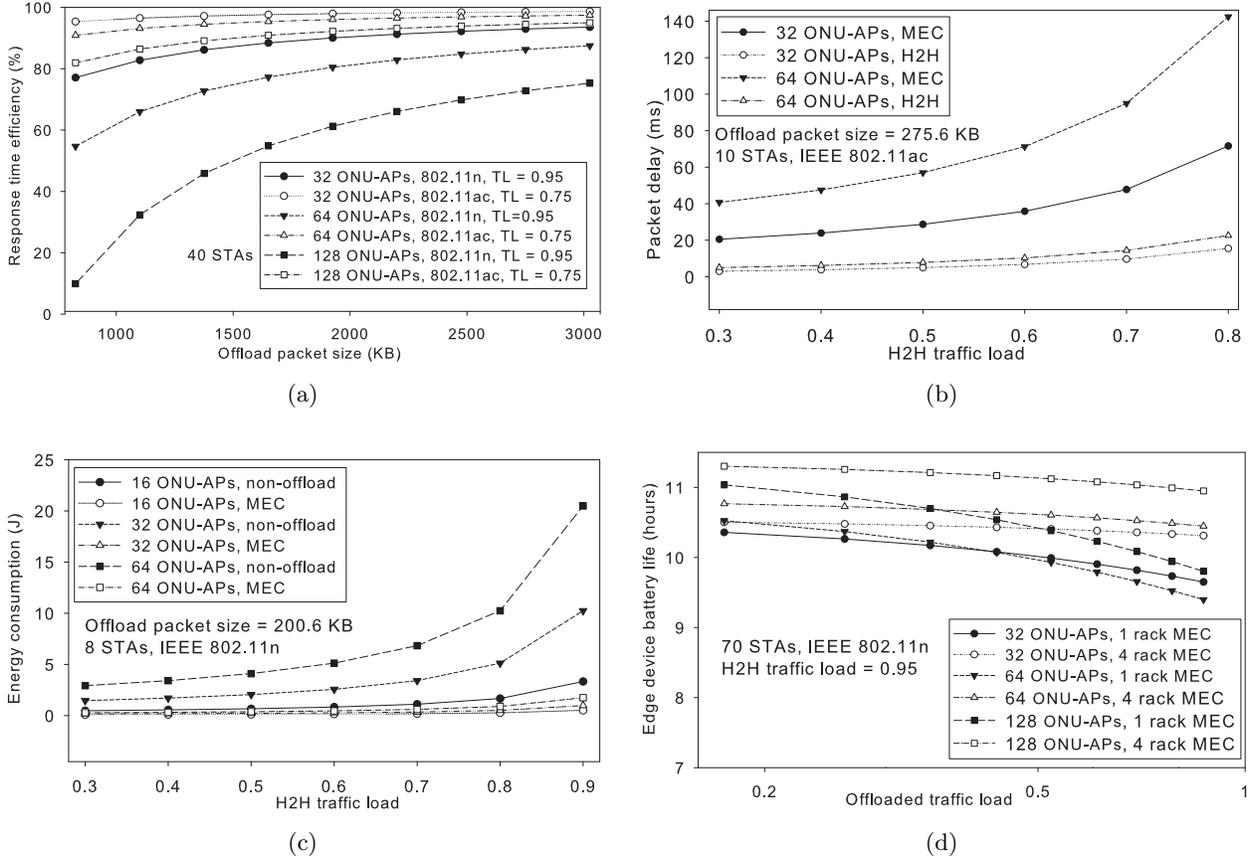


Figure 5.5: a) response time efficiency vs. offload packet size; b) mean packet delay of co-existing H2H and MEC traffic; c) energy consumption vs. H2H traffic load for MEC and non-MEC scenarios; d) battery life of edge devices vs. offloaded traffic load. (TL: H2H traffic load).

Figure 5.5(c) compares the average energy consumption between the MEC scenario and local execution scenario (i.e., non-offload). The energy consumption is a function of H2H traffic load, polling cycle time, and number of ONU-APs and STAs. It is shown in Fig. 5.5(c) that MEC-over FiWi-based offloading significantly reduces the energy consumption of edge devices. For instance, for 32 ONU-APs and an H2H traffic load of 0.8, the energy consumption in the MEC scenario is 10.13 times less than in the non-offload scenario. Finally, Fig. 5.5(d) shows the battery life of edge devices as a function of offloaded traffic load and number of MEC servers. Remarkably, by employing the proposed unified resource management scheme and sleep mode scheduling, up to 11.035 h and 11.302 h of battery life can be obtained with 1-rack and 4-rack MEC server, respectively. This verifies that an MEC-over-Ethernet-based FiWi network with the proposed scheme helps prolong the battery life of edge devices significantly. Note that due to the TDMA nature of the proposed scheme, employing

backhaul links with a higher capacity (e.g., 10G-EPON) in the proposed MEC over FiWi network would not affect the obtained findings.

5.5 Conclusions

This chapter introduces the novel concept of MEC over FiWi networks. Besides several benefits of MEC, a number of interesting research challenges in terms of network integration and coordination, distributed resource management, coexistence of H2H and MEC traffic, cloud and cloudlet coexistence, reliability, and mobility are discussed. Three envisioned design scenarios for MEC over FiWi networks are studied, followed by a novel unified resource management scheme proposed for MEC-over-Ethernet-based FiWi networks. The obtained results show the significant benefits of MEC over FiWi networks. For instance, for a typical scenario, a response time efficiency of 95.36 percent can be achieved compared to non-offloading scenario. Importantly, the mean MEC packet delay of 20.50 ms is obtained for the considered scenario, while allowing for efficient H2H/MEC coexistence without degrading network performance. Moreover, the battery life of edge devices is prolonged up to 11.30 h by employing the proposed solution. While TDMA scheduling has been shown to be effective in MEC over Ethernet-based FiWi networks, resource management schemes for the other two design scenarios and an in-depth performance analysis, especially with wireless backhaul technologies (e.g., millimeter-wave, microwave) would be an interesting topic for future research.

Chapter 6

Mobile-Edge Computing vs. Centralized Cloud Computing in Fiber-Wireless Access Networks

6.1 Introduction

With the advent of a wide variety of smart devices, the Internet of Things (IoT), and emerging 5G applications, tremendous volumes of data will be generated at the network edge. These data may be transmitted for storage and processing on conventional centralized clouds, even though the majority of data can be pre-processed at the network edge [31]. A conventional cloud faces several challenges such as single point of failure, lack of location awareness, reachability, and wide area network (WAN) latencies, among others [13]. Cloudlets were envisioned to support resource-intensive applications by providing access to computing and storage resources for mobile devices, e.g., battery-constrained devices [15]. Indeed, a cloudlet may be viewed as a small data center in a box. Further, a cloudlet runs open-source derivatives of the widely used OpenStack cloud computing

This chapter is based on the following publication:

[109] B. P. Rimal, D. Pham Van, and M. Maier. Mobile-Edge Computing vs. Centralized Cloud Computing in Fiber-Wireless Access Networks, *Proc., IEEE INFOCOM, Workshop on 5G & Beyond - Enabling Technologies and Applications*, pp. 991-996, San Francisco, CA, USA, Apr. 2016.

[110] B. P. Rimal, D. Pham Van, and M. Maier, Mobile-Edge Computing vs. Centralized Cloud Computing Over a Converged Fiber-Wireless Broadband Access Network, *IEEE Transactions on Network and Service Management*, in first revision.

platform, called OpenStack++ (<http://elijah.cs.cmu.edu>). In light of this, *Mobile-Edge Computing (MEC)* has recently emerged [42]. MEC is all about providing access to cloud-like computing, storage, and networking capabilities as well as IT service environment at the network edge within the Radio Access Network (RAN), in close proximity to subscribers. More importantly, MEC is considered an intermediate step for the evolution towards 5G networks [26]. In addition, MEC holds great promise to open new frontiers for network operators, applications, service and content providers, allowing them to flexibly deploy innovative and disruptive services in the area of IoT, immersive environments, and big data analytics, just to name a few.

A conventional centralized cloud has high storage and processing capabilities but it poses large latency. Conversely, MEC may offer lower latency but it has limited computing and storage capabilities compared to a conventional cloud. Therefore, centralized clouds and decentralized cloudlets (i.e., MEC) may coexist and be complementary to each other in order to support a more diverse set of emerging applications and services (e.g., delay-tolerant, mission-critical, and location-aware) in 5G networks.

Being one of the most successful and widely deployed access technologies [2], passive optical networks (PONs) have been recently receiving significant attention in multi-disciplinary research areas, e.g., small-cell backhaul [201] and the relatively new research area of cloud computing in broadband access networks. For instance, the feasibility of using optical broadband access networks, i.e., PONs, as cloud computing servers was studied in [187]. A cloud integrated hybrid wireless-optical broadband access network (WOBAN) was studied in [202]. Recently, the integration of coverage-centric 4G mobile networks and capacity-centric fiber-wireless (FiWi) broadband access networks was introduced in [29] with a focus on backhaul reliability and WiFi offloading. Further, a micro-data center localization scheme was proposed in [203] in the context of fog computing. Note, however, that none of the existing studies deals with MEC and centralized cloud in FiWi networks.

Given the wired/wireless network integration and decentralization trend in 5G networks, FiWi networks present a promising architecture to support both centralized cloud and emerging MEC. Such an envisioned cloud and MEC enabled FiWi network is termed *CM-FiWi network* in this work. Note that existing MEC studies, e.g., MEC Industry Specification Group, focus on the MEC implementation in the context of 4G LTE, whereas this paper examines MEC over WiFi networks due to their their low costs, wide deployments, and high capacity (100 times higher than cellular

networks [32]). CM-FiWi network offers several benefits. From an economical perspective, it would allow mobile operators to not only share existing fiber and wireless infrastructures, but also provide a single communication platform for multiple purposes such as cloud services (MEC, centralized cloud) and broadband access to both fixed and wireless users via fiber backhaul sharing, thereby reducing capital and operational expenditures (CAPEX/OPEX). Further, CM-FiWi provides the ability to realize end-to-end control of the entire network with back/fronthaul taken into account. From a technical viewpoint, a single unified resource management mechanism can be designed to efficiently operate such a highly integrated network. This helps reduce overall network management complexity (e.g., resource management, bandwidth allocation). However, designing integrated CM-FiWi networks is challenging, given the symbiotic coexistence of both broadband traffic and cloud data. Such coexistence further diversifies communication characteristics and requirements. It is worth mentioning that no existing study deals with the cloud/MEC coexistence as well as the coexistence of MEC and human-to-human (H2H) traffic (e.g., triple-play voice, video, and data traffic) in FiWi networks. On the other hand, there exist reliability issues of the optical fiber backhaul (e.g., feeder and distribution fiber cuts) and MEC server (fault within the MEC platform, virtual machine failure, link failures to/from access point). Survivability is becoming increasingly important for both service providers and end customers due to the large amount of carried traffic in high-speed TDM PONs and their significantly increased number of ONUs (up to hundreds) [33]. In case of a service outage, data loss would be extremely high because of the very high capacity of the fiber links [34]. The envisioned network may suffer from widespread service outages. Many of such occurrences yield highly correlated (time and space) node and link failures [35]. Thus, providing an acceptable level of reliability is a critical issue in CM-FiWi networks.

This chapter aims at designing a unified resource management scheme to integrate computation offloading activities with the underlying FiWi operations. The envisioned network architecture is built on the integration of Ethernet PON (EPON), Gigabit-class very high throughput (VHT) WLAN (IEEE 802.11ac), and cloud/MEC servers. To deal with the symbiotic coexistence of H2H and centralized cloud and MEC traffic, this chapter proposes to incorporate computation offload transmissions between edge devices (e.g., cell-phone, PDAs, sensors, wearables) and MEC/cloud servers into the FiWi dynamic bandwidth allocation (DBA) process by leveraging time division multiple access (TDMA) scheduling. It is worth noting that TDMA is a widely deployed access scheme in EPON/10G-EPON and WLAN, e.g., [36–40]. We note that a number of previous studies

on FiWi networks, e.g., [29], consider CSMA/CA as random access control protocol without QoS guarantees. However, the MEC applications considered in this work require deterministic access in order to minimize delay and energy consumption. In addition, it was recently shown in [108] that TDMA scheduling is suitable for delay-sensitive MEC applications (e.g., face detection, video analytics). Therefore, the envisioned CM-FiWi network is designed using dynamic TDMA and polling. Furthermore, this paper explores the opportunities and challenges of survivability in both PON backhaul and MEC in CM-FiWi networks. Towards this end, localized optical redundancy strategies and MEC survival schemes are designed for given fiber and MEC failure probabilities.

To the best of the authors' knowledge, this chapter is the first to comprehensively investigate the performance of two-level cloud-MEC integrated EPON-WLAN based FiWi networks. Towards this end, the main contributions of this chapter are as follows:

- Two-level cloud-MEC architecture is envisioned in FiWi broadband access networks. The CM-FiWi not only creates an infrastructure platform for mobile operators or third-party service providers to offer MEC services from the edge of networks, but also represents a cost-effective solution for supporting diverse types of traffic (H2H, both cloud and MEC). Further, an overview of various aspects of network and service management is presented to highlight different management and service functions in CM-FiWi networks.
- A novel unified resource management scheme is proposed to effectively manage H2H and both cloud traffic types (cloud and MEC) at the same time by leveraging TDMA scheduling with multi-polling. Both MEC and cloud traffic are scheduled outside the transmission slot of H2H traffic.
- Key performance metrics to evaluate MEC are defined, which are instrumental in understanding the interplay between FiWi broadband access network, cloud, and MEC.
- Finally, this chapter presents a comprehensive analytical framework to analyze the performance of the proposed unified resource management scheme. In addition, a probabilistic analytical model is developed to estimate the survivability of the optical backhaul, wireless front-end network, and MEC and study the advantageous impact of various advanced localized fiber-lean backhaul and MEC redundancy strategies.

The remainder of the chapter is structured as follows. Section 6.2 describes in greater detail the architecture of CM-FiWi communications infrastructures and our proposed unified resource management scheme, whose performance is modeled and analyzed in Section 6.3. Section 6.4 presents our obtained results and findings. Finally, Section 6.5 draws conclusions and highlights future research directions.

6.2 CM-FiWi Communication Infrastructures

This section describes the introduced architecture of CM-FiWi communications infrastructures and provides an overview of different aspects of network and service management in CM-FiWi networks. Further, this section presents our proposed unified resource management scheme in greater detail.

6.2.1 Architecture

Fig. 6.1 depicts the architecture of CM-FiWi communications infrastructures. The optical fiber backhaul consists of a widely deployed IEEE 802.3ah EPON/IEEE P802.3av10G-EPON with an extended fiber range of 10-100 km between the central optical line terminal (OLT) at the central office and remote optical network units (ONUs). The optical backhaul provides end users with broadband access and carries the aggregated traffic of the front-end wireless mesh network (WMN). The OLT connects to two subsets of ONUs in a tree-and-branch topology. However, fiber link failures may still happen due to fiber cuts and thus links might become unavailable for routing traffic, as analyzed in greater detail in Section 6.3.

The first subset of ONUs is located at the premises of residential or business subscribers, providing FTTx (e.g., fiber-to-the-home) services to wired subscribers. The second subset of ONUs is equipped with a wireless mesh portal point (MPP)/gateway to interface with the WMN, where aggregated packets enter the optical backhaul. Mesh points (MPs) are relaying nodes supporting wireless mesh services, such as mesh routing selection and forwarding packets over multiple hops on behalf of other MPs, whereby packets are finally relayed to the OLT or MEC server through an ONU-MPP. From the OLT, packets are routed either to the Internet or conventional (centralized) cloud (see Fig. 6.1). Mesh access points (MAPs) are a special type of MP that provide wireless access to IEEE 802.11-compliant mesh clients (end users), i.e., edge devices/STAs such as cell-phone,

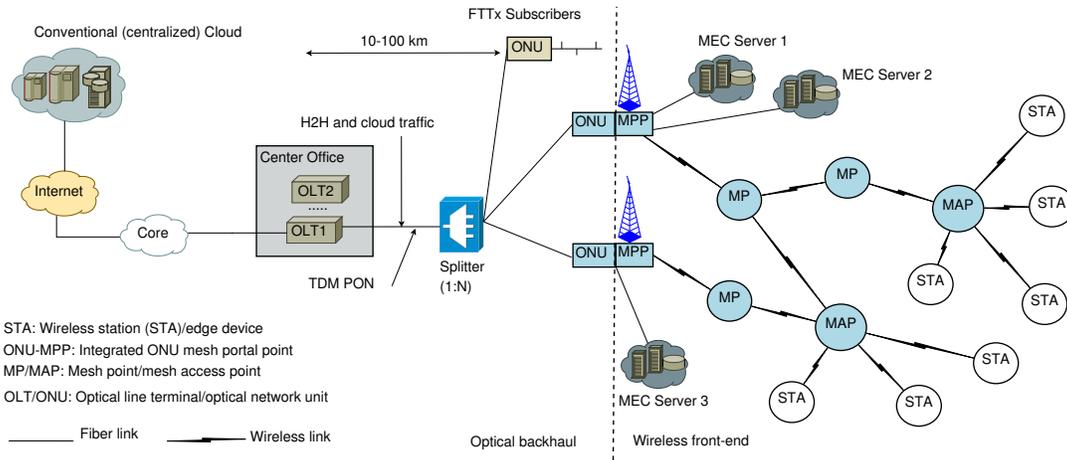


Figure 6.1: Architecture of CM-FiWi communications infrastructures with fiber backhaul sharing.

PDAs, sensors, wearables within their coverage area. The locations of ONU-MPPs, MPs, and MAPs may be strategically planned [41].

Note that in a conventional topology, STAs connect directly to APs rather than to MAPs. The integration of an ONU and an AP is hereafter called “ONU-AP”. Interested readers are referred to [40] for the functional block diagram of a similar ONU-AP architecture. However, note that different from [40], the proposed ONU-AP architecture and resource management solutions have different functionalities. They deal with the coexistence of H2H traffic and traffic from both MEC and cloud. The front-end segment of the CM-FiWi may employ standard wireless technologies such as Wi-Fi. Note that 4G LTE/LTE-A may also be used at the wireless front-end of CM-FiWi networks, which is left for future work.

One or more MEC servers may be connected to the ONU-MPP via dedicated reliable fiber links to provide cloud services at the edge of FiWi access networks (see Fig. 6.1). The MEC server (or system) consists of an MEC hosting infrastructure management system (e.g., server, virtualization layer, and virtualization manager), MEC application platform management system (e.g., service registry, communication services), and application management systems (e.g., virtualized machine for MEC, MEC App). For further details of each component of the MEC server, interested readers are referred to [42]. The envisioned CM-FiWi network is to provide a holistic end-to-end communications infrastructure. The CM-FiWi single network infrastructure can be used for broadband service and both types of cloud services in a cost-effective manner, as explained in greater detail next. Towards this end, the network and service management plays a very important role in CM-

Table 6.1: Summary of different aspects of network and service management in CM-FiWi networks

	Description
Network management	CM-FiWi network manages both optical and wireless access networks. Recall that cellular networks (e.g., LTE/LTE-A) in the front-end of CM-FiWi may also be deployed.
Service management	CM-FiWi network manages both cloud services (MEC, cloud) and broadband access. In addition, since CM-FiWi inherits the principles of EPON, it is capable of supporting multiple existing services, such as edge video services, context-aware services, mobile backhaul, IP services, among others. For further details on the services that PON can support, readers are referred to [204].
Management paradigm	CM-FiWi network uses a hybrid management paradigm. More specifically, while the OLT manages its underlying ONU-MPPs in a centralized fashion, ONU-MPPs are decentralized entities.
Technologies used in the management process (communications protocol)	CM-FiWi has a tree and branch topology, where downstream uses a point-to-multipoint and upstream a multipoint-to-point connection. The IEEE 802.3ah MPCP protocol provides auto-discovery, registration, and ranging operations and a comprehensive signaling (control plane). At layer L2, dynamic bandwidth allocation algorithms are built on top of the MPCP layer, as discussed in Section 6.2.2. At layer L3, a proactive routing protocol based on the delay-aware routing algorithm (DARA) [36] may be adopted in CM-FiWi networks (see Sections 6.2.2 and 6.3)
Methods used to address the management problem	For fault management (e.g., link failure, MEC failure), backhaul and MEC redundancy schemes may be designed and their performance may be analyzed via probabilistic analysis
Economic aspects	Since CM-FiWi provides a unified management solution and the integrated network creates a platform for different types of services as mentioned above, it helps reduce both CAPEX and OPEX.

FiWi networks, which consist of different types of networks in support of a wide variety of services to be managed, communications protocols, methods used to address the management problems, and economic aspects. Table 6.1 summarizes the different aspects of network and service management in the considered CM-FiWi networks.

6.2.2 Unified Resource Management Scheme

It is important to ensure that different and diverse edge devices have access to network resources (e.g., bandwidth, storage) at the edge of CM-FiWi networks. A resource management scheme in CM-FiWi networks is needed to operate efficiently (e.g., without degrading QoS of H2H traffic) and unleash the full potential of the envisioned network architecture. This section introduces such a unified resource management scheme by detailing its operation and resource allocation, and scheduling algorithm.

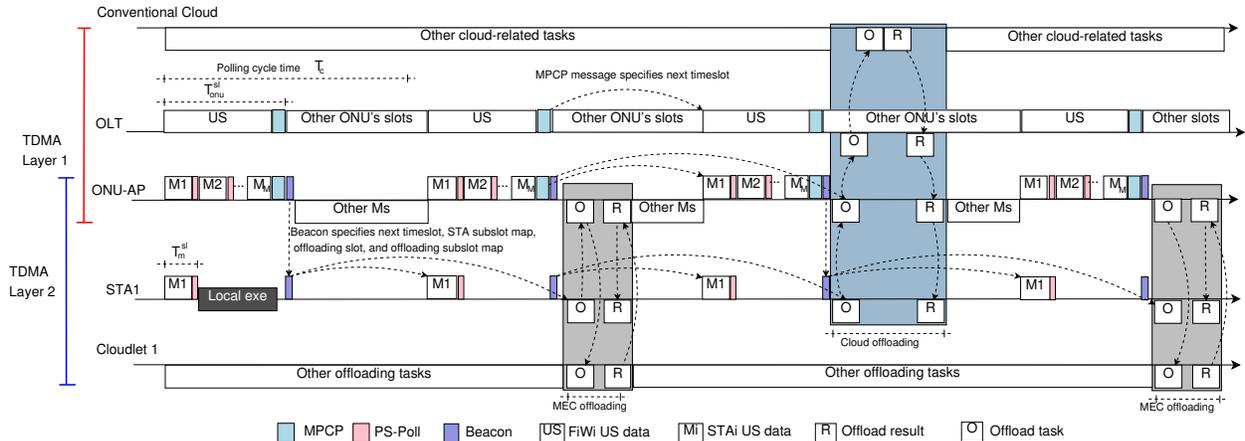


Figure 6.2: Illustration of the proposed unified resource management scheme for CM-FiWi network.

6.2.2.1 General Description

Fig. 6.2 illustrates the operation of the proposed unified resource management scheme. The system is designed via two TDMA tiers with polling for the whole network in a unified manner. The first TDMA tier is designed for the optical backhaul based on IEEE 802.3ah EPON, where the OLT allocates bandwidth and schedules an upstream (US) timeslot for each ONU-AP, while broadcasting downstream (DS) frames to all ONU-APs. Each ONU-AP discards received DS frames that are not destined to it. In the second TDMA tier, the ONU-AP allocates bandwidth in subslots and schedules transmissions of both broadband H2H and MEC traffic of all edge devices destined to it. During its assigned timeslot, an ONU-AP sends its US data frames (e.g., cloud offload/H2H data frame) to the OLT and the ONU-AP receives DS data frames (e.g., offload result, DS H2H data traffic) from the OLT and immediately broadcasts them to its associated STAs.

In a conventional topology (without mesh), edge devices/STAs connect directly to ONU-APs instead of MAPs, which we refer to as *single-hop scenario*. This is an important topology of interest in many deployment scenarios of CM-FiWi networks. As a possible example, in CM-FiWi based mobile, patient health monitoring data can be offloaded to a hospital MEC server via ONU-AP (WiFi access point) for real-time data analytics. Those data may be needed in critical applications such as remote surgery. Indeed, the single-hop scenario is the best way to achieve low latency in CM-FiWi networks for delay-sensitive MEC applications. Unlike in the single-hop scenario, multi-hop scenarios (i.e., with mesh) are mostly suitable for delay-tolerant MEC applications. Furthermore,

the goal of considering the multi-hop scenario is to investigate the maximum number MEC hops that still outperforms conventional cloud-based computation offloading.

Signaling messages: The OLT allocates and schedules timeslots for ONU-APs via the IEEE 802.3ah multipoint control protocol (MPCP). MPCP uses the two polling messages **REPORT** and **GATE**. In this work, the **REPORT** message is extended by using its pad/reserved bits to include additional information (computation offloading) needed for the OLT to schedule the next timeslot of an ONU-AP. Further, the **GATE** message is extended by using its pad/reserved bits to include an offload flag, which is needed to distinguish in DS between H2H frames and offload result frames.

The ONU-AP allocates US subslots to its edge devices by means of **Beacon** and **PS-Poll** control frames, as specified in IEEE 802.11 WLAN standards. The **Beacon** frame is extended by using its optional bits to include a number of slot-related parameters, i.e., start and duration. The broadcast **Beacon** frame contains the ONU-AP's timeslot and an US subslot map, i.e., subslot start and duration of all edge devices. Each edge device sends its required bandwidth for H2H traffic/offload traffic to the ONU-AP by using an extended **PS-Poll** frame that contains an offload flag to notify the ONU-AP about its offload request.

Synchronization: For network synchronization, ONU-MPPs and edge devices assign their local clocks to the OLT global clock, as specified in the timestamp mechanism of the EPON standard. When the OLT receives a **REPORT** from the ONU-MPP, it uses the received timestamp value to calculate the round-trip time between OLT and ONU-MPP. To aid readability, Table 6.2 summarizes the notations, their description, and default values used in this chapter.

6.2.2.2 Cloud Offloading Operation

The transmission of cloud traffic is scheduled outside the ONU-AP's timeslot within a polling cycle time to allow for the cloud/H2H coexistence without degrading H2H network performance (see Fig. 6.2). Once the ONU-AP receives a **PS-Poll** from an edge device, it decides where (cloud or MEC) to offload the traffic based on given QoS and other requirements. In the case of cloud offloading, it embeds the computation offload request in the **REPORT** message sent to the OLT at the end of its timeslot. The start and duration of computation offload subslots are notified by means of a **Beacon** broadcast to edge devices, which is one polling cycle time T_c before the actual

Table 6.2: Notation, description, and default values

Notation	Description	Value/Unit
C, C_{cloud}, C_{mec}	ONU-AP, cloud, MEC transmission capacity	6900 Mbit/s, 10 Gb/s, 10 Gb/s
N	Number of ONUs in system	16, 32, 64, 128
M	Number of wireless STAs	8-100
olt_clk	OLT clock	s
$T_{prop1}, T_{prop2}, T_{prop3}, T_{prop4}$	Air propagation delay between STA and ONU-AP, fiber propagation delay between ONU-AP and OLT, fiber propagation delay between OLT and conventional cloud, fiber propagation delay between ONU-AP and MEC server	0.00033 ms, 0.05 ms, 50 ms, 0.01 ms
T_g, T_c	Guard time between two consecutive slots, PON polling cycle time	1 μ s, ms (variable)
$T_{pon}^{ms}, T_{ul}^{ms}$	Transmission of MPCP message and STA PS-Po11	0.512 μ s, 0.12 μ s
$B_u, T_{start}^o, T_{sl}^o$	buffer backlog, Start and duration of an offload subslot map	s
T_{start}^r, T_{sl}^r	Start and duration of an offload result subslot	s
$P_f^{MPP}, P_f^{MP}, P_f^{MAP}$	Failure probability of an MPP, MP, and MAP	10^{-7}
$\bar{D}_{cloud}, \bar{D}_{mec}$	Offload packet mean delay of cloud, MEC	ms
$\bar{D}_{h2h}^{e2e}, \bar{D}_{h2h}^{e2e'}$	End-to-end mean H2H packet delay without/with front-end WMN	ms
$\bar{R}_{cloud}, \bar{R}_{mec}$	Average response time of cloud, MEC	ms
$\bar{\delta}_{cloud}, \bar{\delta}_{mec}$	Average response time efficiency of cloud, MEC	%
$\bar{\gamma}_{cloud}, \bar{\gamma}_{mec}$	Average offload gain-overhead ratio of cloud, MEC	
CCR_{cloud}, CCR_{mec}	Communication-to-computation ratio (CCR) of cloud, MEC	

computation offload transmission, as shown in Fig. 6.2. When the OLT receives offloaded traffic from the ONU-AP, it relays it to the conventional cloud for execution of the offloading tasks using a fiber link. The OLT waits for the results while coordinating other H2H traffic transmissions. It is assumed that the conventional cloud always has sufficient resources for offloaded tasks and these tasks are assigned to a server upon arrival. After performing computation, the conventional cloud sends the results to the OLT. The OLT uses a one-bit flag (0 or 1) to distinguish DS between H2H frames and offload result frames. Once the OLT receives the results, it sets the flag to 1 and immediately broadcasts them to ONU-MPPs. Each ONU-MPP identifies and filters out its results based on its unique logical link identifier (LLID). The ONU-MPP then broadcasts the results to its associated edge devices. The operations of the OLT will be further detailed in Algorithm 1 shortly.

6.2.2.3 MEC Offloading Operation

Similar to cloud offloading, the transmission of MEC traffic is scheduled outside the ONU-AP's timeslot within a cycle to allow for H2H/MEC coexistence. In this case, the ONU-AP itself determines MEC subslots without informing the OLT. Once the ONU-AP is notified of its next timeslot and cycle time via a **GATE**, it computes both conventional H2H and MEC offload subslots for its edge devices and notifies them via a broadcast **Beacon** (see Fig. 6.2). The edge device sends its H2H and MEC traffic to the ONU-AP within its respective assigned subslots. During an offload subslot, the ONU-AP receives offloaded traffic from the devices and immediately routes it to the MEC server using a dedicated point-to-point fiber link (see also Fig. 6.1). The MEC server performs the execution of offloaded tasks and then sends results back to the ONU-AP using the same fiber link. Once the ONU-AP receives the results from the MEC server, it broadcasts them to its associated edge devices. An edge device receives the broadcast frames from the ONU-AP and filters its destined frames. Note that the small distance from an ONU-AP to MEC server allows for a much smaller propagation delay compared to the one incurred in the case of conventional cloud offloading, i.e., between OLT and cloud server.

6.2.2.4 Algorithm Executed by OLT

The centralized algorithm executed by the OLT, in which the resource allocation is carried out for its associated ONU-APs, is shown in Algorithm 6.1. The OLT assigns a transmission timeslot to an ONU-AP and generates a **GATE** to notify it. A **GATE** message includes *onu_start*, T_{sl} , and an offload subslot map. The timeslot duration is determined based on the request reported by the ONU-AP through a **REPORT** (see lines 1-11). If the OLT clock is in the data interval of the current timeslot, the OLT receives H2H data frames from the ONU-AP (lines 13-14). During an offloading transmission subslot (i.e., in the case of remote cloud offloading), the OLT receives the offloaded packet from an ONU-AP and forwards it immediately to the remote cloud for the execution (lines 15-16). When the OLT receives control message from the remote cloud (i.e., $CMSG_RCVD = TRUE$) that means it will receive offloading results. When the OLT clock is in the offload interval, the OLT receives offload results from the remote cloud and sets $Flag = 1$ to indicate an offloading transmission. The result is sent immediately to the ONU-AP (lines 17-20). Afterwards, the OLT expects to receive a new **REPORT** message and the above steps are repeated. The algorithm executed by an ONU-AP

Algorithm 6.1 Algorithm executed by OLT

```
1: if (REPORT_RCVD = TRUE) then
2:    $B_u \leftarrow$  extracted US buffer backlog, COFlag, and most recently scheduled offload slot  $T_{scd}^o$ 
   from the REPORT
3:    $B_{req} = B_u + T_{pon}^{msg}$ 
4:    $T_{sl} = \min \{B_{req}, T_{sl}^{max}\}$ 
5:    $onu\_start = \max \{(T_{scd}^u + T_g + 1), T_{scd}^o\}$ 
6:    $T_{scd}^u = onu\_start + T_{sl}$ 
7:   for all ( $i \leftarrow 2, N$ ) do
8:      $T_{start}^o[i] = onu\_start[i - 1] + T_{sl}^o[i - 1] + 1$ 
9:      $T_{sl}^o[i] = B_u^o[i]$ 
10:  end for
11:  Generate a GATE message with  $onu\_start$ ,  $T_{sl}$ , and remote cloud offload subslot map
    $\{(T_{start}^o[i], T_{sl}^o[i])\}$ 
12:  Timestamp the GATE with olt_clk and send it to the ONU-AP
13: else if ( $onu\_start \leq olt\_clk \leq onu\_start + T_{sl} - T_{pon}^{msg}$ ) then
14:  Receive H2H US data frames from the ONU-AP
15: else if ( $T_{start}^o[i] \leq olt\_clk \leq T_{start}^o[i] + T_{sl}^o[i]$ ) then
16:  Receive US offload task of  $ONU_i$  and send it immediately to the cloud
17: else if (CMMSG_RCVD[ $i$ ] = TRUE) then
18:  Extract  $T_{start}^r[i], T_{sl}^r[i]$  from the CMMSG
   # receive offload result control message from remote cloud
19: else if ( $T_{start}^r[i] \leq olt\_clk \leq T_{start}^r[i] + T_{sl}^r[i]$ ) then
20:  Receive offload results from the remote cloud, set Flag = 1, then broadcast it immediately
   to the ONU-AP
21: end if
```

and STA can be designed in a similar way. However, due to space constraints, these algorithms are not shown.

6.2.2.5 Multi-Hop Scenario Operation

In the case of a multi-hop WMN scenario of CM-FiWi, ONU-MPPs are connected with mesh routers (MP) wirelessly and MAPs provide access to associated STAs. The overall operation described above for the single-hop scenario can be extended to the multi-hop scenario. To do so, a contention-free multi-polling system is designed following a similar approach, as presented in [40]. Note, however, that the work in [40] neither considers cloud services nor deals with a multi-hop WMN scenario. A modification is therefore needed, as described in the following.

An ONU-MPP schedules transmission subslots and allocates the requested bandwidth to its associated MPs, which may be multiple hops away from it. Similarly, MPs and MAPs schedule

transmission subslots and allocate bandwidth to their associated MAPs and STAs, respectively. Similar to the single-hop scenario, the ONU-MPP, MP, and MAP allocates US subslots to MP, MAP, and STA, respectively by means of the extended `Beacon` and `PS-Poll` control frames. In order to route packets between nodes in WMN a routing protocol is essential. Different routing algorithms for FiWi access networks were studied in the literature, e.g., clustered and localized routing (CluLoR) [205] and proactive routing protocol based on a delay-aware routing algorithm (DARA) [36]. Among them, (DARA) [36] may be adopted in CM-FiWi networks. In DARA, every wireless router is modeled as a queue that maintains global state information and advertises the states of all of its outgoing links using link state advertisements (LSAs) [36]. When a STA transmits its data packet to its associated MAP during the assigned transmission subslots, the packet is forwarded through a number of intermediate MPs until the packet reaches the ONU-MPP. Since each MP receives the LSAs, it identifies the shortest routing path based on DARA.

6.3 Performance Analysis

This section defines and analyzes performance metrics in terms of H2H and MEC packet delay, aggregated response time efficiency, offload gain-overhead ratio, communication-to-computation ratio (CCR), and survivability. Recall from Section 2.1 that these performance metrics of interest reflect key attributes of MEC.

Definition of Performance Metrics

Task: Each task is associated with some data load, which contains input data and a set of operations. Those operations are performed on this data in order to yield a result.

Traffic: Traffic represents the length of the packet (data load) moving from the source node across the network at a given time.

Aggregate response time efficiency: The efficiency of computation offloading depends on both transmission and computation aspects of the network. The offloading is significant only if the time to execute a computation task at the edge device locally is longer than the response time of offloading that task onto a MEC/cloud server. This difference is defined as *computation offload*

gain. The average response time efficiency of cloud $\bar{\delta}_{cloud}$ and MEC $\bar{\delta}_{mec}$ are defined as the ratio of computation offload gain and the response time of tasks that are executed at the edge device locally.

Computation offload gain-overhead ratio: The average computation offload gain-overhead ratio $\bar{\gamma}_{cloud}$ and $\bar{\gamma}_{mec}$ are defined as the ratio of computation offload gain and the computation offload overhead incurred by the communication protocols.

Communication-to-computation ratio: Communication-to-computation ratio (CCR) of both cloud services, i.e., CCR_{cloud} and CCR_{mec} are defined as the ratio of the time required to transmit computation offload packet through CM-FiWi network and the computation time required to execute the offloaded task on the MEC and cloud, respectively.

Survivability: Survivability is the ability of the network to provide services even in the presence of link or node failures. In CM-FiWi networks, MEC reliability depends not only on optical fiber cuts but also on offloaded task execution failure, MEC server failure, and network failure (i.e., transmission failure).

Assumptions

The following assumptions are made in the analysis:

Traffic model: STAs, OLT, centralized cloud server, and MEC server act as traffic sources and destinations. Whereas ONUs, MPPs, MPs, and MAPs forward traffic without generating their own traffic.

Buffer: It is assumed that ONUs and MPs/MAPs always have sufficient buffer to serve the demand of STAs and thus no packets are dropped in the CM-FiWi network.

Task execution model: Computationally intensive applications such as the face detection application is modeled as a set of fine-grained tasks, similarly to [97]. Each task has a computation load that is represented by the number of instructions needed to be executed. Further, it is assumed that the data load of a computation task is fragmented into packets of fixed length. The MEC/Cloud executes a computation intensive task after reassembling all the packets associated with the task. Regarding data integrity, it is worth mentioning that the OLT assigns a Logical Link

Identifier (LLID) to the ONU-AP during the discovery process that allows it to distinguish frames received from and to target frames at a specific ONU-AP. On the other hand, an ONUAP (edge node) uses a frame check sequence based on a 32-bit cyclic redundancy check (CRC) process to guarantee data integrity. Moreover, in this work, tasks execution models are classified into three categories: *a) Cloud-only*: Delay-tolerant applications and compute-intensive tasks are offloaded onto the centralized cloud server, *b) MEC-only*: Delay-sensitive applications (e.g., face recognition, critical-IoT applications, Tactile Internet applications [32], virtual reality) are offloaded onto MEC servers, and *c) Local*: If the latency and energy consumption are higher for executing the tasks on MEC and cloud servers, it is better to execute tasks locally on the edge device.

Location of STAs, MAPs, and MEC server: STAs are uniformly randomly and independently scattered in the coverage area of an MAP (equipped with omnidirectional antennas with a typical reach of 100 m). MPs and MAPs are also uniformly randomly and independently distributed over a given area. One or multiple MEC servers may be connected to each MPP. The placement of MEC servers depends on a number of factors, including performance criteria (e.g., latency), physical deployment constraints, and costs. For simplicity, it is assumed that one MEC server is connected to each MPP.

Failure probability: It is assumed that link failures in an MAP/MP/MPP region and optical fiber cuts occur in an independent manner.

Transmission Time of Offload Packets

The wireless channels are assumed to be block fading with additive white Gaussian noise. The computation offload transmission time consists of the time that the computation offload packet P^{tx} (bits) is transmitted from an edge device to an ONU-AP and the time the computation offload packet is sent from the ONU-AP to the MEC server/conventional cloud via a dedicated optical fiber link. The result receiving time is defined as the sum of time intervals that the result packet P^{rx} (bits) is transmitted from an MEC server/conventional cloud to an ONU-AP. It follows that:

$$T_{cloud}^{ofl} = \frac{(P^{tx} + P^{rx})}{B \log_2(1 + \frac{P}{\sigma} d^{-\beta})} + \frac{2(P^{tx} + P^{rx})}{C_{cloud}}, \quad (6.1)$$

$$T_{mec}^{ofl} = \frac{(P^{tx} + P^{rx})}{B \log_2(1 + \frac{\mathcal{P}}{\sigma} d^{-\beta})} + \frac{(P^{tx} + P^{rx})}{C_{mec}}, \quad (6.2)$$

where T_{cloud}^{ofl} and T_{mec}^{ofl} denote the sum of time for computation offload packet transmit and result reception while computation offloading on centralized cloud and MEC server, respectively. C_{cloud} is the transmission capacity of the link between the OLT and cloud server and is assumed to be the same as that of the PON, whereas C_{mec} is the transmission capacity of the link between the ONU-AP and MEC server.

The denominator of the first component in Eqs. (6.1) and (6.2) is henceforth represented by $r_d = B \log_2(1 + \frac{\mathcal{P}}{\sigma} d^{-\beta})$. r_d is the maximum bandwidth that a wireless communication link can provide, which is defined by the Shannon capacity. \mathcal{P} , d , and B represent the transmitted power, Euclidean distance between the sender and receiver, and bandwidth shared by all nodes, respectively. Further, the interference σ is assumed to be constant throughout the CM-FiWi network with an attenuation exponent $\beta \geq 2$.

Delay Analysis

This subsection presents an end-to-end delay analysis of both clouds (remote cloud and MEC) and H2H traffic.

1) *PON polling cycle time and offload subslot*: For analytical tractability, H2H packets are assumed to arrive at data buffers according to a Poisson process at an aggregate arrival rate λ . We denote the first and second moments of packet service time of ONU-AP_{*i*} by \bar{X} and \bar{X}^2 . We denote the first and second moments of reservation time by \bar{V} and \bar{V}^2 . In addition, all service times and reservation times are considered independent. Let $\rho^{h2h} = \lambda \bar{X}$ be the aggregated H2H traffic load (i.e., utilization factor).

The EPON is modelled as a polling system with reservations, similarly to [40]. The ONU-AP and its associated edge devices are modelled as an N user M/G/1 queue with reservations and vacations, where the ONU-AP goes on vacation, i.e., it does not serve H2H traffic of edge devices after its assigned timeslot (see Fig. 6.3). From a H2H transmission perspective, the ONU-AP time line is decomposed into data interval, reservation interval, and vacation interval (see

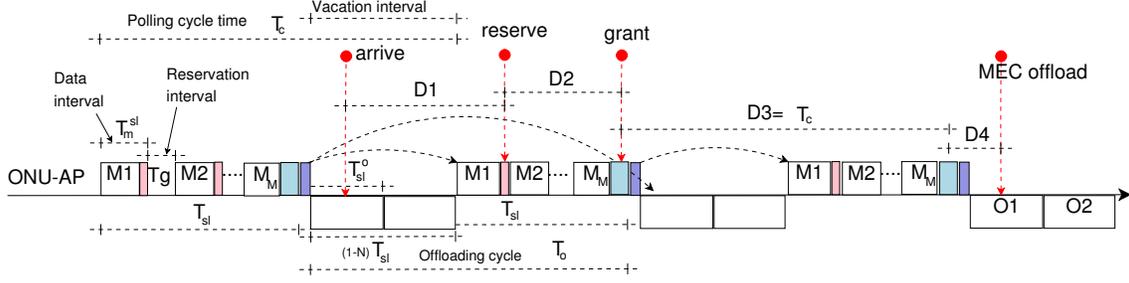


Figure 6.3: Components of packet delay of both conventional H2H and MEC traffic.

Fig. 6.3). The vacation interval is the duration outside the timeslot plus MPCP messaging times, i.e., $(N - 1)T_{sl} + T_{pon}^{msg}$. The reservation interval is $T_{wl}^{msg} + T_g$, where T_g is the guard time that is used to avoid collisions between consecutive uplink transmissions. For the vacation-reservation system, within a polling cycle time T_c , the portion of non-data H2H traffic is $NM(T_{wl}^{msg} + T_g)$, which corresponds to $1 - \rho^{h2h}$. Therefore, T_c is expressed as:

$$T_c = \frac{N[M(T_{wl}^{msg} + T_g)]}{1 - \rho^{h2h}}, \quad (6.3)$$

where $0 \leq \rho^{h2h} < 1$. Let T_{sl}^o denote a computation offload slot, which is defined as the sum of computation offload packet transmission and result reception time. T_{sl}^o is derived based on T_c and cloud/MEC traffic loads (ρ^{cloud}, ρ^{mec}) as:

$$\rho^{cloud} = \rho^{mec} = \frac{M \cdot T_{sl}^o}{T_c} \Rightarrow T_{sl}^o = \frac{\rho^{mec} \cdot T_c}{M}. \quad (6.4)$$

By substituting T_c from Eq. (6.3) in Eq. (6.4), T_{sl}^o becomes:

$$T_{sl}^o = \left(\frac{\rho^{mec}}{1 - \rho^{h2h}} \right) [N(T_{wl}^{msg} + T_g)]. \quad (6.5)$$

Clearly, T_{sl}^o depends on both offloaded and H2H traffic.

2) *MEC Offload Mean Delay*: In the single-hop scenario, edge devices/STAs connect directly to ONU-APs. The mean computation offload packet buffering delay \bar{D}_{mec} is decomposed into four components (see Fig. 6.3) and calculated as follows:

$$\begin{aligned} \bar{D}_{mec} &= \bar{D}_1 + \bar{D}_2 + \bar{D}_3 + \bar{D}_4 \\ &= \frac{T_c}{2} + \frac{T_c}{2N} + T_c + \frac{1}{2} \frac{\rho^{mec} T_c}{N}, \end{aligned} \quad (6.6)$$

where, as shown in Fig. 6.3, \bar{D}_1 denotes the time interval between a computation offload packet arrival and a PS-Poll request sent by a given STA for reserving its subslot. \bar{D}_2 denotes the time interval between the PS-Poll and a received Beacon frame. \bar{D}_3 denotes the time interval between the received Beacon frame and completion of the most recent scheduled ONU-AP timeslot, i.e., when a new Beacon frame is received. \bar{D}_4 is the time interval between the second Beacon frame and the actual transmission time of the packet.

In the multi-hop scenario, a mesh topology is considered, where an edge device/STA connects to an MAP (see Fig. 6.1). In addition to the previous single-hop analysis, it is also interesting to study the performance in this multi-hop WMN scenario. It is assumed that the packets are served by the MP/MAP in a first-come-first-served (FCFS) manner. The WMN in CM-FiWi is modeled by a directed connectivity graph $G(\mathcal{V}, \mathcal{E})$, where \mathcal{V} is the set of $V = |\mathcal{V}|$ mesh nodes (i.e., MPs and MAPs) and \mathcal{E} is the set of point-to-point links. Each link $i \in \mathcal{E}$ can carry offloaded packets and has an average capacity of $C_i \geq 0$. The mean packet delay equals the sum of the delays on each link in the path between source node and destination node, which is computed as follows:

$$\bar{D}_{path} = \sum_{i=1}^N (D_i^{tx} + D_i^{prop} + D_i^{syn} + D_i^{que}), \quad (6.7)$$

where D_i^{tx} , D_i^{prop} , D_i^{syn} , D_i^{que} , and N denote the transmission delay (packet service time) on link i , propagation delay, time-slot synchronization delay, queuing delay at node, and the total number of hops in the path that a packet traverses, respectively. The propagation delay is considered negligible compared to the transmission delay because nodes are close to each other in the front-end segment of CM-FiWi networks. Time-slot synchronization delay is associated with the TDM-based operation of the wireless channel, where each MP/MAP sends packets to its neighboring MPs at the pre-assigned time slot. D_i^{syn} is estimated as $\frac{1}{2\mu C_i}$ [36]. D_i^{tx} is computed as $\frac{1}{\mu C_i}$, where $\frac{1}{\mu}$ and C_i denote the average packet size and the capacity of link i , respectively. For purposes of estimating the queuing delays, each MP/MAP inside the WMN is modeled as an M/M/1 queue, similarly to [36]. Thus, D_i^{que} is computed as $\frac{\rho_i}{\mu C_i - \lambda_i}$ [206], where ρ_i and λ_i denote the link load and packet arrival rate, respectively. After substituting these values, Eq. (6.7) can be rewritten as follows:

$$\bar{D}_{path} = \sum_{i=1}^N \left(\frac{1}{\mu C_i} + \frac{1}{2\mu C_i} + \frac{\rho_i}{\mu C_i - \lambda_i} \right). \quad (6.8)$$

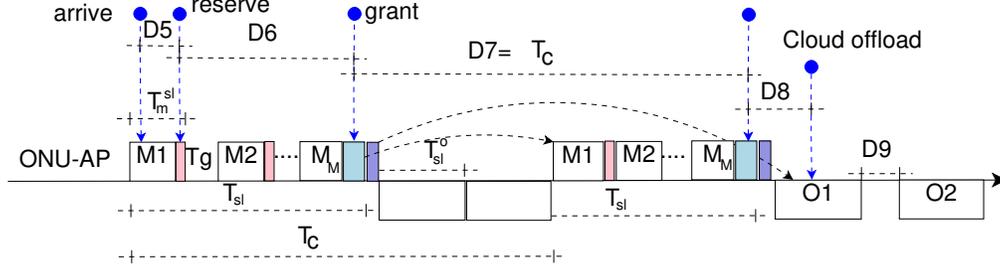


Figure 6.4: Components of packet delay of cloud traffic.

Since a packet also experiences all the delay components in the single-hop scenario as detailed earlier, the average MEC delay in the multi-hop scenario is the sum of the single-hop delay and \bar{D}_{path} for $(N - 1)$ hops. Thus, from Eqs. (6.6) and (6.8), the average MEC delay can be expressed as follows:

$$\bar{D}_{mec} = \begin{cases} \frac{T_c}{2N} (3N + 1 + \rho^{mec}), & \text{if single hop} \\ \left[\frac{T_c}{2N} (3N + 1 + \rho^{mec}) \right] + \sum_{i=1}^{N-1} \left[\frac{1}{\mu C_i} + \frac{1}{2\mu C_i} + \frac{\rho_i}{\mu C_i - \lambda_i} \right], & \text{if multi-hop} \end{cases} \quad (6.9)$$

3) *Cloud Offload Mean Delay*: The queuing delay of the task at the cloud server is assumed to be negligible. The mean computation offload packet buffering delay \bar{D}_{cloud} is decomposed into five components (see Fig 6.4). In a conventional tree-based topology (i.e., single-hop front-end), \bar{D}_{cloud} is computed by adding the delay components from \bar{D}_5 to \bar{D}_9 . Cloud offload mean delay, including front-end WMN in CM-FiWi, is computed by adding delays for single-hop scenario and front-end WMN (i.e., Eq. (6.8) for $(N - 1)$ -hops). Thus, cloud offload mean delay with a single- and multiple-hop front-end WMN is computed as follows:

$$\bar{D}_{cloud} = \begin{cases} \left(\frac{T_c}{2} + \frac{T_c}{2N} + T_c + \frac{1}{2} \frac{\rho^{cloud} T_c}{N} \right) + T_{prop3}, & \text{if single-hop} \\ \left(\frac{T_c}{2} + \frac{T_c}{2N} + T_c + \frac{1}{2} \frac{\rho^{cloud} T_c}{N} \right) + T_{prop3} + \sum_{i=1}^{N-1} \left[\frac{1}{\mu C_i} + \frac{1}{2\mu C_i} + \frac{\rho_i}{\mu C_i - \lambda_i} \right], & \text{if multi-hop} \end{cases} \quad (6.10)$$

where \bar{D}_5 , \bar{D}_6 , \bar{D}_7 , and \bar{D}_8 are defined the same as \bar{D}_1 , \bar{D}_2 , \bar{D}_3 , and \bar{D}_4 , respectively. \bar{D}_9 (i.e., T_{prop3}) is the fiber propagation delay between the OLT and the conventional centralized cloud.

4) *Mean End-to-End H2H Delay*: Given the smallest packet delay among various scheduling disciplines (e.g., gated, exhaustive, limited service) at high network loads [207], in the proposed

system gated service discipline (i.e., T_{max}^{sl} in Algorithm 1 is assumed to be $+\infty$) is employed. In gated service, each ONU-AP is granted a transmission window equal to its reported amount of upstream traffic.

The mean end-to-end H2H packet delay (i.e., from STA to OLT and vice versa) with single-hop front-end in CM-FiWi network is computed by adding up: a) the mean queuing delay at ONU-AP, similarly to [128] and [40], b) mean packet time \bar{X} , c) average propagation delay between ONU-AP and OLT, T_{prop2} , and vice versa, and d) air propagation delay T_{prop1} between STA and ONU-AP, and vice versa. Similarly, the mean end-to-end H2H delay across both the fiber backhaul and WMN front-end segment in CM-FiWi network is computed by summing up H2H packet delay for the single-hop scenario and WMN front-end delay (i.e., Eq. (6.8) for $(N-1)$ hops). Thus, the mean end-to-end H2H delay with single-hop and multiple-hop front-end are expressed as follows:

$$\bar{D}_{h2h}^{e2e} = \begin{cases} \frac{\lambda \bar{X}^2}{2(1-\rho^{h2h})} + \frac{(3N-\rho^{h2h})\bar{V}}{2(1-\rho^{h2h})} + \frac{\sigma_v^2}{2\bar{V}} + \bar{X} + 2T_{prop2} + \frac{(P^{tx}+P^{rx})}{r_d} + 2T_{prop1}, & \text{if single hop} \\ \frac{\lambda \bar{X}^2}{2(1-\rho^{h2h})} + \frac{(3N-\rho^{h2h})\bar{V}}{2(1-\rho^{h2h})} + \frac{\sigma_v^2}{2\bar{V}} + \bar{X} + 2T_{prop2} + \frac{(P^{tx}+P^{rx})}{r_d} + 2T_{prop1} + \\ \sum_{i=1}^{N-1} \left[\frac{1}{\mu C_i} + \frac{1}{2\mu C_i} + \frac{\rho_i}{\mu C_i - \lambda_i} \right], & \text{if multi-hop,} \end{cases} \quad (6.11)$$

where σ_v^2 denotes the variance of each reservation time.

Aggregate Response Time Efficiency

The aggregate response time includes four components: the time to wait for connectivity before sending a packet (buffering delay), packet transmission time, execution time of tasks onto a MEC/cloud server, and result receiving time. The mean response time of executing the offloaded task onto the remote cloud server \bar{R}_{cloud} , MEC server \bar{R}_{mec} , and at the edge device \bar{R}_{local} in the single-hop scenario are given as:

$$\bar{R}_{cloud} = \bar{D}_{cloud} + T_{cloud}^{ofl} + T_{cloud}^{exe} + 2(T_{prop1} + T_{prop2} + T_{prop3}), \quad (6.12)$$

$$\bar{R}_{mec} = \bar{D}_{mec} + T_{mec}^{ofl} + T_{mec}^{exe} + 2(T_{prop1} + T_{prop4}), \quad (6.13)$$

$$\bar{R}_{local} = T_{local}^{exe}, \quad (6.14)$$

where T_{prop4} represents the fiber propagation delay between the ONU-AP and MEC server. The average time to execute a computation task at cloud server \bar{T}_{cloud}^{exe} , MEC server \bar{T}_{mec}^{exe} , and edge device \bar{T}_{local}^{exe} are calculated as follows:

$$T_{cloud}^{exe} = \frac{C'_{cloud}}{S_{cloud}}, T_{mec}^{exe} = \frac{C'_{mec}}{S_{mec}}, T_{local}^{exe} = \frac{C'_{local}}{S_{local}}, \quad (6.15)$$

where C'_{cloud} , C'_{mec} , and C'_{local} denote the number of CPU cycles needed to execute a task at the cloud, MEC, and edge device, respectively. Further, S_{cloud} , S_{mec} , and S_{local} represent the CPU clock speed (cycles/second) of cloud/MEC server and edge device, respectively. Note that the number of CPU cycles depends on the packet size and application.

The average response time efficiency of cloud $\bar{\delta}_{cloud}$ and MEC $\bar{\delta}_{mec}$ are defined as the ratio of offload gain and the response time of tasks that are executed at the edge device locally, which are expressed as follows:

$$\bar{\delta}_{cloud} = \frac{\bar{R}_{local} - \bar{R}_{cloud}}{\bar{R}_{local}}, \quad (6.16) \quad \bar{\delta}_{mec} = \frac{\bar{R}_{local} - \bar{R}_{mec}}{\bar{R}_{local}}. \quad (6.17)$$

The higher the ratio, the smaller the response time compared to the local execution scenario, thus the better the computation offloading. By substituting Eqs. (6.1), (6.10), and (6.15) into Eq. (6.12), \bar{R}_{cloud} is obtained as follows:

$$\begin{aligned} \bar{R}_{cloud} = & \frac{T_c}{2N}(\rho^{cloud} + 3N + 1) + \frac{(P^{tx} + P^{rx})}{r_d} + \frac{C'_{cloud}}{S_{cloud}} \\ & + \frac{2(P^{tx} + P^{rx})}{C_{cloud}} + 4T_{prop3} + 2(T_{prop1} + T_{prop2}). \end{aligned} \quad (6.18)$$

Similarly, by substituting Eqs. (6.2), (6.6), and (6.15) into Eq. (6.13), \bar{R}_{mec} is obtained as follows:

$$\bar{R}_{mec} = \frac{T_c}{2N}(\rho^{mec} + 3N + 1) + \frac{(P^{tx} + P^{rx})}{r_d} + \frac{(P^{tx} + P^{rx})}{C_{mec}} + \frac{C'_{mec}}{S_{mec}} + 2(T_{prop1} + T_{prop4}). \quad (6.19)$$

By substituting Eqs. (6.18) and (6.14) into Eq. (6.16) and Eqs. (6.19) and (6.14) into Eq. (6.17), $\bar{\delta}_{cloud}$ and $\bar{\delta}_{mec}$ are obtained.

Computation Offload Gain-Overhead Ratio

The offload gain-overhead ratio provides insights into how efficient the proposed scheme is. $\bar{\gamma}_{cloud}$ and $\bar{\gamma}_{mec}$ for a single-hop scenario are computed as follows:

$$\bar{\gamma}_{cloud} = \frac{\bar{R}_{local} - \bar{R}_{cloud}}{\bar{D}_{cloud} + T_{cloud}^{ofl}}. \quad (6.20) \quad \bar{\gamma}_{mec} = \frac{\bar{R}_{local} - \bar{R}_{mec}}{\bar{D}_{mec} + T_{mec}^{ofl}}. \quad (6.21)$$

By substituting Eqs. (6.1), (6.10), (6.14), and (6.18) into Eq. (6.20) and Eqs. (6.2), (6.6), (6.14), and (6.19) into Eq. (6.21), $\bar{\gamma}_{cloud}$ and $\bar{\gamma}_{mec}$ are obtained.

Communication-to-Computation Ratio (CCR)

A measure of granularity is the ratio of communication to computation. Communication-to-computation ratio (CCR) of both cloud services, i.e., CCR_{cloud} and CCR_{mec} for a single-hop scenario are computed as follows:

$$CCR_{cloud} = \frac{\bar{D}_{cloud} + T_{cloud}^{ofl}}{T_{cloud}^{exe}}. \quad (6.22) \quad CCR_{mec} = \frac{\bar{D}_{mec} + T_{mec}^{ofl}}{T_{mec}^{exe}}. \quad (6.23)$$

By substituting Eqs. (6.1), (6.10), and (6.15) into Eq. (6.22) and Eqs. (6.2), (6.6), and (6.15) into Eq. (6.23), CCR_{cloud} and CCR_{mec} are obtained. Since different applications have different execution and computation times, CCR for each application also varies accordingly. Based on the value of CCR, the size of tasks may be adjusted for efficient offloading. Further, it can be seen from Eqs. (6.6), (6.22), and (6.23) that the polling cycle time has an important impact on the value of CCR.

Survivability Analysis

In CM-FiWi, network elements (i.e., splitter) are reliable due to their passive (i.e., unpowered) nature. However, PONs with their typical tree-and-branch topology may suffer from link (outdoor feeder and distribution fiber) failures due to fiber cuts that disconnect a single or multiple ONUs and their attached subscribers from the OLT. Recall that MEC should meet design requirements such as high reliability. However, an MEC server may fail due to power outage or malfunctioning. More importantly, providing a reliable MEC service is vital for delay-sensitive MEC applications in CM-FiWi networks. There exists a few studies on reliability in the context of PONs, e.g., [34, 208, 209],

as well as cloud computing in general, e.g., [210]. Different from these studies, from an economical viewpoint, this chapter considers mainly two types of localized fiber lean backhaul redundancy schemes and a MEC protection scheme, as discussed in detail in the following.

a) Interconnection fiber (IF) scheme: Four types of protection schemes were defined in ITU-T Recommendation G.983.1 to improve the reliability in of PONs against link and node failures by means of redundancy. However, these schemes do not include direct inter-ONU communications. It was shown in [211] that the IF protection scheme can provide high connection availability (i.e., 99.999%) between ONUs and OLT. In this work, IFs are deployed to interconnect neighboring ONUs pairwise (see ONU_{N-1} and ONU_N in Fig. 6.5). Each pair of ONUs can protect itself mutually by using the corresponding IF in the case of a distribution fiber cut of either ONU. Note that IF does not mean connecting all ONUs in series. At the downside, an additional optical switch is required at an IF-upgraded ONU. For sparse population densities, IFs are a promising approach to achieve high connection availability between ONUs and OLT in a cost-effective manner [33, 211]. Using fiber mesh connections among ONUs is another possibility for achieving improved survivability. Note, however, that from a practical deployment point of view, this approach would be more expensive than the IF scheme due to larger number of extra optical switches and fiber links.

b) Redundant back-up fiber links scheme: In this scheme, redundant back-up fiber links are deployed between each node pair of each link (i.e., outdoor feeder and distribution back-up fibers (FF and DF), as illustrated in Fig. 6.5)

c) MEC protection scheme: This paper proposes survival schemes for MEC, including a redundant MEC server and a back-up fiber connection between the ONU-MPP and an MEC server (see Fig. 6.5). The survivability analysis of MEC, optical fiber backhaul, and WMN is detailed in the following.

1) MEC: Failures during packet transmission and task execution, fiber cuts between ONU_k and MEC server, and MEC server failure are considered in the MEC survivability analysis. Let $STA = \{1, 2 \dots N_{st}\}$, $ONU = \{1, 2 \dots N_{on}\}$, $MEC = \{1, 2 \dots N_{me}\}$, and $OLT = \{1, 2 \dots N_{ol}\}$ denote the sets of STAs, ONUs, OLTs, respectively, where the last element of each set indicates its number of elements. Let $P_{f(ij)}$ denote the failure probability of the connection between i and j , where $i, j \in (STA \cup ONU \cup OLT \cup MEC)$. The FiWi failure probability between ONU and

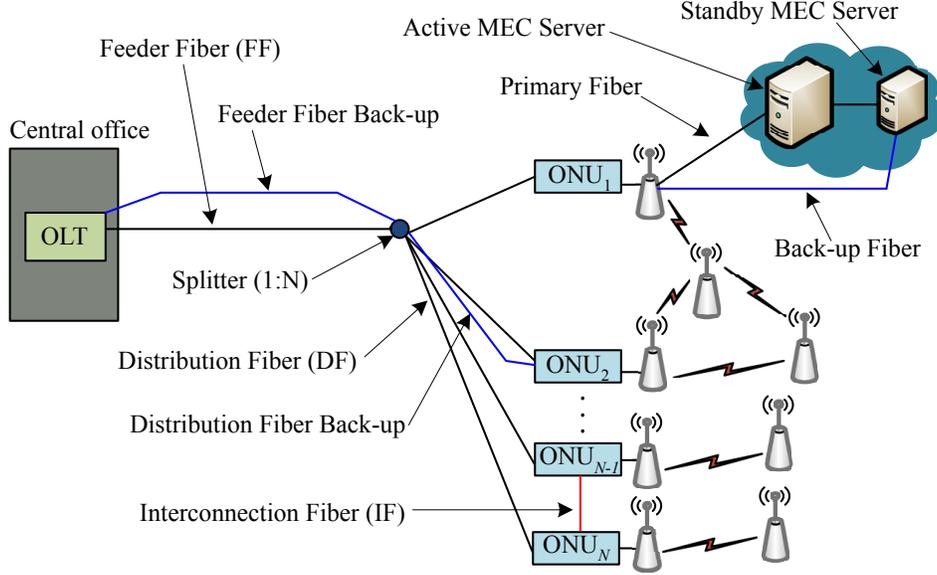


Figure 6.5: Illustration of optical backhaul reliability, MEC reliability, and wireless protection by utilizing redundancy techniques.

MEC server is computed as:

$$P_{f(ij)}^{ONU_n} = 1 - \prod_{n=1}^N (1 - P_{C(ij)})(1 - P'_{C(ij)}), \quad (6.24)$$

where $P_{C(ij)}$ and $P'_{C(ij)}$ represent the fiber cut probability of the primary and back-up fibers between ONUs and MEC server, respectively.

The reliability of each individual element (i.e., packet transmission and offloaded task execution) in MEC is derived by using the following definition [210]:

$$\mathcal{R}^{MEC} = \prod_{l=1}^L \exp\{-(\text{failure rate of elements}) \cdot (\text{working duration of the element})\}, \quad (6.25)$$

where l is the index of network elements. By substituting Eqs. (6.2) and (6.15) into Eq. (6.25) and including the reliability of MEC server, Eq. (6.25) is rewritten as follows:

$$\mathcal{R}^{MEC} = \prod \left[\exp\left\{-F_{tx}^{MEC} \cdot \left(\frac{P^{tx}}{C_{mec}}\right)\right\} \cdot \exp\left\{-F_{ofl}^{MEC} \cdot \left(\frac{C'_{mec}}{S_{mec}}\right)\right\} \right] \cdot \prod (1 - P_f^a)(1 - P_f^s), \quad (6.26)$$

where F_{tx}^{MEC} , F_{ofl}^{MEC} , P_f^a , and P_f^s denote the transmission failure rate, offloaded packet execution failure rate, failure probability of active MEC server, and failure probability of standby MEC server,

respectively. The FiWi connectivity probability of ONU_k with reliable MEC service is computed from Eqs. (6.24) and (6.26) as follows:

$$\mathcal{P}_{C(ij)}^{ONU_k} = \prod_{n=1}^N (1 - P_{f(ij)}^{ONU_n}) \cdot \mathcal{R}^{MEC}. \quad (6.27)$$

2) *Optical Fiber Backhaul Segment*: The FiWi failure probability between ONU_i and the OLT under fiber redundancy schemes can be computed as follows:

$$P_{f(ij)}^{ONU_n} = 1 - \prod_{n=1}^N (1 - P_{C(is)})(1 - P'_{C(is)}) \cdot (1 - P_{C(sj)})(1 - P'_{C(sj)}), \quad (6.28)$$

where $P_{C(is)}$, $P'_{C(is)}$, $P_{C(sj)}$, and $P'_{C(sj)}$ denote the fiber cut probability of the distribution, backup distribution, feeder, and back-up feeder fibers, respectively, whereby s indicates the splitter in the PON. Note that without loss of generality, as shown in Figs. 6.1 and 6.5, a single-stage configuration is considered. Note, however, that Eq. (6.28) can be easily extended for any cascaded multi-stage configuration (e.g., long-reach PON).

3) *WMN Front-end Segment*: A similar approach presented in [29] is followed to estimate the survivability in the WMN front-end of CM-FiWi network. The collocated ONU-MPPs use the WMN to first route traffic wirelessly from an ONU to another ONU and then optically forward it to the OLT. Let $P_f^{Path_i^{w(y,x)}}$ be the probability that the i -th wireless path $Path_i^{w(y,x)}$ between ONU_x and ONU_y fails. It is also considered that NW_i MPs are wirelessly connected to MPP_y and MPP_x via $Path_i^{w(y,x)}$ and MP_{iz_k} is the MP that is k wireless hops away. $P_f^{Path_i^{w(y,x)}}$ is calculated similarly to [29]:

$$P_f^{Path_i^{w(y,x)}} = (1 - P_f^{MPP_y})(1 - P_f^{MP_{iz_1}}) \dots (1 - P_f^{MPP_x})(1 - P_f^{MP_{iz_{NW_i}}}). \quad (6.29)$$

Next, the failure probability of ONU_y collocated with an MPP is estimated similarly to [29]:

$$P_f^{ONU_y} = \prod_{\forall |x \leftrightarrow y} \left[1 - \left(1 - \prod_{i=1}^{N_{WP(y,x)}} P_f^{Path_i^{w(y,x)}} \right) \cdot (1 - P_f^{ONU_x}) \right], \quad (6.30)$$

where $N_{WP(y,x)}$ is the number of paths connecting ONU_y and ONU_x wirelessly.

Overall, for a given failure probability of ONU, MPP, MP, MAP, and MEC, the FiWi connectivity probability of STA_k in CM-FiWi networks is computed as follows:

$$\begin{aligned}
P_{C(ij)}^{STA_k} &= (1 - P_f^{MAP_q})(1 - P_f^{MP_1}) \cdots (1 - P_f^{MP_l})(1 - P_f^{MPP_y}) \\
&\quad \left(1 - \prod_{\forall |x \leftrightarrow y} \left[1 - \left(1 - \prod_{i=1}^{N_{WP(y,x)}} P_f^{Path_i^{w(y,x)}} \right) (1 - P_f^{ONU_x}) \right] \right), \\
&\quad \left(\prod_{n=1}^N (1 - P_{C(ij)})(1 - P'_{C(ij)}) \right) \cdot \left(\prod_{n=1}^N (1 - P_f^{ONU_n}) \cdot \mathcal{R}^{MEC} \right),
\end{aligned} \tag{6.31}$$

where STA_k is connected to MAP_q, which in turn is connected to MPP_y via a number of intermediate MPs, i.e., MP₁, ⋯ MP_l, whereby the values in the last term are substituted from Eqs. (6.24) and (6.26).

6.4 Results

This section presents the results of the performance evaluation based on the defined metrics in Section 6.3.

Configurations: The H2H traffic load (intensity) is varied from 0.3 to 0.9. The traffic load at an integrated ONU-AP is normalized and the maximum traffic load must be less than 1 for stability, as in [128], [40]. C_{cloud} and C_{mec} are set to 10 Gb/s. The PS-Po11 frame size is set to 20 Bytes and T_{wl}^{msg} to 0.16 μs . \bar{X} is assumed to be equal to 5.09 μs [128]. MEC servers may be assembled from the same type of off-the-self servers as in a conventional cloud data center with 3.2 GHz of CPU clock speed. The computing capability of the cloud is considered 1000 times higher than MEC. Amazon EC2 US west-1 is considered as a cloud service provider. The physical (PHY) line rate of WLAN is 6900 Mbit/s. The considered edge devices are HP iPAQ PDAs with a 400-MHz Intel processor. For computation offloading, a face detection application is considered using OpenCV [212]. Other default parameter values are listed in Table 6.2.

Results: In the following, results for a wide variety of network configurations are presented. Fig. 6.6 shows the delay performance as a function of H2H traffic load for an increasing number of ONU-APs $N \in \{32, 64, 128\}$. When H2H traffic load increases, both \bar{D}_{cloud} and \bar{D}_{mec} increase. A similar trend can be observed in the case of H2H packet delay. It is observed that computation

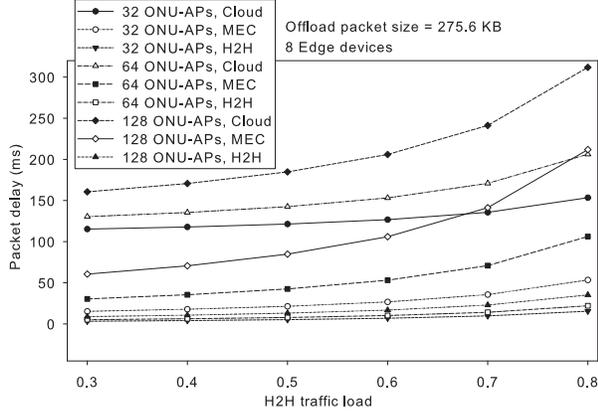


Figure 6.6: Delay performance for single-hop scenario.

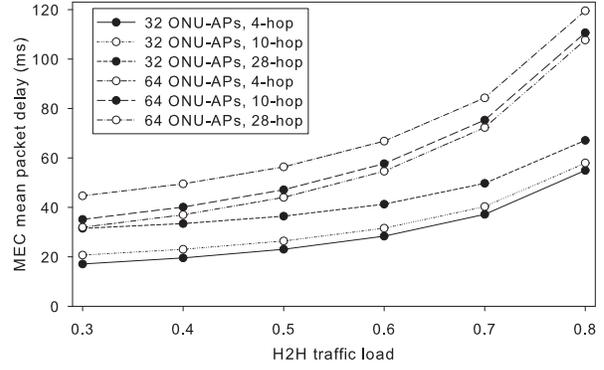


Figure 6.7: Impact of H2H traffic load on MEC packet delay performance in multi-hop scenario.

offload packet delay is higher in the cloud than in MEC. For example, in case of 32 ONU-APs, MEC and cloud delays are below 53.42 ms and 153.42 ms, respectively, for all values of H2H traffic load. This can be explained by Eqs. (6.10) and (6.6). The delay performance is not only affected by the PON polling cycle time but also significantly affected by the propagation delay and aggregate traffic load at ONU-AP. For instance, for a given threshold of 30 ms (as an example) of the MEC traffic delay, the upper bound of permissible aggregate H2H traffic load of 32 ONU-AP should be less than 0.7. Further, as shown in Fig. 6.6, H2H delay is significantly lower than cloud packet delay. This is because the computation offloading transmissions/activities are incorporated outside the H2H timeslots, resulting in a longer waiting time (see also Fig. 6.2 and Eqs. (6.10) and (6.11)). Beside the propagation delay between ONU-AP and OLT, the long distance between OLT and cloud data center is also an important factor that heavily affects the delay performance. Fig. 6.6 indicates that MEC experiences a higher delay compared to H2H because edge devices have to wait at least one polling cycle time to perform computation offloading activities (see also Fig. 6.2). Note that a network reach between the OLT and ONU-APs of 20 km is considered in this work. Alternatively, a long-range PON (LR-PON) may be used to extend the fiber reach up to 100 km in order to increase the coverage area of optical access networks. However, this will result in an increased propagation delay because of the higher round-trip-time of LR-PONs, which will eventually affect the overall network performance.

Importantly, Fig. 6.6 provides insights where to offload a computation task to. For instance, for a typical case of 32 ONU-APs, H2H traffic load less than or equal to 0.5, MEC experiences a delay of less 21.39 ms, whereas cloud has 121.40 ms. This means that many delay-sensitive applications

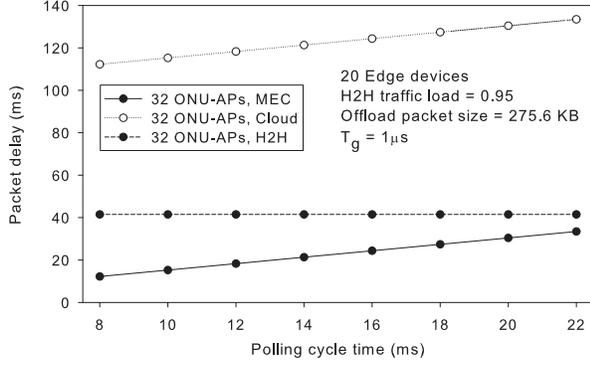


Figure 6.8: Delay performance as function of polling cycle time.

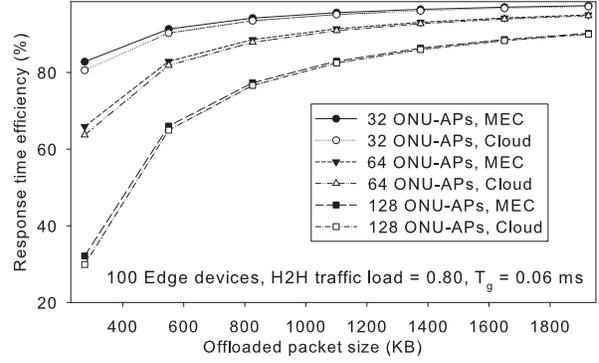


Figure 6.9: Response time efficiency vs. offloaded packet size.

can be offloaded on the MEC integrated FiWi network rather than on the conventional cloud. It is better to offload delay-tolerant applications on the cloud when H2H traffic load is greater than 0.5. The computation offloading of applications may not be practical in both MEC and cloud when H2H traffic load is greater than 0.8 because of significantly longer delay. Therefore, in such a case, tasks should be locally executed on edge devices.

Fig. 6.7 shows the MEC delay performance as a function of H2H traffic load and the number of wireless hops for an increasing number of ONU-MPPs $N \in \{32, 64\}$. We observe that when H2H traffic load increases, \bar{D}_{mec} increases in the multi-hop scenario as well, similar to the single-hop scenario described in Fig. 6.6 above. In addition, as expected, Fig. 6.7 shows that MEC delay increases when the number of wireless hops $\mathcal{N} \in \{4, 10, 28\}$ increases (see Eq. (6.9)). Importantly, MEC with a maximum of 4 hops still outperforms a conventional cloud for a typical scenario of 32 ONU-APs and an H2H heavy traffic load of 0.8 (see Figs. 6.6 and 6.7). This demonstrates that delay-tolerant applications can be offloaded onto 4 hops MEC instead of the conventional cloud for a better network performance. In addition, this indicates that a powerful multi-core MEC server can be deployed at an aggregation point, where multiple ONU-APs are located close together sharing a single MEC platform. This approach centralizes resources and thus helps reduce both CAPEX and OPEX without experiencing significant latency.

Fig. 6.8 shows the delay performance as a function of the polling cycle T_c under a given traffic load. We observe that when the value of T_c increases the values of \bar{D}_{mec} and \bar{D}_{cloud} increase linearly. In the case of remote cloud offloading, cloud traffic should wait for at least one polling cycle time before the actual offload transmission. When priority is given to H2H traffic, MEC traffic also needs

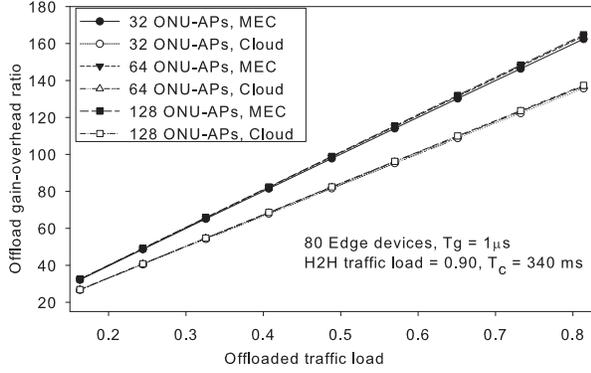


Figure 6.10: Offload gain-overhead ratio vs. offloaded traffic load.

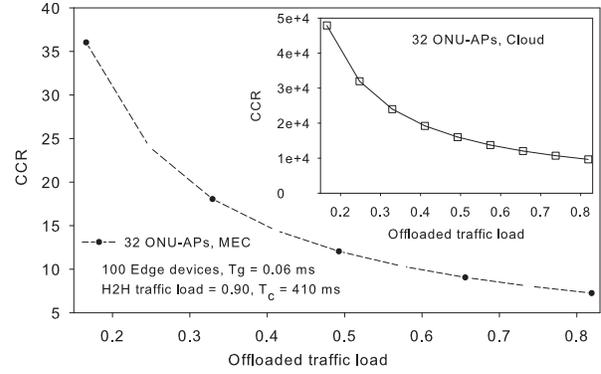


Figure 6.11: CCR vs. offloaded traffic load.

to wait before the actual offload transmission. Therefore, if T_c is longer, both types of cloud traffic should wait longer, see Figs. (6.2) and (6.4) as well as Eqs. (6.6) and (6.10).

Fig. 6.9 compares the response time efficiency of cloud and MEC offloading scenarios. The response time efficiency increases with increased computation offload packet sizes (see Fig. 6.9). Interestingly, for increasing computation offload packet sizes the response time efficiency asymptotically approaches 100%. In the considered scenario, $\bar{\delta}_{mec}$ is always higher than $\bar{\delta}_{cloud}$. For instance, for a typical case of 32 ONU-APs and an offloaded packet size of 825.60 KB, $\bar{\delta}_{mec}$ and $\bar{\delta}_{cloud}$ equal 94.19% and 93.52%, respectively. This translates into a delay reduction of 94.19% and 93.52% with respect to the local execution scenario.

Fig. 6.10 shows that computation offload gain-overhead ratio is a linear function of the offloaded traffic load in both cloud and MEC scenarios. Note that the higher the offload gain-overhead ratio, the more beneficial the computation offloading. For the considered scenario, $\bar{\gamma}_{mec}$ is always higher than $\bar{\gamma}_{cloud}$. For instance, for a typical scenario of 32 ONU-APs and offloaded traffic load of 0.5, $\bar{\gamma}_{mec}$ and $\bar{\gamma}_{cloud}$ equal 81.21 and 61.37, respectively. That further indicates the effectiveness of MEC offloading. In addition, Fig. 6.10 provides important insights into the computation offloading strategies. Recall that communication overhead can be minimized by designing an efficient resource management scheme to handle both broadband access and cloud traffic. The low packet delay and high offload gain-overhead ratios in Figs. 6.6 and 6.10 verify the feasibility and effectiveness of the cloud and MEC enabled FiWi networks and our proposed resource management scheme.

Fig. 6.11 shows the impact of offloaded traffic load on CCR for a typical scenario of 32 ONU-APs. For increasing offloaded traffic loads, the value of CCR decreases. This is because the offloaded

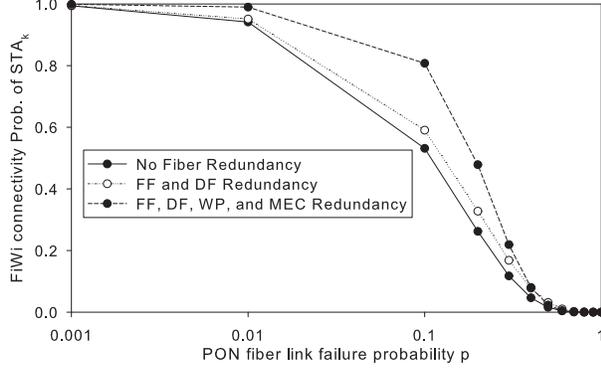


Figure 6.12: FiWi connectivity probability of STA_k vs. optical backhaul failure probability p with different survival schemes.

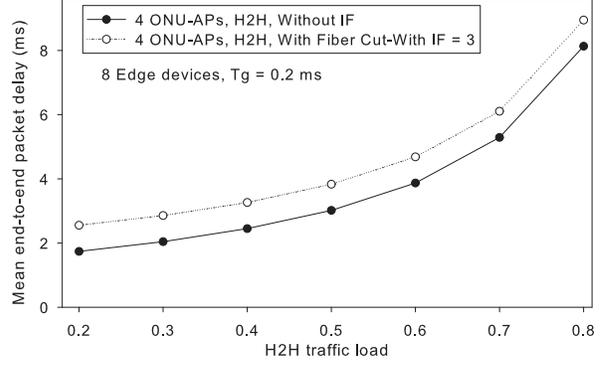


Figure 6.13: Mean end-to-end delay vs H2H traffic load with/without IF scenario.

execution time is inversely proportional to CCR (see Eqs. (6.23) and (6.22)). The values of CCR_{mec} are always less than the values of CCR_{cloud} for all considered offloaded traffic loads. This is because CCR depends not only on the task granularity but also on the values of incurred delay (i.e., \bar{D}_{mec} and \bar{D}_{cloud}) and T_c (see Eqs. (6.23) and (6.22)). Further, the values of \bar{D}_{mec} and \bar{D}_{cloud} depend on T_c , as described in Eqs. (6.9), (6.10), and Fig. 6.8 above. It is worth mentioning that we observed a similar trend with other values of ONU-APs (not shown in Fig. 6.11).

The benefits of the survivability schemes for the FiWi connectivity probability of STAs are shown in Fig. 6.12, where an ONU wirelessly connects to its neighboring ONUs via 3-hop WMN paths in the wireless protection scheme. Fig. 6.12 shows that by jointly employing FF, DF, IF, wireless protection, and MEC redundancy schemes, the FiWi connectivity is significantly improved compared to the scenario of no fiber redundancy for a wide range of PON fiber link failure probabilities. For instance, for a typical scenario with a PON fiber link failure probability of $p = 0.1$, a FiWi connectivity probability of STAs of 0.81 is achieved. Hence, CM-FiWi networks help provide highly reliable broadband access and MEC services. Finally, Fig. 6.13 depicts the mean end-to-end H2H packet delay performance of CM-FiWi networks with 3 IF-interconnected ONUs versus H2H traffic load. The end-to-end H2H packet delay is slightly higher for the IF scheme than in the case of no IF. This is due to the total number of inter-ONU hops and transmission and propagation delays between ONUs.

6.5 Conclusions and Outlook

This chapter studied the feasibility of designing the symbiotic coexistence of H2H and cloud services (both centralized and MEC scenarios). The chapter investigated the performance gains obtained from the implementation of conventional cloud and MEC in FiWi broadband access networks. A novel multi-tier TDMA-based unified resource management scheme was proposed to schedule cloud, MEC, and conventional H2H traffic at the same time. A comprehensive performance analysis was conducted to evaluate packet delay, aggregated response time efficiency, computation offload gain-overhead ratio, and CCR ratio. In addition, a probabilistic analytical model was developed to estimate the survivability of the optical fiber backhaul, WMN front-end, and MEC in CM-FiWi networks. The presented solution helps reduce MEC offload packet delay without affecting the performance of H2H traffic. The obtained results show that for a typical system of 32 ONU-APs and H2H traffic load of 0.6, MEC and cloud packet delay of 26.73 ms and 126.72 ms can be achieved, respectively, without degrading network performance under H2H traffic. The obtained results also provide insights into finding the optimal decision about when and where to offload computation tasks under different H2H traffic load scenarios. More precisely, our results show that it is better to offload delay-sensitive applications on MEC up to a H2H traffic load of 0.5. Based on the obtained results, we also suggest that the maximum number of MEC hops should not exceed 4 in CM-FiWi networks for acceptable network performance. Further, our results demonstrate the effectiveness of the presented survival schemes by providing edge devices/STAs with highly fault-tolerant FiWi connectivity. By jointly employing FF, DF, IF, wireless protection, and MEC redundancy, the FiWi connectivity probability of STAs is achieved higher even in the case of higher PON fiber link failure probabilities. The proposed solution might need to be revisited for other use cases of MEC such as IoT and smart cities, where a huge number of edge devices enforce network synchronization issues.

The concept of MEC is in its early stage. MEC may redefine how services are implemented in fixed/mobile networks. There are several directions for future research in CM-FiWi networks. For instance, a few studies have begun to explore optical networking support for software-defined networking (SDN), referred to as software-defined optical networks (SDONs), as an emerging concept that helps leverage the flexibility of SDN control for supporting networking applications with an underlying optical network infrastructure [213]. Since MEC, SDONs, and the concept of network function virtualization (NFV) are complementary, they can be deployed independently from each

other or jointly. CM-FiWi should support these concepts at the edge. As all three concepts are based on virtualization, they can share the same virtualized infrastructure for not only exploitation of virtualization to dynamically pool resources but also enhanced flexibility in provisioning and configuration of 3G/4G/5G connectivity services. Another potential direction could be to use time and wavelength division multiplexing (TWDM) PON in CM-FiWi networks. The asynchronous delivery oriented resource allocation proposed in [214] may be applied to improve the resource allocation in this context.

Chapter 7

Emerging Deployment Scenario of Cloud-Cloudlet Enhanced FiWi Access Networks: The Tactile Internet

My key contributions in this work were as follows: a) fully engaged in the brainstorming phase, initial phase of formulating the ideas, and writing a raw draft; b) creating Fig. 7.1(b), c) major contributions in Section 7.3.2, including Figs. 7.4 and 7.5, and d) identifying open research challenges in Section IV.

7.1 Introduction

The IEEE Digital Senses Initiative (DSI) is the newest initiative by the Technical Activities Board Future Directions Committee, launched in June 2015. DSI is dedicated to advancing technologies that capture and reproduce various stimuli (e.g., sight, hearing, touch, smell, and taste) from the outside world and let humans as well as machines perceive and react to the combined stimuli in various ways. An interesting early example is the commercially available oPhone, which allows smartphone users to send digital scent messages with more than 300,000 unique aroma combi-

This chapter is based on the following publication:

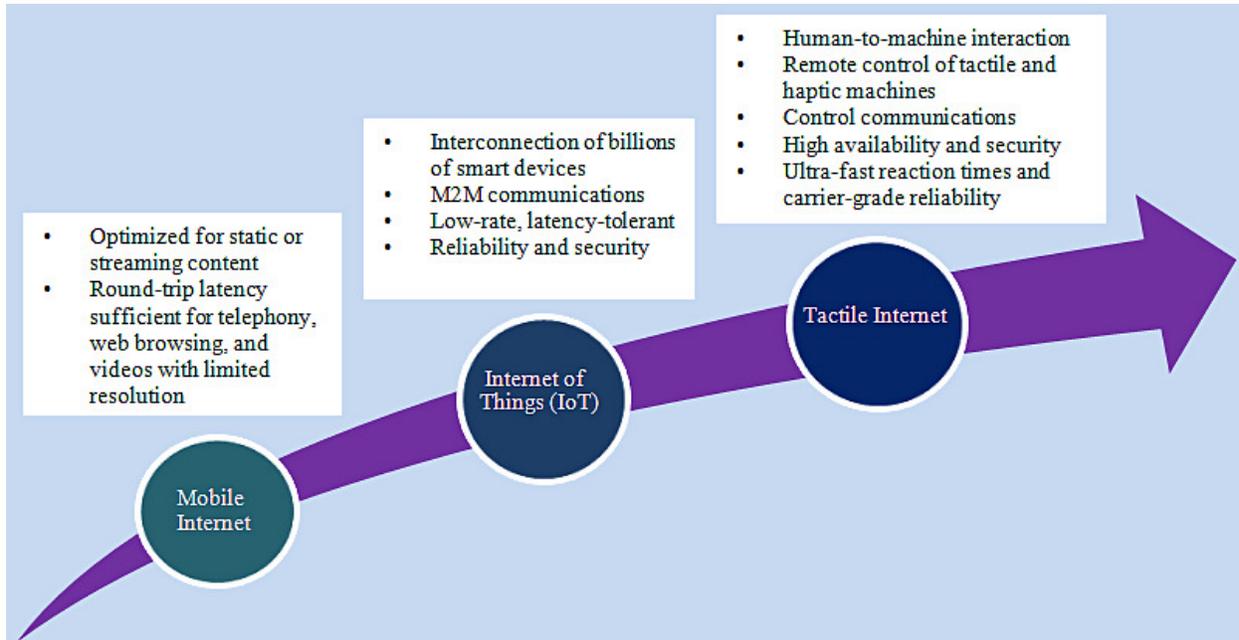
[111] M. Maier, M. Chowdhury, B. P. Rimal, and D. Pham Van. The Tactile Internet: Vision, Recent Progress, and Open Challenges. *IEEE Communications Magazine*, vol. 54, no. 5, pp. 138-145, May 2016.

nations [215]. Another example is remote-presence robots, e.g., Suitable Technologies' *BeamPro*, which consist of a flat screen and video camera mounted on a mobile pedestal.

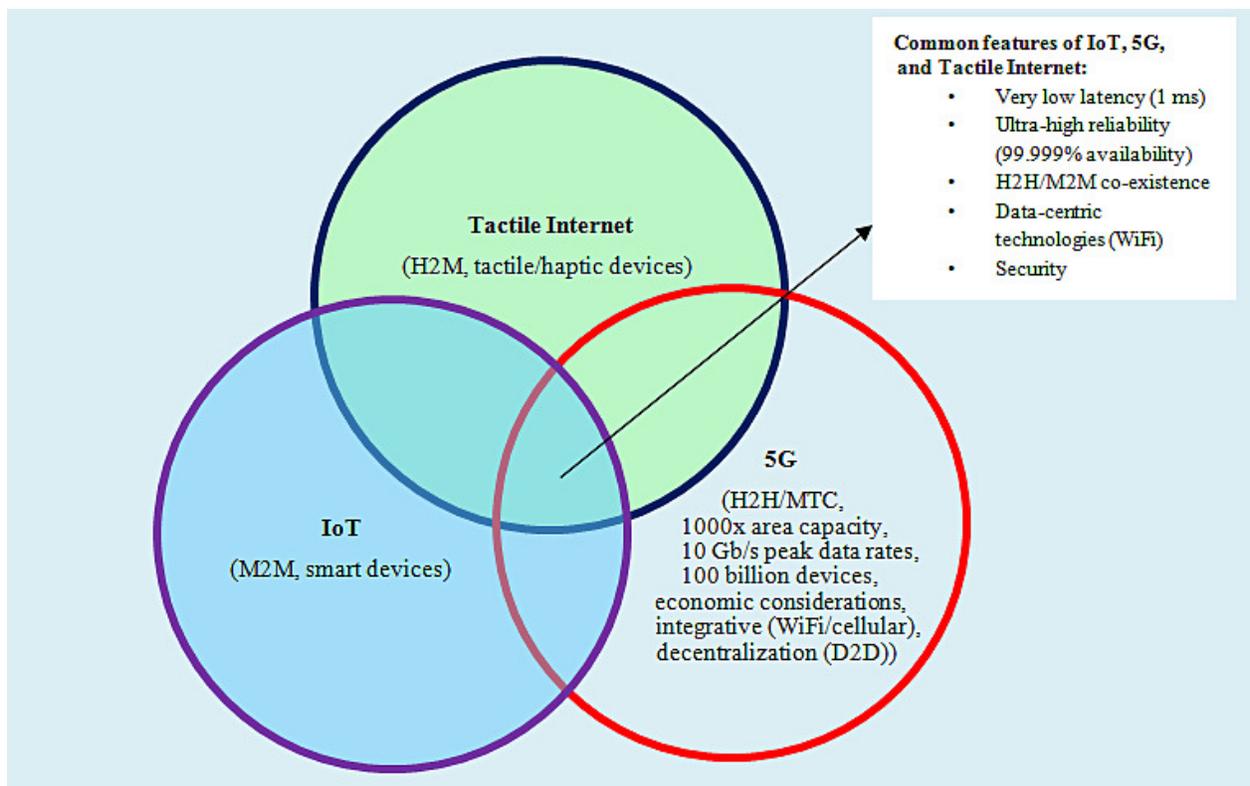
In an interview in 2013, we reflected on a future economic “golden age” of technological convergence in the 2020s, where important tasks of everyday life may be increasingly done by robots [43]. As a personal example, we envisioned the desirable possibility of not only monitoring but also acting from Canada remotely via the Internet in support of our elderly parents living in Germany. This vision of the Internet is now widely known as the so-called *Tactile Internet*, a term first coined by G. P. Fettweis in early 2014 [28, 44]. The Tactile Internet is expected to have the potential to create a plethora of new opportunities and applications that reshape our life and economy. A preliminary market analysis has revealed that the potential market could extend to US\$20 trillion worldwide, which is around 20% of today's worldwide GDP [45].

In various real-time cyber-physical systems (CPSs), including virtual and augmented reality, an extremely low round-trip latency of below 1 ms is required. An important CPS example is the smart grid and its fast response time requirements in the event of (cascading) power network failures. Current cellular and WLAN systems miss this target by at least one order of magnitude. A round-trip latency of 1 ms can potentially move today's mobile broadband experience into the new world of the Tactile Internet. Beside voice and data communications, current 4G mobile networks enable real-time access to richer content and enable early applications of machine-to-machine (M2M) or machine type communication (MTC). Once machines become connected, the next natural leap is to have them controlled remotely. This will generate a completely new paradigm for control communications to steer/control elements of our surroundings and environment [28]. A round-trip latency of 1 ms in conjunction with carrier-grade robustness and availability will enable the Tactile Internet for steering and control of real and virtual objects [44]. However, the Tactile Internet comes with a caveat: It should amplify the differences between machines and humans. By building on the areas where machines are strong and humans are weak, the machines are more likely to complement humans rather than substitute for them. The value of human inputs will grow, not shrink, as the power of machines increases [216].

To facilitate a better understanding of the Tactile Internet, it is helpful to compare it to the emerging Internet of Things (IoT) and 5G mobile networks and elaborate on their commonalities and subtle differences. To begin with, it is worthwhile to mention that the concept of IoT is far from



(a)



(b)

Figure 7.1: The Tactile Internet: a) revolutionary leap of the Tactile Internet (in compliance with ITU-T Technology Watch Report [46]); b) the three lenses of IoT, 5G, and the Tactile Internet: Commonalities and differences.

novel. In fact, the term “Internet of Things” was coined by Kevin Ashton from MIT no less than 20 years ago in 1995. However, it is only recently that we are witnessing the explosive growth of the IoT [199]. Figure 7.1(a) shows the revolutionary leap of the Tactile Internet according to a recent ITU-T Technology Watch Report [46]. The high availability and security, ultra-fast reaction times, and carrier-grade reliability of the Tactile Internet will add a new dimension to human-to-machine interaction by enabling tactile and haptic sensations. On the other hand, future 5G networks will have to be able to cope with the unprecedented growth of mobile data traffic as well as the huge volumes of data from the smart devices that will power the IoT. Towards this end, the 5G technology vision foresees 1000-fold gains in area capacity, 10 Gb/s peak data rates, and connections for at least 100 billion devices. The key challenge of 5G wireless access and core network architectures is to make it possible to address novel machine-centric use cases that are currently not addressed by cellular networks. Potential 5G applications range from industry, robots and drones, virtual and augmented reality, to healthcare, road traffic, and smart grid [217]. Some of these envisioned 5G applications require very low latency on the order of 1 ms or less and ultra-high reliability with essentially guaranteed availability. Thus, beside very low latency, 5G should enable connectivity, whose reliability will have to be orders of magnitude higher than in current radio access networks. Unlike the previous four generations, 5G will be also highly integrative. The integrative vision of 5G will lead to an increasing integration of cellular and WiFi technologies and standards. Another important aspect of the 5G vision is decentralization by evolving the cell-centric architecture into a device-centric one and exploiting intelligence at the device side (human or machine), for example via device-to-device (D2D) communication or user equipment (UE) assisted mobility.

Clearly, the discussion above shows that there is a significant overlap among IoT, 5G, and the Tactile Internet, though each one of them exhibits its unique characteristics. For illustration, Fig. 7.1(b) provides a view of the aforementioned commonalities and differences through the three lenses of IoT, 5G, and the Tactile Internet. The major differences may be best expressed in terms of underlying communications paradigms and enabling devices. IoT relies on M2M communications with a focus on smart devices (e.g., sensors and actuators). In co-existence with emerging MTC, 5G will maintain its traditional human-to-human (H2H) communications paradigm for conventional triple-play services (voice, video, data) with a growing focus on the integration with other wireless technologies (most notably WiFi) and decentralization. Conversely, the Tactile Internet will be centered around human-to-machine (H2M) communications leveraging tactile/haptic devices. More

importantly, despite their differences, IoT, 5G, and the Tactile Internet seem to converge toward a common set of important design goals:

- Very low latency on the order of 1 ms.
- Ultra-high reliability with an almost guaranteed availability of 99.999 percent.
- H2H/M2M coexistence.
- Integration of data-centric technologies with a particular focus on WiFi.
- Security.

We note that there already exist recent excellent surveys on the Tactile Internet, most notably the aforementioned [44] and [217], which elaborate on its rationale and potential. However, both of these surveys take a rather 5G-centric approach with a focus on the wireless front-end and do not report on any early results and obtained findings. Conversely, this survey tries to approach the Tactile Internet from various angles and differs from previous Tactile Internet surveys in a number of ways. Specifically, our survey touches on the importance of high-speed fault-tolerant fiber backhaul infrastructures, as well as complementary technologies and techniques such as WiFi offloading and cloudlets, given that state-of-the-art robots, for example, Aldebaran's humanoid robot NAO, rely on WiFi and next-generation robots such as Softbank's Pepper, announced to become available for order in North America starting 2016, will be based on advanced cloud technologies. In addition, we provide a comprehensive up-to-date survey of results on lowering the delay and increasing the reliability performance of integrated fiber-wireless communications and control infrastructures based on data-centric Ethernet technologies in support of future Tactile Internet applications, including new results on emerging mobile-edge computing (MEC). The reported results are instrumental in providing insights into possible realizations of the Tactile Internet vision.

The remainder of the chapter is structured as follows. The following section further elaborates on the Tactile Internet vision by briefly reviewing its anticipated impact on society and important design guidelines. Then we provide an up-to-date survey of recent progress and enabling technologies proposed for the Tactile Internet. Following that we identify several open challenges and outline future research directions. Finally, we conclude the chapter.

7.2 Tactile Internet: Vision and Design Guidelines

The vision of the Tactile Internet and its potential impact on society is expected to add a new dimension to human-to-machine interaction in a variety of different application fields, including healthcare, education, and smart grid. For a detailed description the interested reader is referred to [46]. The information and communications infrastructure enabling the envisioned Tactile Internet has to meet a number of design requirements. First and foremost, it has to provide a very low end-to-end *latency* of 1 ms and the highest possible *reliability* for real-time response. It also has to ensure both data *security* and the availability and dependability of systems, without violating the very low latency requirement due to additional encryption delays. These key design objectives of the Tactile Internet can be only accomplished by keeping tactile applications local, close to the users, which calls for a distributed (i.e., *decentralized*) service platform architecture based on cloudlets and mobile-edge computing (to be discussed in more detail shortly). Furthermore, *scalable* procedures at all protocol layers are needed to reduce the end-to-end latency from sensors to actuators. Importantly, the Tactile Internet will set demanding requirements for future *access networks* in terms of latency, reliability, and also capacity (e.g., high data rates for video sensors). Wired access networks are partly meeting these requirements already, but wireless access networks are not yet designed to match these needs. According to the ITU-T Technology Watch Report on the Tactile Internet [46], scaling up research in this area will be essential, ushering in new ideas and concepts to boost access networks' inherent redundancy and diversity to address the stringent latency and reliability requirements of Tactile Internet applications.

7.3 Tactile Internet: Recent Progress

We have seen in the previous section that the Tactile Internet will set demanding requirements in particular for the design of future wired and wireless access networks. In [61], we recently introduced our concept of fiber-wireless (FiWi) enhanced LTE-Advanced (LTE-A) heterogeneous networks (HetNets), where the traditional barriers between coverage-centric 4G mobile networks and capacity-centric FiWi broadband access networks based on low-cost data-centric optical fiber and wireless Ethernet technologies are removed. We elaborated on emerging trends and identified important open research challenges to unleash the full potential of FiWi enhanced LTE-A HetNets,

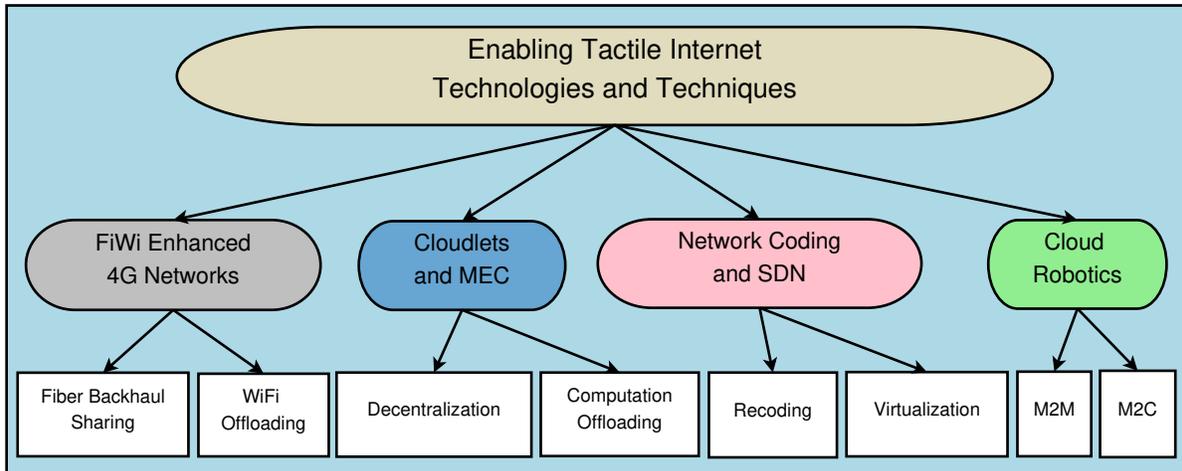


Figure 7.2: Taxonomy of enabling Tactile Internet technologies and techniques.

including their convergence with other technologies and economic sectors for future non-incremental FiWi research. In the future, robots may become parts of our digital-age extended self just as online avatars are today. The adoption rate of low-cost domestic service robots, for example, robotic vacuum cleaners such as iRobot’s *Roomba*, is growing rapidly due to the consumers’ desire to save time spent in unpaid household work. Moreover, inexpensive general-purpose robots such as *Baxter* developed by Rethink Robotics are now able to learn new routines by simply guiding the robot arms through the motions without any need for programming.

More recently, in [30], we elaborated on the role of FiWi access networks for conventional clouds and emerging *cloudlets* (i.e., decentralized entities at the edge of the Internet), thereby highlighting the limitations of traditional radio-over-fiber (RoF) networks to meet the aforementioned trend toward decentralization in future 5G networks. We revisited our early FiWi vision of the year 2008, where we advocated that the focus of access network research should shift from bridging the notorious first/last mile bandwidth bottleneck to the exploitation of distributed storage and processing capabilities, thereby creating unforeseen services and applications that help stimulate innovation, generate revenue, and improve the quality of our every-day lives. Toward this end, we proposed so-called radio-and-fiber (R&F) networks, which are based on decentralized (optical and wireless) Ethernet technologies and perform medium access control (MAC) protocol translation at the optical-wireless interface. Beside protocol translation, the distributed processing and storage capabilities inherently built into R&F networks at the optical-wireless interface may be exploited for a number of additional tasks, for example, cognitive assistance, augmented reality, or face recognition and navigation for cloud robotics. R&F may become the FiWi network type of choice

in the light of future 5G mobile networks moving toward decentralization based on intelligent base stations and cloudlets. In fact, as we shall see shortly, there is a growing desire among industry players to reap the benefits of mobile-cloud convergence by extending today's unmodified cloud to a decentralized two-level cloud-cloudlet architecture based on emerging *mobile-edge computing (MEC)* capabilities.

In the remainder of this section, we provide a more detailed description of FiWi enhanced LTE-A HetNets, cloudlets, and MEC as well as an up-to-date survey on other enabling technologies and techniques proposed for the Tactile Internet according to the taxonomy shown in Fig. 7.2.

FiWi Enhanced LTE-A HetNets

In [29], we investigated the performance gains obtained from unifying coverage-centric 4G mobile networks and capacity-centric FiWi broadband access networks based on data-centric Ethernet technologies with resulting fiber backhaul sharing and WiFi offloading capabilities in response to the unprecedented growth of mobile data traffic. We evaluated the maximum aggregate throughput, offloading efficiency, and in particular the delay performance of FiWi enhanced LTE-A HetNets, including the beneficial impact of various localized fiber-lean backhaul redundancy and wireless protection techniques, by means of probabilistic analysis and verifying simulation. In our study we paid close attention to fiber backhaul reliability issues stemming from fiber faults of an Ethernet passive optical network (EPON) and WiFi offloading limitations due to WiFi mesh node failures as well as temporal and spatial WiFi coverage constraints.

For illustration, Fig. 7.3(a) depicts the average end-to-end delay performance of FiWi enhanced LTE-A HetNets vs. aggregate throughput for different WiFi offloading ratio (WOR) values, whereby $0 \leq \text{WOR} \leq 1$ denotes the percentage of mobile user traffic offloaded onto WiFi. The presented analytical and verifying simulation results were obtained by assuming a realistic LTE-A and FiWi network configuration under uniform traffic loads and applying minimum (optical and wireless) hop routing. For further details the interested reader is referred to [29]. For now, let us assume that the reliability of the EPON is ideal, that is, no fiber backhaul faults occur. However, unlike EPON, the WiFi mesh network may suffer from wireless service outage with probability 10^{-6} . We observe from Fig. 7.3(a) that for increasing WOR the throughput-delay performance of FiWi enhanced LTE-A HetNets is improved significantly. More precisely, by changing WOR from 0.1 to 0.57 the

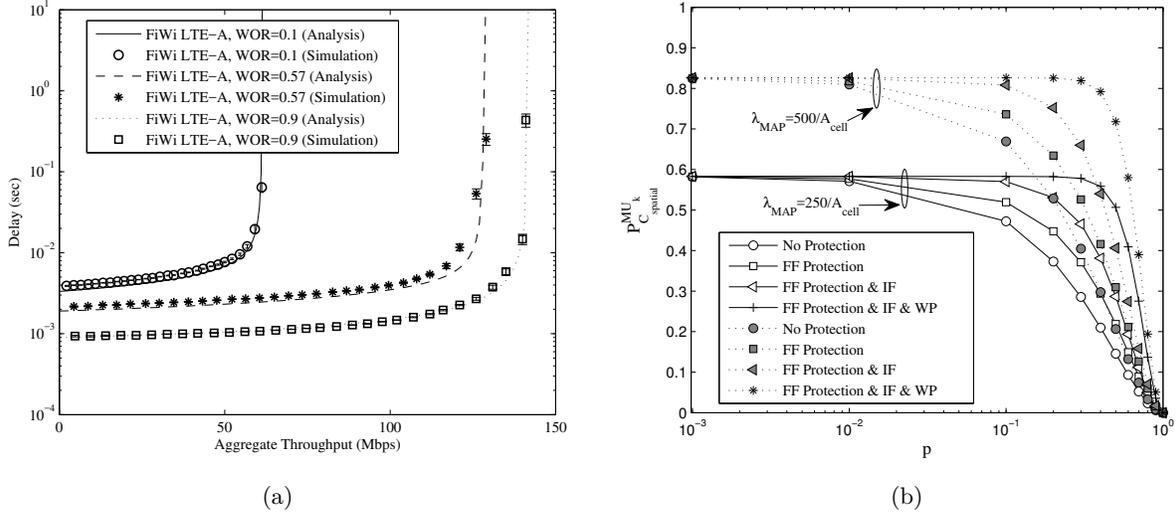


Figure 7.3: FiWi enhanced LTE-A HetNets performance: a) Average end-to-end delay vs. aggregate throughput for different WiFi offloading ratio (WOR); b) FiWi connectivity probability of a mobile user vs. EPON fiber link failure probability p .

maximum achievable aggregate throughput increases from about 61 Mb/s to roughly 126 Mb/s (at an average end-to-end delay of $10^0 = 1$ second), that is, the maximum achievable aggregate throughput has more than doubled. More importantly, further increasing WOR to 0.9 does not result in an additional significant increase of the maximum achievable aggregate throughput, but it is instrumental in decreasing the average end-to-end delay and keeping it at a very low level of 10^{-3} second (1 ms) for a wide range of traffic loads. Thus, this result shows that WiFi offloading the majority of data traffic from 4G mobile networks is a promising approach to obtain a very low latency on the order of 1 ms.

Figure 7.3(b) shows the beneficial impact of the various considered fiber-lean backhaul redundancy (FF: feeder fiber, IF: interconnection fiber between optical network units) and wireless protection (WP) schemes on the FiWi connectivity probability of a mobile user for a conventional two-stage EPON, whereby p denotes its fiber link failure probability. We observe that FF protection in conjunction with IF and WP are able to keep the FiWi connectivity probability of the mobile user essentially flat, though it is lowered when decreasing the density of deployed WiFi mesh access points, λ_{MAP} , from 500 to 250 in a considered cell coverage area of $A_{cell} = 3 \times 3$ km². Figure 7.3(b) clearly shows that the use of FF protection together with IF and WP enables mobile users to be reliably connected to FiWi for an EPON fiber link failure probability p as high as 10^{-1} and beyond, thus demonstrating the ultra-high reliability of mobile user connectivity to the FiWi

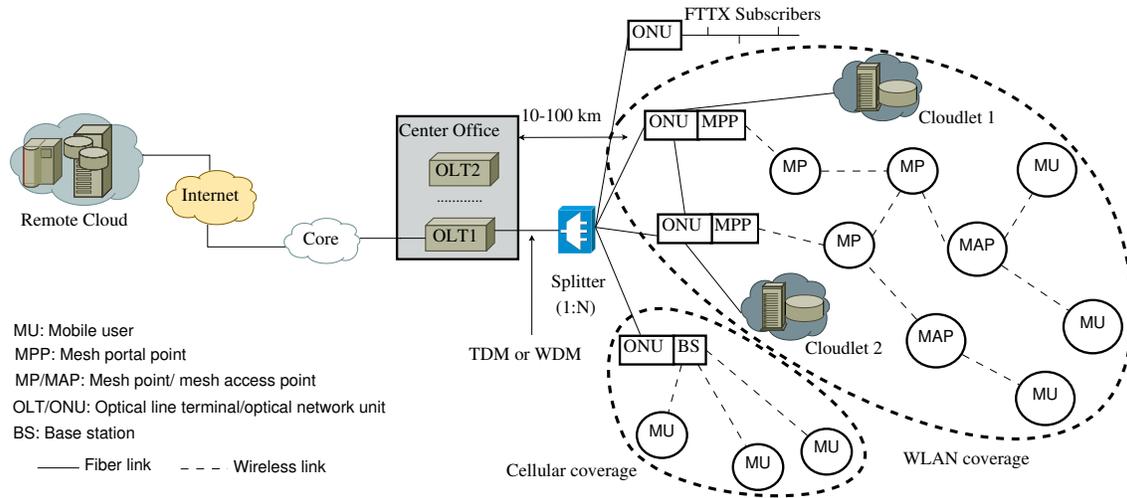


Figure 7.4: Generic architecture of cloudlet enhanced FiWi access networks for mobile-edge computing.

access network, which is key to benefit from the aforementioned WiFi offloading and resultant very low latency performance of FiWi enhanced LTE-A HetNets.

Cloudlets and Mobile-Edge Computing

According to [16], only the concept of locally available cloudlets will enable us to realize the vision of the Tactile Internet. Even at the speed of light (e.g., in optical fiber access networks), 1 ms of round-trip propagation delay requires a cloudlet within 150 km. Cloudlets may be viewed as decentralized proxy cloud servers with processing and storage capabilities, just one or more wireless hops away from the mobile user. Cloudlet research has tended to focus on WiFi in the past, though recently there has been growing interest among cellular network providers. Figure 7.4 illustrates an example of cloudlet enhanced FiWi access networks, where cloudlets may be co-located with WiFi mesh portal points (MPPs) that interface with the optical network units (ONUs) of a shared fiber backhaul, as discussed above.

The importance of cloudlets can be seen in many end-to-end latency-sensitive applications such as augmented reality, real-time cognitive assistance, or face recognition on a mobile device. For instance, to manage and offload high volumes of data, Akamai recently developed the *Edge Redirector Cloudlet*, which is an early example of commercial applications of the cloudlet concept. In September 2014, the so-called *mobile-edge computing (MEC)* industry initiative introduced a reference architecture in order to list challenges that need to be overcome and facilitate the implementation

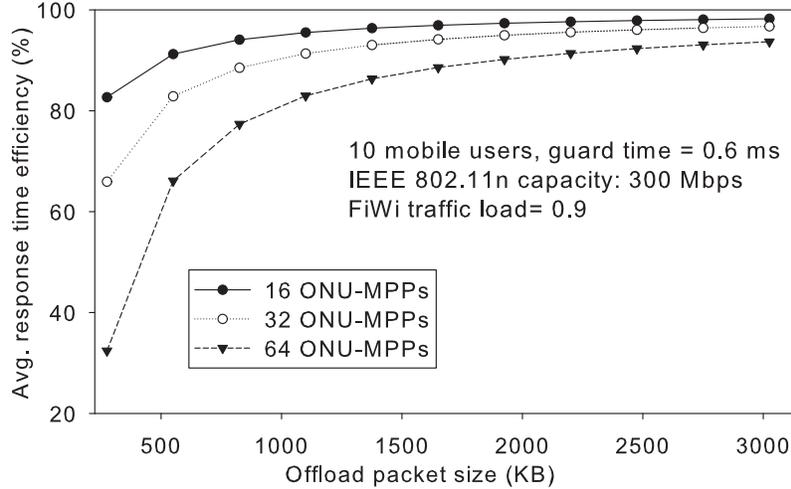


Figure 7.5: Average offload response time efficiency vs. offload packet size.

of MEC [42]. MEC provides IT and cloud computing capabilities within the radio access network (RAN) in close proximity to mobile subscribers. MEC aims at transforming mobile base stations into intelligent service hubs by exploiting proximity, context, agility, and speed in order create a new value chain and stimulate revenue generation.

MEC is expected to enable a wide range of new services and applications. Among others, some important use cases include mobile unified communications, distributed content and DNS caching, RAN-aware content optimization, positioning over LTE (PoLTE), IoT, M2M, video analytics, augmented reality, and optimized local content. It uniquely allows mobile operators, service and content providers, over-the-top (OTT) players, and independent software vendors (ISVs) to tap into local content and real-time information about local access network conditions.

In [30], we elaborated on the deployment of both clouds and cloudlets in FiWi enhanced LTE-A HetNets to increase throughput, reduce end-to-end latency, and improve scalability by means of computation offloading. Recently, we built on this preliminary work by studying the coexistence of conventional broadband and MEC traffic in such a highly converged network. Our obtained results indicate that the use of cloudlets at the edge of FiWi access networks enable us to bring the vision of the Tactile Internet closer to reality by means of MEC, thereby achieving a significantly reduced end-to-end latency and an enhanced overall network performance. For illustration, Fig. 7.5 shows the achievable *average offload response time efficiency* for computation offloading onto cloudlets. In general, computation offloading should be performed if the time required to execute a given task on the mobile device locally is much longer than the response time of offloading the task onto a cloudlet.

This time difference is referred to as offload gain. The average offload response time efficiency is defined as the ratio of offload gain and the response time of tasks that are locally executed on mobile devices. In the following, we assume that the data load of a computation task is fragmented into packets of fixed size and an application is subdivided into a number of fine-grained tasks. Figure 7.5 depicts the achievable offload response time efficiency for different offload packet sizes. We observe that as the offload traffic load increases gradually the overall response time efficiency increases. Figure 7.5 shows that for increasing offload packet sizes the average overall offload time efficiency asymptotically approaches 100 percent. For instance, for a typical case of $N = 16$ and an offload packet size of 1100.60 KB, the average overall offload response time efficiency equals 95.50 percent. This translates into a delay reduction of 95.50 percent compared to the delay obtained in a non-offloading scenario without MEC.

Network Coding and Software Defined Networking

In [218], the authors proposed the integration of network coding and software defined networking (SDN) as a viable approach to meet the Tactile Internet's very low latency requirement. The authors claimed that the extensive use of a more flexible network coding mechanism such as random linear network coding (RLNC) throughout the network can improve the latency performance and reduce the frequency of required packet retransmissions. RLNC is the most general form of network coding, whose main characteristics are recoding and a sliding window based operation. The recoding enables the so-called compute-and-forward approach, where each node in the network resets its coding strategy based on current network conditions for next-hop communication. The complexity of recoding is far simpler than alternative end-to-end (E2E) and hop-by-hop (HbH) coding strategies. This is due to the fact that with E2E coding each relay node needs to store and forward each successfully received packet, whereas with HbH coding each relay node performs full encoding and decoding of all incoming data packets. Conversely, unlike E2E and HbH coding which work on blocks of packets, RLNC applies the sliding window approach, which is beneficial for improving the end-to-end delay performance.

In order to provide deeper insights into achievable latency performance gains and validate their presented theoretical results, the authors implemented a network coding capable software router as a virtual network function (VNF). SDN and virtualization are commonly considered a promising

approach to enable the flexible and automated deployment of VNF in networks. More specifically, the authors used ClickOS NFV platforms and deployed their software router on the Click modular router platform. The RLNC encoder, recoder, and decoder Click elements and fully-fledged compute-and-forward routers were developed by using the Kodo library and its built-in modules, respectively. Furthermore, a prototype was developed by using the extensible service chain prototyping environment (ESCAPE) for the seamless integration of network coding and SDN for a three-hop scenario. The obtained results in [218] show that if the channel is error prone RLNC achieves a lower latency than E2E and HbH coding. Moreover, it was shown that RLNC and HbH coding increase the total number of conveyed packets in the network linearly with the loss probability, whereas E2E coding increases it exponentially. By contrast, if there are no losses RLNC and E2E coding exhibit the same latency performance, while HbH results in an increased latency. The experiments on the compute-and-forward software router verified that RLNC outperforms E2E and HbH coding, offering gains up to 6x and 16x.

Cloud Robotics

Providing robotic services to support daily human activities, especially for the elderly and persons with disabilities, through socially interactive behaviors has been an emerging topic in robotics research, where robotic services consist of systems, devices, and robots that provide the following three functions: sensation, actuation, and control [219]. In this study, the authors presented a cloud-based robotic platform to continuously support human daily activities. Several key technological issues were identified for continuous robotic services such as multi-robot management, multi-area management, user attribute management, and service coordination management. Based on these issues, a cloud robotic based prototype was proposed, referred to as the ubiquitous networked robot platform (UNR-PF), which enables multi-location robotic services via distributed task coordination and control of multiple robots and sensor devices.

To extend the capabilities of both tele-operated and multi-robot based networked robotics, the authors in [17] proposed a cloud robotic system architecture that leverages the combination of an ad-hoc cloud formed by M2M communications among participating robots and an infrastructure cloud enabled by machine-to-cloud (M2C) communications between the robots and the remote cloud, as depicted in Fig. 7.6. M2M communications was used to enable a team of networked robots

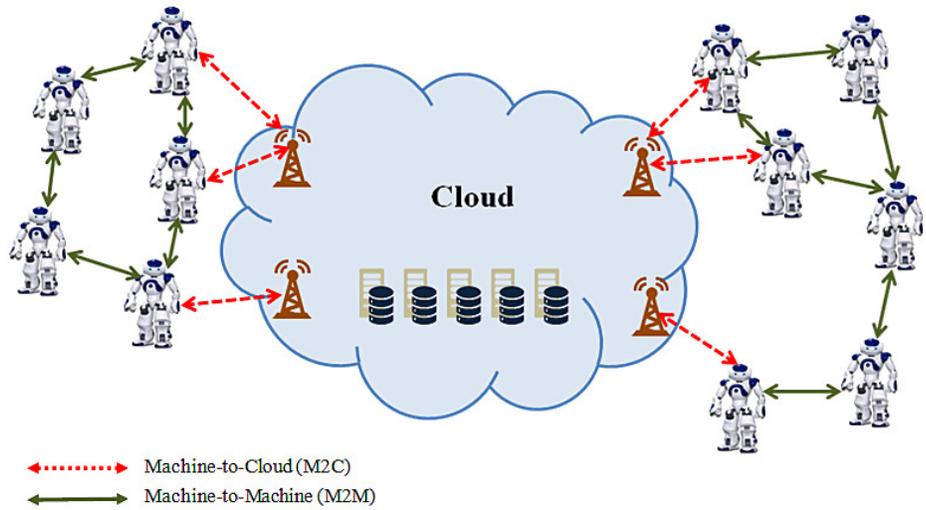


Figure 7.6: Two-tier cloud robotics systems architecture based on M2M and M2C communications.

to complete tasks cooperatively in a distributed fashion by sharing computation/storage resources and exchanging information via the wireless communication network. M2C communications makes it feasible to learn from the shared history of all cloud-enabled robots. Furthermore, the authors proposed the use of gossip routing protocols for the considered two-tier M2M/M2C communications in cloud robotics. The potentially high latencies of distributed routing protocols based on gossip algorithms may be significantly mitigated by using the infrastructure cloud as a central super node for M2M/M2C communications. The authors also developed an optimization framework for task execution strategies that minimize the robots' energy consumption while completing their assigned tasks within a given deadline.

7.4 Open Challenges and Future Research Directions

The Tactile Internet is still in its infancy. A number of open research challenges need to be tackled in order to realize its vision. Beside physical layer issues such as waveform selection and robust modulation schemes, intelligent control and user plane separation and coordination techniques will be vital to reduce signaling overhead and air interface latency. The design of advanced resource management techniques for the support of H2R traffic in R&F based FiWi access networks without degrading network performance is another promising area of future research. Furthermore, highly adaptive network coding techniques along with scalable routing algorithms may play a major role

in providing QoS with enhanced security against malicious activities. Although network coding and SDN hold promise to reduce end-to-end latency in support of the Tactile Internet, further investigations are needed to explore the use of the sliding window approach in multi-path SDN based networks to also improve their throughput and resilience performance. In cloud robotics, the major challenges to be addressed include trust, privacy, security, as well as dependability and safety, given that networked robotic services are not limited to cyberspace but also interact with the physical world. Despite the wide deployment of industrial and service robots, real-time robot applications still suffer from several problems such as inefficiency in service completion. An exciting avenue for future work is the development of collaborative robots with advanced machine learning intelligence to perform collaborative work among truly autonomous distributed humanoid robots. Other important issues for the Tactile Internet include resource management and task allocation schemes (optimal online/offline scheduling), failure handling, mobility of robots, haptic feedback (multimodal or multisensory) based remote robot steering and control applications, as well as flexible service coordination among robots.

The overarching goal of the Tactile Internet should be the production of new goods and services by means of empowering (rather than automating) machines that complement humans rather than substitute for them [216]. Or as Nicholas Carr puts it: relying on computers to fly our planes, find our cancers, design our buildings, audit our businesses is all well and good, but what happens when machines fail and humans have become increasingly deskilled due to automation [220]? In the future, coworking with robots will favor geographical clusters of local production (“inshoring”) and will require human expertise in the coordination of the human-robot symbiosis for the sake of inventing new jobs humans can hardly imagine or did not even know they wanted done. FiWi enabled H2R communications may be a stepping stone to merging mobile Internet, IoT, and advanced robotics with automation of knowledge work and cloud technologies, which together represent the five technologies with the highest estimated potential economic impact in 2025 [61].

7.5 Conclusions

The Tactile Internet will be centered around H2M communications by leveraging devices that enable haptic and tactile sensations. Similar to IoT and 5G, it demands very low latency, ultra-high reliability, H2H/M2M coexistence, integration of data-centric technologies, and security. As the

power of machines increases, the Tactile Internet should help complement humans rather than substitute for them, thus empowering them by providing a growth path based on increased output rather than reduced inputs due to automation. In particular, research on the design of future wired and wireless access networks based on decentralized cloudlets and MEC capabilities will be essential for the coordination of the human-robot symbiosis via FiWi enabled H2R communications. This chapter comprehensively surveyed the recent progress on FiWi enhanced 4G mobile networks, cloudlets, MEC, network coding, SDN, and cloud robotics, with a focus on their significant latency and reliability performance gains.

Chapter 8

Conclusions and Outlook

This chapter summarizes the important contributions of the dissertation and outlines possible directions for future work.

8.1 Conclusions

There is a growing awareness among industry players of reaping the benefits of mobile-cloud convergence by extending today's unmodified cloud to a decentralized two-level cloud-cloudlet architecture based on emerging MEC capabilities. In light of 5G mobile networks moving toward decentralization based on cloudlets, intelligent base stations, and MEC, the inherent distributed processing and storage capabilities of R&F networks can be exploited for new applications. This doctoral thesis is built on FiWi broadband access networks focusing on three major trends, namely, green wired and wireless communications, the integration of clouds and cloudlets in FiWi networks, and the Tactile Internet.

In FiWi access networks, energy efficiency issues must be addressed in a comprehensive fashion that takes into account not only wireless front-end but also optical backhaul segments to extend the battery life of wireless devices and allow operators to reduce their OPEX, while not compromising on QoS. This thesis proposed a novel energy conservation scheme for EPON-WLAN based FiWi access networks (ECO-FiWi) that jointly schedules power-saving modes of wireless stations and access points and optical network units to reduce their energy consumption. ECO-FiWi maximizes

the overall network performance by leveraging TDMA to synchronize the power-saving modes and incorporate them into the DBA process. The proposed ECO-FiWi scheme achieves significant amounts of energy saving (more than 70% in typical scenarios), while limiting upstream delay and incurring manageable delay to downstream traffic.

Different research facets of cloud and cloudlet technologies were explored with a particular focus on workflow scheduling in multi-tenant cloud computing environments and cloud and cloudlet enhanced FiWi access networks. More specifically, Chapter 2 was dedicated to designing a novel four-layered architecture for a workflow scheduling system and a novel CWSA in a multi-tenant cloud computing environment. The obtained results show that the proposed CWSA scheduling outperforms other existing scheduling policies in the literature. Importantly, CWSA was shown to utilize computational resources properly by reducing idle time of cloud resource nodes. Further, a proof-of-concept experiment was conducted by employing real-world scientific workflow applications. The proof-of-concept experiment indicates that the proposed CWSA scheduling policy offers significant improvements for larger workflow applications. Importantly, a key lesson learned from this study is that multi-tenancy helps improve the utilization of resources.

In Chapter 3, we examined the performance of cloudlet enhanced FiWi access networks for MEC based on the integration of EPON, WLAN, and cloudlets. Novel cloudlet-aware resource management algorithms were proposed in order to handle both conventional FiWi and MEC traffic at the same time. A comprehensive analytical framework was developed to evaluate the packet delay performance of both traffic types as well as response time efficiency, offload gain-overhead ratio, energy efficiency, and battery life. The obtained results provide insights into finding the optimal response time efficiency and energy efficiency for a given offload traffic load. For a typical scenario of 32 ONU-APs and an offload packet size of 825.60 KB, the delay is reduced by 86.76% compared to the delay obtained in a non-offload scenario. Further, the envisioned network yields more than 78% of energy savings for edge devices and the battery life of edge devices is extended by up to 7.03 hours in typical scenarios.

Furthermore, the thesis introduced the novel concept of MEC over FiWi networks. Beside several benefits of MEC, a number of interesting research challenges in terms of network integration and coordination, distributed resource management, coexistence of H2H and MEC traffic, cloud and cloudlet coexistence, reliability, and mobility were discussed. Three envisioned design scenarios for

MEC over FiWi networks were studied, followed by a novel unified resource management scheme proposed for MEC over Ethernet-based FiWi networks. The obtained results show the significant benefits of MEC over FiWi networks. For instance, for a typical scenario, a response time efficiency of 95.36% can be achieved. Importantly, a mean MEC packet delay of 20.50 ms is obtained for the considered scenario, while allowing for efficient H2H/MEC coexistence without degrading network performance. Moreover, the battery life of edge devices is prolonged up to 11.30 hours by employing the proposed solution.

In Chapter 6, we studied the coexistence of H2H and cloud services (both centralized and MEC scenarios) in cloudlet enhanced FiWi access networks. A novel two-layer TDMA-based unified resource management scheme was proposed to schedule cloud, MEC, and conventional H2H traffic at the same time. A comprehensive performance analysis was conducted to evaluate packet delay, response time efficiency, and offload gain-overhead ratio. While the proposed network is resilient against fiber cuts, the central OLT forms a single point of failure. To provide survivability in an efficient manner for both FiWi and MEC segments, different protection schemes were proposed and evaluated. The presented solutions help reduce offload packet delay without affecting the network performance for H2H traffic. The obtained results show that for a typical system of 32 ONU-APs and H2H traffic load of 0.6, an MEC and cloud packet delay of 26.73 ms and 126.72 ms can be achieved, respectively, without degrading the network performance for H2H traffic.

Lastly, we investigated an emerging application scenario of cloud-cloudlet enhanced FiWi access networks. In particular, the thesis explored the feasibility of the concept of the Tactile Internet and elaborated on its vision. A taxonomy of enabling Tactile Internet technologies and techniques was presented. In addition, the role of cloudlet/MEC enhanced FiWi access networks in support of the Tactile Internet was investigated in technically greater detail. The obtained results indicate that the use of cloudlets/MEC at the edge of FiWi access networks is instrumental in bringing the vision of the Tactile Internet closer to reality by achieving a significantly reduced end-to-end latency and an enhanced network reliability.

8.2 Outlook

Given the growing progress on and interest in 5G networks, FiWi access networks have great potential to enable the emergence of a variety of novel applications and creating new exciting opportunities. In particular, the integration of cloud and cloudlet in FiWi access networks represents a new topic in communications network research. Consequently, each of the contributions made in this dissertation has opportunities for further improvements. Some of the potential directions for future research that may build upon the work in this dissertation are discussed in the following.

Homogeneous vs. Heterogeneous Machines: Following the discussions in Chapters 5 and 6, the current implementation of the algorithms, models, and host configuration assume homogeneity of all machines in the cloudlet/MEC and the public remote cloud. However, previous research shows that the cloud environment is mostly a heterogeneous environment with machines of the same category having different capabilities. In a similar way, a cloudlet/MEC may be deployed using multiple servers with a certain heterogeneity. As part of the future work, the proposed models and results can be extended taking into account machines with heterogeneous capabilities.

Fast Moving vs. Quasi-Stationary Users: Following the discussions in Chapters 5 and 6, the proposed models and algorithms do not take mobility into account. It is evident that computation offloading onto cloudlets is challenging because of the intermittent connection between cloudlets and mobile users due to user mobility. An interesting extension of the proposed solutions would be to incorporate a broad range of realistic mobility scenarios with connectivity predictions. More specifically, the impact of fast moving and quasi-stationary users on the performance of cloud-cloudlet (or MEC) enhanced FiWi access networks would be an interesting topic.

Scalable Unified Resource Management for Fog Computing: This dissertation has explored three key aspects of the integration of cloud and cloudlets in FiWi access networks: unified resource management, architecture, and remote or surrogate execution. Many other possible research directions exist, notably, the most promising is *Fog computing over FiWi access networks*. Recall that Chapter 6 explored the unified resource management for managing the cloud and cloudlet integration. However, in the context of massive connectivity of edge devices, the proposed solution might need to be revisited for Fog computing, where scalability is one of the major issues.

For instance, one possible solution would be the implementation of a hybrid access mechanism by enhancing the proposed layered-TDMA approach with polling.

FiWi Enabled Human-to-Robot Communications: As MEC over FiWi is quite a new concept introduced in this work, there are a number of areas in which it could play a role as an enabler. For example, as explored in [61], the so-called *FiWi enabled Human-to-Robot Communications (H2R)* concept could play an important role in the future. Recall that Chapter 5 explored MEC empowered FiWi in the 5G era, which might be one way of realizing FiWi based H2R communications. Since the tasks performed by humans and robots are generally different, the involved tasks should be categorized accordingly. Machine learning plays a vital role in H2R communications. For example, monitoring, supervision, and recovery are crucial capabilities for controlling H2R communications. Machine learning helps automatize those capabilities, particularly non-verbal communications and imitation learning mechanisms for robots [221]. Since machine learning requires significant resources (CPU, RAM), such resource-intensive functionalities can be offloaded onto cloudlet/MEC empowered FiWi access networks. To do this, one could improve the unified resource management strategies explored in Chapters 5 and 6 by incorporating specific characteristics of robot traffic.

References

- [1] G. Kramer and G. Pesavento. Ethernet passive optical network (EPON): Building a next-generation optical access network. *IEEE Communications Magazine*, 40(2):66–73, Feb. 2002.
- [2] G. Kramer, M. De Andrade, R. Roy, and P. Chowdhury. Evolution of optical access networks: Architectures and capacity upgrades. *Proceedings of the IEEE*, 100(5):1188–1196, May 2012.
- [3] F. Effenberger, D. Cleary, O. Haran, G. Kramer, R. D. Li, M. Oron, and T. Pfeiffer. An introduction to PON technologies. *IEEE Communications Magazine*, 45(3):S17–S25, Mar. 2007.
- [4] M. Maier. The escape of sisyphus or what post ng-pon2 should do apart from neverending capacity upgrades. *Photonics*, 1(1):47–66, Mar. 2014.
- [5] K. Smith and Y. R. Zhou. Optical transmission innovation for long term broadband internet growth. *IEEE ComSoc Technology News (CTN) [Online]*: <http://www.comsoc.org/ctn/strengthening-backbone-5g-and-beyond>, Aug. 2016.
- [6] M. S. Gast. *802.11 Wireless Networks: The Definitive Guide*. O’Reilly Media, Inc., second edition, May 2005.
- [7] D. Astely, E. Dahlman, A. Furuskär, Y. Jading, M. Lindström, and S. Parkvall. LTE: The evolution of mobile broadband. *IEEE Communications Magazine*, 47(4):44–51, Apr. 2009.
- [8] S. Parkvall, A. Furuskär, and E. Dahlman. Evolution of LTE toward IMT-Advanced. *IEEE Communications Magazine*, 49(2):84–91, Feb. 2011.
- [9] S. Cherry. Telecom: Edholm’s law of bandwidth. *IEEE Spectrum*, 41(7):58–60, July 2004.
- [10] M. Martin, N. Ghazisaidi, and M. Reisslein. The audacity of fiber-wireless (FiWi) networks (invited paper). In *Proc., Third International Conference on Access Networks (AccessNets)*, pages 1–10, Oct. 2008.
- [11] S. Aissa and M. Maier. Towards seamless fiber-wireless (FiWi) access networks: Convergence and challenges. In *Proc., International Conference on Transparent Optical Networks “Mediterranean Winter” (ICTON-MW ’07)*, pages 1–6, Dec. 2007.
- [12] D. C. Kilper, G. Atkinson, S. K. Korotky, S. Goyal, P. Vetter, D. Suvakovic, and O. Blume. Power trends in communication networks. *IEEE Journal of Selected Topics in Quantum Electronics*, 17(2):275–284, Mar. 2011.

- [13] B. P. Rimal and E. Choi. A service-oriented taxonomical spectrum, cloudy challenges and opportunities of cloud computing. *International Journal of Communication Systems*, 25(6): 796–819, June 2012.
- [14] N. Fernando, S. W. Loke, and W. Rahayu. Mobile cloud computing: A survey. *Future Generation Computer Systems*, 29(1):84–106, Jan. 2013.
- [15] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies. The case for VM-based cloudlets in mobile computing. *IEEE Pervasive Computing*, 8(4):14–23, Oct. 2009.
- [16] M. Satyanarayanan, R. Schuster, M. Ebling, G. Fettweis, H. Flinck, K. Joshi, and K. Sabnani. An open ecosystem for mobile-cloud convergence. *IEEE Communications Magazine*, 53(3): 63–70, Mar. 2015.
- [17] G. Hu, W. P. Tay, and Y. Wen. Cloud robotics: Architecture, challenges and applications. *IEEE Network*, 26(3):21–28, May/June 2012.
- [18] W. Hu, Y. Gao, K. Ha, J. Wang, B. Amos, Z. Chen, P. Pillai, and M. Satyanarayanan. Quantifying the impact of edge computing on mobile applications. In *Proc., 7th ACM SIGOPS Asia-Pacific Workshop on Systems, APSys '16*, pages 5:1–5:8, 2016.
- [19] M. Maier and M. Lévesque. Dependable fiber-wireless (FiWi) access networks and their role in a sustainable third industrial revolution economy. *IEEE Transactions on Reliability*, 63(2): 386–400, June 2014.
- [20] S. Tombaz, P. Monti, F. Farias, M. Fiorani, L. Wosinska, and J. Zander. Is backhaul becoming a bottleneck for green wireless access networks? In *Proc., IEEE Int. Conf. Commun. (ICC)*, pages 4029–4035, June 2014.
- [21] L. Valcarenghi, D Pham Van, Pier Giorgio Raponi, Piero Castoldi, D.R. Campelo, S. Wong, S. Yen, Leonid G. Kazovsky, and S. Yamashita. Energy efficiency in passive optical networks: where, when, and how? *IEEE Network*, 26(6):61–68, Nov. 2012.
- [22] B. P. Rimal and M. A. El-Refaey. A framework of scientific workflow management systems for multi-tenant cloud orchestration environment. In *Proc., IEEE WETICE*, pages 88–93, June 2010.
- [23] C. J. Guo, W. Sun, Y. Huang, Z. H. Wang, and B. Gao. A framework for native multi-tenancy application development and management. In *Proc., IEEE CEC/EEE*, pages 551–558, July 2007.
- [24] C. D. Weissman and S. Bobrowski. The design of the force.com multitenant internet application development platform. In *Proc., ACM SIGMOD*, pages 889–896, June 2009.
- [25] H. Ballani, P. Costa, T. Karagiannis, and A. Rowstron. Towards predictable datacenter networks. In *Proc., ACM SIGCOMM*, pages 242–253, Aug. 2011.
- [26] ETSI Industry Specification Group (ISG). Mobile-edge computing – A key technology towards 5G. *White Paper, no. 11*, pages 1–16, Sept. 2015.
- [27] ETSI Industry Specification Group (ISG). Mobile-edge computing (MEC); technical requirements, group specification. *ETSI GS MEC 002, v1.1.1*, pages 1–40, Mar. 2016.

- [28] G. Fettweis and S. Alamouti. 5G: Personal mobile Internet beyond what cellular did to telephony. *IEEE Communications Magazine*, 52(2):140–145, Feb. 2014.
- [29] H. Beyranvand, M. Lévesque, M. Maier, and J. A. Salehi. FiWi enhanced LTE-A HetNets with unreliable fiber backhaul sharing and WiFi offloading. In *Proc., IEEE INFOCOM*, pages 1275–1283, Apr. 2015.
- [30] M. Maier and B. P. Rimal. The audacity of Fiber-Wireless (FiWi) networks: Revisited for clouds and cloudlets (invited paper). *China Communications*, 12(8):33–45, Aug. 2015.
- [31] H. Chang, A. Hari, S. Mukherjee, and T. V. Lakshman. Bringing the cloud to the edge. In *Proc., IEEE INFOCOM Wkshps*, pages 346–351, Apr. 2014.
- [32] G. Fettweis and S. Alamouti. 5G: Personal mobile internet beyond what cellular did to telephony. *IEEE Communications Magazine*, 52(2):140–145, Feb. 2014.
- [33] M. Maier. Survivability techniques for NG-PONs and FiWi access networks. In *Proc., IEEE ICC*, pages 6214–6219, June 2012.
- [34] B. Kantarci and H. T. Mouftah. Availability and cost-constrained long-reach passive optical network planning. *IEEE Transactions on Reliability*, 61(1):113–124, Mar. 2012.
- [35] S. Neumayer, G. Zussman, R. Cohen, and E. Modiano. Assessing the vulnerability of the fiber infrastructure to disasters. *IEEE/ACM Transactions on Networking*, 19(6):1610–1623, Dec. 2011.
- [36] S. Sarkar, H. H. Yen, S. Dixit, and B. Mukherjee. A novel delay-aware routing algorithm (DARA) for a hybrid wireless-optical broadband access network (WOBAN). *IEEE Network*, 22(3):20–28, May 2008.
- [37] P. Chowdhury, B. Mukherjee, S. Sarkar, G. Kramer, and S. Dixit. Hybrid wireless-optical broadband access network (WOBAN): Prototype development and research challenges. *IEEE Network*, 23(3):41–48, May 2009.
- [38] P. Djukic and P. Mohapatra. Soft-TDMAC: A software TDMA-based MAC over commodity 802.11 hardware. In *Proc., IEEE INFOCOM*, pages 1836–1844, Apr. 2009.
- [39] V. Sevani, B. Raman, and P. Joshi. Implementation-based evaluation of a full-fledged multihop TDMA-MAC for WiFi mesh networks. *IEEE Transactions on Mobile Computing*, 13(2):392–406, Feb 2014.
- [40] D. Pham Van, B. P. Rimal, M. Maier, and L. Valcarenghi. Design, analysis, and hardware emulation of a novel energy conservation scheme for sensor enhanced FiWi networks (ECO-SFiWi). *IEEE Journal on Selected Areas in Communications*, 34(5):1645–1662, May 2016.
- [41] S. Sarkar, H. H. Yen, S. Dixit, and B. Mukherjee. Hybrid wireless-optical broadband access network (WOBAN): Network planning using Lagrangean relaxation. *IEEE/ACM Transactions on Networking*, 17(4):1094–1105, Aug. 2009.
- [42] ETSI Industry Specification Group (ISG). Mobile-edge computing - introductory technical white paper. In *ETSI*, pages 1–36, Sept. 2014.
- [43] M. Maier. Un âge d’or économique dont le numérique est la trame de fond. *Webzine PlanèteINRS.ca [Online]*, Dec 2013.

- [44] G. P. Fettweis. The Tactile Internet: Applications and challenges. *IEEE Vehicular Technology Magazine*, 9(1):64–70, Mar. 2014.
- [45] M. Dohler and G. Fettweis. The Tactile Internet – IoT, 5G and cloud on steroids. *Telefonica Blog*, 30 Oct. 2014.
- [46] ITU-T. The Tactile Internet. *ITU-T Technology Watch Report*, pages 1–24, Aug. 2014.
- [47] ITU-T. 40-Gigabit-capable passive optical networks (NG-PON2): General requirements. *ITU-T G.989.1 Recommendation*, pages 1–26, Mar. 2013.
- [48] D. Nasset. NG-PON2 technology and standards. *IEEE/OSA Journal of Lightwave Technology*, 33(5):1136–1143, Mar. 2015.
- [49] C. F Lam. Fiber to the home: Getting beyond 10 Gb/s. *OSA Optics and Photonics News*, 27(3):22–29, Mar. 2016.
- [50] IEEE 802.3 Next Generation Ethernet Passive Optical Network (NG-EPON) Study Group. URL <http://www.ieee802.org/3/NGEPONSG/>. Accessed on Oct. 25, 2016.
- [51] CPRI. Common public radio interface (CPRI); interface specification, v 7.0. pages 1–128, Oct. 2015.
- [52] N. J. Gomes, P. Chanclou, P. Turnbull, A. Magee, and V. Jungnickel. Fronthaul evolution: From CPRI to Ethernet (invited paper). *Optical Fiber Technology*, 26:50–58, Dec. 2015.
- [53] China Mobile Research Institute. C-RAN: The road towards green RAN. *White Paper*, Oct. 2011.
- [54] N. Shibata, T. Tashiro, S. Kuwano, N. Yuki, Y. Fukada, J. Terada, and A. Otaka. Performance evaluation of mobile front-haul employing Ethernet-based TDM-PON with IQ data compression [invited]. *IEEE/OSA Journal of Optical Communications and Networking*, 7(11):B16–B22, Nov. 2015.
- [55] L. L. Van-Etter. Design and installation challenges and solutions for passive optical lans. *White Paper, 3M Communication Markets Division*, pages 1–9, 2015.
- [56] Nokia Passive Optical LAN. URL <https://networks.nokia.com/solutions/passive-optical-lan>. Accessed on Aug. 25, 2016.
- [57] Network Strategy Partners, LLC (NSP). Transformation of the enterprise network using passive optical LAN. *White Paper*, pages 1–13, May 2009.
- [58] T. Koonen. Fiber to the home/fiber to the premises: What, where, and when? *Proceedings of the IEEE*, 94(5):911–934, May 2006.
- [59] NTT Corporation. Flexible access system architecture (FASA). *White Paper, Ver. 1.0*, pages 1–29, June 2016.
- [60] M. Maier and N. Ghazisaidi. *FiWi Access Networks*. Cambridge University Press, 2012.
- [61] M. Maier. FiWi access networks: Future research challenges and moonshot perspectives. In *Proc., IEEE ICC, Workshop on Fiber-Wireless Integrated Technologies, Systems and Networks*, pages 371–375, June 2014.

- [62] Fiber-Wireless Integration Technical Subcommittee. URL <http://ifwt.committees.comsoc.org/>. Accessed on Sept. 20, 2016.
- [63] L. Kazovsky, S.-W. Wong, T. Ayhan, K.M. Albeyoglu, M. R. N. Ribeiro, and A. Shastri. Hybrid optical-wireless access networks. *Proceedings of the IEEE*, 100(5):1197–1225, May 2012.
- [64] B. P. Rimal and M. Maier. Workflow scheduling in multi-tenant cloud computing environments. *IEEE Transactions on Parallel and Distributed Systems*, 28(1):290–304, Jan. 2017.
- [65] N. Ghazisaidi and M. Maier. Fiber-Wireless (FiWi) access networks: Challenges and opportunities. *IEEE Network*, 25(1):36–42, Jan./Feb. 2011.
- [66] Aptilo Networks. URL <http://www.anpdm.com/newsletterweb/44475F447248435D4474424A59/494A514279494B5B4173414445E43>. Accessed on Dec. 5, 2016.
- [67] IEEE 802.11 Task Group AX. Status of project IEEE 802.11ax high efficiency WLAN (HEW). URL http://www.ieee802.org/11/Reports/tgax_update.htm. Accessed on Dec. 9, 2016.
- [68] B. Bellalta. IEEE 802.11ax: High-efficiency WLANs. *IEEE Wireless Communications*, 23(1):38–46, Feb. 2016.
- [69] M. Décina. Future of networks. In *Proc. IEEE Technology Time Machine (TTM)*, Oct. 2014.
- [70] J. G. Andrews. Seven ways that HetNets are a cellular paradigm shift. *IEEE Communications Magazine*, 51(3):136–144, Mar. 2013.
- [71] P. Bhat, S. Nagata, L. Campoy, I. Berberana, T. Derham, G. Liu, X. Shen, P. Zong, and J. Yang. Lte-advanced: An operator perspective. *IEEE Communications Magazine*, 50(2):104–114, Feb. 2012.
- [72] H. Beyranvand, M. Lévesque, M. Maier, J. A. Salehi, C. Verikoukis, and D. Tipper. Toward 5G: FiWi enhanced LTE-A HetNets with reliable low-latency fiber backhaul sharing and WiFi offloading. *IEEE/ACM Transactions on Networking*, 25(2):690–707, Apr. 2017.
- [73] D. Pham Van, B. P. Rimal, and M. Maier. Fiber optic vs. wireless sensors in energy-efficient integrated FiWi smart grid networks: An energy-delay and TCO comparison. In *Proc., IEEE INFOCOM*, pages 1–9, Apr. 2016.
- [74] D. Pham Van, B. P. Rimal, J. Chen, P. Monti, L. Wosinska, and M. Maier. Power-saving methods for internet of things over converged fiber-wireless access networks. *IEEE Communications Magazine*, 54(11):166–175, Nov. 2016.
- [75] D. Pham. Van, B. P. Rimal, and M. Maier. Power-saving scheme for PON LTE-A converged networks supporting M2M communications. In *2015 IEEE International Conference on Ubiquitous Wireless Broadband (ICUWB)*, pages 1–5, Oct. 2015.
- [76] D. Pham Van, B. P. Rimal, S. Andreev, T. Tirronen, and M. Maier. Machine-to-machine communications over FiWi enhanced LTE networks: A power-saving framework and end-to-end performance. *IEEE/OSA Journal of Lightwave Technology*, 34(4):1062–1071, Feb. 2016.

- [77] J. Manyika, M. Chui, J. Bughin, R. Dobbs, P. Bisson, and A. Marrs. Disruptive technologies: Advances that will transform life, business, and the global economy. *McKinsey Global Institute, Report*, pages 1–176, May 2013.
- [78] Nokia Siemens Networks. Liquid applications system description. 1(DN09139526):1–40, July 2014.
- [79] D. McGrath and A. Banathy. Introduction to the micro data center (mdc). *White Paper, Panduit Corp., WW-CPWP-16*, pages 1–10, Apr. 2012.
- [80] Cisco Systems Inc. Cisco IOx. URL <https://developer.cisco.com/site/iox/>. Accessed on Oct. 20, 2016.
- [81] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli. Fog computing and its role in the Internet of Things. In *Proc., First Edition of the MCC Workshop on Mobile Cloud Computing*, pages 13–16, Aug. 2012.
- [82] B. Han, P. Hui, V. S. A. Kumar, M. V. Marathe, J. Shao, and A. Srinivasan. Mobile data offloading through opportunistic communications and social participation. *IEEE Transactions on Mobile Computing*, 11(5):821–834, May 2012.
- [83] K. Lee, J. Lee, Y. Yi, I. Rhee, and S. Chong. Mobile data offloading: How much can WiFi deliver? *IEEE/ACM Transactions on Networking*, 21(2):536–550, Apr. 2013.
- [84] F. Rebecchi, M. Dias de Amorim, V. Conan, A. Passarella, R. Bruno, and M. Conti. Data offloading techniques in cellular networks: A survey. *IEEE Communications Surveys & Tutorials*, 17(2):580–603, May 2015.
- [85] F. Mehmeti and T. Spyropoulos. Performance modeling, analysis, and optimization of delayed mobile data offloading for mobile users. *IEEE/ACM Transactions on Networking*, 25(1):550–564, Feb 2017.
- [86] 3GPP TS 24.312. Access Network Discovery and Selection Function (ANDSF), Management Object (MO). pages 1–55, V13.3.0, June 2016.
- [87] 3GPP TR 23.829. Technical specification group services and system aspects; local IP access and selected IP traffic offload (LIPA-SIPTO). pages 1–43, Oct. 2011, Rel. 10, V10.0.1.
- [88] 3GPP TS 23.261. LTE: IP flow mobility and seamless wireless local area network (WLAN) offload; stage 2. pages 1–24, May 2011, Rel. 10, V10.1.0.
- [89] H. Soliman. Mobile IPv6 support for dual stack hosts and routers. *Internet Engineering Task Force (IETF) Request For Comments (RFC): 5555*, pages 1–41, June 2009.
- [90] C. B. Sankaran. Data offloading techniques in 3GPP Rel-10 networks: A tutorial. *IEEE Communications Magazine*, 50(6):46–53, June 2012.
- [91] R. Wolski, S. Gurun, C. Krintz, and D. Nurmi. Using bandwidth data to make computation offloading decisions. In *Proc., IEEE International Symposium on Parallel and Distributed Processing*, pages 1–8, Apr. 2008.
- [92] M. Satyanarayanan. Pervasive computing: Vision and challenges. *IEEE Personal Communications*, 8(4):10–17, Aug. 2001.

- [93] R. Balan, J. Flinn, M. Satyanarayanan, S. Sinnamohideen, and H.-I Yang. The case for cyber foraging. In *Proc., ACM SIGOPS European Workshop*, pages 87–92, July 2002.
- [94] R. K. Balan, D. Gergle, M. Satyanarayanan, and J. Herbsleb. Simplifying cyber foraging for mobile devices. In *Proc., 5th International Conference on Mobile Systems, Applications and Services (MobiSys)*, pages 272–285, June 2007.
- [95] D. P. Anderson, J. Cobb, E. Korpela, M. Lebofsky, and D. Werthimer. Seti@home: An experiment in public-resource computing. *Communications of the ACM*, 45(11):56–61, Nov. 2002.
- [96] K. Kumar, J. Liu, Y.-H. Lu, and B. Bhargava. A survey of computation offloading for mobile systems. *Mobile Networks and Applications*, 18(1):129–140, 2013.
- [97] E. Cuervo, A. Balasubramanian, D.-Ki Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl. MAUI: Making smartphones last longer with code offload. In *Proc., ACM MobiSys*, pages 49–62, June 2010.
- [98] B.-G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti. CloneCloud: Elastic execution between mobile device and cloud. In *Proc., ACM EuroSys*, pages 301–314, Apr. 2011.
- [99] S. Kosta, A. Aucinas, P. Hui, R. Mortier, and X. Zhang. ThinkAir: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading. In *Proc., IEEE INFOCOM*, pages 945–953, Mar. 2012.
- [100] M.-R. Ra, A. Sheth, L. Mummert, P. Pillai, D. Wetherall, and R. Govindan. Odessa: Enabling interactive perception applications on mobile devices. In *Proc., 9th international conference on Mobile systems, applications, and services*, MobiSys ’11, pages 43–56, June 2011.
- [101] J. Flinn, S. Park, and M. Satyanarayanan. Balancing performance, energy, and quality in pervasive computing. In *Proc., 22nd International Conference on Distributed Computing Systems*, ICDCS ’02, pages 217–226, July 2002.
- [102] K. Yang, S. Ou, and H. H. Chen. On effective offloading services for resource-constrained mobile devices running heavier mobile internet applications. *IEEE Communications Magazine*, 46(1):56–63, Jan. 2008.
- [103] R. Kemp, N. Palmer, T. Kielmann, and H. Bal. Cuckoo: A computation offloading framework for smartphones. In *Proc., 2nd International Conference on Mobile Computing, Applications, and Services (MobiCASE)*, pages 59–79, Oct. 2010.
- [104] Y.-Y. Su and J. Flinn. Slingshot: Deploying stateful services in wireless hotspots. In *Proc., 3rd International Conference on Mobile Systems, Applications, and Services*, MobiSys ’05, pages 79–92, June 2005.
- [105] K. Ha, P. Pillai, W. Richter, Y. Abe, and M. Satyanarayanan. Just-in-time provisioning for cyber foraging. In *Proc., 11th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys ’13, pages 153–166, June 2013.
- [106] D. Pham Van, B. P. Rimal, M. Maier, and L. Valcarenghi. ECO-FiWi: An energy conservation scheme for integrated fiber-wireless access networks. *IEEE Transactions on Wireless Communications*, 15(6):3979–3994, June 2016.

- [107] B. P. Rimal, D. Pham Van, and M. Maier. Cloudlet enhanced fiber-wireless access networks for mobile-edge computing. *IEEE Transactions on Wireless Communications*, IEEE Xplore Early Access, DOI: 10.1109/TWC.2017.2685578.
- [108] B. P. Rimal, D. Pham Van, and M. Maier. Mobile-edge computing empowered fiber-wireless access networks in the 5G era. *IEEE Communications Magazine*, 55(2):192–200, 2017.
- [109] B. P. Rimal, D. Pham Van, and M. Maier. Mobile-edge computing vs. centralized cloud computing in fiber-wireless access networks. In *Proc., IEEE INFOCOM, Workshop on 5G & Beyond - Enabling Technologies and Applications*, pages 991–996, Apr. 2016.
- [110] B. P. Rimal, D. Pham Van, and M. Maier. Mobile-edge computing vs. centralized cloud computing over a converged fiber-wireless broadband access network. *IEEE Transactions on Network and Service Management*, in revision.
- [111] M. Maier, M. Chowdhury, B. P. Rimal, and D. Pham Van. The Tactile Internet: Vision, recent progress, and open challenges. *IEEE Communications Magazine*, 54(5):138–145, May 2016.
- [112] J. G. Andrews, S. Buzzi, Wan Choi, S. V. Hanly, A. Lozano, A. C. K. Soong, and J. C. Zhang. What will 5G be? *IEEE Journal on Selected Areas in Communications*, 32(6):1065–1082, June 2014.
- [113] H. Beyranvand, W. Lim, M. Maier, C. Verikoukis, and J. A. Salehi. Backhaul-aware user association in FiWi enhanced LTE-A heterogeneous networks. *IEEE Transactions on Wireless Communications*, 14(6):2992–3003, June 2015.
- [114] F. Aurzada, M. Lévesque, M. Maier, and M. Reisslein. FiWi access networks based on next-generation PON and Gigabit-class WLAN technologies: A capacity and delay analysis. *IEEE/ACM Transactions on Networking*, 22(4):1176–1189, Aug. 2014.
- [115] S.-L Tsao and C.-H. Huang. A survey of energy efficient MAC protocols for IEEE 802.11 WLAN. *Computer Communications*, 34(1):54–67, Jan. 2011.
- [116] A. M. Kholaf, T. D. Todd, P. Koutsakis, and M. N. Smadi. QoS-enabled power saving access points for IEEE 802.11e networks. In *Proc., IEEE Wireless Commun. Netw. Conf. (WCNC)*, pages 2331–2336, Mar. 2008.
- [117] P. Chowdhury, M. Tornatore, S. Sarkar, and B. Mukherjee. Building a green wireless-optical broadband access network (WOBAN). *IEEE/OSA Journal of Lightwave Technology*, 28(16): 2219–2229, Aug. 2010.
- [118] B. Kantarci and H. T. Mouftah. Energy efficiency in the extended-reach fiber-wireless access networks. *IEEE Network*, 26(2):28–35, Mar. 2012.
- [119] K. Miyanabe, H. Nishiyama, N. Kato, H. Ujikawa, K.-I. Suzuki, and N. Yoshimoto. Synchronized power saving mechanisms for battery-powered mobile terminals in smart FiWi networks. In *Proc., IEEE Veh. Conf. Technol. (VTC)*, pages 1–5, Sept. 2014.
- [120] A. Barradas, N. Correia, J. Coimbra, and G. Schütz. Load adaptive and fault tolerant framework for energy saving in fiber-wireless access networks. *IEEE/OSA Journal of Optical Communications and Networking*, 5(9):957–967, Sept. 2013.

- [121] J. Coimbra, G. Schütz, and N. Correia. Energy efficient routing algorithm for fiber-wireless access networks: A network formation game approach. *Computer Network*, 60:201–216, Feb. 2014.
- [122] M. Ahmed, I. Ahmad, and D. Habibi. Service class resource management for green wireless-optical broadband access networks (WOBAN). *IEEE/OSA Journal of Lightwave Technology*, 33(1):7–18, Jan. 2015.
- [123] R. P. Liu, G. J. Sutton, and I. B. Collings. WLAN power save with offset listen interval for machine-to-machine communications. *IEEE Transactions on Wireless Communications*, 13(5):2552–2562, May 2014.
- [124] D. Pham Van, L. Valcarenghi, M. P. I. Dias, K. Kondepu, P. Castoldi, and E. Wong. Energy-saving framework for passive optical networks with ONU sleep/doze mode. *OSA Optics Express*, 23(3):A1–A14, Feb. 2015.
- [125] D. Pham Van, L. Valcarenghi, M. Chincoli, and P. Castoldi. Experimental evaluation of a sleep-aware dynamic bandwidth allocation in a multi-ONU 10G-EPON testbed. *Optical Switching and Networking*, 14, Part 1:11–24, Aug. 2014.
- [126] G. Kramer, B. Mukherjee, and G. Pesavento. IPACT: A dynamic protocol for an Ethernet PON (EPON). *IEEE Communications Magazine*, 40(2):74–80, Feb. 2002.
- [127] M. Bokhari and P. Saengudomlert. Analysis of mean packet delay for upstream transmissions in passive optical networks with sleep mode. *Optical Switching and Networking*, 10(3):195–210, July 2013.
- [128] S. Bharati and P. Saengudomlert. Analysis of mean packet delay for dynamic bandwidth allocation algorithms in EPONs. *IEEE/OSA Journal of Lightwave Technology*, 28(23):3454–3462, Dec. 2010.
- [129] R. Zheng, J. C. Hou, and L. Sha. Performance analysis of power management policies in wireless networks. *IEEE Transactions on Wireless Communications*, 5(6):1351–1361, June 2006.
- [130] H. Lei and A. A. Nilsson. Queuing analysis of power management in the IEEE 802.11 based wireless LANs. *IEEE Transactions on Wireless Communications*, 6(4):1286–1294, Apr. 2007.
- [131] Y.-H. Zhu and V. C. M. Leung. Efficient power management for infrastructure IEEE 802.11 WLANs. *IEEE Transactions on Wireless Communications*, 9(7):2196–2205, July 2010.
- [132] D. P Bertsekas, R. G. Gallager, and P. Humblet. *Data Networks*. Prentice-Hall International New Jersey, second edition, Jan. 1992.
- [133] J. A. Hernandez, I. Seoane, R. Romeral, and D. Larrabeiti. A note on the potential energy savings by extending the average cycle times in passive optical networks. In *Proc., IEEE 17th Eur. Conf. Netw. Opt. Commun. (NOC)*, pages 1–4, June 2012.
- [134] L. M. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner. A break in the clouds: towards a cloud definition. *SIGCOMM Computer Communication Review*, 39(1):50–55, Jan. 2009.
- [135] I. Foster, Yong Zhao, I. Raicu, and Shiyong Lu. Cloud computing and grid computing 360-degree compared. In *Proc., IEEE Grid Comput. Environments Wksp*, pages 1–10, Nov. 2008.

- [136] D. Fernandez-Baca. Allocating modules to processors in a distributed system. *IEEE Transactions on Software Engineering*, 15(11):1427–1436, Nov. 1989.
- [137] J. Yu, R. Buyya, and K. Ramamohanarao. Workflow scheduling algorithms for grid computing. In *Metaheuristics for Scheduling in Distributed Computing Environments*, pages 173–214. Aug. 2008.
- [138] Y. Zhao, X. Fei, I. Raicu, and S. Lu. Opportunities and challenges in running scientific workflows on the cloud. In *Proc., IEEE Cyber-Enabled Distrib. Comput. and Knowledge Discovery*, pages 455–462, Oct. 2011.
- [139] Y. Gil, E. Deelman, M. Ellisman, T. Fahringer, G. Fox, D. Gannon, C. Goble, M. Livny, L. Moreau, and J. Myers. Examining the challenges of scientific workflows. *Computer*, 40(12): 24–32, Dec. 2007.
- [140] J. Livny, H. Teonadi, M. Livny, and M. K Waldor. High-throughput, kingdom-wide prediction and annotation of bacterial non-coding RNAs. *PLoS ONE*, 3(9):e3197, Sept. 2008.
- [141] F. S. Hsieh and J. B. Lin. A dynamic scheme for scheduling complex tasks in manufacturing systems based on collaboration of agents. *Applied Intelligence*, 41(2):366–382, Sept. 2014.
- [142] H. Topcuoglu, S. Hariri, and M. Y. Wu. Performance-effective and low-complexity task scheduling for heterogeneous computing. *IEEE Transactions on Parallel and Distributed Systems*, 13(3):260–274, Mar. 2002.
- [143] A. Rădulescu and A. J. C. van Gemund. On the complexity of list scheduling algorithms for distributed-memory systems. In *Proc., ACM Supercomput.*, pages 68–75, June 1999.
- [144] H. M. Fard, R. Prodan, J. J. D. Barrionuevo, and T. Fahringer. A multi-objective approach for workflow scheduling in heterogeneous environments. In *Proc., IEEE/ACM CCGrid*, pages 300–309, May 2012.
- [145] S. Darbha and D. P. Agrawal. Optimal scheduling algorithm for distributed-memory machines. *IEEE Transactions on Parallel and Distributed Systems*, 9(1):87–95, Jan. 1998.
- [146] R. Bajaj and D. P. Agrawal. Improving scheduling of tasks in a heterogeneous environment. *IEEE Transactions on Parallel and Distributed Systems*, 15(2):107–118, Feb. 2004.
- [147] A. Gerasoulis and T. Yang. A comparison of clustering heuristics for scheduling directed acyclic graphs on multiprocessors. *Journal of Parallel and Distributed Computing*, 16(4):276 – 291, Dec. 1992.
- [148] J.-C. Liou and M. A. Palis. An efficient task clustering heuristic for scheduling DAGs on multiprocessors. In *Proc., Resource Management, Symp. of Parallel and Distrib. Processing*, pages 152–156, Oct. 1996.
- [149] K. Bessai, S. Youcef, A. Oulamara, Claude Godart, and S. Nurcan. Bi-criteria workflow tasks allocation and scheduling in cloud computing environments. In *Proc., IEEE CLOUD*, pages 638–645, June 2012.
- [150] T. He, S. Chen, H. Kim, L. Tong, and K.-W. Lee. Scheduling parallel tasks onto opportunistically available cloud resources. In *Proc., IEEE CLOUD*, pages 180–187, June 2012.

- [151] Z. Xiao, W. Song, and Q. Chen. Dynamic resource allocation using virtual machines for cloud computing environment. *IEEE Transactions on Parallel and Distributed Systems*, 24(6):1107–1117, June 2013.
- [152] S. T. Maguluri, R. Srikant, and L. Ying. Stochastic models of load balancing and scheduling in cloud computing clusters. In *Proc., IEEE INFOCOM*, pages 702–710, Mar. 2012.
- [153] T. R. Browning and A. A. Yassine. Resource-constrained multi-project scheduling: Priority rule performance revisited. *International Journal of Production Economics*, 126(2):212–228, Mar. 2010.
- [154] D. Shue, M. J. Freedman, and A. Shaikh. Performance isolation and fairness for multi-tenant cloud storage. In *Proc., USENIX OSDI*, pages 349–362, Oct. 2012.
- [155] S. Pandey, L. Wu, S. Guru, and R. Buyya. A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments. In *Proc., IEEE Advanced Information Networking and Applications*, pages 400–407, Apr. 2010.
- [156] M. A. Rodriguez and R. Buyya. Deadline based resource provisioning and scheduling algorithm for scientific workflows on clouds. *IEEE Transactions on Cloud Computing*, 2(2):222–235, Apr. 2014.
- [157] Z. Wu, X. Liu, Z. Ni, D. Yuan, and Y. Yang. A market-oriented hierarchical scheduling strategy in cloud workflow systems. *Journal of Supercomputing*, 63(1):256–293, Jan. 2013.
- [158] H. M. Fard, R. Prodan, and T. Fahringer. A truthful dynamic workflow scheduling mechanism for commercial multicloud environments. *IEEE Transactions on Parallel and Distributed Systems*, 24(6):1203–1212, June 2013.
- [159] G. Juve, E. Deelman, K. Vahi, G. Mehta, B. Berriman, B. P. Berman, and P. Maechling. Scientific workflow applications on amazon EC2. In *Proc., IEEE E-Science Wksp*, pages 59–66, Dec. 2009.
- [160] J. Jin, J. Luo, A. Song, F. Dong, and R. Xiong. BAR: An efficient data locality driven task scheduling algorithm for cloud computing. In *Proc., IEEE/ACM CCGrid*, pages 295–304, May 2011.
- [161] D. Yuan, Y. Yang, X. Liu, and J. Chen. A data placement strategy in scientific cloud workflows. *Future Generation Computer Systems*, 26(8):1200–1214, Oct. 2010.
- [162] E. Deelman, J. Blythe, Y. Gil, C. Kesselman, G. Mehta, K. Vahi, K. Blackburn, A. Lazzarini, A. Arbree, R. Cavanaugh, and S. Koranda. Mapping abstract complex workflows onto grid environments. *Journal of Grid Computing*, 1(1):25–39, Mar. 2003.
- [163] Q. Zhu and G. Agrawal. Resource provisioning with budget constraints for adaptive applications in cloud environments. *IEEE Transactions on Services Computing*, 5(4):497–511, Oct./Dec. 2012.
- [164] L. Ramakrishnan, C. Koelbel, Yang suk Kee, R. Wolski, D. Nurmi, D. Gannon, G. Obertelli, A. YarKhan, A. Mandal, T. M. Huang, K. Thyagaraja, and D. Zagorodnov. VGrADS: enabling e-science workflows on grids and clouds with fault tolerance. In *Proc., IEEE High Performance Comput. Networking, Storage and Analysis*, pages 1–12, Nov. 2009.

- [165] K. Plankensteiner and R. Prodan. Meeting soft deadlines in scientific workflows using resubmission impact. *IEEE Transactions on Parallel and Distributed Systems*, 23(5):890–901, May 2012.
- [166] S. Abrishami, M. Naghibzadeh, and D. H. J. Epema. Deadline-constrained workflow scheduling algorithms for infrastructure as a service clouds. *Future Generation Computer Systems*, 29(1):158–169, Jan. 2013.
- [167] W. Tsai, X. Sun, Q. Shao, and G. Qi. Two-tier multi-tenancy scaling and load balancing. In *Proc., IEEE ICEBE*, pages 484–489, Nov. 2010.
- [168] T. Kwok and A. Mohindra. Resource calculations with constraints, and placement of tenants and instances for multi-tenant SaaS applications. In *Proc., Int. Conf. on Service-Oriented Comput.*, pages 633–648, Dec. 2008.
- [169] J. Espadas, A. Molina, G. Jiménez, M. Molina, R. Ramírez, and D. Concha. A tenant-based resource allocation model for scaling software-as-a-service applications over cloud computing infrastructures. *Future Generation Computer Systems*, 29(1):273–286, Jan. 2013.
- [170] D. Klusacek, H. Rudova, R. Baraglia, M. Pasquali, and G. Capannini. Comparison of multi-criteria scheduling techniques. In *Grid Computing: Achievements and Prospects*, pages 173–184. Springer US, 2008.
- [171] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya. Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Software: Practice and Experience*, 41(1):23–50, Jan. 2011.
- [172] M. Mao and M. Humphrey. Auto-scaling to minimize cost and meet application deadlines in cloud workflows. In *Proc., IEEE/ACM Supercomputing*, pages 1–12, Nov. 2011.
- [173] M. Mao and M. Humphrey. A performance study on the vm startup time in the cloud. In *Proc., IEEE CLOUD*, pages 423–430, June 2012.
- [174] M. Malawski, G. Juve, E. Deelman, and J. Nabrzyski. Cost-and deadline-constrained provisioning for scientific workflow ensembles in IaaS clouds. In *Proc., IEEE High Performance Comput., Networking, Storage and Analysis*, pages 1–11, Nov. 2012.
- [175] G. Juve, A. Chervenak, E. Deelman, S. Bharathi, G. Mehta, and K. Vahi. Characterizing and profiling scientific workflows. *Future Generation Computer Systems*, 29(3):682–692, Mar. 2013.
- [176] M. V. Barbera, S. Kosta, A. Mei, , and J. Stefa. To offload or not to offload? the bandwidth and energy costs of mobile cloud computing. In *Proc., IEEE INFOCOM*, pages 1285–1293, Apr. 2013.
- [177] T. Orphanoudakis, E. Kosmatos, J. Angelopoulos, and A. Stavdas. Exploiting PONs for mobile backhaul. *IEEE Communications Magazine*, 51(2):27–34, Feb. 2013.
- [178] S. Davy, J. Famaey, J. Serrat-Fernandez, J.L. Gorricho, A. Miron, M. Dramitinos, P.M. Neves, S. Latre, and E. Goshen. Challenges to support edge-as-a-service. *IEEE Communications Magazine*, 52(1):132–139, Jan. 2014.

- [179] Intel Corporation. Increasing mobile operators' value proposition with edge computing. *Technical Brief*, pages 1–6, Aug. 2013.
- [180] Michael Till Beck, Martin Werner, Sebastian Feld, and S Schimper. Mobile edge computing: A taxonomy. In *Proc., Sixth Int. Conf. on Advances in Future Internet*, pages 48–54, Nov. 2014.
- [181] Nokia Networks. Intelligent base stations. *White Paper*, pages 1–20, Feb. 2013.
- [182] A. Ceselli, M. Premoli, and S. Secci. Cloudlet network design optimization. In *Proc., IFIP Networking*, pages 1–9, May 2015.
- [183] NGMN Alliance. Further study on critical C-RAN technologies. pages 1–93, Mar. 2015.
- [184] K. Ha, Y. Abe, Z. Chen, W. Hu, B. Amos, P. Pillai, and M. Satyanarayanan. Adaptive vm handoff across cloudlets. Technical report, Carnegie Mellon Univ., Pittsburgh, PA, USA, Tech. Rep. CMU-CS-15-113, pages 1-27, June 2015.
- [185] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield. Live migration of virtual machines. In *Proc., 2nd ACM/USENIX Symposium on Networked Systems Design & Implementation*, pages 273–286, May 2005.
- [186] K. Kumar and Yung-Hsiang Lu. Cloud computing for mobile users: Can offloading computation save energy? *Computer*, 43(4):51–56, Apr. 2010.
- [187] Y. Luo, F. Effenberger, and M. Sui. Cloud computing provisioning over passive optical networks. In *Proc., IEEE Int. Conf. on Commun. in China (ICCC)*, pages 255–259, Aug. 2012.
- [188] M. P. McGarry, M. Reisslein, and M. Maier. Ethernet passive optical network architectures and dynamic bandwidth allocation algorithms. *IEEE Communications Surveys and Tutorials*, 10(3):46–60, Sept. 2008.
- [189] J. Zheng and H. T. Mouftah. A survey of dynamic bandwidth allocation algorithms for ethernet passive optical networks. *Optical Switching and Networking*, 6(3):151–162, July 2009.
- [190] A. G. Sarigiannidis, M. Iloridou, P. Nicopolitidis, G. Papadimitriou, F.-N. Pavlidou, P. G. Sarigiannidis, M. D. Louta, and V. Vitsas. Architectures and bandwidth allocation schemes for hybrid wireless-optical networks. *IEEE Communications Surveys & Tutorials*, 17(1):427–468, Mar. 2015.
- [191] G. Kramer. *Ethernet Passive Optical Networks*. McGraw Hill Professional, Mar. 2005.
- [192] L. Rao, X. Liu, L. Xie, and W. Liu. Minimizing electricity cost: Optimization of distributed internet data centers in a multi-electricity-market environment. In *Proc., IEEE INFOCOM*, pages 1–9, Mar. 2010.
- [193] D. Cheng, C. Jiang, and X. Zhou. Heterogeneity-aware workload placement and migration in distributed sustainable datacenters. In *Proc., IEEE International Parallel and Distributed Processing Symposium*, pages 307–316, May 2014.
- [194] T. Zhao, S. Zhou, X. Guo, Y. Zhao, and Z. Niu. A cooperative scheduling scheme of local cloud and internet cloud for delay-aware mobile cloud computing. In *Proc., IEEE Globecom Wkshps*, pages 1–6, Dec. 2015.

- [195] D. Gross, J. F. Shortle, J. M. Thompson, and C. M. Harris. *Fundamentals of Queueing Theory*. Wiley, fourth edition, Aug. 2008.
- [196] G. Bradski and A. Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media, Inc., Sept. 2008.
- [197] M. Simsek, A. Aijaz, M. Dohler, J. Sachs, and G. Fettweis. 5G-enabled tactile internet. *IEEE Journal on Selected Areas in Communications*, 34(3):460–473, Mar. 2016.
- [198] K. Tanaka, A. Agata, and Y. Horiuchi. IEEE 802.3av 10G-EPON standardization and its research and development status. *IEEE/OSA Journal of Lightwave Technology*, 28(4):651–661, Feb. 2010.
- [199] S. Andreev, O. Galinina, A. Pyattaev, M. Gerasimenko, T. Tirronen, J. Torsner, J. Sachs, M. Dohler, and Y. Koucheryavy. Understanding the IoT connectivity landscape: A contemporary M2M radio technology roadmap. *IEEE Communications Magazine*, 53(9):32–40, Sept. 2015.
- [200] V. Suryaprakash, P. Rost, and G. Fettweis. Are heterogeneous cloud-based radio access networks cost effective? *IEEE Journal on Selected Areas in Communications*, 33(10):2239–2251, Oct. 2015.
- [201] C. Ranaweera, M. G. C. Resende, K. Reichmann, P. Iannone, P. Henry, B. J. Kim, P. Magill, K. N. Oikonomou, R. K. Sinha, and S. Woodward. Design and optimization of fiber optic small-cell backhaul based on an existing fiber-to-the-node residential access network. *IEEE Communications Magazine*, 51(9):62–69, Sept. 2013.
- [202] A. Reaz, V. Ramamurthi, M. Tornatore, and B. Mukherjee. Cloud-Integrated WOBAN: An offloading-enabled architecture for service-oriented access networks. *Computer Networks*, 68: 5–19, Aug. 2014.
- [203] B. Yang, Z. Zhang, K. Zhang, and W. Hu. Integration of micro data center with optical line terminal in passive optical network. In *Proc., 21st OptoElectronics and Commun. Conf. held jointly with Int. Conf. on Photonics in Switching*, pages 1–3, July 2016.
- [204] J. i. Kani, F. Bourgart, A. Cui, A. Rafel, M. Campbell, R. Davey, and S. Rodrigues. Next-generation pon-part i: Technology roadmap and general requirements. *IEEE Communications Magazine*, 47(11):43–49, Nov. 2009.
- [205] Y. Dashti, A. Mercian, and M. Reisslein. Evaluation of dynamic bandwidth allocation with clustered routing in fiwi networks. In *Proc. IEEE 20th International Workshop on Local Metropolitan Area Networks*, pages 1–6, May 2014.
- [206] L. Kleinrock. *Queueing Systems, Volume II: Computer Applications*. Wiley-Interscience, May 1976.
- [207] F. Aurzada, M. Scheutzow, M. Herzog, M. Maier, and M. Reisslein. Delay analysis of Ethernet passive optical networks with gated service. *OSA Journal of Optical Networking*, 7(1):25–41, Jan. 2008.
- [208] N. Ghazisaidi, M. Scheutzow, and M. Maier. Survivability analysis of next-generation passive optical networks and Fiber-Wireless access networks. *IEEE Transactions on Reliability*, 60(2): 479–492, June 2011.

- [209] S. Huang, C. U. Martel, and B. Mukherjee. Survivable multipath provisioning with differential delay constraint in telecom mesh networks. *IEEE/ACM Transactions on Networking*, 19(3): 657–669, June 2011.
- [210] Y.-S. Dai, B. Yang, J. Dongarra, and G. Zhang. Cloud service reliability: Modeling and analysis. *Proc., IEEE Pacific Rim Int’l Symp. Dependable Computing (PRDC)*, pages 1–17, Nov. 2009.
- [211] J. Chen, L. Wosinska, C. M. Machuca, and M. Jaeger. Cost vs. reliability performance study of fiber access network architectures. *IEEE Communications Magazine*, 48(2):56–65, Feb. 2010.
- [212] Open source computer vision library (OpenCV). <https://www.willowgarage.com/pages/software/opencv/>, 2015.
- [213] A. S. Thyagaturu, A. Mercian, M. P. McGarry, M. Reisslein, and W. Kellerer. Software defined optical networks (sdons): A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 18(4):2738–2786, July 2016.
- [214] G. Wang, R. Gu, J. Zhang, H. Li, and Y. Ji. Asynchronous delivery oriented efficient resource allocation in twdm-pon enabled fronthaul. In *Proc., IEEE Online Conf. on Green Commun.*, pages 45–50, Nov. 2016.
- [215] oPhone. URL <http://www.oNotes.com>. Accessed on Sept. 30, 2015.
- [216] E. Brynjolfsson and A. McAfee. *The Second Machine Age: Work, Progress, and Prosperity in a Time of Brilliant Technologies*. W. W. Norton and Company, Jan. 2014.
- [217] K. Moskvitch. Tactile Internet: 5G and the cloud on steroids. *IET Engineering & Technology*, 10(4):48–53, May 2015.
- [218] D. Szabó, A. Gulyás, F. H. P. Fitzek, and D. E. Lucani. Towards the Tactile Internet: decreasing communication latency with network coding and software defined networking. In *Proc., European Wireless*, pages 428–433, May 2015.
- [219] K. Kamei, S. Nishio, N. Hagita, and M. Sato. Cloud networked robotics. *IEEE Network*, 26(3):28–34, May/June 2012.
- [220] N. Carr. All can be lost: The risk of putting our knowledge in the hands of machines. *The Atlantic [Online]*: <http://www.theatlantic.com/magazine/archive/2013/11/the-great-forgetting/309516/>, Nov. 2013.
- [221] V. Klingspor, J. Demiris, and M. Kaiser. Human-robot-communication and machine learning. *Applied Artificial Intelligence*, 11(7):719–746, Nov. 1997.