UQAC
Université du Québec
à Chicoutimi

Conception d'un modèle architectural collaboratif pour l'informatique omniprésente à la périphérie des réseaux mobiles

Par Ghassan Fadlallah

Thèse présentée à l'Université du Québec à Chicoutimi en vertu d'un protocole d'entente avec l'Université du Québec en Outaouais, en vue de l'obtention du grade de Philosophie de Doctorat (Ph. D.) en sciences et technologies de l'information

Québec, Canada

RÉSUMÉ

Le progrès des technologies de communication pair-à-pair et sans fil a de plus en plus permis l'intégration de dispositifs portables et omniprésents dans des systèmes distribués et des architectures informatiques de calcul dans le paradigme de l'internet des objets. De même, ces dispositifs font l'objet d'un développement technologique continu. Ainsi, ils ont toujours tendance à se miniaturiser, génération après génération durant lesquelles ils sont considérés comme des dispositifs de facto. Le fruit de ces progrès est l'émergence de l'informatique mobile collaborative et omniprésente, notamment intégrée dans les modèles architecturaux de l'Internet des Objets. L'avantage le plus important de cette évolution de l'informatique est la facilité de connecter un grand nombre d'appareils omniprésents et portables lorsqu'ils sont en déplacement avec différents réseaux disponibles.

Malgré les progrès continuels, les systèmes intelligents mobiles et omniprésents (réseaux, dispositifs, logiciels et technologies de connexion) souffrent encore de diverses limitations à plusieurs niveaux tels que le maintien de la connectivité, la puissance de calcul, la capacité de stockage de données, le débit de communications, la durée de vie des sources d'énergie, l'efficacité du traitement de grosses tâches en termes de partitionnement, d'ordonnancement et de répartition de charge.

Le développement technologique accéléré des équipements et dispositifs de ces modèles mobiles s'accompagne toujours de leur utilisation intensive. Compte tenu de cette réalité, plus d'efforts sont nécessaires à la fois dans la conception structurelle tant au matériel et logiciel que dans la manière dont il est géré. Il s'agit d'améliorer, d'une part, l'architecture de ces modèles et leurs technologies de communication et, d'autre part, les algorithmes d'ordonnancement et d'équilibrage de charges pour effectuer leurs travaux efficacement sur leurs dispositifs.

Notre objectif est de rendre ces modèles omniprésents plus autonomes, intelligents et collaboratifs pour renforcer les capacités de leurs dispositifs, leurs technologies de connectivité et les applications qui effectuent leurs tâches. Ainsi, nous avons établi un modèle architectural autonome, omniprésent et collaboratif pour la périphérie des réseaux. Ce modèle s'appuie sur diverses technologies de connexion modernes telles que le sans-fil, la radiocommunication pair-à-pair, et les technologies offertes par LoPy4 de Pycom telles que LoRa, BLE, Wi-Fi, Radio Wi-Fi et Bluetooth. L'intégration de ces technologies permet de maintenir la continuité de la communication dans les divers environnements, même les plus sévères. De plus, ce modèle conçoit et évalue un algorithme d'équilibrage de charge et d'ordonnancement permettant ainsi de renforcer et améliorer son efficacité et sa qualité de service (QoS) dans différents environnements. L’évaluation de ce modèle architectural montre des avantages tels que l’amélioration de la connectivité et l’efficacité d’exécution des tâches.

ABSTRACT

Advances in peer-to-peer and wireless communication technologies have increasingly enabled the integration of mobile and pervasive devices into distributed systems and computing architectures in the Internet of Things paradigm. Likewise, these devices are subject to continuous technological development. Thus, they always tend to be miniaturized, generation after generation during which they are considered as de facto devices. The success of this progress is the emergence of collaborative mobiles and pervasive computing, particularly integrated into the architectural models of the Internet of Things. The most important benefit of this form of computing is the ease of connecting a large number of pervasive and portable devices when they are on the move with different networks available.

Despite the continual advancements that support this field, mobile and pervasive intelligent systems (networks, devices, software and connection technologies) still suffer from various limitations at several levels such as maintaining connectivity, computing power, ability to data storage, communication speeds, the lifetime of power sources, the efficiency of processing large tasks in terms of partitioning, scheduling and load balancing.

The accelerated technological development of the equipment and devices of these mobile models is always accompanied by their intensive use. Given this reality, it requires more efforts both in their structural design and management. This involves improving on the one hand, the architecture of these models and their communication technologies, and, on the other hand, the scheduling and load balancing algorithms for the work efficiency.

The goal is to make these models more autonomous, intelligent, and collaborative by strengthening the different capabilities of their devices, their connectivity technologies and the applications that perform their tasks. Thus, we have established a collaborative autonomous and pervasive architectural model deployed at the periphery of networks. This model is based on various modern connection technologies such as wireless, peer-to-peer radio communication, and technologies offered by Pycom's LoPy4 such as LoRa, BLE, Wi-Fi, Radio Wi-Fi and Bluetooth. The integration of these technologies makes it possible to maintain the continuity of communication in the various environments, even the most severe ones. Within this model, we designed and evaluated a load balancing and scheduling algorithm to strengthen and improve its efficiency and quality of service (QoS) in different environments. The evaluation of this architectural model shows payoffs such as improvement of connectivity and efficiency of task executions.

TABLE DES MATIÈRES

LISTE DES TABLEAUX

LISTE DES FIGURES

LISTE DES ABRÉVIATIONS

P2P

Peer-to-Peer

5G

5ème Génération

LoRa

Long Range

AP

Access Point (en français point d'accès)

BLE

Bluetooth Low Energy

GA

Genetic algorithms

SA

Simulated annealing

MILP

Mixed Integer Linear Programming

VNS

Variable Neighborhood Search

ANN

Artificial Neural Network

CM

Corrective Maintenance

PM

Preventive Maintenance

RFID

Radio-Frequency Identification

SOS

Stochastic Online Scheduling

IoT

Internet of Things (en français internet des objets)

VMs

Virtual Machines

LPWAN

Low-Power Wide-Area Network (en français réseau étendu à faible consommation d'énergie)

LTE

Long-Term Evolution

LTE-A

LTE Advanced

MANET

Mobile Ad hoc Network

VANET

Vehicular Ad hoc Network

ML

Machine Learning (en français apprentissage automatique)

VDTNs

Vehicular Delay-Tolerant Networks

LCFP

Longest Cloudlet Fastest Processing Element

SCFP

Shortest Cloudlet Fastest Processing Element

FCFS

First Come First Serve

PL

Path Loss (en français Perte de propagation)

SJF

Shortest Job First

QoS

Quality of Service (en français qualité de service)

RR

Round Robin

MAC

Mobile Ad hoc Cloud Computing

MEC

Mobile Edge Computing

NFC

Near Field Communication

CPM

Critical Path Method

CPA

Critical Path Analysis

eNodeB

evolved Node B

PSO

Particle Swarm Optimization

Wi-Fi

Wireless Fidelity

Wi-Fi Direct

Wi-Fi Peer-to-Peer (P2P)

WiMAX

Worldwide Interoperability for Microwave Access

DÉDICACE

Je dédie cette thèse à ma famille.

À mes chers qui sont partis et dont je suis toujours inondé par leur amour.

À mes chers qui m'entourent de leur gentillesse et de leurs soins qui me donnent la détermination et la persévérance.

À ceux qui sont mon amour, mes espoirs, mes rêves et mes belles fleurs qui fleuriront dans un futur proche.

REMERCIEMENTS

Mes remerciements chaleureux avec ma grande gratitude s'adressent à mes directeurs de thèse, Dr Hamid Mcheick et Dr Djamal Rebaine pour leur soutien, leurs conseils, leur patience, leur confiance et leurs encouragements tout au long de ma thèse.

Je tiens à remercier l'Université du Québec à Chicoutimi, en particulier le Département d'informatique et de mathématiques et au directeur de 3ème cycle, le professeur Bob-Antoine Jerry Ménélas, pour m'avoir offert un environnement de travail propice aux études et à la recherche.

Je suis également très reconnaissant envers toutes les personnes qui m'ont aidé dans mon cheminement et qui ont contribué, directement ou indirectement, à la réalisation de cette thèse.

INTRODUCTION

Le monde contemporain se caractérise par l'accélération d’innovations technologiques dans divers domaines, notamment les communications sans fil, les appareils mobiles intelligents et omniprésents, l'informatique collaborative omniprésente et les technologies de l'information. En conséquence, tout le monde qui nous entoure est déjà connecté via un nombre énorme de dispositifs (objets) qui peuvent être contrôlés et gérés à distance. Cela conduit à une expansion radicale de l'Internet des objets (IoT), de l'intelligence artificielle, de la blockchain et de technologies multiples qui gèrent différents aspects de notre vie quotidienne. En effet, ces appareils sont en train de prendre une place de plus en plus importante dans l'ensemble de nos activités tels que : emplois, marchés, milieux d'affaires, science, culture, littérature, sport, etc.

L'informatique omniprésente, également appelée informatique ubiquitaire ou encore ambiante, intègre des capacités de calculs dans des dispositifs omniprésents connectés aux réseaux et disponibles en permanence. Elle permet à ces objets de communiquer efficacement et effectuer des tâches utiles permettant aux utilisateurs la portabilité, autonomie et disponibilité des services sans être présents dans un centre de services particulier. Contrairement à l'informatique de bureau, l'informatique omniprésente peut se produire, à tout moment, n'importe où, avec n'importe quel appareil, sur n'importe quel réseau et dans n'importe quel format de données. Par conséquent, l'informatique omniprésente renforce les systèmes distribués en y intégrant de petits dispositifs omniprésents à faible coût.

L'avancement des technologies de miniaturisation des dispositifs et des technologies de communication sans fil et Pair-à-pair (P2P) dans les réseaux mobiles est un facteur fondamental de l'informatique omniprésente mobile. Ainsi, effectuer des tâches informatiques est devenue possible pendant que l'utilisateur est en mouvement. L'informatique omniprésente mobile permet l'interaction d'utilisateurs distants et mobiles aux diverses ressources pour effectuer différentes activités à l'aide d'une variété de dispositifs omniprésents. Cette interaction est devenue une technique efficace pour traiter des problèmes complexes, allant de la santé et des soins à domicile à la surveillance de l'environnement et aux systèmes de transport intelligents [1]. Ces problèmes peuvent être décomposés en petits sous-problèmes qui sont adéquatement traités sur les appareils mobiles et omniprésents dans les modèles architecturaux distribués de l'Internet des objets (IoT). Les dispositifs omniprésents évoluent très rapidement, leurs fonctions s'enrichissent constamment, tandis que leurs capacités de communication s'accroissent. Dans ce paradigme, ces objets (utilisateurs, matériels et logiciels) sont dynamiques et changent de manière imprévisible [2].

Le développement des systèmes mobiles omniprésents a conduit à des conceptions innovantes pour les réseaux mobiles et leurs technologies de communication qui connectent leurs dispositifs. Ainsi, de nouveaux modèles architecturaux ont été conçus pour mieux gérer les différents services offerts par l'informatique omniprésente mobile et les techniques d'interaction entre ces dispositifs. Nous distinguons de nombreuses architectures qui ont été établies récemment dans le cadre de l'Internet des Objets et de l'informatique omniprésente mobile sur les réseaux mobiles tels que cloud, fog, edge, mobile edge computing, mobile cloudlet, etc. [3].

Le cloud computing (informatique dans les nuages) est l'architecture principale du paradigme de l'Internet des objets. Il permettait, comme les autres architectures de ce paradigme, l'accès à des services des technologies de l'information à travers des réseaux d'appareils connectés par l'Internet avec des diverses technologies de communication, notamment sans fil. Le cloud computing a contribué avec succès au renforcement de l'informatique mobile omniprésente et à l'amélioration de l'efficacité des réseaux de dispositifs mobiles et omniprésents aux différents niveaux, y compris la vélocité de communication et les capacités de transfert de données, des processus de calculs informatiques et de stockage. Cependant, il a fait face à de sérieux problèmes en raison de la centralisation de ses services et de l'énorme quantité de données échangées entre les appareils et les serveurs centraux. De nombreuses autres architectures émergentes, qui reposent sur des serveurs intermédiaires et à proximité de la périphérie, ont été proposées pour surmonter ces problèmes telles que fog, edge et mobile edge computing (MEC) [4]. Les architectures fog et edge computing [5] et, plus récemment, mobile cloudlet et mobile edge computing (MEC) ainsi que de nombreux autres modèles architecturaux, deviennent de plus en plus importants dans le paradigme de l'IoT en tant qu'applications informatiques collaboratives distribuées via les réseaux mobiles.

En raison de l'intensité excessive des connexions et de l'échange de requêtes d'informations et de données, le problème rencontré par le cloud computing a été rapidement transféré au reste des architectures intermédiaires (edge et fog) et à celles construites près de la périphérie en tant que solutions. Ces architectures émergentes rencontraient, à leur tour, le même problème et nécessitaient la construction d'architectures moins spacieuses et plus proches de la périphérie. Ainsi, la construction des architectures à proximité s'est poursuivie dont les plus importantes sont : Cloudlet, collaborative fog computing (CoFog), Mobile ad hoc cloud computing (MAC), MoCCA, mobile cellular cloud architecture, mClouds, MobiCloud, mobile cloudlet, multi-group networking, smart groups formation in Wi-Fi Direct multi-group networks, MANET, VANET, etc. [6-16].

L'efficacité des dispositifs dans ces modèles architecturaux devient disproportionnée par rapport à la demande intense et excessive des communications, surtout dans les événements extrêmes dans lesquels les centres de services de communication, les stations de base et les points d'accès peuvent tomber en panne, être surchargés, isolés ou même détruits. Par conséquent, la solution efficace dans ces situations peut être réalisée en utilisant des systèmes ayant des réseaux locaux et autonomes composés d'appareils mobiles et de ressources omniprésentes qui peuvent être connectées via leurs propres connexions radio P2P [16]. Ces appareils à faible coût et présents en permanence dans nos poches peuvent être regroupés instantanément et dynamiquement dans des réseaux en temps réel [17]. Ces réseaux pourraient être utilisés dans diverses situations - comme le plan d'évacuation dans les zones dangereuses présenté à la Section II.6.1 - pour fournir aux utilisateurs des moyens efficaces et prometteurs pour gérer leur travail à tout moment et n'importe où dans différents environnements. Pour ces raisons ainsi que pour les améliorations technologiques futures attendues, nous croyons que les systèmes ayant ces appareils et réseaux seront les outils de facto pour soutenir le paradigme de l'informatique collaborative mobile intelligente.

Le rythme accéléré du développement technologique impose de réels défis, en termes de les assimiler dans des systèmes bien gérés et utilisés dans divers domaines, notamment dans l'informatique omniprésente sur les réseaux mobiles. La raison est due aux progrès rapides dans le développement d'architectures et de techniques de modèles informatiques, de dispositifs mobiles omniprésents et de technologies de communication sans fil et radio de poste à poste (P2P) [17]. Le P2P pourrait utiliser diverses technologies sans fil à courte portée telles que Bluetooth, Wi-Fi Direct, LTE Direct et NFC (near field communication) [18, 19]. Les technologies émergentes peuvent étendre la portée des communications P2P et sans fil comme celles offertes par Lopy4 de Pycom. En effet, la mise en réseau est devenue un foyer d'appareils omniprésents qui peuvent être connectés n'importe où et à tout moment. Bientôt, avec les technologies clés telles que 5G et la communication par lumière visible, les réseaux mobiles permettront aux gens d'utiliser la technologie P2P évoluée à une vaste échelle. En conséquence, le nombre d'appareils mobiles et omniprésents ayant une durée de vie de la batterie plus longue et une latence réduite augmentera considérablement [20].

Le mode dynamique des réseaux omniprésents, défini par leur capacité d'autoconfiguration dynamique et d'interopérabilité [21, 22], séduit et attire les utilisateurs. La raison en est qu'il leur donne plus de liberté et de flexibilité en termes de lieu et de temps pour communiquer avec ces réseaux afin de traiter une tâche spécifique tout en poursuivant leurs différents travaux. De ce fait, et en plus du développement continu des dispositifs de ces réseaux, la demande pour leur utilisation augmente de façon exponentielle (croissance rapide et continue), en particulier dans les applications collaboratives complexes [2, 23, 24]. Malheureusement, les progrès technologiques de ces dispositifs ne progressent pas proportionnellement à cette demande pour leur utilisation. Par conséquent, cette disparité peut être compensée en améliorant les architectures émergentes en termes des performances des applications informatiques collaboratives s'exécutant sur ces appareils en termes d'optimisation de la vitesse de réponse et du temps d'achèvement des tâches. Le moyen le plus efficace d'y parvenir est i) d'établir une architecture permettant de décomposer les services en couches plus simples à évoluer, ii) d'améliorer les techniques d'ordonnancement et d'équilibrage de charge des tâches, et iii) de maintenir la connectivité des réseaux, en particulier dans les environnements difficiles, pour assurer la continuité des activités pendant le temps requis.

La technique d'ordonnancement consiste à organiser l'ordre d'exécution des tâches sur les appareils disponibles, tandis que l'équilibrage des charges vise à répartir les charges de travail entre les ressources informatiques [3, 25].

Le processus d'ordonnancement s'intéresse à l'allocation de tâches aux appareils au cours du temps, selon un environnement déterministe, stochastique ou en ligne où les paramètres qui décrivent le problème sont respectivement connus à l'avance, des variables aléatoires ou connus seulement au moment de la prise de décision [26].

L'amélioration du processus d'ordonnancement nécessite une meilleure compréhension de l'environnement à étudier et dépend de son état s'il est statique (fixe et non modifiable) ou dynamique (réévalué en ligne pour répondre aux changements). De même, les critères de traitement de l'environnement si local ou distribué peuvent être classés en différents types : (1) contraintes dures (les contraintes doivent être strictement respectées), (2) contraintes molles : il y a une flexibilité pour les respecter, (3) préemptif : les tâches peuvent être interrompues avant l'achèvement, ou (4) non-préemptif : les tâches sont exécutées sans interruption jusqu'à l'achèvement [3, 27, 28].

Les problèmes d'ordonnancement surviennent dans diverses applications du monde réel. Ils ont été initialement étudiés dans le secteur industriel, tels que la fabrication et l'assemblage de gros générateurs, et cela a conduit à l'introduction de l'analyse du chemin critique (CPA) et de la méthode du chemin critique (CPM) [29, 30]. Récemment, les problèmes d'ordonnancement sont devenus un domaine de recherche actif dans la communauté de l'IoT [31], en particulier au niveau informatique dans les modèles architecturaux collaboratifs omniprésents comme Cloud et Fog computing [32, 33].

Comme ces modèles architecturaux émergents sont intégrés aux systèmes distribués, nous avons concentré notre étude sur l'ordonnancement de modèles de machines parallèles.

Dans le cadre de cette thèse, nous nous intéressons à l'établissement d’un système mobile collaboratif à la périphérie, efficace en qualité de service. Il est basé sur des technologies de connexion modernes telles que le sans-fil, la radio P2P, en outre celles offertes par Lopy4 de Pycom, pour maintenir la connectivité dans les différents environnements, même dans les plus difficiles [34, 35]. Pour atteindre ces objectifs, nous avons conçu un modèle collaboratif mobile et omniprésent, basé sur un style architectural en couches [36], qui permet aux utilisateurs de communiquer dans diverses situations. À cette fin, nous avons pris en compte les limitations de capacité de leurs dispositifs en termes de stockage de données, de calculs et d'autonomie attendue de la batterie.

La plate-forme de communication dans ce modèle se compose de différentes technologies de connexion, où la priorité est pour le Wi-Fi avec ses stations de base et ses points d'accès. Puis pour les communications radio P2P telles que Wi-Fi Direct et Bluetooth. En cas de pannes diverses du Wi-Fi, nous l'avons alterné par des cartes microcontrôleurs Lopy4 de Pycom comme points d'accès que nous avons configuré pour offrir une large portée de radio Wi-Fi sur des kilomètres. Lopy4 est en fait une plate-forme IoT de classe entreprise exceptionnelle. Il fonctionne avec LoRa, Sigfox, Wi-Fi et Bluetooth. Il possède des fonctionnalités d'un processeur puissant, BLE et radio Wi-Fi de pointe. Aussi, il peut se doubler comme une passerelle Nano LoRa avec une consommation d'énergie ultra-faible (une fraction par rapport aux autres microcontrôleurs connectés) [37]. Ces Lopys sont répartis à des distances suffisantes entre eux (plus de 1 km selon la configuration et l’installation de l’antenne). Chacun d'eux communique avec les dispositifs à proximité via Wi-Fi ou Bluetooth et également avec le reste des Lopys via leur radio Wi-Fi. Les dispositifs à proximité peuvent se connecter aussi les uns aux autres via Wi-Fi Direct.

Pour surmonter ces limitations et assurer l'efficacité de ce système, nous avons d'abord introduit et étudié les techniques et algorithmes d'ordonnancement et d'équilibrage de charge récemment publiés. Par conséquent, nous avons développé et conçu un algorithme d'ordonnancement basé sur une approche gloutonne. Cet algorithme s'adapte bien aux différents environnements en termes de rapidité, qualité et facilité de prise de décision à chaque étape [38-40]. Cet algorithme permet également aux petits appareils mobiles non seulement de gérer des tâches à la mesure de leurs capacités, mais aussi d'équilibrer leur charge.

CHAPITRE I. MOTIVATION, PROBLÉMATIQUES, OBJECTIFS ET MÉTHODOLOGIE

I.1. MOTIVATION

De nos jours, grâce au développement rapide des technologies de communication et d'électroniques, le monde assiste à une large diffusion des dispositifs omniprésents et des appareils intelligents mobiles portables et mettables. Ces appareils sont devenus étroitement liés à la vie quotidienne et s'y attacheront davantage à l'avenir avec la forte demande pour leur adoption et leur utilisation, notamment grâce au développement attendu de ces technologies ainsi que d'autres dans divers domaines. En fait, ces appareils ont fini par dominer et gérer notre vie quotidienne avec leurs diverses activités, travaux et tâches et ils deviendront indispensables. Il est donc nécessaire que les chercheurs et les praticiens se concentrent sur l'intégration de ces dispositifs dans les architectures de l'Internet des Objets (IoT) comme Cloud, Fog, Edge computing et les autres architectures qui sont de plus en plus s'installent sur des périphéries des réseaux. Ainsi, il est important de renforcer la capacité de gestion et de contrôle de ces dispositifs, en plus d'améliorer leur facilité d'utilisation et leur portée dans les zones rurales. Par conséquent, il est nécessaire que la recherche comprenne également le développement d'algorithmes et de logiciels pour le maintien de leur connexion dans diverses circonstances.

De même, la motivation du travail présenté dans cette thèse provient également des cas critiques et difficiles dans lesquels les communications sont interrompues pour diverses raisons, y compris les catastrophes naturelles, les guerres, et les attaques terroristes. La situation dans tous ces cas nécessite le maintien de la connexion et de la communication entre toutes les personnes concernées n'importe où via les appareils disponibles, ainsi que d'alléger et de faciliter les applications qu'ils utilisent.

I.2. PROBLÉMATIQUE

Malgré l'avènement de la technologie sans fil 5G, les réseaux sont toujours confrontés à de nombreux défis, notamment : la topologie dynamique (les utilisateurs d'appareils sont libres de se déplacer arbitrairement), le routage, les limitations de la bande passante, la « Qualité de service (QoS) », la sécurité et le travail en ligne « Internetworking », la vulnérabilité des canaux et la limitation des ressources en termes d'énergie, des calculs et de stockage [1]. En conséquence, plusieurs pistes de recherche ont été lancées avec la cinquième génération de la communication sans fil. Les exigences de performance pour cette génération ont été définies en termes de capacité, efficacité spectrale, efficacité énergétique, débit de données et débit cellulaire moyen [2].

Ces évolutions technologiques s'accompagnent toujours de tendances à l'utilisation extensive des réseaux. Ce qui affecte négativement les solutions pour réduire ces défis, en particulier dans le domaine de l'informatique omniprésente collaborative. Cette utilisation extensive cause donc de nouveaux défis au niveau d'échange de données ainsi que d'exécution efficace des applications à travers les réseaux. En fait, cela pose le problème de l'accomplissement de grosses tâches. Ceci nécessite de les diviser en sous-tâches plus petites, qui à leur tour sont planifiées et leurs charges sont équilibrées entre les dispositifs du réseau.

Les recherches dans ce domaine sont de plus en plus florissantes, tant logicielles que matérielles comme en témoigne la littérature associée [3]. Par conséquent, il y a encore de la place pour des développements innovants dans ce domaine, comme nous l’espérons l’avoir démontré au fil de cette thèse.

Le principal problème que nous avons abordé dans cette thèse se situe généralement dans le contexte du domaine émergent de l'architecture des systèmes distribués, et de l'informatique omniprésente collaborative à la périphérie des réseaux mobiles intelligents. Ces architectures peuvent être classées en trois catégories : i) Principales-centralisées en tant que cloud, ii) Intermédiaires-centralisées comprenant entre autres fog, edge computing et mobile edge computing, et iii) Autres architectures périphériques telles que les technologies émergentes introduites ces dernières années, en particulier les réseaux mobiles ad-hoc à la périphérie (multigroupes, cloudlet mobile, etc.) [4].

Les dernières architectures périphériques qui ont été introduites dans les systèmes mobiles intelligents souffrent de problèmes en termes de déficience de communication, de consommation d'énergie, de stockage de données, de capacité de calcul et de décharge/charge entre les dispositifs périphériques et les centres de services du cloud [5]. A noter que le cloud computing a été créé comme une solution de l'échange de méga données et de l'informatique distribuée collaborative et omniprésente. Mais du fait de sa centralisation, il a fait face à de nombreux problèmes, notamment ceux éloignés du centre vers les périphéries. Par conséquent, les architectures émergentes fog et edge computing sont conçues comme des architectures permettant de résoudre ces problèmes de cloud à proximité de la périphérie. Ainsi, d'autres nouveaux modèles architecturaux ont été conçus en tant que solutions pour des problèmes dans des zones à proximité de la périphérie, notamment cloudlet, mobile cloud computing, mobile cloudlet, mobile ad hoc computing, smart P2P multi-group networks, etc.

Malgré l'avancement dans ces architectures émergentes, nous avons identifié de nombreux enjeux dans ce domaine de l'informatique omniprésente collaborative à la périphérie des réseaux, à savoir :

1. La portée réduite des dispositifs de réseau mobile en termes de connexion poste à poste P2P en cas de réduction et même de perte de la connectivité sans fil avec les stations de base.

2. La perte totale de l'opérabilité ou de l'efficacité de ces dispositifs lors d'événements graves tels que des catastrophes naturelles, des actes de terrorisme, etc. en raison de dysfonctionnements ou de pannes des stations de base de connexion sans fil.
3. La bande passante de connexion limitée en raison de la congestion résultant du déchargement et du chargement des messages (tâches et données) vers les serveurs centralisés fixes ou mobiles [6].
4. La limitation des appareils mobiles en termes de puissance et de capacité de calcul et de stockage des données [7].
5. La lacune réside dans la façon dont les applications existantes sont conçues comme des applications génériques qui ne sont pas destinées à ces appareils de capacité modérée pour accomplir de grandes tâches. Cette lacune est liée à la partition, d'ordonnancement et d'équilibrage de charge proportionnellement aux capacités limitées des dispositifs.

I.3. OBJECTIFS

L'objectif de cette thèse est de promouvoir la collaboration entre le nombre énorme et croissant de dispositifs mobiles et omniprésents (objets), favorisant ainsi l'informatique collaborative mobile et omniprésente dans le paradigme de l'Internet des objets. L'approche suivie pour réaliser cet objectif consiste à i) gérer efficacement la connexion de ces appareils via diverses technologies de communication, même sans l'utilisation de stations de base, et ii) ainsi améliorer l'exécution des tâches dans divers environnements difficiles. À cette fin, nous avons conçu un modèle architectural collaboratif omniprésent sur la périphérie des réseaux dans le paradigme de l'Internet des objets. Il est également constitué d'un réseau de dispositifs mobiles intelligents et omniprésents, notamment à la périphérie. En plus de faciliter et d'accélérer l'achèvement des tâches, il est destiné à remédier à la perturbation de la communication des réseaux, qui atteint les limites d'interruption dans différentes situations. L'objectif visé dans ce cas est le maintien de la connectivité dans ces réseaux via diverses technologies de communication peer-to-peer (P2P) (Wi-Fi Direct, Bluetooth) et sans fil (Wi-Fi) renforcées par d'autres technologies pouvant être utilisées comme stations de base privées.

Ceci peut être réalisé par Lopy4 de Pycom qui dispose d'une connectivité radio Wi-Fi, Bluetooth (compatible BLE), LoRa et Sigfox. Lopy4 peut être configuré en mode LoRa brut pour envoyer des paquets directement entre LoPy4. L'intégration de ces technologies fait partie de l'objectif de notre modèle de maintenir une connexion continue entre ses dispositifs atteignables dans les diverses circonstances, même les difficiles. Ainsi, notre objectif est de contribuer au renforcement des capacités et donc des performances des réseaux de périphériques mobiles en développant et en améliorant leurs structures en termes de logiciels, d'équipements et de technologies de communication. Aussi, notre objectif, comme susmentionné, est également d'accélérer l'achèvement des tâches en tirant parti du domaine de l'informatique collaborative [8-11]. Ceci en améliorant et en intégrant les dernières technologies d'équilibrage de charge et d'ordonnancement, qui doivent s'adapter au contexte de ces tâches, notamment à la périphérie des réseaux.

Les objectifs spécifiques de cette recherche sont :

1. Maintenir la connexion des appareils en diverses situations même les plus difficiles. (O1)
2. Rendre les réseaux et leurs utilisateurs plus autonomes. (O2)
3. Compenser le déchargement des dispositifs vers les serveurs Cloud et d'autres architectures intermédiaires et vice versa (c.à.d. le chargement depuis ces serveurs aux dispositifs), en créant et utilisant d’autres serveurs plus proches de la périphérie. (O3)
4. Améliorer la capacité des appareils à économiser de l'énergie et à accélérer l'exécution des tâches qui y sont effectuées grâce au développement de techniques d'ordonnancement et d'équilibrage de la charge. (O4)

Notre objectif premier dans cette ce présent travail est de contribuer à l'amélioration de l'informatique collaborative sur de vastes réseaux mobiles. L'idée principale est de proposer des solutions pratiques et efficaces conduisant à un environnement de calculs collaboratifs entre les différents dispositifs de ces réseaux.

Notre travail vise à améliorer la communication entre les dispositifs des réseaux et d'accroître le partage des ressources.

À cette fin, nous avons orienté notre étude sur la conception des solutions aux problèmes de collaboration entre utilisateurs, en tenant compte des limites de stockage et de la durée de vie de la batterie des dispositifs, tout en renforçant la puissance de calcul, qui est relativement faible.

I.4. MÉTHODOLOGIE

Nous proposons dans cette thèse une méthodologie de développement de moyens efficaces pour promouvoir les architectures omniprésentes émergentes à la périphérie. Notre objectif porte sur la proposition d'une approche intégrée pour concevoir une architecture omniprésente aux deux niveaux suivants:

1. La structure des appareils de technologie sans fil pour assurer la continuité de la connectivité et réduire la densité excessivité des communications.
2. L’architecture logicielle pour améliorer l’efficacité des applications qui y sont exécutées et aussi faciliter et accélérer leur utilisation.

Notre méthodologie de recherche s’est faite à travers une série d'étapes pour atteindre les objectifs spécifiques (O1, O2, O3 et O4) basés sur diverses technologies, techniques et algorithmes.

Nous avons suivi la technique de collecte de données et d’informations existant dans la littérature, les revues spécialisées et les sites Internet via leurs moteurs de recherche et les bibliothécaires. Quant à l'analyse des données, nous avons suivi une méthode d'analyse par liste de critères les plus importants qui influencent les performances, d'une part, les approches algorithmiques d'ordonnancement, et, d'autre part, les technologies de communication sans fil dans leur contexte d'utilisation. Nous avons effectué un ensemble d’expériences sur des données sous forme d’unités quantifiables pour sélectionner les meilleurs critères.

Nous avons utilisé le modèle architectural multicouche (Chapitre 2), les algorithmes d'ordonnancement (Chapitres 3 et 4), les technologies et protocoles de communication sans fil et radio P2P, l'architecture collaborative omniprésente inter et intra groupes d'objets des réseaux autonomes périphériques au sein de l'Internet des objets (IoT) (Chapitres 5 et 6). Ces étapes sont : 1) exploration du problème et de son contexte pour bien le définir, 2) cueillette et formalisation des connaissances pertinentes, 3) étude et analyse des solutions existantes, et 4) un cycle d’itération de trois étapes: i) conception des nouvelles solutions plus efficaces, ii) prototypage dans le cas d’architecture logicielle et iii) évaluation et validation pour établir une solution plus avantageuse.

I.4.1) EXPLORATION DU PROBLÈME ET DE SON CONTEXTE

Cette étape concerne l'architecture du réseau en termes d'appareils et de technologies de communication dans l'environnement dans lequel ils opèrent. Le but visé est d'identifier les composants visibles et cachés de ce système afin que toutes ses spécifications soient prises en compte.

I.4.2) RECUEIL ET FORMALISATION DES CONNAISSANCES PERTINENTES

Les spécifications et les capacités de chaque technologie de communication et de chaque appareil affilié et fonctionnant dans ce système sont décrites de manière formelle. Il en est de même pour les exigences que le système doit réaliser en tant que spécifications fonctionnelles, ainsi que pour les performances décrivant des contraintes temps réel. Les exigences sont ensuite traduites en conceptions des solutions sous la forme d'algorithmes et d'applications qui sont conçus, programmés, implémentés et évalués au cours des étapes suivantes.

I.4.3) ÉTUDE ET ANALYSE DES SOLUTIONS EXISTANTES

Après l'exploration des solutions existantes et en cours de construction, ces solutions sont étudiées et analysées sur la base des résultats de l'étude et de l'analyse des spécifications et exigences déjà collectées à l'étape précédente. En conséquence, les lacunes et les défauts de ces solutions sont identifiées et des conclusions tirées pour les remédier et les améliorer. Ainsi, des idées et des conceptions de nouvelles solutions plus efficaces seront proposées et réalisées dans les étapes suivantes.

I.4.4) CONCEPTION DE NOUVELLES SOLUTIONS, PROTOTYPAGE ET VALIDATION

Dans cette étape de notre méthodologie, les données obtenues au cours des premières étapes d'exploration, d'étude et d'analyse sont traitées dans un cycle d'itérations en trois parties (Figure I-1). La première partie est consacrée à la conception de nouvelles solutions. Dans cette partie, ces données et les idées qui s'en inspirent sont transformées et présentées sous forme de solutions plus efficaces, simples et rapides à mettre en œuvre. Le résultat se présente sous forme de solutions en termes de la création de la plate-forme physique des technologies de communication et de celle des logiciels. Au niveau de l'établissement de notre réseau de communication, les dispositifs qui fournissent une technologie de radiocommunication autonome avec des portées différentes, notamment celles qui sont relativement longues, sont recherchés et sélectionnés. Quant à la plateforme logicielle, les conceptions sont construites sous forme d'algorithmes. Puis, ces derniers sont transformés en prototypes implémentés par des applications. Par la suite, ces prototypes et plateformes de communications déjà établis sont soumis à une étape d’évaluation et validation, et continuent à itérer dans leur cycle jusqu'à leurs évaluations et validation.

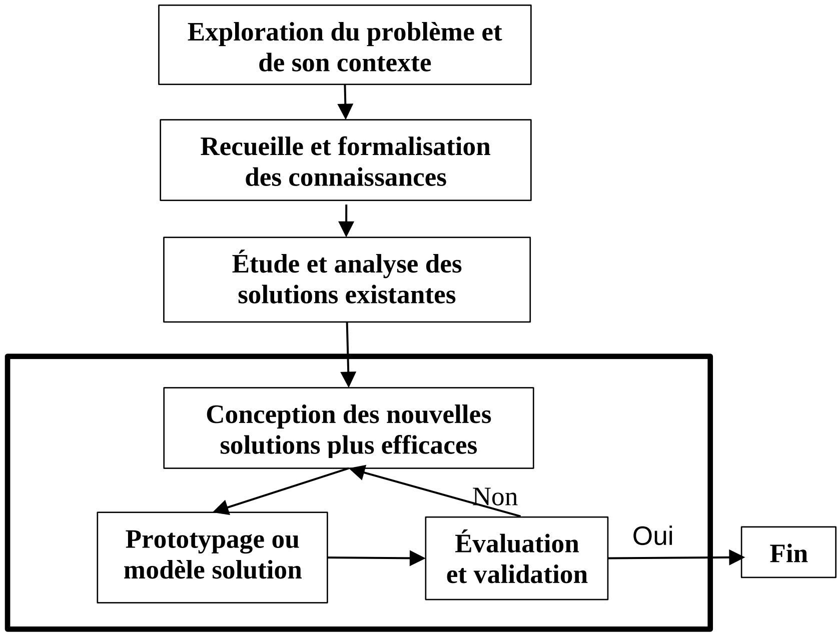


Figure I-1 Étapes de la méthodologie de recherche. © 2021 Ghassan Fadlallah.

I.5. CONTRIBUTION

Nous avons dans le cadre de cette thèse conçu, évalué, et validé un modèle collaboratif pour améliorer des architectures omniprésentes à la périphérie des réseaux dont les ressources de leurs objets sont limitées. L'efficacité de ce modèle architectural a été renforcée par l'introduction d'une structure de communication qui lui permet de fonctionner dans différentes circonstances et conditions. En plus, ce modèle adopte des stratégies de collaboration avec des algorithmes qui réduisent le temps d'achèvement des tâches.

Dans le cadre de ce modèle, une approche collaborative a été explorée et développée dans laquelle nous avons amélioré l'utilisation, la disponibilité et l'efficacité des ressources omniprésentes et des dispositifs mobiles. Le but est de concevoir et réaliser l'implémentation d'une stratégie de collaboration et des algorithmes d'ordonnancement de tâches dans des systèmes de dispositifs omniprésents collaboratifs.

L'apport de ce modèle est donc de maintenir la connectivité entre ces appareils dans diverses situations, notamment dans les environnements difficiles. En effet, les résultats de l'étude expérimentale que nous avons menée ont prouvé l'efficacité de ce modèle pour le maintien de la connectivité grâce à la technologie offerte par Lopy4 de Pycom, sur une portée de plusieurs kilomètres sous forme d'un réseau de trois sortes de fonctionnements du Lopy: Lopy émetteur, Lopy pont et Lopy récepteur. Comme travaux futurs, nous cherchons à intégrer dans ce modèle d'autres technologies modernes émergentes de radiocommunication qui pourraient être compatibles avec Lopy4 pour plus d'efficacité.

Notre contribution offre une approche dédiée adaptée aux dispositifs omniprésents largement utilisés dans le paradigme IoT, en particulier ceux à la périphérie des réseaux. En outre de maintenir la connectivité, cette approche a donné lieu à la conception d'un nouvel algorithme heuristique d'ordonnancement basé sur l'approche "Greedy". Elle l'a ensuite comparé à d'autres algorithmes heuristiques qui ont les mêmes caractéristiques et fréquemment utilisés dans la littérature. Les principaux avantages de cette approche sont mis en évidence par le partitionnement des tâches de grandes tailles en sous-tâches ayant des tailles proportionnelles aux capacités des dispositifs. Par la suite, chacun de ces deux ensembles a été regroupé en trois catégories: large, moyenne et petite, tout en tenant compte de la taille des sous-tâches et des capacités des appareils dans les sous-groupes de la même catégorie. Cela permet une recherche parallèle pour obtenir la solution souhaitée qui rend notre algorithme plus rapide dont les solutions générées sont de meilleure qualité. Par conséquent, notre approche permet l'économie d'énergie en plus d'améliorer la performance de ces appareils.

Dans cette thèse, nous avons conçu un modèle architectural collaboratif mobile et omniprésent permettant d’améliorer la performance de ce type d'architectures à la périphérie des réseaux de deux manières :

1. Renforcement de l'architecture de réseau mobile intelligent à la périphérie en termes d'équipements et de technologies et protocoles de communication modernes disponibles tels que les technologies de communication sans fil, la communication de poste à poste (P2P) et la technologie Lopy4 de Pycom pour une communication sans fil directe entre ces dispositifs. Ce réseau constitue une variété d'appareils tels que des capteurs, des smartphones et des tablettes, ainsi que de petits ordinateurs portables de grande capacité qui peuvent être utilisés comme des micro-serveurs locaux. Ces appareils peuvent être regroupés selon leurs étendues de connexions pour créer une série de groupes connectés.
2. Conception et implémentation des logiciels sous forme d'algorithmes et d'applications afin de : créer des groupes, sélectionner un gestionnaire de groupe, échanger des données et des messages entre les appareils. Ainsi que partitionner les grands travaux en petites tâches, ordonnancer ces tâches et équilibrer leur charge. Ces logiciels sont testés et validés via des simulations des situations de catastrophe, par exemple, en envoyant des messages d'un appareil à un autre en suivant le réseau de nœuds (appareils) le plus rapide à l'aide de notre propre mode de communication basé sur le Lopy4 de Pycom utilisé comme point d'accès privé.

I.6. ORGANISATION DE LA THÈSE

Le reste de cette thèse se compose de cinq chapitres. Chaque chapitre est un article publié dans une revue ou une conférence du domaine.

Dans le deuxième chapitre, nous passons en revue l'informatique mobile collaborative et son impact sur les performances et l'évolutivité des systèmes distribués et donc sur le développement de l'Internet des objets, ainsi que ses nombreuses architectures informatiques collaboratives telles que cloud, cloudlet, fog, edge, mobile edge, mobile cloudlet computing, etc. Ces architectures sont renforcées par les progrès rapides des technologies de communication et des dispositifs mobiles intelligents avec la tendance croissante à leur utilisation. Nous passons d'abord en revue leurs modèles architecturaux en couches en discutant les problèmes de leurs limites et de leurs défis. Par conséquent, nous introduisons un nouveau modèle architectural multicouche qui répond à ces problèmes et un scénario d'un cas d’urgence pour illustrer et mettre en évidence ses besoins et sa manière de fonctionnement à travers les différentes couches architecturales.

Le troisième chapitre, en guise d’une étude approfondie de la littérature, passe en revue les techniques et les algorithmes récents d'ordonnancement et d'équilibrage de charge publiés dans la littérature suivis d’une discussion critique.

Dans le quatrième chapitre, nous présentons un nouvel algorithme de type glouton ou vorace (Greedy en anglais) d'ordonnancement et d'équilibrage de charge pour les architectures collaboratives, notamment à la périphérie des réseaux dans l'IoT. Cet algorithme s'adapte aux capacités des appareils de ces réseaux et améliore leur efficacité.

Dans le cinquième chapitre, nous passons en revue les normes de protocoles de communication actuels et leurs rôles dans le renforcement des réseaux mobiles dans les différentes architectures de l'informatique collaborative au sein du paradigme de l'IoT. Plus précisément, nous avons illustré comment configurer et profiter d'une technique de maintien une connexion efficace même entre les dispositifs mobiles omniprésents, notamment à la périphérie. Ainsi, pour établir un modèle autonome omniprésent collaboratif à la périphérie des réseaux.

Le sixième chapitre est dédié à l'informatique collaborative omniprésente dans l'Internet des objets (IoT), dont les modèles architecturaux émergents souffrent toujours de limitations de capacité en termes de puissance de calcul, mémoire, stockage, connectivité, latence et bande passante. Par conséquent, l'objectif principal que nous nous sommes fixés est l’amélioration et le maintien de la connectivité ainsi que l’amélioration des performances de ces systèmes, tout en fournissant les services requis et attendus dans les environnements difficiles. Ceci est particulièrement important dans les situations de catastrophes et d’environnements extrêmes où la communication est utilisée pour faciliter et améliorer les opérations de sauvetage. Nous avons ainsi établi et complété le modèle architectural collaboratif omniprésent à la périphérie du réseau. C'est un modèle autonome et flexible qui prend en compte les limitations des ressources en tant que technologies de connectivité et dispositifs. L'étude expérimentale que nous avons conduite montre que ce modèle fonctionne efficacement même sans accès aux stations de base sans-fil, en exploitant les technologies de connexion P2P émergentes telles que Wi-Fi Direct et celles offertes via LoPy4 de Pycom.

La dernière partie de notre travail est la conclusion. Dans cette partie, nous avons résumé la thématique sur laquelle nous avons travaillé. Ensuite, nous avons présenté les principaux objectifs que nous nous sommes fixés, les résultats obtenus et esquissé les travaux futurs.

CHAPITRE II. MODÈLE ARCHITECTURAL OMNIPRÉSENT

Ghassan Fadlallah[1], Hamid Mcheick[1*], Djamal Rebaine[1]
[1]Département d'informatique et de mathématique, Université de Québec à Chicoutimi, QC G7H 2B1, Canada
*Correspondant : Hamid Mcheick (courriel: Hamid_Mcheick@uqac.ca).

Résumé : L'informatique mobile collaborative est aujourd'hui l'un des paradigmes de l'informatique les plus populaires en raison de son impact sur les performances et l'expansion des systèmes distribués et donc sur le développement de l'Internet des objets. En raison des progrès rapides des technologies de communication et des appareils mobiles intelligents avec la tendance croissante à leur utilisation, plusieurs architectures d'informatique collaborative mobile ont été émergés telles que cloud, cloudlet, fog, edge, mobile edge, mobile cloudlet computing, etc. Elles ont contribué efficacement à améliorer et organiser l'expansion de l'Internet des objets. Dans cet article, nous passons d'abord en revue les modèles architecturaux actuels en couches et discutons leurs limites et défis. Ensuite, nous présentons un nouveau modèle architectural des réseaux autonomes collaboratifs d'appareils mobiles utilisant la communication peer-to-peer qui peut être impliqué dans de nombreux domaines. Enfin, un scénario de situation d'urgence est présenté pour illustrer et mettre en évidence ces exigences telles que la stabilisation et le maintien de la connectivité en plus de l'ingénierie des données et services, ainsi que le déroulement du processus de sauvetage à travers les différentes couches d'architecture.

 Procedia Computer Science. 155. 201-209. 10.1016/j.procs.2019.08.030.

Préambule

Notre étude se concentre sur le développement de nouvelles normes de connectivité pour les appareils intelligents mobiles dont les ressources sont confrontées à de nombreuses limitations. Elle aborde également les problématiques de l'informatique collaborative mobile, en particulier à la périphérie des réseaux dans des architectures telles que fog - edge - mobile cloud computing, mobile cloudlet, smart multi-groups devices, etc.

L'objectif de notre étude est de contribuer à l'amélioration de l'informatique collaborative sur des réseaux mobiles autonomes d'appareils intelligents connectés via les technologies standards sans fil ou radio peer-to-peer comme Wi-Fi Direct.

Nous avons conçu un projet avec une architecture en couches pour faciliter et organiser à la fois les processus de sa construction et de son utilisation. Son architecture est constituée de couches suivantes : application de tâches, ingénierie des services comprenant deux domaines : les modules de calcul et d'opérations, ingénierie des données consistant en formatage et unification, adaptation et de la connectivité qui inclut l'infrastructure réseau et la communication.

L'article consiste à une introduction suivie par les sections suivantes: i) Revue de l'architecture IoT, ii) Travaux connexes de modèles d'architecture récents supportant l'Internet des objets (IoT), iii) Limites et défis des modèles d'architectures existantes, iv) Présentation de notre solution : Réseaux autonomes collaboratifs d'appareils mobiles utilisant la connectivité peer-to-peer, v) Architecture en couches proposée de notre approche, et vi) Conclusion.

LAYERED ARCHITECTURAL MODEL FOR COLLABORATIVE COMPUTING IN PERIPHERAL AUTONOMOUS NETWORKS OF MOBILE DEVICES

Ghassan Fadlallah[1], Hamid Mcheick[1*], Djamal Rebaine[1]

[a]University of Quebec at Chicoutimi boulevard de l'Université, Chicoutimi, G7H 2B, Canada

*Correspondant: Hamid Mcheick (courriel: Hamid_Mcheick@uqac.ca).

ABSTRACT Collaborative mobile computing is today one of the most popular paradigms of computing because of its impact on the performance and expansion of distributed systems and therefore on the development of Internet of Things. Due to the rapid progress in technologies of communication and smart mobile devices with the growing trend towards its use, many architectures of mobile collaborative computing have emerged to improve and organize the expand of Internet of Things, such as cloud, cloudlet, fog, edge, mobile edge, mobile cloudlet computing, etc. In this paper, we first review their current layered architectural models and discuss their limits and challenges. Then, we will present a new architectural model: Collaborative autonomous networks of mobile devices using peer-to-peer Communication that can be applied in many areas. Finally, a scenario of an emergency is presented to illustrate and highlight its requirements, such as supporting connectivity and engineering of data and services, and how the rescue process is addressed across different architecture layers.

Keywords: Smart devices, smart cities, communication technologies, communication challenges, peer-peer communication requirements, mobile collaborative frameworks, fog computing, Internet of Things (IoT).

II.1. INTRODUCTION

The rapid technological development in the fields of hardware and software for telecommunications and smart devices has necessitated the development of new communication standards adapted to resource-constrained devices in various areas of energy, communication and computing capabilities.

Recently, potential efforts are being made for reducing computing costs and increasing computing power in mobile devices that experiencing many limitations. This trend has witnessed industries and academia to significantly address issues of collaborative mobile computing at the networks periphery like fog, edge, mobile cloud, mobile cloudlet computing, smart multi-groups devices etc.

These technologies have differences in their architectural layers. Some of them have focused on certain structural phases, while others have found them to be unnecessary. These differences are evident as we move from centralized techniques to the periphery ones.

The objective of the present study is to contribute to the improvement of collaborative computing on autonomous or individual mobile networks of smart devices connected via standard wireless or Wi-Fi Direct technologies. This to make these networks more and more efficient and smart to meet the different requirements, especially urgent needs as in the case of disasters [3]. Therefore, we designed a layered architectural of this project to ease and organize both of its building and using processes.

In this context, our project focuses on layers of task application, service engineering including two fields: computing modules and operation, data engineering consisting of formatting and unification, adaptation and connectivity that includes network infrastructure and communication.

The rest of the document contains the following sections and a conclusion: A review of IoT architecture, Related works of recent architecture models supporting Internet of Things (IoT), Limits and challenges of existing architectures models, overview of our project: collaborative autonomous networks of mobile devices using peer-to-peer communication and layered architectural of our proposed approach.

II.2. IOT ARCHITECTURE REVIEW

The IoT paradigm, as a formal and abstract general framework, has no single consensus on architecture that is universally agreed upon. Several researchers have proposed different architectures. The earlier one defines the main idea of the IoT composed of three layers as follows [13, 14, 15]:

1) The perception layer as a physical layer, which is equipped with devices for detecting and collecting information about the environment in which it identifies other smart objects.
2) The network layer dedicated to connection to other objects and to the transmission and processing of devices data.
3) The application layer is responsible for providing the user by application specific services, in which IoT can be deployed, such as smart homes, smart cities and smart healthcare.

Another architecture more recent is the five layers architecture, which includes the processing and business layers additionally of the first one [14, 13–17]. The five layers are perception, transport, processing, application, and business layers. The perception and application layers have the same role as the architecture with three layers. The other layers are:

1) The transport layer transfers the data between the perception and processing layers via networks.

2) The processing layer or middleware layer stores, analyzes, and processes huge amounts of data from the transport layer. It can manage and provide a diverse set of services to the lower layers.

3) The business layer manages the entire IoT system, including applications, business and profit models, and user privacy.

Ning and Wang [14, 16] proposed another architecture inspired by the layers of processing in the human brain. It consists of three parts by analogous between (i) the human brain with the processing and data management unit or the data center, (ii) the spinal cord with the distributed network of data processing nodes and smart gateways, (iii) the network of nerves with the networking components and sensors.

A Reference Model of IoT architecture proposed by CISCO consists of 7 layers[1]:

1) Physical devices & controllers: the “things” in IoT.
2) Connectivity: communication & processing units.
3) Edge Computing: data element analysis & transformation.
4) Data accumulation: storage.
5) Data abstraction: aggregation & access.
6) Application: reporting, analytics and control.
7) Collaboration & processes: involving people & business processes.

[1] https://www.cisco.com/c/dam/global/en_ph/assets/ciscoconnect/pdf/bigdata/jim_green_cisco_connect.pdf

Based on these formal and general IoT structures the main companies operating in the field of information technologies have developed their own architectures that support and reinforce the IoT paradigm. This is illustrated by the emergence of the following structures: Cloud, Fog, Edge, Mobile cloud, Mobile Edge Computing, etc.

II.3. RELATED WORKS

From a new perspective Azam *et al.* [21] considered that "other than sensors and IoT nodes, smartphones are also going to be part of IoT." They illustrated a combined architecture (CoT) that is composed of IoT and cloud computing. They declared: "CoT will play an important role in this regard, not only in delivering the service, however, managing it." The CoT architecture as it is an integration of "IoTs and cloud data communication", consists of three layers: IoT layer, cloud layer and access layer. This architecture inherits cloud problems and it has many challenges at the levels of protocol support, energy efficiency, resource allocation, identity management, IPv6 (for identification of communicating objects) deployment, service discovery, quality of service provisioning, location of data storage, security and privacy and communication of unnecessary data.

Aliyu, *et al.* [6] focused on five-layered architecture of mobile cloud computing (MCC) including task application, perception, network infrastructure, Internet and communication, and computation layers.

The task application layer represents applications of resource consuming that includes: mobile healthcare, mobile learning, mobile commerce, mobile safety, mobile gaming and mobile social-media. This layer operates applications of management, service management, offloading decision module (ODM), service-based data management, authentication, and authorization.

Perception layer represents mobile devices (MDs), such as smartphones, personal device assistance (PDA), IPAD, laptops, etc. The perception layer is represented at physical layer of the OSI model. It also handles the heterogeneity of MDs in terms of communication. Further, deals with physical connection of the MDs.

Network infrastructure layer represents devices that used for internetworking/routing technologies, cloudlet devices and their functionalities in MCC. The infrastructure can serve as gateway for the MCC connection offloading. These gateway devices include cellular base stations, cellular satellite, access points etc.

Internet and Communications layer handles the different internet technologies for communication between MDs and the computation layer in cloud. It serves as a link that uses TCP, UDP and IP protocols for this communication.

Computation layer is represented by sophisticated datacenters, servers of conventional Cloud, and task offloading manager (TOM) and their functionalities in MCC. It is dubbed as Cloud layer and it also consists of the CC services with no mobility.

A popular technique is presented in [5] for extending the natural capabilities of mobile devices this is Code offloading. It migrates processor-intensive tasks to resource-rich surrogates. MobiCOP is a fully functional code offloading framework of Android devices. It offers implementation of all modules expected of such a system, including a remote execution environment, a decision-making engine, and a communication layer. The offloading operation is performed via this layer by implementing a suspend-wait-resume scheme on the client and contacting, through either Wi-Fi or Bluetooth, a server deployed on another mobile device MobiCOP's communication layer has been designed by considering the fact of mobility under unreliable network conditions to minimize traffic and power consumption.

The end-to-end (E2E) network slicing is considered in [4] as a foundation to support diversified 5G services and as a key to 5G network architecture evolution. It is stated in [4] that "based on network functions virtualization (NFV) and software defined network (SDN), physical infrastructure of the future network architecture consists of sites and three-layer data centers (DCs) (consist of computing and storage resources). Sites support multiple modes (such as 5G, LTE, and Wi-Fi) in the form of macro, micro, and pico base stations to implement the RAN real-time function." The central DC is the bottom layer that is closest in relative proximity to the base station side. The local DC is the middle layer, and the upper layer is the regional DC, with each layer of arranged data centers connected through transport networks. The authors proceed with a 3D layered architecture of IoT that consists of the following layers: physical, network communication, processing, storage, abstraction, service, application and collaboration with processes.

A generic multi-layer is presented in [1] as the ROS-JADE integration architecture for the manufacturing entity in the future factories. It can be represented as an individual cyber physical production system (CPPS). This work contributes to the definition of a generic multi-layer architecture for enabling the MARS (multi-agent robotic systems) social abilities in ATVs (autonomous transport vehicle) and, thus, fulfilling MR1 and MR2 (MR: main requirement).

This architecture comprises 4 layers: social, cognitive, operative and functional. The main goal of the higher layer is to offer the services of the ROS entity to other agents in the environment. The intermediate layers are designed for efficient execution, while the inner control of the robotic device is constituted in the lowest layer. "This layer (functional) deals with the basic ATV control (sensors, actuators, robotic algorithms...) and is implemented by means of a RF (robotic framework). The intermediate layers are responsible for abstracting the social behavior from the ATV functionality and, at the same time, they oversee pre-processing and storing the information needed at the social layer for achieving fast negotiation response.

Besides, these layers are also in charge of transmitting information and events between the functional and social layers. "

By leveraging fog computing aggregation of services as a driver for more sophisticated IoT applications, a new architecture has been developed, viz. the CoFog [2]. It is a service layer that provides ways to dynamically define and create services based on predefined templates. These services can be aggregated through formal mechanisms called operations. An operation represents a relationship between a given collaboration request and the services that can be used to fulfill that request. With mathematical formulas, a service (or more) can be composed, transformed or aggregated to dynamically create new services. There are two types of operations: conservative and non-conservative. The second type produces a new type of data. However, the first type of operation retains the same type of data. In this way a conservative operation can be applied recursively to obtain the desired results.

Aggregation services are important mechanisms in the CoFog architecture that allow it to discover and extract such services represented in a data sharing model. This model gives objects and applications connected to IoT the opportunity to discover the services offered in other Fog nodes. In the case where a fog node is unable to provide the requested services, it transmits the request to the neighboring nodes listed in its whitelist. In this way, any nearby Fog node can be used to satisfy the request. In terms of security, two of the main access control models have been studied and extended to integrate the collaboration aspects into the CoFog architecture: role-based access control and access control models based on attributes.

The architectural model of CoFog consists of the following layers: Adaptation, formatting & unification, operation and service where their main services are as follows:

The adaptation layer provides an abstract interface with the underlying resource infrastructure. In addition it provides generic means for defining virtual devices and objects.

The Formatting and unification layer provide methods for describing information and data filtering mechanisms. It generates a unified and consistent view for standardizing filtered data. In addition, it handles the heterogeneity of the infrastructure from a data semantic perspective.

The operations layer provides, through a combination of processes, a devices virtualization mechanism based on the request for dynamic and / or static services offered and contextual information.

The service layer provides, by leveraging the capabilities of the lower layers (such as, formatting, filtering, unification and virtualization of resources and data), a dynamic service orchestration to manage heterogeneity. In addition, it must offer an abstract and generic API to accelerate and facilitate the deployment of systems based on the Fog service.

A computational model of a multilayer architecture is described in [7]. It is dedicated for improving the performance of devices using mobile cloud computing. The research described in this work presents a computational model of a multilayer architecture for increasing the performance of devices using the mobile cloud computing paradigm. The main novelty of this work lies in the definition of a comprehensive model in which all the computing platforms that are available along the network layers are involved in outsourcing the application workload. It provides a generalization of the MCC (mobile cloud computing) field, which allows handling the complexity of scheduling tasks in such complex scenarios.

II.4. LIMITS AND CHALLENGES OF EXISTING ARCHITECTURES MODELS

IoT deals with a heterogeneous environment where the main requirements for its architecture are scalability, interoperability, openness, and modularity. This architecture should enable data analytics, easy and scalable management functionalities, cross-domain interactions, multi-systems integration, etc. There are numerous proposed IoT architecture models based on layered structures [19, 20]. Still, there is no common or general architecture, which provides full interoperability.

Through a specific study of current architecture models that allowed recognition of the range of methods that support the Internet of Things, it was possible, in the one hand, to better understand the tools, techniques, and methodologies to meet the developer's requirements. And, on the other hand, to solve real life problems by building and disseminating IoT powerful concepts. These are helpful to fill gaps within the current trends of architecture to fit the exact IoT power. So, numerous studies and researches have been carried out with the aim of finding possible solutions to integrate the gap in the current architectures of the Internet of Things to be at its full potential [18].

In this context, we highlight especially the efforts needed to bridge the gap in the Internet of Things in the emerging paradigm that results from the interaction and integration of mobile devices at the peripheries of Internet networks.

Our study of IoT existing architecture models was done through the framework of the approaches that support it. This study aims to, on the one hand, improve the understanding of related tools, technologies and methodologies to meet the requirements of users and developers, and, on the other hand, fill the lacuna in current architectural trends to assess the IoT power, especially at the networks edge.

The peripheral networks suffer on several levels more than other networks located closer to the service centers. The communication with these centers is according to the number of base stations or access points to cross. Especially, since this issue has the greatest impact on most other problems of these networks such as low capacity in response time power, computing and storage. Therefore, the establishment of new architectures is required to bring services closer to peripherals, to reduce access to the main cloud center. Thus, emerging architectures are created such as fog, edge, mobile edge, mobile cloud, cloudlet, etc. These architectures, which use communication technologies based on switches, routers, gateways, and base stations, have contributed to strengthen the collaborative mobile computing and the Internet of Things. This is realized, when communication is maintained, by reducing network congestion and enhancing its connection and consequently optimizing the latency and the tasks achievement times. However, these architectures still have limitations, among others, the lack of tasks unloading to central or peripheral servers [11], the loss of user mobility in case of cloudlet, the big number of hops (passing packets between devices, when the number is greater than 2) [10], the limitation of centers resources in capacity and budget which raises a great competition between applications [9] and the dysfunction of Wi-Fi. So, what happens if the connection fails when the base stations and access points are out of service due to a disaster or if it is not offered as in rural areas? This question is illustrated in Figure II-1.

In response to this question, some approaches, as mobile cloudlet and intelligent multi-groups architectures, are taking advantage of emerging direct communication technologies such as Wi-Fi Direct to bridge the gap in the collaborative mobile computing paradigm caused by the failure or interruption of the wireless connection based on access points or base stations. Unfortunately, these new architectures based on the principle of smart devices groups are having more problems and limitations. For example, intelligent multi-groups architecture does not have intergroup communication while mobile cloudlet supports two groups only. They don’t unload to Cloud. Their group owners are non-replaceable [8, 12]. Their resources may be limited in computation, storage and energy more than previous architectures capacity. Managing these limited capacity resources is a critical challenge requiring considerable effort in tasks scheduling and load balancing at the applications level.

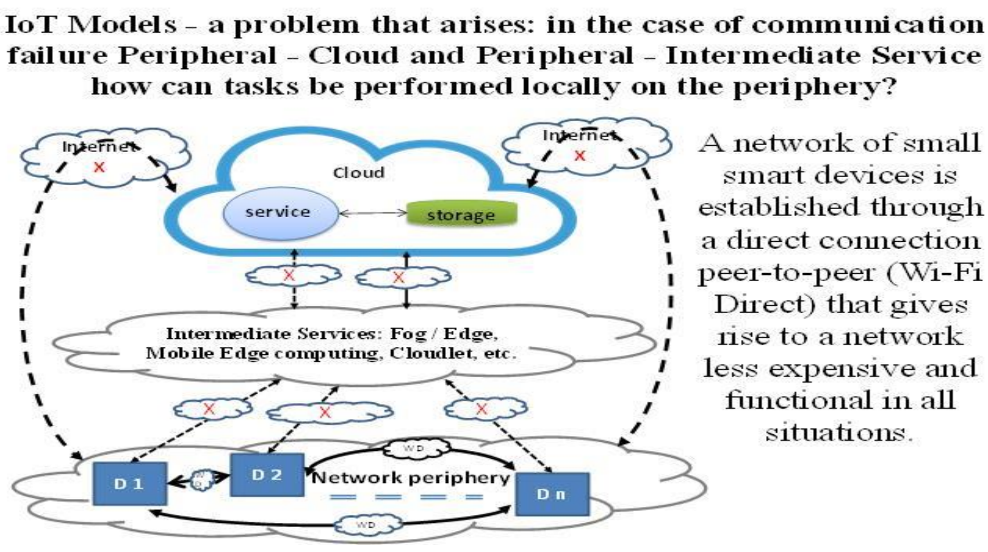


Figure II-1 Failure of cloud and intermediate services. © 2021 Ghassan Fadlallah.

II.5. AN OVERVIEW OF OUR PRESEARCH

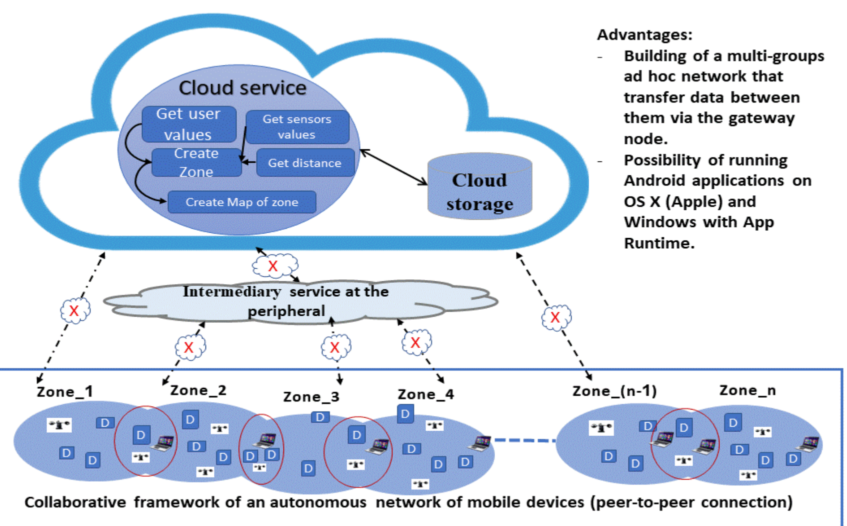


Figure II-2 Collaborative autonomous network using P2P connection. © 2021 Ghassan Fadlallah.

This project (see Figure II-2) aims to perform the needed tasks in any circumstances under extreme events (e.g.: natural disasters, extreme weather, vegetation in rural areas, etc.)

which can cause partial or total failures of networks and systems (in the event of disasters, Siberian attack, rural areas or failures in base stations). “This approach addresses the problems of the poor extent of the mobile network in the case of using a peer-to-peer connection. As well as the lacuna in performing relatively large size tasks using small smart devices.

This is partly due to partitioning, scheduling and distributing tasks [3].” Our approach should provide networks solutions, especially for zones on the edge of networks. These solutions will integrate Wi-Fi Direct, Wi-Fi Aware, Wi-Fi, and the Pycom Lopy4 technology to maintain and increase connectivity. In fact, where Wi-Fi Direct allows connection of intra-groups and not inter-groups, except tightly via gateway nodes between two adjacent groups, a circuit integrating Wi-Fi and Lopy4 makes it possible between group-owners for a few kilometers.

II.6. Layered architectural Model of our proposed approach

The autonomous mobile network at the edge is supposed to be a common platform supporting a wide range of application domains. This requires interoperability and resource virtualization capabilities to operate in each of these areas. In this context, our project focuses (see Figure II-3) on layers of (i) task application, (ii) service engineering including two fields: computing modules and operation, (iii) data engineering consisting on formatting and unification, (iv) adaptation and (v) connectivity that includes network infrastructure & communication.

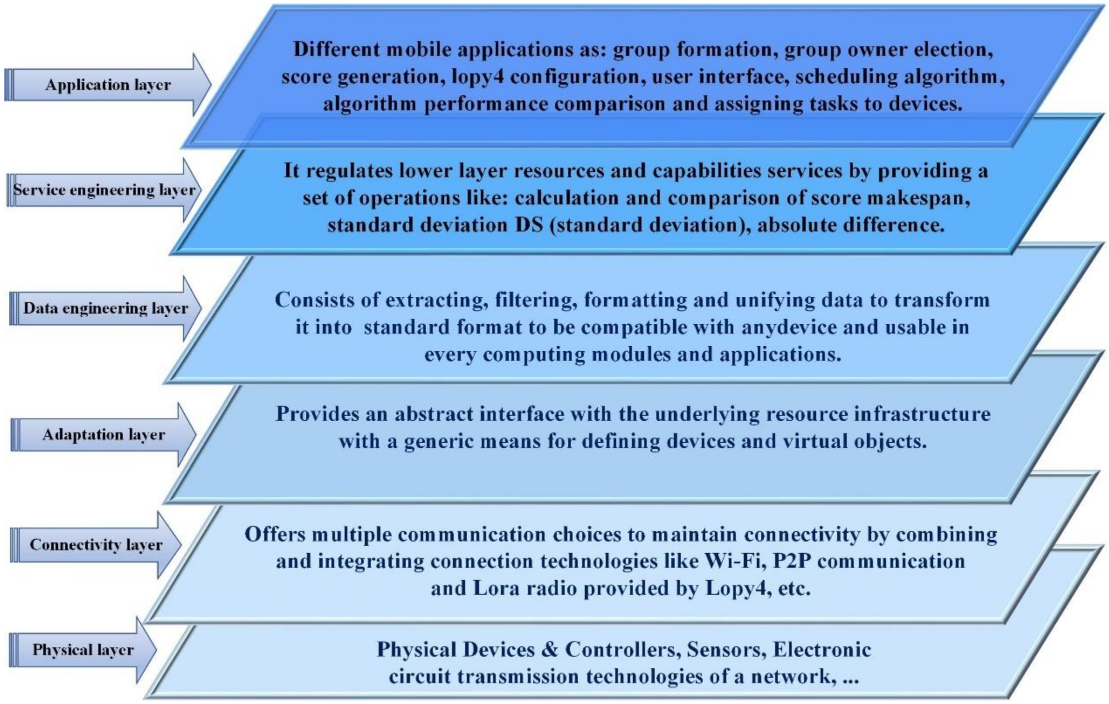


Figure II-3 Layered architectural of our proposed approach. © 2021 Ghassan Fadlallah.

i. Task application layer: The task application provides integration and connection to various mobile applications which may require significant resources for their own account. It creates user-specific applications that we need to code such as priority, security, creating device groups, finding the message path, etc. It allows tasks to be partitioned into small parts to scheduling and load balancing them on appropriate devices using other available applications.

ii. Service engineering layer: The autonomous mobile network architecture at the edge should offer a fast-lightweight service provided by computing and operational levels. It consists on leveraging lower layers functionalities such as formatting, filtering and unifying data as well as virtualizing resources. This is done by a set of operations applied to existing computing modules and operations to meet desired services.

iii. Data engineering layer: It consists of extracting, filtering, unifying and formatting data. Then, to transform data description into standard format (YAML, JSON, CSV, HDF5, etc.) to be compatible with any device and usable in every computing module and application.

iv. Adaptation layer: It provides an abstract interface with the underlying resource infrastructure. It provides generic means for defining devices and virtual objects.

v. Connectivity layer: It consists of the ability of different manufacturing entities (machines, robots, warehouses, operators, etc.) to communicate with each other. It offers several communication choices to keep the connection like Wi-Fi, Wi-Fi Direct, Wi-Fi Aware with different connection technologies that increases the connection range of devices like Lopy4, etc.

II.6.1) A SCENARIO OF A CASE STUDY

In a harsh environment, when the wireless internet (Wi-Fi, LTE' etc.) and cell phone connection services are shutdown, unavailable or failed, the emergency monitoring office establishes an urgent communication line, via an alternative technology such as Wi-Fi Direct, with the police, ambulance, and firefighter vehicles' drivers, which are under service to evacuate an area at risk. It sends them a message containing, in addition to the necessary data captured by sensors, the latitude and longitude of this area. This to explore their paths towards this place and return to the office their information as well as some sensors data. Thus, they can choose which of them can effectively involve in this task. This requires (see Figure II-4), on the one hand, to obtain the coordinates of this area with sensors indications and to convert them to the standard data format. Then to filter the data of the drivers to choose the destinations of the messages to be sent and finally select the abstract communication interface with them. On the other hand, each driver extracts the information from the office message, reads its coordinates and the indications of the captured sensors. Then, he converts them to the standard format and chooses the appropriate application to perform the necessary calculation and afterward sends this information to the office. This requires establishing communication by means of an alternative wireless connection technology, for example, Wi-Fi Direct or Bluetooth reinforced by Lora wireless technology through Lopy4.

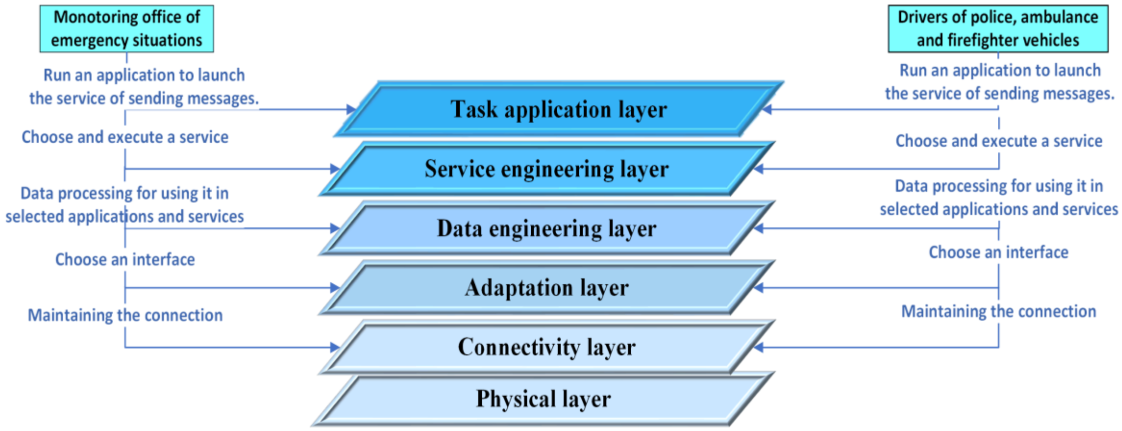


Figure II-4 A scenario of how our proposed approach works. © 2021 Ghassan Fadlallah.

CONCLUSION

In this paper, a layered architectural design model and associated works acting in different fields of collaborative mobile computing are presented. This model can help and be applied in many areas such as smart city, smart home, police office, military management, smart parking, etc. In addition, it facilitates their realization and use.

This layered architectural design model adopts appropriate and effective communication technologies to establish and maintain the connection in extreme events. In this way, we presented our contribution to solve the problem of communication between different group owners in Wi-Fi Direct, as they have the same Internet Protocol (IP) addresses: 192.168.49.1. This is realized by integrating technologies such as Lopy4 into a Wi-Fi connection circuit to create our special chain of Wi-Fi hotspot; where each one is a few kilometers away from the other. In future work, many applications and services of layers in this architecture model will be developed in different areas. They are intended to establish, in the case of wireless connection failure, an autonomous network between the multi-groups of mobile devices.

REFERENCES

[1] J. Martin, O. Casquero, B. Fortes, M. Marcos. “A generic multi-layer architecture based on ROS-JADE integration for autonomous transport vehicles,” *Sensors* 2019, vol. 19, no. 1, art. no. 69, Mar. 2019, doi: 10.3390/s19010069.

[2] J. Abdelaziz. « Un cadre architectural pour la collaboration dans l’internet des objets ; une approche basée sur l’informatique en brouillard, » Thèse de doctorat, Université du Québec à Chicoutimi, 2018.

[3] G. Fadlallah, Dj. Rebaine, H. Mcheick. “Scheduling problems from workshop to collaborative mobile computing: A state of the art,” *international journal of computer science and information security (IJCSIS)*, vol. 16, no. 1, Jan. 2018.

[4] O. Vermesan, M. Eisenhauer, M. Serrano, P. Guillemin, H. Sundmaeker, E. Tragos, J. Valino, B. Copigneaux, M. Presser, A. Aagaard, R. Bahr, E. Darmois “The next-generation internet of things: Hyperconnectivity and embedded intelligence at the edge,” in *next generation internet of things distributed intelligence at the edge and human machine-to-machine cooperation*, River publishers, pp.19-102, 2018.

[5] J. I. Benedetto, G. Valenzuela, P. Sanabria, A. Neyem, J. Navón, C. Poellabauer “MobiCOP : A scalable and reliable mobile code offloading solution,” *wireless communications and mobile computing*, nol. 2018, article ID 8715294, 18 pages, 2018. [Online]. Available: https://doi.org/10.1155/2018/8715294

[6] A. Aliyu, M. Tayyab, A. H. Abdullah, U. M. Joda, O. Kaiwartya. “Mobile cloud computing : Layered architecture,” 2018 *seventh ICT international student project conference (ICT-ISPC)*, Publisher: *IEEE,* pp. 1-6, 2018.

[7] H. Mora, F. J. Mora Gimeno, M. T. Signes-Pont, B. Volckaert. “Multilayer architecture model for mobile cloud computing paradigm,” *Complexity*, Vol. 2019, art. 3951495, pp. 1-13, 2019.

[8] D. Fesehaye, Y. Gao, K. Nahrstedt, G. Wang "Impact of cloudlets on interactive mobile cloud applications," In: *IEEE 16th international enterprise distributed object computing conference (EDOC)*. IEEE; Beijing, China, pp. 123-32, 2019.

[9] X. Wu, R. Dunne, Q. Zhang, W. Shi. "Edge computing enabled smart firefighting: opportunities and challenges," *in proceedings of HotWeb'17*, San Jose / Silicon Valley, CA, USA, pp. 1-6, Oct. 14, 2017.

[10] A. A. Mehanna, M. I A. Abdel-Fattah, S. Abdel-Gaber. "M.Cloudlet: A mobile cloudlet model using Wi-Fi Direct," *international journal of computer science and information security (IJCSIS)*, vol. 14, no. 11, pp. 972-979, Nov. 2016.

[11] "Number of mobile phone users worldwide from 2015 to 2020," Statista research department, Nov 23, 2016. [Online]. Available: https://www.statista.com/statistics/274774/forecast-of-mobile-phone-users-worldwide/

[12] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, M. Ayyash. « Internet of things: A survey on enabling technologies, protocols and applications, » in *IEEE communications surveys tutorials*, vol. 17, no. 4, pp. 2347-2376, 2015.

[13] I. Mashal, O. Alsaryrah, T.-Y. Chung, C.-Z. Yang, W.-H. Kuo, and D. P. Agrawal. "Choices for interaction with things on Internet and underlying issues," ad hoc networks, vol. 28, pp. 68- 90, 2015.

[14] P. Sethi, S. R. Sarangi. "Internet of Things: Architectures, protocols, and applications", *journal of electrical and computer engineering*, vol. 2017, pp.1-25, 2017.

[15] M. Wu, T.-J. Lu, F.-Y. Ling, J. Sun, and H.-Y. Du. "Research on the architecture of internet of things," in *proceedings of the 3rd international conference on advanced computer theory and engineering (ICACTE '10)*, *IEEE*, Chengdu, China, vol. 5, pp. 484-487, Aug. 2010.

[16] H. Ning and Z. Wang. "Future internet of things architecture: like mankind neural system or social organization framework?" *IEEE communications letters*, vol. 15, no. 4, pp. 461-463, 2011.

[17] R. Khan, S. U. Khan, R. Zaheer, and S. Khan. "Future internet : The internet of things architecture, possible applications and key challenges," in *proceedings of 2012 10th international conference on frontiers of information technology (FIT 12),* pp. 257-260, Dec. 2012.

[18] P. P. Ray. "A survey on Internet of Things architectures," *Journal of king saud university - computer and information sciences*, vol. 30, no. 3, pp. 291-319, Jul. 2018.

[19] A. Čolaković, M. Hadžialić "Internet of things (IoT): A review of enabling technologies, challenges, and open research issues," *Computer networks*, vol. 144, pp. 17-39, 24 Oct. 2018.

[20] C. Sarkar, S.N Akshay Uttama Nambi, R.V. Prasad , A. Rahim , R. Neisse , G. Baldini. "DIAT: a scalable distributed architecture for IoT," in *IEEE Internet of Things journal*, vol. 2, no. 3, pp. 230-239, Jun. 2015, doi: 10.1109/JIOT.2014.2387155.

[21] M. Aazam, E.-N. Huh, M. St-Hilaire, Ch.-H. Lung, I. Lambadaris. "Cloud of things: Integration of IoT with cloud computing," Springer international publishing, pp. 77-94, 2016.

CHAPITRE III. PROBLÈMES D'ORDONNANCEMENT : UN ÉTAT DE L'ART

Ghassan Fadlallah, Djamal Rebaine, Hamid Mcheick
Département d'informatique et de mathématique, University of Quebec at Chicoutimi
555, boul. De l'Université, Saguenay, Québec, Canada G7H 2B1
Ghassan-mustapha.fadlallah1@uqac.ca Djamal_Rebaine@uqac.ca Hamid_Mcheick@uqac.ca

Résumé : Cet article passe en revue la littérature la plus récente reliée aux problèmes d'ordonnancement. La motivation de ce travail vient de la nécessité d'avoir, dans un article, une idée globale de ces problèmes avec une vue approfondie des tendances de recherche impliquées. Plusieurs schémas d'ordonnancement sous différentes contraintes et critères d'optimisation sont discutés. Nous avons constaté que le développement technologique rapide au niveau des machines et équipements s'accompagne d'une utilisation intensive de ces appareils. Cela nécessite l'enrichissement et l'amélioration des algorithmes d'ordonnancement où la tendance s'oriente de plus en plus vers les algorithmes heuristiques et approximatifs. Comme les techniques d'ordonnancement s'étendent des ateliers aux secteurs cloud, fog et edge de l'informatique mobile collaborative, nous disputons qu'ils n'ont pas encore été utilisés efficacement dans son troisième segment : les réseaux mobiles autonomes. Ces réseaux peuvent jouer le rôle le plus efficace, dans des situations catastrophiques, pour surmonter le problème du trafic de communication téléphonie/internet avec le coût le moins cher ou gratuit. Nous visons à attirer l'intention d'orienter la recherche sur les problèmes d'ordonnancement vers ce segment de l'informatique mobile collaborative qui devient indispensable dans ces cas (par exemple, les inondations en Oregon, les tremblements de terre, les attaques terroristes, etc. lorsque presque tout est endommagé ou inaccessible, sauf nos petits dispositifs mobiles et ressources omniprésentes).

Contribution associée :

Ghassan Fadlallah, Djamal Rebaine, Hamid Mcheick «Scheduling problems from workshop to collaborative mobile computing: A state of the art» chez International Journal of Computer Science and Information Security (IJCSIS), vol. 16, no. 1, pp. 47-69, January 2018.

Préambule

Ce chapitre consiste en un article publié dans un journal. Il se compose de la littérature récente sur le sujet de l'ordonnancement. Ce travail vise à rassembler, dans cet article, les idées de recherche les plus importantes qui ont abordé ce sujet. Ensuite, extraire un regard approfondi sur leurs tendances pertinentes et les orientations futures dans le domaine.

Vu que le développement technologique des machines et des appareils augmente leur utilisation intensive, alors l'enrichissement et l'amélioration des algorithmes d'ordonnancement est la solution efficace pour renforcer leurs performances. Tandis que leur tendance est de plus en plus vers les algorithmes heuristiques et approximatifs.

Les techniques d'ordonnancement s'étendent des ateliers aux secteurs cloud, fog et edge computing et autres segments. Nous visons à attirer l'intention d'orienter la recherche dans celui des réseaux mobiles à la périphérie. Ce sujet est traité dans l'article comme suit :

Les sections 2, 3 et 4 présentent respectivement des « surveys » et des articles récents sur les problèmes d'ordonnancement dans les domaines de l'industrie, le cloud et edge computing, et les réseaux mobiles à la périphérie. La section 5 met en évidence la discussion des idées, suggestions et tendances les plus importantes dans ces « surveys » et articles. Dans la section 6, nous concluons les articles examinés en termes de la tendance récente de l'ordonnancement qui s'oriente de plus en plus vers les algorithmes heuristiques et approximatifs.

SCHEDULING PROBLEMS FROM WORKSHOP TO COLLABORATIVE MOBILE COMPUTING: A STATE OF THE ART

Ghassan Fadlallah, Djamal Rebaine, Hamid Mcheick
Département d'informatique et de mathématique, University of Quebec at Chicoutimi
555, boul. De l'Université, Saguenay, Québec, Canada G7H 2B1
Ghassan-mustapha.fadlallah1@uqac.ca Djamal_Rebaine@uqac.ca
Hamid_Mcheick@uqac.ca

Abstract This survey reviews the latest literature related to scheduling problems which is closely related to load balancing problems. It is noted that they are often used with the same meaning. In fact, it is not efficient to use one without the other. This is because the scheduling problem is to determine the order of tasks execution on available devices, while load balancing seeks to balance these tasks between these devices. The motivation of this work comes from the need to have, in one paper, a comprehensive idea of these problems with an in-depth view of the involved research tendencies. Several scheduling schemes under different constraints and optimization criteria are discussed. We observed that the rapid technological development at the level of machinery and equipment is accompanied by intensive use of these devices. This requires the enhancement and improvement of scheduling algorithms, and the tendency is more and more towards the heuristic and approximate algorithms. As the scheduling schemes range from workshops to Cloud, Fog and Edge computing segments of the collaborative mobile computing, we argue that they have not yet been used effectively in its third segment: individual mobile networks. These networks can play the most effective role, in catastrophic situations, to overcome the problem of telephony/internet communication traffic with the cheapest or free cost. We aim to motivate research on scheduling issues to this segment of collaborative mobile computing that becomes indispensable in urgent these cases as: Oregon, floods, earthquake, terrorist attacks, etc., when almost everything is damaged or not accessible except our small mobile devices and ubiquitous resources.

III.1. INTRODUCTION

Accelerated technological development constantly imposes real challenges in various fields, especially in mobile networks because of the rapid advances in computer architecture, mobile devices and wireless communications. This has led to the transition from big devices network (desktops and laptops) to small mobile devices connected by high bandwidth wireless. The advances in hardware and software technologies have sparked increased interest in the use of these mobile devices within the large scale parallel and distributed systems in different fields such as databases, defense, real-time and commercial applications. The performance of any system designed to operate a large number of devices depends on the tasks scheduling satisfying the workload distribution across these devices.

Scheduling involves allocating resources over time to perform a collection of tasks [4, 6]. The need of scheduling started first in factories and industries before becoming a de facto technique in multiprocessor computers.

Due to the explosion of mobile and wireless technologies, and despite the fourth generation (and soon the fifth) deployment of wireless communication systems, several challenges still remain to be solved. These challenges include the spectrum crisis, high energy consumption, the ever-increasing demand for high data rates and the mobility required by new wireless applications.

The latest technological advances in this area can be a solution to the intensive use of applications on mobile devices. As this situation has captured users, because it gives them freedom in terms of place of work and time to pursue their jobs and interests. This increases the demand for using these devices. Unfortunately, the rate of this growing demand is still higher than the technological growth. Therefore, this reinforces the trend of improving scheduling and load balancing techniques.

According to Leung and Anderson in [1], a scheduling process involves modeling a range of different environments which differ in the way the information is released. They distinguish paradigms: the static scheduling, when all jobs with related information are available at the start of the horizon, and the dynamic scheduling where jobs have different release or available times. The authors pointed out that the decision maker must optimize (usually minimize) a given objective function. There are different categories of policies classified generally into: a) the class of static policies; when the decision maker has to specify at the outset all actions to be taken during the evolution of the process, b) the class of dynamic policies; decisions are made at any time as a function of all the information that has become available up to that point.

In scheduling paradigm, a distinction is also made between i) offline deterministic scheduling, ii) stochastic scheduling, and iii) online scheduling [1,7,8,9].

In offline deterministic scheduling, all information or data with regard to the problem is known *a priori* including: the number of jobs, their release dates, due dates, weights, etc. The resulting problem is known as a combinatorial optimization problem subject to some given constraints.

In stochastic scheduling the number of jobs is fixed and known in advance. However, most or all of the parameters describing a job such as processing times, release dates and due dates are considered as random variables from known distributions.

In online scheduling, there is even less information known before hand, it is released gradually to the decision maker. The decision maker knows nothing in advance about the release dates or processing times.

To summarize, offline deterministic scheduling deals with perfect information, stochastic scheduling with input that is partially stochastic, while online deterministic scheduling deals with input that is known gradually as it arrives to the system. However, a model which mixes the above models may also be the subject of interesting studies. Indeed, Vredeveld [2] addressed the stochastic online scheduling (SOS) model. In this model, jobs arrive in an online manner and as soon as a job becomes known, the scheduler only learns about the probability distribution of the processing time and not the actual processing time. Both online scheduling and stochastic scheduling are special cases of this model.

Let us note that deterministic scheduling models are based on predictive approaches that do not take into account the presence of disturbances and evaluate the scheduling solution in terms of estimated data. In practice, however, this sort of scheduling becomes quickly unfeasible and returns poor performance. Indeed, in practice, scheduling environments are usually subject to significant amounts of randomness.

As a result it is not of interest to spend an enormous amount of time figuring out a supposedly optimal solution when within a few hours random events will change the structure of the problem or the list of jobs [5]. So, the hypothesis of determinism of scheduling problems is considered as restrictive, and the problem of scheduling with uncertainty management has been raised and is of interest to several researchers [3]. This has led to motivate research in the dynamic scheduling methods which consist in (re)allocating resources at run time [128] *i.e.*, make decisions in real time given the state of resources and the progress of different tasks over time[2]. These methods use approaches other than the predictive ones, which until recently were known as proactive, reactive and hybrid which includes two sub-types: predictive-reactive and proactive-reactive approaches. The proactive approach computes by anticipation a scheduling solution by taking into account *a priori* knowledge about probable uncertainties. The reactive approach, another on-line approach, builds real-time scheduling solutions by taking into account any kind of uncertainty that may arise. Finally, it is possible to combine on-line (reactive) and off-line (proactive) approaches in order to get the advantages offered by the two models [17].

III.1.1) DEFINITIONS AND NOTATIONS IN THE SCHEDULING PROBLEMS SCHEME

The characteristics of jobs may be of the following: preemptive or non-preemptive, resumable or non-resumable, independent or linked. The latter is represented usually by a precedence graph. This graph is a directed acyclic graph that specifies the precedence constraints between tasks execution [136]. A job is called non-preemptive if the processing on this job which is assigned to a machine is processed until its completion. On the other hand, if this processing is interrupted before its completion and reassigned to either the same machine or some other machine, that type of job is called preemptive [10].

[2] Groupe d'Ordonnancement Théorique et Appliqué (GOThA)

A job is said to be resumable, if it has been interrupted due to a machine non-availability period, it can be resumed without needing to be restarted after the machine becomes available again. However, the non-resumable job has to be restarted every time it is interrupted [135].

Polynomial time [155]: The time complexity of an algorithm is said to be of polynomial time if the running time of this algorithm is O(p(n)), where p(n) is a polynomial and n is the size of the input of the problem being solved.

The setup time is defined in [20] as the time required to prepare the necessary resource, as machines or people, to perform a task or job operation. The setup cost is the cost to set up any resource used prior to the execution of a task (for more details see e.g. Allahverdi and Soroush [20]).

In the reminder of this paper, we will be using the following notations [127]: C_i, $F_i = C_i - r_i$, $L_i = C_i - d_i$, w_i, $T_i = \max(0, L_i)$, $E_i = \{d_i - C_i\}$, r_i, p_i, d_i are respectively completion time, flow time, lateness, relative weight, tardiness, earliness, release date, processing time, due date of job i. Table III-1 summarises the most used criteria of performance to evaluate the quality of a scheduling solution.

Tableau III-1 THE PERFORMANCE MEASURES

Measure	Max	Total	Total weighted	Average	Average weighted
Completion time C_i	$C_{max} = \max_{1 \le i \le n} C_i$	$C_{total} = \sum_{i=1}^{n} C_i$	$wC_{total} = \sum_{i=1}^{n} w_i C_i$	$C_{average} = \frac{1}{n}\sum_{i=1}^{n} C_i$	$wC_{average} = \frac{1}{n}\sum_{i=1}^{n} w_i C_i$
Flow time $F_i = C_i - r_i$	$F_{max} = \max_{1 \le i \le n} F_i$	$F_{total} = \sum_{i=1}^{n} F_i$	$wF_{average} = \frac{1}{n}\sum_{i=1}^{n} w_i F_i$	$F_{average} = \frac{1}{n}\sum_{i=1}^{n} F_i$	$wF_{average} = \frac{1}{n}\sum_{i=1}^{n} w_i F_i$

Lateness $L_i = C_i - d_i$	$L_{max} = \max_{1 \le i \le n} L_i$	$L_{total}= \sum_{i=1}^{n} L_i$	$wL_{average}= \frac{1}{n}\sum_{i=1}^{n} w_i L_i$	$L_{average}= \frac{1}{n}\sum_{i=1}^{n} L_i$	$wL_{average}= \frac{1}{n}\sum_{i=1}^{n} w_i L_i$
Tardiness $T_i = \max(0, L_i)$	$T_{max} = \max_{1 \le i \le n} T_i$	$T_{total}= \sum_{i=1}^{n} T_i$	$wT_{average}= \frac{1}{n}\sum_{i=1}^{n} w_i T_i$	$T_{average}= \frac{1}{n}\sum_{i=1}^{n} T_i$	$wT_{average}= \frac{1}{n}\sum_{i=1}^{n} w_i T_i$
Earliness $E_i = \{d_i - C_i\}$	$E_{max} = \max_{1 \le i \le n} E_i$	$E_{total}= \sum_{i=1}^{n} E_i$	$wE_{average}= \frac{1}{n}\sum_{i=1}^{n} w_i E_i$	$E_{average}= \frac{1}{n}\sum_{i=1}^{n} E_i$	$wE_{average}= \frac{1}{n}\sum_{i=1}^{n} w_i E_i$

Competitive ratio is the way to evaluate the performance of an online algorithm. The idea is to evaluate the quality of an on-line algorithm compared to an algorithm that receives the complete information. A competitive ratio is defined as [156]:

$$\max_{e \in E} \frac{Cost\ of\ executing\ the\ plan\ that\ does\ not\ know\ e\ in\ advance}{Cost\ of\ executing\ the\ plan\ that\ knows\ e\ in\ advance}$$

Scheduling problems may be solved using either meta-heuristic algorithms or heuristic algorithms [157] *i.e* iterative or constructive methods that may deliver approximate solutions within a reasonable time. The popular ones are as follows:

- Genetic algorithms are initially developed to meet specific needs in biology. In the context of combinatorial optimization applications, an analogy is developed between an individual in a population and a solution of a problem in the global solution space [157].
- The simulated annealing method; used in metallurgy to improve the quality of a solid and seeks a state of minimal energy that corresponds to a stable structure of the solid. The simulated annealing method is designed to solve local minima problems. [158].
- PSO (Particle Swarm Optimization algorithm) is a cooperative, population-based global search swarm intelligent metaheuristic, presented by Kennedy and Eberhart in 1995 [33]. This is a powerful optimization technique for solving multimodal continuous optimization problems [34]. In addition it is a population based stochastic optimization technique which has become popular due to its effectiveness and low computational cost.

- Longest processing time rule (LPT): Jobs with large processing time values are prioritized for scheduling. So, tasks are organized in descending order of their processing times. Shortest processing time rule (SPT): Jobs with small processing time values are prioritized for scheduling. So, tasks are organized in ascending order of their processing times [7]

Contingent schedule: “a contingent schedule allows different task resource assignments depending on how the execution of the schedule has proceeded so far. A contingent schedule can be viewed as a tree that assigns a possibly different task resource assignment for every possible execution [117].”

III.1.2)CONTEXT OF THE RESEARCH TOPIC

The main objective in the machines’ scheduling theory is to find the best solutions everywhere in that broad scope with various areas, such as production, medical, military, informatics services and telecommunications, etc. This scheme aimed to adopt different classifications of these problems each of which has been named, over the years, by different names among the following: static, deterministic, predictive, offline, stochastic, dynamic, proactive, reactive, online, etc.

In the area of mobile networking, cloud computing and fog or edge computing, which are the recent emerging domains of parallel machines scheduling, it is well justified to focus our study on the multi-machine model, dedicated machines and parallel machine model [7,10].

The parallel machine models are usually classified as follows:

1. Identical parallel machines: all the machines are identical in terms of their speed. Every job will take the same amount of processing time on each of the machines.
2. Uniform (proportional) parallel machines: the machines have different speeds, whereas for each job, its processing times on the machines are inversely proportional to the speeds of those parallel machines.

3. Unrelated parallel machines: the machines have different speeds with arbitrary processing times of jobs. In this type of scheduling, there is no relation amongst the processing times of a job on the parallel machines.

Researches on scheduling problems have expanded over decades. These researches differ according to the perspective from which they are based, since there is no common perspective to address this issue. In fact, many surveys have been published over the last few years treating these problems of several points of view.

This paper is an attempt towards a comprehensive overview of dealing with scheduling problems based on a significant number of surveys and individual articles that have been published in the literature over the past decade.

This survey will focus on scheduling problems with multiple parallel machines or devices. This is motivated by the fact that the problem of scheduling, as mentioned above, has become, according to the accelerated technological progress, an evident requirement in recently emerging fields, such as modern factories, mobile networks, cloud computing, fog computing, smart cities, etc.

The remainder of this paper is organized as follows. Section 2, 3 and 4 present respectively recent surveys and articles on scheduling problems in i) industry, ii) cloud and edge computing, and iii) mobile networks. These sections are presented in two forms: the first details some important works, while the second summarizes other surveys in a table illustrating their most important characteristics such as: reference, scheduling area, context, scheduling environment, conclusion, suggestions and tendencies. Section 5 highlights the discussion of the most important ideas, suggestions and tendencies in these surveys. In Section 6 we conclude the examined papers in terms of the recent scheduling trends in emerging areas such as cloud, edge / fog and mobile computing.

III.2. SURVEYS AND ARTICLES ON CLASSICAL AND INDUSTRIAL SCHEDULING PROBLEMS

Many research algorithms and surveys have been published in the area of scheduling problems. In this section, we analyse and discuss briefly these works to understand the different approaches and algorithms used for their resolutions.

Saidy *et al.* [11] studied scheduling problems under various constraints such as activity duration, release dates, due dates and precedence constraints, and the availability of resources (or machines) that could affect these problems. In fact, these machines might not be available during certain time periods called holes for convenience [12]. Their study is based on the classification of these problems into two classes: deterministic and stochastic.

In this survey [11], Saidy *et al.* presented machine scheduling problems that have been studied in the literature with availability constraints in the resumable, semi-resumable and non-resumable cases, within different environments: single machine, parallel machines, flow shop, job shop, open shop, flexible flow shop and flexible job shop. They mentioned papers dealing with single machine and also parallel machine problems, having resumable, non- resumable and “crossable” availability constraints. They have defined as “crossable” the unavailability period that allows an operation to be interrupted and resumed after a period of time. Although, an unavailability period that prevents the interruption of any operation, even if the operation is resumable, as “non-crossable”.

Saidy *et al.* stated that in stochastic models the following parameters are not known before time: the processing times, the release dates, the starting time and the duration of the unavailability period. However, they assumed that the distributions of the processing times, due dates (deadlines), repair time and time at which breakdown occurs are known at time 0.

The uptime and downtime of machines and the jobs processing requirement are assumed to be independent identically distributed random variables.

They pointed out that heuristics are used for problems in both cases: when machines are continuously available and when breakdowns occur. At their days, they considered that this document may be a good reference for those interested in sequencing and job scheduling issues in the context of limited resource availability.

Most of the heuristics with error bound analysis have been gathered in this study [11], noting that these heuristics, in some cases, produce optimal solutions. As well, the known polynomial (P) and pseudo-polynomial (pseudo-p) models were summarized in a single table whose results show that they are applicable to simpler problems with equivalent performance measures. The authors concluded that "if availability constraints come from unexpected breakdowns, fully online algorithms will be needed; but in case of preemptive scheduling, many results of optimality concern the best nearly online algorithms. It is an open question to look for the optimality results from fully on-line algorithms and specific availability patterns, or at least to compute performance bounds." Finally, they stated that a direction is to assume that one operation cannot be interrupted at any point of time, but only at given instants. Furthermore, they suggested that "authors work on more complicated problems such as sequencing n jobs on m resources in the flow job or open job environment [11]."

Chaari *et al.* [13] noted that in the real-world several types of hard-to-predict risks must be considered in scheduling problems, and that scheduling under uncertainty allows taking these kinds of risks into account.

They considered that the numerical values (e.g., execution time, machine speed) which are used in the scheduling methods as uncertain, incomplete or imprecise.

Consequently, they focused on the dynamic scheduling more than on the deterministic scheduling problems because it is the real scheduling case.

Different types of scheduling approaches under uncertainty environment exist in the literature as well as several dedicated typologies [13]. Most of these typologies are based on the distinction between different scheduling approaches among the following: stochastic, fuzzy [151] proactive and reactive or hybrid which includes two subtypes: predictive-reactive and pro-reactive approaches.

They proposed a global classification schemes technically independent and encompass new kind of scheduling algorithms under uncertainty [13]:

1. Proactive (also known as robust) scheduling approaches: this kind of approach tries to anticipate uncertainty while developing flexibility, in order to produce a schedule, or a family of schedules, that is relatively insensitive to uncertainty [14]. In proactive approaches, five different techniques can be identified: techniques based on robustness measures, redundancy-based techniques, probabilistic methods, contingent scheduling and optimization-based techniques.
2. Reactive scheduling approaches: these approaches are often used in highly perturbed real time environment when off-line scheduling becomes rapidly unfeasible. In this context, decision-making must be very fast and intuitively easy for users to understand [15]. Different methods are used to solve reactive scheduling problems involve distributed (multi-agent systems) or centralized approaches, priority rules and dynamic choice of priority rules. These approaches may exploit priority local criteria when a decision must be made in real-time.
3. Hybrid approaches: these approaches can be subdivided into predictive-reactive and proactive-reactive approaches.

A. Predictive - reactive approaches: These approaches, used to support risks, have two scheduling phases:

i) First phase: a deterministic schedule is set up off-line.

ii) Second phase : this schedule is used and adapted on-line. The on-line phase requires making scheduling decisions one at a time while the schedule is being built. These decisions are then adapted in real time to take disturbances into account [16].

Scheduling methods can be constructed by answering two questions: “When to reschedule?” and “How to reschedule?”

B. Proactive - reactive approaches: In the proactive-reactive approaches, in contrast to predictive-reactive approaches, no rescheduling is done on-line; instead, one among several pre-estimated schedule solutions is chosen. This is what can make it possible to build a set of static schedules such that it is easy, in the event of risk, to pass from one to the other.

- “A new mixed technique presented by [17] combines a proactive approach with a reactive approach to deal with scheduling problem under uncertainty. In the proactive phase, the authors built a robust baseline schedule that minimizes the schedules distance defined as the sum of the absolute deviations between the baseline and expected schedules. The robust baseline schedule contains some built-in flexibility in order to minimize the need of complex search procedures for the reactive scheduling approach.”

The authors presented an updated and enriched classification scheme for the various approaches for scheduling in an uncertain environment. The more important are as follows [13]. The first is about the notion of predicting uncertain events and measuring their impact on scheduling. The second concerns combining multi-agent approaches with optimization techniques for dynamic scheduling and dynamic control. This prospective has a lack of optimality is due to decisions that lead to the myopic behavior of decision-making entities [18]. A possible solution would be to increase the intelligence of these entities by introducing optimization techniques in the decisional process. The third set of prospective research concerns the integration of new emerging technologies into existing products such as RFID (radio-frequency identification), mechatronics and embedded infotronics. This integration enables the products to participate in the decision-making in a dynamic scheduling context. The fourth set of prospective concerns the new possibilities for designing and dimensioning systems based upon scheduling performances. This takes in account the agility perspective and the issues of evolution and improvement in the production system's (re)design & (re)engineering process.

Kaabi *et al.* [7] focused on results related to the specific real industrial problem of parallel machine scheduling under availability constraints. The authors present parallel machines problems under different constraints with the appropriate solving algorithms along with their complexities under various objective functions. By emphasizing on certain concepts, they pointed out the following:

- Parallel machines scheduling is at hand when machines of similar type and eventually slightly different in characteristics are available in multiple numbers. Jobs can be processed over these machines simultaneously.
- The machines may be subject to accidental breakdowns, periodic preventive maintenance (mainly non-availability), tool changes, workers availability, and availability of the resources used by the machines, and so on.
- There are two classes of problems considered depending on whether the scheduling of preventive maintenance activities is determined before the scheduling of jobs or jointly with the scheduling of jobs. Class 1 : in the case that the two activities of the production scheduling and the maintenance "planning" are generally planned and executed separately in real manufacturing systems. Class 2 : when maintenance and production services collaborate in order to maximize the system productivity. Maintenance strategies can be broadly classified into corrective maintenance (CM) and preventive maintenance (PM).

- The job scheduling under maintenance constraints was generally applied to a single machine and multi-machine models. In this paper, the authors deal only with scheduling parallel machines subject to availability constraints.

The authors use multiple tables to represent: Most often used notations in scheduling in Table III-1 The performance measures, main results of one machine scheduling under availability constraints in Table III-2 and main results on scheduling identical parallel machines under availability constraints in Table III-3 [7]. Subsequently we present two tables summarising the results extracted from this survey concerning the scheduling problems running over uniform and unrelated parallel machines under availability constraints.

Tableau III-2 SCHEDULING PROBLEMS OVER UNIFORM PARALLEL MACHINES

Measure to be optimized	Solving algorithm	Improvement of complexity or competitive ratio	Ref.
$Q_{1,2}\|online\|C_{max}$ (to denote the problem of online scheduling on two uniform parallel machines where one machine is periodically unavailable to minimize the makespan)	Optimal algorithm (as declared by authors)	An online scheduling is investigated on m parallel machines and on two uniform parallel machines where there is one machine periodically unavailable. In the latter case the length of each available period is normalized to 1 while the speed of the other one is s>0. If s ≥ 1: speed of the 2nd machine, the proposed algorithm is optimal with a competitive ratio 1+1/s. In the case where 0 < s < 1, the authors proposed some lower bounds on competitive ratio.	[129]
		If s=1 and jobs arrive in decreasing sequence and proved that proposed is optimal with competitive ratio 3/2.	[130]

Tableau III-3 SCHEDULING PROBLEMS OVER UNRELATED PARALLEL MACHINES

Measure to be optimized	Solving algorithm	Improvement of complexity or competitive ratio	Ref.
Total machine load on m unrelated parallel machines with maintenance activity (ma) $Rm \left\| P_{jn}, ma \leq h_{jk} \right\| \sum_{i=1}^{m} Ci$	Two efficient algorithms	Complexity O(n^{m+3})	[133]
Total machine load $Rm \left\| P_{jn} = P_{ij} + ab_{ji}, ma \leq h_{jk} \right\| \sum_{i=1}^{m} Ci$			
Minimizing the total completion time or the total machine load on m unrelated parallel machines	An algorithm that considers a deterioration of maintenance activities	Complexity O(n^{m+3})	[131], [132]
Minimizing the total completion time on m unrelated parallel machines	An algorithm that reconsiders simultaneously a deterioration effects and deteriorating multi-maintenance activities.	Find jointly the optimal maintenance frequencies, the optimal maintenance positions, and the optimal job sequences with a polynomial time algorithm	[134]

The authors concluded that "the maintenance activities can be planned in a flexible or in a non-deterministic ways. In fact, the machines are subject to random breakdowns." In addition, the assumption of non resumable jobs needs to be taken into account for many real life problems. "Considering more realistic constraints such as online scheduling, resumable jobs, and nondeterministic availability constitute interesting research directions [7]."

The survey paper [19] of Allahverdi provides an extensive review of about 500 papers that have appeared since the mid 2006 to the end of 2014, including static, dynamic, deterministic, and stochastic environments. These survey papers classify scheduling problems based on shop environments as single machine, parallel machine, flow shop, job shop, or open shop. It further classifies the problems as family and non-family as well as sequence-dependent and sequence-independent setup times/costs. In this paper the focus is on the setup times/costs [20] factor ignored by most of the existing scheduling literature. Allahverdi drew up several tables that present the articles references with their criteria or measures to be optimized and their used approaches. These tables are built based on the shop environments as single machine, parallel machine, etc., and according to the problems classification as family and non-family as well as sequence-dependent and sequence-independent setup times/costs. The author summarized the results as follows:

- Heuristics solutions methods have been more used than (the double of) the exact methods.
- The genetic algorithm has been the first one used among the heuristics followed by the simulated annealing (SA).
- Among the exact solutions methods, the mixed integer programming (MIP) and the branch and bound (B&B) are the most used methods.

He concluded the need for:

- More research on scheduling problems with explicit consideration of setup times/costs.
- Considering family setup time for the parallel and job shop environments.

- Addressing the sequence-dependent scheduling problems in single machine environments with family setup times.
- Addressing more scheduling problems with multiple criteria.
- Addressing more scheduling problems with uncertain setup times.

In [111] the dynamic job shop scheduling that considers random job arrivals and machine breakdowns was studied. “Considering an event driven policy rescheduling, is triggered in response to dynamic events by variable neighborhood search (VNS). A trained artificial neural network (ANN) updates parameters of VNS at any rescheduling point. As well, a multi-objective performance measure is applied as objective function that consists of makespan and tardiness. The proposed method is compared with some common dispatching rules that have been widely used in the literature for dynamic job shop scheduling problem.”

This paper [119] is characterized by its study of the static scheduling problem of m identical parallel machines with a common server and sequence dependent setup times. In fact, according to the best knowledge of the authors, it is the first such study in the literature. The authors focused, in their study, on the comparison of the performance of the proposed a mixed integer linear programming (MILP) model, simulated annealing (SA) and genetic algorithm (GA) based solution approaches with the performance of basic dispatching rules such as: shortest processing time first (SPT) and longest processing time first (LPT) over a set of randomly generated problem instances.

The MILP model is presented with the SPT and LPT dispatching rules for the problem to minimize the makespan. However, according to the authors, the MILP model is not able to solve the large scaled problem instances due to the NP-hard nature of the problem. For this reason, simulated annealing (SA) and genetic algorithm (GA) based solution approaches are proposed for solving the large scaled problem instances. As a result, based on the computational experiments, the proposed GA is generally the most efficient and effective (followed by the AS approach) in solving this problem. As the problem size increases, the GA approach finds better solutions with smaller standard deviations. Genetic algorithms (GA) are the basis of stochastic optimization algorithms, as they can also be used for machine learning.

Kia *et al.* studied in [6] a dynamic flexible flow line problem with sequence-dependent setup times for minimizing the mean flow time and mean tardiness. By applying genetic programming framework and choosing proper operators, four new composite dispatching rules are proposed to solve this NP-hard problem. To examine scheduling rules performances, a discrete-event simulation model is made considering four new heuristic rules and the six adapted heuristic rules from the literature.

In [117], Rintanen considered that the contingent approach recognizes that different schedules are needed under different contingencies and computes them either off-line before the execution phase or on-line as information about the contingencies becomes available. This is the most general approach, eliminating the limitations (incompleteness, sub-optimality) of the other approaches at the cost of increased complexity." He investigated the properties of some classes of contingent scheduling problems. In these problems assignments of resources to tasks depend on resource availability and other facts that are only known fully during execution. Therefore, the off-line construction of one fixed schedule is insufficient. He demonstrated generally that contingent scheduling is most likely outside the complexity class NP "Their results prove that standard constraint-satisfaction and SAT (Site acceptance testing) frameworks are in general not straightforwardly applicable to contingent scheduling."

Skutella *et al.* address in [120] two main characteristics encountered generally in real-world scheduling problems: the heterogeneous processors and a certain degree of uncertainty about the sizes of jobs. They studied, for the first time according to their best knowledge, a scheduling problem that combines the classical unrelated machine scheduling model with stochastic processing times of jobs.

For the stochastic version of the unrelated parallel scheduling problem and with the objective of weighted sum of completion times $R\,|(r_{ij}\,)|\sum\,w_jC_j$, they calculated in polynomial time a scheduling policy with a performance guarantee of $(3\,+\,\Delta)\,/\,2\,+\,\varepsilon$ using a novel time-indexed linear programming relaxation. They showed that when jobs also have individual release dates, their bound is $(2\,+\,\Delta)\,+\,\varepsilon$ where, Δ is an upper bound on the squared coefficient of variation of the processing times and $\varepsilon\,>\,0$ is arbitrarily small. They showed that the dependence of the performance guarantees on Δ is tight.

On the deterministic side, the current best known approximation algorithms for unrelated parallel machines have respectively performance guarantees 3/2 and 2 for the problem without and with release dates [121, 122, 124, 125]. Improving these bounds is considered one the most important open problems in scheduling (see Schuurman and Woeginger [126]).

On the Stochastic front, the authors stated that they consider for the first time the stochastic variant of unrelated parallel machine scheduling. In stochastic scheduling, it is asked to compute a non-anticipatory scheduling policy which must make its decisions at an indicated time based on the observed past up to this time as well as the a priori knowledge of the input data of the problem. Here, the processing time of a job j on machine i is given by random variable P_{ij}. The authors assume that the random variables P_{ij} are stochastically independent across jobs. For any given non-anticipatory scheduling policy, the possible outcome of the objective function $\sum\,w_jC_j$ is a random variable. Then, the goal is to minimize its expected value, which by linearity of expectation equals $\sum\,w_j\mathbb{E}[C_j]$.

The authors mentioned that for the first time they completely departed from the linear programming relaxation of Mohring *et al.* [123], and showed how to put a novel, time-indexed linear programming relaxation to work in stochastic machine scheduling. According to the authors, this approach will inspire further research for other stochastic optimization problems in scheduling and related areas. In addition, they showed how to overcome the difficulty that scheduling policies feature a considerably richer structure including complex dependencies between the executions of different jobs which cannot be easily described by time-indexed variables. As a result, they presented the first time-indexed LP relaxation for stochastic scheduling on unrelated parallel machines. Here, they calculated the probability value of a job j being started on machine i at time t which can be represented by the time-indexed variable x_{ijt}. The situation is complicated in the stochastic context, and it requires a fair amount of information on the exact probability distributions of the random variables. Some other surveys on classical and industrial scheduling problems are summarized in Table III-4.

Tableau III-4 PROPERTIES FROM ADDITIONAL SURVEYS OF SCHEDULING

Authors / Reference	Scheduling problems context	Scheduling problems environment	Conclusion and suggestions
Samia Ourari / [3]	Deterministic to distributed Scheduling approaches based on cooperation	Scheduling approaches under uncertainties: reactive, proactive and proactive-reactive approaches. They differ according to how uncertainty is taken into account, either offline or on-line, single machine & job shop	It is important to address the problem of managing uncertainty in scheduling in order to reduce the gap between the theory (estimated or expected scheduling) and the practical field (scheduling really implemented or adopted).

Janiak *et al.* / [105]	Offline & online scheduling	Single machine & parallel machine scheduling, Just-in-Time (JIT) scheduling models, PERT/CPM (Program Evaluation Research Technique)/(Critical Path Method) scheduling	A currently noticeable trend in this area is the concept of combining pure due window scheduling problems with other new and trendy phenomena, like e.g., learning or aging effects, deteriorating jobs, maintenance activities, etc. These models need practical trend more than theoretical trend and among their future trends are: the analysis of scheduling problems with due windows (multiple due windows) and preemptive jobs or precedence constraints.

III.3. SCHEDULING PROBLEMS IN CLOUD COMPUTING

Masdari *et al.* [32] addressed scheduling problems in the Cloud environment that, like other environments, still lacks adequate and effective solutions for load balancing and scheduling tasks and workflows. The tasks or jobs are mapped to the appropriate virtual machines (VMs) which are generated virtually from the single physical machine to optimize some given scheduling measures. Various heuristic, metaheuristic and exact algorithms are applied to study the Cloud scheduling problem. In this paper, the authors present an in-depth analysis of the particle swarm optimization (PSO)-based task and workflow scheduling schemes, proposed in the literature, for the cloud environment. Moreover, they provide a classification of the proposed scheduling schemes based on the type of the PSO algorithms illuminating their objectives, properties and limitations. In the particle swarm optimization algorithm, the swarm of particles is randomly generated initially, and each particle position in the search space represents a possible solution and has a fitness value and velocity to determine the speed and direction of its moves. By moving and updating position and velocity, particles get an optimized solution [35, 37]. The authors claim that, following the repeated advances (called iteration), the particle swarm gradually approaches the optimal location [41].

The authors present the papers in literature that propose the PSO algorithms with different schemes:

- In Standard PSO schemes such as Guo *et al.* [39], Zhang *et al.* [42], Yang *et al.* [38], Huang *et al.* [35], Pandey *et al.* [57], Wu *et al.* [40] and Jianfang *et al.* [36].
- In Multi-Objective[3] PSO schemes, several contributions were cited in the literature: Netjinda *et al.* [43], Wang *et al.* [139], Ramezani *et al.* [60] and Yassa *et al.* [44].
- In Bi-Objective[4] PSO schemes; the papers of Beegom *et al.* [48] and Verma *et al.* [50].
- In Hybrid[5] PSO schemes; the papers of Zhan *et al.* [46], Kaur *et al.* [54], Visalakshi *et al.* [52], Krishnasamy and Gomathi *et al.* [49], Xue *et al.* [53], JieHui *et al.* [55] and Xiaoguang *et al.* [51].
- In Learning PSO schemes which use learning PSO for scheduling in cloud environment; the papers of Zuo *et al.* [47] and Chen *et al.* [45].

[3] This is the process of simultaneously optimizing two or more conflicting objectives subject to a number of constraints.

[4] This is the process which solves optimization models for two objective functions respectively.

[5] To overcome some of the limitations of the PSO, one or more algorithms such as Genetic Algorithm, Ant Colony, etc., are integrated with the PSO algorithm.

- In Jumping PSO Schemes (which is proposed to optimize the load balancing, the speed-up ratio, and the makespan) the paper of Chitra *et al.* [59].
- In Modified PSO Schemes (it is an improved PSO to overcome the drawbacks of the standard PSO or to increase its performance) the paper of Tarek *et al.* [56], Zhao *et al.* [61], Pragaladan *et al.* [58] and Abdi *et al.* [62].

Properties of these PSO-based scheduling schemes are illustrated in five tables in this survey. Based on these tables, Tableau III-5 shows in the form of a report a summary of how the schemes studied respond to these properties.

Tableau III-5 PROPERTIES REPORT OF SCHEDULING SCHEMES STUDIED IN [32]

PSO Scheduling scheme	References of scheduling schemes studied	The rate of scheduling schemes studied in this survey and which deal with these properties.					
		Objectives				Scheduling type	
		Minimizing cost	Minimizing task execut on time	QoS support	Minimizing makespan	Task	Workflow
Standard	[39, 42, 38, 35, 57, 40, 35]	6/7[6]	-	2/7	4/7	3/7	4/7
Multi Objective	[43, 60, 44, 139]	4/4	2/4	2/4	1/4	3/4	1/4
Bi-Objective	[48, 50]	2/2	1/2	-	1/2	1/2	1/2
Hybrid	[46, 54, 52, 49, 53, 55, 51]	1/7	5/7	1/7	1/7	6/7	1/7

[6] e.g., 6 scheduling schemes out of a total of 7 dealt with “Minimizing cost”.

Learning	[47, 45]	-	-	2/2	-	1/2	1/2
Jumping	[59]	-	-	-	1/1	-	1/1
Modified	[56, 61, 58, 62]	3/4	2/4	-	1/4	3/4	1/4

The authors stated that metaheuristic algorithms present better results than deterministic algorithms (with a particular given input, they produce always the same output) in terms of the quality. Likewise, they find approximate solutions faster than traditional exhaustive algorithms in terms of the computation time [138].

Finally, the authors concluded that scheduling is a critical process to map the cloud tasks to the VMs and to reduce their cost and time of execution. Moreover, the future research on scheduling problem should consider, investigate, evaluate and enhance various security related factors in the task and workflow scheduling solutions. As well, the heterogeneous resources functionality, the load balancing on VM (virtual machines) and data center network to reduce their energy consumption. As well, scheduling tasks and workflows on hybrid and federated clouds should be studied more.

Ramathilagam and Vijayalakshmi [63] noted that there is no typical effective task scheduling algorithm employed in the cloud environment. Furthermore, the well-known task planners have great difficulty in being implemented in a large-scale distributed environment due to the high communication charge. This requires the building of compatible and applicable job scheduling algorithms and load balancing techniques in this large scale environment. “To balance the load in cloud the resources and workloads must be scheduled in an efficient manner. A variety of scheduling algorithms are used by load balancers to determine which backend server to send a request to [152].” Cloud computing approaches employed the latest technology which is increasing considerably. Job scheduling is one of the processes done with the aim to efficiently enhance the functioning of cloud computing atmosphere with achieving maximum profit. In this paper, the authors have investigated and discussed concisely several scheduling algorithms and issues in cloud computing. These algorithms fall into two groups: static and dynamic. Both have their own merits and demerits.

In addition, the authors [63] surveyed various types of task scheduling algorithms in cloud computing. These algorithms are included and compared in Tableau III-1 in Section 5 of their survey. They declared that the heuristic based algorithm, that's belonging to a subset of meta-heuristic approach, is one of the important means to achieve the optimal or near optimal solution of task scheduling in the cloud environment. Many task scheduling techniques that employed in the cloud environment are classified into the following three categories:

1. Traditional techniques which are simple and deterministic, but they get stuck in local optima [67]: First come first serve (FCFS), round robin (RR) and shortest job first (SJF) etc.
2. Heuristic techniques which are used to find the optimal or near optimal solution by using a sample space of random solutions are: min-min, max-min, enhanced max-min [66] and priority based min-min etc. These techniques give better results as compared to the traditional approaches [69].
3. Meta-heuristic techniques make use of random solution space for tasks scheduling. The principal difference between heuristic and meta-heuristic is the first one is problem specific while the second one is problem independent [70]. meta-heuristics generally have functional similarities with the aspects of the science of life (biology): (a) meta-heuristics based on gene transfer: Genetic algorithms and transgenic algorithm; (b) meta-heuristics based on interactions among individual insects: Ant colony optimization, firefly algorithm, marriage in honey bees optimization algorithm, artificial bee colony algorithm; and (c) meta-heuristics based on biological aspects of alive beings: Tabu search algorithm, simulated annealing algorithm, particle swarm optimization algorithm and artificial immune system [153].

Based on this extensive survey, the authors concluded that there remain many problems and issues to enhance as the need of scheduling techniques that covers all requirements accurately. Another issue which is a very vital in scheduling algorithms is the energy efficiency (energy consumption, energy savings, energy sufficiency, etc.). For multiple workflows, metrics like reliability and availability should also be considered.

Sharma *et al.* describe in [64] the work done in the field of task scheduling algorithms. These algorithms are classified as follows:

1. Efficient task scheduling algorithm : Sindhu *et al.* [65] proposed an enhanced task scheduling algorithm to minimize the completion time of cloudlets. Their approach has two algorithms named as, longest cloudlet fastest processing element (LCFP) and shortest cloudlet fastest processing element (SCFP).
2. Improved min-min algorithm : Kaur *et al.* [68] proposed an improved min-min algorithm to achieve the maximum resource utilization in distributed environment. This algorithm consists of two phases: the first one is similar to the traditional min-min algorithm, in which minimum completion time of each task is calculated. In the second phase, tasks are rescheduled to make selection of those resources which have been unutilized for a long period of time.
3. Enhanced max-min algorithm: In order to optimize the task scheduling in cloud computing environment, Santosh *et al.* [73] proposed an enhanced max-min algorithm that consists of two algorithms using respectively the arithmetic and geometric means for calculating average time of job execution instead of maximum completion time. Then the job which has execution completion time just greater than the calculated average time is selected. If jobs are independent of each other, then arithmetic mean gives the best time average. However, on the contrary the geometric mean calculates the best average of time.
4. Selective Algorithm: A selective algorithm has been proposed by Kobra Etminani *et al.* [74] for ensuring the QOS. This algorithm uses the advantages of the two basic scheduling algorithms min-min and max-min and tries to overcome their disadvantages. The selective parameter is the standard deviation of the completion time of unassigned tasks in Meta task.
5. Optimized task scheduling algorithm: for improving scalability in the cloud environment, Shubham Mittal *et al.* [72] proposed an optimized task scheduling algorithm. The authors

took into account five algorithms (min-min, max-min, RASA, improved max-min and enhanced max-min).

6. Improved task scheduling algorithm: Abdul Razaque *et al.* [71] proposed an improved task scheduling algorithm to achieve the proper utilization of the network bandwidth in the cloud computing environment. They use a non-linear programming model for assigning a proper number of tasks to each virtual machine.
7. Cloud-based workflow scheduling algorithm: Bhaskar Prasad Rimal *et al.* [75] proposed a cloud-based workflow scheduling algorithm in order to enhance the workflow in a multi-tenant cloud environment. The authors have defined the workflow as a new service layer and the fourth one on the top of the infrastructure. They used a directed acyclic graph (DAG) to represent the workflow in the systems. The labels on the nodes and the edges represent respectively the costs of computation and of communication. This algorithm uses the ideal time of resources, reduces the makespan, properly utilizes resources and minimizes the cost.
8. Task scheduling algorithm based on quality of service (QoS): A task scheduling – quality of service (TS-QOS) algorithm has been proposed by Xiaonian Wu *et al.* [76] for optimizing the service quality in cloud computing. Firstly, the algorithm computes the priority of each task on the basis of certain parameters and then it sorts the entire list of tasks according to their priority. The task having minimum completion time is considered as highest priority task and gets the resource first for job completion. Three indexes are taken in account for measuring the performance of this algorithm: i) the makespan, ii) the average waiting time of the longest task (average latency), iii) the load balancing index (LBI) used to determine the loading conditions of the system and maximum system loading capacity [154].
9. Task scheduling algorithms with multiple factors: Nidhi Bansal *et al.* [77] proposed a comparison between traditional scheduling methods, i.e., FCFS, optimization method, QoS-driven, ABC (activity based costing) and priority based algorithms, etc., by using CloudSim as a simulator. The authors showed that the resource utilization and cost factor are the main

criteria in any scheduling algorithm to deal at best. As well, they stated that optimization based methods performed better as compared to the traditional methods.

As a result of their study, the authors concluded that there is no such heuristic approach which can fulfill all the required parameters. However, they can perform better when some particular parameter among resource utilization, execution time for each task, and workflow, and so on are considered at a time.

Abbas and Zhang in [83] study the mobile edge computing (MEC), which is an emergent architecture that extends the cloud computing services to the edge of networks leveraging mobile base stations. It can be applied to mobile, wireless and wireline scenarios, using end-users' software and hardware platforms located at the network edge.

The mobile networks suffer from low storage and energy capacity, low bandwidth, and high latency [84]. Moreover, exponential growth of the emerging Internet-of-Things (IoT) technology is foreseen to further stumble cellular and wireless networks [85]. The edge computing fog computing) [86], has begun to be of paramount significance, especially mobile edge computing (MEC) in mobile cellular networks. MEC is equipped with better offloading techniques that characterize the network with low-latency and high-bandwidth.

The basically contribution of this paper is surveying MEC. A few MEC survey reports such as [87] and [88] exist in the literature. It provides a brief overview of different attributes of MEC and identifies the major open research challenges in MEC. In addition, it presents an extensive survey on mobile edge computing focusing on its general overview.

Subsequently, several research efforts were recently carried out in the area of MEC. They are classified according to different domains:

1. Offload computation: This is a way to improve the capacity of mobile devices by transferring the computation to higher resourceful servers that are located at a different location [92]. The improvement mobile devices and networks will still not be able to cope with the

increased demand on these devices. As a result, mobile devices will always have to compromise with their limited resources, such as resource-poor hardware, insecure connections, and energy computing tasks [89].

- In 2015, many algorithms or prototypes are proposed such as i) edge accelerated web browsing (EAB) prototype proposed by Takahashi *et al.* [91]. It is designed for web applications using a better offloading technique. ii) An algorithm-based design, called successive convex approximation (SCA) proposed by Ardellitti *et al.* [90]. This algorithm optimizes computational offloading on multiple densely deployed radio access points. iii) FemtoCloud system proposed by Habak *et al.* [93] which forms a cloud of orchestrated co-located mobile devices that are self-configurable into a correlative mobile cloud system
- In 2016, other algorithms or prototypes are proposed: a) the efficient computation offloading model designed by Chen *et al.* [94] using a game theoretic approach in a distributed manner. Game theory is a persuasive tool that helps simultaneously connected users to make the correct decision when connecting a wireless channel based on the strategic interactions. b) the contract-based computation resource allocation scheme proposed by Zhang *et al.* [95]. It improves the utility of vehicular terminals which intelligently utilize services offered by MEC service providers under low computational conditions.

2. Low Latency: MEC is equipped with better offloading techniques that characterize the network with low-latency and high-bandwidth. So, it is one of the promising edge technologies that can improve user experience by providing high bandwidth and low latency.

- In 2015, Nunna *et al.* [97] proposed a real-time context aware collaboration system by combining MEC with 5G networks. This integration of MEC and 5G helps to empower real time collaboration systems utilizing context-aware application platforms. These systems require context information combined with geographical information and low latency communication.

- In 2016, two schemes are proposed: i) REPLISOM designed by Abdelwahab *et al.* [96] is an edge cloud architecture and (long term evolution) LTE enhanced memory replication protocol to avoid latency issues. LTE bottleneck occurs when allocating memory to a large number of IoT devices in the backend cloud servers. ii) Kumar *et al.* [99] proposed a vehicular delay tolerant network-based smart grid data management scheme. The authors investigated the use of vehicular delay-tolerant networks (VDTNs) to transmit data to multiple smart grid devices exploring the MEC environment.

3. Storage : To overcome their device storage limitation, end-users may utilize MEC resources.
   - In 2016, Jararweh *et al.* [98] proposed a framework connects software defined system components to MEC to further extend MCC (mobile cloud computing) capabilities, software defined system for mobile edge computing (SDMEC). The components jointly work cohesively to enhance MCC into the MEC services.
4. Energy efficiency: The MEC architecture is created to reduce energy consumption of user devices by migrating compute intensive tasks to the network edge.
   - Many schemes are developed in this field:
   - In 2014, an opportunistic peer-to peer mobile cloud computing framework was proposed by Wei Gao [103]. The probabilistic framework is composed of peer mobile devices connected via their short-range radios. Based on their available capacity, these mobile devices are able to share energy and computational resources. The author proposed the probabilistic method to estimate the opportunistic transmission status of the network ensuring that the resulting computation is timely delivered to its initiator.
   - In 2015: i) an architecture that integrates MEC to voice over LTE called ME-VoLTE was proposed by Beck *et al.* [101]. The encoding of video calls is offloaded to the MEC server located at the base station (eNodeB). The offloading of video encoding through external services helps escalating battery lifetime of the user equipment. Encoding is high computational-intensive and hence is very power consuming. ii) El-Barbary *et al.* [100] proposed DroidCloudlet; an architecture to enhance mobile battery lifetime by migrating

data-intensive and compute-intensive tasks to rich-media. Based on commodity mobile devices DroidCloudlet is legitimized with resource-rich mobile devices that take the load of resource-constraint mobile devices. iii) in 2016 Jalali *et al.* [102] proposed a flow-based and time based energy consumption model. They conducted number of experiments for efficient energy consumption using centralized nano Data Centers (nDCs) in a cloud computing environment. But, the authors claim that nDCs energy consumption is not yet been investigated.

The authors [83] concluded that as a recent technology platform, little research has been specifically done in MEC. In fact, there are some open issues in MEC that need to be addressed. Many researchers interested by MEC have studied some problems of these issues that belong in several areas including: Resource optimization, transparent application migration, Web Interface, security, pricing, network openness, multi-services and operations, robustness and resilience.

We summarise, in Tableau III-6, significant properties extracted from some other surveys of scheduling problems in cloud environment: context, environment, conclusion and suggestions.

Tableau III-6 ADDITIONAL SURVEYS OF SCHEDULING IN CLOUD ENVIRONMENT

Authors / Reference	Scheduling problems context	Scheduling problems environment	Conclusion and suggestions
S. Yi *et al.* / [106]	Static, dynamic, real time and	The main issues: Fog networking (is heterogeneous): Internet of Things, software-defined	Fog computing will evolve with the rapid development in underlying IoT, edge devices, radio access techniques, SDN, NFV, VM and

	Heuristic scheduling	networking, network function virtualization (NFV) to create flexible and easy maintaining network environment. quality of service (QoS): connectivity, reliability, capacity, and delay. Interfacing and programming model. Computation offloading. accounting, billing and monitoring. Provisioning and resource management: Application-aware provisioning, resource discovery and sharing. Security and privacy	Mobile cloud. We think fog computing is promising but currently need joint efforts from underlying techniques to converge to "fog computing".
Deshmane *et al.* / [107].	Static, dynamic, real time and heuristic scheduling	Enhanced max-min, Improved genetic, scalable heterogeneous earliest-finish-time (SHEFT) Improved cost-based, resource-aware-scheduling, innovative transaction intensive cost-constraint scheduling, algorithms and multiple QoS constrained scheduling	There is a need to implement a new scheduling algorithm to minimize the execution time and improve availability and reliability in a cloud computing environment. The improvement can also be done with building algorithms that take user preferences while scheduling. Also, one more aspect can help improving the design of

		strategy of multi-workflows (MQMW)	algorithm, which can include new factors such as inter-node bandwidth etc., that have not been considered for resources matching.
Ahmed *et al.* / [88]	Static, dynamic, real time and scheduling	The contributions of this article are as follows: (a) survey of the state of-the-art research. (b) Preview taxonomy based on various parameters such as characteristics, actors, access technologies, applications, objectives, computational platforms, and key enablers. (c) Identification of various open challenges related to the Mobile Edge Computing that impede or prevent the successful deployment.	The open research challenges in mobile edge computing: Standard protocol, simulation platform, mobility management, heterogeneity, pricing model, scalability and security
Singh *et al.* / [109]	Dynamic / static scheduling and allocation of resources	Resource scheduling algorithms (RSA) and dynamic RSAs, bargaining based RSA, compromised cost and time based RSA, cost based RSA, dynamic and adaptive based RSA, energy based RSA, hybrid based RSA, nature	Recent research has shown that resource scheduling algorithms using resource provisioning mechanisms and applying the effective resource provisioning technique. On the basis of existing research, it is necessary to fully understand

		inspired and bio-inspired based RSA, optimization based RSA, profit based RSA, priority based RSA, SLA and QoS based RSA, time based RSA and VM based RSA Resource scheduling aspects and resource distribution policies	QoS requirements for workload for better allocation of resources rather than to detect workload and resources. It is necessary to find the progress in the search on the cloud itself before finding the advanced search in the scheduling of resources.
Manpreet Kaur / [110]	Static & dynamic scheduling	Min-Min, Max-Min, RASA, shortest job first, heuristic, dominant resource priority and multi-objective task scheduling algorithms	Future work would be to continue the multi-objective scheduling improvement. The authors have done the non- dominated sorting of virtual machines (VMs) according to MIPS. In future they aim to take other parameters of VMs also to sort them for better performance.

III.4. SCHEDULING PROBLEMS IN MOBILE NETWORKS

Many surveys have been published recently with regard to scheduling problems in mobile networks. In this section, we address these surveys briefly.

Mahidhar *et al.* [21] addressed the problem of the wireless sensor network (WSN) by providing dynamic scheme. WSN is a highly distributed network of small and light nodes. This problem is demonstrated by the limited battery life of the nodes. Sensor nodes spend their energy in transmitting and receiving the data, as well as, in the relaying of the packets. This implies designing the routing algorithm that maximizes the lifetime of the network. Packet scheduling is important in WSN to maintain fairness based on data priority and to reduce end-to-end delay. The authors proposed the dynamic multilevel priority (DMP) packet scheduling scheme with the bit rate classification; the data is divided into three categories as high, moderate and low bit rate. They also proposed the threshold value check mechanism to prevent deadlock situations. To provide security they implement the RC6 security algorithm.

Another important implication in real-time WSN data transmission is the packet scheduling at node sensors that ensures the delivery of different packages according to their priority and fairness without any delay. This saves battery power by reducing sensors working time.

The authors also presented various existing real time scheduling schemes which are as follows [21]:

1. Dynamic conflict free transmission scheduling (DCQS): is a query based novel scheduling technique, designed to support in network data aggregation and in response to the workload changes it can dynamically adapt to the transmission [23, 26].
2. Nearest job next (NJN): It consists of the mobile element (ME); server and client. The client is the one which request the service and it is a simple and intuitive discipline which is adopted by the ME to select the next request to be served or the next client [22].
3. Traffic pattern oblivious scheduling (TPO): to handle efficiently a wide variety of the traffic pattern by using a single TDMA (Time-Division Multiple Access) schedule [27].
4. Dynamic multilevel priority packet scheduling (DMPPS): it consists of three levels of the priority queues, the data is placed in the priority queue based on the priority, the last level of the virtual hierarchy does not have the priority queue and the levels are formed based on the hop distance from the base station [25].
5. First come first serve (FCFS): it is the simplest packet scheduling algorithm in which packets are processed as they come [24].

By comparing these schemes, the authors concluded that DMPPS is the better one. They presented a literature review table for five articles [25,29,31,28,30]. This table shows the objective, key issues, and advantages which are summarized by: reducing average energy consumption, balancing the nodes energy consumption, minimizing the delay at nodes and increasing network life. They also illustrated the adaptive staggered sleep protocol ASLEEP protocol which is efficient for the power management in wireless sensor network. This protocol adjusts dynamically the node sleep schedulers to match the network demand. The node adjusts its active period dynamically [21, 137].

The scheduling scheme proposed in [21] has three levels of the priority queues; the last level of the virtual hierarchy does not have the priority queue. The data packet classification is done as i) real time data given as priority 1, ii) non real time remote data, received from the lower level nodes, given as priority 2, iii) non real time local data, sensed from the node itself, given as priority 3. The TDMA scheme is used to process the data packet sensed by the node which are at the different levels. The conclusion drawn from this paper is that one of the advantages of their DMP scheme is its dynamicity to the changing requirement of the wireless sensor network application. The proposed threshold value, to check mechanism at the time of the priority level when the data arrives at the high priority queue, helps to reduce the deadlock situation.

In his survey [78], Nimbalkar introduces opportunistic scheduling for effective load balancing in multipath traffic network. This is a technique that aims to maximize throughput and packet delivery ratio by exploiting short-term variation in path condition.

Opportunistic Scheduling works to achieve two objectives simultaneously: selecting the user with the best channel conditions and satisfying the fairness constraints over long-term scales. Thus, many algorithms are developed, as in [79, 80, 81, 82] which exploit high-quality channels to realise the fairness use of the multiple channels available.

Opportunistic scheduler takes into account some criteria such as maximum delay, minimum throughput, maximum response time, maximum latency in the way to impose fairness on the channels. So, certain characteristics should be contained in a good scheduling algorithm such as maximum resource utilization, maximum throughput, minimum turnaround time, minimum waiting time, and minimum response time.

The opportunistic scheduling techniques has been studied and classified in five categories. The following table (Tableau III-7) summarizes their characteristics.

Tableau III-7 OPPORTUNISTIC SCHEDULING OF MULTICONSTRAINT NETWORK [78]

Category	Approach	Optimized measures	Channels / scheduling categories	Performance	Resolution settings	Ref.
Fairness	Proportional-fair sharing approach	Total throughput	Two competing channels	Maximize total throughput	The channel rate history	[140]
	Multichannel Fair Scheduler	Optimal throughput	Multiple wireless channels / deterministic and probabilistic	Maximize throughput	This model uses adaptive control framework to develop opportunistic fair wireless schedulers.	[141]
	Indexed to optimal	QoS and throughput	Multiple user QoS	Optimal solution of throughput		[79]

	solution of throughput					
	Heuristic opportunistic scheduling policy	QoS, throughput and short-term fairness	Multiple interface system	Throughput performance of the heuristic policy is comparable to that of the long- term optimal policy	Heuristic opportunistic scheduling policy	
	This model uses Lyapunov optimization framework for stochastic network optimization	Stronger delay & efficient throughput utility	Stochastic	Opportunistic scheduling that guarantees a bounded worst-case delay	Network which has time-varying channel condition	[80]
Delay	Using radio resource allocation in OFDMA	Delay sensitive traffic, system capacity and throughput fairness		This model provides fairness with respect to the realizable throughput per user, packet dropping ratios	OFDMA (Orthogonal frequency-division multiple access) based network	[81]

				and packet delay distributions		
	An opportunistic scheduling based on multi-user diversity effect	Total throughput		Total system throughput is maximized	This scheduling mechanism can result in higher spectrum utilization	[82]
Quality of Service	A time slotted system where time is the resource to be shared among all users	Maximize the system performance stochastically	Multiple channels / stochastic	Users scheduling, to transmit slots at each time, that optimize the network performance.		[108]
	Model uses a time-slotted system where time is the resource to be shared	Throughput value	Multiple channels / stochastic	The performance of a user's channel condition by enlarging the stochastic process value.		[142]

	A model whereby the scheduling mechanism is based on preventing from transmitting under adverse conditions.	Maximize user utility measure, e.g., communication rate for efficient utilization of the available communication resources.	Shared wireless channels	Communication rate	Resources of a wireless channel network	[143]
	Reinforcement learning framework to design distributed adaptive opportunistic routing problem (d-AdaptOR)	Minimizing the average per packet cost for routing a packet from source to destination	Wireless multi-hop network	Packet routing from source to destination	Distributed adaptive opportunistic routing problem (d-AdaptOR)	[144]
	Low complexity adaptive	An identical throughput guarantee	Time-slotted networks	An approximate throughput guarantee	This model develops an expression for the	[145]

Throughput	scheduling algorithms				approximate throughput guarantee violation probability for users in time-slotted networks	
	A simple algorithm for networks with short-lived flows	Throughput optimal	Wireless networks with short-lived flows	Performance of the channel flows transmission	Wireless channel network	[146]
	Opportunistic Multipath Scheduling (OMS)		Multipath routing uses multiple alternative paths in the network.	OMS minimize the delay and improves overall throughput	Multiple network paths	[147]
	A new model of opportunistic scheduling mechanism	Maximizing the system overall throughput	Wireless network with hybrid links	To avoid starvation of the link having a much lower transmission rate	Wireless channel network	[148]

.	A model of distributed opportunistic scheduling (DOS) under average delay constraint	Maximize the overall throughput or the throughput of every link according to its own individual delay constraint	Ad-hoc network of wireless channels	Optimize the throughput performance in ad-hoc network	Homogeneous/heterogeneous scenarios with saturated/non-saturated stations	[149]
	Distributed opportunistic scheduling (DOS)	Maximize the throughput in a network of channels with a certain access probability	Wireless network	Improve the throughput performance of the wireless network	Homogeneous/heterogeneous scenarios with saturated/non-saturated stations	[150]

In [112], the authors write "we present a centralized integrated approach for: 1) enhancing the performance of an IEEE 802.11 infrastructure wireless local area network (WLAN), and 2) managing the access link that connects the WLAN to the Internet. Our approach, which is implemented on a standard Linux platform, and which we call ADvanced Wi-fi internet service EnhanceR (ADWISER), is an extension of our previous system WLAN

Manager (WM). ADWISER addresses several infrastructure WLAN performance anomalies such as mixed-rate inefficiency, unfair medium sharing between simultaneous TCP uploads and downloads, and inefficient utilization of the internet access bandwidth when Internet transfers compete with LAN–WLAN transfers, etc. The approach is via centralized queueing and scheduling, using a novel, configurable, cascaded packet queueing and scheduling architecture, with an adaptive service rate."

In the objective of managing inter-cell interference with a centralized controller, Ramos-Cantor *et al.* addressed in [116] the problem of coordinating scheduling decisions among multiple base stations in an -LTE-Advanced downlink network. In order to solve the coordinated scheduling problem an integer non-linear program is formulated. It only makes use of the specific measurement reports defined in the 3GPP standard. Unlike most existing approaches, it does not rely on exact channel state information. The authors proposed an equivalent integer linear programming reformulation of the coordinated scheduling problem, which can be solved efficiently by commercial solvers. The performance of the proposed coordinated scheduling approaches is analyzed by extensive simulations of medium to large size networks. The available analytical results show fundamental limits in cooperation due to interference outside the cluster.

A centralized scheduling approach was proposed in [114] to manipulate centralized coordination among heterogeneous agents. In this study, the center agent acts as an information collector, processer and resource scheduler. The main contribution of this center is enacting centralized scheduling to run well. A clustering analysis based on artificial immune algorithm is applied to process information, moreover a series of schemes are suggested to ensure smooth scheduling.

In [113], it is indicated that the data-scale computing for analytical workloads is becoming increasingly popular. Due to the high operational costs heterogeneous applications are forced to share cluster resources to achieve economies of scale. Existing approaches of scheduling large and diverse workloads are tackled in two alternative ways: (1) the solutions offer strict, secure enforcement of scheduling invariants (fairness, capacity) for heterogeneous applications; and (2) the distributed solutions offer scalable, efficient scheduling for homogeneous applications. The authors proposed Mercury, a hybrid resource management framework that supports the full spectrum of scheduling, from centralized to distributed. Mercury exposes a programmatic interface that allows applications to trade-off between scheduling overhead and execution guarantees. The authors stated that their framework harnesses this flexibility by opportunistically utilizing resources to improve task throughput. In addition, experimental results on production-derived workloads show gains of over 35% in task throughput. These benefits can be translated by appropriate application and framework policies into job throughput or job latency improvements. They have implemented and contributed mercury as an extension of apache hadoop / YARN.

Fu *et al.* introduced in [115] En-Omega. This is a novel hierarchical hybrid design of schedulers to address the serious job starvation problem that triggers especially in heavily loaded clusters. In En-Omega the fully distributed schedulers can be enhanced with a central scheduler. This can provide a global fairness to the jobs from different schedulers and simultaneously reduce the average latency of all the jobs sharply. To reduce the overhead, in En-Omega design, the central scheduler will be activated only when the cluster is heavily loaded. Furthermore, the cache used for central queuing and the scoring policy used in central scheduling are all load-aware. En-Omega was evaluated based on Google trace and experimental results show that, compared to the baseline design, this method can reduce the average latency of starving jobs up to 90% with reasonable overhead.

In [118], the authors pointed out that the traditional distributed wireless video scheduling is based on perfect control channels where instantaneous control information from the neighbors is available. They mentioned the difficulty to obtain this information in practice, especially for dynamic wireless networks. They found that the two approaches - distortion-minimum scheduling aiming to meet the long term video quality demands and the other that focusing on a minimum delay - can't be applied directly. Then they went to investigate the distributed wireless video scheduling with delayed control information (DCI). First, they translate this scheduling problem into a stochastic optimization rather than a convex optimization problem in the way to exploit in a tractable framework. Next, they consider two classes of DCI distributions to study the relationship between the DCI and scheduling performance and provide a general performance property bound for any distributed scheduling. These classes are: i) the class with finite mean and variance, and ii) a general class that does not employ any parametric representation. Thereafter, it will be created a class of distributed scheduling scheme to achieve the performance bound by making use of the correlation among the time-scale control information. The main contributions are presented at the theoretical and technical level.

1. Theoretical level: an appropriate Lyapunov function based on observed DCI is presented to establish the scheduling performance in terms of DCI. This leads to the positive Harris recurrence property of the network Markov process. This represents the most challenging part of this work since it needs to prove an effective time scale separation between the network state dynamics and scheduling decision dynamics. To make this possible, they design an increase function of queue-size to capture the correlation of the DCI.
2. Technical level: a distributed video scheduling scheme in terms of DCI is proposed. This scheme only utilizes local queue-length information to make scheduling decisions. As well, it only requires each node to perform a few logical operations at each scheduling decision.

The authors concluded that they provided a general performance property bound for any distributed scheduling. Importantly, they designed a class of distributed online scheduling scheme to achieve the optimal performance bound by making use of the correlation among the time-scale control information.

Some other surveys of scheduling problems in mobile networks, whose important characteristics are summarized in Tableau III-8, are Context, Environment, Conclusion and Suggestions.

Tableau III-8 ADDITIONAL SURVEYS OF MOBILE NETWORKS SCHEDULING

Authors / Reference	Area	Context	Scheduling problems environment	Conclusion and suggestions
Akashdeep *et al.* / [104]	Mobile networks	Point-to-multipoint (PMP) and mesh mode for wireless broadband access in networks.	Scheduling techniques for IEEE 802.16 networks in PMP mode and their approaches which may be divided into sub categories such as traditional, hierarchal, cross layer approaches, dynamic schedulers and soft computing based.	There are some of the areas still not quite explored namely the application of soft computing/optimization techniques like genetic algorithm, neural networks, fuzzy logic etc. Using these approaches together with information from higher layers can act as a major contributor in the field of scheduling.

III.5. DISCUSSION

Through recent surveys presented in the present paper and others more ancient, we conclude that the scheduling problem has been developed by balancing tasks on one machine, on a few machines and then over very large number of machines. This is developed with time, over decades, from heavy manufacturing to various areas of light industries passing recently in networks of mobile devices going to cloud, fog and edge computing.

There are many different algorithms designed by researchers or adopted from several domains, especially from mathematics, and then developed to solve scheduling problems. In this paper, we are investigating their advantages in terms of efficiency and speed of achieving the desired optimal scheduling policy.

Based on what was published, we find that the criterion to meet is to accomplish the desired job efficiently in the shortest possible time, taking into account the capacity of equipment used in terms of energy and treatment. It is obvious that researchers prefer an exact optimal policy as a resolution approach to solving scheduling problems. However, if this policy has a great complexity and requires more effort, energy and time, researchers aim to find another policy close to it simpler and more economy. This is the case of small mobile devices in networks that support the latest technologies, for example cloud and fog computing. especially, for the communication systems of ubiquitous resources that have several challenges still cannot be accommodated such as: high energy consumption, the continuously increasing demand for high data rates and the mobility required by new wireless applications. So, according to the fact that these devices suffered of poor energy and storage or computing capacity, the trend is the use of heuristic and meta-heuristic methods.

We mention the success achieved through the use of cloud computing in solving the problems of communication as well as the storage and transfer of data. Then, the use of fog and edge computing to solve the problems experienced by cloud computing. But, we find that these means become ineffective in the case of heavy and excessive communications or the case of natural or terrorist disasters that may destroy their service centers. Therefore, we believe that the solution in these cases is the network of ubiquitous resources and mobile devices that can be established anytime and anywhere. This can also be used in normal situations because of their very low cost and their permanent presence with us.

III.6. CONCLUSION

Scheduling and load balancing problems are the most important issues for sharing tasks between machines or devices. The most recent systems, such as Internet objects, cloud computing, fog and edge, whose main issue is sharing tasks, are rapidly evolving because of technological advances and excessive demand for their use.

In this paper, we presented a collection of recent surveys and articles that are interested in scheduling problems. We presented different scheduling issues, and investigated their advantages and disadvantages based on many criteria, such as context, environment, optimizing function, used algorithms, suggestions and proposed improvement. This may give rise to a document which summarizes most recent researches in this field in order to present a comprehensive idea of scheduling problems.

The main objective of this effort is to facilitate the work of researchers and readers who investigate this model as well as those working in this field. As well, to attract attention to enhance and thus take advantage of newly emerging areas that are in dire need of scheduling such as cloud, fog and edge computing, especially the network of ubiquitous resources and mobile devices.

As a result, we have concluded that efforts in this area focus on optimizing the desired measures and achieving jobs in the shortest possible time. However, this seems to be unattainable by using exact and optimal policies of scheduling and load balancing tasks on the available devices. This is due to the algorithms complexity that gives these exact solutions. Therefore, another way is adopted, in the literature, by means of heuristic and approximate algorithms which search optimal policies close to the exact ones, which decreases considerably the execution time. We recommend and urge to focus the future works to develop this type of algorithms.

Finally, we aim to attract researchers to enhance scheduling issues in the individual mobile networks segment of collaborative mobile computing. The main factor behind this trend is to create a practical alternative to the applications of information, communications, task scheduling and many more, for example in cases of disasters and terrorism when it becomes impossible to move from place to another and use the centers dedicated to these applications. Another reason, to be realized, is the facilities and benefits that these issues can provide to users in terms of implementing and executing their applications, and thus managing their various jobs, through mobile networks at any time and wherever they are. This is for nowadays, but for years to come, we claim that they will be the de facto devices due to the anticipated future technological improvements.

REFERENCES

[1] J. Leung, J. Anderson, « *Handbook of Scheduling: Algorithms, Models, and Performance Analysis*, » CRC/Chapman-Hall, 2004.

[2] T. Vredeveld, "Stochastic online scheduling", *Computer Science-Research and Development*, vol. 27, no 3, p. 181-187, 2012.

[3] S. Ourari, « De l'ordonnancement déterministe à l'ordonnancement distribué sous incertitudes, » Thèse de doctorat, Université Paul Sabatier - Toulouse III, Toulouse, France, 2011.

[4] K. R. Baker, « *Introduction to sequencing and scheduling*, » Wiley, New York, 1974.

[5] M. Pinedo, "*Scheduling: Theory, algorithms and systems*," 2nd edition, Prentice Hall, 2008.

[6] H. Kia, S. H. Ghodsypour, H. Davoudpour, "New scheduling rules for a dynamic flexible flow line problem with sequence-dependent setup times," *Journal of Industrial Engineering International*, vol. 13, 2017, doi: 10.1007/s40092-017-0185-y.

[7] J. Kaabi, Y. Harrath, "A survey of parallel machine scheduling under availability constraints", *International Journal of Computer and Information Technology*, vol. 3, no. 2, pp. 238-245, 2014.

[8] Y. Ma, C. Chu, C. Zuo, "A survey of scheduling with deterministic machine availability constraints", *Computers & Industrial Engineering*, vol 58, pp. 199-211, 2010.

[9] Z. Tan, Y. He, "Optimal online algorithm for scheduling on two identical machines with machine availability constraints," *Information Processing Letters*, vol 83, pp. 323-329. 2002.

[10] P. Senthilkumar, S. Narayanan, « Literature Review of Single Machine Scheduling Problem with Uniform Parallel Machines, » *Intelligent Information Management*, vol 2, pp. 457-474, 2010.

[11] H. R. D. Saidy, M. T. Taghavi-Fard, "Study of Scheduling Problems with machine Availability Constraint," *Journal of Industrial and Systems Engineering*, vol. 1, no. 4, pp. 360-383, 2008.

[12] J. Blazewicz, J. Breit, P. Formanowicz, W. Kubiak, G. Schmidt, "Heuristic algorithms for the two-machine flowshop problem with limited machine availability," *Omega Journal*, vol 29; pp. 599-608, 2001.

[13] T. Chaari, S. Chaabane, N. Aissani, D. Trentcsaux, "Scheduling under uncertainty: Survey and research directions", *In Proceedings of Advanced Logistics and Transport (ICALT)*. *IEEE*, pp. 229-234, 2014.

[14] X. Lin, S. L. Janak, C. A. Floudas, « A new robust optimization approach for scheduling under uncertainty: I. Bounded uncertainty," *Computers & Chemical Engineering*, *FOCAPO 2003 Special issue*, vol. 28, no. 6, pp. 1069-1085, 2004.

[15] E. Szelke, R. Kerr, "Knowledge-based reactive scheduling," *Production Planning & Control*, vol. 5, no. 2, pp. 124-145, 1994.

[16] P. Cowling, D. Ouelhadj, S. Petrovic, "Dynamic scheduling of steel casting and milling using multi-agents", *Production Planning & Control*, vol. 15, no. 2, pp. 178-188, 2004.

[17] C.W. Wu, K.N. Brown, J.C. Beck, "Scheduling with uncertain durations: Modeling Beta-robust scheduling with constraints," *Computers and Operations Research*, vol. 36, pp. 2348-2356, 2009.

[18] D. Trentesaux, "Distributed control of production systems," Engineering Applications of *Artificial Intelligence*, vol. 22, no. 7, pp. 971-978, 2009.

[19] A. Allahverdi, "The third comprehensive survey on scheduling problems with setup times/costs", *Euroepan Journal of Operational Research*, vol. 246, pp. 345-378, 2015.

[20] A. Allahverdi, H. M. Soroush, "The significance of reducing setup times/setup costs", *European Journal of Operational Research*, vol. 187, pp. 978-984, 2008.

[21] R. Mahidhar and A. Raut, "A Survey on Scheduling Schemes with Security in Wireless Sensor Networks", *Procedia Computer Science Journal*, vol. 78, pp. 756-762, 2016.

[22] L. He, Z. Yang. « Evaluating Service Disciplines for On-Demand Mobile Data Collection in Sensor Networks, » *IEEE Transactions on mobile computing*, vol. 13, no. 4; Apr. 2014.

[23] R. Gomathi, N.Mahendran, "An Efficient Data Packet Scheduling Scheme in Wireless Sensor Networks", *IEEE 2nd international conference on electronics and communication system (ICECS)*, 2015.

[24] G. D. Celik, E. H. Modiano, "Controlled mobility in stochastic and dynamic wireless networks," *Queueing Systems: Theory and Applications,* vol. 72, no 3, pp. 251-277, 2012. [Online]. Available: https://doi.org/10.1007/s11134-012-9313-y

[25] N. Nasser, L. Karim, T. Taleb, "Dynamic Multilevel Priority Packet Scheduling Scheme for Wireless Sensor Network", *IEEE Transactions on wireless communications*, vol.12, no. 4; Apr. 2013.

[26] J. A. Stankovic, C.-G. Roman, « Dynamic Conflict-Free Transmission Scheduling for Sensor Network Queries », *IEEE Transactions on mobile computing*, vol. 10, no. 5, May 2011.

[27] A. Somasundara, A. A. Ramamoorthy, "Mobile Element Scheduling for Efficient data collection", in *wireless sensor network with Dynamic Deadline*, Department of Electrical Engineering, *IEEE International Real-Time Systems Symposium*, pp. 1052-8725, 2004.

[28] G.Bergmann, M.Molnar, L.Gonczy, B. Cousin, "Optimal Period Length for the CQS Sensor Network Scheduling Algorithm," in *Prac. 2010 International Conf. Netw. Services*, pp. 192-199, 2010.

[29] B. Nazir and H. Hasbullah, “Dynamic Sleep Scheduling for Minimizing Delay in Wireless Sensor Network,” *2011 Saudi International Electronics, Communications and Photonics Conference (SIECPC)*, pp. 1-5, 2011, doi: 10.1109/SIECPC.2011.5876935.

[30] A.R.Swain, R.C.Hansdah, and V.K.Chouhan, “An Energy Aware Routing Protocol with Sleep Scheduling for Wireless Sensor Networks,” in *Proc. 2010 IEEE International Conf. Adv. Inf. Netw. Appl*, pp. 933-940, 2010.

[31] F.Liu, C. Tsui, and Y. J. Zhang, “Joint Routing and Sleep Scheduling for Lifetime Maximization of Wireless Sensor Networks”, *IEEE Trans. Wireless Communication*, vol. 9, no. 7, pp. 2258-2267, 2010.

[32] M. Masdari, F. Salehi, M. Jalali, and M. Bidaki, “A survey of PSO based scheduling algorithms in cloud computing”, *J. Netw. Syst. Manage.*, vol. 25, no. 1, pp. 122-158, 2016.

[33] J. Kennedy, R. C. Eberhart, “Particle Swam Optimization”, *Proceedings of ICNN'95 - International Conference on Neural Networks*, vol.4, pp. 1942-1948, 1995, doi: 10.1109/ICNN.1995.488968.

[34] P. Shelokar, P. Siarry, V. K. Jayaraman, B. D. Kulkarni, “Particle swarm and ant colony algorithms hybridized for improved continuous optimization”, Applied Mathematics and Computation, vol. 188, no. 1, pp. 129-142, 1 May 2007.

[35] J. Huang, K. Wu, L. K. Leong, S. Ma, M. Moh, “A tunable workflow scheduling algorithm based on particle swarm optimization for cloud computing”, Criterion, pp. 12-14, 2013.

[36] C. Jianfang, C. Junjie, Z. Qingshan, “An optimized scheduling algorithm on a cloud workflow using a discrete particle swarm,” *Cybernetics and information technologies*, vol. 14, no. 1, Sofia, 2014.

[37] R.C. Eberhart, Y. Shi, “Particle swarm optimization: developments, applications and resources,” In: *Evolutionary Computation, Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546)*, vol. 1, pp. 81-86 2001, doi: 10.1109/CEC.2001.934374.

[38] X. Qin, Z. Yang, W. Li, Y. Yang, "Optimized task scheduling and resource allocation," in *cloud computing using PSO based fitness function, Inf. Technol. J.*, vol. 12, pp. 7090-7095, 2013.

[39] L. Guo, S. Zhao, S. Shen, C. Jiang, "Task scheduling optimization in cloud computing based on heuristic algorithm", *journal of networks*, vol. 7, no. 3, pp. 547-553, Mar. 2012.

[40] Z. Wu, Z. Ni, L. Gu, X. Liu, « A revised discrete particle swarm optimization for cloud workflow scheduling, » In *2010 International Conference on Computational Intelligence and Security*, pp. 184-188, 2010, doi: 10.1109/CIS.2010.46.

[41] X. Wen, M. Huang, J. Shi, "Study on Resources Scheduling Based on ACO Algorithm and PSO Algorithm in Cloud computing," In *2012 11th International Symposium on Distributed Computing and Applications to Business, Engineering & Science*, pp. 219-222, 2012, doi: 10.1109/DCABES.2012.63.

[42] G. Zhang, X. Zuo, "Deadline constrained task scheduling based on standard-PSO in a hybrid cloud", In *International Conference in Swarm Intelligence,* Springer, Berlin, Heidelberg, pp. 200-209, 2013.

[43] N. Netjinda, T. Achalakul, B. Sirinaovakul, "Cloud provisioning for workflow application with deadline using discrete PSO", *ECTI Transactions on Computer and Information Technology (ECTI-CIT)*, vol. 7, no 1, pp. 43-51, 2013.

[44] S. Yassa, R. Chelouah, H. Kadima, B. Granado, "Multi-objective approach for energy-aware workflow scheduling in cloud computing environments," *The Scientific World Journal*, vol. 2013, 2013. doi:10.1155/2013/ 350934.

[45] W.-N. Chen, J. Zhang, "A set-based discrete PSO for cloud workflow scheduling with user-defined QoS constraints," In: *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, pp. 773-778, 2012.

[46] S. Zhan, H. Huo, "Improved PSO-based task scheduling algorithm in cloud computing," *Journal of Information & Computational Science*, vol. 9, no 13, pp. 3821-3829, 2012.

[47] X. Zuo, G. Zhang, W. Tan, “Self-adaptive learning PSO-based deadline constrained task scheduling for hybrid IaaS cloud,” *IEEE transactions on automation science and engineering*, vol. 11, no 2, pp. 564-573, 2013.

[48] A.A. Beegom, M. Rajasree, “A particle swarm optimization based pareto optimal task scheduling in cloud computing,” in *advances in swarm intelligence*, pp. 79-86. Springer, 2014.

[49] K. Krishnasamy, B. Gomathi, “Task scheduling algorithm based on hybrid particle swarm optimization in cloud computing environment,” *journal of theoretical and applied information technology*, vol. 55 no. 1, pp. 33-38, 10th Sept. 2013.

[50] A. Verma, S. Kaushal, “Bi-criteria priority based particle swarm optimization workflow scheduling algorithm for cloud,” In: *2014 recent advances engineering and computational sciences (RAECS),* pp. 1-6, 2014. doi: 10.1109/RAECS.2014.6799614.

[51] Y. Xiaoguang, C. Tingbin, Z. Qisong, “Research on cloud computing schedule based on improved hybrid PSO,” in computer science and network technology *(ICCSNT)*, pp. 388-391, 2013.

[52] P. Visalakshi, S. Sivanandam, “Dynamic task scheduling with load balancing using hybrid particle swarm optimization,” Int. *J. Open Problems Compt. Math.* vol. 2, pp. 475-488, 2009.

[53] S.-J. Xue, W. Wu, “Scheduling workflow in cloud computing based on hybrid particle swarm algorithm,” *TELKOMNIKA Indones. J. Electr. Eng.* vol. 10, pp. 1560-1566, 2012.

[54] E.S. Sharma, G. Kaur, “Optimized utilization of resources using improved particle swarm optimization based task scheduling algorithms in cloud computing,” *Int. J. Emerg. Technol. Adv. Eng.* vol. 4, pp. 110-115, 2014.

[55] J.U. JieHui, B.A.O. WeiZheng, W.A.N.G. ZhongYou, W.A.N.G. Ya, L.I. WenJuan, “Research for the task scheduling algorithm optimization based on hybrid PSO and ACO for cloud computing,” int. *j. grid distrib. comput.* vol. 7, pp. 87-96, 2014.

[56] M.Z. Zahraa Tarek, F.A. Omara, “Pso optimization algorithm for task scheduling on the cloud computing environment,” int. *j. comput. technol.* vol. 13, 2014.

[57] S. Pandey, L. Wu, S.M. Guru, R. Buyya, "A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments," 2010 24th *IEEE International conference on advanced Information networking and applications*, pp. 400-407, 2010. doi: 10.1109/AINA.2010.31.

[58] R. Pragaladan, R. Maheswari, "Improve workflow scheduling technique for novel particle swarm optimization in cloud environment," *Int. j. eng. res. gen. sci.* vol. 2, no. 5, pp. 675-680, 2014.

[59] S. Chitra, B. Madhusudhanan, G. Sakthidharan, P. Saravanan, "Local minima jump PSO for workflow scheduling in cloud computing environments," in *advanced in computer science and its applications*, pp. 1225-1234. Springer 2014.

[60] F. Ramezani J., Lu, F. Hussain, "Task scheduling optimization in cloud computing applying multiobjective particle swarm optimization," in *Service-oriented computing*, pp. 237-251. Springer 2013.

[61] G. Zhao, "Cost-aware scheduling algorithm based on PSO in cloud computing environment," *International journal of grid & distributed computing*, vol. 7, pp. 33-42, 2014.

[62] S.A.M. Solmaz Abdi, S. Sharifian, "Task scheduling using modified PSO algorithm in cloud computing environment," In International conference on machine learning*, Electrical and mechanical engineering (ICMLEME'2014)*, vol. 4, no. 1, pp. 8-12, 2014.

[63] A. Ramathilagam, K. Vijayalakshmi, "A survey of scheduling algorithm in cloud computing environment," *International science press, IJCTA*, vol. 9, no. 36, pp. 137-145, 2016.

[64] N. Sharma, S. Tyagi, S. Atri, "A survey on heuristic approach for task scheduling in cloud computing," *International journal of advance research in computer science*, vol. 8, pp. 1089-1092, Mar.-Apr. 2017.

[65] S. Sindhu, S. Mukherjee, "Efficient task scheduling algorithms for cloud computing environment," *Springer-verlag Berlin Heidelberg*, pp. 79-83, 2011.

[66] U. Bhoi, P. N. Ramanuj, "Enhanced max-min task scheduling algorithm in cloud computing," *International journal of application or innovation in engineering and management (IJAIEM)*, vol. 2, no. 4, pp. 259-264, Apr. 2013.

[67] Shimpy, J. Sidhu, “Different scheduling algorithms in different cloud environment,” *International journal of advanced research in computer and communication engineering (IJARCCE)*, vol. 3, no. 9, pp. 8003-8006, Sept. 2014.

[68] R. kaur, P. K. Patra, « Resource allocation with improved min-min algorithm, » *International journal of computer applications (IJCA)*, vol. 76, pp. 61-67, Aug. 2013.

[69] R. J. S. Raj, S. V. M. Prasad, “Survey on variants of heuristic algorithms for scheduling workflow of Tasks,” *International conference on circuit, power and computing technologies [ICCPCT]*, IEEE, pp. 1-4, 2016, doi: 10.1109/ICCPCT.2016.7530288.

[70] G. Pate, R. Mehta, “A survey on various task scheduling algorithm in cloud computing,” *International journal of advanced research in computer engineering & technology (IJARCET)*, vol. 3, no. 3, pp. 715-717, Mar. 2014.

[71] A. Razaque, N. R. Vennapusa, N. Soni, G. S. Janapati, k. R. Vangala, “Task scheduling in cloud computing,” in *2016 IEEE Long island systems, applications and technology conference (LISAT),* pp. 1-5, 2016, doi: 10.1109/LISAT.2016.7494149.

[72] S. Mittal, A. Katal, "An Optimized task scheduling algorithm in cloud,” *IEEE 6th International advanced computing conference*, pp. 197-202, 2016.

[73] B. Santhosh, D. H. Manjaiah, “An improved task scheduling algorithm based on max-min for cloud computing,” *International journal of innovative research in computer and communication engineering (IJIRCCE)*, vol. 2, no. 2, pp. 84-88, May 2014.

[74] K. Etminani, M. Naghibzadeh, « A min-min max-min selective algorihtm for grid task scheduling, » *2007 3rd IEEE/IFIP International conference in central Asia on internet*, pp. 1-7, 2007, doi: 10.1109/CANET.2007.4401694.

[75] B. P. Rimal, M. Maier, “Workflow scheduling in multi-tenant cloud computing environments,” *IEEE Transaction on parallel and distributed systems*, pp. 1-14, 2015.

[76] X. Wu, M. Deng, R. Zhang, B. Zeng, S. Zhou, « Task scheduling algorithm based on QoS-driven in cloud computing, » *Elsevier publications*, pp. 1162-1169, 2013.

[77] N. Bansal, A. Awasthi, S. Bansal, “Task scheduling algorithms with multiple factors in cloud computing environment,” *Springer India*, pp. 619-627, 2016.

[78] P. N. Nimbalkar, “A survey on opportunistic scheduling for traffic networks with multiple constraints,” *International journal of innovative research in computer and communication engineering*, vol. 5, no. 5, May 2017. [Online]. Available: www.ijircce.com.

[79] Kulkarni, S. Suresh, C. Rosenberg, “Opportunistic scheduling: generalizations to include multiple constraints, multiple interfaces, and short term fairness,” *Wireless networks*, vol. 11, no. 5, pp 557-569, Sept. 2005. [Online]. Available: https://doi.org/10.1007/s11276-005-3512-y

[80] Neely, J. Michael, “Opportunistic scheduling with worst case delay guarantees in single and multi-hop networks,” *2011 proceedings IEEE INFOCOM*, pp. 1728-1736, 2011, doi: 10.1109/INFCOM.2011.5934971.

[81] Khattab, KF Ahmed, K. MF Elsayed, “Opportunistic scheduling of delay sensitive traffic in OFDMA-based wireless network,” *proceedings of the 2006 International Symposium on World of Wireless, Mobile and Multimedia Networks*. IEEE Computer Society, pp.10-19, 2006, doi: 10.1109/WOWMOM.2006.81.

[82] Liu, Xin, E. KP Chong, N. B. Shroff, « Optimal opportunistic scheduling in wireless networks ». *2003 IEEE 58th Vehicular technology conference. VTC 2003-Fall (IEEE Cat. No.03CH37484),* vol.3, pp. 1417-1421, 2003, doi: 10.1109/VETECF.2003.1285258.

[83] N. Abbas, Y. Zhang, A. Taherkordi, T. Skeie, “Mobile edge computing: A survey,” in *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 450-465, Feb. 2018, doi: 10.1109/JIOT.2017.2750180.

[84] G. Orsini, D. Bade, and W. Lamersdorf, “Computing at the mobile edge: Designing elastic android applications for computation offloading,” in *2015 8th IFIP Wireless and mobile networking conference (WMNC)*, pp. 112-119, Oct 2015.

[85] E. Borgia, R. Bruno, M. Conti, D. Mascitti, A. Passarella, “Mobile edge clouds for information-centric iot services,” *in 2016 IEEE Symposium on computers and communication (ISCC)*, pp. 422-428, Jun. 2016.

[86] K. Kai, W. Cong, L. Tao, “Fog computing for vehicular ad-hoc networks: paradigms, scenarios, and issues,” *The Journal of China universities of posts and telecommunications*, vol. 23, no. 2, pp. 56- 96, 2016.

[87] R. Roman, J. Lopez, M. Mambo, « Mobile edge computing, fog *et al.*: A survey and analysis of security threats and challenges, » *Future generation computer systems*, vol. 78, part 2, pp. 680-698, Jan. 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X16305635

[88] A. Ahmed E. Ahmed, “A survey on mobile edge computing,” in *2016 10th International conference on intelligent systems and control (ISCO)*, pp. 1-8, 2016.

[89] D. Kovachev, T. Yu, R. Klamma, “Adaptive computation offloading from mobile devices into the cloud”, in *2012 IEEE 10th International symposium on parallel and distributed processing with applications*, pp. 784-791, Jul. 2012.

[90] S. Sardellitti, G. Scutari, S. Barbarossa, “Joint optimization of radio and computational resources for multicell mobile-edge computing,” *IEEE Transactions on signal and information processing over networks*, vol. 1, no. 2, pp. 89-103, Jun. 2015.

[91] N. Takahashi, H. Tanaka, R. Kawamura, “Analysis of process assignment in multi-tier mobile cloud computing and application to edge accelerated web browsing,” in *2015 3rd IEEE International conference on mobile cloud computing, services, and engineering,* pp. 233-234, 2015, doi: 10.1109/MobileCloud.2015.23.

[92] K. Kumar, J. Liu, Y.-H. Lu, B. Bhargava, “A survey of computation offloading for mobile systems,” *Mobile networks and applications*, vol. 18, no. 1, pp. 129-140, 2013. [Online]. Available: http://dx.doi.org/10.1007/s11036-012-0368-0

[93] K. Habak, M. Ammar, K. A. Harras, E. Zegura, “Femto clouds: Leveraging mobile devices to provide cloud service at the edge,” in *2015 IEEE 8th International conference on cloud computing*, pp. 9-16, Jun. 2015.

[94] X. Chen, L. Jiao, W. Li, X. Fu, “Efficient multi-user computation offloading for mobile-edge cloud computing,” *IEEE/ACM Transactions on networking*, vol. 24, no. 5, pp. 2795-2808, Oct. 2016.

[95] K. Zhang, Y. Mao, S. Leng, A. Vinel, Y. Zhang, "Delay constrained offloading for mobile edge computing in cloud-enabled vehicular networks," in *2016 8th International workshop on resilient networks design and modeling (RNDM)*, pp. 288-294, Sept 2016.

[96] S. Abdelwahab, B. Hamdaoui, M. Guizani, T. Znati, "Replisom: Disciplined tiny memory replication for massive iot devices in lte edge cloud," *IEEE Internet of Things Journal*, vol. 3, no. 3, pp. 327-338, Jun. 2016.

[97] S. Nunna, A. Kousaridas, M. Ibrahim, M. Dillinger, C. Thuemmler, H. Feussner, A. Schneider, "Enabling real-time context-aware collaboration through 5g and mobile edge computing," in *Information technology - new generations (ITNG)*, pp. 601-605, Apr. 2015.

[98] Y. Jararweh, A. Doulat, A. Darabseh, M. Alsmirat, M. Al-Ayyoub, E. Benkhelifa, "Sdmec: Software defined system for mobile edge computing," in *2016 IEEE International conference on cloud engineering workshop (IC2EW)*, pp. 88-93, Apr. 2016.

[99] N. Kumar, S. Zeadally, J. J. P. C. Rodrigues, « Vehicular delay tolerant networks for smart grid data management using mobile edge computing, » *IEEE Communications magazine*, vol. 54, no. 10, pp. 60-66, Oct. 2016.

[100] A.E.H.G. El-Barbary, L.A.A. El-Sayed, H.H. Aly, M.N. El-Derini, "A cloudlet architecture using mobile devices," in *2015 IEEE/ACS 12th International conference of computer systems and applications (AICCSA)*, pp. 1-8, Nov 2015.

[101] M. T. Beck, S. Feld, A. Fichtner, C. Linnhoff-Popien, T. Schimper, "Me-volte : Network functions for energy-efficient video transcoding at the mobile edge," in *Intelligence in next generation networks (ICIN)*, pp. 38-44, Feb 2015.

[102] [102] F. Jalali, K. Hinton, R. Ayre, T. Alpcan, R. S. Tucker, "Fog computing may help to save energy in cloud computing," IEEE Journal on selected areas in communications, vol. 34, no. 5, pp. 1728-1739, May 2016.

[103] [103] W. Gao, "Opportunistic peer-to-peer mobile cloud computing at the tactical edge," in *2014 IEEE Military communications conference*, pp. 1614-1620, Oct 2014.

[104] [104] Akashdeep, Karanjeet S. Kahlon, Harish Kumar, "Survey of scheduling algorithm in IEEE 802.16PMP Network," *Egyptian informatics journal*, vol.15, pp. 25-36, 2014.

[105] A. Janiaka,W. A. Janiakb,T. Krysiaka,T. Kwiatkowski, “A survey on scheduling problems with due windows,” *European journal of operational research*, vol. 242, no. 2, pp. 347-357 16 Apr. 2015.

[106] S. Yi, Z. Qin, and Q. Li, “Security and privacy issues of fog computing: A survey,” in *International conference on wireless algorithms, systems and applications (WASA*), Springer, Cham. pp. 685-695, 2015.

[107] KA Deshmane and BT Pandhare, “Survey on scheduling algorithms in cloud computing,” *International journal of emerging trend in engineering and basic sciences*, vol. 2, no. 1, pp. 210-2015, 2015.

[108] L. Xiaojun, E. K. P. Chong, N. B. Shroff, “Opportunistic transmission scheduling with resource-sharing constraints in wireless networks,” *IEEE Journal on selected areas in communications*, vol. 19, no. 10, pp. 2053-2064, Oct. 2001, doi: 10.1109/49.957318.

[109] S. Singh, I. Chana, “A survey on resource scheduling in cloud computing: issues and challenges,” *J. grid computing*, vol. 14, no. 2, pp. 217-264, 2016. [Online]. Available: http://dx.doi.org/10.1007/s10723-015-9359-2 .

[110] M. Kaur, “Survey of task scheduling algorithms in cloud computing,” *International journal of innovative research in computer and communication engineering*, vol. 5, no. 4, Apr. 2017. [Online]. Available: www.ijircce.com .

[111] M. A. Adibi, M. Zandieh, M. Amiri, “Multi-objective scheduling of dynamic job shop using variable neighborhood search,” *Expert systems with applications*, vol. 37, no. 1, pp. 282-287, 2010.

[112] M. Hegde, P. Kumar, K. R. Vasudev, N. N. Sowmya, S. V. R. Anand, A. Kumar, J. Kuri, “Experiences with a centralized scheduling approach for performance management of IEEE 802.11 Wireless LANs,” *IEEE/ACM Trans. Netw*, vol. 21, no. 2, pp. 648-662, 2013.

[113] K. Karanasos, S. Rao, C. Curino, C. Douglas, K. Chaliparambil, G. M. Fumarola, S. Heddaya, R. Ramakrishnan, S. Sakalanaga, “Mercury : Hybrid centralized and distributed scheduling in large shared clusters,” In *Proceedings of the 2015 USENIX conference on*

usenix annual technical conference (Berkeley, CA, USA, 2015), USENIX ATC ’15, USENIX Association, pp. 485-497, 2015.

[114] W. Niu, Y. Xu, J. Wu, Y. Tan (2015) “A centralized scheduling approach to multi-agent coordination,” In *Proceedings of the 21st International conference on industrial engineering and engineering management*, pp. 171-175, 2014.

[115] D. Fu, J. Yang, X. Ling, H. Zhang, “Load-aware hybrid scheduling in large compute clusters,” *IEEE Symposium on computers and communication (ISCC)*, pp. 791-796, 2016, doi: 10.1109/ISCC.2016.7543833.

[116] O. D. Ramos-Cantor, J. Belschner, G. Hegde, M. Pesavento, “Centralized coordinated scheduling in LTE-advanced networks,” LTE-advanced networks. *J wireless com network 2017*, vol. 122, 2017. https://doi.org/10.1186/s13638-017-0904-5

[117] J. Rintanen, “Scheduling with contingent resources and tasks,” *twenty-third International conference on automated planning and scheduling*, pp. 189-196, Rome, Italy, Jun. 2013.

[118] L. Zhou, Z. Yang, Y. Wen, and J. J. P. C. Rodrigues, « Distributed wireless video scheduling with delayed control information, » *IEEE Trans. circuits syst. video technol.*, vol. 24, no. 5, pp. 889-901, May 2014.

[119] A. Hamzadayi, G. Yildiz, “Modeling and solving static m identical parallel machines scheduling problem with a common server and sequence dependent setup times,” *Computers & industrial engineering*, vol.106, pp. 287-298, 2017.

[120] M. Skutella, M. Sviridenko, M. Uetz, “Unrelated machine scheduling with stochastic processing times,” *Math. oper. res.*, vol. 41, no. 3, pp.851-864, 2016.

[121] F. A. Chudak, “A min-sum 3/2-approximation algorithm for scheduling unrelated parallel machines,” *Journal of scheduling*, vol. 2, pp. 73-77, 1999.

[122] A. S. Schulz, M. Skutella, « Scheduling unrelated machines by randomized rounding, » *SIAM Journal on discrete mathematics*, vol.15, pp. 450-469, 2002.

[123] Mohring, R. H., A. S. Schulz, M. Uetz, “Approximation in stochastic scheduling: The power of LP-based priority policies,” *Journal of the ACM*, vol. 46, pp. 924-942, 1999.

[124] J. Sethuraman, M. S. Squillante, "Optimal scheduling of multiclass parallel machines," *Proceedings of the 10th annual ACM–SIAM symposium on discrete algorithms*, pp. 963-964, 1999.

[125] M. Skutella, "Convex quadratic and semidefinite programming relaxations in scheduling," *Journal of the ACM*, vol. 48, pp. 206-242, 2001.

[126] P. Schuurman, G. J. Woeginger, "Polynomial time approximation algorithms for machine scheduling : Ten open problems," *Journal of scheduling*, vol. 2, pp. 203-213, 1999.

[127] E.O. Oyetunji, "Some Common Performance Measures in Scheduling Problems: Review Article," *Research journal of applied sciences, engineering and technology*, vol. 1, no. 2, pp. 6-9, 2009.

[128] D. M. Abdelkader, F. Omara, "Dynamic task scheduling algorithm with load balancing for heterogeneous computing system," *Egyptian informatics journal*, vol. 13, pp. 135-145, 2012. [Online]. Available: www.elsevier.com/locate/eij , www.sciencedirect.com

[129] M. Liu, F. Zheng, C. Chu, Y. Xu, "Optimal algorithms for online scheduling on parallel machines to minimize the makespan with a periodic availability constraint," *Theoretical computer science*, vol. 412, pp. 5225-5231, 2011.

[130] D. Xu, Z. Cheng, Y. Yin, H. Li, "Makespan minimization for two parallel machines scheduling with a periodic availability constraint," *Computers & operations research*, vol. 36, pp. 1809-1812, 2009.

[131] T. C. E. Chang, C. J. Hsu, D. L. Yang, "Unrelated parallelmachine scheduling with deterioring maintenance activities," *Computers & industrial engineering* vol. 60, pp. 602-605, 2011.

[132] S. Jeng, C. J. Hsu, D. L. Yang, 'Unrelated parallel-machine scheduling with deteriorating maintenance activities," *Computers & industrial engineering*, vol. 62, pp. 1141-1143, 2012.

[133] D. L. Yang, T.C.E Cheng, S. J. Yang, C. J. Hsu, “Unrelated parallel-machine scheduling with aging effects and multi-maintenance activities,” *Computers & operations research*, vol. 39, no. 7, pp. 1458-1464, 2012, doi: org/10.1016/j.cor.2011.08.017.

[134] S. J. Yang, “Unrelated parallel-machine scheduling with deterioration effects and deteriorating multi-maintenance activities for minimizing the total completion time,” *Applied mathematical modelling*, vol. 37, no. 5, pp. 2995-3005, 2013.

[135] S. Gawiejnowicz, A. Kononov, “Complexity and approximability of scheduling resumable proportionally deteriorating jobs,” *European journal of operational research*, vol. 200, no. 1, pp. 305-308, 2010.

[136] D. Dolev, M. K. Warmuth “Scheduling precedence graphs of bounded height,” *Journal of algorithms*, vol. 5, pp. 12, 1984, doi: 10.1016/0196-6774(84)90039-7.

[137] G. Anastasi, M. Conti, M. Di Francesco, “Extending the lifetime of wireless sensor networks through adaptive sleep,” *IEEE Trans. industrial informatics*, vol.5, no. 3, pp. 351-365, 2009.

[138] C.-W. Tsai, J.J. Rodrigues : “Metaheuristic scheduling for cloud: A survey,” *IEEE Syst. J.* vol. 8, pp. 279-291, 2014.

[139] Y. Wang, J. Wang, C. Wang, X. Song, “Research on resource scheduling of cloud based on improved particle swarm optimization algorithm,” in *Advances in brain inspired cognitive systems*, pp. 118-125, 2013.

[140] H. Kushner, P. Whiting, « Convergence of proportional-fair sharing algorithms under general conditions, » *IEEE Transaction on wireless communication*, vol. 3, no. 4, 2004.

[141] L. Yonghe, E. Knightly, “Opportunistic fair scheduling over multiple wireless channels,” *INFOCOM 2003. twenty-second annual joint conference of the IEEE computer and communications*. IEEE societies. vol. 2, 2003.

[142] L. Xin, E. K.P Chong, N. B. Shroff, “A framework for opportunistic scheduling in wireless networks,” *Computer networks*, vol. 41, no. 4, pp. 451-474, 2003.

[143] G. Konstantinos *et al.*, “Opportunistic scheduling of control tasks over shared wireless channels,” *Cyber-physical systems (ICCPS)*, *2014 ACM/IEEE 5th International*

conference on cyber-physical systems (with CPS Week 2014), pp. 48-59, Apr. 2014. [Online]. Available: https://doi.org/10.1109/ICCPS.2014.6843710

[144] A. A. Bhorkar *et al.*, « Adaptive opportunistic routing for wireless ad hoc networks, » *IEEE/ACM Transactions on networking,* vol. 20, no. 1, pp. 243-256, 2012.

[145] Rasool, Jawad, *et al.*, "Opportunistic scheduling policies for improved throughput guarantees in wireless networks," *EURASIP Journal on wireless communications and networking* 2011, vol.1, pp. 1-18, 2011.

[146] L. Shihuan, L. Ying, R. Srikant, "Throughput-optimal opportunistic scheduling in the presence of flow-level dynamics," *IEEE/ACM Transactions on networking,* vol. 19, no. 4, pp. 1057-1070, 2011.

[147] C. Coskun, E.W. Knightly, "Opportunistic traffic scheduling over multiple network paths," INFOCOM 2004. *twenty-third annual joint conference of the IEEE computer and communications societies*, vol. 3, pp. 1928-1937, 2004.

[148] M. Wenguang, X. Wang, S. Wu, "Distributed opportunistic scheduling with QoS constraints for wireless networks with hybrid links," *IEEE Transactions on vehicular technology,* vol. 65, no. 10, pp. 8511-8527, 2016.

[149] T. Sheu-Sheu *et al.*, "Distributed opportunistic scheduling for ad-hoc communications under delay constraints," *2010 Proceedings IEEE, INFOCOM*, pp. 1-9, 2010.

[150] Garcia-Saavedra, Andres, *et al.*, "Distributed opportunistic scheduling: A control theoretic approach," *2012 Proceedings IEEE INFOCOM*, pp. 540-548, 2012, doi: 10.1109/INFCOM.2012.6195795.

[151] [151] W. S. Herroelen, R. Leus, "Project scheduling under uncertainty: Survey and research potentials", *European journal of operational research*, vol. 165, no. 2, pp. 289-306. 2005.

[152] S.B. Shaw, AK.Singh, "A survey on scheduling and load balancing techniques in cloud computing environment," *International conference on computer and communication technology (ICCCT)*, pp.87-95, 26-28 Sept. 2014.

[153] J. A. Ruiz-Vanoye, O. Díaz-Parra, “Similarities between meta-heuristics algorithms and the science of life,” *Central european journal of operations research*, vol. 19, no 4, pp. 445-466, 2011.

[154] T. Lantharthong, N. Rugthaicharoencheep, “Network reconfiguration for load balancing in distribution system with distributed generation and capacitor placement,” *World academy of science, engineering and technology international journal of electrical and computer engineering*, vol. 6, no. 4, pp. 409-414, 2012.

[155] B. Kaliski, “*Polynomial time*,” In: van Tilborg H.C.A. (eds) Encyclopedia of cryptography and security. Springer, Boston, May 2005.

[156] C. H. Papadimitriou, “*Computational complexity*,” Reading, Mass. : Addison-Wesley, 1994, ISBN 0-201-53082-1.

[157] K. Merhoum, M. Djeghaba, “Algorithme génétique pour le problème d’ordonnancement de type job-shop,” *4th International conference on computer integrated manufacturing, CIP’2007 Sétif*, Algeria, 03-04 Nov. 2007.

[158] H. B. Othman, Sur l’ordonnancement d’ateliers job-shop flexibles et flow-shop en industries pharmaceutiques Optimisation par algorithmes génétiques et essaims particulaires, Thèse de doctorat délivré conjointement par l’École centrale de Lille et l’École nationale d’ingénieurs de Tunis, Jul. 2009.

CHAPITRE IV. **UNE APPROCHE GLOUTONNE**

Ghassan Fadlallah, Djamal Rebaine and Hamid Mcheick *

Department of Computer Science and Mathematics, University of Quebec at Chicoutimi,

Chicoutimi, QC, G7H 2B1, Canada.

Ghassan-Mustapha.Fadlallah1@uqac.ca (G.F.); Djamal_Rebaine@uqac.ca (D.R.);

Hamid_Mcheick@uqac.ca (H.M.)

*****Correspondent: Hamid_Mcheick@uqac.ca .

Résumé : Les appareils intelligents et omniprésents ont récemment connu un développement technologique accéléré aux niveaux matériel, logiciel et des connexions sans fil. La promotion de divers types d'informatique mobile collaborative nécessite une mise à niveau de la connectivité réseau avec les technologies sans fil, aussi la communication améliorée pair-à-pair. L'informatique mobile nécessite également des méthodes d'ordonnancement appropriées pour accélérer l'implémentation et le traitement de diverses applications informatiques en gérant mieux les ressources du réseau. Les techniques d'ordonnancement sont pertinentes pour les modèles architecturaux modernes qui supportent le paradigme IoT, en particulier les architectures informatiques mobiles collaboratives intelligentes à la périphérie du réseau. A cet égard, des techniques d'équilibrage de charge sont également devenues nécessaires pour exploiter toutes les capacités des ressources disponibles et donc la rapidité de l'implémentation. Cependant, étant donné que le problème d'ordonnancement et d'équilibrage de charge, que nous avons abordé dans cette étude, est connu pour être NP-difficile, l'approche heuristique est bien justifiée. Nous avons ainsi conçu et évalué et validé un algorithme d'ordonnancement glouton et d'équilibrage de charge pour améliorer l'utilisation des ressources. Nous avons mené une étude comparative avec les

algorithmes « the longest cloudlet fact processing (LCFP), shortest cloudlet fact processing (SCFP), » et « Min-Min heuristic ». Le choix de ces trois algorithmes est basé sur l'efficacité et la simplicité de leurs mécanismes, comme rapporté dans la littérature, pour allouer des tâches aux appareils. La simulation que nous avons menée a montré la supériorité de notre approche sur ces algorithmes en ce qui concerne le critère de temps de réalisation global.

Contribution associée :

Ghassan Fadlallah, Djamal Rebaine, Hamid Mcheick, « Greedy Scheduling Approach for Peripheral Mobile Intelligent Systems ». IoT **2021**, 2, pp. 249-274. https://doi.org/10.3390/iot2020014

Préambule

Avec le développement technologique rapide des appareils mobiles et intelligents omniprésents et de leurs réseaux et connexions, cet article met en évidence la nécessité d'améliorer l'informatique mobile collaborative qui y est implémentée. Cela se fait en intégrant des technologies modernes de communication sans fil ainsi de pair-à-pair. L'informatique mobile nécessite également l'amélioration des techniques de planification et d'équilibrage de charge appropriées pour accélérer la mise en œuvre et le traitement de diverses applications informatiques offrant une meilleure gestion des ressources réseau. Les techniques d'ordonnancement sont pertinentes pour les modèles architecturaux modernes qui supportent le paradigme IoT, en particulier les architectures informatiques mobiles collaboratives intelligentes à la périphérie des réseaux. A cet égard, ces techniques sont devenues indispensables en exploitant toutes les capacités des ressources disponibles afin de renforcer la rapidité d'exécuter leurs tâches.

Vu qu'en abordant, dans cette étude, le problème d'ordonnancement et d'équilibrage de charge qui est connu pour être NP-difficile, l'approche heuristique est bien justifiée. Nous avons ainsi conçu, évalué et validé un algorithme d'ordonnancement gourmande et d'équilibrage de charge pour optimiser et améliorer l'utilisation des ressources. Nous avons mené une étude comparative par simulation, basé sur l'efficacité et la simplicité du mécanisme d'allouer des tâches aux appareils, des algorithmes suivants: « the longest cloudlet fact processing (LCFP), shortest cloudlet fact processing (SCFP), » et « min-min heuristic ». Les résultats ont montré la supériorité de notre approche sur ces algorithmes en ce qui concerne le critère de temps de réalisation global.

Cet article est organisé en six sections suivantes : Introduction occupe la section 1. La section 2 présente des travaux connexes, en particulier ceux pertinents pour notre algorithme proposé. La section 3 présente les algorithmes gloutons comme solution dans notre approche. La section 4 décrit notre contribution à l'ordonnancement et à l'équilibrage de charge dans les systèmes mobiles autonomes périphériques. A travers une étude expérimentale intensive, la section 5 valide notre approche, puis analyse et entreprend une comparaison entre notre solution et trois autres algorithmes. Enfin, la section 6 présente nos conclusions et perspectives.

A GREEDY SCHEDULING APPROACH FOR PERIPHERAL MOBILE INTELLIGENT SYSTEMS

Ghassan Fadlallah, Djamal Rebaine and Hamid Mcheick *

Department of Computer Science and Mathematics, University of Quebec at Chicoutimi, Chicoutimi, QC, G7H 2B1, Canada;

Ghassan-Mustapha.Fadlallah1@uqac.ca (G.F.); Djamal_Rebaine@uqac.ca (D.R.); Hamid_Mcheick@uqac.ca (H.M.)

* Correspondent: Hamid_Mcheick@uqac.ca .

Citation: Fadlallah, G.; Rebaine, D.; Mcheick, H. A Greedy Scheduling Approach for Peripheral Mobile Intelligent Systems. IoT **2021**, 2, 249-274. https://doi.org/10.3390/iot2020014

Academic Editor: Luca Bedogni

Received: 1 April 2021
Accepted: 27 April 2021
Published: 30 April 2021

Abstract: Smart, pervasive devices have recently experienced accelerated technological development in the fields of hardware, software, and wireless connections. The promotion of various kinds of collaborative mobile computing requires an upgrade in network connectivity with wireless technologies, as well as enhanced peer-to-peer communication. Mobile computing also requires appropriate scheduling methods to speed up the implementation and processing of various computing ap-plications by better managing network resources. Scheduling techniques are relevant to the modern architectural models that support the IoT paradigm, particularly smart collaborative mobile computing architectures at the network periphery. In this regard, load-balancing techniques have also become necessary to exploit all the available capabilities and thus the speed of implementation. However, since the problem of scheduling and load-balancing, which we addressed in this study, is known to be NP-hard, the heuristic approach is well justified. We thus designed and validated a greedy scheduling and load-balancing algorithm to improve the utilization of re-sources. We conducted a comparison study with the longest cloudlet fact processing (LCFP), shortest cloudlet fact processing (SCFP), and min-min heuristic algorithms. The choice of those three algorithms is based on the efficiency and simplicity of their mechanisms, as reported in the literature, for allocating tasks to devices. The simulation we conducted showed the superiority of our approach over those algorithms with respect to the overall completion time criterion.

Keywords: IoT; scheduling; makespan; greedy algorithms; pervasive systems; LCFP; SCFP; Min-Min algorithms.

IV.1. INTRODUCTION

Continual and accelerated innovation in communications and information technologies has led to an expansion of the modern Internet and its applications and extensions, such as the Internet of things (IoT). The IoT continues to grow rapidly. It has become the basis for the so-called fourth industrial revolution and the digital transformation of business and society [1]. It is an emerging paradigm aimed at providing appropriate and smart systems of objects [2]. Making objects smart may be interpreted in two ways. The first interpretation is about entrusting computing power to an object, making it autonomous by allowing it to make its own choices. The second interpretation, for its part, consists of allowing objects to communicate with the outside world and, if necessary, to communicate with machines that can calculate and make decisions. These machines can be located in cloud centers to limit their economic impact [3]. However, the proliferation of connected objects generates a huge amount of data that is difficult for resource limited IoT objects to manage. In addition, the exchange and security of this data is a challenge when it is loaded and uploaded between peripheral devices and cloud server centers. In this regard, artificial intelligence (AI) can make a significant contribution to solving this problem, where it can become, along with the IoT, an important achievement [4].

The IoT supports a wide range of distributed systems that are interconnected with various devices such as servers, database centers, and computers. This is in addition to more powerful and compact portable devices such as smart cell phones and personal digital assistants (PDA) [5]. It is obvious to researchers and specialists that the demand for these devices has increased steadily. Unfortunately, technological progress has not always followed this tendency. Therefore, scheduling and load balancing techniques [6-9] have been developed to close this technological gap.

The scheduling process is concerned with the allocation of tasks to devices over time, whatever the environment may be (deterministic, stochastic, or online), with respect to a given criterion. For the sake of clarity, let us recall the definitions of these environments [10]:

1. Deterministic scheduling: the data defining the problem are known in advance;
2. Stochastic scheduling: all or most of the parameters that describe the problem are random variables over known distributions.
3. Online scheduling: all or some of the parameters describing the model are known only at the time the decision has been made.

Improving the scheduling process requires a better understanding of the environment under study and depends on whether it is static (fixed and non-modifiable scheduling) or dynamic (re-evaluated online to respond to changes). Likewise, the environment treatment criteria on a local or distributed level can be classified into different types: (1) hard constraints (the constraints must be strictly respected), (2) soft constraints (there is a flexibility with respect to the constraints), (3) preemptive (tasks may be interrupted before completion), or (4) non-preemptive (tasks are executed without interruption until completion) [11-13].

Scheduling problems occur in various real-world applications, such as industrial production, hoist scheduling, airport control towers, assignment of processes on processors, and data transfer services. Scheduling problems were originally studied in the industrial sector, such as the manufacture and assembly of large generators, and this led to the introduction of critical path analysis (CPA) and the critical path method (CPM) [14,15]. Recently, scheduling problems has emerged as an active field of research in the IoT community [16] and in mobile collaborative computing models via architectural models of cloud, fog, and edge computing [6,17]. As these emerging architectural models of computing are distributed systems, we focused our study on scheduling parallel machine models.

The development of mobile devices and communication technologies has contributed to their growing utilization in collaborative applications that are complex and resource intensive [18-20]. This has led to innovative designs in mobile collaborative computing and interaction techniques between devices. Among the most important closely related techniques are those for the scheduling and load balancing of tasks. Indeed, the scheduling problem consists of organizing the task execution order on ready devices, while load balancing aims to balance the tasks between devices [11]. As pointed out in Mishra *et al.* [21], the load balancing task determines where certain applications should be executed; it is used for distributing workloads across computing resources.

In this paper, we focus on establishing collaborative mobile systems based on modern technologies such as Pycom's Lopy4 and direct radio communication by devices to maintain wireless connectivity in different environments, even in the harshest ones [22, 23]. In the present study, we develop a new scheduling algorithm based on a greedy approach, which is well suited to these environments in terms of speed, quality, and ease of decision making at every step [24-26]. This approach enables small mobile devices to process tasks that match their capabilities where they have resource constraints in the IoT paradigm. Therefore, we adopted a strategy that consists of dividing the task and device groups into corresponding subgroups in terms of task size and device capacity. This algorithm aims to ensure load balancing through a technique that chooses among these corresponding subgroups the devices with the longest execution time first to allocate them to the appropriate tasks.

We conducted an experimental study (simulation) to compare, with respect to criteria based on the overall completion times, three existing solutions with our approach to improve the scheduling and load balancing techniques on the available devices of these systems. Let us note that the choice of these three algorithms was mainly based on the simplicity, efficiency, and speed, as well as the mechanism for allocating devices to tasks [27, 28]. We believe that the present study provides a path to further research on the development and efficient use of mobile and pervasive computing devices in crucial and urgent situations, such as highly intensive communications that require high bandwidth, smart healthcare, smart cities, harsh cases, and smart cars.

This paper is organized as follows. Section 2 presents the related works. Section 3 introduces greedy algorithms as a solution in our approach. Section 4 describes our contribution to scheduling and load balancing in peripheral autonomous mobile systems. Through an intensive experimental study, Section 5 validates our approach, then analyzes and undertakes a comparison between our solution and three other algorithms. Finally, Section 6 presents our concluding remarks and perspectives.

IV.2. RELATED WORKS

In this section, we present a review of the literature on scheduling techniques applied in the IoT field. The algorithms presented in [11] show that the tendencies in this area are evolving more toward approximate (near-optimal) solutions that are generated within a reasonable time. The main reason for these trends is that most scheduling problems are difficult to solve from the computational point of view [29]

Le *et al.* [30] pointed out that source code optimization is an emerging technique used to reduce energy consumption in parallel with increasing the storage and power capacity of mobile devices. This improvement may be achieved by partitioning tasks into smaller tasks.

Tableau IV-1 Classification of task scheduling algorithms

Categories of scheduling algorithms	**Description**
Immediate scheduling	Direct scheduling of new tasks-upon their arrival on VMs.
Batch scheduling of tasks	The tasks are pregrouped into batches before they are sent.
Static scheduling	The strategies of scheduling within this environment are usually based on information known beforehand about the system's global state.
Dynamic scheduling	Does not require current information about the global state of the system. In dynamic scheduling, tasks are distributed in terms of the capacity of the available VMs.
Preemptive scheduling	Portions of a task executed on a resource are resumed later on the same or another resource.
Non-preemptive scheduling	Requires processing the entire task without interruption until its completion on the same resource.

The classification of task scheduling algorithms differs depending on the viewpoints of the authors. The algorithms fall into six categories [31], which are presented in Tableau IV-1.

In cloud computing, there are three levels suitable for the task scheduling system [31]:

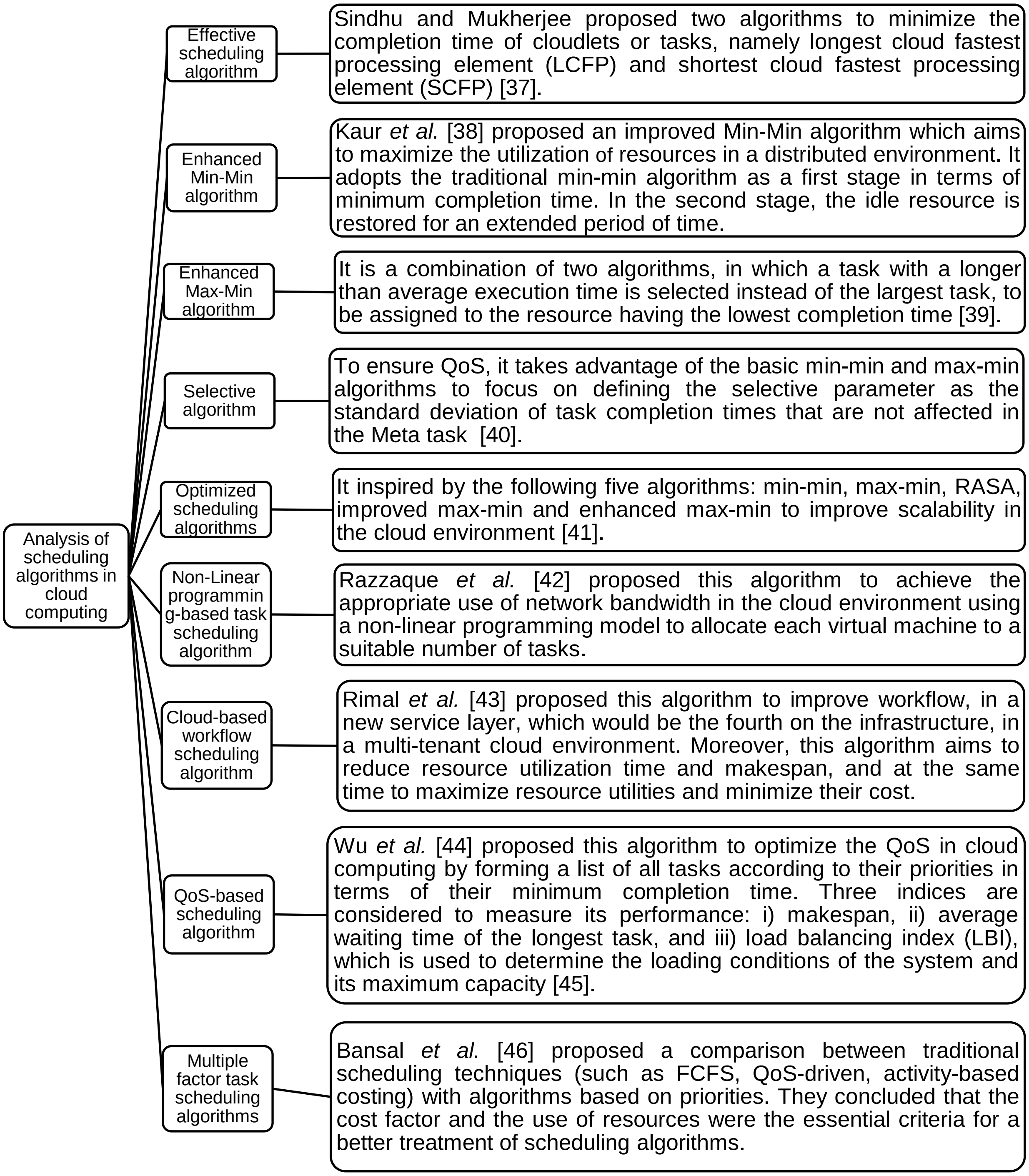
1. The first level is the set of tasks (cloudlets) to be executed.
2. The second level is the process of appropriately allocating resources to tasks to optimize the use of these resources with respect to the makespan or overall completion times.
3. The third level is the use of a set of virtual machines (VMs) to perform tasks.

In Sharma *et al.* [32], it is stated that there are several task scheduling techniques available for scheduling tasks in a cloud environment under the following categories:

1. Heuristic techniques can be divided into two subcategories. The first one is a traditional technique for scheduling various tasks, such as FCFS, round robin (RR), and SJF. This approach is simple and imperative, but it always approaches the local optima [33]. The second category uses a random sample to find the optimal or near-optimal solution. Some of its techniques are min-min, max-min, improved max-min [34], and min-min based on priorities. These techniques generate better results than traditional approaches [35];
2. Metaheuristic algorithms generally have functionalities that are like aspects of biological science. They are classified into three categories: (1) metaheuristics, such as genetic and transgenic algorithms, based on gene transfer; (2) metaheuristics based on insect behaviors and their interactions, such as ant colony optimization, the firefly algorithm, bee marriage optimization algorithm, and bee colony algorithm; and (3) metaheuristics based on aspects of biological life, such as the tabu search algorithm, simulated annealing algorithm, optimization algorithm for particle swarms, and artificial immune system [36].

The main difference between heuristic and metaheuristic is that the former is problem-specific and generates, step by step, only one solution, while the latter is problem-independent and generates several solutions [32].

Recently, improvements made to scheduling algorithms in cloud computing were discussed and investigated extensively in the literature (see, for example, Sharma *et al.* [32]). The analysis of scheduling algorithms is summarized in Scheme 1.



Scheme 1 Categories of several task scheduling techniques in loud environment [37-46].

The need for the fair sharing of resources in wireless networks increases with both the increased demand for their use and the cost of their equipment as it becomes difficult to install more devices (Sherin *et al.* [47]). Therefore, the scheduling process becomes the most important factor for improving wireless resource management. Multiple users of the system can access their shared resources efficiently by this scheduling process. These resources are mostly limited when compared with the increasing number of users. Scheduling algorithms offer equitable user access and are essential for ensuring the provision of quality of service (QoS). Let us observe that researchers and practitioners are becoming more concerned with packet scheduling algorithms, which are highly essential for QoS. The design of scheduling algorithms has become more complex due to several factors, including the dynamism of mobile ad hoc networks (MANETs) and vehicular ad hoc networks (VANETs) with frequent topology changes and breakdowns in connectivity. A MANET has many applications, including disaster management, emergency relief, vehicular ad hoc network services (VANETs), war field communications, mobile teleconferencing, and electronic payments. QoS is essential in these real-time applications due to the limited resources and dynamic topology. QoS indicates the service performance level provided by the network to the end user. Depending on the application, it can be based on bandwidth, delay, packet loss, throughput, overhead, jitter, and so on.

One of the key features of ad hoc networks is their capability of operating without a standard infrastructure, and they can be deployed easily for various applications. However, it is difficult to provide QoS for all packets because of the dynamic nature of nodes [48]. Generally, scheduling algorithms manage changes in queuing dynamics and indicate the packets to be submitted from the queue. To ensure the QoS parameters, these algorithms are determined based on network requirements [49]. There are two major problems in multi-hop wireless networks: the packet (or package of information, as it is known in the jargon of computer networks [48]) and the channel scheduling. In this area, scheduling algorithms, generally considered to be non-preemptive, are based on analysis of the QoS parameters, such as the throughput rate, fairness in the network, bounded delay, and jitter. The traditional scheduling approach, which takes advantage of the priority lists, is mainly used in mobile ad hoc networks.

Wireless scheduling algorithms in mobile ad hoc networks are classified according to their application domains into two general classes: packet scheduling and channel-aware scheduling [47], as presented in Tableau IV-2 and Tableau IV-3 and Scheme 2.

Tableau IV-2 Class of channel-aware scheduling algo. in MANET

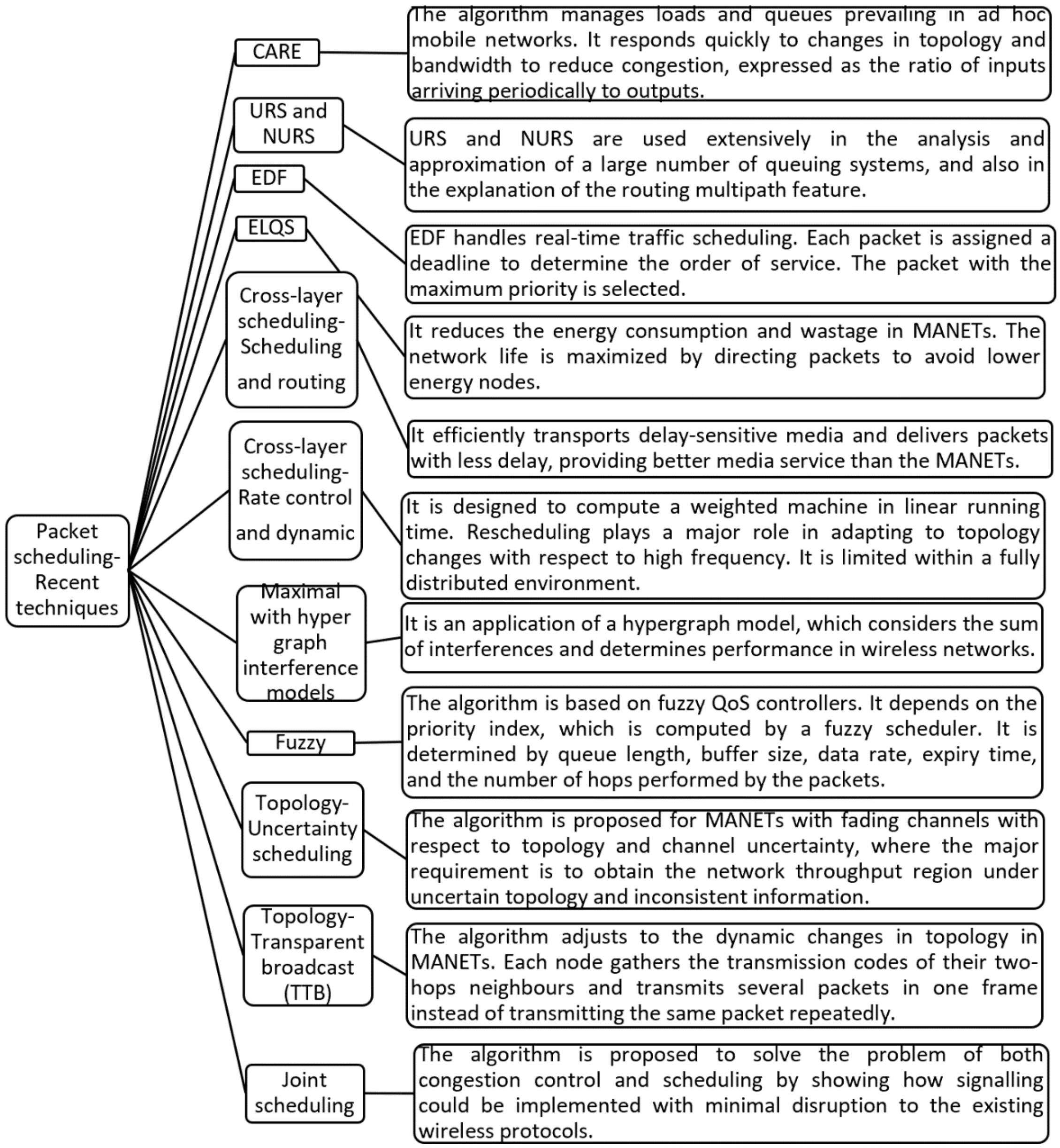
Class	Algorithm	Description
Channel-aware scheduling	CaSMA	CaSMA concentrates on the awareness and coordination of the end-to-end channel condition to reduce the accumulation of packets in the network and avoid congestion by increasing the number of completely served packets.
	Channel-aware AOMDV	It uses a preemptive handoff technique. In addition, it uses the non-fade duration to select the path during the route. Each node contains a table, which gives the information about the signal strength of the previously received packet.
	PALM	Power-aware link maintenance is based on an ad hoc on-demand distance vector routing algorithm and is responsible for power control with route maintenance. It establishes the routing mechanism in MANETs (Mobile ad hoc networks).

Tableau IV-3 Packet scheduling algorithms techniques in MANET

Class	Subclass	Algorithm	Description
Packet scheduling	Existing techniques	FIFO	In first-in, first-out, all packets are inserted into a single queue and processed in the order of their arrival. The delay is directly proportional to the length of the queue.
		Priority queuing	Packets are categorized and then grouped into queues with different priorities. Whereas high-priority packets are first processed, low-priority packets are likely to be dropped.
		WFQ	In the weighted fair queuing, packets are scheduled with bandwidth requirements and placed in the respective queues. Packets with a smaller end time are chosen as the next packet for transmission.
		CBWFQ	Class-based weighted fair queuing extends the WFQ functionality and supports user-defined traffic classes. CBWFQ services the class queue fairly based on the weight assigned to the queued packets.
		WHS	Weighted-hop scheduling gives high priority to data packets with only a few remaining hops to pass. A weighted round robin scheduler is used instead of static priority to give a chance to all service classes.
		WDS	The weighted distance scheduling algorithm considers the physical distance using a GPSR, where each data packet contains the destination address.
		RR	The round robin scheduling algorithm preserves the per-flow queues. It provides equal service opportunity among flows.

	Greedy scheduling	Each node redirects its own packet before forwarding the other nodes' packets, which are processed based on FIFO scheduling.
	LLQ	Low-latency queuing has a single strict priority queue for placing separate traffic classes in. This queue allows traffic-sensitive delay treatment before processing other queues. All other queues are regulated by the (percentage of) bandwidth.

Class/subclass	Algorithm/sub-algorithm	Description



Scheme 2 Class of recent techniques for packet scheduling algorithms in mobile ad hoc networks

IV.3. GREEDY APPROACH

Greedy algorithms are a problem-solving paradigm in which local optimal choices are made at every step, and they are expected to yield an optimal overall solution. However, in many problems, they only produce near-optimal solutions [50]. Let us recall that the greedy approach is well justified for NP-hard problems, and it is suited as a solving approach for real problems. However, it is worth noting that for problems with an optimal substructure, greedy algorithms are able to find globally optimal solutions [51].

For the objective of minimizing an unloading cost function, Mazouzi *et al.* [52] studied discharging policies to choose the tasks that must be discharged before selecting the assigned cloudlet, depending on the network and system resources. The unloading cost is declared to be a combination of task execution time and energy consumption. This problem is represented with mixed binary programming. As this problem is NP-hard, a distributed linear relaxation-based heuristic approach is proposed. Likewise, a greedy heuristic algorithm is proposed to solve the subproblems, and it calculates the best cloudlet selection and bandwidth allocation based on the task unloading costs.

Ayanzadeh *et al.* [51] proposed a novel hybrid approach, the quantum-assisted greedy algorithm (QAGA), to improve the performance of physical quantum annealers in the process of finding the global minimum in an Ising Hamiltonian model. Ising models are widely used in many areas of science. They routinely apply a Hamiltonian equation to, among other subjects, alloy thermodynamics and the thermal properties of solids [53]. The QAGA algorithm leverages the quantum annealers to better select candidates at each stage of the greedy algorithm. In fact, it consists of using a quantum annealer at each step to give rise to samples of the ground state of the problem. These samples are used to estimate the probability distribution of the problem variables. Then, it fixes these variables with insignificant uncertainties to move to the next step, where quantum annealing will solve a smaller problem using scattered couplings [51]. In their experimental study, the authors used a D-Wave 2000Q quantum computer with 2000 qubits and new control features to solve larger problems. The results showed that the QAGA approach could find samples with remarkably lower energy values compared to the better-known improvements in the field of quantum annealing.

In their study, Durmus *et al.* [54] aimed to support the assumption that the classic algorithms generate optimal solutions while the greedy heuristic algorithms generate proximate solutions. They argued that not only do the dimensions of problems increase, but the dimensions and number of constraints in packet programs are also limited, so it is difficult for the classic algorithms to provide the appropriate results. However, regardless of the dimensions or the number of constraints, the greedy approach generates appropriate results. As shown in the literature, both the former and the newer versions, such as the ones proposed by Akçay *et al.* [55] and Zhou *et al.* [56], work efficiently. Indeed, greedy algorithms are more efficient in comparison with other complex algorithms; they produce solutions to everyday problems within a reasonable time. Durmus *et al.* [54] applied greedy and some classical algorithms to the problems of integer linear programming and then compared the differences and similarities of the obtained results. Throughout this work, they classified the most-used algorithms in the literature, as displayed in Tableau IV-4, to which we added other greedy approaches and algorithms [51,57].

Tableau IV-4 Algorithm classifications and descriptions

Algorithm Category	Algorithm Name	Category Description
Classical algorithms (exact solution algorithms)	Rounding and graphical method, cutting plane, branch-bound, Balas, Lagrange, branch and cut, Benders decomposition method, and all-integer integer programming.	These algorithms are known in the literature as exact solution algorithms. Their drawback is often their high computational cost. However, they are effective for small and medium instances.
Metaheuristic algorithms	Steepest descent, Dantzig and Ramser's method, and tabu search, genetic, simulated annealing, ant colony, artificial bee colony, particle swarm optimization, and artificial neural networks.	These methods provide solutions close to the optimal one. They can solve a large variety of problems. They are conceptually simple. They can be flexed and adapted according to the problem under study. For example, in genetic algorithms, an analogy is developed between an individual in a population and a solution of a problem in the global solution space. In addition, the simulated annealing method is inspired by the process used in metallurgy to cool down steel. Likewise, particle swarm optimization is a cooperative, population-based global search swarm intelligent metaheuristic and population-based stochastic optimization technique, which is used in solving multimodal continuous optimization problems. Moreover, the tabu search algorithm selects a

		new search movement in such a way that temporally forbids the evaluation of previous solutions [11].
Greedy algorithms	Greedy algorithms are approaches or techniques in which we consider only one choice at each stage. Their strategy is to seek the best for the current state. It is known in the literature that they may produce optimal global solutions. We may cite the Dijkstra, Kruskal, Prim and Huffman algorithms [58].	They are the most-used algorithms in everyday life. They tackle problems with a given objective function; their strategy is to select each stage variable that has the most benefit. They are quite easy to apply and implement, the computational costs are quite low, and they can be applied to all kinds of problems.

IV.4. SCHEDULING ALGORITHMS IN PERIPHERAL AUTONOMOUS PERVASIVE SYSTEMS

The scheduling problems we discuss in this paper generally occur in the context of collaborative pervasive architectures within mobile and intelligent distributed systems. We focus our study particularly on the emerging architectures and technologies introduced in recent years, especially at the periphery of autonomous mobile systems.

IV.4.1) AUTONOMOUS MOBILE AND PERVASIVE ARCHITECTURE AT THE PERIPHERY

The problem with these architectures is how to make autonomous smart mobile systems more efficient in terms of completing tasks quickly while respecting their limitations (energy consumption, data storage, computing, connection preservation, and mutual charging between terminals, cloud centers, or other structures). The present study aims to efficiently address these issues in terms of partitioning, scheduling, and load balancing. This requires relying solely on these devices to accomplish the required tasks. This should consider their modest capabilities for innovation, on the one hand, in the system architecture in terms of equipment and maintaining connectivity and, on the other hand, in improving the scheduling and load balancing processes [59]. The scheduling problem we are addressing in this study consists in transferring as many tasks as possible to be managed on the appropriate devices of a peripheral autonomous mobile system. This problem is described as follows.

Assume there are *m* available devices $d_i, i = 1, \dots, m$ and *n* tasks $t_j, j = 1, \dots, n$. With no priority, the *n* tasks are processed by the *m* devices according to their due dates $ddt_j, j = 1, \dots, n$. The devices are also assigned to the tasks based on the device's ability to accomplish the task. This is done when the device free execution time is greater than the task execution time, as illustrated by Tableau IV-5.

Tableau IV-5 Parameters of devices and tasks

Task	t_1	t_2	t_3	t_4	t_5	-	-	-	t_n
ddt_j	4	2	6	8	10	-	-	-	5
fdt_j	5	8	3	11	9	-	-	-	13
Device	d_1	d_2	d_3	d_4	d_5	-	-	-	d_m
ftd_i	3	5	14	12	15	-	-	-	9

As usual, we assume that no device may process more than one task at a time, and no task is processed by more than one machine at a time. We seek an assignment of these tasks to these devices mainly to minimize the overall completion time. This kind of problem is known in scheduling theory as the parallel machine problem. This problem is known to be NP-hard with respect to most of the scheduling criteria, including the criterion we are addressing [29].

IV.4.2) Scheduling Algorithms Based on a Greedy Approach

By comparing different classes of scheduling algorithms, we concluded that greedy algorithms are appropriate for performing tasks on mobile systems within the autonomous mobile architecture at the network's periphery or edges. The reason is that these systems are compatible with greedy algorithms, which are suited for use as a selection algorithm to prioritize options in a search in simple problems. They are generally used in situations where the number of possibilities for improvement is too large to be considered in a meaningful way within a reasonable time [10]. Consequently, to remedy the problems of these networks, it is necessary to design an appropriate algorithm inspired by these greedy algorithms, which require reasonable processing time. For this purpose, we designed a scheduling algorithm based on the greedy approach, composed of one choice at each stage. This approach seems to be appropriate for avoiding complex calculations. Our algorithm was inspired by those like FCFS [46], the expected time to compute (ETC) matrix, minimum execution time (MET), minimum completion time (MCT) [60], Min-Min algorithm for task scheduling [33], longest cloudlet fastest processing (LCFP), and shortest cloudlet fastest processing (SCFP), proposed by Sindhu *et al.* [37].

To evaluate the performance of our algorithm, we selected three algorithms for comparison in an experimental study with our own, with respect to the quality of the generated solutions, simplicity of calculation, and speed of execution. These algorithms were chosen to cover the different mechanisms for allocating devices to tasks. They are described as follows:

1. The Min-Min algorithm selects the smaller tasks, with respect to processing time, to be executed first on the appropriate devices [61];

2. The LCFP algorithm chooses the longest cloudlet or task to be executed on the fastest processing element.
3. The SCFP algorithm directs the shortest cloudlet or tasks to the fastest processing element.

The approach we developed to solve the problem described above consists of the following steps: (1) problem modeling, (2) describing the greedy algorithm, and (3) the operating mechanism for solving the problem.

Before closing this subsection, let us say a word on the time complexity of the three algorithms. Each of these algorithms is dominated by the sorting procedure with respect to either the set of tasks or speed of the devices. However, the assignment of the tasks to the devices can be implemented in O(nm). Therefore, the time complexity of a sorting procedure is O (nm + max(m log m, n log n).

IV.4.2.1) PROBLEM MODELING

Our approach consists of several steps. The first step begins with breaking down large tasks into tasks based on the device's capabilities. Likewise, based on a criterion for the number of instructions, in the case of applications, and a criterion for size in the case of data. Then, the second step groups and classifies them as well as the network devices into three categories: large, medium, and small. In the third step, the tasks are classified into three categories-large, medium, and small-in such a way that the sizes of the tasks in a category are proportional to the capacities of the devices of the analogous category (Figure IV-1). These capacities are estimated based on a device score calculation. The most important aspect of a device's capacity is that it has sufficient execution time to accomplish the task assigned to it.

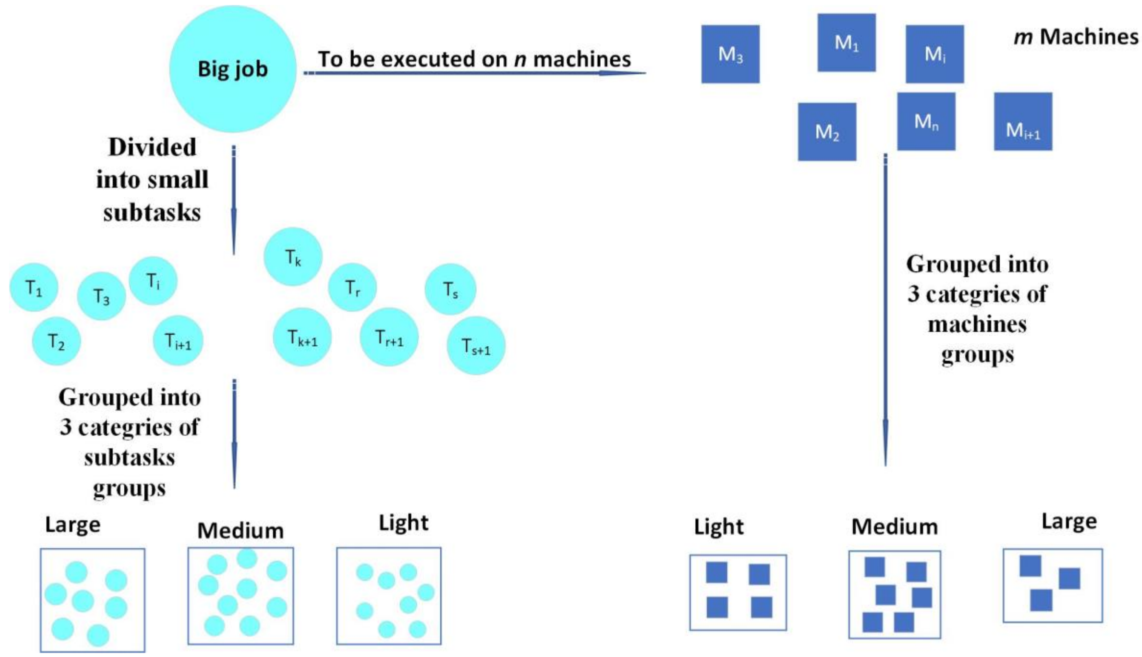


Figure IV-1 Devices & tasks Grouped based on device's capabilities. © 2021 Ghassan Fadlallah.

The score for each device *d* is determined as a function of the battery energy (ValBattery), storage (ValStok), bandwidth (ValBandwidth), RAM (ValRam), and CPU (ValCPU). To each of these factors is assigned a constant as a percentage value, depending on its importance for device effectiveness. The score is calculated as follows:

$$ValScore = ValBattery * PrcBattery + ValStok * PrcStoke + ValBandwidth * PrcBandwidth + ValRam * PrcRam + ValCPU * PrcCPU$$

This formula is implemented as in Algorithm1: CalculatingScores, which is called within Algorithm4: SchedulingThroughAnalogousSubgroups. It takes a device *d* as a parameter and obtains the capacity values of its factors previously mentioned to return its score value (ValScore).

The variables (criteria) used in our algorithms (Algorithms 1-4) are described in Tableau IV-6.

Tableau IV-6 The variables and their description

Variable	Explanation
ValBattery	Battery energy
ValStok	Storage capacity
ValBandwidth	Bandwidth
ValRam	RAM capacity
ValCPU	CPU capacity
ValScore	Score Value
TaskTime	Task execution time
DeviceTime	Available device execution time

Algorithm 1 CalculatingScores (d)

Begin

get factor capacity values of *d*;

$ValScore = ValBattery * PrcBattery + ValStok * PrcStoke + ValBandwidth * PrcBandwidth + ValRam * PrcRam + ValCPU * PrcCPU$;

return $ValScore$;

End

Ejaz *et al.* [62] formally defined the component execution time in terms of the processor speed, average number of instructions in the component, file size, number of processes running simultaneously on the mobile device, propagation delay, and transmission delay.

Based on [62], we supposed that the available device execution time would be proportional to its score as it depends on the same factors: the bandwidth and the capacities of the battery, storage or warehousing, and computing (RAM and CPU speed). This enabled us to estimate the execution time of a task as well as the available execution time of a device. Thus, we propose Algorithm: DeviceHasExecutionTime for the function that checks if this device has enough time to perform this task for use in our greedy Algorithm 4:

SchedulingThroughAnalogousSubgroups. Algorithm 2 uses the following parameters: TaskTime (task rpocessing time), DeviceTime (available device execution time), and a constant β (0.85 ≤ β ≤ 0.90), used to ensure that the device's ready execution time is sufficiently larger than the task processing time. It then returns a Boolean value B that indicates whether the machine has sufficient time to process the task in a non-preemptive mode.

Algorithm 2 DeviceHasExecutionTime (TaskTime, DeviceTime)

Begin
Boolean B = False;
Double β such that 0.85 ≤ β ≤ 0.92;
If TaskTime ≤ β*DeviceTime
B = True;
End if
Return B;
End

IV.4.2.2) PROPOSED GREEDY ALGORITHM

We propose a new approach to map tasks to devices such as laptops, smartphones, tablets, and similar machines, based on their categories and execution times. The approach we propose consists of partitioning each of the groups of tasks and devices into three subgroups and then sorting them in descending order, according to the processing times and device available execution time: LT (longest tasks), MT (medium tasks), ST (shortest tasks), LD (large devices), MD (medium devices), and SD (small devices). The next phase maps tasks from their subgroup to the appropriate devices in the subgroup of the same category and, if necessary, to other subgroups of devices having greater capabilities. Therefore, the tasks of the LT subgroup are mapped to the devices of the LD device subgroup. The MT subgroup tasks are then assigned to MD subgroup devices and, if required, to LD subgroup devices. The ST subgroup tasks are also mapped to the SD subgroup of devices and, if necessary, to MD and then LD device subgroups, as illustrated in (Figure IV-2).

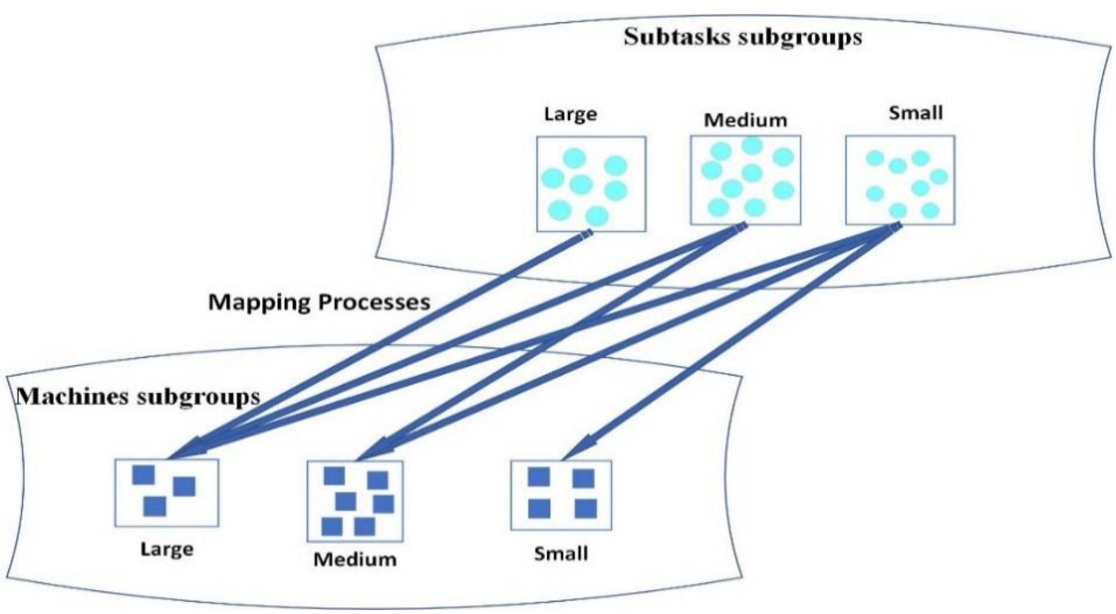


Figure IV-2 Mapping tasks to appropriate devices. © 2021 Ghassan Fadlallah.

Using these group patterns, we aimed to initiate and direct parallel processes of assigned devices to tasks in order to reduce the task completion times. We proceeded by restricting the search to the subgroup of devices that was similar to the subgroup of tasks or to larger subgroups of devices. Due to our prior classification, smaller sets of devices did not contain the appropriate devices. As previously mentioned, three parallel processes could be launched. The first was from the LT subgroup to the LD subgroup, but only if the devices in the MD and SD subgroups had no available execution time to perform LT tasks in a non-preemptive mode. Likewise, a second process could be started from the MT subgroup to the MD and LD subgroups. In parallel, a third process could go from the ST subgroup to the SD, MD, and LD subgroups.

The next phase, after the partitioning and the generation of these subgroups, was to proceed through the assignment of the devices to the tasks via Algorithm4: Scheduling ThroughAnalogousSubgroups.

We structured this algorithm in three parts, which could be executed in parallel with the aim of reducing the overall completion time of the tasks and thus save the device's energy. Many techniques, such as multithread, message passing interface (MPI), or ExaMPI, were used in parallel [63-65]. All the three parts were involved in allocating devices to subtasks from their similar groups at the same time viz. Part I from the large group of subtasks LT to the large group of devices LD, Part II from the medium group of subtasks MT to the medium group MD, and if necessary, to the large group of the devices LD, and Part III from the small group of subtasks ST to the small group SD, and then, as needed, to the MD and LD groups of devices. In addition, we performed load balancing on the devices by updating their information and sorting them again in descending order by the available execution time of the devices.

We also facilitated the understanding and effective use of this algorithm, as mentioned above, through the following algorithms, which will be called by the main algorithm:

- Algorithm1: CalculatingScores;
- Algorithm2: DeviceHasExecutionTime;
- Algorithm3: AllocatingDeviceToTask.

They act as follows. Algorithm 1 calculates the score of a device that would be proportional to its available runtime. This time will be compared to the time of Algorithm 2 to complete a subtask and check whether it is sufficient to execute it. If so, Algorithm 3 will allocate this device to this task.

Algorithm3 takes a subset of devices and a task as parameters, and it calls Algorithm2 to check whether the device in this subset has enough time to perform the corresponding task. When such a device is found, it is assigned to the task.

Therefore, the ready execution time of the device is reduced by the execution time of the task. Next, the SubGroupDevice device subgroup is sorted to perform load balancing.

Algorithm 3 AllocatingDeviceToTask (task, SubGroupDevice)

```
Boolean      B = false;
Double       Dim, DeviceTime, TaskExecutionTime;
Begin
Dim ← length (SubGroupDevice);
             For i ← 0 to Dim-1 Do
                  Get DeviceTime of device_i;
                  Get TaskExecutionTime of task;
                  If DeviceHasExecutionTime (task, device_i) == True
                       Assign device_i to task;
                       B = True;
```

```
                    DeviceTime = DeviceTime – TaskExecutionTime;
                    Set DeviceTime to device_i;
                    Sort SubGroupDevice;
                End If
            End For
            Return B;
        End
```

Algorithm 4 works, as is explained above, on the six sorted subgroups already generated (LT,

MT, and ST and LD, MD, and SD) to produce a plan for allocating the devices of those subgroups to their tasks while ensuring true load balancing.

Algorithm 4 SchedulingThroughAnalogousSubgroups

```
        Integer    isdone = 0, lg, j;
        String task;
        CalculatingScores (LD), CalculatingScores (MD), CalculatingScores (SD);
        while isdone < 3
Part I      while LT ≠ Ø Go
                lg ← length (LT);
                For j == 0, j < lg Do
                    If AllocatingDeviceToTask (task_j, LD) == True
                        Remove task_j from LT;
                    End If
                End For
                If LT == Ø
                    isdone++;
                End If
            End While
Part II     while MT ≠ Ø Go
                lg ← length (MT);
                For j == 0, j < lg Do
                    If AllocatingDeviceToTask (task_j, MD) == True
                        Remove task_j from MT;
                    Else
                      If AllocatingDeviceToTask (task_j, LD) == True
                        Remove task_j from MT;
                      End If
                End For
```

```
                    If MT == Ø
                        isdone++;
                    End If
              End While
Part III      while ST ≠ Ø Go
                    lg ← length (ST);
                    For j == 0, j < lg Do
                        If AllocatingDeviceToTask (task_j, SD) == True
                              Remove task_j from ST;
                        Else
                              If AllocatingDeviceToTask (task_j, MD) == True

                                 Remove task_j from ST;
                              Else
                                 If AllocatingDeviceToTask (task_j, LD) == True

                                    Remove task_j from ST;
                                 End If
                         End If
                    End For
              If ST == Ø
                 isdone++;
              End If
         End While
     End While
```

IV.4.2.3) OPERATING MECHANISM

Our greedy scheduling algorithm is designed to build the subgroups, as described in Algorithm4. The operating mechanism is carried out by launching a search from a task subgroup in a similar subgroup of devices and the larger one, if needed, to allocate the appropriate devices to the tasks. It is based on the technique of parallelism, considering the number of processors and kernels of the devices in addition to their load-balancing factor (as illustrated in Figure IV-3). The score of a device is used as a factor to determine the availability of execution time, which is needed to accomplish specific tasks, as shown in Algorithm2. This score represents the sum of the percentage of battery usage and the capacity of storage, computing, and connectivity (or bandwidth), as illustrated in Algorithm1. The mechanism we used to map tasks into the appropriate devices in similar subgroups reduced the completion times. This improves the system performance of mobile devices as it reduces the limited energy consumption of this system's resources.

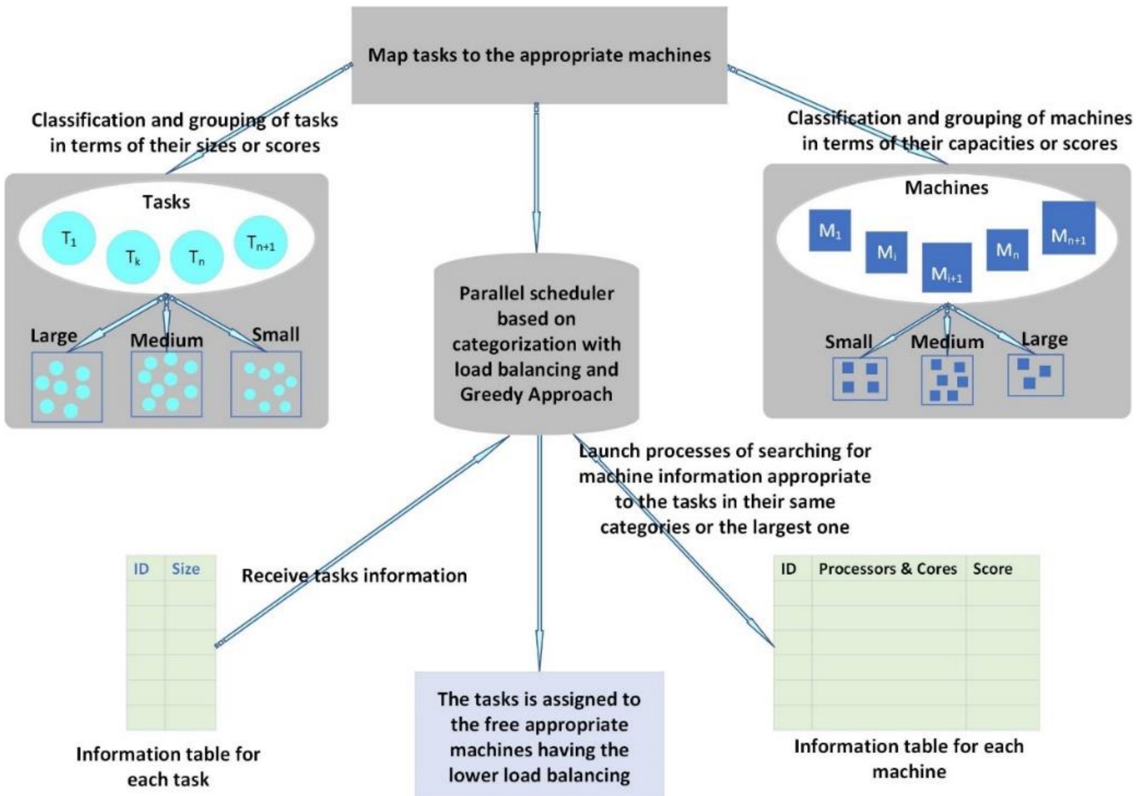


Figure IV-3 Mechanism for mapping tasks to appropriate devices. © 2021 Ghassan Fadlallah.

We close this section by mentioning the time complexity of our algorithm. To divide the tasks into the three classes, we needed to sort the tasks. This step could be done in O(n log n). The same argument goes for the three groups of devices with respect to their capacity scores. This step could be done in O(m). The rest of the algorithm (i.e., the assignment of those groups of tasks to the corresponding group of devices) could be implemented in $O(nm)$. Therefore, the overall time complexity of our algorithm was $O(nm + n \log n)$.

IV.5. SIMULATION

We present in this section the results of a simulation study we conducted to compare the performance of Algorithm4, as described above, with LCFP, SCFP, and Min-Min. In these algorithms, a variety of factors and their proper values are considered, such as the processing time, bandwidth, number of tasks, number of devices, sending and receiving tasks, waiting times in the queues of the devices, and capacity of each device.

Note that the simulation did not consider some factors in mobile systems such as device communication, bandwidth, the sending and receiving of tasks through networks, and their waiting time in the queues of the devices.

Even though we used a single computer in this experimental study instead of several devices to process all the tasks, we accurately calculated the values of the estimated factors of algorithm performance. We compared and analyzed the four algorithms based on the following criteria:

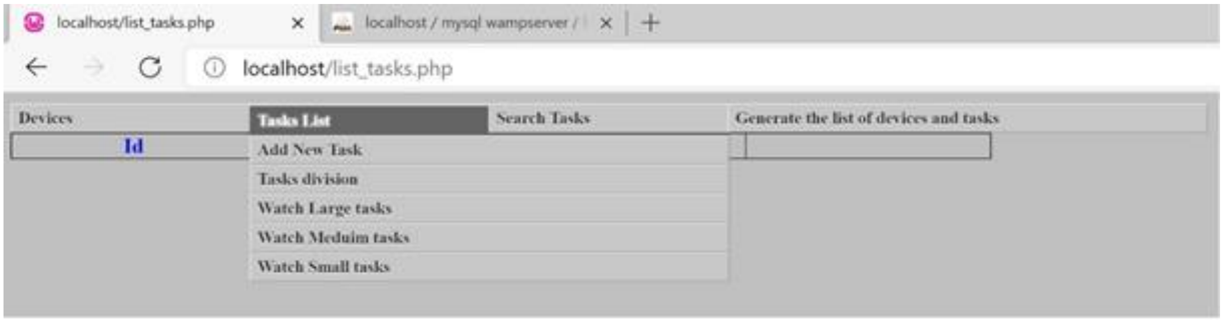
1. The overall completion time, known as the makespan M ($\mathrm{M} = \max(C_i : i = 1, \ldots, m)$, where C_i denotes the completion time of device i;
2. The standard deviation SD (SD = $\sqrt{\frac{1}{m}\sum_{i=1}^{m}(C_i - \bar{M})^2}$), , $\bar{M} = \frac{1}{m}\sum_{j=1}^{m} C_j$
3. The absolute difference AD between the maximum completion time M ($\mathrm{M} = \max(C_i : i = 1, \ldots, m)$) and the minimum completion time L ($\mathrm{L} = \min(C_i : i = 1, \ldots, m)$) (i.e., AD = |M-L|).

We then computed the percentage for the number of times that each algorithm produced the minimum values of these factors (makespan, SD, and AD). The value of the makespan indicates the time at which the entire set of tasks is completed, thus representing how efficiently the devices are used. The value AD represents the range of the completion times of the devices, thus representing how well-balanced the generated solution is. The standard deviation expresses the dispersion of the completion times of the devices around the makespan. A low value of standard deviation indicates that the completion times tend to be close to the value of the makespan, which is to say that the schedule is well-balanced.

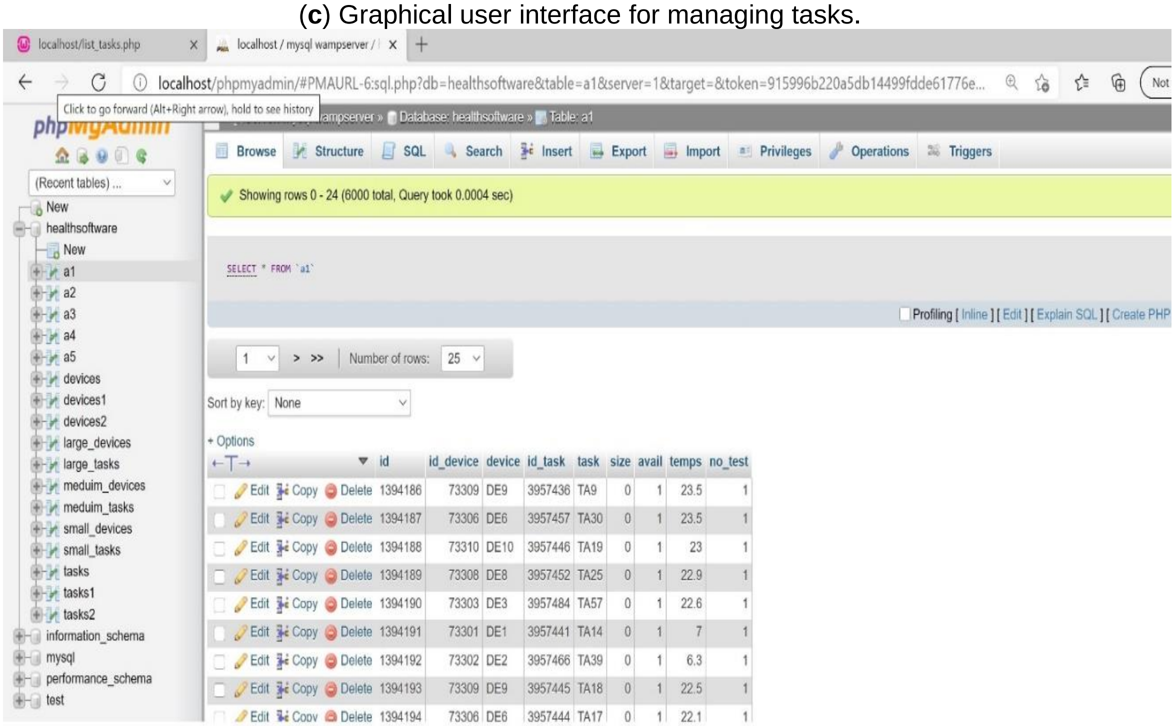
The experimental study we conducted used the WampServer web platform, an SQL server, and the PHP programming language to implement and validate the proposed algorithms. It was implemented on a laptop with the following characteristics: Intel® Core™ i7 processor, x64-based processor, Windows 10 Pro 64-bit OS, 2.3 GHz CPU, and 16 GB of RAM. Figure IV-4 presents the graphical interface for the simulation.

(**a**) Graphical interface for generating the list of devices and the tasks assigned to them.

(**b**) Graphical interface for choosing algorithms.



(**c**) Graphical user interface for managing tasks.



(d) Graphical user interface of the SQL database.

Figure IV-4 Four user interfaces in the software application menu. © 2021 Ghassan Fadlallah.

To ensure the reliability of our method, it is necessary to mention the external factors that might have affected the tested algorithms. Among those factors, we may mention the computer self-configuration resources in execution times. Thus, they influenced the operating system in terms of scheduling their execution in conjunction with these algorithms. As these factors operate automatically while the algorithms are executed, they affect the length of the processing time. Therefore, considering these factors, in terms of the impacts they may have had on the implementation of the algorithms, enabled us to verify and confirm the credence of the results.

For each group of m devices (m = 60, 40, 30, 20, 10, 5), we randomly generated from [11,24] the estimated available time values for each device. Then, for each group of tasks (n = 200, 100, 80, 60, 40, 20, 10), we generated 100 instances, for which the processing times were randomly drawn from [11,53] and on which the four algorithms were executed.

For each instance, we compared the makespan values of the four algorithms. We increased by one the score of the algorithm with the minimum makespan. If two algorithms had the same minimum makespan, we compared their standard deviations and increased by one the score of the algorithm with the minimum standard deviation. If they again produced the same minimum standard deviations, we compared their absolute difference and increased by one the score of the algorithm with the minimum value. If they still had the same absolute difference, we increased by one the score of the algorithm with the minimum score. The best algorithm was the one that had the highest score over the 100 instances.

In the following, we first present a comparison of the minimum makespan, the standard deviation, and the absolute difference with respect to the completion times of the four algorithms (LCFP, SCFP, Min-Min, and Algorithm 4). Next, we also present, for the same instances, the results with respect to the running times of these algorithms.

Displayed in Figure IV-5 are the different tables that summarize the results of the experimental study which satisfied the criteria (completion time M, standard deviation SD, and absolute difference AD).

Tasks Number	LCFP	SCFP	Min-Min	Our Algorithm
200	**0.51**	0	0	**0.49**
100	0	0	0	**1**
80	0	0	0	**1**
60	0.25	0	0	**0.75**
40	**0.67**	0	0	0.33
30	**0.51**	0	0	**0.49**
20	0.31	0	0	**0.69**
10	0.19	0	0	**0.81**

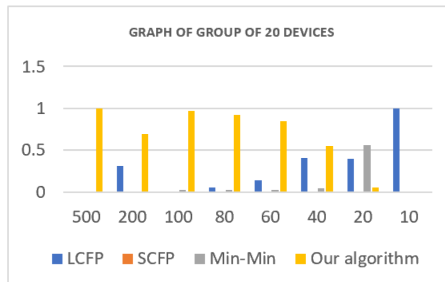
(a) **Group of 5 Devices.**

Tasks Number	LCFP	SCFP	Min-Min	Our Algorithm
200	**1**	0	0	0
100	0	0	0	**1**
80	0.44	0	0	0.56
60	0.101	0	0	**0.899**
40	0.47	0	0	**0.53**
20	**0.92**	0	0.01	0.06
10	0.28	0.32	0.1	0.1

(b) **Group of 10 Devices.**

Tasks Number	LCFP	SCFP	Min-Min	Our Algorithm
200	0.31	0	0	**0.69**
100	0	0	0.027	**0.973**
80	0.054	0	0.027	**0.919**
60	0.136	0	0.023	**0.841**
40	0.408	0	0.041	**0.551**
20	0.393	0	**0.554**	0.054
10	1	0	0	0

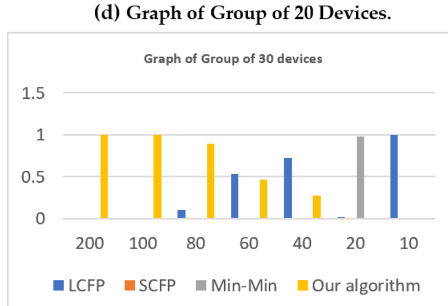
(c) Group of 20 Devices.



(d) Graph of Group of 20 Devices.

Tasks Number	LCFP	SCFP	Min-Min	Our Algorithm
200	0	0	0	**1**
100	0	0	0	**1**
80	0.107	0	0	**0.893**
60	**0.533**	0	0	0.467
40	**0.724**	0	0	0.276
20	0.022	0	0.978	0
10	**1**	0	0	0

(e) Group of 30 Devices.



(f) Graph of Group of 30 Devices.

Tasks Number	LCFP	SCFP	Min-Min	Our Algorithm
200	0	0	0	**1**
100	0	0	0	**1**
80	**0.8**	0	0	0.2
60	1	0	0	0
40	**0.696**	0	0	0.304
20	0	0	**1**	0
10	**1**	0	0	0

(g) Group of 40 Devices.

Tasks Number	LCFP	SCFP	Min-Min	Our Algorithm
200	0	0	0	**1**
100	0.077	0	0	**0.923**
80	**0.533**	0	0	0.467
60	**1**	0	0	0
40	**1**	0	0	0
20	**1**	0	0	0
10	**1**	0	0	0

(h) Group of 60 Devices.

Figure IV-5 Algorithm's performance relative to solutions quality. © 2021 Ghassan Fadlallah.

Figure IV-5 illustrates the performance of the four algorithms for various number of

Tasks Number	LCFP	SCFP	Min-Min	Our Algorithm
200	1.752	1.944	1.878	**0.679**
100	0.273	0.304	0.291	**0.058**
80	0.154	0.163	0.155	**0.05**
60	0.125	0.152	0.146	**0.084**
40	0.078	0.093	0.086	**0.065**
20	0.029	0.039	0.035	**0.02**
10	0.011	0.016	0.014	**0.009**

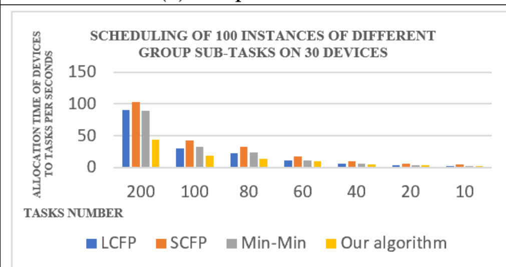
(a) Group of 5 Devices.

Tasks Number	LCFP	SCFP	Min-Min	Our Algorithm
200	1.109	1.298	1.157	**0.441**
100	0.319	0.361	0.31	**0.185**
80	0.193	0.224	0.198	**0.121**
60	0.155	0.19	0.16	**0.089**
40	0.071	0.109	0.078	**0.053**
20	**0.026**	0.055	0.03	0.033
10	**0.012**	0.018	0.013	0.02

(b) Group of 20 Devices.

Tasks Number	LCFP	SCFP	Min-Min	Our Algorithm
200	0.902	1.032	0.898	**0.445**
100	0.297	0.422	0.333	**0.186**
80	0.231	0.323	0.235	**0.139**
60	0.117	0.181	0.119	**0.096**
40	0.065	0.095	0.065	**0.049**
20	0.036	0.063	**0.035**	0.041
10	**0.023**	0.044	0.026	0.031

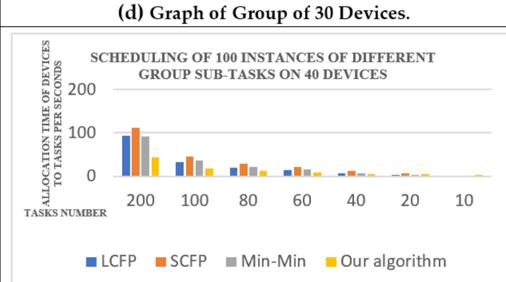
(c) Group of 30 Devices.



(d) Graph of Group of 30 Devices.

Tasks Number	LCFP	SCFP	Min-Min	Our Algorithm
200	0.931	1.129	0.922	**0.43**
100	0.335	0.462	0.371	**0.173**
80	0.197	0.294	0.212	**0.129**
60	0.138	0.214	0.155	**0.097**
40	0.061	0.121	0.063	**0.054**
20	0.035	0.061	**0.034**	0.046
10	**0.019**	0.022	0.021	0.028

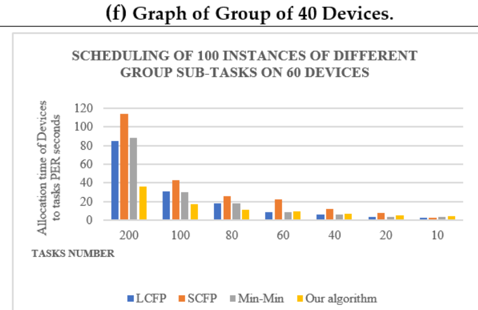
e) Group of 40 Devices.



(f) Graph of Group of 40 Devices.

Tasks Number	LCFP	SCFP	Min-Min	Our Algorithm
200	0.852	1.137	0.88	**0.362**
100	0.313	0.433	0.298	**0.17**
80	0.185	0.262	0.179	**0.116**
60	**0.086**	0.221	**0.086**	0.097
40	**0.063**	0.123	0.064	0.075
20	**0.039**	0.076	0.04	0.054
10	**0.03**	0.033	0.034	0.045

(g) Group of 60 Devices.

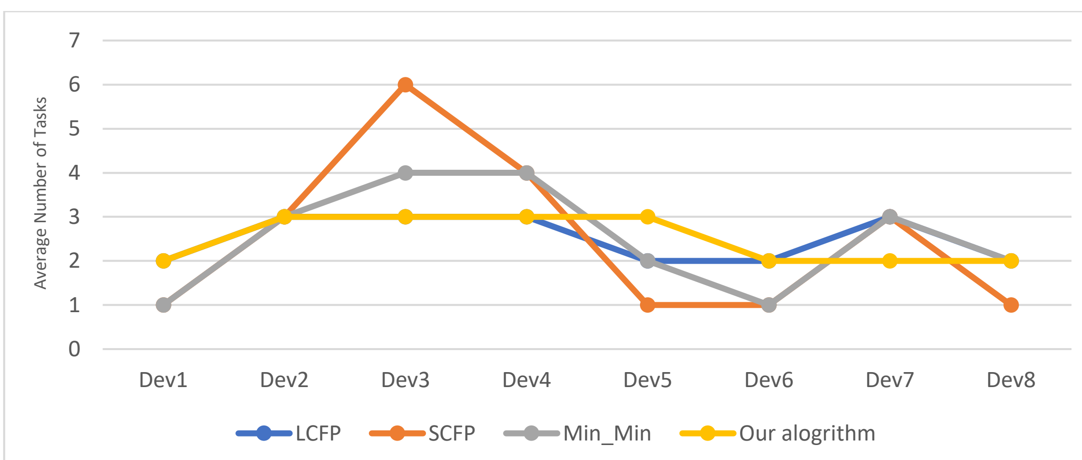


(h) Graph of Group of 60 Devices.

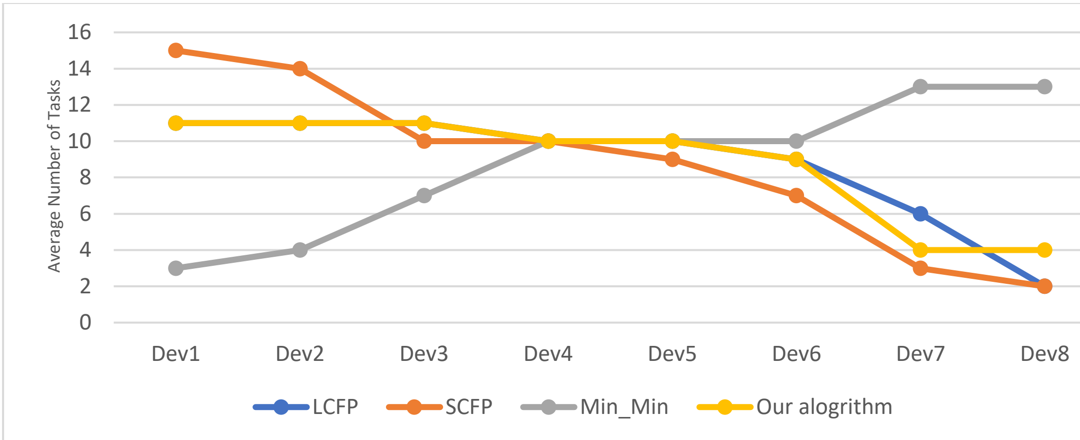
Figure IV-6 Algorithm's performance relative to running times. © 2021 Ghassan Fadlallah.

devices and tasks. Whereas, Figure IV-6 summarizes the average running times of these four algorithms.

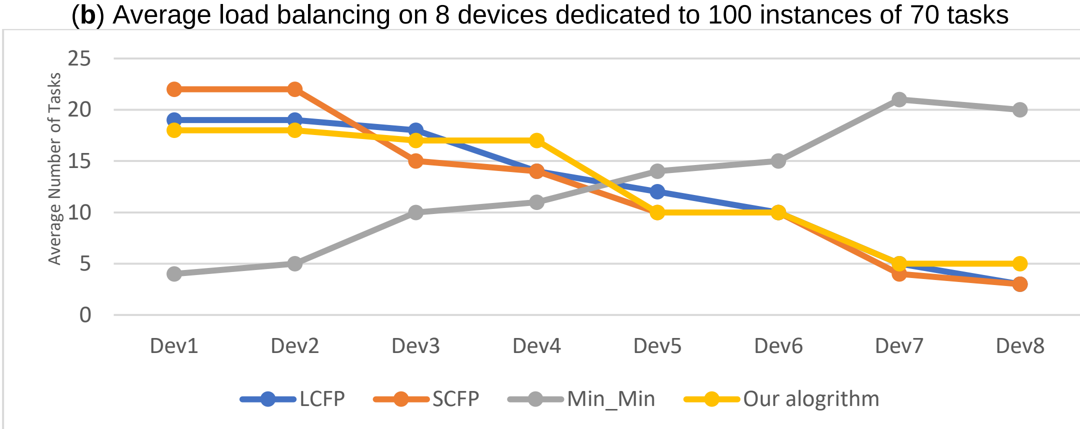
Regarding the scheduling solution, our algorithm succeeded in reducing the completion time of the tasks. At the same time, it improved the device load balancing by preparing devices with larger execution times to be chosen first by the algorithm and accomplish the appropriate tasks. Figure IV-7 shows our algorithm performance compared with the other algorithms for load balancing in the experimental results, with respect to 8 devices and 20, 70, 100, and 200 tasks. For the correct interpretation of these graphic lines in Figure IV-7, we proceeded from the fact that the ideal line representing the best performance is the horizontal line crossing them in the middle. Therefore, the best algorithm is the one with the closest graphic line to that ideal line, illustrated by our algorithm for most of the cases.



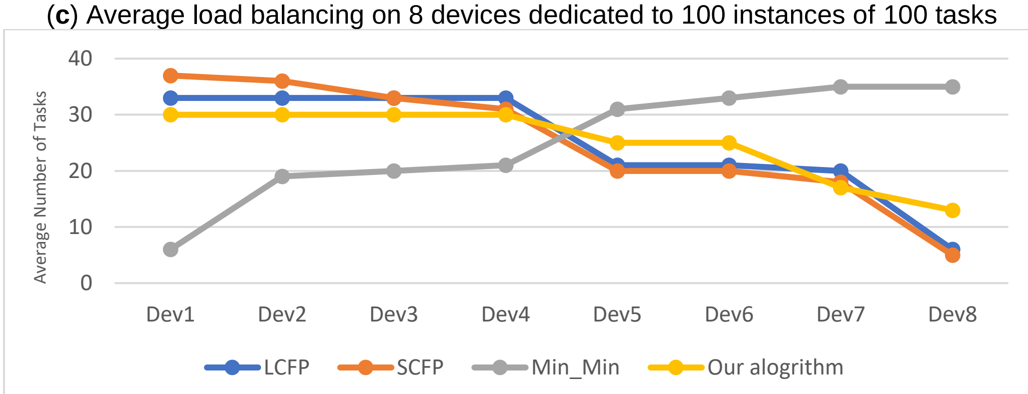
(**a**) Average load balancing on 8 devices dedicated to 100 instances of 20 tasks



(**b**) Average load balancing on 8 devices dedicated to 100 instances of 70 tasks



(**c**) Average load balancing on 8 devices dedicated to 100 instances of 100 tasks



(**d**) Average load balancing on 8 devices dedicated to 100 instances of 200 tasks

Figure IV-7 Experimental load balancing of the four algorithms. © 2021 Ghassan Fadlallah.

The results in these above tables clearly show the superiority of our algorithm. Indeed, our algorithm became more advantageous whenever the difference between the number of tasks and the number of devices became larger (i.e., in cases where the number of tasks is greater than the number of devices). This advantage is directly proportional to the increase in this difference in favor of tasks. Whenever this difference is large, the results generated by our algorithm are the best. The main reason for this is that the device and task groups were partitioned into three similar and relatively small subgroups. This reduced the time it took to navigate through them and find the appropriate devices to execute the tasks. The performance of the remaining algorithms, in decreasing order, is as follows: LCFP, Min-Min, and SCFP. From the above tables, it is easy to observe that our algorithm outperformed the other three algorithms with respect to three criteria: makespan, standard deviation, and the maximum of the absolute difference in completion times, whatever the number of devices or the number of tasks.

To summarize the results, our contribution is made through a dedicated approach adapted to the ubiquitous devices widely used in the IoT paradigm, particularly at the periphery of their communication networks. The goal was to build a new heuristic algorithm and compare it with the other scheduling heuristic algorithms mostly used in the literature. The main advantages of this approach are brought up through the reduction of the size of the tasks to be performed and breaking them into smaller subtasks. In addition, the process of dividing the groups of subtasks and devices into smaller sub-groups makes it possible for a parallel search to get the desired solution. Proceeding as such makes our algorithm faster and the quality of the generated solutions much better. Consequently, our approach saves a lot of energy, in addition to improving the efficiency of these devices.

IV.6. CONCLUSION AND FUTURE WORKS

The effectiveness of any scheduling algorithm depends, among other things, upon the overall completion time. Partitioning large tasks into smaller tasks plays an important role in getting these tasks done with shorter completion times. Greedy scheduling algorithms are a popular and widely used approach that produces a balanced, near-optimal solution for scheduling problems. Based on this idea, we proposed a new greedy scheduling algorithm for autonomous smart mobile systems, especially those found at the network's periphery. It takes advantage of the ability of these devices to maintain connectivity through a P2P radio connection. This feature allows devices to continue to communicate and therefore to work with each other, even in the event of wireless failures or communication flooding on the network. Thus, they can always perform the necessary tasks at any time and in any situation. This algorithm is based on the idea of dividing large tasks into smaller tasks to facilitate and speed up the scheduling process and thus their rapid completion. It creates similar subgroups in terms of the size of the tasks and devices. Then, it allocates tasks to devices between similar subgroups and, if necessary, to the larger device subgroups. We have shown through a preliminary study that this algorithm is compatible with mobile systems having resource constraints. Due to the limited capacity of devices, they do not tolerate the long process of computing. Therefore, they are not able to adopt a complex scheduling method. Instead, they require a series of simple scheduling strategies. The preliminary study we conducted showed that our algorithm had a clear advantage over the LCFP, SCFP, and Min-Min algorithms. On the one hand, our algorithm generated small average times for allocating devices to tasks. On the other hand, our algorithm outperformed the three comparison algorithms we presented with respect to the makespan, standard deviation, and range of completion times criteria. Likewise, significant improvements in load balancing could also be added to these advantages.

Current research provides reliability in reducing the completion time for urgent requests and thus in speeding up their completion, especially in harsh environments. For future work, we aim to apply our algorithm to more scenarios and situations. In particular, we aim to use it with more autonomous intelligent mobile systems having a large number of tasks to be performed on multiple devices. These advantages of the proposed method could be added to other functionalities, such as maintaining connectivity and processing via parallelism.

Author Contributions: G.F. designed the proposed greedy algorithms and implemented the four algorithms; G.F., D.R and H.M. analyzed and validated the computational algorithms; G.F. compared these algorithms and wrote the manuscript in consultation with H.M. and D.R.; All authors read and agreed the published version of the manuscript.

Funding: This work is partially funded by the Natural Sciences and Engineering Research council of Canada (NSERC), the 'Fund of the Université du Québec à Chicoutimi (FUQAC) and the Décanat de la Recherche et de la Créaction (UQAC).

Institutional Review Board Statement: Not applicable but the 'Université du Québec à Chicoutimi requires that All research involving the participation of human beings, whether funded or not, conducted or supervised by its professors, employes, and students, be ethically reviewed. Links: http://recherche.uqac.ca/cer/ (accessed on Mar. 2021).

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable but we generated the data in a random way to simulate and compare the algorithms.

Conflicts of Interest: The authors declare that there is no conflict of interest associated with this publication.

REFERENCES

[1] A. Rayes, S. Salam "*Internet of Things from hype to reality-the road to digitization*," Springer international publishing: New York, NY, USA, 2016; doi:10.1007/978-3-319-99516-8.

[2] E.O. Makori, "Promoting innovation and application of internet of things in academic and research information organizations". *Libr. Rev*. vol. 66, pp. 655-678, 2017, doi:10.1108/L-01-2017-0002.

[3] G. Fadlallah, H. Mcheick, Dj. Rebaine, M. Adda, "Towards mobile collaborative autonomous networks using peer-to-peer communication." In *proceedings of the ICSENT 2018: Proceedings of the 7th international conference on software engineering and new technologies*, Hammamet, Tunisie, pp. 1-8, 26-28 December 2018, doi:10.1145/3330089.3330107.

[4] A. Ghosh, D. Chakraborty, A. Law "Artificial intelligence in Internet of Things," *CAAI Trans. intell. technol*. vol. 3, pp. 208-218, 2018, doi:10.1049/trit.2018.1008.

[5] C.S.R. Prabhu "Overview - Fog computing and Internet-of-Things (IOT)," *EAI Endorsed trans. cloud syst*., vol. 3, no. 10, 2017**,** doi:10.4108/eai.20-12-2017.154378.

[6] T. Wang, X. Wei, C. Tang, J. Fan, "Efficient multi-tasks scheduling algorithm in mobile cloud computing with time constraints," *peer-to-peer netw. appl*., vol. 11, pp. 793-807, 2018, doi:10.1007/s12083-017-0561-9.

[7] J. Lim, D. A Lee "load balancing algorithm for mobile devices in edge cloud computing environments," *Electronics*, vol. 9, no. 4, 2020.

[8] C. Arun, K. Prabu "Load balancing in mobile cloud computing a review," *Int. j. comput. sci. eng*., vol. 6, pp. 460-465, 2018, doi:10.26438/ijcse/v6i12.460465.

[9] W. Xianglin "Application scheduling in mobile cloud computing with load balancing," *J. appl. math. 2013*, vol. 2013, art. 590872, doi:10.1155/2013/409539.

[10] T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein "*Introduction to algorithms*," 3rd ed.; MIT Press: Cambridge, MA, USA, 2009.

[11] G. Fadlallah, Dj. Rebaine, H. Mcheick "Scheduling problems from workshop to collaborative mobile computing: A state of the art," *Int. j. compute. sci. inf. secur*., vol. 1*6*, pp. 47-69, 2018.

[12] P. Černý, E.M. Clarke, T.A. Henzinger, A. Radhakrishna, L. Ryzhyk, R. Samanta, T. Tarrach "From non-preemptive to preemptive scheduling using synchronization synthesis," *formal methods in system design,* vol. 50, no. 2, pp. 97-139, 2017, doi:10.1007/s10703-016-0256-5.

[13] P. Hosein, S. Boodhoo "Event scheduling with soft constraints and on-demand re-optimization," in *Proceedings of the 2016 IEEE international conference on knowledge engineering and applications (ICKEA)*, Singapore, pp. 62-66, 28-30 September 2016, doi:10.1109/ICKEA.2016.7802993.

[14] P. A Weaver "Brief history of sheduling-back to the future," in *proceedings of the myPrimavera conference*, Canberra, Australia, pp. 1-24, 4-6 April 2006.

[15] F.K. Levy, G.L. Thompson, J.D. Wiest "ABCs of the critical path method," *Harv. bus. rev*., vol. 4*2*, pp. 98-108, 1963.

[16] P. Suresh, J.V. Daniel, V. Parthasarathy, R.H. Aswathy "A state of the art review on the Internet of Things (IoT) history, technology and fields of deployment," in *Proceedings of the 2014 international conference on science engineering and management research (ICSEMR)*, Chennai, India, pp. 1-8, November 2014.

[17] D. Rahbari, M. Nickray "Low-latency and energy-efficient scheduling in fog-based IoT applications," *Turk. j. electr. eng. comput. sci*., vol. 27, pp.1406-1427, 2019.

[18] W. Zhang, Y. Wen, D.O. Wu "Collaborative task execution in mobile cloud computing under a stochastic wireless channel," *IEEE Trans. wirel. commun*., vol. 14, pp. 81-93, 2015.

[19] G. Coulouris, J. Dollimore, T. Kindberg, G. Blair "*Distributed systems: Concepts and design*," 5th ed.; Addison Wesley: Boston, MA, USA, 2012.

[20] C.M. Wang, S.F. Hong, S.T. Wang, H.C. Chen “A dual-mode exerciser for a collaborative computing environment,” in *proceedings of the 11th Asia-Pacific software engineering conference*, Busan, Korea, pp. 240-248, 30 November-3 December 2004.

[21] S. Mishra, N. Mathur “Load balancing optimization in LTE/LTEA cellular networks: A review,” *arXiv* 2014, arXiv:1412.7273.

[22] Pycom LoPy4 development board datasheet. [Online]. Available: https://docs.pycom.io/datasheets/development/lopy4 (accessed on 31 Jan. 2021).

[23] G.M. Bianco, A. Mejia-Aguilar, G. Marrocco “Radio wave propagation of LoRa systems in mountains for search and rescue operations,” in *proceedings of the 2020 XXXIIIrd general assembly and scientific symposium of the international union of radio science*, Rome, Italy, pp. 1-3, 29 August-5 September 2020, doi:10.23919/URSIGASS49373.2020.9232231.

[24] Y. Lou, J. Chen, L. Zhang, D. Hao “A survey on regression test-case prioritization,” *Advances in computers*. vol. 113, pp.1-46, 2019.

[25] Logistic application of greedy algorithms, Vargo, January 2, 2013. [Online]. Available: https://vargosolutions.com/logistic-application-greedy-algorithms/ (accessed on Jan. 10, 2021).

[26] J. Le “Greedy algorithm and dynamic programming,” *Experfy*, November 5, 2018. [Online]. Available: https://www.experfy.com/blog/bigdata-cloud/greedy-algorithm-dynamic-programming/ (accessed on Jan. 10, 2021).

[27] B.H. Malik, M. Amir, B. Mazhar, S. Ali, R. Jalil, J. Khalid “Comparison of task scheduling algorithms in cloud environment,” *international journal of advanced computer science and applications*, vol. 9, no 5, pp. 384-390, 2018. doi:10.14569/IJACSA.2018.090550.

[28] S. Zhuravlev, J.C. Saez, S. Blagodurov, A. Fedorova, M. Prieto “Survey of energy-cognizant scheduling techniques,” *IEEE trans. parallel distrib. syst*., vol. 24, pp. 1447-1464, 2013.

[29] P. Brucker “*Scheduling algorithms*,” 4th ed.; Springer: Berlin/Heidelberg, Germany, pp. 1-367, 2004, ISBN 978-3-540-20524-1.

[30] H. A. Le, A.T. Bui, N.T. Truong “An approach to modelling and estimating power consumption of mobile applications,” *Mob. netw. appl*., vol. 24, pp. 124-133, 2019.

[31] T. Aladwani “*Types of task scheduling algorithms in cloud computing environment, scheduling problems-new applications and trends*,” da Rosa Righi, R., Ed.; IntechOpen: London, UK; pp. 1-12, 2020.

[32] N. Sharma, S. Tyagi, S. Atri “A survey on heuristic approach for task scheduling in cloud computing,” *Int. J. Adv. Res. Comput. Sci*., vol. 8, pp. 1089-1092, 2017.

[33] E. Shimpy, M.J. Sidhu “Different scheduling algorithms in different cloud environment,” *Int. j. adv. res. comput. commun. eng*., vol. 3, pp. 8003-8006, 2014.

[34] U. Bhoi, P.N. Ramanuj “Enhanced max-min task scheduling algorithm in cloud computing,” *Int. j. appl. innov. eng. manag*., vol. 2, pp. 259-264, 2013.

[35] R.J.S. Raj, S.V.M. Prasad “Survey on variants of heuristic algorithms for scheduling workflow of tasks,” in *proceedings of the international conference on circuit, power and computing technologies [ICCPCT]*, Nagercoil, India, pp. 1-4, 18-19 March 2016.

[36] J.A. Ruiz-Vanoye, O. Díaz-Parra “Similarities between meta-heuristics algorithms and the science of life,” *J. cent. eur. J. oper. res*., vol. 19, pp. 445-466, 2011.

[37] S. Sindhu, S. Mukherjee “Efficient task scheduling algorithms for cloud computing environment,” in *high performance architecture and grid computing*; Springer: Berlin/Heidelberg, Germany; vol. 169, pp. 79-83, 2011.

[38] R. kaur, P.K. Patra “Resource allocation with improved min-min algorithm,” *int. j. comput. appl*., vol. 76, pp. 61-67, 2013.

[39] B. Santhosh, D.H. Manjaiah “An improved task scheduling algorithm based on max-min for cloud computing,” *int. J. innov. res. comput. commun. eng*., vol. 2, pp. 84-88, 2014.

[40] K. Etminani, M. A Naghibzadeh “Min-min max-min selective algorithm for grid task scheduling,” in *proceedings of the 2007 3rd IEEE/IFIP international conference in central Asia on internet*, Tashkent, Uzbekistan, pp. 1-7, 26-28 Sept. 2007.

[41] S. Mittal, A. Katal "An optimized task scheduling algorithm in cloud computing," in *proceedings of the IEEE 6th international conference on advanced computing (IACC)*, Bhimavaram, India, pp. 197-202, 27-28 Feb. 2016.

[42] A. Razaque, N.R. Vennapusa, N. Soni, G.S. Janapati, K.R. Vangala "Task scheduling in Cloud computing," In *Proceedings of the 2016 IEEE Long island systems, Applications and technology conference (LISAT)*, Farmingdale, NY, USA, 29 Apr. 2016.

[43] B.P. Rimal, M. Maier "Workflow scheduling in multi-tenant cloud computing environments," *IEEE Trans. parallel distrib. Syst*., vol. 28, pp. 290-304, 2017.

[44] X. Wu, M. Deng, R. Zhang, B. Zeng, S. Zhou "A task Scheduling Algorithm Based on QoS-driven in cloud computing," *Procedia comput. sci.*, vol. 17, pp. 1162-1169, 2013.

[45] T. Lantharthong, N. Rugthaicharoencheep "Network reconfiguration for load balancing in distribution system with distributed generation and capacitor placement," *World acad. sci. eng. technol. int. j. electr. comput. eng*., vol. 6, pp. 396-401, 2012.

[46] N. Bansal, A. Awasthi, S. Bansal "Task scheduling algorithms with multiple factor in cloud computing environment," in *Information systems design and intelligent applications*; Springer: New Delhi, India; vol. 433, pp. 619-627, 2016.

[47] B.C. Sherin, E.A. Mary Anita "A survey of scheduling algorithms for wireless ad-hoc networks," *Int. j. adv. sci. eng.*, vol. 4, pp. 776-787, 2018.

[48] N.I.M. Enzai, F. Anwar, O. Mahmoud "Evaluation study of QoS-enabled AODV," in *proceedings of the international conference on computer and communication engineering*, Kuala Lumpur, Malaysia; pp. 1254-1259, 13-15 May 2008.

[49] J.F. Kurose, K.W. Ross "*Computer networking a top down approach featuring the internet*," 2nd ed.; Addison Wesley: Boston, MA, USA, 2004.

[50] R.A. DeVore, V.N. Temlyakov "Some remarks on greedy algorithms," *Adv. comput. math.*, vol. *5*, pp. 173-187, 1996.

[51] R. Ayanzadeh, M. Halem, J. Dorband, T. Finin "Quantum-assisted greedy algorithms," *arXiv*, arXiv:1912.02362, 2019.

[52] H. Mazouzi, N. Achir, K. Boussetta "DM2-ECOP: An efficient computation offloading policy for multi-user multi-cloudlet mobile edge computing environment," *ACM trans. internet technol*., vol. 19, pp. 1-24, 2019, doi:10.1145/3241666.

[53] W. Huang, D.A. Kitchaev, S. Dacek, Z. Rong, A. Urban, S. Cao, C. Luo, G. Ceder "Finding and proving the exact ground state of a generalized Ising model by convex optimization and MAX-SAT," *Phys. Rev. B*, vol. 94, no. 13, 2016.

[54] B. Durmus, O. Guneri, A. Incekirik "Comparison of classic and greedy heuristic algorithm results in integer programming: Knapsack problems," *Mugla j. sci. technol*., vol. 5, pp. 34-42, 2019, doi:10.22531/muglajsci.469475.

[55] Y. Zhou, X. Chen, G. Zhou "An improved monkey algorithm for a 0-1 knapsack problem," *Appl. soft comput*., vol. 38, pp. 817-830, 2016.

[56] M. Pinedo "Offline deterministic scheduling, stochastic scheduling, and online deterministic scheduling: A comparative overview," in *Handbook of scheduling: Algorithms, models, and performance analysis*, Chapman & Hall/CRC: London, UK, 2004, ch. 38, pp. 38-1-38-12.

[57] S.A. Curtis "The classification of greedy algorithms," *Sci. comput. program*., vol. 49, pp. 125-157, 2003.

[58] S. Sahni, E. Horowitz "*Fundamentals of computer algorithms*," Computer science series, W. H. Freeman and Company New York,1984.

[59] Y. Chen, S. Yang, J. Hwang, M. Wu "An energy-efficient scheduling algorithm for real-time machine-to-machine (M2M) data reporting," In *Proceedings of the IEEE global communications conference*, Austin, TX, USA; pp. 4442-4447, 8-12 December 2014.

[60] S. Sheikh, A. Nagaraju “A comparative study of task scheduling and load balancing techniques with MCT using ETC on computational grids,” *Indian j. sci. technol*., vol. 10, pp. 1-14, 2017, doi:10.17485/ijst/2017/v10i32/110751.

[61] N. Sharma, S. Tyagi, S. Atri “A comparative analysis of min-min and max-min algorithms based on the makespan parameter,” *Int. j. adv. res. comput. sci*., vol. 8, pp. 1038-1041, 2017.

[62] E. Ahmed, A. Naveed, S.H. Ab Hamid, A. Gani, K. Salah “Formal analysis of seamless application execution in mobile cloud computing,” *J. supercomput.*, vol. 73, pp. 4466-4492, 2017.

[63] G. Xie, X. Xiao, H. Peng, R. Li, K. Li “A survey of low-energy parallel scheduling algorithms,” *IEEE Transactions on sustainable computing,* pp. 1-20, 2021, [Online]. Available: https://ieeexplore.ieee.org/document/9351758 .

[64] B.A. Mahafzah, R. Jabri, O. Murad “Multithreaded scheduling for program segments based on chemical reaction optimizer,” *Soft comput.*, vol. *25*, pp. 2741-2766, 2021, doi:10.1007/s00500-020-05334-4.

[65] A. Skjellum, M. Rüfenacht, N. Sultana, D. Schafer, I. Laguna, K. Mohror “ExaMPI: A modern design and implementation to accelerate message passing interface innovation,” Springer international publishing, 2020, ch. 11, pp 153-169.

CHAPITRE V. VERS DES RÉSEAUX AUTONOMES OMNIPRÉSENTS

Ghassan Fadlallah† Hamid Mcheick, Djamal Rebaine, Mehdi Adda*
Ghassan-mustapha.fadlallah1@uqac.ca, Hamid_Mcheick@uqac.ca,
DJamal_Rebaine@uqac.ca
Département d'informatique et de mathématiques, université du Québec à Chicoutimi, Chicoutimi, Québec, Canada.
Mehdi_Adda@uqar.ca
Département de Mathématiques, Informatique et Ingénierie, université du Québec à Rimouski, Rimouski, Québec, Canada

Résumé : Compte tenu du développement technologique émergeant dans les domaines des télécommunications et des appareils mobiles intelligents, le nombre de ces appareils connectés dans le monde augmente rapidement. De plus, la tendance à utiliser ces petits appareils est en constante augmentation proportionnellement à celle de leurs capacités et de leur efficacité. Les progrès dans ces domaines ont été une incitation et une raison pour améliorer les systèmes distribués, l'Internet des objets (IoT) et l'informatique collaborative mobile. Les progrès des technologies aux niveaux matériel et logiciel ont nécessité le développement de nouvelles normes de communication adaptées aux appareils qui sont limités en ressources surtout dans les secteurs d'énergie, de puissance de calcul, de mémoire et de bande passante. Ces normes de communication jouent un rôle crucial dans l'amélioration de l'IOT et d'autres architectures d'informatique collaborative mobile telles que Cloud, Fog, Edge et Mobile Edge computing. Dans cet article, nous passerons en revue les normes et protocoles de communication actuels en montrant son rôle dans le renforcement des réseaux de communication mobile dans les différentes architectures de l'informatique collaborative mobile et dans les différentes circonstances. Plus précisément, nous illustrerons comment ces normes et protocoles peuvent maintenir une connexion efficace même entre les appareils mobiles à la périphérie. Aussi, comment ils peuvent établir des réseaux mobiles autonomes en utilisant la communication peer-to-peer via Wi-Fi Direct et d'autres technologies.

Contribution associée :

Ghassan Fadlallah, Hamid Mcheick, Djamal Rebaine and Mehdi Adda. 2018. Towards Mobile Collaborative Autonomous Networks using peer-to-peer communication. In *Proceedings of ACM ICSENT conference (ICSENT'18). ACM, Hammamat, Tunis, 7 pages.* https://doi.org/10.1145/3200842.3200852

Préambule

Le nombre d'appareils mobiles intelligents et omniprésents connectés augmente rapidement, tout comme la tendance à les utiliser. Ceci est en vue du développement technologique émergent dans l'industrie des appareils et dans le domaine des communications. Les progrès dans ces domaines ont été à la fois une incitation et une raison d'améliorer les systèmes distribués, l'Internet des objets (IoT) et l'informatique collaborative mobile. Les progrès des technologies aux niveaux matériel et logiciel ont nécessité le développement de nouvelles normes de communication adaptées aux dispositifs limités en ressources, notamment en termes d'énergie, de puissance de calcul, de mémoire et de bande passante.

Dans cet article, nous montrons le rôle crucial de ces normes de communication dans le renforcement de la connexion dans les réseaux mobiles et les différentes architectures de l'informatique collaborative mobile telles que Cloud, Fog, Edge et Mobile Edge computing. Plus précisément, nous illustrerons comment ces normes et protocoles peuvent maintenir une connexion efficace entre les appareils mobiles même à la périphérie des réseaux. Ainsi, établir des modèles autonomes de l'informatique collaborative mobile à la périphérie des réseaux en intégrant la communication pair-à-pair via Wi-Fi Direct et d'autres technologies.

Nous avons présenté et expliqué notre approche proposée et ses sujets connexes à travers une introduction dans la première section et les six sections ayant successivement les titres suivants:

i) IoT : un aperçu, ii) Ordonnancement dans des architectures supportant l'Internet des objets (IoT), iii) Protocoles et technologies de communication dans l'Internet des objets (IoT), iv) Architectures et technologies informatiques supportant l'IoT, v) Nouvelle approche proposée : cadre collaboratif d'un réseau autonome d'appareils mobiles, et vi) Conclusion.

TOWARDS MOBILE COLLABORATIVE AUTONOMOUS NETWORKS USING PEER-TO-PEER COMMUNICATION*

Ghassan Fadlallah†
Department of Computer Science and Mathematics
Université du Québec à Chicoutimi
Chicoutimi, Quebec, Canada
Ghassan-mustapha.fadlallah1@uqac.ca

Hamid Mcheick, Djamal Rebaine
Department of Computer Science and Mathematics
Université du Québec à Chicoutimi
Chicoutimi, Quebec, Canada
Hamid_Mcheick@uqac.ca
DJamal_Rebaine@uqac.ca

Mehdi Adda
Department of Mathematics, Computer Science and Engineering
Université du Québec à Rimouski
Rimouski, Quebec, Canada
Mehdi_Adda@uqar.ca

ABSTRACT

Given the emerging technological development in the fields of telecommunication and smart mobile devices, the number of connected devices around the world is increasing rapidly. Moreover, the tendency to use these small devices is increasing steadily as their capabilities and efficiency increase. Progress in these areas has been an incentive and a reason for enhancing distributed systems, Internet of things and mobile collaborative computing. The advances in hardware and software technologies have necessitated the development of new communication standards adapted to devices constrained in resources in all areas, energy, computing power, memory and bandwidth. These communication standards have a crucial role in enhancing the IOT and other architectures of mobile collaborative computing such as Cloud, Fog, Edge, and Mobile Edge Computing. In this paper, we will review the current communication standards and protocols by showing its role in strengthening mobile communication networks in the different architectures of mobile collaborative computing and under the different circumstances. More specifically, we will illustrate how they can maintain an efficient connection even between mobile devices on the periphery and how they can establish autonomous mobile networks by using peer-to-peer communication via Wi-Fi Direct and other technologies.

CCS CONCEPTS

Computer systems organization; Embedded systems; Networks; Network reliability

KEYWORDS: Emerging architectures; Mobile Edge computing; open fog; cloudlet; mobile cloudlet; mobile collaborative Frameworks; Internet of Things (IoT); Peer-Peer communication

ACM Reference format:
Ghassan Fadlallah, Hamid Mcheick, Djamal Rebaine and Mehdi Adda. 2018. Towards Mobile Collaborative Autonomous Networks using peer-to-peer communication. In *Proceedings of ACM ICSENT conference (ICSENT'18). ACM, Hammamat, Tunis, 7 pages.* https://doi.org/10.1145/3200842.3200852

V.1. INTRODUCTION

Continuous and rapid technological innovation, especially in the fields of communication and information technology, has led to a radical expansion of modern Internet as the Internet of Things IoT.

In the current context of proliferation of connected objects, their expected numbers in 2019 and 2025 are respectively about 26.66 and 75.44 billion [11]. These Internet-connected objects are the infrastructure of modern models and emerging architectures of mobile collaborative computing (MCC) such as: Cloud computing and its derivatives and branches: fog, open fog, mobile edge, mobile edge computing, cloudlet, etc.

These architectures still have some drawbacks in terms of maintaining connectivity and making mobile and collaborative object networks (Internet of Things IoT) intelligent and autonomous. When there is a failure in the network, collaborative distributed systems must me smart and autonomous enough to exchange messages between different mobiles, devices and central systems. Making these systems smart can be achieved in many ways. The first issue is to balance their loads and provide computing power to each object to make it self-contained by performing its own tasks and choosing its collaboration with other objects. The second issue is to provide the capacity of communication for objects outside, depending on the need to communicate with a machine capable of performing their queries (calculation, storage and decisions). These machines can be a cloud server, an intermediate peripheral server (Fog or Edge, etc.) or surrounding objects.

We propose an emerging architectural framework to maintain connectivity and perform tasks collaboratively between these objects, especially in the case of connection problem due to an extreme event or the large number of connected objects that can reach more than 75 billion by 2025 [11]. This framework can reduce bidirectional loading to central service centers in case of cloud or those intermediate in case of fog, edge, mobile edge, etc. We opt for a reference architecture providing small objects with more autonomy and more collaboration with their close counterparts to answer the following questions:

- How can the connectivity of objects and networks be maintained despite different connection problems?
- How can the load of these small peripheral objects be reduced to the central service centers (Cloud) or intermediaries?
- How the energy consumption of these objects can be decreased by reducing their usage time based on the by development of more efficient scheduling and load-balancing algorithms?

The objective of our study is to contribute to the improvement of collaborative computing on the devices of autonomous or individual mobile networks connected to standard wireless or Wi-Fi Direct technology, making these networks more and more efficient and smart to meet the different requirements, especially urgent needs as in the case of disaster situations.

To achieve this goal, we will first establish a network of mobile devices that can operate independently from wireless base stations. Then we will break down the work to be done into small tasks according to the potentials and capabilities of these devices that will achieve them. Finally, we intend to develop an efficient scheduling solution for these tasks while ensuring load balancing according to the deterministic, stochastic or online environment.

The rest of the document contains the following sections and a conclusion: IoT: an overview, scheduling into architecture supporting Internet of Things (IoT), protocols and communication technologies into Internet of Things (IoT), computing architectures and technologies supporting IoT and new proposed approach: Collaborative framework of an autonomous network of mobile devices.

V.2.IOT: AN OVERVIEW

IoT plays an important role in improving our daily life in its various aspects. Using embedded ubiquitous devices, IoT allows things to act intelligently and make consensus decisions that benefit many applications. By this way, IoT transforms physical objects to smart when exploiting their underlying technologies such as ubiquitous and pervasive computing, embedded devices, communication technologies, sensor networks, Internet protocols and applications.

Salman and Jain [14] argued that, "as embedded powerful sensors and others, these devices transform from being passive observers to actively communicating, collaborating, computing and making critical decisions. However, these fundamental technologies, new computing paradigms, data analytics, lightweight communication, and internet protocols point to the need for specialized standards and communication protocols to handle the resulting challenges. Indeed, IoT standards and protocols are developed for different layers of the network stack including, among others, the following layer protocols: data link, medium access control (MAC), network routing, and network encapsulation, network session" [12, 14].

An absolute plethora of applications, devices and projects aims to fully exploit the IoT. In this way, and in frank terms, Cloud computing as a developed platform, is the vessel by which so much trouble, many problems and extra work can be avoided. However, most distributed object (DO) projects are, in some way, totally cloud-based to solve certain problems.

Therefore, we need more enhancement of cloud infrastructure to catch up with the amount of IoT developments. As well, growing up the cloud storage on a gargantuan level is needed to handle the IoT data output. Likewise, Cloud processing, the performance and response to execute requests have to be equal or better than the analysis and functionality requirements of devices [15].

V.2.1) IOT ELEMENTS

In [12], Al-Fuqaha *et al.* presented a model showing the six main elements needed to provide IoT functionalities. In this figure, we added the categories of these elements with their examples as shown in Figure V-1.

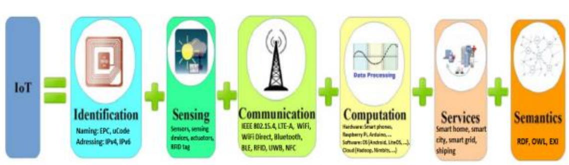


Figure V-1 Six main elements needed to provide IoT functionality. © 2021 Ghassan Fadlallah & IEEE Com.

V.2.2) IOT: UBIQUITOUS SMART DEVICES AND THEIR COLLABORATIVE COMPUTING

Collaborative computing is a term that describes a variety of activities where remote users interact with each other using distributed devices. These may be desktops and laptops, but due to the rapid technological development, the current and future trend is the use of portable and wearable sophisticated and smart devices (or pocket devices).

Moreover, the use of these small computing devices will eventually become so omnipresent in daily life that it will barely be noticeable. Indeed, to enable the collaboration of daily tasks, the collaborative environment should be lightweight and have ubiquitous components supporting a wide variety of devices and modes of interaction [1]. This computing paradigm still suffers from problems such as: interoperability, heterogeneity, security, failures, transparency, throughput, and connection traffic.

Ubiquitous infrastructure is ranked fifth in Gartner Inc.'s 2018 emerging technologies hype cycle. This is one of five distinct emerging technology trends revealed in this cycle, which blur the lines between humans and machines.

As it was argued: “The advent and mass popularity of cloud computing and its many variants have enabled the establishment of a permanent, available and unlimited infrastructure computing environment. This trend is enabled by the following technologies: 5G, Carbon Nanotube, Deep Neural Network ASICs, Neuromorphic Hardware and Quantum Computing. Technologies supporting ubiquitous infrastructure are on track to reach the peak and move fast along the Hype Cycle. 5G and deep neural network ASICs, in particular, are expected to reach the plateau in the next two to five years.” As well, “IoT platforms have crossed the peak by now, and we believe that they will reach maturity in the next five to 10 years.”[7]

In addition, the 2018 Gartner Top 10 Strategic Technology Trends report indicates that the fifth trend is Cloud to the Edge. It states that: “Connectivity and latency challenges, bandwidth constraints and greater functionality embedded at the edge favours distributed models. Enterprises should begin using edge design patterns in their infrastructure architectures - particularly for those with significant IoT elements. A good starting point could be using colocation and edge-specific networking capabilities. Edge computing speaks to a computing topology that places content, computing and processing closer to the user/things or “edge” of the networking. Cloud is

[7] https://www.gartner.com/en/newsroom/press-releases/2018-08-20-gartner-identifies-five-emerging-technology-trends-that-will-blur-the-lines-between-human-and-machine

a system where technology services are delivered using internet technologies, but it does not dictate centralized or decentralized service delivering services. When implemented together, cloud is used to create the service-oriented model and edge computing offers a delivery style that allows for executions of disconnected aspects of cloud service."[8]

V.3. SCHEDULING INTO ARCHITECTURE SUPPORTING INTERNET OF THINGS (IOT)

A powerful data management of cloud computing derived models (fog, edge, …) gives IoT an opportunity to revolutionize our lives. However, to meet the users' expectations, the traditional cloud computing server schedulers, which are not ready to provide services to IoT according to heterogeneous devices and not standardized applications, should be improved to efficiently schedule and allocate IoT requests. To efficiently provide desired services by considering priorities of requests, great efforts are needed on levels of the dynamic scheduling approach and scheduling algorithms. These algorithms and related analysis build efficient server schedulers which are adaptable to homogeneous and heterogeneous environments by considering system performance metrics, such as: drop rate, throughput, and utilization in IoT [15].

[8] https://www.gartner.com/smarterwithgartner/gartner-top-10-strategic-technology-trends-for-2018/

V.3.1) TOWARD NEW SCHEDULING PROCESSING METHODOLOGIES

It is well known that the need for tasks scheduling is directly proportional to the increasing use of dedicated devices to achieve them. Also, the smaller the devices, the greater the need to develop scheduling techniques by taking their limitations into account the capacity of the calculation, battery, memory, hard disk, CPU. To these are added the connection limitations such as availability, bandwidth, traffic, breakdown, etc.

An effective scheduling process depends, on the one hand, on the partition of the job to be performed into tasks compatible with the capabilities of the devices involved and on the other hand, on the load balancing of these devices already affected by partitioned tasks.

The scheduling process can usually be represented by a precedence graph. This graph is an oriented acyclic graph that specifies precedence constraints between tasks execution [2].

At the heart of the scheduling process is an algorithm on which it is based. These algorithms rely on the indispensable integration between tasks scheduling and devices load balancing. Figure V-2 summarizes and gives a wide overview of the state of these algorithms in the different architectures of the IoT. They classified according to two criteria: the environments and the quality of solutions.

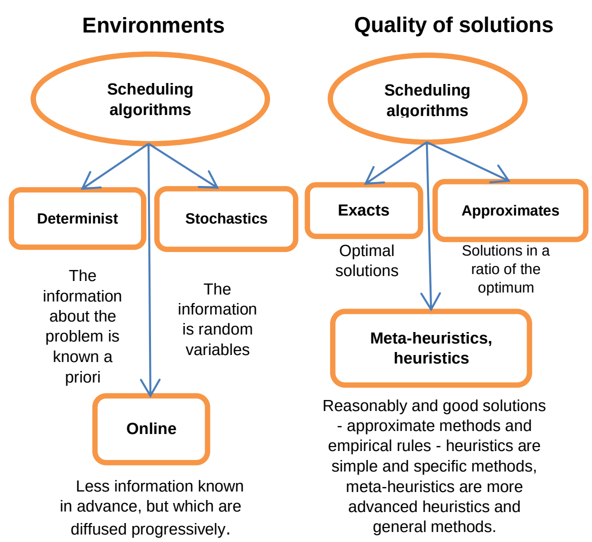


Figure V-2 Scheduling algorithms by environments & QoS. © 2021 Ghassan Fadlallah.

As we concluded in our article [3], scheduling algorithms tend to the approximate heuristic and meta-heuristic approaches, especially to obtain solutions close to the optimum in a short time. This is a critical factor in the completion of the job in progress in terms of the response speed, especially by small smart devices. Those devices require light algorithms whose execution is only a few units of time. Indeed, these small devices with limited capabilities require simple and uncomplicated scheduling algorithms that do not require much time to set up and run.

V.4. PROTOCOLS AND COMMUNICATION TECHNOLOGIES INTO INTERNET OF THINGS

Protocols and communication technologies for the IoT networks pose a real challenge to their functioning. However, many protocols and communication techniques have been established, especially the collaborative ones, and have reinforced these networks with big gains.

Tableau V-1 Connection technologies Part 1

Wireless connection technologies		
Name	**Description**	**Base station**
Wi-Fi	Wireless Fidelity is a wireless communication protocol standardized by IEEE 802.11. It has a frequency of 2.4GHz and 5 GHz, a range of 25-50 m (exterior: 75m-300m) and a bit rate of 11 Mbps-1,3 Gbps.	Yes
LTE	consists of two parts: a radio part (eUTRAN) and an EPC core (Evolved Packet Core). The radio part consists of eNode B, local or remote antennas, optical fiber links to remote antennas and IP links connecting eNode B to each other and with the core network via a backhaul network	Yes
LTE Direct	LTE Direct is a potential new standard peer-to-peer cellular technology which would make location-based services faster and more efficient. LTE Direct finds nearby devices directly over the air.	Non
WiMax	WiMAX is defined by IEEE 802.16, with frequencies of 2 - 66 GHz, a range of 50 km (line of sight) or 1 - 7 km (no line of sight) and data rates of 70 - 240 Mbps.	Yes
Flashlinq	FlashLinq system uses an OFDMA TDD wireless mode using a dedicated spectrum and can handle up to 1 000 devices simultaneously in a radius of about 1 kilometer.	Non
SigFox	The standard Sigfox has a frequency of 900MHz, a range of 30-50km (campaign), 3-10km (city) and a debit of 10-1000bps.	Yes

Tableau V-2 Connection technologies Part 2

Wireless connection technologies		
Name	**Description**	**Base station**
6LoWPAN	Network protocol based on IPv6 with low power consumption. IPv6 is the evolution of IPv4. It has been designed taking into account the specified mobile devices and automation.	Yes
LoRaWAN	The LoRaWAN standard has a variable frequency, a range of 2-5 km (city) -15 km (campaign) and a bit rate of 0.3-50 kbps.	Yes
ZigBee	The standard is ZigBee 3.0 which is based on IEEE802.15.4. It has a frequency of 2.4GHz, a range of 10-100m and a debit of 250kbps.	Yes
Z-Wave	The standard Z-Wave Alliance Standard ZAD12837 / ITU-T G.9959 has a frequency of 900MHz (ISM), a range of 30m and a bitrate of 9.6 / 40 / 100kbits	Yes
Bluetooth BLE	The standard Bluetooth 4.2 is with a 2.4 GHz (ISM) of frequency, a 50-150m of range and a debit of 1Mbps (Smart/BLE).	Non
Wi-Fi Direct	Offers P2P Wireless without Hot Spots; it's is Wi-Fi without the internet bit; it's a part of the standard Wi-Fi radio.. It knows what's nearby.	Non

Tableau V-3 Some connection protocols

Protocol	Description
User Datagram Protocol (UDP)	A transport layer protocol defined for use with the IP. It provides an unreliable service that offers no guarantee of delivery or protection against duplication.
Constrained Application Protocol (CoAP)	A specialized web transfer protocol for use with constrained nodes and constrained networks in the Internet of Things. It is designed for machine-to-machine (M2M).
Message Queuing Telemetry Transport (MQTT)	A standard for IOT communication that saves battery power and gives a good QoS. It supports several millions of connection and the creation of a broker cluster.
Transmission Control Protocol (TCP)	A connection-oriented protocol that provides a reliable transport service between pairs of processes using the network layer service provided by IP.

In the following, we present two tables (Tableau V-1, Tableau V-2 and Tableau V-3) of some communication protocols and technologies.

V.5. COMPUTING ARCHITECTURES AND TECHNOLOGIES SUPPORTING IOT

IoT is defined as interconnected and uniquely addressable objects that are managed via web pages [5; 6; 16]. These objects are supported via distributed systems within the collaborative mobile computing using a variety of models and architectures, including: cloud, fog, edge, mobile edge computing cloudlet, mobile cloudlet, multi-groups of smart devices, etc.

In this context, Hung says that: “When considering the IoT, we think of machines, vehicles, buildings and consumer goods, all connected to the Cloud” [4]. As well, in this same article, He asks: “Yet, will this cloud approach be the predominant architecture for the IoT? Will we add sensors and actuators to monitor and control things, and then rely on the cloud to provide computing resources and storage? Not always.”

From what has been published about them in the literature, there is no need to redefine and interpreted in detail the architectures of Cloud and its derivatives or branches as: Fog, edge, mobile edge computing, cloudlet, mobile cloudlet, etc. Instead, we must document and explain the evolution of these technologies and the reasons for their successive emergence in time, by addressing previously unmet needs. This is what we tried to illustrate summarily in the following figures: Figure V-3, Figure V-4, Figure V-5 and Figure V-6.

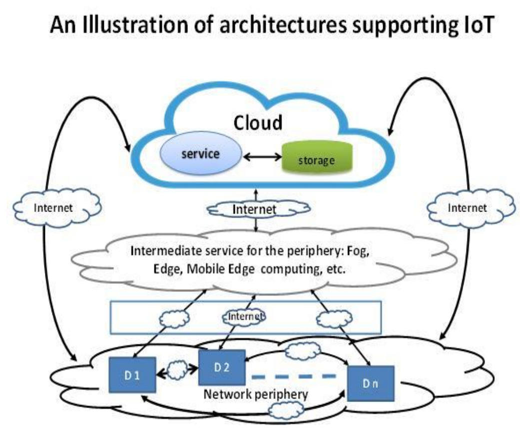


Figure V-3 Architecture supporting IoT. © 2021 Ghassan Fadlallah.

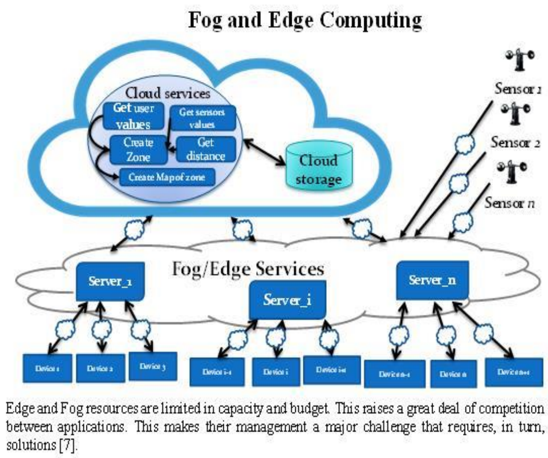


Figure V-4 Intermediate services: Fog & Edge Computing. © 2021 Ghassan Fadlallah.

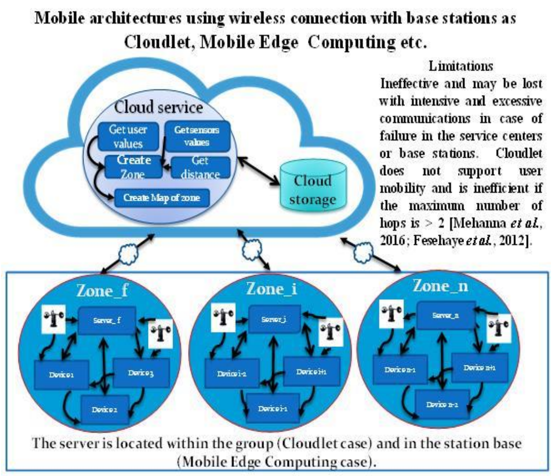


Figure V-5 Mobile architectures based on base stations. © 2021 Ghassan Fadlallah.

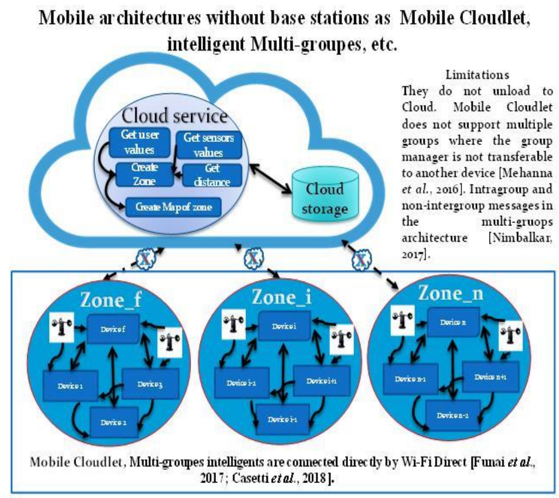


Figure V-6 Architecture with direct P2P connection. © 2021 Ghassan Fadlallah.

The evolution of these technologies and the reasons for their emergence are not the only thing shown in these figures. They also shed light and draw attention to the problems that these technologies have and are still experiencing during their development stages. These problems become more complex as we go towards the periphery of communication networks, since they constitute these technologies' infrastructure.

V.5.1) ARCHITECTURES: LIMITS AND CHALLENGES

The peripheral networks suffer on several levels, as doing other networks located closer to the service centers. Basically, the communication with these centers is according to the number of base stations or access points to cross. Especially, since this issue has the greatest impact on most other problems of these networks such as low capacity in response time power, computing and storage. Therefore, the establishment of new architectures is required to bring services closer to the peripherals, to reduce access to the main cloud center. Thus, emerging architectures are created such as fog, edge, mobile edge, mobile cloud, cloudlet, etc. These architectures, which use communication technologies based on switches, routers, gateways, and base stations, have contributed to strengthen the collaborative mobile computing and IoT by reducing latency to Cloud if communication is maintained. However, they still have limitations, among others, the lack of tasks unloading to central or peripheral servers [9], the loss of user mobility in case of Cloudlet, the big number of hops (passing packets between devices, when number is greater than 2) [8], the limitation of centers resources in capacity and budget which raises a great competition between applications [7] and the dysfunction of Wi-Fi. So, what happens if the connection fails when the base stations are out of service due to a disaster or do not exist as in rural areas? This question is illustrated in the following Figure V-7.

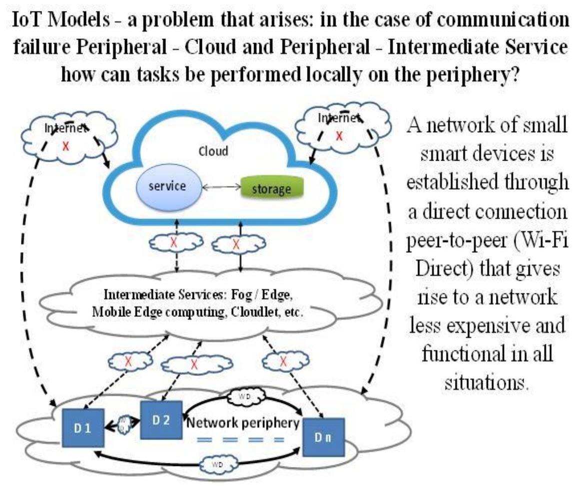


Figure V-7 Central and Intermediate Servers Connection Failure. © 2021 Ghassan Fadlallah.

In response to this question, some approaches (see figure), as mobile cloudlet and intelligent multi-groups architectures, are taking advantage of emerging direct communication technologies such as Wi-Fi Direct to bridge the gap in the collaborative mobile computing paradigm caused by the failure or interruption of the wireless connection based on access points or base stations. Unfortunately, these new architectures based on the principle of smart devices groups are having more problems and limitations. For example, Intelligent Multi-Groups

Architecture does not have intergroup communication while mobile cloudlet supports two groups only. They do not unload to Cloud. Their group owners are non-replaceable [10, 6]. Their resources may be limited in computation, storage and energy more than previous architectures capacity. Managing these limited capacity resources is a critical challenge requiring considerable effort in tasks scheduling and load balancing at the applications level.

V.6. OUR APPROACH: COLLABORATIVE AUTONOMOUS NETWORKS FRAMEWORK

In this context, we propose an approach (see Figure V-8) that should perform the needed tasks in many situations, such as the case of partial or total failures of networks and systems. These failures should be handled when extreme events can happen (e.g.: natural disasters, extreme weather, vegetation in rural areas, etc.). This approach addresses the problems of the partial or total loss of the connection (in the event of disasters, Siberian attack, rural areas or failures in the base stations), as will its limitation or the insufficiency of its bandwidth. In addition, we should consider the poor extent of the mobile network in the case of using a peer-to-peer connection. As well as the lacuna in performing relatively large size tasks using small smart devices. This is partly due to partitioning, scheduling, and distributing tasks. These problems were at the origin of our project: Collaborative model or framework of an autonomous network of mobile devices.

This research project consists of establishing an autonomous mobile network of smart devices connected directly by their own wireless capabilities using Wi-Fi Direct, LoraWAN or other similar technologies if available. To this end, the research project, which benefits from the advantages of existing architectures and technologies in this area, aims to overcome their inherent difficulties mentioned above (in particular, failures of systems networks). Its specific objectives are:

1. Maintaining device connectivity in any case and time.
2. Making objects (devices) more autonomous.
3. Compensating for upload to cloud and other middle architectures.

4. Maintaining the capacity of the devices as energy, calculation and storage, etc.

To achieve these objectives, our approach will be done on two levels of resources:

1. Hardware: Infrastructure and Devices Architecture.
2. Software: applications (formation of groups, election of manager, …) and scheduling (including partitioning and distribution) of tasks.

Networks, to be built at the hardware level, consist of a variety of devices: smartphones, tablets, sensors and laptops of small size and large capacity. These devices are grouped, according to their ranges, in the form of chain of groups on several zones, taking advantage of the available modes of connections, more specifically Wi-Fi Direct, offered for free.

Group building would follow this plan:

1. The evacuation plan is supposed to be pre-established and ready for execution. It is therefore possible to initiate local device groups via the Cloud service.
2. Local exploration by each device or other existing peers in its scope using Wi-fi-Direct.

Communication, information exchange, negotiation between devices and convention to

3. form groups, and to elect their managers according to the context and their capacities to carry out (perform) the targeted jobs.

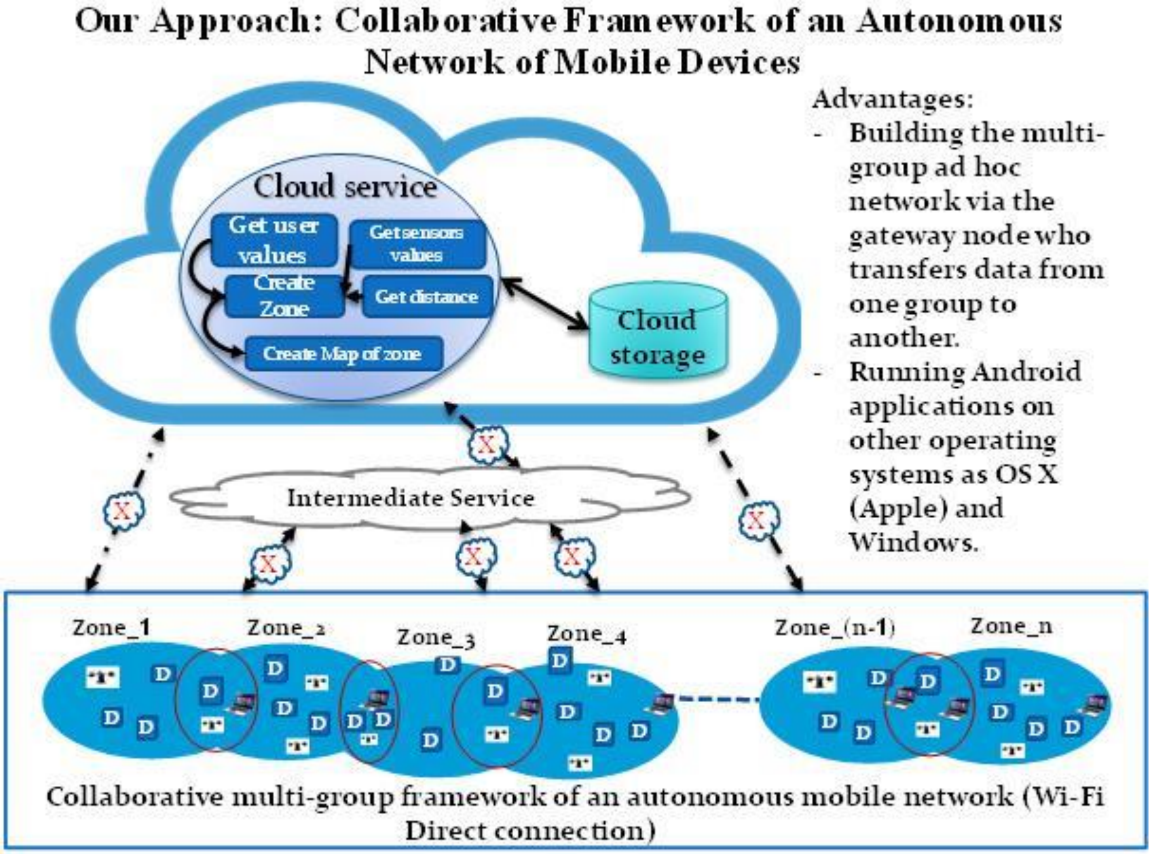


Figure V-8 Our Network Model using P2P connection. © 2021 Ghassan Fadlallah.

4. Periodic collection of information by current group managers and selection of future managers based on the best total score on energy, storage and calculation factors. The calculation of the score is done by an allocation, already made based on the literature, of a constant value to each of these factors according to its importance for devices effectiveness. The score of each device is then calculated as the sum of the multiplications of each of these values by the current capacity of the corresponding factor.

The last three steps are repeated periodically. The period is to be determined according to a given situation (in seconds, minutes, etc.).

For the scheduling of tasks plan, it is given as follows:

1. Partitioning the work to be done into tasks that adapt to devices capabilities.
2. Collecting information on these tasks: processing time, expiry date, start date, etc.
3. Establishing precedence relationships between these tasks.
4. Assigning existing tasks to available devices:
5. If there are tasks in the Waiting group owner (GO) status:
   a. Select pairs of task and least loaded device that its capacity is proportional to the size of this task (inspired by LCFP, SCFP and MCT algorithms).
   b. This task must be performed entirely without interruption on this device.
   c. Update status of tasks: Waiting, processing, or completed.

V.7. CONCLUSION

Resilient communication in system networks is still an issue in many cases, such as extreme events (e.g. natural disasters, terrorist attacks, etc.). We conclude that peer-to-peer direct communication technologies between smart devices can fill the gap caused by station-based wireless connection failure or insufficient bandwidth or failure networks. Therefore, it is necessary to develop and strengthen these technologies in terms of the scope of broadcasting and throughput. It is expected that the WIFI-Direct and LoraWAN technologies, coupled with the accelerated evolution of smart devices, can give rise to new and effective technologies or architectures that will work in any situation and at any time, especially during extreme events such as disasters. Therefore, a network of small smart devices established by a direct connection (Wi-Fi Direct) peer-to-peer giving rise to an inexpensive network can be functional in all situations.

In this paper, we proposed a mobile collaborative framework using peer-to-peer systems. An algorithm to build a group was given and a scheduling plan was described to perform the task based on the capacity of device and the size of the task (inspired by LCFP, SCFP and MCT algorithms). This framework was used to execute tasks in clients' sides and helps to collaborate mobile devices in many failure situations in the system networks, such as extreme events. In the future works, we will describe the details of this framework and validate and improve these algorithms.

ACKNOWLEDGMENTS

This work was sponsored and funded by Les Fonds de l'Université du Québec à Chicoutimi (FUQAC), Chicoutimi (Quebec), Canada.

REFERENCES

[1] J. Abdelaziz, H. Mcheick, "Toward a collaborative computing environment", *Conférence francophone de systèmes collaboratifs (SysCo)*, Sousse, Tunis, 28-30 Sept., 2012.

[2] D. Dolev, M. K. Warmuth, "Scheduling precedence graphs of bounded height," *Journal of algorithms*, vol. 5; no. 1, pages: 12, 1984.

[3] G. Fadlallah, Dj. Rebaine, H. Mcheick, "Scheduling problems from workshop to collaborative mobile computing: A state of the art," *International journal of computer science and information security (IJCSIS)*, vol. 16, no. 1, Jan. 2018.

[4] M. Hung, "Leading the IoT-gartner insights on how to lead in a connected world," 2017, [Online]. Available: https://www.gartner.com/imagesrv/books/iot/iotEbook_digital.pdf.

[5] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context aware computing for the Internet of Things: A survey," *IEEE communications surveys & tutorials*, vol. 16, no 1, pp. 414-454, 2013.

[6] C. Funai, C. Tapparello, W. Heinzelman, "Enabling multi-hop ad hoc networks through Wi-Fi Direct multi-group networking," in *Proceedings of the international conference on computing, networking and communications*, Santa Clara, CA, USA, pp. 491-497, 26-29 Jan. 2017.

[7] C. H. Hong, B. Varghese, "Resource management in fog/edge computing: a survey on architectures, infrastructure, and algorithms," *ACM computing surveys (CSUR)*, vol. 52, no 5, pp. 1-37, 2019.

[8] D. Fesehaye, Y. Gao, K. Nahrstedt, G. Wang, "Impact of cloudlets on interactive mobile cloud applications," In: *IEEE 16th international enterprise distributed object computing conference (EDOC)*. IEEE; Beijing, China, pp. 123-32, 2012.

[9] X. Wu, R. Dunne, Q. Zhang, W. Shi, "Edge computing enabled smart firefighting: opportunities and challenges," *In Proceedings of HotWeb'17*, San Jose / Silicon Valley, CA, USA, 6 pages, Oct. 14 2017.

[10] A. A. Mehanna, M.I A. Abdel-Fattah, S. Abdel-Gaber, M.Cloudlet: "A mobile cloudlet model using Wi-Fi Direct," *International journal of computer science and information security (IJCSIS)*, vol. 14, no. 11, Nov. 2016.

[11] "Number of mobile phone users worldwide from 2015 to 2020," Statista research department, Nov 23, 2016. [Online]. Available: https://www.statista.com/statistics/274774/forecast-of-mobile-phone-users-worldwide/

[12] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, M. Ayyash, "Internet of things: A survey on enabling technologies, protocols and applications," in *IEEE Communications surveys tutorials*, vol. 17, no. 4, pp. 2347-2376, 2015.

[13] A. Stanford-Clark, H.L. Truong, "MQTT for sensor networks (MQTT-SN) protocol specification", *International business machines (IBM) Corporation version*, vol. 1, no 2, 2013.

[14] T. Salman, R. Jain, “A survey of protocols and standards for Internet of Things,” *Advanced computing and communications*, vol. 1, no. 1, Mar. 2017

[15] H. S. Narman, M. S. Hossain, M. Atiquzzaman, H. Shen, “Scheduling internet of things applications in cloud computing,” *Annals of telecommunications*, vol. 72, no 1-2, pp. 79-93, 2017.

[16] A. Coulon, “L'Internet des Objets - Un gisement à exploiter,” La Lettre d’ADELI n°78, Hiver 2010.

CHAPITRE VI. UN MODÈLE ARCHITECTURAL COLLABORATIF OMNIPRÉSENT

Ghassan Fadlallah, Hamid Mcheick * and Djamal Rebaine
Department of Computer Science and Mathematics, University of Quebec at Chicoutimi,
Chicoutimi, QC, G7H 2B1, Canada
Ghassan-Mustapha.Fadlallah1@uqac.ca (G.F.); Hamid_Mcheick@uqac.ca (H.M.); Djamal_Rebaine@uqac.ca (D.R.)
* Correspondent: Hamid_Mcheick@uqac.ca

Résumé : L'informatique collaborative omniprésente au sein de l'Internet des objets (IoT) a progressé rapidement au cours de la dernière décennie. Néanmoins, les modèles architecturaux émergents et leurs applications et dispositifs souffrent encore d'une capacité limitée dans des domaines tels que la puissance, l'efficacité informatique, la mémoire, la connectivité, la latence et la bande passante. Le développement technologique se poursuit dans les secteurs du matériel, des logiciels et des communications sans fil. Leur communication se fait généralement via Internet et sans fil via des stations de base. Cependant, ces modèles sont parfois sujets à des pannes de connectivité et à une couverture limitée. Les modèles qui intègrent des appareils dotés de technologies de communication pair-à-pair (P2P) sont d'une grande importance, en particulier dans les environnements difficiles. Cependant, leurs dispositifs à puissance limitée sont répartis de manière aléatoire à la périphérie où leur disponibilité peut être limitée et arbitraire. Malgré ces limitations, leurs capacités et leur efficacité sont en constante croissance. L'accélération du développement dans ces domaines peut être obtenue en améliorant les architectures et les technologies de l'informatique collaborative omniprésente, qui fait référence à la collaboration de dispositifs informatiques mobiles et embarqués. Pour améliorer l'informatique collaborative mobile, en particulier dans les modèles agissant à la périphérie du réseau, nous nous intéressons à la modernisation et au renforcement de la connectivité à l'aide des technologies sans fil et de la communication P2P. Par conséquent, l'objectif principal de ce document est d'améliorer et de maintenir la connectivité et de renforcer les performances de ces systèmes omniprésents tout en fournissant les services requis et attendus dans un environnement sévère. Ceci est particulièrement important dans les situations catastrophiques et les environnements difficiles, où la connectivité est utilisée pour faciliter et améliorer les opérations de sauvetage. Ainsi,

nous avons établi un modèle architectural collaboratif mobile résilient comprenant un réseau périphérique autonome de dispositifs omniprésents en tenant compte des contraintes de ces ressources. En maintenant la connectivité de ses appareils, ce modèle peut fonctionner indépendamment des stations de base sans fil en tirant parti des technologies de connexion P2P émergentes telles que Wi-Fi Direct et celles activées par LoPy4 de Pycom telles que LoRa, BLE, Sigfox, Wi-Fi, Radio Wi-Fi et Bluetooth. De même, nous avons conçu quatre algorithmes pour construire un groupe d'appareils, calculer leurs scores, sélectionner un gestionnaire de groupe et échanger des messages inter et intra-groupe. L'étude expérimentale que nous avons menée montre que ce modèle continue de fonctionner efficacement, même dans des circonstances telles que la panne de la connectivité sans fil due à un événement extrême ou à une congestion due à la connexion d'un grand nombre de dispositifs.

Contribution associée :

Ghassan Fadlallah, Hamid Mcheick * and Djamal Rebaine, « A pervasive collaborative architectural model at the network's periphery », Internet of Things (IoT) journal, special issue: Emerging Trends and Challenges in Fog and Edge Computing for the Internet of Things, Editors: Benoit Parrein & Bastien Confais, vol.2, no. 3, pp. 524-548, September 6, 2021. https://doi.org/10.3390/iot2030027.

Préambule

Cet article s'intéresse à la façon de maintenir les connexions dans les modèles architecturaux émergents dans le paradigme de l'Internet des objets (IoT) et entre leurs applications et leurs dispositifs mobiles et omniprésents. Ces dispositifs, répartis de manière aléatoire à la périphérie des réseaux, souffrent d'une capacité limitée dans des domaines tels que la puissance, la mémoire, la connectivité, la latence et la bande passante. Au sein de ces modèles architecturaux c'est l'informatique collaborative omniprésente qui a été progressée rapidement au cours de la dernière décennie. Cependant, il souffre toujours d'une capacité limitée dans des domaines tels que l'efficacité informatique, la connectivité, le temps de réponse et la bande passante. Le développement technologique se poursuit dans l'industrie matériel, des logiciels et des communications sans fil. Ils communiquent généralement en ligne et sans fil via des stations de base. Cependant, ces modèles sont parfois sujets à des pannes de connexion et à une couverture limitée. Les modèles qui incluent des appareils dotés de technologies de communication pair-à-pair sont d'une grande importance, en particulier dans les environnements difficiles.

Dans cet article, nous avons intensifié nos efforts pour améliorer et maintenir la connectivité et renforcer les performances de ces systèmes omniprésents tout en fournissant les services requis et attendus dans des environnements sévères. Ainsi, nous avons établi un modèle architectural collaboratif mobile résilient comprenant un réseau périphérique autonome de dispositifs omniprésents en tenant compte des contraintes de ces ressources.

En maintenant la connectivité de ses appareils, ce modèle peut fonctionner indépendamment des stations de base sans fil en tirant parti des technologies de connexion pair-à-pair telles que Wi-Fi Direct et celles activées par LoPy4 de Pycom telles que LoRa, BLE, Sigfox, Wi-Fi, Radio Wi-Fi et Bluetooth.

Afin de rendre notre modèle architectural collaboratif fonctionnel et complet, nous avons conçu quatre algorithmes pour construire un groupe d'appareils, calculer leurs scores, sélectionner un gestionnaire de groupe et échanger des messages inter et intra-groupe.

Notre étude expérimentale montre que ce modèle continue de fonctionner efficacement, même dans des événements extrêmes souffrant des problèmes sérieux de la connectivité. Le processus de création et d'activation de notre modèle a été détaillé dans les six sections suivantes: l'Introduction, les travaux connexes, le problème des modèles architecturaux omniprésents à la périphérie des réseaux, le modèle architectural collaboratif complet proposé, notre étude pilote, et enfin la conclusion.

A PERVASIVE COLLABORATIVE ARCHITECTURAL MODEL AT THE NETWORK'S PERIPHERY

Ghassan Fadlallah, Hamid Mcheick * and Djamal Rebaine
Department of Computer Science and Mathematics, University of Quebec at Chicoutimi, Chicoutimi, QC, G7H 2B1, Canada.
Ghassan-Mustapha.Fadlallah1@uqac.ca (G.F.); Hamid_Mcheick@uqac.ca (H.M.); Djamal_Rebaine@uqac.ca (D.R.)
* Correspondent Hamid_Mcheick@uqac.ca

Abstract: Pervasive collaborative computing within the Internet of Things (IoT) has progressed rapidly over the last decade. Nevertheless, emerging architectural models and their applications still suffer from limited capacity in areas like power, efficient computing, memory, connectivity, latency and bandwidth. Technological development is still in progress in the fields of hardware, software and wireless communications. Their communication is usually done via the Internet and wireless via base stations. However, these models are sometimes subject to connectivity failures and limited coverage. The models that incorporate devices with peer-to-peer (P2P) communication technologies are of great importance, especially in harsh environments. However, their power-limited devices are randomly distributed on the periphery where their availability can be limited and arbitrary. Despite these limitations, their capabilities and efficiency are constantly increasing. Accelerating development in these areas can be achieved by improving architectures and technologies of pervasive collaborative computing, which refers to the collaboration of mobile and embedded computing devices. To enhance mobile collaborative computing, especially in the models acting at the network's periphery, we are interested in modernizing and strengthening connectivity using wireless

Citation: Fadlallah, G.; Mcheick, H; Rebaine, D. A collaborative pervasive architectural model at the network's periphery. *IoT* **2021**, *2*, x. https://doi.org/10.3390/xxxxx

Academic Editor:

Received: date
Accepted: date
Published: date

technologies and P2P communication. Therefore, the main goal of this paper is to enhance and maintain connectivity and improve the performance of these pervasive systems while performing the required and expected services in a challenging environment. This is especially important in catastrophic situations and harsh environments, where connectivity is used to facilitate and enhance rescue operations. Thus, we have established a resilient mobile collaborative architectural model comprising a peripheral autonomous network of pervasive devices that considers the constraints of these resources. By maintaining the connectivity of its devices, this model can operate independently of wireless base stations by taking advantage of emerging P2P connection technologies such as Wi-Fi Direct and those enabled by LoPy4 from Pycom such as LoRa, BLE, Sigfox, Wi-Fi, Radio Wi-Fi and Bluetooth. Likewise, we have designed four algorithms to construct a group of devices, calculate their scores, select a group manager, and exchange inter and intra-group messages. The experimental study we conducted shows that this model continues to perform efficiently, even in circumstances like the breakdown of wireless connectivity due to an extreme event or congestion from connecting a huge number of devices.

Keywords: Pervasive architectural models; collaborative computing in IoT; connectivity of IoT objects; harsh environments; construct group of nodes; resilient and rescue systems; pervasive computing.

VI.1. INTRODUCTION

Continuous and rapid technological innovation, especially in the areas of telecommunications and information technology, has led to a radical expansion of the Internet throughout the world. This has given rise to the Internet of Things (IoT). Moreover, portable smart devices that use the Internet and various wireless technologies have become popular. The facilities in cities, including buildings, offices, factories, markets, home appliances, machines, security systems and cars are connected to each other. This is what Ashton [1] first called the IoT in 1999. IoT supports a wide variety of networks interconnected through a variety of devices, including servers, databases, computers, powerful portable and wearable devices like smartphones and personal digital assistants.

The improvement of the Internet has accelerated the development of practical solutions to the various problems of distributed systems, such as heterogeneity, openness, connectivity, security, scalability, failures, transparency and quality of service [2]. Advances in technology like device miniaturization and both P2P and wireless communication have enabled the integration of pervasive devices into distributed systems. These enhance and expand collaborative mobile computing by using small low-cost devices (wearable/portable) present in the physical environments of users, including homes, offices, and even natural environments.

This model of computing allows the interaction of remote and mobile users to perform different activities using a variety of pervasive devices. This interaction has become an effective technique for dealing with complex problems by breaking them down into smaller sub-problems processed adequately on large networks. The pervasive devices develop very quickly and their functions are enriched and enhanced while their capabilities of communication are increased. In this paradigm, all users, hardware and software are dynamic, and they change these capabilities in unpredictable ways [2, 3].

The development of collaborative systems has led to innovative designs for mobile networks and their devices' technologies. In addition, they enhance mobile collaborative computing and the techniques of interaction between these devices.

We distinguish many architectures that have been established recently in pervasive collaborative computing on distributed mobile networks, such as cloud, fog and edge computing [4].

Cloud computing, the main architecture supporting the IoT, has successfully helped to solve the problems of collaborative computing in areas such as communication, computing capabilities and storage and transfer of data. Similarly, fog, edge and mobile edge computing (MEC) [5] solved the problems faced by cloud computing when offloading to central servers and vice versa. Fog and edge computing architectures [6] (and recently MEC), with several other architectural models, have gained primordial importance in mobile networks.

However, the effectiveness of devices in these structures became disproportionate to the intense demand for communications. This effectiveness is needed even more in cases like natural disasters, industrial accidents and terrorist attacks, which can isolate or destroy their communication service centers, base stations and access points. Therefore, the solution for situations of disconnected communication is most likely realized using networks that comprise mobile devices and have ubiquitous resources that can be connected through their capabilities of P2P connection. These devices can be regrouped instantly in networks in real time and in a variety of positions. Besides extreme events, these networks could be used in normal situations due to their very low cost and their permanent presence in our pockets. These networks provide users with facilities and benefits to manage their work anytime and anywhere. For these reasons, and with the emergence of new technologies expected in the future, we claim that these devices and networks will be the de facto tools to support the paradigm of smart mobile collaborative computing.

The pace of technological development constantly imposes real challenges in a variety of areas, particularly in mobile networks. This is due to the rapid advances in the architecture and techniques of IT models, mobile devices and communication tools, such as wireless and P2P, through self-diffusion. P2P communication allows access to devices without the need for wireless infrastructure such as access points or evolved base stations (eNBs). P2P can use various short-range wireless technologies [7] like Bluetooth, Wi-Fi Direct, LTE Direct (defined by the Third Generation Partnership Project (3GPP) [8]), near field communication (NFC) and proximity services as needed. The mobile networks can be composed of devices that connect spontaneously using these types of mobile communication, and this offers performance services in a variety of areas [9]. In addition, emerging Radio Wi-Fi technology enabled by Pycom's LoPy4 can extend the range of P2P and wireless communication. As a result, networking has become a resource of pervasive devices that can be connected anytime and anywhere. With key technologies such as spatial modulation, millimeter wave, visible light communication and 5G, cellular networks will soon enable people to use P2P technology globally. The objective is to increase the number of connected devices with longer battery life and greatly reduced latency [10].

The IoT paradigm is defined as a dynamic global self-configuration and interoperable network [11,12]. It fascinates and even seduces its users. Because dynamic situations give these users more freedom in terms of place of work and time to continue their different jobs, the demand for the use of these devices or objects is steadily increasing. Likewise, the development of pervasive devices is leading to an intensification of their use in increasingly complex collaborative applications, as well as to an intensive use of resources. [2,13,14]. Unfortunately, technological progress does not follow this demand proportionately. For example, the number of extant mobile devices, including phones and tablets, has increased from more than 7.7 billion in 2014 to more than 12.1 billion in 2018 [15]. Moreover, some expect that, by 2025, there will be up to 74 billion connected IoT devices [16] or perhaps as many as 100 billion [17]. Therefore, since the technological advancement of these devices lags behind the increasing demand for their use, it can be compensated by improving the performance of the applications that these devices will achieve in terms of the speed of response and the completion of tasks. This can be done using job scheduling and load balancing techniques and by maintaining connectivity in networks, especially in harsh environments, to ensure the continuity of activities for the required time.

Therefore, this paper seizes the opportunity of emerging architectures such as edge, fog, cloud, mobile ad hoc cloud computing and multi-group networking to build an autonomous architecture of pervasive, collaborative computing in the IoT paradigm. The main idea of the approach is to establish a peripheral autonomous network of mobile devices using their own P2P radio communication and the radio wireless enabled by Pycom’s LoPy4. Therefore, the present work aims to:

1. Reduce the risk of total or partial interruption of network connectivity caused by breakdowns or overwhelming (a huge number of) connections. This involves the improvement and

continuity of maintaining the connectivity between mobile devices, regardless of the state of the network.

2. Increase the resilience and reliability of sharing data, information and computations between these devices (objects).

To achieve these goals, we have designed a pervasive mobile network based on a collaborative layered architectural model [18] that allows users to communicate in various situations. For this purpose, we have taken into account the capacity limitations of their devices in terms of data storage, computing, and expected battery life. This model has been designed to enhance computing capacity, improve response time and increase the use of ubiquitous surrounding resources. The development of this architectural model includes the following steps:

(a) Design an architecture sample to maintain connectivity, even during extreme events and harsh environments.

(b) Establish a multi-group network connected by device-to-device communication, such as Wi-Fi Direct and other communication technologies that have a larger range connection, such as those of Pycom's LoPy4.

(c) Propose four new algorithms and validate them through the required applications needed to manage the proposed model, such as dividing tasks and balancing their load. Among these applications is one to program the Lopys as transmitter, bridge and receiver.

Pervasive computing is the natural successor of mobile computing systems. It is an emerging field of research that offers revolutionary models for the most recent computing paradigms. Pervasive computing has led to tremendous advances in technologies such as computing and mobile devices, as well as technologies of communication, wireless networking, middleware programing, smart spaces and distributed and embedded systems.

The present study fosters collaboration between groups of objects through intra-group and inter-group (of devices) communication in the IoT model, where most of these objects are no longer managed by a system administrator. Note that collaborative computing can be defined as “a fertile blend of technologies and techniques which facilitate people working together via computer-assisted means” [19]. Pervasive computing is an emerging field of distributed systems, where objects are small, mobile and connected by wireless and radio P2P technologies. It can be divided into three sub-classes: systems of ubiquitous computing, mobile systems and sensor networks [20]. Our research combines collaborative and pervasive computing to (i) embrace a pervasive collaborative model for IoT and (ii) facilitate and assist cooperation between people, especially in harsh environments, to exploit the capabilities of each object (device). Indeed, we think the interesting applications come from the collaboration of objects with minimal human collective intervention rather than from individual ones.

Other technologies are used in this paper. Let us introduce them briefly. LoRa is a wireless technology that enables communications over long distances, but at a low data rate. It uses sensors and actuators for P2P and IoT applications. It can use the radio spectrum to enable low power broadband communication between remote devices and network gateways. LoRa is the de facto platform for low power wide area network (LPWAN) technology for IoT. The LoRaWAN® specification, designed by Pycom (Guildford, UK), is a networking protocol designed to enable rapid establishment of public or private IoT networks anywhere using bidirectional communication, mobility and localization services with end-to-end security [21]. Wi-Fi-direct is a Wi-Fi ad hoc standard for peer-to-peer wireless connections that allows two devices to communicate without an access point, router, or Internet connection.

The remainder of this paper is organized as follows. Section 2 describes related work, and discusses the problem of pervasive architectural models at the periphery. Section 3 describes the proposed pervasive collaborative architectural model. Section 4 presents the experimental study we conducted. Finally, concluding remarks are presented in Section 5.

VI.2. RELATED WORKS OF PERVASIVE ARCHITECTURAL MODELS AND THEIR LIMITS

This section surveys the pervasive architectural models and identifies their limits.

VI.2.1) PERVASIVE ARCHITECTURAL MODELS

Maintaining connectivity between devices in pervasive computing, and particularly in the IoT paradigm, is important for serving people in various situations, especially in rural and harsh environments. For example, connectivity must be maintained with a smart car while it is driving as well as during a live and continuous transfer of data if it is necessary to make quick decisions. In addition, maximizing coverage areas, especially in these environments, is needed to ensure communications between pairs of devices over at least one connection. Many approaches and algorithms are designed to increase connectivity and maximize coverage in networks. Most require a central station [22,23] with advanced resources to control the devices in the network, or they require homogenous devices [24,25]. Recently, Mcheick *et al.* [26] designed an algorithm CMHWN to provide connectivity between at least two devices and to add flexibility in controlling the overall displacement in the network. Ghaddar *et al.* [27] designed and validated a connectivity approach to minimize the movement of devices to save energy and support the self-organization of each object (device). Wang *et al.* proposed a communication system using only Wi-Fi Direct technology [28]. However, these algorithms and systems do not handle heterogeneous communication technologies and require a central station to maintain connectivity. Therefore, many existing technologies can be considered in contemporary architectures to maximize coverage and maintain connectivity in pervasive computing.

In addition, major research efforts have been conducted in the fields of cloud, fog, edge and mobile edge computing and their branching at the peripheral level. The fundamental structure in this area is cloud computing. The main problem of the cloud is that its services are centralized, which sometimes slows down processes and can cause total or partial shutdowns due to a huge number of connections. This problem is the basis for other approaches, which transform these centralized services into decentralized ones through intermediate and peripheral service centers. In this way, several architectural models were developed: fog, edge and mobile edge computing, cloudlet, and collaborative fog computing [29,30].

Table VI1 reviews the most recent and widely used of these architectural models.

Tableau VI-1 Classification of contemporary architectural models

Architectural Model	**Description and Analysis of These Models**
Mobile edge computing (MEC)	The main advantage of MEC is to deploy mini cloud servers in each communication base station, especially at the edge of mobile networks. MEC is applied to mobile devices in these edge networks to reduce delays by allowing computational and storage capabilities at the edge of mobile networks instead of being at central servers [5,31]. MEC has better discharge techniques to edge servers, giving the network low latency and high bandwidth [6]. In addition, it provides efficient storage and energy consumption through many architectures such as the software-defined system for mobile edge computing (SDMEC) for storage [32] and ME-VoLTE for computing, which integrates MEC with long-term evolution (LTE) [33].
Cloudlet	A cloudlet is a proximal cloud that brings computing capacities closer to the mobile devices it serves to avoid latency. A cloudlet is formed by a cluster of multi-core computers, a very powerful multi-core server and a high bandwidth wireless LAN with high connectivity [34,35]. “Cloudlets are decentralized and widely dispersed Internet infrastructure whose compute cycles and storage resources can be leveraged by nearby mobile computers [34].”

Collaborative fog computing (CoFog)	CoFog provides dynamic services based on predefined templates using formal mechanisms called operations. An operation represents a relationship between a collaboration request and the services that can fulfill that request. There are two types of operations: conservative, which retains the same type of data, and non-conservative, which creates a new type. CoFog relies on the ability to discover, recover and provide security services using role-based and attribute-access models [19].
Mobile cellular cloud architecture (MoCCA)	MoCCA [29,36] uses small scale micro-servers loaded on mobile phones to build its own mobile cloud architecture. Its resources are smartphones, base stations, base station controllers and mobile switching centers. Its main concerns are (i) connectivity, (ii) computational limitations, (iii) churn, because of user mobility and device volatility, (iv) lack of power; and (v) user incentives.
mClouds	With mClouds [29,37], mobile devices become essential components of cloud computing architectures. An mCloud uses specific resource-discovery procedures, incentive management and distributed processing to execute locally whenever possible.
MobiCloud	MobiCloud [29,38] is a trustworthy collaborative mobile cloud administration system that enables the efficient and collaborative management of mobile smart phone resources. Its objective is to use mobile devices even when there is no internet connectivity. MobiCloud architecture has participant nodes and a field control node (a cloud agent) that provides centralized cloud controller functionalities by locating the required resources.

Mobile cloudlet (M.Cloudlet)	M.Cloudlet uses its own servers without base station services to provide cloud services in order to dispense the connection to remote cloud central servers or central ones of fixed cloudlets. To build a network of mobile cloudlet, M.Cloudlet uses Wi-Fi Direct to connect nearby mobile devices [39]. The M.Cloudlet architecture has two main phases: (i) build M.Cloudlet by discovering and negotiating neighboring devices to form their groups and choose their owners, then exchange information, (ii) control and manage devices, including their tasks and spontaneous lists of participating devices [39].
Multi-group networking	This supports Wi-Fi Direct connections between multiple devices in groups, then interconnects different groups to create ad hoc multi-hop networks. However, the Wi-Fi Direct standard [40] defines only intra-group communications, where the group owner (GO) is central to all communications in its group. A Wi-Fi Direct device called a gateway node can operate as a member of more than one group at a time. In a proposed time-sharing mechanism, the gateway node can switch between two or more groups to implement multi-group communication scenarios using only Wi-Fi Direct [41].
Building smart multi-group networks by Wi-Fi Direct	Independent group formation aims to achieve the efficient use of devices while maintaining connectivity. The main obstacle to the widespread adoption of P2P communication is the lack of coordination of services. Efforts are focused on the potential of Wi-Fi Direct as a P2P communication technology in medium- and large-level frameworks using unrooted Android devices. It functions by (i) designing multi-groups interconnected logical topology by tunnelling the application layer and (ii) proposing a fully distributed intelligent group formation mechanism to construct networks, considering the group’s durability, the devices’ power consumption, the bandwidth average and network extent [42].

Mobile ad hoc cloud computing (MAC)	MAC computing "is still in its infancy" [29]. Several definitions of MAC have been proposed [43-45], such as "MAC enables the use of many nearby resource-rich mobile devices to provide computational services in the vicinity" [45]. MAC roots are in mobile cloud computing (MCC) but also in opportunistic computing [43] or distributed computing with the caveats of intermittent connectivity and delay tolerance. The ad hoc cloud features are developed by Kirby *et al.* [46] as self-administration in terms of flexibility, performance and balancing potentially contradictory policy goals. MAC challenges include (i) limited computing and energy resources of mobile devices and their ability to perceive context and location, (ii) network connectivity, (iii) security and (iv) partitioning and application loading. In addition, there are specific challenges, as shown in Figure VI-1, introduced in the MAC context as mobile devices are the only resources [29].

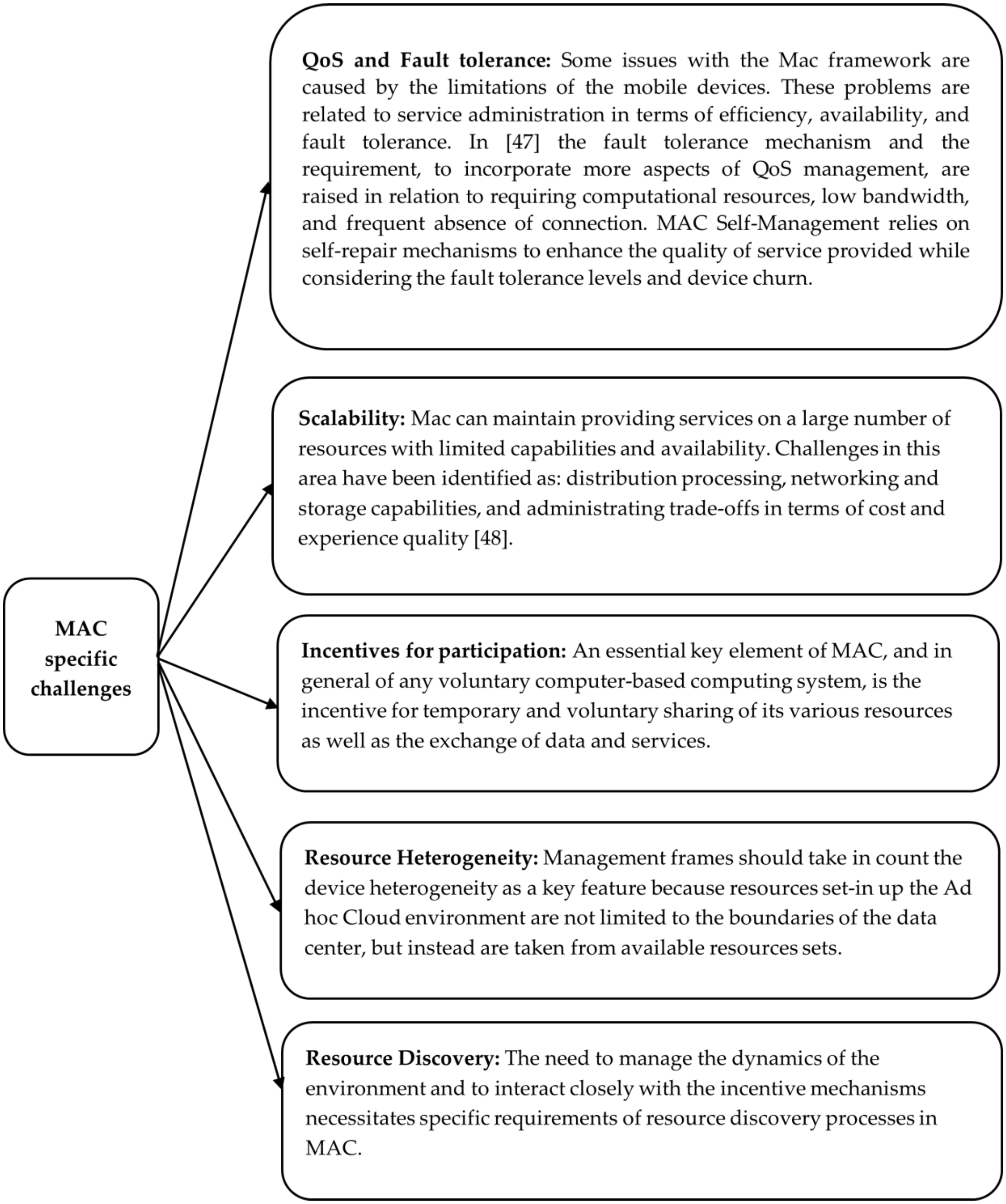


Figure VI-1 MAC specific challenges. © 2021 Ghassan Fadlallah.

VI.2.2) ISSUES OF PERVASIVE ARCHITECTURAL MODELS

The main problem of our research occurs from the emerging architecture field of distributed systems, especially pervasive collaborative computing at the periphery of a network. These architectures can be classified into three categories: (i) main centralized as cloud, (ii) peripherally centralized including fog, edge computing and mobile edge computing, and (iii) other peripheral architectures such as emerging technologies introduced in recent years, especially mobile ad hoc networks at the periphery (MANET, VANET, multi-groups networks, mobile cloudlet, etc.) [49]. The peripheral architectures introduced most recently make smart mobile systems more efficient in terms of communication, power consumption, data storage and computation capability while they reduce peripheral device discharge to cloud service centers. Note that cloud computing was created as a solution for mega data exchange and distributed collaborative and ubiquitous computing. Fog and edge computing were also designed as system architectures to address the problems of cloud computing near peripheral devices. Still other architectures have been designed as solutions for more distant peripheral devices including cloudlet, mobile cloud computing, mobile cloudlet, mobile ad hoc cloud computing, smart P2P and multi-group networks.

These peripheral architectures provide access to a large number of devices. The main issue is to maintain and enhance connectivity using the offered technologies, which include (i) P2P technology such as WI-FI-Direct, and LTE-Direct and (ii) wireless technology using base stations such as Wi-Fi and Lora.

The emergence of the Internet of Things (IoT) and its widespread deployment as a new model that processes a huge amount of data has led to problems with collaborative structures. These include many applications, such as management, communicating and exchanging data between remote users at the periphery of networks away from their hubs or centers.

Centralized cloud computing and other computing architectures at the levels closest to the periphery were, in their turn, effective solutions to these problems because of the proliferation of data centers and smart micro-devices that reach the network edges. However, the efficiency of these architectures has declined in proportion to the increase in requests from a huge number of users and the colossal amount of data to be processed. Therefore, it has caused a serious bottleneck in terms of connectivity and communication, especially for device users at the periphery. The networks in these paradigms generally comprise small mobile devices that suffer from the problem of limited capacity, as mentioned earlier.

Following a thorough study and analysis of these architectural models, we highlight the following facts about this area: (i) there is a strong trend towards the development and evolution of small, smart mobile devices; (ii) there is excessive demand to use them, and (iii) researchers have made diligent efforts to improve their performance and functionality and overcome their disadvantages. The main research questions to address on the limits of these structures are:

- How to maintain the connectivity of network devices in various situations?
- How to reduce the tasks of loading and unloading these small peripheral devices with central or intermediate service centers (cloud, fog, edge) to avoid the communication bottleneck?
- How to reduce the energy consumption of these devices by minimizing their time of use and improving task scheduling and load balancing algorithms?

We identified five emerging issues in this area of intelligent and collaborative mobile computing at the periphery of networks:

(1) Limited connection bandwidth because of congestion from loading and unloading messages (tasks and data) to fixed or mobile center servers [29].

(2) The limitations of mobile devices in terms of power, computing ability and data storage [39].

(3) The reduced range of mobile network devices in terms of device-to-device connection when wireless connectivity with base stations is lost.

(4) The complete loss of operability or effectiveness of these devices during severe events such as natural disasters and terrorist attacks because of malfunctions or failures in the base network stations.

(5) The gap in performing relatively large jobs using these devices. This is partly due to the inadequate partitioning and scheduling tasks relative to the capabilities and loads of the equipment.

VI.3. COLLABORATIVE ARCHITECTURAL MODEL OF MOBILE PERVASIVE COMPUTING

Pervasive computing aims to improve the human experience and quality of life without a clear awareness of the underlying communications and computing technologies [50]. This section is about the architectural model we proposed, the description of the structure of the connectivity and the algorithms to manage the construction and communication of the device groups. We also address briefly the security related to the access of the network.

VI.3.1) SPECIFIC OBJECTIVES

With the large number of mobile device technologies that contribute significantly to most sectors, there are still unmet needs to ensure the availability, permanence and continuity of wireless connectivity in several environments. The standard wireless connection technologies that use base stations and access points can be interrupted because of failures for a variety of reasons. These include a lack of power, natural disasters and terrorist attacks. Moreover, communication might not be available in certain overcrowded areas (for example, a concert or conference hall), or even to manage smart car communications. As a result, the trend is to address this deficiency by taking advantage of the emergence of device-to-device (P2P) communication technologies such as Bluetooth, Wi-Fi Direct and LTE-Direct. Based on these technologies, an autonomous local communication system can be developed to provide an instant communication environment for mobile users so they can communicate with each other [28]. Consequently, this type of radio communication technology has become the main concern in the recent development of wireless connections, particularly 5G [51].

The objective of the present study is to design an autonomous pervasive collaborative architectural model (APCAM) consisting of a network of ubiquitous smart mobile devices. In this model, data and information are exchanged between devices using device-to-device radio communication (such as Wi-Fi Direct) along with various available wireless communication technologies. This approach helps to strengthen the capabilities and then the performance of mobile peripheral networks by developing and improving their structure at both the hardware and software levels. On the one hand, this model relies on recent work in the field of collaborative mobile computing [36,41,42,52] and on the latest hardware communication technologies such as device-to-device radio communication and Pycom's LoPy4 that can be used as private wireless base stations. On the other hand, it relies on the latest load balancing and scheduling techniques, which must be adapted to its context, especially at the periphery of networks.

The specific objectives of our model are:

(1) Maintain the device's connectivity in harsh environments and at any time.

(2) Make objects (devices) and their networks more autonomous.

(3) Compensate for loading/unloading to/from the service centers of cloud and other edge (middle and peripheral) architectures.

(4) Maintain the capacity of the devices such as energy, computing ability and storage.

Therefore, for this paper, we considered these aspects in these two areas:

(a) Smart mobile network peripheral architectures in terms of technologies, protocols, and applications of networking. There is a variety of devices such as sensors, smartphones, and tablets, in addition to small laptops, that have a capacity to be used as local micro-servers. These devices can be grouped according to their ranges. In this way, a chain of groups is established in several zones, taking advantage of the available connection modes such as wireless, device-to-device and Pycom's LoPy4 technologies.

(b) Software in the form of algorithms and applications to carry out tasks such as the creation of groups, the election of a group manager, and exchanging data and messages between nodes, as well as partitioning, scheduling and load balancing of tasks. These algorithms and applications are tested and validated by simulating harsh situations, for example, by sending messages from one device to another far away by combining the P2P connection technologies of devices and by establishing a proper mode of wireless technology based on communication technologies offered by Pycom's LoPy4.

VI.3.2) AUTONOMOUS PERVASIVE COLLABORATIVE ARCHITECTURE MODEL

We built this model based on the aforementioned technologies for connecting smart mobile devices. It consists of reassembling these devices in groups connected through the communication technologies available according to their ranges in their zones. Among the most important of these technologies is the Wi-Fi Direct, which uses the gateway node as a means of communicating between these groups (please see details in Figure VI-2 of the reference [18]). These devices can operate independently of base stations through their own wireless radio technology by integrating with the access points provided by Pycom's LoPy4. Thus, this model can play a crucial role in the case of losing communication system infrastructure as base stations. This is especially important in harsh situations, when an emergency event occurs, where for example smart car communications and natural disasters must be managed. In such cases, the primary concern is maintaining connectivity.

The architectural model we are proposing has the following layers (see details in the Fig. 3 of the reference [18]): (i) task application; (ii) service engineering, including two fields: computing modules and operations; (iii) data engineering, consisting of formatting and unification; (iv) adaptation; (v) connectivity, which includes network infrastructure and communication; and (vi) physical [18]. This architecture uses layer pattern architecture to separate the services into different modules. A layer offers a set of cohesive services (modules) exposed into public interfaces.

This APCAM model uses the following integrated wireless technologies that include Wi-Fi in conjunction with a device-to-device connection, such as Wi-Fi Direct, and it is enhanced with wireless connectivity provided by Pycom's LoPy4. LoPy4 is a powerful CPU, Bluetooth low energy (BLE) and state-of-the-art Wi-Fi radio with a range of 1 km [53]. Therefore, these are used by our model as special base stations, especially in extreme situations where Wi-Fi wireless technologies are lost. Our research aims to overcome the difficulties associated with creating an efficient mobile network system by direct radio communication (P2P) between devices that can be clustered into a chain of groups as shown in Figure VI-2. This network can be created automatically and spontaneously from distributed groups of smart devices by adopting an assembly mechanism based on the extended coverage of connections. This mechanism also allows devices to define their role so they can create an efficient logical topology for intergroup communications [42]. It consists of identifying smart proximity zones according to factors such as the context of the problem, the distance between devices, communication range, network traffic and resources capacities. This makes the model promising as it can be used for effective real-time communication between groups.

The APCAM model combines communication technologies to ensure the continuity of communications in various situations. In this way, by using Pycom's LoPy4, we established our model of communication by integrating several modern wireless and P2P radio technologies, including Wi-Fi, Wi-Fi Direct, Bluetooth and Lora. Our technique manages the communication between devices, especially those deployed in areas outside the coverage of Wi-Fi, in terms of the available alternative communication technologies.

The network of the APCAM model can be classified into two categories according to their ability to communicate with LoPy4. Therefore, our strategy is based on devices that are not in the communication coverage range of LoPy4 and that can be connected through device-to-device radio communication technology (Figure VI-2). We have also adopted another strategy for devices in the communication coverage range of LoPy4 to integrate device-to-device radio communication and Radio Wi-Fi technology offered by LoPy4 (Figure VI-3). Both strategies rely on clustering devices into groups, each of which chooses its own manager or group owner to communicate with each other. These strategies include designing appropriate algorithms and applications to operate effectively to manage groups of devices and their intra- and interconnections, as well as sharing their information. It is a collaborative distributed environment that enables the proper functioning of the network, considering the availability of the participating devices, their energy and computing capabilities, the average speed of data transfers and the connection coverage area [42].

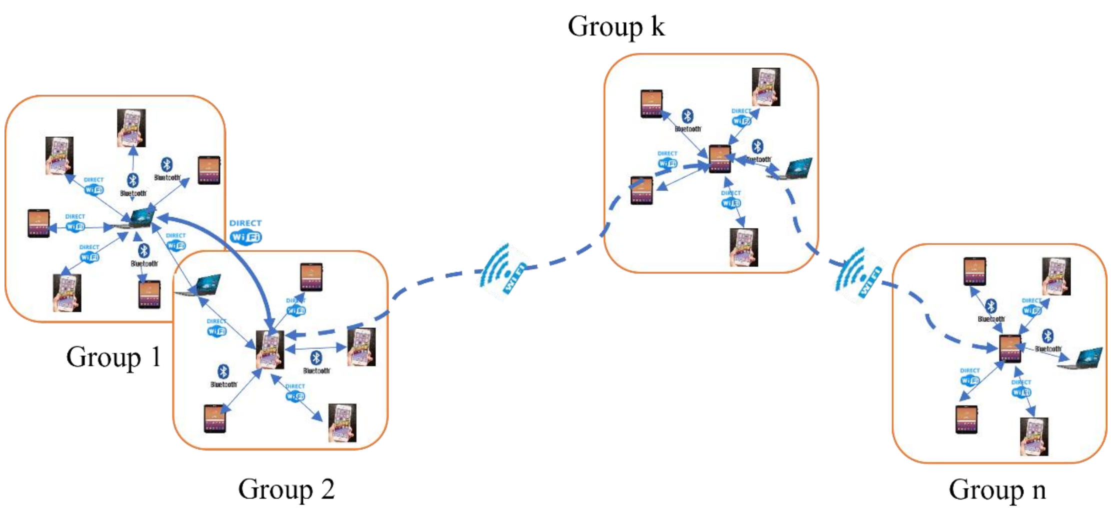


Figure VI-2 Devices' groups communicating by Wi-Fi and peer-to-peer. © 2021 Ghassan Fadlallah.

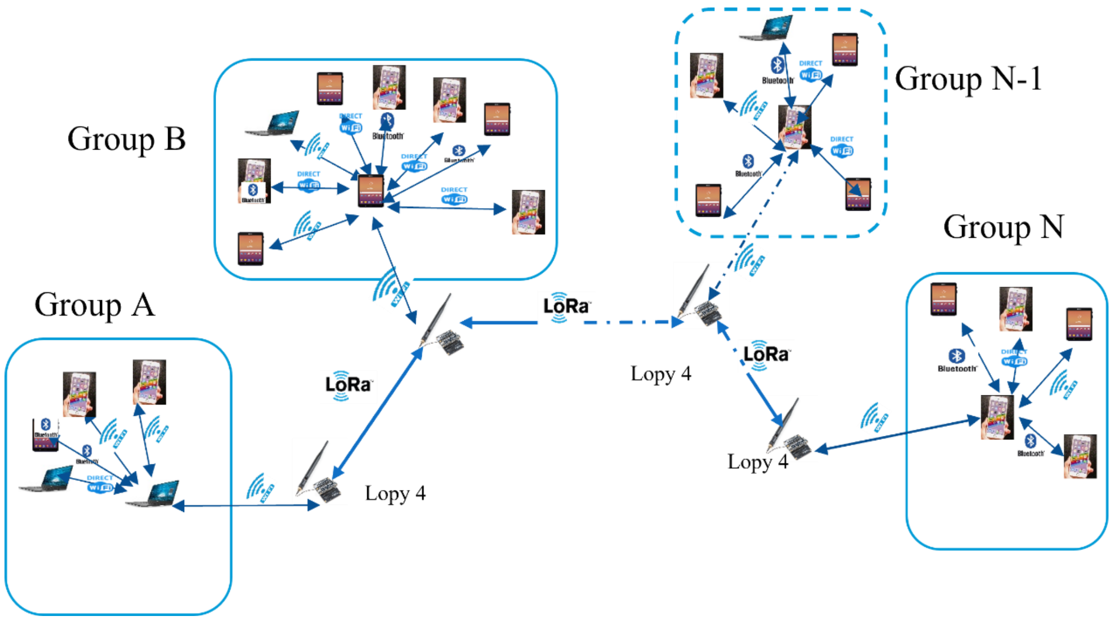


Figure VI-3 Devices' groups communicating by LoPy4. © 2021 Ghassan Fadlallah.

VI.3.3) STRUCTURE OF THE CONNECTIVITY MODEL

This communication model of maintaining connectivity is based on a network of many Pycom's LoPy4 microcontroller boards used as mini station bases for communication. These LoPy4s are programmed in this model in three modes with python in visual studio code. These modes are: LoPy transmitter, LoPy bridge (receiver/transmitter) and LoPy receiver (see Figure VI-4).

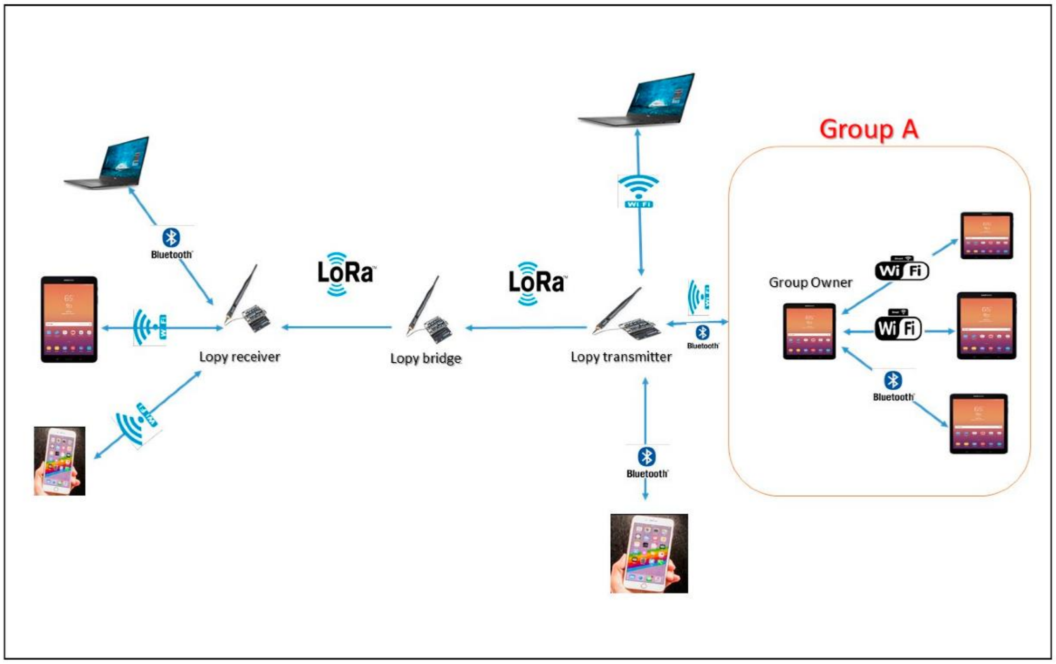


Figure VI-4 Connectivity Model Structure. © 2021 Ghassan Fadlallah.

VI.3.4) COMMUNICATION BEHAVIOURS OF THE APCAM MODEL

The devices can communicate, through the mobile applications implemented there, with the LoPy4s directly or through their group owners according to the following scenario (see Figure VI-5). The LoPy transmitter can receive data through Wi-Fi or Bluetooth.

(i) Bluetooth: LoPy4 can receive the information by Bluetooth using any device opening its Bluetooth option.

(ii) Wi-Fi: we created LoPy4 as a Wi-Fi access point such that any device (Android, IOS, Windows or Linux) that connects to this Wi-Fi can communicate with LoPy4 at this address: http://192.168.4.1/test using HTTP post socket.

(iii) Wi-Fi Direct: The devices are split into groups using mobile applications that we have implemented there based on Wi-Fi Direct technology. They use these applications to indicate their group owners. The group owner stores all information received from other group members in a local database. Then, it sends this information to the LoPy4 using a HTTP post socket.

As soon as the LoPy4 transmitter receives the information by Bluetooth or Wi-Fi, it transmits it to the LoPy4 bridge using the Lora socket.

The LoPy bridge is used to connect two remote LoPys: the LoPy transmitter and the LoPy receiver. It receives the information from the LoPy transmitter and sends it to the LoPy receiver. If it is not functional, the LoPy receiver cannot receive the information.

The LoPy receiver is also used as a Wi-Fi access point. All the devices (Android, IOS, Windows and Linux) that are connected to it can have the information that it receives from the LoPy4 transmitter via the LoPy4 bridge in the form of a table.

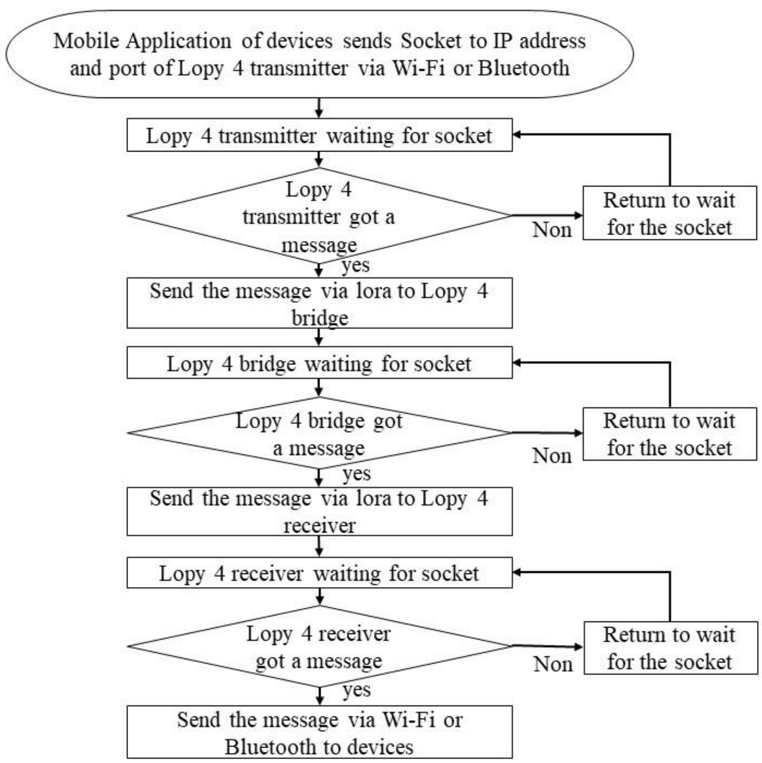


Figure VI-5 A visualization of the APCAM Model. © 2021 Ghassan Fadlallah.

VI.3.5) CONNECTIVITY ALGORITHMS TO MANAGE THE APCAM MODEL

The processes of synthesizing, managing and connecting materials in our network model are based on the following algorithms, which design the network as a series of mobile device groups. All the variables used in these algorithms are summarized in Table VI-2.

(1) Algorithm to calculate the score of device d: CalculatingScores(d): This algorithm, as presented in [54], calculates the score of each device based on five criteria. This score is required to select a group manager (or a group owner) and in the process of routing messages between devices. For the sake of clarity and completeness, it is described below.

Algorithm 1 CalculatingScores(d)

```
Begin
   get factor capacity values of d;
   ValScore = ValBattery ∗ PrcBattery + ValStock ∗ PrcStock + ValBandwidth ∗
PrcBandwidth + ValRam ∗ PrcRam + ValCPU ∗ PrcCPU;
   return ValScore;
End
```

(2) Algorithm that allows a device to create a list of its connected devices and their scores: This algorithm allows a device such as "d" to create a list of devices connected to it with their scores. When device "q" is connected to device "d", it sends its resource information to device "d." Then, device "d" calculates the score of device "q" by applying the algorithm CalculatingScores(q), then it adds device q to its list identified by two parameters: IP address and score.

Algorithm 2 Creating_List_Devices_Scores(d)

int i = 0, j = 0, score = 0;
while connection is active and $i \leq n$ // In case of Wifi and P2P, a group does not exceed n = 12 devices
{
  If device q is connecting to device d {
    $list_d[i][j] = ip\_q$;
    score = CalculatingScores(q);
    j = j+1
    $list_d[i][j] = score$;
    i = i+1;
    j = j-1; // value of j is 0 or 1, ex: $list_d[0][0] = ip\_q$,
        // $list_d[0][1] = score$, $list_d[1][0] =$, $list_d[1][1] =$, etc.
  }//End if
} //End while
Return $list_d$
End

(3) Algorithm to sort the list device scores (Sorting_Devices_by_Scores): Bubble sorting is an algorithm that progressively moves up the largest elements of an array [55].

Algorithm 3 Sorting_Devices_By_Scores ($\mathrm{ip_d}, list_d$)

// sorting based on the scores

```
int i, j, n, k;
Begin
  n = list_d.length;
  For (i = n; i> = 1; i --)
  {
     For (j = 2; j <= i; j ++)
     {
        k = 1; // rank of the line
        If list_d[k][j−1] > list_d[k][j] Then
        {
          tmp_scr = list_d[k][j−1];
          tmp_dvc = list_d[k+1][j−1];
          list_d[k][j−1] = [list]_d [k][j];
          list_d[k][j] = tmp_scr;
          list_d[k+1][j] = tmp_dvc;
        } End If
     } End For
  } End For
  return list_d;
End
```

(4) Algorithm to select the group manager: Selecting_Group_Manager: Let us randomly choose one of the *n*-connected devices $d_i, i = 1, \ldots, n$ to be the group owner (GO). It periodically receives data, including scores, from other members at various moments in time, such as every 5 min. At each of these moments, the GO begins to compare iteratively the scores of the members of its group. It starts by comparing itself with resources and scores such as d_i with d_{i+1} Consequently, the new GO becomes the device with the highest score. Again, by increasing index *i* by 1, the group owner GO will be compared with d_{i+1} to determine the new group owner until *i* = *n* - 1.

Algorithm 4 Selecting_Group_Manager

```
int n;
Begin
    list_d = Sorting_Devices_By_Scores(list_d);
    n = list_d.length;
    ip_go = list_d[1][n]; // Max(list_d)
    return ip_go; // Go : group owner
End
```

(5) Algorithm 5 Sending_Message (IP, IP) uses the previous algorithms to reach the target user (object). This algorithm routes a message from device A to target device B in areas covered by device-to-device radio connections such as Lora and Wi-Fi Direct. This algorithm is an iterative algorithm in which the device that has the message (at the

beginning it is A) creates its list of devices connected to it with their scores. Then, it looks for device B to check if B can be reached using this device's list to send it the message and accomplish its task. Otherwise, it searches through its list to find the device with the next highest score to B. This algorithm takes advantage of the device's mobility to allow the device carrying the message to move to find in its list of connected devices the appropriate one to send this message. It keeps doing this until it meets the intended device B. We have a dynamic graph because the nodes can be entered in and out of it continuously. These devices are all in constant motion. Therefore, the device carrying the message must move to communicate with the others. Depending on the latitude and longitude of the nodes between B, the device carrying the message and the candidate to carry it, we determine their positions relative to each other. Then, the distances are compared between B and each of the two nodes. The carrier of the message sends it to the other if it has the shortest distance to B. For example, let M be the device carrying the message and N be the candidate to carry it. We calculate the distances MB and NB, and if NB < MB then M sends the message to N.

Algorithm 5 for Sending_Message (ip_A, ip_B)

Local variables:
int n;
double n = 0; $distance_{TR} = 0$; $x_1 = 0$; $x_2 = 0$; $y_1 = 0$; $y_2 = 0$;
    $distance_{ipB} = 0$; $distance_{ip_{max}B} = 0$;
String ip_{act};// IP of the device being processed
String [][] $list_{act}$; // Array of the devices connected to the device being processed
String [][] $list_{act\_srt}$; // Array of devices sorted according to scores and connected to the device being processed
String min_scr; // threshold score Boolean flag = false
Begin
  ip = ip_A;$x_1 = x_{ip}$; $y_1 = y_{ip;}$
    $x_2 = x_B$; $y_2 = y_B$;
    $distance_{TR} = (x_2 - x_1)^2 - (y_2 - y_1)^2$; // distance between transmitter and receiver (Here, A and B)
  While ip != ip_B{
    $list_{act}$ = Creating_List_Devices_Scores(ip); // creating the list connected to the device
    $list_{act\_srt}$ = Sorting_Devices_By_Scores(ip, $list_{act}$);
    n = $list_{act\_srt}[0]$.length; // number elements in this array
    // check if B belongs to the list of devices connected to the device being processed
   For (i = n−1; i ≥ n; i--) {

```
ipmax = list_act_srt[1][i]; // starting by the device having the maximum score
If ipmax = ip_B; //the device being processed can
   //communicate the message directly to B
    Exit //as the message has arrived at its
   // destination B
Else{
       x1 = x_ip; y1 = y_ip;
       x3 = x_ipmax; y3 = y_ipmax;
       distance_ipB = (x_B − x1)^2 − (y_B − y1)^2; // distance between ip and B
       distance_ipmaxB = (x_B − x3)^2 − (y_B − y3)^2; // distance between ipmax and B
       If distance_ipmaxB < distance_ipB and min_scr ≤ list_act_srt[0] [i] {
           //ipmax is closer to B and its score > threshold score
           // the message is deposited at ipmax
           ip = ipmax;
         }//End If
 }//End If
} // End For
} // End While
End Algorithm
```

Tableau VI-2 APCAM models' models and description

Variable	Description
ValBattery	Battery energy
ValStock	Storage capacity
ValBandwidth	Bandwidth
ValRam	RAM capacity
ValCPU	CPU capacity
ValScore	Score value
PrcBattery	Percentage of battery influence
PrcStock	Percentage of storage influence
PrcBandwidth	Percentage of bandwidth influence
PrcRam	Percentage of RAM influence
PrcCPU	Percentage of CPU influence
Ip_d, ip_q, or ip_go	The IP address of device d, q or group owner
$list_d(2,n)$	An array of two rows and n columns for device d
$n = list_d.length$	Number of devices in $list_d$
tmp_dvc	Temporary device
tmp_scr	Temporary score

Before closing this section, let us say a word about the running times of these algorithms, which are, as we will see shortly, very efficient. Indeed, Algorithm 1 is obviously of $O(1)$-time. The time complexity of Algorithm 2 is dominated by the *while* loop iterated at most *n* times. Since the core of this loop takes $O(1)$-time, it follows that Algorithm 2 is $O(n)$. The time complexity of Algorithm 3 is dominated by the two loops nested one inside the other. Again, as the corps of the two loops is $O(1)$-time, it then follows that the global running time of Algorithm

3 is $O(n^2)$. Algorithm 4 is obviously of $O(1)$-time. Now, for Algorithm 5, there are two things to distinguish. First, the inner loop *for* runs in $O(n)$-time, whereas the running time of the outer loop while depends on the input. Indeed, as long as condition (ip != ip_B) is met, the loop while does not terminate. The device carrying the message moves to find an appropriate place in the list of devices connected to it to receive this message. This continues until it meets the intended device B. Therefore, in the worst case, the test will be done in $O(n)$ time, as we have *n* devices. Therefore, the overall time complexity of Algorithm 5 is $O(n^2)$.

VI.3.6) OPERATING MECHANISM AND SECURITY CONTROLS

We implemented applications of the previous algorithms with others that are needed to manage the model at the levels of devices, their groups, and message circulation through and between these groups. These applications must be installed on the participating devices. Among them, one is to form groups and elect their owners. Another is the client/server application where the server is supposed to be the owner of the group which sends its IP address and its communication gateway number to clients who are members of that group and to other group owners. Then, through these coordinates they exchange messages and information. The third one of these applications is to program the Lopys as transmitter, bridge and receiver.

The model is generally designed to be open for public use in harsh situations to allow participation and rescue for everyone. Thus, the architecture is proposed with exposed and unencrypted IP addresses such as http://192.168.4.1/test. In practice, this makes it non-secure and therefore dangerous to put into service. To remedy this situation of a public Lopy network, we have used a password for participants that is currently “Wi-Fi”. Likewise, we can change the public IP address of the page to a private one. To make the system more secure, we can also add an identification page to identify each user. Thus, each user must have a login and a password. For more security, we can further create a database via a registration form to the network of Lopys to verify the connection of the participants. However, it must be taken into account that the more important the security measures, the lower the chances are of the general public participating in and benefitting from this model. Therefore, the lower the chances are of rescuing people in harsh and catastrophic situations.

VI.4. EXPERIMENTAL STUDY

Relying on the type of the available communication technology, we classified the experimental study we conducted into two categories. Category A uses device-to-device radio communication technology only, such as Wi-Fi Direct, and Category B combines this technology with others that are available via Pycom's LoPy4, such as Radio Wi-Fi, Lora and Bluetooth. In both cases, we used a combination of mobile phones, tablets and laptops. In case A, we used mobiles and tablets, whereas in case B, we used laptops. The characteristics of several of the devices we used in this experimental study are:

Mobiles: (i) Smartphone Samsung Galaxy S21: CPU Octa-core (1 × 2.9 GHz Cortex-X1 & 3 × 2.80 GHz Cortex-A78 & 4 × 2.2 GHz Cortex-A55)-International, OS Android 11, One UI 3.1, 256 GB RAM, 512 GB storage. (ii) Smartphone Samsung Galaxy S6: Processor Octa-core, 2100 MHz, ARM Cortex-A57 and ARM Cortex-A53, 64-bit-14 nm, RAM 3 GB LPDDR4, OS Android 7.0 Nougat, 64 GB storage. (iii) Smartphone LG K40: OS Android 8.1 (Oreo), LG UX 7, CPU Octa-core 2.0 GHz Cortex-A53, 2 GB RAM, 32 GB storage.

Tablets: (i) Samsung Galaxy Tab A 10.1: Samsung Exynos 7904 CPU, 2 GB RAM, Operating System Android 9 with One UI, 32 GB storage. (ii) Samsung Galaxy Tab A 8.0: Processor Qualcomm Snapdragon 429 4 × 2 GHz, Cortex-A53, 2 GB RAM, Operating System Android 9.0 Pie, 32 GB storage. (iii) Samsung Galaxy Tab 7.0 Plus: processor 1.5 GHz quad-core, 1.5 GB of RAM, Operating System Android, 8 GB storage.

Laptops: (i) Laptop 1 with the following characteristics: Intel® Core™ i7 processor, x64-based processor, Windows 10 Pro 64-bit OS, 2.3 GHz CPU, and 16 GB of RAM. (ii) Laptop 2 with the following characteristics: Intel® Core™ i7 6820HP CPU, 2.7 GHz, x64-based processor, 64-bit OS, and 16 GB of RAM. (iii) MacBook Pro with the following characteristics:

Intel Core i7 quad core 2.8 GHz, macOS High Sierra and 16 Go of integrated memory LPDDR3 2 133 MHz.

Let us point out the data we used are generated live from the devices between which they were exchanged. The exchanged messages contain the information of the devices and the users.

VI.4.1) CATEGORY A

This category includes two types of techniques to achieve and enhance speed and communication in transmitting information:

(1) Finding the shortest path of maintained communication between two devices in the network through which information can be transferred as messages from one device to another. Ensuring the connectivity of the shortest path nodes are presented in the following three case studies [56]:

- (i) The nodes are obviously connected.
- (ii) Nodes must be moved.
- (iii) Using and moving a drone with the capability of P2P wireless technology.

(2) Establishing a network by forming a chain of adjacent groups of interfering devices within the available communication range. A device belonging to two groups can act as a gateway between the owners of these groups. The message must be sent in an intelligent way, taking into consideration the distance to the destination device and the continuity of communication through nearby devices. In the following, we present the application interface of forming groups and selecting their owner. An analysis of the data available from the device's interface in Figure VI-6 shows that the information about devices in its connection range is periodically communicated to it. In addition, the GO is also changed when the score varies. Different parts are used to show the most important components of the graphical interfaces of our system.

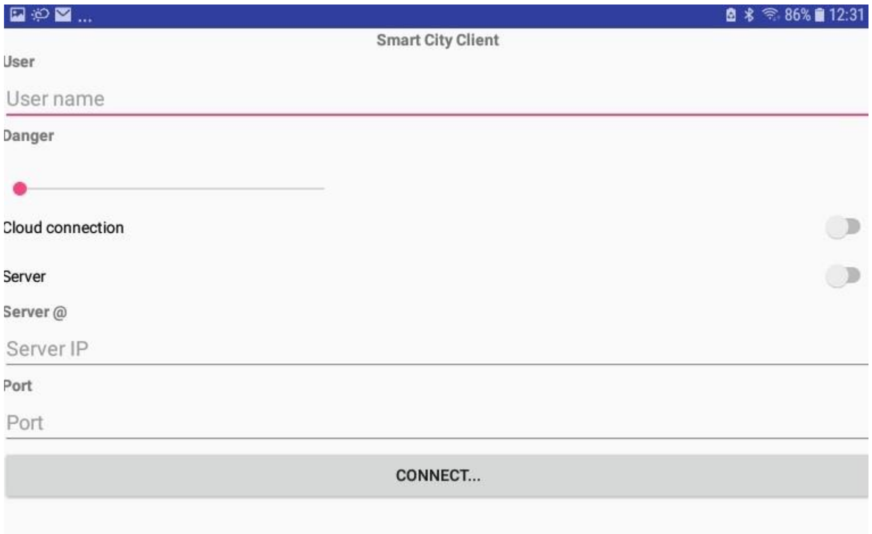


Fig. VI-6-Part 1. Client menu of the device's connection applications.

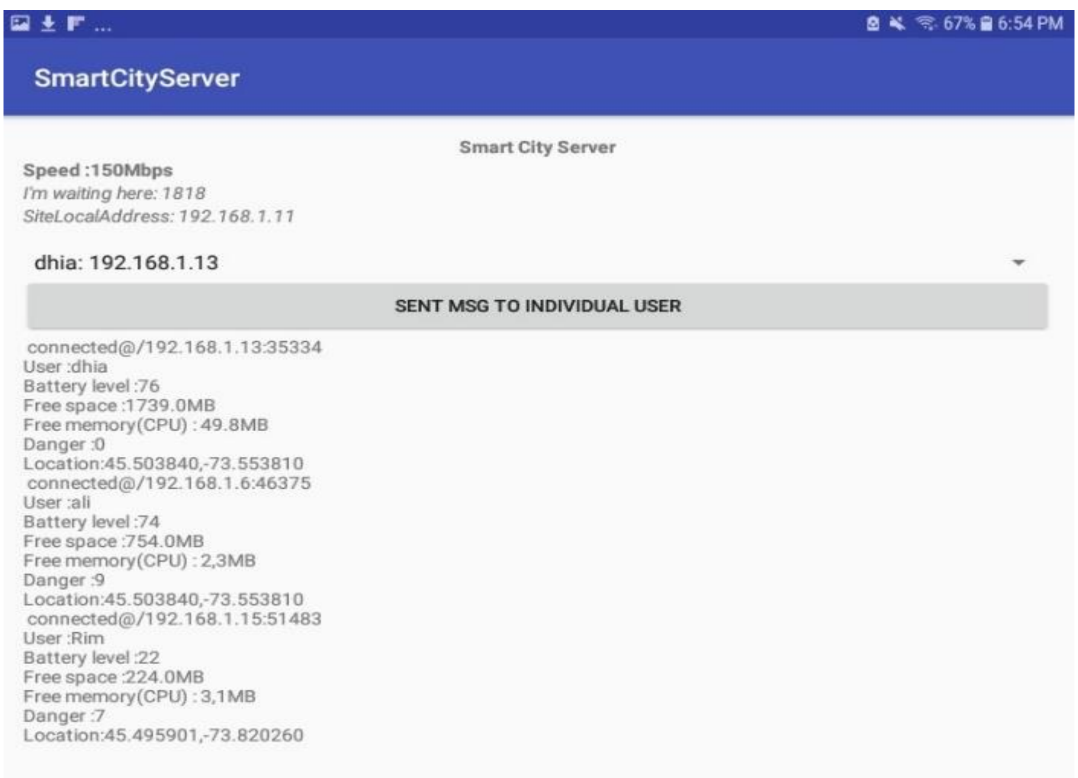


Fig. VI-6-Part 1 Server menu of the device's connection applications.

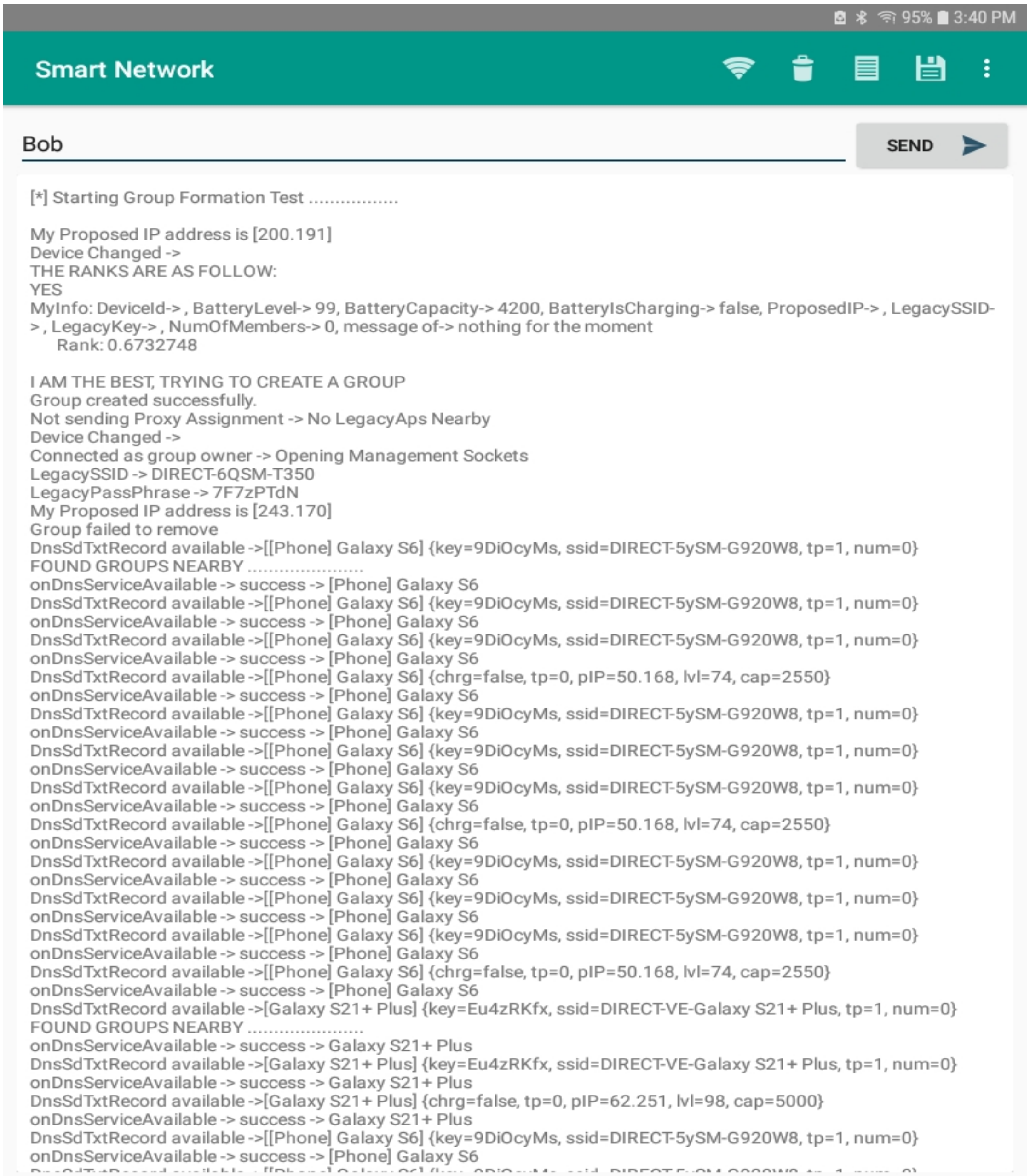


Fig. VI-6-Part 2: Forming a group of devices and selecting the group owner.

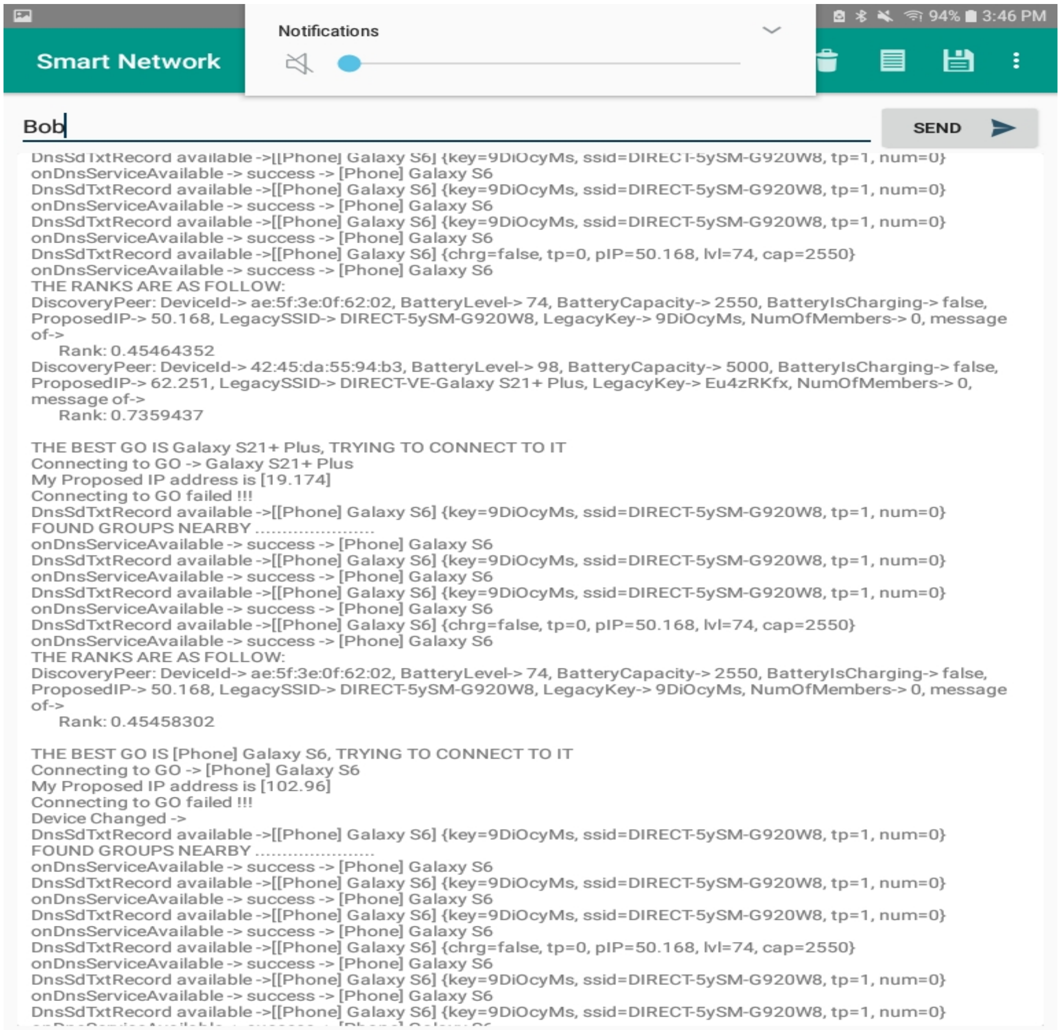


Fig. VI-6-Part 3: Sequel of Part 2, menu for forming a group of devices and selecting the group owner.

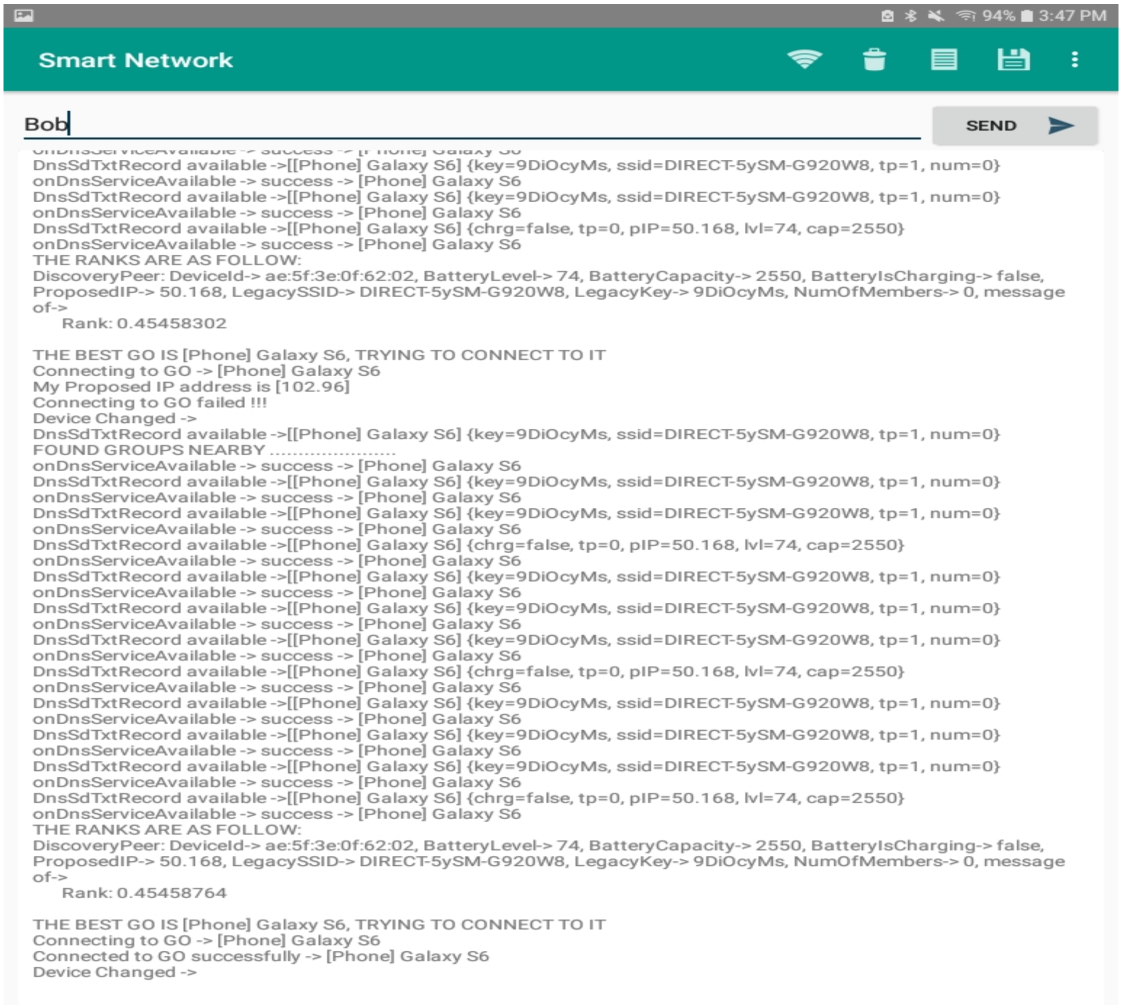


Fig. **Vi-6-**Part 4: Sequel of Part 3, menu for forming a group of devices and selecting the group owner.

Figure VI-6 Application interfaces used on devices. © 2021 Ghassan Fadlallah.

VI.4.2) CATEGORY B

This category integrates connection technology provided by Pycom's LoPy4 with existing Wi-Fi and device-to-device radio communication.

We designed and implemented a mobile application with a graphic interface (Figure VI-7) to exchange information through our model of connectivity (Figure VI-5). In the experimental study, we used a network like the one illustrated in Figure VI-5. We installed the LoPy bridge in a given position. Then, we placed both the LoPy transmitter and the LoPy receiver in cars that drove away from the LoPy bridge in opposite directions. We connected the devices to the LoPy transmitter whenever we travelled 150 m away from the LoPy bridge in both directions. The LoPy bridge received the device's information from the LoPy transmitter and transferred it to the LoPy receiver, which transmitted it to its nearby devices as shown in Tableau VI-3 and Tableau VI-4.

Because the team of volunteers could be scattered and distant in the experimental study, we used the same device in different places with the names of different users.

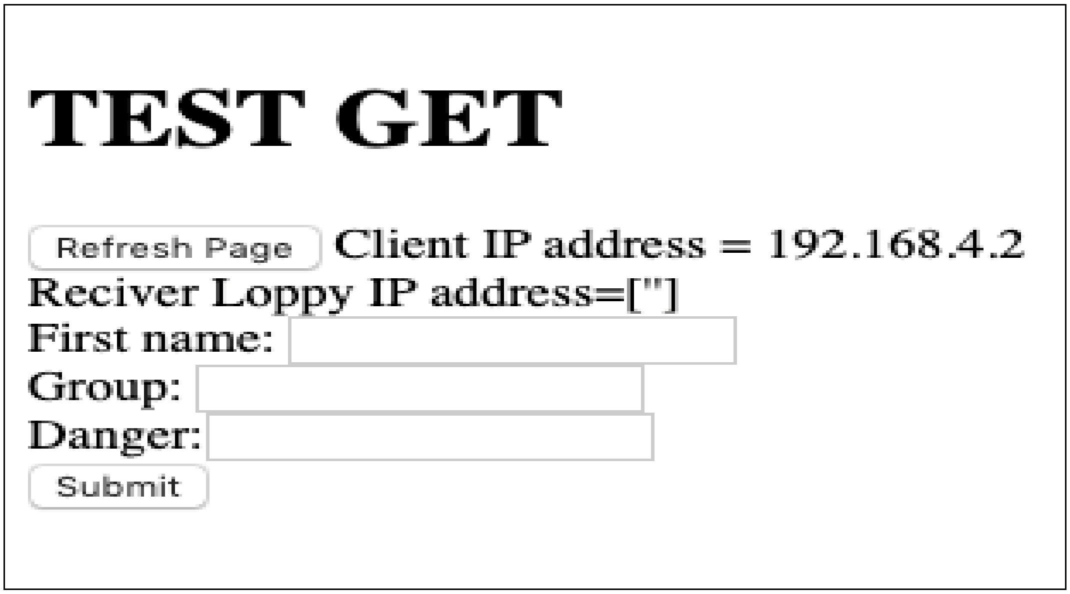


Figure VI-7 Info. transfer from LoPy transmitter to LoPy bridge. © 2021 Ghassan Fadlallah.

Throughout this experimental study, our autonomous model is shown to be efficient in providing communication and exchanging information over a range of more than two kilometres. As long as this can be done using a LoPy bridge, this range can be extended proportionally by using additional LoPy bridges.

Tableau VI-3**:** First experimentation study, information required from devices near the LoPy transmitter sent via the LoPy bridge to the LoPy receiver with 192.168.4.1/test. This communication model can be applied between transmitter–bridge as well as between bridge–receiver.

Tableau VI-3 First experimental study

Full Name	Group	Danger	IP
Eric Bouchard	G1	5	192.168.4.2
DIRECT-H8SM-G920W8	G1	1	192.168.4.3
Jad Tremblay	G2	3	192.168.4.6
Jean Richard	G2	4	192.168.4.2
Hugo Michel	G3	7	192.168.4.2
Hanna Hugo	G3	6	192.168.4.2
Luc Besson	G3	6	192.168.4\xae2

Table VI-4**:** Second experimentation study, information required from devices near the LoPy transmitter sent via the LoPy bridge to the LoPy receiver with 192.168.4.1/test. This communication model can be applied between transmitter–bridge as well as between bridge–receiver.

Tableau VI-4 Second experimental study

Full Name	Group	Danger	IP
Ali Jabril	G7	7	192.168.4.3
Mohamad Mickeal	G4	2	192.168.4.3
Alain Legof	N/A	4	Bluetooth
Sam	G5	3	192.168.4.2
Alain April	G3	7	192.168.4.3
DIRECT-ueSM-G920W8	G1	1	192.168.4.5
Dihya Messaoudi	N/A	4	Bluetooth
DIRECT-ueSM-G920W8	G1	1	192.168.4.5

The experimental study we conducted shows the efficiency of our communication model, which has many advantages over existing models, even those using 5G technology in terms of connection coverage. In fact, the 5G coverage does not exceed 500 m, while it exceeds 1 km through one Pycom’s LoPy4. In fact, the 5G coverage range tests on the millimeter wave band produced results in the order of 500 m around the pylon. This requires an important installation of MIMO (multiple input, multiple output) antennas for a single deployment of 5G standalone [57]. Furthermore, our model creates a topology to enhance inter-and intra-group communication. In addition, it selects the objects (devices) with the greatest resources to increase the duration of the connectivity and the functionality of the model. These objects are replaced dynamically by others if their resources decline. Moreover, even if there is no standard wireless or Internet connection, this model can continue to help people communicate using a combination of technologies such as Wi-Fi-Direct, Bluetooth and that of Pycom's LoPy4 from Lora.

VI.5. CONCLUSION AND FUTURE WORK

Ubiquitous smart and pervasive devices are the core elements of the Internet of Things, which is expanding, enhancing and integrating with distributed systems and mobile collaborative computing. The essential factor on which these paradigms are built and enhance their effectiveness is the communication technologies. Connectivity in networks is being improved and modernized by introducing and enhancing more modern wireless technologies as well as P2P connectivity. The main issue of various environments (harsh, urban or rural environment) is to maintain connectivity between objects (users), through both inter- and intra-group connectivity.

In this paper, we presented an autonomous collaborative pervasive architectural model of communication between ubiquitous smart mobile devices, especially those at the ends of networks. We designed four algorithms to increase the availability of communication between objects, taking into account many constraints such as the distributed environment and the mobility and the heterogeneity of these objects.

This model is based on the use of modern wireless communication technologies, in particular device-to-device radio communication. To overcome the short range of radio waves, we integrated Pycom's LoPy4 technology into this model as base stations and access points. The contribution of this model is that it maintains the permanence of the connectivity between its devices in various situations, especially harsh ones. In fact, the results of the experimental study proved the effectiveness of this model in maintaining connectivity using LoPy4 technology across a range of several kilometers in the form of LoPy transmitter, LoPy bridge and LoPy receiver. The proposed distributed collaborative and architectural model mainly improves the performance and efficiency in terms of overcoming difficulties of connectivity and management of pervasive networks, which are automatically and dynamically established in peripheral areas, especially in harsh environments.

For future work, we intend to include other emerging modern technologies of radio communication that could be compatible with LoPy4 so they can be integrated into this model for more effectiveness.

Author Contributions: G.F. designed the pervasive architectural model and its managing algorithms, then he implemented them; G.F., H.M. and D.R. analyzed and validated the computational algorithms; G.F. compared these algorithms and wrote the manuscript in consultation with H.M. and D.R. All authors have read and agreed the published version of the manuscript.

Funding: This work was partially funded by the Natural Sciences and Engineering Research council of Canada (NSERC), Le Fond de l'Université du Québec à Chicoutimi (FUQAC) and Le Décanat de la Recherche et de la Créaction (UQAC).

Institutional Review Board Statement: Not applicable but the University of Quebec at Chicoutimi requires that all research involving the participation of human beings, whether funded or not, conducted or supervised by its professors, employees or students be ethically reviewed. Links: http://recherche.uqac.ca/cer/ (accessed on 3 Sept. 2021).

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable. However, we generated live data from information of the used devices and from messages sent through intermediate devices.

Conflicts of Interest: The authors declare no conflicts of interest associated with this publication.

Références

[1] T. Kramp, R. van Kranenburg, S. Lange "Introduction to the Internet of Things," in *Enabling things to talk*; Bassi, A., Bauer, M., Fiedler, M., Kramp, T., van Kranenburg, R., Lange, S., Meissner, S. Eds.; Springer: Berlin/Heidelberg, Germany, pp. 1-10, 2013. [Online]. Available: https://doi.org/10.1007/978-3-642-40403-0_1

[2] G. Coulouris, J. Dollimore, T. Kindberg, G. Blair *"Distributed systems: concepts and design*," 5th ed.; Addison Wesley: Boston, MA, USA, 2012.

[3] L'informatique de poche dans l'entreprise ; Les cahiers d'e-Caria, [Online]. Available: https://dochero.tips/linformatique-de-poche-dans-lentreprise.html (accessed on Sept. 3, 2021).

[4] G. Fadlallah, Dj. Rebaine, H. Mcheick "Scheduling problems from workshop to collaborative mobile computing: A state of the art," *Int. j. comput. sci. inf. secur. (IJCSIS)*, vol. 16, pp. 47-69, 2018.

[5] N. Abbas, Y. Zhang, A. Taherkordi, T. Skeie "Mobile edge computing: A survey," *IEEE Internet things journal*, vol. 5, no 1, pp. 450-465, 2017. doi: 10.1109/JIOT.2017.2750180.

[6] K. Kai, W. Cong, L. Tao "Fog computing for vehicular ad hoc networks: Paradigms, scenarios, and issues," *J. China univ. posts telecommun*, vol. 23, no. 2, pp. 56-96, 2016.

[7] U.N. Kar, D.K. Sanyal "An overview of device-to-device communication in cellular networks," *ICT Express*, vol. 4, no 4, pp. 203-208, 2018.

[8] 3GPP, 3rd Generation partnership project (3GPP). [Online]. Available: http://www.3gpp.org/ (accessed on Apr. 11, 2021).

[9] A. Saranya, K. Rajasekaran, S. Chandra "An overview of mobile ad hoc networks," *International journal of computer science and information technology research*, vol. 4, no. 3, pp: 245-250, Month: July - September 2016, [Online]. Available: https://www.researchpublish.com/issue/IJCSITR/Issue-3-July-2016-September-2016

[10] G. Rakshith, R. Mahesh "New technology for machine-to-machine communication in Softnet towards 5G," *Int. j. wirel. mobile netw. (IJWMN)*, vol. 9, no. 2, pp. 1-23, 2017.

[11] R. Jones *"The international telecommunication union*," IEEE: New York, NY, USA, pp. 1248, 2019.

[12] C. Hennebert, G. Baldini, T. Peirce, M. Botterman, M. Talacchini, A. Guimaraes Pereira, M. Handte, D. Rotondi, H. Pohls "*IoT governance, privacy and security issues*," European research cluster on the internet of things, [Online]. Available: http://www.internet-of-things-research.eu/pdf/IERC_Position_Paper_IoT_Governance_Privacy_Security_Final.pdf (accessed on Sept. 3, 2021).

[13] Z. Weiwen, W. Yonggang, W. Dapeng "Collaborative task execution in mobile cloud computing under a stochastic wireless channel," *IEEE Transactions on wireless communications*, vol. 14, no 1, pp. 81-93, 2014, doi: 10.1109/TWC.2014.2331051.

[14] C.M. Wang, S.F. Hong, S.T. Wang, H.C. Chen "A dual-mode exerciser for a collaborative computing environment," In *Proceedings of the 11th Asia-Pacific, software conference*, IEEE computer society, Busan, Korea, pp. 240-248, 30 Nov.-3 Dec. 2004, doi: 10.1109/APSEC.2004.3.

[15] S. Radicati "Mobile Statistics Report: 2014-2018," *The radicati group, INC.* [Online]. Available: https://www.radicati.com/?page_id=54 (accessed on Feb. 3*, 2014*).

[16] Available online: https://www.statista.com/statistics/274774/forecast-of-mobile-phone-users-worldwide/ (accessed on Sept. 3, 2021).

[17] C. Rose, S. Eldridge, L. Chapin "The Internet of Things: An overview-understanding the issues and challenges of a more connected world," *Internet soc. (ISOC)* 2015. [Online]. Available: https://www.internetsociety.org/wp-content/uploads/2017/08/ISOC-IoT-Overview-20151221-en.pdf (accessed on Sept. 3, 2021).

[18] G. Fadlallah, H. Mcheick, D. Rebaine "Layered architectural model for collaborative computing in peripheral autonomous networks of mobile devices*,"* In *Proceedings of the 16th international conference on mobile systems and pervasive computing (MobiSPC)*, Halifax, NS, Canada, 19-21 Aug. 2019. In *Procedia Computer Science*, vol. 155, pp. 201-209, 2019, [Online]. Available: https://doi.org/10.1016/j.procs.2019.08.030

[19] J. Abdelaziz « Un cadre architectural pour la collaboration dans l'Internet des Objets. Une approche basée sur l'informatique en brouillard, » Thèse de doctorat, Université du Québec à Chicoutimi, Chicoutimi, QC, Canada, 2018.

[20] M.V. Steen, A.S. Tanenbaum "*Distributed systems,*" *3rd edition*, Pearson Education: London, UK, 2017.

[21] Lora technology. Available online: https://canada.newark.com/wireless-lora-technology (accessed on Sept. 3, 2021).

[22] M.; Elhoseny, A.E. Hassanien "Expand mobile WSN coverage in harsh environments," in *Dynamic wireless sensor networks*; Springer: Cham, Switzerland; pp. 29-52, 2019.

[23] H.P. Gupta, P.K. Tyagi, M.P. Singh "Regular node deployment for k -coverage in m -connected wireless networks," *IEEE Sensors journal,* vol. 15, no. 12, pp. 7126-7134, Dec. 2015, doi: 10.1109/JSEN.2015.2471837.

[24] D. Sharma, V. Gupta "Improving coverage and connectivity using harmony search algorithm in wireless sensor network," In *Proceedings of the 2017 international conference on emerging trends in computing and communication technologies (ICETCCT)*, Dehradun, pp. 1-7, India, 17-18 Nov. 2017.

[25] S.T. Hasson, A.A.R. Finjan "A suggested angles-based sensors deployment algorithm to develop the coverages in WSN," In *Proceedings of the 2018 2nd international conference on inventive systems and control (ICISC)*, Coimbatore, India, pp. 547-552, 19-20 Jan. 2018.

[26] H. Mcheick, M.S.B. Hatoum, Ghaddar, A. "CMHWN: Coverage maximization of heterogeneous wireless network," In *Proceedings of the 16th ACS/IEEE international conference on computer systems and applications AICCSA'19*, Abu Dhabi, United Arab Emirates, pp. 1-7, 3-7 Nov. 2019, doi: 10.1109/AICCSA47632.2019.9035307.

[27] A. Ghaddar, M.S.B. Hattoum, G. Fadlallah, H. Mcheick "MCCM: An approach for connectivity and coverage maximization," *Future internet*, vol. 12, no. 2, pages 19, 2020, [Online]. Available: https://doi.org/10.3390/fi12020019

[28] Z. Wang, F. Li, X. Wang, T. Li, T. Hong “A WiFi-Direct based local communication system,” in *Proceedings of the 2018 IEEE/ACM 26th international symposium on quality of service (IWQoS)*, Banff, AB, Canada, pp. 1-6, 4-6 Jun. 2018, doi: 10.1109/IWQoS.2018.8624171.

[29] A. J. Ferrer, J.M. Marques, J. Jorba “Towards the decentralized cloud: Survey on approaches and challenges for mobile, ad hoc, and edge computing,” *ACM computing surveys*, vol. 51, no. 6, pp 1-36, Nov. 2019. [Online]. Available: https://doi.org/10.1145/3243929

[30] G. Fadlallah, H. Mcheick, Dj. Rebaine, M. Adda “Towards mobile collaborative autonomous networks using peer-to-peer communication,” in *Proceedings of the 7th international conference on software engineering and new technologies (ICSENT 2018)*, Hammamet, Tunisia, art. no. 11, pp. 1-8, 26-28 Dec. 2018, ACM: New York, NY, USA, 2018, ISBN 978-1-4503-6101-9, doi:10.1145/3330089.3330107.

[31] A. Ahmed, E. Ahmed “A Survey on mobile edge computing,” In *Proceedings of the 2016 10th international conference on intelligent systems and control (ISCO)*, Coimbatore, India, pp. 1-8, 7-8 Jan. 2016, doi: 10.1109/ISCO.2016.7727082.

[32] Y. Jararweh, A. Doulat, A. Darabseh, M. Alsmirat, M. Al-Ayyoub, E. Benkhelifa “Sdmec: Software defined system for mobile edge computing,” In *Proceedings of the 2016 IEEE international conference on cloud engineering workshop (IC2EW)*, pp. 88-93, 4-8 Apr. 2016.

[33] M.T. Beck, S. Feld, A. Fichtner, C. Linnhoff-Popien, T. Schimper “Me-volte: Network functions for energy-efficient video transcoding at the mobile edge,” In *Proceedings of the 2015 18th international conference on intelligence in next generation networks*, pp. 38-44, 2015. doi: 10.1109/ICIN.2015.7073804.

[34] M. Satyanarayanan, P. Bahl, R. Caceres, N. Davies “The case for vm-based cloudlets in mobile computing,” *Pervasive comput.* 2009, vol. 8, pp. 14-23.

[35] D. Fesehaye, Y. Gao, K. Nahrstedt, G. Wang “Impact of cloudlets on interactive mobile cloud applications,” In *Proceedings of the IEEE 16th international enterprise distributed object computing conference (EDOC)*, Beijing, China, pp. 123-132, 10-14 Sept. 2012, doi: 10.1109/EDOC.2012.23.

[36] A.M.G. Masson, "MoCCA: A mobile cellular cloud architecture," *J. cyber secur. mobil.*, vol. 2, pp. 105-125, 2013. [Online]. Available: https://doi.org/10.13052/jcsm2245-1439.221

[37] E. Miluzzo, R. Cáceres, Y.F. Chen "Vision: mClouds computing on clouds of mobile devices," In *Proceedings of the third ACM workshop on mobile cloud computing and services MCS*; Lake District, UK, pp. 9-14, 25 Jun. 2012, [Online]. Available: https:/doi.org/10.1145/2307849.2307854 (accessed on Sept. 3, 2021).

[38] A. Hammam, S. Senbel "A reputation trust management system for ad hoc mobile clouds," *Intelligent systems reference library*, vol. 70, pp. 519--539, 2014.

[39] A.A. Mehanna, M.A. Abdel-Fattah, S.M. Abdel-Gaber "Cloudlet: A mobile cloudlet model using Wi-Fi Direct," *Int. j. comput. sci. inf. secur. (IJCSIS)*, vol. 14, 2016.

[40] Wi-Fi Alliance, P2P task group. *Wi-Fi peer-to-peer (P2P) technical specification*; version 1.5. [Online]. Available: https://cse.iitkgp.ac.in/~bivasm/sp_notes/wifi_direct_2.pdf (accessed on Sept. 3, 2021).

[41] C. Funai, C. Tapparello, W.Heinzelman "Enabling multi-hop ad hoc networks through Wi-Fi Direct multi-group networking," In *Proceedings of the international conference on computing, networking and communications*, Santa Clara, CA, USA; pp. 491-497, 26-29 Jan. 2017.

[42] C. Casetti, C.F. Chiasserini, Y. Duan, P. Giaccone, A.P. Manriquez "Data connectivity and smart group formation in Wi-Fi Direct multi-group networks," *IEEE Trans. netw. serv. manag.*, vol. 15, pp. 245-259, 2018.

[43] I. Yaqoob, E. Ahmed, A. Gani, S. Mokhtar, M. Imran "Mobile ad hoc cloud: A survey," *Wirel. commun. mob. comput.*, vol. 16, pp. 2572-2589, 2016.

[44] V. Balasubramanian, A. Karmouch "An infrastructure as a service for mobile ad hoc cloud," in *Proceedings of the IEEE 7th annual computing and communication workshop and conference (CCWC)*, Las Vegas, NV, USA, pp. 1-7, 9-11 Jan. 2017.

[45] I. Yaqoob, E. Ahmed, A. Gani, S. Mokhtar, M. Imran "Heterogeneity aware task allocation in mobile ad hoc cloud," *IEEE Access,* vol. 5, pp. 1779-1795, 2017.

[46] G. Kirby, A. Dearle, A. Macdonald, A. Fernandes "An approach to ad hoc cloud computing," *arXiv preprint arXiv:1002.4738*, 2010.

[47] M. Shiraz, A. Gani, R.H. Khokhar, R. Buyya "A review on distributed application processing frameworks in smart mobile devices for mobile Cloud computing," *IEEE commun. surv. tutor*., vol. 15, pp. 1294-1313, 2013, *[Online]. Available:* https://doi.org/10.1109/SURV.2012.111412.00045

[48] D.T. Hoang, C. Lee, D. Niyato, P. Wang "A survey of mobile cloud computing: Architecture, applications, and approaches," *wirel. commun. mob. comput*., vol. 13, pp. 1587-1611, 2013, [Online]. Available: http://dx.doi.org/10.1002/wcm.1203

[49] B.C. Sherin, E.A.M. Anita "A survey of scheduling algorithms for wireless ad-hoc networks," *Int. j. adv. sci. eng*., vol. 4, pp. 776-787, 2018. [Online]. Available: www.mahendrapublications.com (accessed on Sept. 3, 2021).

[50] Pervasive and mobile computing, science direct journal. Publisher: Elsevier. [Online]. Available: https://www.researchgate.net/journal/Pervasive-and-Mobile-Computing-1574-1192 (accessed on Sept. 3, 2021).

[51] T.S. Rappaport, Y. Xing, G.R. MacCartney, A.F. Molisch, E. Mellios, J. Zhang "Overview of millimeter wave communications for fifth-generation (5G) wireless networks-with a focus on propagation models," IEEE Trans. antennas propag., vol. 65, pp. 6213-6230, 2017.

[52] T. Oide, T. Abe, T. Suganuma "Infrastructure-less communication platform for off-the-shelf android smartphones," *Sensors* 2018, vol. 8, no. 3, 2018.

[53] Pycom Enterprise, Description and technical details of Pycom's Lopy4. [Online]. Available: https://static6.arrow.com/aropdfconversion/c500cfa0a971d49ac7d694a083bb5f20650e211a/pgurl_6269436244227100.pdf (accessed on Apr. 12, 2021).

[54] G. Fadlallah, Dj. Rebaine, H. Mcheick "A greedy scheduling approach for peripheral mobile intelligent systems," *IoT*, vol. *2*, pp. 249-274, 2021. [Online]. Available: https://doi.org/10.3390/iot2020014

[55] Le tri a Bulle. [Online]. Available: https://rmdiscala.developpez.com/cours/LesChapitres.html/Cours4/TBchap4.6.htm (accessed on Sept. 3, 2021).

[56] G. Fadlallah, H. Mcheick, Dj. Rebaine "Internet-of-Things (IoT) shortest path algorithms and communication case studies for maintaining connectivity in harsh environments," *2020 International symposium on networks, computers and communications (ISNCC2020)*, Montreal, QC, Canada, pp. 1-8, 20-22 Oct. 2020.

[57] Viavi enterprise, technologie 5G, [Online]. Available: https://www.viavisolutions.com/fr-fr/technologie-5g (accessed on Sept. 3, 2021).

CONCLUSION

Nos travaux de recherche se sont concentrés sur l'étude de l'environnement de la communication entre divers dispositifs mobiles et omniprésents en termes du mode et de la qualité de leur connectivité au sein du paradigme de l’Internet des objets (IoT). Ainsi, cette étude met en évidence leur collaboration, leur rapidité et leur qualité de réalisation des applications dans diverses conditions difficiles. Nos études se sont particulièrement focalisées sur les dispositifs opérant à la périphérie des réseaux dans l’espace de l’IoT, afin d'améliorer la qualité et la fiabilité de service en milieu sévère et difficile comme par exemple ceux de catastrophes et ruraux. Le but de cette recherche est de pallier le problème de perte de communication entre ces dispositifs en contribuant à assurer sa continuité dans ces environnements. Ceci renforce la coopération entre ces dispositifs pour communiquer les informations nécessaires et de fournir des services appropriés à la population, notamment dans les opérations de sauvetage.

En général, les réseaux sans fil sont le moyen le plus évident à nos jours qui fournit des services de communications partout dans les divers milieux. Le Wi-Fi est l'une des technologies les plus utilisées à bord de ces réseaux sans fil. Elle se caractérise par sa large gamme d'utilisation, en raison de sa facilité de déploiement et de son faible coût. Cela s'ajoute à sa structure dynamique qui favorise la mobilité des utilisateurs tout en restant connecté grâce à ses antennes et ses stations de base. Malheureusement, ces dernières peuvent être soumises à une congestion potentielle du trafic de communications et de dysfonctionnements qui bloquent ou ralentissent les technologies de communication, notamment dans des environnements difficiles et catastrophiques.

Nous avons surmonté les difficultés mentionnées ci-dessus dans ces environnements qui entravent les performances de ces appareils dans le cadre de l'Internet des objets. En conséquence, nous avons travaillé sur cette base afin d'améliorer les architectures émergentes dans lesquelles ces dispositifs sont intégrés avec les technologies modernes de communication et d'informatique collaborative mobile distribuée à travers le paradigme de l'IoT. À cette fin, nous avons travaillé simultanément sur deux niveaux :

Le premier niveau se concentre sur le maintien et la continuité de la communication dans les différents environnements difficiles. Alors, l'intégration des technologies de communication directe peer-to-peer inclues celles de Wi-Fi Direct, Bluetooth et celles fournies par LoPy4 de Pycom peut combler la lacune causée par une défaillance de connexion sans fil, une bande passante insuffisante ou des réseaux défaillants. Pour cela, nous avons établi un modèle architectural en six couches pour maintenir la connectivité entre ses dispositifs mobiles intelligents omniprésents, en particulier ceux situés à la périphérie des réseaux. De même, nous avons conçu plusieurs algorithmes pour gérer ces objets et leurs groupes, en tenant compte de nombreuses contraintes telles que leurs caractéristiques de mobilité et d’hétérogénéité dans un environnement distribué.

Le second s'intéresse à l'amélioration de la technique d'ordonnancement et d'équilibrage des tâches afin d'accélérer leur réalisation tout en préservant les capacités déjà limitées des dispositifs. À cet égard, nous avons conclu que les efforts dans ce domaine se concentrent sur l'optimisation des mesures souhaitées de la réalisation des tâches dans les plus brefs délais. Mais, cela semble très difficile à atteindre en utilisant des politiques exactes et optimales d'ordonnancement et d'équilibrage des tâches. Cela est dû à la complexité des algorithmes qui donnent des solutions exactes. Par conséquent, une autre piste est adoptée, dans la littérature, au moyen d'algorithmes heuristiques et approchés qui recherchent des politiques optimales proches de l'exacte, ce qui diminue considérablement le temps d'exécution. En conséquence, nous avons proposé un nouvel algorithme d'ordonnancement "Greedy" ou glouton pour les systèmes mobiles intelligents autonomes, en particulier ceux trouvés à la périphérie du réseau. Cet algorithme tire parti de la capacité de ces dispositifs à maintenir la connectivité via une connexion radio P2P. Cette fonctionnalité permet aux dispositifs de continuer à communiquer et donc de collaborer entre eux, même en cas de panne ou de congestion de la communication sans fil. Ainsi, ils peuvent toujours effectuer les tâches nécessaires à tout moment et dans n'importe quelle situation.

Objectifs accomplis

L'objectif principal de cette thèse est de concevoir un modèle autonome d'architecture mobile collaborative à la périphérie des réseaux qui soit efficace ayant une qualité de service (QoS) adaptée et réalisée dans différents environnements. Ce modèle s'appuie sur de nombreuses technologies de communication modernes telles que le sans-fil, la radio P2P et celles fournies par Lopy 4 de Pycom, ayant une portée de quelques kilomètres, pour maintenir la continuité de communication dans divers environnements, même les sévères. Cet objectif principal a inclus la continuité de la communication entre les dispositifs et le renforcement des performances des réseaux périphériques mobiles en les améliorant tant au niveau matériel que logiciel. Cet objectif peut être réalisé en intégrant et améliorant les dernières technologies d'équilibrage de charge et d'ordonnancement afin d'optimiser leurs métriques.

Comme nous l'avons susmentionné, nous avons identifié l'objectif principal de cette thèse sous forme de quatre sous-objectifs spécifiques sur lesquels nous avons concentré nos efforts. Ensuite, nous avons publié notre recherche et les résultats dans les cinq articles présentés dans cette thèse qui ont montré l'étendue du succès dans la réalisation de ces sous-objectifs classés en deux catégories. La première est l'établissement du mécanisme de communication dans le modèle conçu à la périphérie du réseau qui comprend le sous-objectifs (O1, O2 et O3) suivants:

1. Maintenir la connexion des appareils en toutes les situations à tout moment, (O1)
2. Rendre les réseaux et leurs utilisateurs plus autonomes, (O2)
3. Compenser le chargement/déchargement vers/depuis les serveurs Cloud et d'autres architectures intermédiaires et périphériques. (O3)

La deuxième catégorie est le renforcement de la performance de ce modèle qui constitue le sous-objectif (O4) suivant:

4. Améliorer la capacité des appareils à économiser de l'énergie et à accélérer l'exécution des tâches qui y sont effectuées grâce au développement de techniques d'ordonnancement et de répartition équitable de la charge. (O4)

Dans le cadre de notre recherche nous avons étudié les différentes architectures informatiques de calcul ou « computing architectures » dans le paradigme de l'IoT avec leurs modes de connectivités et leurs modèles de couches. Selon nos études et analyses de ces architectures et technologies de communication, nous avons établi un modèle architectural autonome collaboratif omniprésent à la périphérie du réseau. Les études expérimentales dans ces articles prouvent que ce modèle réalise les sous-objectifs spécifiques (O1), (O2) et (O3) (voir les chapitres 2, 5 et 6 où nous avons travaillé et réalisé ses trois sous-objectifs).

De même, nos études sur les algorithmes d'ordonnancement et les approches d'équilibrage de charge appliquées spécifiquement dans les architectures informatiques dans l'IoT ont abouti à la conception de notre nouvel algorithme « gourmand ». Cet algorithme prend en compte les caractéristiques du modèle que nous avons construit en termes de capacités de ses dispositifs et ses technologies de connexion. Les résultats des études expérimentales que nous avons menées et publiées sur cet algorithme ont prouvé qu'il atteint le sous-objectif spécifique (O4) (voir les chapitres 3 et 4).

Avantages, limitations et orientations futures

La vision de l'Internet des Objets (IoT) indique que les architectures dominantes et gérant ce paradigme souffrent de problèmes au niveau des communications ainsi qu'au niveau des applications qui les administrent et mettent en œuvre leurs tâches.

Il est bien établi que les systèmes de communication sont toujours en développement rapide et sont riche en diversité. Cependant, leurs services sans fil ou radio pair-à-pair sont sujets à perturbation, détérioration et interruption, malgré leur développement constant. Ces systèmes de communication sont toujours sujets à des dysfonctionnements, aussi avancés soient-ils, avec l'émergence de la 5G et de la 6G dans le futur, en termes de débit de télécommunication, de temps de latence très court et de nombre de connexions simultanées par une surface couverte. La courte portée P2P de connexion est l'une des lacunes qui n'a pas été comblée malgré ces dernières générations de développement des technologies de la communication qui reposent toujours sur les stations de base. Par conséquent, il était nécessaire de chercher la mise en place d'un modèle architectural omniprésent permettant de maintenir la continuité de la communication, en particulier dans des environnements difficiles causés par de nombreuses raisons, y compris les catastrophes naturelles.

Avec la diversité des technologies de communication sans fil, nous avons principalement adopté celles qui ont l'auto-capacité de communication radio pair-à-pair pour être intégrer avec Wi-Fi si disponible. La raison en est que dans des environnements difficiles, les stations de base de Wi-Fi peuvent être trop chargées ou endommagées en premier temps. Alors, elles restent hors de service pendant une longue période de temps en attendant d'être réparées. Donc, le Lopy 4 de Pycom a été choisi comme point d’accès alternatif, car il peut donner lieu à une longue portée de plus de kilomètres.

Bien que le modèle conçu ait démontré sa capacité à maintenir la continuité de la connectivité, il est limité en termes de vitesse de connexion et de portée de certaines de ses technologies adoptées, telles que Wi-Fi Direct et Bluetooth. Ainsi qu'en termes de capacités de ses dispositifs mobiles et omniprésents.

Dans les travaux futurs, nous comptons renforcer ce modèle architectural en accédant à de nouvelles technologies de communication, ayant des capacités plus grandes pouvant être intégrées aux technologies actuelles. En plus, nous visons d'intégrer des dispositifs avec des capacités plus importantes à différents niveaux tels que le calcul informatique, le stockage et la réduction de la consommation d'énergie. De même, nous envisageons d'intégrer dans ce modèle un cadre portable, extensible et évolutif tel que « Distributed Actor Model for Mobile Platforms (DAMMP) » afin d'établir un réseau dynamique d'appareils hétérogènes utilisant plusieurs technologies de communication à la fois.

BIBLIOGRAPHIE

INTRODUCTION

[1] B. hun "Special issue: pervasive computing technology and its applications," *Wireless communications and mobile computing wirel. commun. mob. comput.*, vol. 10, pp, 1281-1282, 2010, doi: 10.1002/wcm.1052, Published online in Wiley Online Library (wileyonlinelibrary.com), [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1002/wcm.1052 (accessed on Oct. 15 2021).

[2] G. Coulouris, J. Dollimore, T. Kindberg, G. Blair, "*Distributed systems: concepts and design*", Fifth edition, published by Addison Wesley, May 2012.

[3] G. Fadlallah, Dj. Rebaine, H. Mcheick, "Scheduling problems from workshop to collaborative mobile computing: A state of the art," *International journal of computer science and information security (IJCSIS)*, vol. 16, no. 1, pages 23, Jan. 2018.

[4] N. Abbas, Y. Zhang, A. Taherkordi, T. Skeie, "Mobile edge computing: A survey," *in IEEE Internet of Things journal*, vol. 5, no. 1, pp. 450-465, Feb. 2018, doi: 10.1109/JIOT.2017.2750180.

[5] K. Kai, W. Cong, L. Tao, "Fog computing for vehicular ad-hoc networks: paradigms, scenarios, and issues," *The journal of China universities of posts and telecommunications*, vol. 23, no. 2, pp. 56-96, 2016.

[6] M. Satyanarayanan, P. Bahl, R. Caceres, N. Davies, "The case for vm-based cloudlets in mobile computing," *Pervasive computing*, IEEE, vol. 8, no. 4, pp. 14 -23, 2009.

[7] D. Fesehaye, Y. Gao, K. Nahrstedt, G. Wang "Impact of cloudlets on interactive mobile cloud applications," In: *IEEE 16th international enterprise distributed object computing conference (EDOC)*, IEEE; Beijing, China, pp. 123-32, 2012.

[8] J. Abdelaziz, "Un cadre architectural pour la collaboration dans l'internet des objets : une approche basée sur l'informatique en brouillard," Thèse de doctorat, Université du Québec à Chicoutimi, Chicoutimi, 2018.

[9] A.J Ferrer, J.M. Marques, J Jorba, "Towards the decentralized cloud: survey on approaches and challenges for mobile, ad hoc, and edge computing," *ACM computing survey*, vol. 51, no. 6, 2019.

[10] A. Hammam, S. Senbel, "A reputation trust management system for ad hoc mobile clouds," *Intelligent systems reference library,* vol. 70, pp. 519-539, 2014.

[11] A. Mishra G. Masson, "MoCCA: A mobile cellular cloud architecture," *Journal of cyber security and mobility*, vol. 2, no. 2, pp.105-125, 2013. [Online]. Available: https://doi.org/10.13052/jcsm2245-1439.221

[12] E. Miluzzo, R. Cáceres, Y. farn Chen "Vision: mClouds computing on clouds of mobile devices," *Proceedings of the third ACM workshop on mobile cloud computing and services MCS*, vol. 12, pp. 9-14, 2012. [Online]. Available: https:/doi.org/10.1145/2307849.2307854.

[13] A. A. Mehanna, M. A. Abdel-Fattah, S. Abdel-Gaber, M.Cloudlet: "A mobile cloudlet model using Wi-Fi Direct," *International journal of computer science and information security (IJCSIS)*, vol. 14, no. 11, pages 8, Nov. 2016.

[14] C. Funai, C. Tapparello, W. Heinzelman, "Enabling multi-hop ad hoc networks through Wi-Fi Direct multi-group networking," In *Proceedings of the international conference on computing, networking and communications*, Santa Clara, CA, USA, pp. 491-497, 26-29 Jan. 2017.

[15] C. Casetti, C.F. Chiasserini, Y. Duan, P. Giaccone, A.P. Manriquez, "Data connectivity and smart group formation in Wi-Fi Direct multi-group networks," *IEEE Transaction on network. service management*, vol. 15, no. 1, pp. 245-259, 2018.

[16] B. C. Sherin, E.A. Mary Anita, "a survey of scheduling algorithms for wireless ad-hoc networks," *International journal of advanced science and engineering*, vol.4, no. 4, pp: 776-787. 2018. [Online]. Available: www.mahendrapublications.com

[17] A. Saranya, K. Rajasekaran, S. Chandra, "An overview of mobile ad hoc networks," *International journal of computer science and information technology research*, vol. 4, no. 3, pp. 245-250, Jul. – Sept. 2016.

[18] U. N. Kar, D. K. Sanyal, "An overview of device-to-device communication in cellular networks," *ICT Express*, vol. 4, no. 4, pp. 203-208, 2018.

[19] 3GPP, 3rd generation partnership project (3GPP). [Online]. Available: http://www.3gpp.org/ (accessed on Apr. 11 2021).

[20] G. Rakshith, R. Mahesh, "New technology for machine to machine communication in softnet towards 5G," *International journal of wireless & mobile networks (IJWMN)*, vol. 9, no. 2, Apr. 2017.

[21] R. W. Jones, "The international telecommunication union," *IEEE Antennas and propagation society international symposium 1997*. digest, vol. 2, pp. 1248, 1997, doi: 10.1109/APS.1997.631791. [CrossRef]

[22] C. Hennebert, G. Baldini, T. Peirce, M. Botterman, M. Talacchini, A. Guimaraes Pereira, M. Handte, D. Rotondi, H. Pohls "*IoT governance, privacy and security issues*," European research cluster on the internet of things, [Online]. Available: http://www.internet-of-things-research.eu/pdf/IERC_Position_Paper_IoT_Governance_Privacy_Security_Final.pdf (accessed on 21 Jan. 2020).

[23] Z. Weiwen, W. Yonggang, W. Dapeng, "Collaborative task execution in mobile cloud computing under a stochastic wireless channel", *IEEE Transactions on wireless communications*, vol. 14, no. 1, Jan. 2015.

[24] C. M. Wang, S. F. Hong, S. T. Wang, H. C. Chen "A dual-mode exerciser for a collaborative computing environment", *11th Asia-pacific software engineering conference,* pp. 240-248, 2004, doi: 10.1109/APSEC.2004.3.

[25] S. Mishra, N. Mathur "Load balancing optimization in LTE/LTEA cellular networks: A review," *arXiv 2014*, arXiv:1412.7273.

[26] T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein "*Introduction to algorithms*," 3rd ed.; MIT Press: Cambridge, MA, USA, 2009.

[27] P. Černý, E.M. Clarke, T.A. Henzinger, A. Radhakrishna, L. Ryzhyk, R. Samanta, T. Tarrach "From non-preemptive to preemptive scheduling using synchronization synthesis,"

Formal methods in system design, vol. 50, no. 2, pp. 97-139, 2017, doi:10.1007/s10703-016-0256-5.

[28] P. Hosein, S. Boodhoo "Event scheduling with soft constraints and on-demand re-optimization," in *Proceedings of the 2016 IEEE international conference on knowledge engineering and applications (ICKEA)*, Singapore, pp. 62-66, 28-30 Sept. 2016, doi:10.1109/ICKEA.2016.7802993.

[29] P. Weaver "A brief history of scheduling-back to the future," in *Proceedings of the myPrimavera conference*, Canberra, Australia, pp. 1-24, 4-6 Apr. 2006.

[30] F.K. Levy, G.L. Thompson, J.D. Wiest "ABCs of the critical path method," *Harv. bus. rev.*, vol. 42, pp. 98-108, 1963.

[31] P. Suresh, J.V. Daniel, V. Parthasarathy, R.H. Aswathy "A State of the art review on the Internet of Things (IoT) history, technology and fields of deployment," in *Proceedings of the 2014 international conference on science engineering and management research (ICSEMR)*, Chennai, India, pp. 1-8, 27-29 Nov. 2014.

[32] T. Wang, X. Wei, C. Tang, J. Fan "Efficient multi-tasks scheduling algorithm in mobile cloud computing with time constraints," *peer-to-peer netw. appl.*, vol.11, pp. 793-807, 2018, doi:10.1007/s12083-017-0561-9.

[33] D. Rahbari, M. Nickray "Low-latency and energy-efficient scheduling in fog-based IoT applications," *Turk. J. Electr. Eng. Comput. Sci.*, vol. 27, pp. 1406-1427, 2019.

[34] Pycom LoPy4 development board datasheet. [Online]. Available: https://docs.pycom.io/datasheets/development/lopy4 (accessed on Jan. 31, 2021).

[35] G.M. Bianco, A. Mejia-Aguilar, G. Morocco "Radio wave propagation of LoRa systems in mountains for search and rescue operations," In *Proceedings of the 2020 XXXIIIrd general assembly and scientific symposium of the international union of radio science*, Rome, Italy; pp. 1-3, 29 Aug.-5 Sept. 2020, doi:10.23919/URSIGASS49373.2020.9232231.

[36] G. Fadlallah, H. Mcheick, Dj. Rebaine, "Layered architectural model for collaborative computing in peripheral autonomous networks of mobile devices," *The 16th International conference on mobile systems and pervasive computing (MobiSPC)*, Aug. 19-21, 2019,

Halifax, Canada, *ScienceDirect, Procedia Computer Science*, vol. 155, pp. 201-209, 2019, [Online]. Available: www.sciencedirect.com (accessed on Oct. 15, 2021).

[37] https://pycom.io/product/lopy4/ (accessed on Oct. 15, 2021).

[38] Y. Lou, J. Chen, L. Zhang, D. Hao "A survey on regression test-case prioritization," *Adv. comput.*, vol. 113, pp. 1-46, 2019.

[39] Logistic application of greedy algorithms, Vargo, Jan. 2, 2013. [Online]. Available: https://vargosolutions.com/logistic-application-greedy-algorithms/ (accessed on Jan. 10, 2021).

[40] J. Le "Greedy algorithm and dynamic programming," *Experfy*, Nov. 5, 2018. [Online]. Available: https://www.experfy.com/blog/bigdata-cloud/greedy-algorithm-dynamic-programming/ (accessed on Jan. 10, 2021).

CHAPITRE 1

[1] A. Saranya, K. Rajasekaran, S. Chandra, "An overview of mobile ad hoc networks," *International journal of computer science and information technology research*, vol. 4, no. 3, pp 245-250, Jul.-Sept. 2016.

[2] C. Wang *et al.*, "Cellular architecture and key technologies for 5G wireless communication networks," in *IEEE Communications magazine*, vol. 52, no. 2, pp. 122-130, Feb. 2014, doi: 10.1109/MCOM.2014.6736752.

[3] J. Abdelaziz, H. Mcheick, "Toward a collaborative computing environment", *Conférence francophone de systèmes collaboratifs (SysCo)*, Sousse, Tunis, 28-30 Sept., 2012.

[4] B. C. Sherin, E.A. Mary Anita, "A Survey of scheduling algorithms for wireless ad-hoc networks," *International journal of advanced science and engineering*, vol.4, no. 4, pp. 776-787. 2018. [Online]. Available: www.mahendrapublications.com

[5] G. Fadlallah, H. Mcheick, Dj. Rebaine, "Layered architectural model for collaborative computing in peripheral autonomous networks of mobile devices," *The 16th International conference on mobile systems and pervasive computing (MobiSPC)*, Halifax, Canada, Aug. 19-21, 2019,

[6] A.J Ferrer, J.M. Marques, J Jorba, "Towards the decentralized cloud: survey on approaches and challenges for mobile, ad hoc, and edge computing," *ACM Computing survey*, vol. 51, no. 6, 2019.

[7] A. A. Mehanna, M. A. Abdel-Fattah, S. Abdel-Gaber, M.Cloudlet: "A mobile cloudlet model using Wi-Fi Direct," *International journal of computer science and information security (IJCSIS)*, vol. 14, no. 11, Nov. 2016.

[8] A. Mishra G. Masson, "MoCCA: A mobile cellular cloud architecture," *Journal of cyber security and mobility*, vol. 2, no. 2, pp. 105-125, 2013. [Online]. Available: https://doi.org/10.13052/jcsm2245-1439.221 .

[9] C. Casetti, C.F. Chiasserini, Y. Duan, P. Giaccone, A.P. Manriquez, “Data connectivity and smart group formation in Wi-Fi Direct multi-group networks,” *IEEE Transaction on network. service management*, vol. 15, no. 1, pp. 245-259, 2018.

[10] C. Funai, C. Tapparello, W. Heinzelman, “Enabling multi-hop ad hoc networks through Wi-Fi Direct multi-group networking,” In *Proceedings of the international conference on computing, networking and communications*, Santa Clara, CA, USA, pp. 491–497, 26–29 Jan. 2017.

[11] T. Oide, T. Abe, T. Suganuma, “Infrastructure-less communication platform for off-the-shelf android smartphones,” *Sensors*, vol. 8, no. 3, 2018.

ANNEXES 1

Certificat Éthique

UQAC

Comité d'éthique de la recherche
Université du Québec à Chicoutimi

APPROBATION ÉTHIQUE

Dans le cadre de l'*Énoncé de politique des trois conseils : éthique de la recherche avec des êtres humains 2* (2014) et conformément au mandat qui lui a été confié par la résolution CAD-7163 du Conseil d'administration de l'Université du Québec à Chicoutimi, approuvant la *Politique d'éthique de la recherche avec des êtres humains* de l'UQAC, le Comité d'éthique de la recherche avec des êtres humains de l'Université du Québec à Chicoutimi, à l'unanimité, délivre la présente approbation éthique puisque le projet de recherche mentionné ci-dessous rencontre les exigences en matière éthique et remplit les conditions d'approbation dudit Comité.

Les membres jugent que ce projet rencontre les critères d'une recherche à risque minimal.

Responsable(s) du projet de recherche :	*Monsieur Hamid Mcheick, Professeur* *Département d'informatique et de mathématiques, UQAC*
Cochercheur(s) :	*Monsieur Hicham Ajami, Étudiant* *Doctorat en sciences et technologies de l'information, UQAC*
Projet de recherche intitulé :	*Conception et adaptation des applications contextuelles.*
No référence du certificat :	*602.577.01*
Financement :	*Conseil de recherche en sciences naturelles et en génie du Canada (CRSNG) - Programme Subventions à la découverte (RGPIN-2017-05521)* *Titre : idem au projet de recherche*

La présente est valide jusqu'au 31 octobre 2018.
Rapport de statut attendu pour le <u>30 septembre 2018 (rapport final)</u>.
N.B. le rapport de statut est disponible à partir du lien suivant : *http://recherche.uqac.ca/rapport-de-statut/*

Date d'émission initiale de l'approbation : *10 octobre 2017*
Date(s) de renouvellement de l'approbation :

[redacted]

Tommy Chevrette,
Professeur et président du Comité d'éthique de la recherche avec des êtres humains de l'UQAC

ANNEXES 2

Renouvellement du certificat Éthique

UQAC
Université du Québec à Chicoutimi

Le 15 mars 2021

RENOUVELLEMENT DE L'APPROBATION ÉTHIQUE

La présente atteste que le projet de recherche décrit ci-dessous a fait l'objet d'un renouvellement de l'approbation éthique émise par le CER-UQAC et qu'il satisfait aux exigences de la politique de l'Université du Québec à Chicoutimi en matière d'éthique de la recherche avec des êtres humains.

Projet # : 2018-151, 602.577.01

Titre du projet de recherche: Conception et adaptation des applications contextuelles Titre pour le financement: idem

Chercheur principal à l'UQAC
Hamid Mcheick, (fonction introuvable)
département d'informatique et de mathématique, (organisation introuvable)

Cochercheur(s)
En provenance de l'UQAC: Hicham Ajami; Konan-Marcelin Kouame; Mohammad Saberi; Mohammad Mahmoud Hassan ENNAB; Ghassan Mustapha Fadlallah

Financement : CRSNG

Date de l'approbation éthique initiale du projet : 10 octobre 2017

Date du prochain renouvellement : 18 mars 2022.
N.B. Un rappel automatique vous sera envoyé par courriel quelques semaines avant l'échéance de votre certificat afin de remplir le formulaire F7 - Renouvellement annuel.

- *Si votre projet se termine avant la date du prochain renouvellement, vous devrez remplir le formulaire* ***F9 - Fin de projet****.*
- *Si des modifications sont apportées à votre projet avant l'échéance du certificat, vous devrez remplir le formulaire* ***F8 - Modification de projet****.*
- *Tout nouveau membre de votre équipe de recherche devra être déclaré au CER-UQAC lors de votre prochaine demande de renouvellement ou lors de la fin de votre projet si le renouvellement n'est pas requis. ATTENTION: Vous devez faire signer une déclaration d'honneur aux personnes ayant accès aux participants (ou à des données nominatives sur les participants) et la conserver dans vos dossiers de recherche.*
- Si vous avez des cochercheurs dans d'autres universités, veuillez leur transmettre ce certificat.
- *Si votre projet est financé, le Décanat de la recherche et de la création sera mis en copie conforme afin de l'informer du renouvellement de votre certification éthique.*

[redacted]

Stéphane Allaire

NAGANO Renouvellement - Approbation du CER-UQAC
Université du Québec à Chicoutimi - 555, boulevard de l'Université, Chicoutimi (Québec), G7H 2B1
1 / 1