Improving medium access for dynamic wireless sensor networks

Georgios Papadopoulos

To cite this version:

Georgios Papadopoulos. Improving medium access for dynamic wireless sensor networks. Networking and Internet Architecture [cs.NI]. Université de Strasbourg, 2015. English. NNT : 2015STRAD034 . tel-01338595

HAL Id: tel-01338595

https://theses.hal.science/tel-01338595v1

Submitted on 28 Jun 2016

UNIVERSITÉ DE STRASBOURG

ÉCOLE DOCTORALE MATHÉMATIQUES, SCIENCES DE L'INFORMATION ET DE L'INGÉNIEUR

Laboratoire ICube – UMR 7357

THÈSE présentée par :

Georgios PAPADOPOULOS

soutenue le : **28 septembre 2015**

pour obtenir le grade de : **Docteur de l'université de Strasbourg**

Discipline/ Spécialité : Informatique

Improving Medium Access for Dynamic Wireless Sensor Networks

THÈSE dirigée par :	
M. NOEL Thomas	Professeur, Université de Strasbourg
RAPPORTEURS :	
Mme. GUÉRIN-LASSOUS Isabelle	Professeur, Université Lyon I
M. DIAS DE AMORIM Marcelo	Directeur de Recherche, LIP6, Paris

---

AUTRES MEMBRES DU JURY :	
M. GALLAIS Antoine	Maître de conférences, Université de Strasbourg
M. WATTEYNE Thomas	Chargé de recherche, INRIA Paris-Rocquencourt

Georgios PAPADOPOULOS

Improving Medium Access for Dynamic Wireless Sensor Networks

Résumé

L'Internet des objets amène des contraintes uniques et une immense variété d'applications. Ceci oblige à être capable d'établir des communications efficaces en énergie (et néanmoins à faible délai) au sein de réseaux fortement dynamiques. Nous nous sommes concentrés sur l'amélioration du contrôle d'accès au medium (MAC), afin d'optimiser la gestion des communications sans fils, principale source de consommation d'énergie dans ces réseaux. Cette thèse discute de l'auto-adaptation de solutions MAC asynchrones et montre qu'une coopération localisée entre objets communicants permet de maintenir un partage efficace de la ressource de communication face à une forte dynamique (trafic, mobilité, pannes). Outre une réflexion menée sur les outils de simulation et d'expérimentation, nous avons conduit des campagnes d'évaluations complètes de nos contributions qui traitent tant des changements de trafic que de la mobilité dans les réseaux très denses.

Résumé en anglais

The Internet of Things brings unique constraints and a huge variety of applications. This forces to be able to establish energy efficient communications (and nevertheless low-delay) within highly dynamic networks. We focused on improving the medium access control (MAC) to optimize the management of wireless communications, the main source of energy consumption in these networks. This thesis discusses the self-adaptation of asynchronous MAC solutions and shows that a localized cooperation between communicating objects can maintain an efficient sharing of the communication resource in highly dynamic networks (traffic, mobility, failures). In addition to a reasoning on the tools of simulation and experimentation, we conducted comprehensive evaluation campaigns of our contributions that address traffic changes and mobility in dense networks.

Améliorations de l'accès au medium dans les réseaux dynamiques de capteurs sans fils

Georgios Z. Papadopoulos

Laboratoroire ICube
Université de Strasbourg, France
gpapadopoulos@unistra.fr

Introduction et Contexte

Les déploiements sans fil deviennent largement utilisés et fournissent aux utilisateurs (ou aux objets) un service d'accès à Internet presque partout, notamment en raison des avancées technologiques qui ont amené non seulement *smartphones* et tablettes PCs, mais aussi d'autres dispositifs à très faible encombrement. Par conséquent, au cours des dernières années, nous avons connu l'émergence d'un nouveau paradigme appelé Internet des Objets (*Internet of things* ou IoT) dans lequel les objets intelligents et connectés construisent des réseaux (sans fils) coopératifs. Ces objets peuvent être déployés ou disseminés presque partout, dans les maisons, les universités, les villes, les champs agricoles, dans la mer, jusque dans le corps humain ou toute autre chose naturelle ou artificielle.

Depuis 2008, le nombre d'objets reliés à l'Internet a ainsi dépassé le nombre de personnes vivant sur Terre. L'Internet des objets s'organise autour de communications de machines à machines toujours plus nombreuses. L'Internet des objets est une vision qui se construit aujourd'hui mais qui a émergé grâce à d'intenses activités de recherche et développement autour des réseaux *ad hoc* mobiles (*Mobile Ad Hoc Network*, MANET) et des réseaux de capteurs sans fils au cours de la dernière décennie.

Observation de l'environnement, télé-médecine, suivi et contrôle des procédés industriels, secours d'urgence, maisons et villes intelligentes sont des exemples d'applications des réseaux de capteurs sans fils qui nécessitent des équipements à faible coût, facilement déployés et multi-fonctionnels. Cela a conduit à une grande variété d'applications dont les nombreux déploiements réussis ont éclipsé les difficultés de gestion distribuée de la ressource commune qu'est l'air pour les communications radio entre objets hétérogènes.

Ainsi, les réseaux de capteurs sans fils absorbent de plus en plus l'attention de la communauté de la recherche et de nombreuses solutions ont été proposées. Elles concernent notamment l'optimisation de la performance du réseau en termes de durée de vie (i.e.,, les systèmes d'économie d'énergie), la collecte fiable de données, l'évitement de congestion ou encore la qualité de service pour l'utilisateur. Cependant, certaines questions cruciales restent sans réponse, en raison des contraintes spécifiques des nœuds capteurs et des liaisons sans fil.

Tout d'abord, les capteurs sont généralement alimentés par batterie. Il est difficile, voire impossible de les recharger au cours d'un déploiement dans des zones difficiles d'accès ou dangereuses. Il est crucial d'éviter le gaspillage inutile d'énergie pour chaque capteur. Cette question motive la communauté de recherche pour l'étude, la conception et le développement de protocoles de communication efficaces en énergie.

En outre, lors d'un déploiement, un réseau de capteurs sans fils peut présenter un certain nombre de défis inattendus tels que les pannes de noeuds, les difficiles conditions environnementales (e.g., volcan, forêt), ou encore les topologies instables (e.g., mobilité, variabilité de la qualité des liens de communication).

Pour surmonter ces problèmes, les protocoles de communication sont tenus de présenter un certain niveau de robustesse et de fiabilité. Par exemple, des protocoles de routage multi-chemins ou opportunistes sont étudiés afin de surmonter les défaillances de communication (e.g., panne de noeud, mauvaise qualité d'un ou plusieurs liens d'un chemin de routage). En outre, il serait idéal pour les protocoles conçus de pouvoir être évalués et vérifiés à la fois par simulation et au travers de campagnes d'expérimentation.

Un nœud capteur (sans fil)

Un capteur effectue des opérations telles que la détection ou le traitement des signaux analogiques issus de son environnement physique (e.g., lumière, température, humidité, mouvement, son, pression, vibration). Il intègre donc un convertisseur analogique-numérique (chargé de numériser le signal analogique continu produit par les capteurs) ainsi que d'autres composants tels qu'une mémoire externe et une source d'alimentation.

Un capteur sans fil est quant à lui constitué d'un micro-contrôleur (e.g., pour le traitement des données et de signalisation de protocole) et d'un émetteur-récepteur radio. Ces appareils, potentiellement énergivores et sujets aux pannes, sont considérés comme étant de très petite taille et permettent ainsi de recueillir des données et de les transmettre à d'autres nœuds grâce aux communications sans fil.

Les réseaux de capteurs sans fils

Les réseaux de capteurs sans fils sont considérés comme des réseaux à basse consommation d'énergie et sujets aux pertes (*Low-power and Lossy Networks (LLNs)*) car ils sont composés de beaucoup de noeuds limités en puissance, mémoire et ressources de traitement. Ainsi, ils devraient être déployés avec une forte redondance, amenant alors des problèmes de partage de ressource (e.g., collisions, interférences radio). Les applications types incluent notamment :

1. ***la surveillance militaire :*** parmi les applications militaires des réseaux de capteurs sans fils se trouvent la détection d'intrusion, la localisation d'ennemis et la détection d'attaque biologique ou chimique.

2. ***l'observation de l'environnement :*** ces applications incluent la surveillance de la faune, de la flore et de l'activité sismique, l'étude de microclimats, la détection des incendies de forêt ou encore la détection d'inondation.

3. ***la surveillance de la santé :*** des réseaux de capteurs corporels peuvent être utilisés pour recueillir des informations sur la santé du patient. Ces applications permettent par exemple d'évaluer la position du corps et l'emplacement de la personne ou encore d'assurer le suivi clinique des patients dans les hôpitaux et les maisons (e.g., rythme cardiaque, oxygénation du sang).

4. ***les applications urbaines et domestiques :*** on compte déjà des déploiements dans plusieurs villes à des fins de régulation de la circulation, de gestion du stationnement (e.g., systèmes SmartGrains et SFpark) ou de surveillance de la pollution de l'air. Les applications domestiques incluent la détection d'ouverture des portes, le contrôle des lumières ou de la température.

5. ***la surveillance industrielle et les*** *smart grids* ***:*** un certain nombre de procédés industriels nécessite la surveillance des produits (e.g., détection d'expiration, gestion de la qualité). Le terme *smart grid* désigne des applications où des capteurs (e.g., compteurs intelligents), disséminés dans le réseau de distribution électrique, permettent de mieux contrôler la production et l'acheminement d'énergie.

6. ***les applications agricoles :*** les déploiements de réseaux de capteurs sans fils dans les exploitations agricoles permettent de surveiller l'humidité, la température ou le pH du mélange eau-nutriments dans un champ de culture et ainsi assurer une maintenance plus efficace.

7. ***les secours d'urgence :*** incendies, séismes et autres catastrophes naturelles sont des situations dans lesquelles la recherche et le secours de victimes peuvent être améliorés grâce à la collecte d'informations en provenance directe du terrain.

L'Internet des objets a permis l'intégration des réseaux de capteurs sans fils à l'Internet, accélérant ainsi grandement leur déploiement au service de nombreuses applications.

Communications dans les réseaux de capteurs sans fils

Pour assurer les fonctions nécessaires aux applications précédemment citées, un grand nombre de capteurs autonomes peuvent être disséminés sur une zone dont ils surveillent les conditions physiques ou environnementales. Ceci passe par la coopération (transmission et réception des mesures d'un capteur mais également de celles des autres capteurs avec lesquels il peut communiquer) afin d'acheminer les données collectées vers une ou plusieurs stations passerelles, appelées puits (i.e., *sinks*). Un puits rassemble ainsi les données et assure la connectivité vers l'Internet, tout en traitant les mesures reçues. Celles-ci sont générées en fonction des exigences de l'application finale. Les déploiements observés s'appuient sur un ou plusieurs paradigmes de communications suivant que les émissions de données soient :

- guidées par le temps (*time-driven*), avec des paquets de données émis à intervalles de temps constants;
- guidées par les événements (*event-driven*), lorsque les transmissions se produisent uniquement en cas de détection d'un événement spécifique, induisant par conséquent un trafic de données dynamique;
- guidées par les requêtes (*query-driven*) si les émissions de données se font en réponse à des demandes issues de l'utilisateur final.

Ces différents paradigmes de communication induisent différents types de trafic au sein du réseau (e.g., constant, sporadique, en rafale).

En outre, en raison des contraintes énergétiques, les radios fonctionnent à des puissances d'émission limitées, imposant ainsi des rayons de communication réduits. Les communications multi-sauts sont alors utilisées pour couvrir de grandes surfaces, potentiellement difficiles à atteindre et d'où les mesures effectuées par les capteurs déployés doivent ensuite atteindre l'une des stations puits. L'organisation de ces communications induit des topologies variées (e.g., ad-hoc, en clusters, centralisées).

Dès lors, on distingue trois types de trafic :

i) multipoint-à-point (ou *convergecast*) où plusieurs capteurs transmettent leurs données en direction des puits (schéma induit en *time-driven* et *event-driven*) ;

ii) point-à-multipoint lorsque le puits envoie des requêtes à un ou plusieurs capteurs du réseau (schéma induit en *query-driven*) ;

iii) Point-à-point lors d'une communication simple de capteur à capteur (ou puits), par exemple, lors de la détection d'événements, ou pour la communication entre nœuds au sein du réseau (schéma induit en *event-driven*) ;.

Motivations et hypothèses

Comme observé à partir des applications actuelles de réseaux de capteurs sans fils, les nœuds sont souvent déployés dans des environnements difficiles à atteindre (e.g., forêts, montagnes, volcans) et doivent opérer de façon autonome dans des densités de communication élevées. Il est alors essentiel de garantir que les systèmes seront toujours alimentés. Par conséquent, l'efficacité énergétique constitue la principale préoccupation dans la perspective de durées de vie étendues pour ces réseaux d'observation ou de surveillance. Ceci a mené à la conception et au développement de nouveaux protocoles dans toutes les couches de la pile de communication.

Dans ce contexte, la principale source de consommation d'énergie (i.e., communications sans fils) est régie par la couche de contrôle d'accès au medium (*medium access control*, MAC). Lors des transmissions, les sources de consommation d'énergie peuvent être classées comme suit :

i) des nœuds en écoute passive scrutent le medium en attente de potentielles réceptions à traiter, ou encore traitent des communications ne leur étant pas destinées ;

ii) deux noeuds ou plus (en cas de chevauchement des zones de communication) tentent simultanément de transmettre un paquet, ce qui résulte en une collision puis en des retransmissions coûteuses ;

iii) enfin, selon le protocole de routage ou de contrôle d'accès au medium, les nœuds doivent transmettre un paquet de contrôle (qui génère des surcoûts en communication) pour pallier d'éventuels problèmes de synchronisation.

Ces aspects relèvent de la responsabilité de la couche MAC, dont l'optimisation participe par conséquent grandement à la réduction de la consommation d'énergie. En effet, la couche MAC s'occupe également d'alterner la radio entre périodes d'activité (transmission et de réception d'un paquet) et de passivité (mode veille). Deux noeuds peuvent communiquer entre eux lorsqu'ils actifs simultanément. Cette alternance de cycles induit un compromis fondamental entre la réduction de la consommation d'énergie (pour maximiser la durée de vie du réseau) et les performances (e.g., latence, débit au sien du réseau). Un protocole MAC annonçant une alternance de 1% signifie que les nœuds ne sont en mode actif que pendant 1% du temps.

Dans ce manuscrit, nous considérons les protocoles MAC avec contention. Dans cette famille de protocoles, les noeuds échantillonnent de façon asynchrone le canal de communication à intervalles réguliers afin de détecter d'éventuels paquets entrants. Entre deux échantillonnages, ils passent en mode veille afin d'économiser leur énergie.

En outre, dans de nombreuses applications (e.g., surveillance de patients, contrôle industriel, observation de la faune), des exigences telles que la mobilité des noeuds et le trafic de données en rafales sont incontournables. Par exemple, dans de telles applications, les noeuds mobiles peuvent être déconnectés de l'infrastructure de communication pendant la majorité du temps, et doivent ainsi rapidement transmettre leurs mesures (i.e., trafic en rafale) lorsque des liens de communication apparaissent vers le(s) puits. Ces circonstances soudaines et non prédictibles de trafic en rafales provoquent certaines anomalies dans le réseau et justifient l'étude de solutions appropriées au niveau MAC.

Dans cette thèse, nous avons donc considéré les protocoles MAC avec contention, afin de gérer les changements de charge de trafic (e.g., décisions locales) et de s'adapter face à des topologies réseau dynamique (e.g., mobilité des capteurs). Plus précisément, nous étudions la famille de protocoles MAC asynchrones, dans lesquels les noeuds échantillonnent de façon asynchrone le canal de communication à intervalles réguliers afin de détecter d'éventuels paquets entrants.

De nombreuses solutions économes en énergie et auto-adaptables au trafic ont été proposées dans la littérature. Cependant, très peu d'entre elles répondent aux besoins induits par le trafic dynamique observé dans des réseaux intégrant des capteurs mobiles. A notre état actuel de connaissance, aucun déploiement de réseaux de capteurs sans fils ne met en place une gestion de la mobilité et de l'auto-adaptation au trafic dynamique.

En outre, compte tenu des caractéristiques des déploiements réalisés au cours des 10 dernières années, nous nous sommes concentrés sur des réseaux multi-sauts, avec un trafic de données multipoint-à-point. Enfin, dans nos procédures d'évaluation de performances, nous avons considéré à la fois les communications guidées par le temps et par l'événement.

Contributions

Cette thèse s'articule autour de l'optimisation des protocoles de contrôle d'accès au medium pour les réseaux de capteurs sans fils dans des conditions de trafic et de topologie dynamiques. Nous souhaitons améliorer la performance globale du réseau (i.e., délai, fiabilité) tout en réduisant la consommation d'énergie. Dans cette optique, nous commençons par estimer le volume de trafic entrant afin d'adapter les paramètres essentiels au niveau MAC avant de nous concentrer sur les communications issues des nœuds mobiles. Toutes les contributions proposées dans cette thèse visent à atteindre les objectifs présentés précédemment :

> ***Afin de débuter nos recherches, nous commençons par caractériser un banc d'essai pour l'Internet des objets.*** Nous explorons le rôle des bancs d'essai dans la conception et le développement de protocoles ou applications pour l'Internet des objets en général et les réseaux de capteurs en particulier. Plus précisément, nous montrons dans quelle mesure l'ajout d'expérimentations améliore significativement la valeur des campagnes d'évaluation des performances. Nous discutons de la capacité à produire des résultats scientifiques et non des preuves de concept uniquement à partir de bancs d'essai ouverts. Dans ce but, nous analysons la reproduction des conditions de déploiements réels.
>
> ***Nous présentons ensuite T-AAD, un algorithme qui s'adapte rapidement aux changements de trafic.*** Il permet la configuration automatique et à la volée des paramètres de la couche MAC. T-AAD est compatible avec tous les protocoles MAC dits à échantillonnage de préambule. Ceci permet une consommation d'énergie réduite pour le récepteur et l'expéditeur tout en réduisant les délais et temps d'occupation du canal, en comparaison aux solutions de la littérature. Nous effectuons une évaluation de performance approfondie, à la fois par simulations et par expérimentations sur le banc d'essai FIT IoT-LAB.
>
> ***Nous proposons ensuite M-ContikiMAC, ME-ContikiMAC et MobiXplore, de nouveaux protocoles de niveau MAC permettant la gestion de la mobilité dans les réseaux très denses.*** M-ContikiMAC étend le protocole ContikiMAC pour permettre la communication de nœud mobile à statique. Il se base sur des transmissions *anycast* qui peuvent engendrer des anomalies réseau, comme des taux élevés de duplication de paquets par exemple. Nous analysons et étudions alors comment mettre en place la co-existence et la coopération des nœuds mobiles avec les nœuds statiques du réseau, tout en mettant en évidence les limitations restantes de M-ContikiMAC. Nous proposons ensuite ME-ContikiMAC où des mécanismes de contrôle des duplications et de réduction des délais sont mis en place. La réduction des duplications permet de limiter le délai à un saut, et donc le temps d'occupation du canal et la consommation d'énergie, en comparaison avec M-ContikiMAC. Nous évaluons cette contribution face à d'autres solutions

récentes et reconnues par la communauté. Enfin, nous proposons MobiXplore, un protocole MAC permettant un relais transparent des communications pour un noeud en situation de mobilité et dont les prochains sauts de routage ne cessent de changer. Toutes les propositions mentionnées précédemment sont conformes aux principes utilisés dans les couches MAC asynchrones et peuvent donc s'adapter aisément aux solutions existantes voire déjà déployées.

Enfin, nous proposons un mécanisme auto-adaptatif qui atténue les duplications de paquets lors d'utilisation de routage opportuniste. Nous proposons un mécanisme qui gère la surdité dans le réseau par l'intermédiaire de configurations MAC hétérogènes parmi les noeuds dans le réseau. Nous utilisons des décisions locales prises par chaque nœud de façon décentralisée.

Les nœuds adaptent alors leurs paramètres MAC de façon dynamique et automatique afin de réduire le trafic inutile, l'occupation du canal de communication et la consommation d'énergie.

Conclusion

L'objectif de cette thèse était de répondre à certaines questions clés posées par les protocoles de contrôle d'accès au médium (MAC) dans les réseaux de capteurs sans fils. Parmi elles, nous pouvons citer la considération d'environnements contraints, des trafics et topologies dynamiques, ainsi que l'amélioration des performances globales de réseaux dans lesquels des transmissions variables et en rafales se produisent.

La couche MAC devant gérer les communications entre capteurs sans fil, elle apparaît comme le premier levier pour la réduction d'une consommation d'énergie essentiellement induite par les opérations de transmission, réception et traitement des données. Nous nous sommes par conséquent concentrés sur l'amélioration de l'accès au medium sans fil pour permettre des communications efficaces en énergie et à faible délai.

Dans ce manuscrit, nous avons favorisé les approches asynchrones plutôt que des méthodes dites synchronisées, principalement en raison de l'efficacité et de la tolérance des protocoles à échantillonnage de préambule dans des réseaux à grande échelle et aux topologies dynamiques. Nous avons ainsi pu considérer des topologies réseaux changeantes (e.g., noeuds mobiles) et certains facteurs d'échelle grâce à la coopération efficace et localisée entre capteurs. Outre une réflexion menée sur les outils de simulation et d'expérimentation, nous avons mené des campagnes d'évaluations complètes qui ont permis de montrer les améliorations apportées par nos contributions.

École doctorale MSII

Improving Medium Access in Dynamic Wireless Sensor Networks

THÈSE

présentée pour l'obtention du

Doctorat de l'Université de Strasbourg
(mention informatique)

par

Georgios Z. PAPADOPOULOS

Composition du jury

Directeur de thèse :	Thomas NOEL, Prof., Université de Strasbourg, France
Co-encadrant de thèse :	Antoine GALLAIS, associate Prof., Université de Strasbourg, France
Rapporteurs :	Isabelle GUÉRIN-LASSOUS, Prof., Université Lyon I, France Marcelo DIAS DE AMORIM, CNRS research director, LIP6, Paris, France
Examinateur :	Thomas WATTEYNE, research scientist, INRIA Paris-Rocquencourt, France

Laboratoire des sciences de l'ingénieur, de l'informatique et de l'imagerie
ICube — UMR 7357

Αφιερωμένο σε όσους πιστεύουν, με υπομονή και επιμονή ...

Acknowledgments

The current thesis is the result of three intensive years of exciting work at the ICube laboratory of the University of Strasbourg in France. First and foremost, I would like to thank Dr. Antoine Gallais and Prof. Thomas Noel for believing in my potential and for giving me the necessary freedom in my research. Thomas, thank you very much for your confidence in my work and decisions. Antoine, I find myself lucky to have had you as my advisor. Thank you for your guidance, patience, constant support and, most importantly, your friendship during these years. I will always remember our long (very long) discussions at the campuses of Illkirch and Esplanade.

I would also want to thank Prof. Isabelle Guérin-Lassous, Prof. Marcelo Dias de Amorim, and Dr. Thomas Watteyne for accepting to be part of my jury, for their questions, useful suggestions and constructive comments.

A big thanks to all my colleagues from the Réseaux team, Oana, Pascal, Guillaume, Fabrice, Julien B., Cosmin, Jean-Jacques, François, Thomas H., Stephan, Philippe, Keoma, Damien, Julien M., Erkan and Pierre. Thank you all for your company during these three years.

Pascal, thank you for your advices that helped me to become a better teacher and, more importantly, for your friendship. Guillaume, thank you very much for your enormous availability, constant technical support, collaboration and co-authorship. Fabrice, thank you for your precious advices, our discussions and support.

Special thanks go also to all my collaborators and co-authors (and friends) Vassilis, Periklis, Julien B., Guillaume, Kosmas and Benjamin. Thank you for your trust and collaboration.

A warm thank to Oana, Julien B. and Cosmin, for our long discussions and (most important) for your friendship.

I deeply like to thank Periklis Chatzimisios, for providing me with uncountable wise advices during the very interesting and long discussions we had, and for his collaboration in my research activity.

Jo, thank you very much for your friendship, support and for sharing with me all these great (memories) moments.

I would like to thank very much my Strasbourg's friends, Eratoula, Asimina, Eleni C., Christos T., Eugenia and Alkis for making this period of my life an experience that I will never forget. I would also want to thank my friends, Vladimiros, Sotiris and Kosmas for being there for me when I needed them.

Eugenia, thank you for your support and for encouraging me to follow my dreams. Thank you for your advices, both personal and professional. Most of all, thank you for your love and care.

This thesis would not have been possible without the help of three persons. I thank my parents, Zourab and Larisa, and my sister, Maria, for their wholehearted support in my decisions, care, love and belief in me throughout my entire life. Thank you for pushing me to be a better person.

Last but not least, I want to thank Alex for the uncountable great evenings of nice company and music at Phonograph.

Abstract

Ad-Hoc and Wireless Sensor Networks (WSNs) have enabled a large variety of applications. Environmental and wildlife monitoring, clinical medical and home-care monitoring, monitoring and control of industrial processes including agriculture, smart houses and cities are just some of the examples of Ad-Hoc and WSN applications, where low-cost, and easily deployed multi-functional sensor nodes is the ideal solution. As a result, during the last years we experience the emergence of a new paradigm called Internet of Things (IoT) in which smart and connected objects cooperatively construct a (wireless) network of things.

However, the unique features of these technologies can pose significant challenges. In clinical medical and home-care applications, requirements such as mobility, bursty traffic and energy-efficiency are appear to be essential. In contrary to the traditional a priori known time-driven traffic patterns, event-driven, where nodes transmit their readings upon detection of a specific event determined by the application layer, networks face occasional, bursty and unanticipated multi-hop data packet transmissions. In wildlife monitoring for instance, the nodes (usually with limited-memory devices) operate under limited internet access for the majority of the time. When a network connection is detected, a surge of traffic should be handled. More specifically, the mobile wireless nodes should immediately upload their stored readings (bursts) at a more powerful device (i.e. sink) before losing again the connection. Such sudden dynamic and bursty traffic cause certain anomalies in the network and fuel the research community to find appropriate solutions.

Since, the Medium Access Control (MAC) layer is in charge for coordinating the communication between wireless sensor nodes. Furthermore, among all operations of a sensor node, Transmission, Reception, Central Processing Unit and Low-power-mode, the communication is the most energy consumed. We therefore, in this dissertation, have focused on improving the access to the wireless medium for low-delay communication in energy efficient manner.

My thesis shows that these competing goals can be balanced by the use of effective algorithms and schemes that enable the protocols (and hence the network) to adapt to current application (i.e. varying traffic load) and network conditions (i.e. mobility). In support of this dissertation, we first have ***i***) studied the role of simulators/emulators and testbeds in the research process cycle, and we identified the means to strengthen their complementarity, ***ii***) designed, developed and evaluated an algorithm that dynamically and automatically reconfigures the MAC parameters depending on the actual and expected traffic load, ***iii***) proposed MAC layer protocol to coordinate the communication between mobile and static nodes even in very dense networks, ***iv***) finally, we demonstrated the advantages of employing low-power MAC protocols in a WSN in terms of latency, reliability, energy consumption and congestion in the network. With mobility-oriented preamble-sampling schemes provided by our new architecture, and the protocol adaptation provided in this dissertation, one can envision new designs at all protocol levels, making wireless sensor networks truly adaptive to changes in both application requirements and network dynamics.

List of publications

The contributions of this thesis have been published or submitted in peer-reviewed international and national journals and conferences.

International Journals and Magazines

- *"Performance Evaluation Methods in Ad-Hoc and Wireless Sensor Networks: A Literature Study"*,
  G. Z. Papadopoulos, K. Kritsis, A. Gallais, P. Chatzimisios and T. Noël,
  In IEEE **Communications Magazine**.

- *"Wireless Medium Access Control under Mobility and Bursty Traffic Assumptions in WSNs"*,
  G. Z. Papadopoulos, V. Kotsiou, A. Gallais, P. Chatzimisios and T. Noël,
  In Springer **Mobile Networks & Applications**.

International Conferences and Workshops

- *"Optimizing the Handover Delay in Mobile WSNs"*,
  G. Z. Papadopoulos, V. Kotsiou, A. Gallais, P. Chatzimisios and T. Noël,
  In IEEE World Forum on Internet of Things (**WF-IoT**) - Milan, Italy.

- *"Live Adaptations of Low-power MAC Protocols"*,
  G. Z. Papadopoulos, A. Gallais, G. Schreiner and T. Noël,
  In ACM Wireless of the Students, by the Students, and for the Students Workshop (**S3**) - Paris, France.

- *"Toward a Packet Duplication Control for Opportunistic Routing in WSNs"*,
  G. Z. Papadopoulos, J. Beaudaux, A. Gallais, P. Chatzimisios and T. Noël,
  In IEEE Global Communications Conference (**GLOBECOM**) - Austin, TX USA.

- *"Enhancing ContikiMAC for Bursty Traffic in Mobile Sensor Networks"*,
  Best Student Paper Award @ Sensor Networks Track,
  G. Z. Papadopoulos, A. Gallais, T. Noël, V. Kotsiou and P. Chatzimisios,
  In IEEE **SENSORS** - Valencia, Spain.

- *"T-AAD: Lightweight Traffic Auto-ADaptations for Low-power MAC Protocols"*,
  Best Paper Award,
  G. Z. Papadopoulos, J. Beaudaux, A. Gallais and T. Noël,
  In IEEE / IFIP Annual Mediterranean Ad Hoc Networking Workshop (**Med-Hoc-Net**) - Piran, Slovenia.

- *"Adding value to WSN simulation using the IoT-LAB experimental platform"*,
  G. Z. Papadopoulos, J. Beaudaux, A. Gallais, Thomas Noël and G. Schreiner,
  In IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (**WiMob**) - Lyon, France.

Demonstrations and Posters in International Conferences

- *"Demo Abstract: Live Adaptations of Low-power MAC Protocols"*,
  G. Z. Papadopoulos, A. Gallais, G. Schreiner and T. Noël,
  Demo abstract in ACM Annual International Conference on Mobile Computing and Networking (**MobiCom**) - Paris, France.

National Conferences

- *"Auto-configuration Proactive de la Couche MAC pour les Réseaux de Capteurs Sans Fil à Trafic Variable"*,
  J. Beaudaux, G. Z. Papadopoulos, A. Gallais and T. Noël,
  Journées Francophones Mobilité et Ubiquité (**UbiMob**) - Nancy, France.

Contents

Chapter 1

Introduction and Context

Wireless deployments are becoming broadly used and provide users (or things) with internet access service almost everywhere, due to the advance of technology that now provides not only smartphones and tablet PCs, but also small and uniquely identifiable devices. Therefore, during the last years we have experienced the emergence of a new paradigm called Internet of Things (IoT) in which smart and connected objects cooperatively construct a (wireless) network of things. Those things can be deployed or embedded nearly everywhere, at homes, universities, cities, agricultural fields, in the sea, even in the human body or any other natural or man-made object.

As a result, in 2008, the number of devices (things) connected to the Internet exceeded the number of people on Earth [Cis]. The IoT revolves around increased Machine-to-Machine (M2M) communication. In fact, IoT is a vision and is being built today.

IoT has emerged thanks to numerous works and intense research and development activities that took place around Mobile Ad-Hoc Network (MANET) and Wireless Sensor Network (WSN) since the previous decade.

Environmental (e.g., in coal mines [LL07]) and wildlife monitoring [Dyo+10], clinical medical and home-care monitoring [Chi+10a], monitoring and control of industrial processes including agriculture [Iov14], military surveillance [Sim+04], emergency rescue [Sha+06], smart houses and cities are just some of the examples of WSNs applications, where low-cost, and easily deployed multi-functional sensor nodes are considered as the ideal solution [Yic+08]. This has led to a large variety of applications whose successful deployments have outshone the difficulties of sharing a common resource (air) among many heterogeneous objects, in a distributed manner.

Thus, WSNs absorb more and more attention from the research community due to their wide range of applications. Over the years, many new solutions for sensor networks have been proposed. Contributions have been mainly focused on optimizing the network performance in terms of lifetime (i.e., energy saving schemes), reliable data collection, congestion avoidance, or quality of service for the user.

However, some critical issues have remained unsolved, due to the specific constraints of the sensor nodes or the wireless links. First of all, the sensor nodes are usually battery-powered, and it is often difficult or impractical to recharge or change the batteries in case they are deployed in hard-to-reach areas [Chi+07] or in unsafe environments [RM04]. Therefore, it is crucial to avoid the unnecessary energy wastage at each sensor. This issue motivates the research community to investigate, design and develop energy-efficient schemes and protocols.

Furthermore, during the experiment, WSNs may present a number of unexpected challenges such as node crashes, heavy environmental conditions (e.g., volcano, forest), or unstable topologies (e.g., mobility, unreliable communication links). To overcome these issues, WSN should be fault-tolerant to those circumstances and communication protocols are required to present a certain level of robustness and reproducibility. For example, multi-path or opportunistic routing

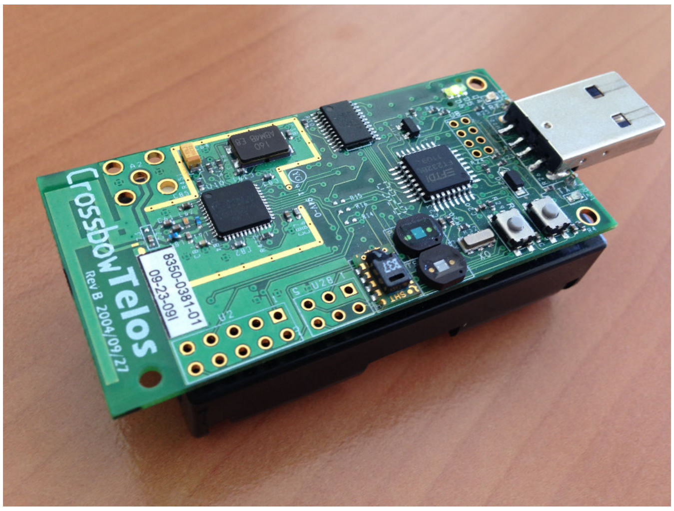


Figure 1.1: A typical sensor node:
Telos ultra-low power wireless module ("mote") with IEEE 802.15.4 wireless transceiver.

protocols are studied in order to overcome communication failures (e.g., node crashing, poor link quality in the path). Moreover, it would be ideal for the designed protocols to be evaluated and verified with both simulation and experimentation campaigns.

1.1 (Wireless) Sensor Node

A sensor device, also known as node or mote, performs operations such as sensing, detecting or responding to analogical inputs of its physical environment. These sensed inputs could be the light, the temperature, the humidity, the motion, the sound, the pressure, the vibration, or any other environmental phenomena. Such a device therefore embeds an Analog-to-Digital Converter (ADC) that digitizes the continual analog signal produced by the sensors. Other main components include an external memory and a power source, usually a battery.

A sensor node is made wireless once a micro-controller (e.g., for data processing and protocol signaling) and a wireless radio transceiver are added. As observed on Figure 1.1, such wireless devices are an extremely energy harvesting and prone to failure. They can be very small in size and thus provide a lightweight and portable detection station that can gather data and communicate with other wireless nodes, in order to transmit its readings.

1.2 Wireless Sensor Networks

Multiple wireless sensor nodes can form a network. WSNs are considered as Low-power and Lossy Networks (LLNs) since they are composed of many embedded nodes with limited power, memory, and processing resources. Thus, they are expected to be densely deployed, which leads to multi-hop interference and time-varying radio link quality. The devices in LLNs communicate by using different types of wireless links, such as the Institute of Electrical and Electronics Engineers (IEEE) 802.15.4 standards [802b].

WSNs are absorbing more and more attention from the research community due to their low cost, small in size, and availability. As a result, hundreds and thousands of such devices are interconnected, which leads to the integration of the WSNs in the emerging IoT.

Typical applications of WSNs include (but are not limited to) [Iov14]:

(1) Military surveillance. WSNs may support efficiently to achieve effective battlefield situational awareness. Some military examples include the use of sensor nodes to detect enemy intrusion, shooters' localization [Sim+04] or nuclear, biological, and chemical attack detection.

(2) Environmental applications: WSNs have found varied applications in the area of habitat and environmental monitoring. Some of the environmental applications are wildlife monitoring [Zha+04], monitoring seismic activity [WA+08], microclimate study of a tree [Tol+05], humidity and temperature monitoring [Lan+06], forest fire detection, flood detection, etc.

(3) Health monitoring: Body sensor networks can be used in mobile health monitoring to collect information about patient's health [Jon+10]. There are many WSN-health applications such as body position measurement and location of the person, clinical monitoring of patients in hospitals and homes (e.g., heart rate and blood oxygenation) [Chi+10b].

(4) Urban and home applications: WSNs have been deployed in several cities for traffic, parking or air pollution monitoring purposes, or at home for light switching, doors detection and temperature adjustment issues.

(5) Industrial monitoring and smart grid: There are number of applications that require monitoring of products (e.g., expired detection, quality). Another example of the possible use of WSNs is to prevent repeats of the contamination of the food supply chain. Furthermore, WSNs are used in smart grid, where the sensors are called smart meters and may use Power-Line Communication for energy measurement and control (e.g., [Tri]).

(6) Terrestrial applications: These type of applications are probably the most common ones, typically deployed in the fields for agricultural purposes [Iov14]. For instance, WSNs can be used to monitor the temperature, moisture, or the PH of water-nutrient mixture in a crop field to enable more efficient maintenance.

(7) Emergency rescue. WSNs are used in emergency search and rescue cases such as fire [Sha+06] and earthquake [KL]. For instance, a sensor network (e.g., temperature or humidity measurement sensors) can be installed in a forest to detect when and where a fire has started. The early detection is essential for the successful reaction of the firefighters.

The Internet of Things paradigm made WSNs connected to the Internet, thus speeding up their deployments and utilization on a daily basis in urban environments, e.g., parking systems like SmartGrains [Sma] and SFpark [Sfp].

1.3 Communications in WSN

Most of the previously exposed application domains rely on WSNs that consist in a large number of spatially distributed autonomous sensor nodes. They allow to monitor physical or environmental conditions and by cooperatively passing (transmit, receive and forward measurements from other nodes) their readings through the network to one or more gateways (i.e., *sinks*) [Aky+02a], [Ver+08]. A typical sink station gathers the data from the sensor nodes, connects the WSN to the Internet, and processes the received measurements locally. Those data are generated according to the end application requirements. In a WSN, there are three types of communication paradigm:

- Time-driven where data packets are transmitted to sink in constant time intervals;
- Event-driven where data packets transmission occurs only when a detection of a specific event takes place, in this case we may observe dynamic traffic as well;
- Query-driven where data packets are collected according to end user's demand.

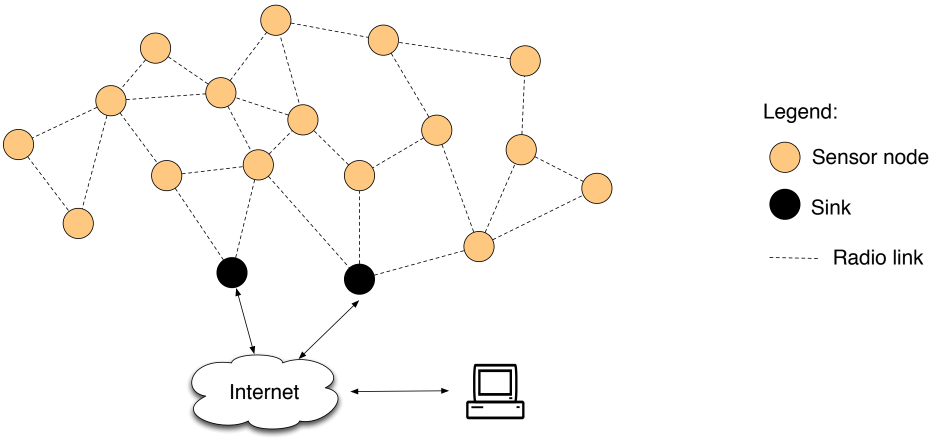


Figure 1.2: Overview of a Wireless Sensor Network. Data collected by sensor nodes are forwarded to a sink station in a multi-hop manner.

However, we may experience deployments with hybrid paradigm of communication [Zha+04], [Dyo+10]. These different communication paradigms can lead to different kinds of traffic in the network, such as Constant Bit Rate (CBR) or non constant (e.g., burst) ones.

Furthermore, due to energy constraints, radios operate under limited transmission powers thus imposing reduced communication ranges. Multi-hop communications are therefore required to cover large and hard-to-reach deployment areas where all measured data must be able to reach one of the sink stations (as illustrated in Figure 1.2). The sensor nodes may establish a network with various topologies, such as ad-hoc networks, clustered networks, or centralized networks.

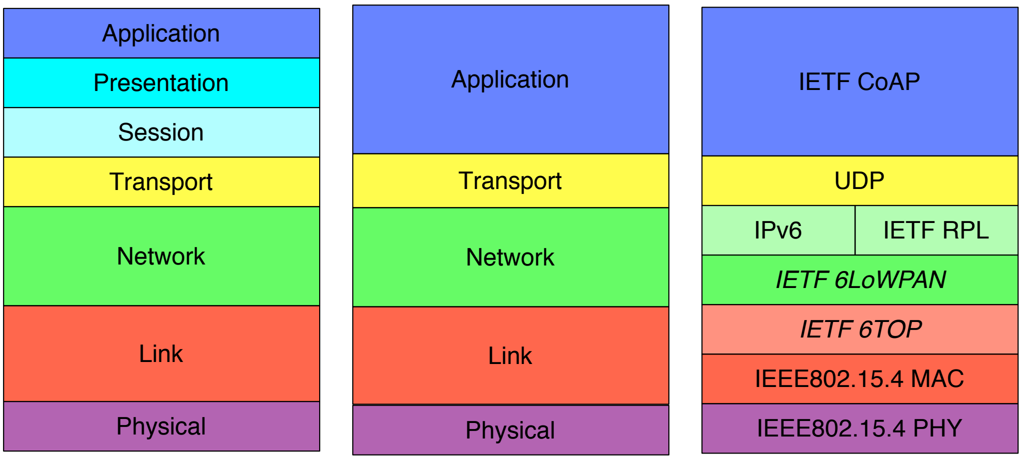
Finally, in a WSN, we can face three types of traffic, *i)* multipoint-to-point (induced by time-driven and event-driven communication paradigms), also called convergecast: the devices sense the environment and send the information back to the sink *ii)* point-to-multipoint (induced by query-driven communication paradigm): used by the sink to send commands to one or more devices in the network *iii)* point-to-point (induced by event-driven communication paradigm): sensor-to-sink communication for instance upon event detection, or for communication between nodes inside the network, where data do not necessarily go towards the sink (e.g., in-network processing, fault-tolerance schemes).

1.4 The Architecture of Wireless Sensor Networks

In order to support such communication paradigms and induced traffic patterns and while being compliant with existing communication networks, numerous works have designed the architecture of typical interconnected sensor nodes. In [Aky+02b], Akyildiz *et al.* introduced one of the first protocol stacks for WSNs (see Figure 1.3b). In fact, they envisioned it based on the seven-layer architecture of Open Systems Interconnection (OSI) model (Figure 1.3a), where the session and presentation layers were not included, since they have not been identified as useful in this type of technology.

Thus, we may consider five main levels:

- **Application layer** that defines a standard set of services and applications;
- **Transport layer** which is responsible for reliable data packet delivery required by the application layer;
- **Network layer** which is responsible for routing the data from the transport layer and directing the process of selecting paths in the network;



(a) OSI Stack

(b) WSN Stack (2001) (Reproduced from [Aky+02b])

(c) Standardized Stack (Reproduced from [AP14])

Figure 1.3: WSN Protocol Stacks

- **Data Link layer** which is responsible for data frame transmission and reception, multiplexing data streams, medium access, and error control;

- **Physical layer** which is responsible for signal transmission and reception over a physical communication medium, including frequency and power selection, signal modulation and data encryption.

Due to the constraints that WSNs present, cross-layer approach has been shown to be more efficient [RI04]. Since the emergence of new and heavy applications in WSNs, the proposed architecture was not well suited. Indeed, it is difficult to maintain the separation between layers during the optimization procedure at the network performance or energy saving.

The IEEE and Internet Engineering Task Force (IETF) Working Groups (WGs) have standardized several protocols for the LLNs, and therefore, for WSNs. Several adaptation layers had to be established by the scientists and engineers, in order for the different standardized protocols to effectively and efficiently interact together. To this aim, by enabling IPv6 over Low power Wireless Personal Area Networks, 6LoWPAN adaptation layer had to be madden in order to provide Internet connectivity to WSN [Mon+07]. Moreover, compression schemes, such as LOWPAN_IPHC Encoding [HT11] were introduced in order to compress the IPv6 header into 2 bytes.

The IETF ROLL WG concluded that existing well-known routing protocols such as Open Shortest Path First (OSPF), Intermediate System to Intermediate System (IS-IS), Ad Hoc On Demand Vector (AODV), and Optimized Link State Routing (OLSR) do not satisfy the specific routing requirements that characterize LLNs, such as multi-hop network, complex traffic patterns, energy-efficiency. As a result, the ROLL WG developed and standardized the IPv6 Routing Protocol for Low-power and Lossy Networks (RPL) [Win+12] to manage networks constructed with up to thousands of devices.

6TOP sublayer [Wan+14], developed by the IETF 6TiSCH working group, which allows interactions between the link layer and the upper layers in the stack to address industrial needs (e.g., applications such as smart metering) with safe slot allocation at the Medium Access Control (MAC) layer, scheduling from the end application. Furthermore, Constrained Application Protocol (CoAP) framework, an application layer protocol, emerged from the CoRE WG in order to manage the limited resources of a device.

The IEEE 802.15.4 family of standards [802b] was adopted to define the Physical (PHY) and MAC layers, which were first introduced in Personal Area Networks (PANs). The first version of the standard was presented in 2003 and was followed by a revision in 2006 and 2011. The Maximum Transmission Unit (MTU) of IEEE 802.15.4 standard is limited to 127 bytes, far smaller than the 1500 bytes supported by Ethernet or WiFi. Since, the single-channel of IEEE 802.15.4 MAC presents reliability issues, and even more, the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) access technique causes overhearing, interference and idle-listening [Pal+13]. The IEEE802.15.4e Timeslotted Channel Hopping (TSCH) MAC amendment has been designed and developed to meet the previously pitfalls [802a].

All these efforts in standardization led to design a new protocol stack for WSNs as we can see in Figure 1.3c.

1.5 Motivation and Assumptions

As exposed throughout this introduction chapter, WSNs present a number of challenges (e.g., energy supplying, network topology), that make the research in this field demanding and exciting.

As observed from current WSN applications (see Section 1.2), wireless sensor nodes are often placed and deployed in a hard-to-reach environments such as forest, mountains, volcanoes, animals, and even more they operate autonomously in high volumetric densities. In those locations, changing or recharging the batteries regularly can be costly, inconvenient, impractical, or even impossible. Thus, it is essential to guarantee that there will be always an adequate energy available to power the system, during the development of a WSN deployment. Hence, whatever the envisioned application, energy-efficiency is the primary concern in WSNs, since the sensor devices are battery-powered and they have to last for long duration [Dyo+10]. Thus, it is critical to investigate energy-saving techniques in order to extend the network lifetime of a WSN experiment. Indeed, the energy supply limitation has prompted the design and development of new protocols in all layers of the communication stack.

As already discussed in Section 1.4, the MAC layer is responsible for controlling the main source of energy consumption, i.e., medium access for wireless transmissions. Indeed, among all operations of a typical wireless sensor node, transmission, reception, Central Processing Unit (CPU) and Low-Power Mode (LPM), the communication is the most energy consumed [Aky+02b], [Eri+09]. The different energy consumption sources can be classified as follows [KM07]: *i)* idle listening sensor nodes unsuccessfully sample the medium for a potential packet reception, or overhear transmissions whose destination is another node. *ii)* two or more nodes (with overlapping communication areas) simultaneously or almost simultaneously transmit a packet, as a result collision occurs in the medium, thus imposing costly retransmissions. *iii)* finally, depending the routing or MAC layers protocol, the sensor nodes must transmit a control packet (i.e., overhead) before transmitting the data packet for synchronization issue.

Therefore, tremendous gain (in reduction of energy consumption) can be achieved at the link layer where MAC manages the communications between nodes, and thus, the radio itself. Indeed, MAC is also responsible for switching the radio device ON and OFF at regular intervals. In other words, it alternates the radio between periods of activity (transmitting and receiving a packet) and passivity (sleep mode). Two nodes in a WSN may communicate with each other only when both of them are in active mode at the same time. This duty-cycling functionality results in a fundamental tradeoff between reducing the energy consumption to maximize the network lifetime and network performance (e.g., latency, throughput). MAC protocols claiming that they offer 1% of duty-cycle means that the sensor nodes are in active mode (i.e., radio turned ON) only 1% of the time. In this manuscript, we consider contention-based MAC protocols [Pap+14b]. In these family of protocols, nodes in the network asynchronously sample the wireless medium for incoming packets at regular intervals. In between, they turn OFF their radio to save energy.

Furthermore, in patient or animal monitoring applications, requirements such as mobility and bursty traffic appear to be essential. In such applications, the sensor nodes operate under limited internet access for the majority of the time. Hence, once a network connection is detected, the mobile nodes should immediately upload their measurements (in burst) to the sink station [Zha+04]. These sudden circumstances of bursty traffic cause certain anomalies in the network and motivate the researchers to further investigate appropriate solutions.

To summaries, the design of MAC protocol faces a number of limitations due to the low memory and computational, and synchronization capabilities of sensor device. We therefore, in this dissertation, consider contention-based MAC protocols, mainly due to the scalability in traffic load changes (e.g., local decisions) and adaptability in dynamic nature of the network topology (e.g., mobility), as well as its energy efficiency performances. More specifically, we investigate the asynchronous-based family of MAC protocols [Can+11]. By employing these protocols, nodes in the network asynchronously sample the wireless medium for incoming packets at regular intervals. A whole range of energy-efficient and traffic auto-adaptation MAC protocols have been devised in the literature (i.e., [Bac+10]). However, very few of them address the needs implied by the presence of variable and bursty traffic in mobility-aware WSNs. As a result, to the best of our knowledge no successful WSN deployment with mobility handling and auto-adaptation to the traffic has been experienced so far.

Furthermore, considering the characteristics of the deployments in the last 10 years [Iov14], we opted to focus on the multi-hop networks, with a many-to-one data traffic. Finally, due to the nature of the WSN deployments, in our performance evaluation procedure we considered both event and time-driven applications.

1.6 Contributions

The purpose of this dissertation is to optimize the medium access schemes for WSNs within constrained conditions, such as variable traffic and dynamic network topology (due to mobility). Moreover, we aim to improve the overall network performance (i.e., delay, reliability) while reducing the energy consumption. To this aim, we first anticipate traffic load variations and dynamics in the network by adapting the MAC layer parameters according to the estimated upcoming traffic volume of the concerned nodes, and then we focus on improving the integration of mobile nodes in unattended static networks, without causing inefficiencies in the network. All proposed contributions in this manuscript, such as new MAC layer protocols and schemes to handle mobility and traffic load changes in WSNs, aimed to fulfill the previously presented goals. Hereafter, we list the contributions presented in this dissertation:

To bootstrap our investigation, we first perform an exhausted characterization of an IoT testbed. We explore the role of testbeds in the design and development of protocols or applications for WSNs and IoT. More specifically, we highlight to what extent the addition of experimentations can significantly improve the value of performance evaluation campaigns. Moreover, we show to what extent open testbeds can produce scientific results and not only proofs of concept. To this aim, we demonstrate how the conditions of real-deployments can be reproduced on it.

Secondly, we introduce T-AAD, a algorithm that quickly adapts to the traffic load. T-AAD scheme automatically configures its MAC layer parameters on-the-fly. T-AAD is compliant with any preamble-sampling based MAC protocol and allows for reduced energy consumption at both the receiver and sender sides, along with delay and channel occupancy reductions, when compared to the state-of-the-art solutions. We perform a thorough performance evaluation, both through simulation and experimental study over FIT IoT-LAB [Pap+13] testbed.

We then propose M-ContikiMAC, ME-ContikiMAC and MobiXplore, new MAC layer protocols to handle mobility especially in very dense networks. M-ContikiMAC extends the statically oriented ContikiMAC protocol to allow for mobile to static node communication. It is based on anycast transmissions which present some anomalies in the network, such as high packet duplications. We further analyze and investigate how to allow mobile nodes to cooperate and co-exist with static nodes in the network, and we highlight the remaining limitations of M-ContikiMAC. We therefore propose ME-ContikiMAC by introducing a number of improvements. Thus, in the optimized version, a delay enhanced scheme and a packet duplication control mechanism are presented to mitigate packet duplications, which in turn reduces the 1-hop delay, channel occupancy, as well as energy consumption when compared to basic M-ContikiMAC as well as against other state-of-the-art solutions (such as MoX-MAC [Ba+14] and MOBINET [Rot+11]). Finally, we propose MobiXplore, a MAC layer scheme that allows a seamless handover. Note that all previously mentioned proposals are compliant with asynchronous MAC layer protocols.

Finally, we propose an auto-adaptive scheme at the MAC layer to mitigate the packet duplications in opportunistic routing. We propose a mechanism that handles the potential deafness in the network through heterogeneous configuration among the nodes in the network. We do so through local decisions in a decentralized fashion at each node. Nodes, dynamically and automatically adapt their MAC layer parameters in order to reduce unnecessary traffic, channel occupancy and energy consumption (while increasing the reliability) due to packet duplication in opportunistic networks.

1.7 Structure of the Thesis

This manuscript is organized in seven chapters. The first Chapter presents an introduction to Wireless Sensor Networks, as well as the motivation and contributions of this manuscript. The next Chapter presents an overview of the current tendency of the validation methodology that authors follow in Ad-Hoc and WSNs. Moreover, it provides a thorough state-of-the-art on MAC layer protocols in WSNs. It also gives the reader the necessary elements for understanding the rest of this manuscript. In Chapter 3, we detail the methodology that we follow to evaluate our proposed schemes and protocols. Furthermore, we provide guidelines to translate simulation campaign to successful experimental deployments. Starting with Chapter 4, each of the chapters presents one of the contributions of this thesis. We begin with the proposal of a new MAC layer protocol, and more specifically, we present the T-AAD, a preamble-sampling based protocol that automatically adapts the MAC parameters on-the-fly. In Chapter 5 we present the M-ContikiMAC, ME-ContikiMAC and MobiXplore, mobility-aware MAC schemes that improve the integration and communication of mobile to static sensors in WSN, without causing inefficiencies in the network. In Chapter 6, we investigate to what extent the packet duplication issue (due to the nature of anycast transmission mode) depends on both the topology density and the nodes MAC configuration, and moreover, to what extent an auto-adaptive scheme can mitigate this issue in opportunistic routing. Finally, Chapter 7 concludes this manuscript by presenting concluding remarks and opening up some perspectives.

Chapter 2

Current Practices and Literature Review

Verification of theoretical analysis is an vital step to the development of an application or a protocol for wireless networks. Most of proposals are evaluated through mathematical analysis followed by either simulation or experimental validation campaigns.

In this investigation, we analyze a large set of statistics on articles published (i.e., 674 papers in total) in Ad-Hoc and WSNs related top representative conferences (i.e., ACM/IEEE IPSN, ACM MobiCom, ACM MobiHoc and ACM SenSys) during the period 2008-2013 (where 596 are related to Ad-Hoc and WSNs) in order to derive the current tendency of the validation methodology that authors follow. More specifically, we focused on exploring the role of simulators and testbeds in the development procedure of protocols or applications for Ad-Hoc, WSNs and IoT technologies. We show that there is a tendency for more and more researchers to rely on custom or open testbeds in order to evaluate the performance of their proposals.

Simulators indeed fail to reproduce actual environment conditions of the deployed systems. Experimentation with real hardware allows our research community to mind the gaps between simulation and real deployment. Still, as experimental approach through custom testbeds results in a low reproducibility level (i.e., 16.5%), we investigate to what extent such performance evaluation methods will be able to bridge those gaps. We finally discuss experimental testbeds and their potential to replace simulators as the cornerstone of performance evaluation procedures.

Furthermore, this Chapter introduces the functionality of the MAC layer and the design characteristics when operated on top of a wireless sensor node. We then perform a thorough literature review and present a general overview of the different categories of MAC protocols for WSNs along with the major solutions that were proposed so far in the literature to leverage the previously presented issues. In this manuscript, we concentrate on contributions that target on runtime auto-adaptations on traffic load changes, mobility-aware networks, as well as energy-efficiency aspects. Furthermore, we investigate the impact of the MAC parameters to upper layers.

2.1 Performance Evaluation Methods for Ad-Hoc and WSNs

Experiences through the past real-world deployments [Lan+06], [Kro+12], [Che+11], [Bar+08b], [Tho+04] have shown that continuing directly with deployments can lead to various unexpected issues (such as node failure or network disconnection). In fact, the majority of the Ad-Hoc, WSN and IoT applications pose significant challenges due to the hardware limitations of sensor nodes (e.g., processor, memory or battery) or constrained environment in which the nodes are deployed (e.g., ocean, mountains) [Iga+15]. It is therefore essential and crucial to verify the protocols at each of design, development and implementation (either by utilizing simulators or emulators or through experimentations by employing testbeds), before being deployed in a real-world deployments.

Simulation evaluation is an essential phase during the design and development of an Ad-Hoc, WSN or IoT infrastructure. However, environments in which Ad-Hoc or sensor networks evolve are often application-specific and too complex to be reproduced precisely. Simulators allow researchers to provide the proof-of-concept for new solutions in a virtual environment by avoiding time-consuming, heavyweight or expensive real-world experimentation. More specifically, simulators allow users to implement some basic assumptions (e.g., link quality, radio propagation, medium interferences, topologies) [Pap+13]. Although, the majority of the simulation models cannot capture real world complexity [Hir+13] [Bar+08b], they are often utilized as a first step. Our purpose is to show that this step is not sufficient to present the consistency of a solution as well as that low cost devices have steered researchers and engineers to enrich performance evaluation with testbeds.

Experimental evaluation is performed either custom or over open testbeds, and exhibits potential unexpected failures and problems that the proposed solutions by researchers would face during real deployments. Even though performing well over testbeds, they remain in vitro deployments with more or less controlled environment conditions. Such a proof of concept must then be transposed into the real world. Designing and setting up a complete Ad-Hoc or WSN system under real conditions that can support robust applications is a very complex task [Kdo+12]. Researchers and production system architects, first need an appropriate plan of deployment and later number of tools, simulators/emulators and testing facilities for real experiments, in order to initially validate their concept or model and then to develop the appropriate infrastructure.

Simulators and testbeds are two important and complementary design and validation tools. In a typical research process cycle, once the modeling phase is done, network researchers and developers continue with the validation procedure in which they evaluate their concept by using either a simulator or an emulator. Later, network engineers and developers may proceed with experimentation to further cross-verify their proposal [Sto08]. Thus, once both the simulation performance and the experimental measurements are satisfactory, which actually means that the carried out mathematical analysis is validated, then real deployments can be initiated. Figure 2.1 illustrates a typical research process cycle as previously described.

In the WSN community there are number of simulators [EL+05] and open testbeds [Glu+11] available for researchers to perform simulation or experimental evaluation of their protocols or applications. However, there is always a concern about the retrieved results from such tools that they may not reflect the mathematical derivations.

Since we face complex environments that are very difficult to be theoretically analyzed and we also take into account the difficulties of setting up a real-world (e.g., large-scale) deployment, simulations are often considered as the optimal approach for studying the performance of wireless networks. Many open source and freely available simulators allow users to have a better control of the nodes by often employing a Graphical User Interface (GUI), and to retain or simplify some assumptions in order to evaluate their solutions. Simulation evaluation is a provisioning procedure during the protocol development. However, even if the simulation performance presents coherent results with mathematical analysis, past real-world deployments show that it is not recommended to proceed directly with real deployment since engineers may face unpredictable phenomena such as node crashing or network disconnection [Bar+08b], [Lan+06]. Intermediate experimentation platforms can therefore be considered to bridge the gap between simulations and real world

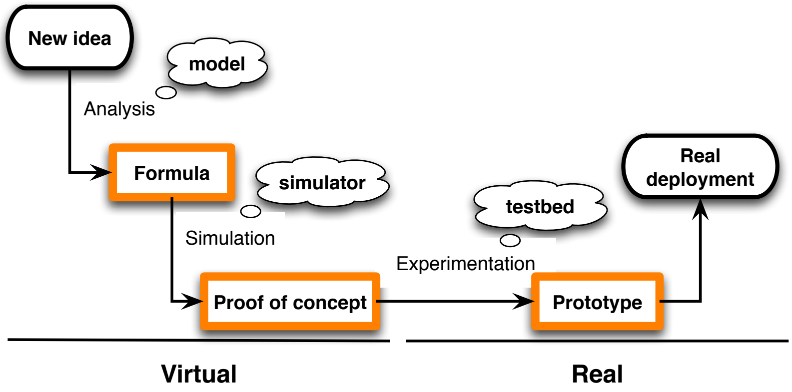
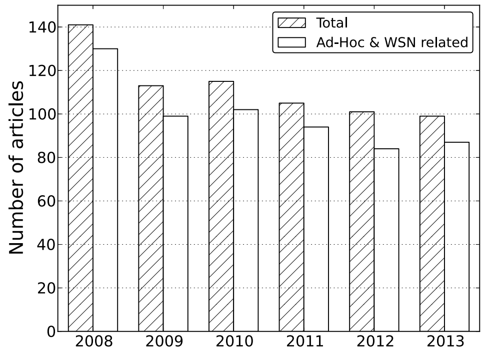


Figure 2.1: A typical research process cycle.

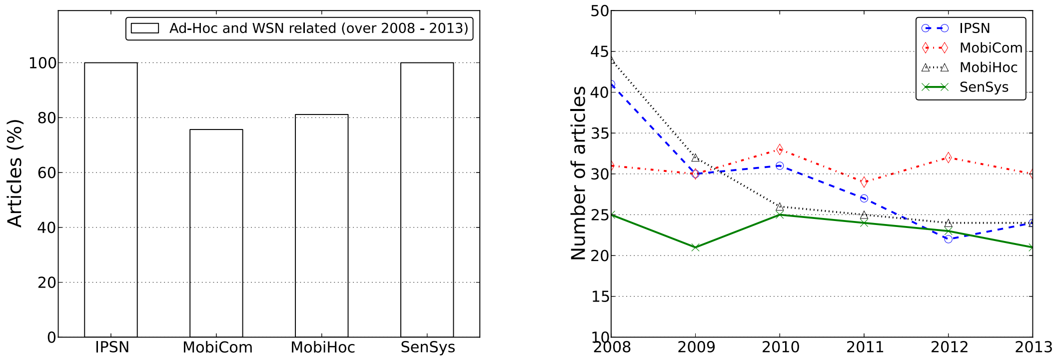
deployments. Nevertheless, while simulations can offer wider sets of assumptions to test and therefore potentially more complete evaluations, testbeds do impose many characteristics (e.g., physical environment, hardware, network topology). Such facilities offer the opportunity to have their solutions facing real conditions, thus being more realistic than those modeled under software simulators. Yet, numerous parameters (e.g., radio dynamics, link stability and symmetry, impact of the weather on communications [Boa+10a]) appear so unpredictable that they may lead to results that can not be reproduced with sufficiently tight confidence intervals. The ambition of obtaining scientific results should then lead researchers to allow for further repeatability of the presented results. As a result, during the simulation evaluation the environmental conditions should not affect the behavior of the nodes. Hence, it would be ideal if the authors first verify their model by employing experimental tests in order to reflect the reality that their proposals would face during real deployment.

2.1.1 A Thorough Literature Study

Throughout this study, we compile a large set of statistics on literature review of 674 articles published in top representative conferences that are strongly related with Ad-Hoc and WSN research fields over the 2008-2013 period [Pap+15c]. In particular, we have studied all articles that have been published at the ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN), ACM Annual International Conference on Mobile Computing and Networking (MobiCom), ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc) and ACM Conference on Embedded Networked Sensor Systems (SenSys) conferences in order to derive the current tendency of the performance evaluation methodology that researchers follow. We especially focus on what extent experiments on testbeds have become a must for performance evaluation of new communication algorithms and protocols. Moreover, we exhibit the tendency for performance evaluation procedures to rely on experiments with real hardware and environment, to the detriment of simulations. The question of scientific results versus proofs of concepts therefore arises. Indeed, we discuss the meaning of reproducibility and of a proof of concept as a prototype being designed to determine feasibility. We also analyze the selection of the evaluation methodology (e.g., simulator, testbed), and simplicity of the overall design that should be provided for validation, understanding and explanation. Furthermore, we investigate and gather the pros both from simulation and experiments so that real-world experiments could lead to reproducible scientific results for our research community. To do so, we go through and study 674 articles in total, published in the conference proceedings for the last six years from 2008 to 2013, out of which 596 are related to Ad-Hoc & WSN (see Figure 2.2a). Indeed, we identified 78 articles that deal with other wireless technologies such as WiFi and WiMAX, that are studied in the context of cellular networks. All of these papers have been



(a) Number of articles per year (all conferences are considered).



(b) Appropriateness of our conference sample.

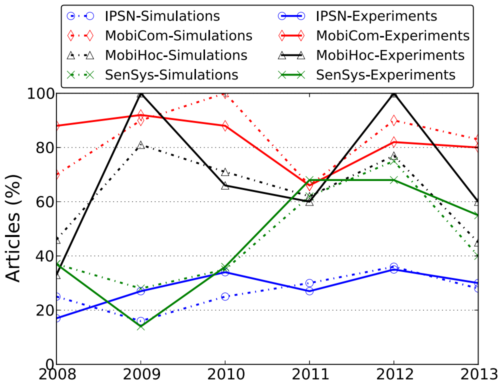
(c) Publication flows over the 2008 - 2013 period.

Figure 2.2: Published articles in ACM/IEEE IPSN, ACM MobiCom, ACM MobiHoc and ACM SenSys from 2008 to 2013.

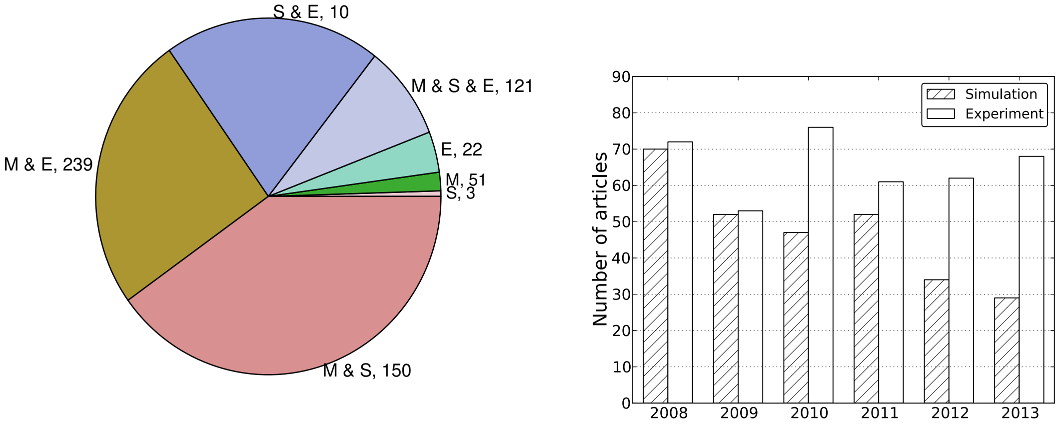
found in MobiCom (i.e., 140 out of 185) and MobiHoc (i.e., 142 articles out of 175) conferences (see Figure 2.2b), which are not entirely dedicated to Ad-Hoc and WSN however have a broad scope on mobility and wireless communications. Thus, we further emphasize our investigation over these 596 articles. During our investigation, we obtain plethora of information for each work and we then categorize the considered articles based on their common features.

Figure 2.2c provides detailed information about the total number as well as the Ad-Hoc & WSN related published articles per proceeding year. We actually observe that, there is a decreasing tendency of published articles in the proceedings, indeed we identified 43 articles less from 2008 to 2013. More specifically, MobiHoc and IPSN reduced the total accepted articles, from 44 to 24 (MobiHoc) and from 41 to 24 (IPSN) respectively, while MobiCom and SenSys kept a steady flow.

Modern technologies introduced the feature of mobility. Consequently, the research community focuses on developing and testing such aspects and scenarios. Our study results justify this trend, owing to the 148 articles (57.7%) that simulated mobile scenarios. Still, our statistical results for MobiHoc and MobiCom, the mobile oriented conferences, show that not all of their articles implement mobility scenarios. For instance, during the 2008 MobiHoc conference we determined only 13 out of 28 simulation-based articles that introduced mobility in their tests. As shown in Figure 2.3a, 57% of articles involving mobility are less induced by our conference sample (half of the conferences, MobiCom and MobiHoc, being theoretically focused on mobility-related topics) than by the global enthusiasm for mobile scenarios.



(a) Mobility scenarios in performance evaluation procedures.



(b) Use of Mathematics (M), Simulations (S), Experiments (E) and their combinations in validation procedures of 596 Ad-Hoc and WSN related articles.

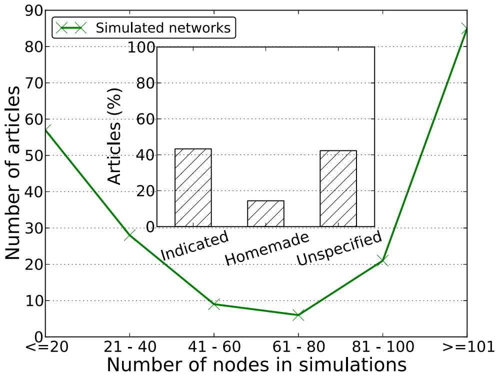
(c) Total simulation versus experimentation evaluated articles.

Figure 2.3: Published articles in ACM/IEEE IPSN, ACM MobiCom, ACM MobiHoc and ACM SenSys from 2008 to 2013.

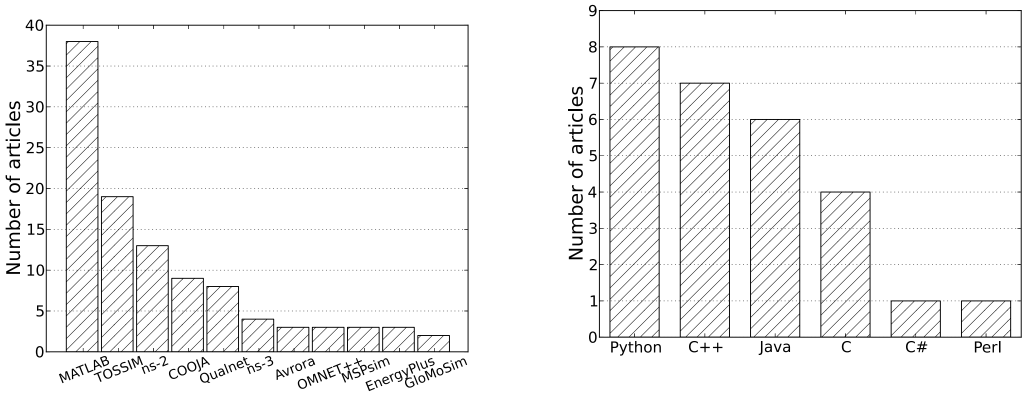
2.1.2 Evaluation Procedures

In this subsection, we expose our analysis on the validation procedures that the authors followed. As a first step, we aimed to categorize the reviewed articles according to the employed evaluation method. In particular, we examine the proportion of simulation, experimental and mathematical (i.e., modeling or analysis) evaluated works. Our primary analysis exposes interesting results. More specifically, our investigation shows that the majority (i.e., 561) of the articles provide an analytical representation of their solution. The remaining 35 have only simulation or experimentation results. Furthermore, 284 verify their proposal by employing simulation evaluation while on the other hand 392 of the articles include experimental evaluation for their validation. Finally, only one out of five (i.e., 20.3%) articles examines all three phases of the research process cycle (i.e., analysis, simulation and experimentation). The number of articles with the previously stated properties (with respect to 596 studied papers) is illustrated in Figure 2.3b.

We now present the characteristics of the articles that we studied. The percentage of simulation versus experiment-based studies (with respect to 596 studied articles) is illustrated in Figure 2.3c. As can be observed, while simulations and experiments used to be equally deployed until 2009, the usage of simulations is decreasing every year (except in 2011) while experimentations still remain present at a relatively stable rate.



(a) Simulator usage and scales of simulated networks.



(b) Popularity of simulators.

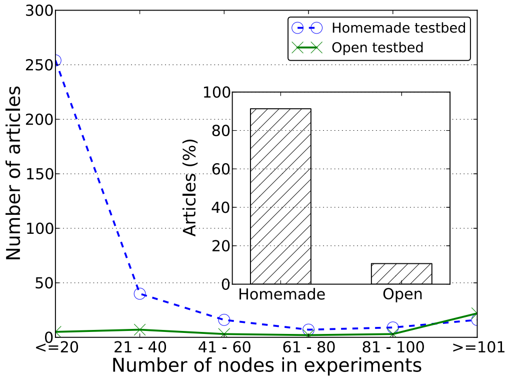
(c) Programming language popularity for custom simulators.

Figure 2.4: Simulation evaluation methods (left) and popularity of simulators (center) and programming languages (right).

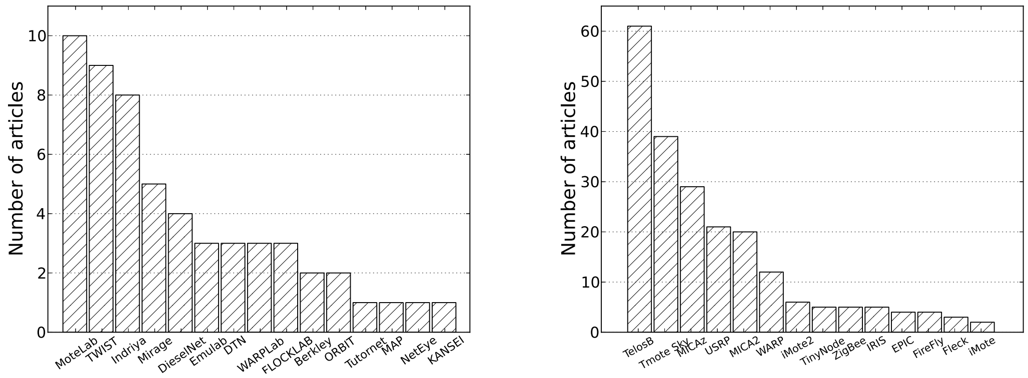
Over the 2008-2013 period, 284 studies followed a simulation evaluation to test their proposal. We noted the simulator usage, the scales of simulated networks and the programming languages used for custom simulators (see Figure 2.4). Only 43.3% are validated through a known simulator while 42.3% of articles did not even provide any information about the tool that their authors have utilized (see Figure 2.4a). Finally, 14.4% (with respect to 284 studied simulation-evaluated articles) developed a homemade simulator (Figure 2.4a), by utilizing programming languages, such as Python and Java (see Figure 2.4c for the distribution of the most popular programming languages).

We are next interested in determining the usage of the simulators. As can be observed in Figure 2.4b, MATLAB is the first choice in our community counting more than 35 articles, followed by TOSSIM which has been reported in almost 20 articles. Furthermore, Network Simulator 2 (ns-2) comes third with 13 articles.

Nowadays, the research community is able to evaluate proposed protocols, models, even new technologies over open testbeds at a very large-scale [Glu+11]. Increasingly, network researchers are using experimentations to enlarge the scope of their performance evaluation, Figure 2.3c. Moreover, as it can be observed in Figure 2.5a, our investigation shows that the majority of the researchers, 91.3%, choose to set up their own testbeds. Even though to the current day, there is a number of open facilities providing to the developers the infrastructure needed for experimental Ad-Hoc, WSN or IoT studies, only 10.7% of the articles use open platforms. Our



(a) Testbed usage and scales of experimented networks.



(b) Popularity of open testbeds.

(c) Motes popularity.

Figure 2.5: Experimental evaluation methods (left) and popularity of open testbeds (center) and motes (right).

compiled statistics tend to show that researchers would rather favor their own setups for small scale deployments. In fact, among the 392 articles exposing experimental results, 78% of them do not exceed 40 nodes for their experimental setup (see Figure 2.5a). Hence, the increased difficulty to apprehend a remote open testbeds (e.g., specific hardware and software, network topology, booking procedure) may have induced researchers to set up their own relatively small scale networks.

Finally, we evaluated the popularity of the devices in homemade experiments. In Figures 2.5b and 2.5c the utility of the open testbeds and motes is presented. Even though a small number of articles experimented over open testbeds, we pointed out the popular open platforms. As observed in Figure 2.5b, Harvard's Motelab comes first (11 articles), followed by TWIST (10 studies). That can be simply explained as those facilities were the first to open up to the scientific community. Regarding the Indriya testbed, even though it was made available only since 2011, it was used in 8 articles. The fact that users can interact with the testbed through the same intuitive web-based interface as MoteLab's, could explain this success among the community.

2.1.3 Reproducibility

We continue our study by investigating the feasibility of reproducing results that are presented in the reviewed articles, both for simulation and experimental campaigns. To proceed so, we looked for some critical information (e.g., simulation setup, simulator indication, simulator details such as version or library, number of nodes), that should be provided by the studied articles. In order to reproduce the proposed solution, we assumed that the authors should provide a complete simulation or experiment settings subsection.

Regarding the simulation based evaluations, while only 43.3% of the articles indicate the simulator, 78.5% of those do provide some details about simulation setups. Among those, 72.5% precise the number of involved nodes. Finally, we decided of non complete setups as soon as there was a lack of critical details regarding the tools used during simulations. For instance, as discussed earlier, MATLAB stands as the most popular software for simulations. In order to use it as a network simulator, researchers must import external libraries (e.g., as developed by the WISLAB* team). It is difficult, if not impossible, to reproduce a simulation study when the version of a publicly available simulator is unknown, and only 21.5% provide us with the employed version or the utilized library of the simulator, which essentially concludes our outcome about the reproducibility of the simulation-evaluated articles.

We followed similar methodology for the experimental-based validations. Taking into account the nature of open platforms, the 42 articles, we consider that these articles overall are reproducible. However, we counted 8 papers where the authors tested their ideas over both custom and open testbeds, with only 3 of them providing enough information to be assumed reproducible. On the other hand, the experimental results that are retrieved through homemade testbeds can be considered as difficult or even impossible to reproduce. This is explained since most of them are deployed in offices, houses or even outdoor installations where the environmental radio activity varies, due to the interpolation of external features such as mobile phones, wireless routers and access points and so on. Nevertheless, owing to the nature (e.g., application layer) of the tested solution, we detected 31 homemade-based studies that may be reproduced. Finally, by summarizing the previous statements, we calculated that only 16.5% (65) of the experimental-based papers present reproducible results.

2.1.4 Scientific results or proofs of concepts ?

Scientific results are expected to be repeatable while a proof of concept is a realization of an idea that demonstrates its feasibility. Our initial investigation shows that most of the authors choose to validate their proposals over experimental evaluation. Our investigation highlights some interesting tendencies in the networking scientific community, especially around Ad Hoc and Wireless Sensor Networks. As previously presented in Section 2.1.1, an increasing number of papers validate their proposals by using experimental evaluations.

We focused on the simulation and experimentation setups in order to determine if they were sufficiently described to allow for repetition of the evaluation procedure. While Kurkowski et al. had focused on MANET, thus, looking for simulation parameters specific to mobility (e.g., speed of nodes, speed delta, pause time, pause delta), we aimed at a larger scope by gathering various sets of setup parameters. This is especially true for all observed experimentations among which setups are highly different (e.g., hardware, physical topologies, radio environment). The reproducibility level of experimental studies is lower than the simulation one. This is even more dramatic as this latest has not varied much since the study of Kurkowski et al.. More specifically, the authors had identified 29.8% of the simulation-based articles that did not identify the simulator used in the research. As mentioned in Section III.A, regarding the 4 conferences we observed over the 2008-2013 period, this proportion has raised to 42.3%. In addition, they had calculated only 12.1% of the articles where the simulator version was mentioned. Furthermore, the authors were concerned that more than 90% of the published results may include bias. As result, they conclude that approximately 12% of the MobiHoc simulation-based results appear to be repeatable. In [Kur+05], numerous pitfalls throughout the simulation lifecycle had already been observed. Those tendencies, as already highlighted by Kurkowski et al., take away from the goals of making the research repeatable, unbiased, realistic, and statistically sound.

---

[*] http://wislab.cz/

As previously observed, over the last six years, less and less papers have actually considered simulations during their performance evaluation process. Still, the simulation phase allows researchers to demonstrate that the main principles of their proposal are indeed effective, before implementing them over a testbed [Sto08]. However, in order for the users to be able to continue their proof of concept validation, we can avoid the necessity that they have to get familiar with various simulators and testbed platforms. Emulators such as TOSSIM[†] or COOJA[‡] were developed to bridge the gap between simulation and experimentation, by being very close to real embedded systems in terms of architecture compilation targets. In fact, by utilizing these simulators, the very same code remains unchanged over the transfer from simulation to experimental campaign.

We are coming to a trade off between realism and reproducibility. More specifically, on the one hand there are more published articles that are closer to real deployment while on the other hand the reproducibility level of the studies decreases. So far, the proportion of papers using experimentations that allow to reproduce the conditions of an experiment remains very low ($< 11\%$). Moreover, all those testbeds are highly different (e.g., hardware, physical topologies, radio environment) and each would require a specific guidance to allow for scientific results to be obtained.

In [Kur+05], authors had proposed a simulation study guidance. If the enthusiasm for experiments in networking scientific papers is to be confirmed, we should also be able to establish such mandatory steps to ensure statistically sound results. The significant number of open access and large-scale testbeds that have been deployed over the recent years [Glu+11], provides appropriate tools and experimental facilities for researchers and engineers to perform real experiments in order to further analyze their protocols. Open testbeds allow users to easily deploy source code (that could be the same with the one of the simulator) on a sensor node and to flash it at no delay. Those open platforms thus allow for more rigorous, transparent and replicable testing of proposed protocols and models.

Researchers, by connecting remotely (e.g., via ssh) to one open platform, may set up and initiate an experiment by using the terminal. Hence, the previously reported simulators along with open testbeds, allow the research community to get a flavor of real deployments while maintaining a unique programming code. More importantly obtaining performance evaluation measurements over large scale network (both for simulation and experiments) can be at no cost at all.

Finally, after following all the previously presented steps, and by obtaining coherent results, researchers may consider to initiate a real deployment by utilizing their verified and refined protocol.

2.1.5 Mobility

Mobility is a key aspect for the future designs. While the majority of existing and used simulators allow to use and create mobility models, testing and executing such scenarios during an experimentation procedure requires to involve and combine advanced and intelligent technologies such as robots. Consequently, very few of the widely popular open platforms do support mobility [Ton+14]. Actually, there is a number of challenges that need to be addressed having mobile robots in a testbed, namely, charging, remote administration and maintenance of the robots. Indeed, robots must be able to reach their docking stations automatically. Conversely, remote users must be able to interact with robots over reliable links (e.g., WiFi). Even though these challenges can be addressed, testbed administrators then face the issue of localizing mobile devices in order to allow for repeatable trajectories. Indoor deployments can not rely on GPS solutions and thus impose distance approximations to be computed based on other available inputs (e.g., received signal strength intensity) or on costly technologies (e.g., 3D camera with range detector sensors for the mapping of the environment). Furthermore, even with perfect localization of all robots,

---

[†] http://tinyos.stanford.edu/tinyos-wiki/index.php/TOSSIM

[‡] http://www.contiki-os.org

trajectories would be very difficult to replay, especially due to the odometry drift. Some 3D cameras using range detector sensor aim at handling this drift. Still they lack to compute the path where not enough landmarks exist in open-space and large-scale environments.

2.1.6 Overview of Evaluation Methodologies

In this study, we reviewed 674 papers that were published in four major and representative conferences in Ad-Hoc and Wireless Sensor Networks, over the 2008-2013 period. We especially focused on the performance evaluation procedures in order to raise the question of whether simulations and experiments lead to scientific results or proofs of concepts. It is undeniable that simulators make the whole process of validation easier, faster and less expensive. On the other hand, with the growing development of open and realistic testbeds, researchers may overcome the technical challenges and economical barriers of real-world deployment to perform a thorough experimental evaluation of their ideas in wide-scale platforms. Simulators and open testbeds are two crucial and complementary design and validation tools; theoretically development process should start from the theoretical analysis by providing bounds and indication of its performance, be validated and verified by simulations and finally confirmed in open testbeds. Hence, once the entire procedure is successfully done and the performance results show coherence, then researchers could push their solution to engineers in order to proceed with real deployments.

This initial study helped us to define our own performance evaluation methodology (as we describe later in Chapter 3). In particular, we aimed at applying both simulation and experimental evaluations in our proposed solutions in this manuscript.

In the following Sections, we first introduce the functionality of the MAC layer and the design characteristics when operated on top of a wireless sensor node. We then perform a thorough literature review and present the major solutions that were proposed so far in the literature to leverage the previously presented issues. Large amount of work has been conducted in the last decade in the area of MAC protocols especially designed for WSNs [Lan07] [KM07], [Bac+10], [Can+11], [DD13]. We will therefore concentrate on contributions that target on runtime auto-adaptations on traffic variations, mobile aware networks, as well as energy-efficiency aspects.

2.2 Controlling Medium Access in WSNs

The MAC is a sub-layer of the Data Link Layer (DLL) specified in the seven-layer of the OSI model (layer 2). More specifically, it is located between the Logical Link Control (LLC) sub-layer (which provides multiplexing and flow control mechanisms) and the network's physical layer. MAC is responsible for coordinating the access to the wireless medium shared by several nodes. In fact, the MAC sublayer provides addressing and channel access control mechanisms that allow sensor nodes to communicate within a network that incorporates a shared medium. To do so, MAC by utilizing a channel access protocol, regulates when a node should transmit or listen the medium. Typically, depending on the type of physical link, the channel provides unicast, multicast (i.e., for wired) or broadcast (i.e., for wireless) communication service. The numerous solutions that have been proposed so far aim at fairness, reliability, scalability, low latency and fair throughput among the nodes in the network. Especially, when considering wireless links, it is essential to properly coordinate the node to avoid interferences or even collisions (simultaneous transmissions on the medium, which result in a jammed signal) or deafness (when the radio of a recipient is turned OFF during the packet transmission).

MAC protocols such as Distributed Coordination Function (DCF) and Point Coordination Function (PCF) (i.e., IEEE 802.11 standard [Ieeb]) are not suitable for WSNs due to number of constraints. In IEEE 802.11 standard, by default all node utilize the DCF scheme, which is based on CSMA/CA [Nam+12]. Carrier Sense Multiple Access (CSMA) is a contention-based protocol where the nodes first sense the medium before transmitting. The main goal is to avoid having stations transmit at the same time, which results in collisions and corresponding retransmissions. As an optional access method, the standard defines the PCF, which is a contention-free protocol

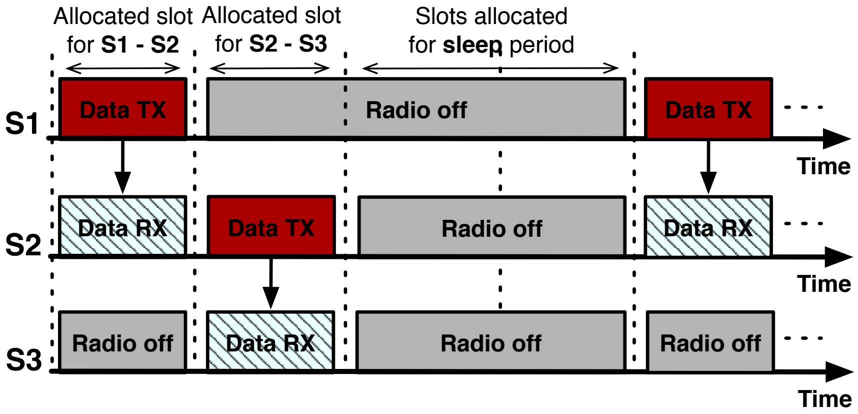


Figure 2.6: An example of slotted protocol.

and enables nodes to transmit data frames synchronously, with regular time delays between data frame transmissions. With PCF, the access point controls which nodes can transmit during any give period of time. However, those protocols suppose that sensor nodes come with high computation resources, memory and synchronization hardware, as well as unlimited energy resources. Moreover, they are not tolerant with multi-hop networks or node failures. Furthermore, in DCF for instance, the stations continuously listen to the channel for potential reception, an operation that rapidly consume the energy resources, and thus these protocols do not address the energy consumption issue.

In WSNs, the access to the medium can be in scheduled-based procedure, decided by a central entity (i.e., sink), or, conversely, can be in random-based in which the transmissions are decided locally, independently at each node. Furthermore, the access could be in hybrid-based that combines the two previously mentioned techniques. Hereafter, we provide a detailed description of the MAC protocols in WSNs.

2.2.1 Scheduled Protocols (Slotted schemes)

In Time Division Multiple Access (TDMA) based slotted schemes, the time is divided into slots distributed among the nodes in the network. At each time slot, a node may transmit or receive a data packet, or power OFF its radio (sleep). These schemes determine a collision-free schedule for small networks. It is therefore particularly suitable to handle periodic traffic. However, it is a complex task to guarantee a collision-free slots for very large networks.

A simple example is depicted in Figure 2.6. It displays a TDMA scheme which divides the communication window into 5 time slots. The communication window is divided into four time slots among three sensor nodes. As can be observed, the time slots are distributed among 3 nodes, namely S1, S2 and S3 (i.e., sink station), where S1 is the origin of the data packets. More specifically, slots are assigned for the communication from S1 to S2 and from S2 to S3, while the 2 other slots are dedicated to switch off the radio of all the sensor nodes. Note that when a node is not involved in a time slot, it turns its radio OFF, into sleep mode. This communication window is duplicated over time as long as the network operates.

MMSN (Multi-Frequency MAC for WSNs) [Zho+06], PEDAMACS [EV06], SMACS (Self-Organizing MAC for Sensor Networks) [Soh+00], TRAMA (Traffic-adaptive MAC) [Raj+06], FLAMA (FLow-Aware Medium Access) [Raj+05] are some of the seminal works. The Timeslotted Channel Hopping (TSCH) mode of the IEEE 802.15.4, which is part of the standardized network stack, is also a scheduled protocol that uses both TDMA and Frequency Division Multiple Access (FDMA) techniques.

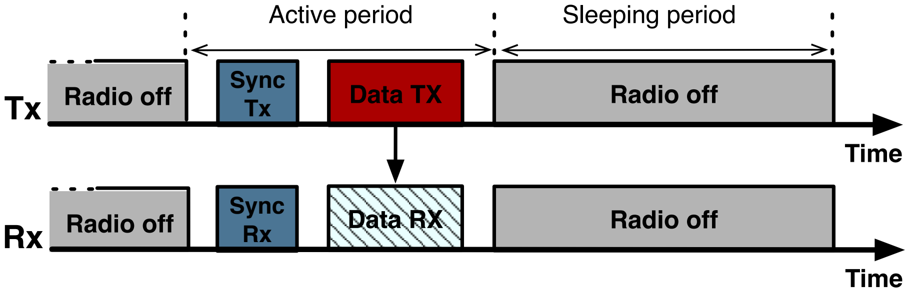


Figure 2.7: An example of MAC protocol with a common active/sleep schedule: the S-MAC protocol.

2.2.2 Protocols with Common Active Periods

Sensor nodes utilizing a common active/sleep period, periodically wake-up in a synchronized manner to transmit or receive data packet. The nodes continuously alternate between active and sleep periods. In fact, a node sends control packets to synchronize the active/sleep periods, by broadcasting a beacon to announce the neighbors to synchronize with it, usually, at the beginning of each active period. Thus, during the sleep periods, the nodes turn their radio OFF to save energy, while during the active periods, the nodes perform the Synchronization (SYNC) procedure as well as data packet transmissions and receptions, by using a contention-based scheme such as CSMA. MAC protocols with common active/sleep periods are more suitable for applications that deal with time-driven traffic (e.g., monitoring).

S-MAC (Sensor MAC) [Ye+02] and T-MAC (Timeout MAC) [DL03] are two representative protocols in common active/sleep-based family of protocols. As illustrated in Figure 2.7, the active period is divided into two consecutive phases: the SYNC period followed by the data packet transmission/reception. The duty cycle of this mechanism depends on the length of the sleep period compared to the active one. With S-MAC, the latter is fixed prior to the network operations, which makes it prone to idle listening and overhearing whenever the traffic load fluctuates. T-MAC improves and mitigates the idle listening of S-MAC by utilizing a very short listening window at the beginning of each active period while the length of active period is dynamically adapted to the traffic, using a timeout. As a result, the sleep period is increased, and consequently more energy is saved. Other protocols that use common active periods are: SWMAC (Separate Wakeup MAC) [Pak+06].

The IEEE 802.15.4 standard [802b] also uses active/sleep periods in its beacon-enabled mode (beacon messages are transmitted at the beginning of the active period). Later, nodes compete for the medium in a contention-access period (i.e., slotted-CSMA) within each time slot. Furthermore, the standard allows for an optional contention-free period in order to guarantee time slots to specific nodes. Finally, it comes with inactive periods which allow the nodes to save energy by switching their radio OFF before the next beacon.

Protocols with common active/sleep periods require high quality hardware to perform the complex clock synchronization. Moreover, there is a scaling issue as the size of the network increases the additional overhead is increasing as well. Furthermore, setting the optimum duration of the active period is not an easy task, while a longer active period decreases the collision probability and increases throughput, it decreases the sleeping period on the other hand. As a result, more energy is consumed. Finally, the active period of the duty-cycle is commonly large because it should give time for transmitting synchronism messages, as well as control and/or data packets.

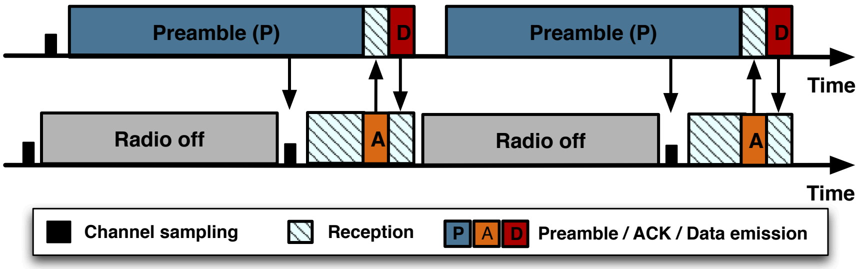


Figure 2.8: An example of preamble-sampling MAC protocol: the B-MAC protocol.

2.2.3 Preamble-Sampling Protocols

In this family of protocols, no synchronization is required. Indeed, nodes in the network asynchronously sample the wireless medium for incoming packets at regular intervals. In between, they turn *OFF* their radio to reduce energy consumption (i.e., duty cycling). More specifically, by employing these protocols, each sensor node has independent, from neighboring nodes, schedule for an awake and sleep period. The transmitter node sends very long preambles, prior the data packet, to ensure that the intended receiver will stay on upon sampling the medium. Preamble-Sampling ALOHA [EH02], Preamble-Sampling CSMA [HC02] and B-MAC [Pol+04] were among the first of the preamble-sampling protocols. As we can see in Figure 2.8 the neighboring node has to periodically wake up and check the medium for possible transmissions. If the receiver node senses the medium busy, it will keep its radio turned *ON* to receive the upcoming data packet. B-MAC, at least, satisfies the goals of non synchronization but then remains sub-optimal in many ways. More specifically, it suffers from the overhearing problem, while the long preamble dominates the energy usage. Indeed a node may wake-up and stay awake unnecessarily due to a long preamble. In overall, many things can be optimized in B-MAC.

Based on the characteristics such as varying and bursty traffic and dynamic WSN topologies, we decided to consider the asynchronous-based approach instead of synchronized ones. Indeed, by employing X-MAC protocol for instance, nodes operate in a fully decentralized fashion, take decisions based on their own criteria (contrary to S-MAC protocol) without utilizing control packets (i.e., overhead) and impacting the neighborhood nodes. Moreover, preamble-sampling family of protocols are tolerant to the network scale compared to scheduled and active/passive methods, because cooperation of nodes required by efficient forwarding will be limited to small groups of nodes. Those approaches can also address the evolving network topology and node mobility (nodes to join or leave a deployed network more effectively). Finally, preamble-sampling protocols are more adequate for applications with low traffic and event-driven traffic pattern (e.g., alarms).

2.2.4 Hybrid Protocols

The hybrid protocols combine the concepts of the scheduled or slotted with preamble-sampling protocols to take advantage of their characteristics. These schemes are particularly suitable for networks under variable traffic patterns. More specifically, when a small number of nodes transmit, contention-based approaches are a better choice, and vice-versa when a large number of nodes transmit, then scheduled protocols present a better performance.

Protocols such as Zebra MAC (Z-MAC) [Rhe+08] or Scheduled Channel Polling (SCP) [Ye+06] utilizes a scheduled TDMA-based scheme (e.g., common active/sleep S-MAC) to reduce the collisions during the high contention periods when the traffic load increases. Conversely, as long as the contention level remains low the protocols switch to CSMA schemes such as B-MAC protocol to mitigate the idle listening. Other main protocols using this hybrid technique are

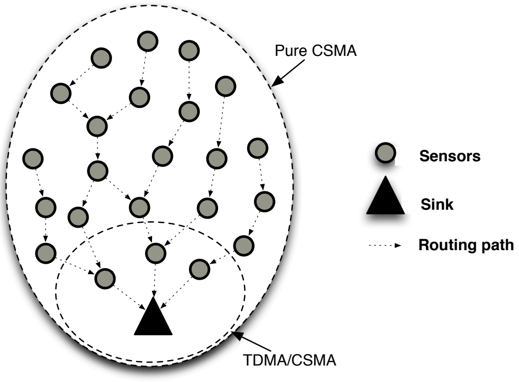


Figure 2.9: Funneling-MAC, combines different schemes in various locations of the network.

MAC family	Main characteristics	Seminal works
Scheduled Protocols	complexity, centralized, synchronized	PEDAMACS [EV06], SMACS [Soh+00]
Protocols with Common Active Periods	energy wastage, synchronized	SMAC [Ye+02], TMAC [DL03]
Preamble-Sampling Protocols	decentralized, asynchronous, simplicity, scalable	Preamble-Sampling ALOHA [EH02], Preamble-Sampling CSMA [HC02], B-MAC [Pol+04]
Hybrid Protocols	scheduled / slotted + preamble-sampling	Z-MAC [Rhe+08], SCP [Ye+06]

Table 2.1: Overview of Medium Access Control protocols in Wireless Sensor Networks

MH-MAC [Ber+07] and Funneling-MAC [Ahn+06], the late one suggests the use of an hybrid CSMA/TDMA algorithm around the sink while pure CSMA scheme (e.g., B-MAC) in the rest of the network (see Figure 2.9).

2.2.5 Overview of MAC Protocols for WSN

Table 2.3 summarizes the prime characteristics of each category along with the seminal works that have been done. Among all presented categories of MAC protocols, the preamble-sampling family of protocols come with some interesting characteristics that make them especially appealing for WSNs. Indeed, the preamble-sampling protocols are less complex and costly when compared to scheduled-based protocols, since no scheduling (and distribution) is needed for the communication between two sensor nodes [KM07]. Moreover, compared to protocols with common active/sleep periods, preamble-sampling protocols consume less energy when the traffic load is low, due to the extremely short channel sampling periods, which allows a node to turn the radio immediately OFF once the medium sampled as idle.

Furthermore, since one of our envisioned applications is wildlife monitoring which includes varying traffic and mobility aspects, we therefore consider the preamble-sampling family of protocols for our investigation due to scalability (e.g., local decisions) and dynamic nature of the network topology (e.g., mobility), in order to design and develop MAC layer solutions to address the previously presented issues.

In the following Section, we present the state-of-the-art of runtime auto-adaptive and mobility-aware works in preamble-sampling based approach.

2.3 MAC under Dynamic and Traffic and Network Topology

WSNs have come to maturity, thus allowing more complex applications (i.e., wildlife monitoring to home-care), varying network topology (e.g., mobility, node failures) and traffic loads which consequently leads to unexpected events. In such event-driven traffic models, where nodes only send their readings upon detection of a specific event determined by the application, nodes store their readings and send them in a row sometime after (i.e., bursty traffic, low 1-hop delay). Furthermore, in such deployment, energy-efficiency is one of the most important parameters, since nodes have to save energy to meet the lifetime requirements of typical applications. It is also important to note that none of the basic MAC protocols (presented in Section 2.2) in each category is able to provide traffic load adaptability by itself. We therefore investigate the runtime traffic auto-adaptations to address the energy/latency trade-off.

In application such as wildlife monitoring [Dyo+10b], [Tho+04], [Zha+04] or clinical medical and home-care [Chi+10a], energy-efficiency is one of the most important parameters, since nodes have to save energy to meet the lifetime requirements of typical applications. To the best of our knowledge, no deployment has taken place without considering the energy efficiency. Furthermore, WSNs have come to maturity, thus offering more complex applications and traffic patterns. In event-driven traffic models, nodes only transmit their measurements upon detection of an event determined by the application, otherwise when the channel is occupied or the network becomes unreachable, nodes may store their collected data and transmit them later at high-rate for short period of time (i.e., burst). This situation implies increased channel occupancy for limited periods of time, at the cost of deteriorating the average network performance.

2.3.1 Toward runtime adaptation

During the previous decade, many contention-based (i.e., asynchronous) MAC protocols were proposed [Can+11]. Preamble-sampling are divided in two prominent categories the Low-Power Listening (LPL) and Low-Power Probing (LPP) respectively. In LPL-based protocols, nodes sleep most of the time and wake-up periodically (i.e., asynchronously) to sample the channel. If a node detects a carrier, it keeps its radio ON to receive the associated message; otherwise it returns to sleep. The transmitter before sending the data packet, sends preambles for a period longer than the sleeping period (of the receiver) so that the receiver can detect the carrier. Conversely, in LPP-based protocols (e.g., Koala [ME+08], RI-MAC [Sun+08]) the opposite happens: the sender first waits for a probe from the intended receiver and then it sends the packet. In this manuscript we focus on the LPL family of protocols, thus, below we review the LPL-based key contributions from the literature that are related to auto-adaptation.

Post B-MAC protocols such as X-MAC [Bue+06], MX-MAC [MH10a], SpeckMAC [WA06] or ContikiMAC [Dun11] replace the long preamble used by B-MAC with series either of strobes (e.g., X-MAC) or of data packets (e.g., ContikiMAC). Hence, the transmitter repeatedly sends strobes or data that contain the address of the receiver. On the other side, the intended receiver replies with an ACK and stays ON until the data transmission is complete. Nodes that have a different address from the one that is indicated in the strobe go back to sleep. ContikiMAC, in addition addresses the false wake-up problem with phase lock and fast sleep techniques which make the transceiver turn OFF the radio for longer period of time. The details of the transmission schedules of X-MAC and ContikiMAC are given in Figure 2.10. Moreover, with BoX-MACs [ML08] or ContikiMAC, a node can be configured so that to keep the radio ON for short time, right after receiving a packet in order to cope with consecutive packets. These protocols greatly reduce overhearing on the nodes and achieve low power operation at both the receiver and transmitter sides. However, they do not meet our goals of automatic and on-the-fly localized adaptation of duty-cycle configurations in the network, since all the nodes of the network operate homogeneously. In [MH10a], Merlin *et al.* present MiX-MAC where nodes switch schedules from a pool of MAC protocols (i.e., X-MAC, MX-MAC and SpeckMAC) based on some parameters (e.g., packet size). However, since it switches between post B-MAC protocols, it has the same disadvantages.

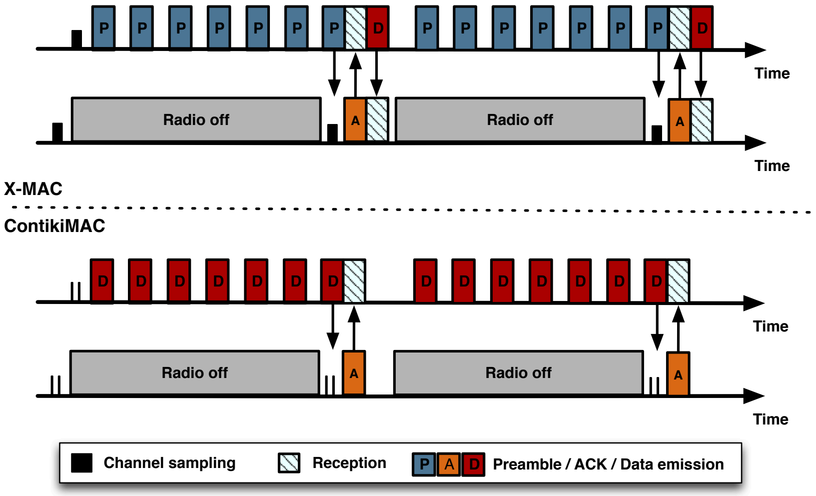


Figure 2.10: X-MAC and ContikiMAC protocols.

In [MH10a], Merlin *et al.* present MiX-MAC where nodes switch schedules from a pool of MAC protocols (i.e., X-MAC, MX-MAC and SpeckMAC) based on some parameters (i.e., packet size). The sender decides which protocol to follow while the receiver does not need to know which protocol is being used, as it simply wake-ups and senses the channel regularly. As a consequence, MiX-MAC requires no overhead, and allows for both energy consumption and latency gains. However, since it switches between post B-MAC protocols, it bears with the same disadvantages.

In [Kun+11], Kuntz *et al.* propose BOX-MAC (for Burst-Oriented X-MAC) where nodes automatically tune their MAC layer parameters (reconfigure the sampling frequency from low to high). They do so for a fixed time (e.g., 10 *sec* for instance), in order more efficiently to handle the burst transmission. This last aspect represents the main drawback of BOX-MAC, where independently of the number of packets to transmit the nodes must stay in high sampling frequency mode for fixed time a priori. Hence, on the one hand nodes consume more energy since they stay in high sampling frequency for longer than needed when few packets are to be transmitted, and on the other hand the latency may is increased when a lot of packets are to be transmitted. As a result, the adaptation is not precisely correlated with the traffic rate.

In [Anw+10], Anwander *et al.* propose the BEAM protocol (Burst-aware Energy-efficient Adaptive MAC) to optimize X-MAC. They do so by appending the payload directly to the strobes, and aggregate multiple consecutive frames into a single packet when they are destined to the same node. Moreover, BEAM introduces an indicator to inform the sender about the buffer state of the receiver, in order to avoid buffer overflow. It also proposes traffic indicators to inform the receiver whether there are more packets to transmit. If yes, the destination doubles its duty-cycle, allowing the sender to reduce the preamble size. This protocol requires *a priori* knowledge of the probability of receiving a packet in any given interval. This can introduce a certain latency in the establishment of a different schedule on the routing path, as the computation of new values would first require the reception of a few packets at a higher or lower rate.

pTunes [Zim+12] dynamically adapts the networks MAC configuration depending on the traffic load. Along with every message sent to the sink, information about the networks status (e.g., traffic load, loss-rate, delays) is added, using piggybacking. Over the successive packet receptions, the sink updates network statistics and computes a MAC configuration corresponding to the actual state of the network. Then, it broadcasts this value to the whole network, so that each node can fit to it. This approach provides gains in terms of performance (e.g., delays, loss-rate) as the network will dynamically adapt to the traffic load. But, it requires a broadcast mechanism relying on node synchronization. Also, this scheme adapts the nodes MAC configuration globally, providing to each of these the same MAC values. Then, the nodes being configured homogeneously over the network, pTunes could fail to address local traffic load changes.

LA-MAC [Cor+12], a Low-Latency Asynchronous MAC protocol, which is based on preamble-sampling family of MAC protocols. Sensor nodes by employing the scalable LA-MAC protocol efficiently adapts their behavior to varying network conditions. More specifically, LA-MAC periodically adapts local organization of channel access based on network dynamics and priority of requests such as the number of active users, age of a burst, burst size and the instantaneous traffic load. LA-MAC is based on network where neighbor nodes are organized in a structure corresponding to the routing information (e.g., DAG, Clustered Tree, mesh). The main concept of the mechanism is an efficient forwarding algorithm based on proper scheduling of children nodes that want to transmit, note that in LA-MAC, the parent node, occasionally becomes a coordinator that schedules transmissions in a localized region. LA-MAC achieves low latency and energy consumption, but it is a distributed scheme since it relies on clustering, which induces an overhead in terms of construction and maintenance of the structure. Indeed, it imposes additional control packets (i.e., SCHEDULE messages) transmissions in broadcast to synchronize the rendezvous of the children nodes in the network.

In MaxMAC [HB10], each node keeps estimating the rate of incoming packets. Nodes change their duty-cycle by allocating so-called Extra Wake-Ups when the rate of incoming packets reaches predefined threshold values, and de-allocate them when the rate drops below the threshold again. Hence, during the first two thresholds nodes double their duty-cycle while in the last (third) threshold, the sensor nodes start operating in a CSMA fashion without going to sleep. The receiving nodes use ACK packets to inform about the change in the duty-cycle and the duration of it. Moreover, switching in the CSMA mode, the receiver node will resulting a higher energy consumption at its side.

In [MH10b], the authors present AADCC (Asymmetric Additive Duty Cycle Control) that is based on the number of consecutive packet transmissions. Hence, each node increments its sleep time by 100 ms when five consecutive packets are transmitted successfully to the destination while for each failed packet the sleeping time is decreased by 250 ms. The aim of this algorithm is to smoothly react to an increase or decrease of channel contention. Jurdak *et al.* [Jur+07] propose Adaptive Low-Power-Listening (ALPL) a cross-layer framework for network-wide power consumption optimization and load balancing in WSN through greedy local decisions. The algorithm periodically gathers neighborhood state information, performs local calculations on the gathered state, and modifies the local configuration of routing and MAC layer accordingly. ZeroCal [Mei+10], is an asynchronous scheduling approach that configures the MAC parameters (i.e., wake-up intervals) between parent and child nodes on the fly to minimize and to balance the energy consumption. To compute the wake-up intervals, an energy consumption model is used based on collecting statistics (e.g.,, packet loss, network topology) on each node and all its children, and is made at fixed intervals or when the number of sent packets of a child exceeds a given threshold. All AADCC, ALPL and ZeroCal algorithms are dedicated to extend the network lifetime. This leads to some nodes using shorter intervals than any of its children in order to prevent preamble misses. Hence, they manage to rapidly adjust duty-cycles on traffic increase, but adapt slowly to reductions in load. Moreover, these proposals may not guarantee the end-to-end delay bounds as the wake-up interval can be extended to save nodal energy. Finally, it takes time and induces communication overhead to adapt, as it requires several messages in order to perform some calculations first and then to change the configuration. This may lead to excessive latency when facing bursty traffic.

2.3.2 Overview of Adaptive MAC Protocols

These solutions may not satisfy our goals of addressing dynamic and bursty traffic, in a highly reactive and low energy consumption manner.

Hence, as summarized in Table 2.2, the static protocols (i.e., preconfigured static MAC values, such as {B, X, MX, Speck, MiX}-MAC solutions) do not provide any on-the-fly auto-adaptation and maintain homogeneous MAC configurations for the whole set of deployed nodes. On the other hand a protocol like pTunes, although dynamically adapting to the traffic load, relies on a centralized decision making system that induces broadcasted control messages from the sink

MAC Protocol	**Traffic dependent**	**Decentralized decisions**	**Heterogeneous configurations**
B ,[Pol+04] X [Bue+06], MX [MH10a], Speck [WA06], MiX [MH10a]-MAC	☐	☑	☐
pTunes [Zim+12]	☑	☐	☐
LA-MAC [Cor+12]	☑	☐	☑
Contiki [Dun11], BoX [ML08]-MAC	☑	☑	☐
MaxMAC [HB10], AADCC [MH10b], ALPL [Jur+07], ZeroCal [Mei+10]	☑	☑	☑
BOX-MAC	☑	☑	☐
BEAM-MAC	☐	☑	☑
T-AAD proposal	☑	☑	☑

Table 2.2: Summary of state-of-the-art contributions : traffic dependency, required knowledge for decision making and possibilities of heterogeneous configurations without endangering network connectivity

node to all nodes. Moreover, it also maintains homogenous LPL configurations, thus failing to cope with non uniform traffic distributions in an energy-efficient manner (e.g., burst traffic upon event occurrence). Regarding ContikiMAC, dynamic and bursty traffic can be handled. But, this proposal relies only on waiting periods after each packet reception. Therefore, no adaptation on the sender is allowed and the same preamble and sampling periods are kept throughout the deployment, thus relying on homogeneous configurations that prevent any further energy savings.

Finally, solutions such as MaxMAC or AADCC appear as the most relevant in our targeted context, being both adaptive and traffic-aware. We therefore selected these solutions as best candidates for further comparisons during our evaluation campaign (see Chapter 4), as detailed throughout this manuscript.

2.3.3 Mobility Oriented Protocols

In WSN applications such as patient [Hac+14] or animal monitoring [Dyo+10], requirements for instance mobility and bursty traffic very often appear to be essential. In such applications, sensor nodes are attached to persons, animals or objects, while the readings (i.e., data packets) of the mobile nodes are delivered to the sink station over a multi-hop path which is consisted with static nodes. Moreover, the whole network might (physically) change, due to mobile nodes that modify the topology. In contrary to the traditional a priori known time-driven traffic patterns, event-driven networks face occasional, bursty and unanticipated multi-hop data transmissions. In wildlife monitoring for instance, the nodes (usually with limited-memory devices) operate under limited internet access for the majority of the time. When a network connection is detected, a surge of traffic should be handled. More specifically, the mobile nodes should immediately upload their stored data (bursts) at a more powerful device before losing again the connection (i.e., sink) [Zha+04]. Such sudden dynamic and bursty traffic cause certain anomalies in the network and fuel the research community to find appropriate solutions.

As previously reported, in this manuscript, we consider contention-based MAC protocols. Many MAC protocols that have been proposed in the literature (i.e., [DD13]) deal with mobility to some extent. However, to the best of our knowledge very few of them address the needs implied by the presence of variable and bursty traffic in a mobility-aware network. As a result, no successful WSN deployment with mobility handling and dynamic traffic has been proposed so far. Hereafter, we present the most recent and relevant approaches related to our investigation.

In [ZD10], the authors present MA-MAC (Mobility-Aware Medium Access Control), an extension of the X-MAC [Bue+06] protocol. In a static scenario, MA-MAC performs similar to X-MAC [Bue+06], while in a mobile environment, MA-MAC defines two thresholds to handle mobility in WSNs. The first threshold is triggered in order the mobile node to initiate a handover, while the second sets an upper limit (i.e., distance) that a mobile node should move before it establishes a new temporary parent. More specifically, the mobile node, being a transmitter, constantly evaluates the Received Signal Strength Indicator (RSSI) values of incoming ACK packets from static nodes. Thus, if the mobile node perceives that the distance between the current receiver and itself exceeds the first threshold, it initiates a neighborhood discovery procedure by transmitting broadcast data packets in which handover requests are embedded. Once it receives an acknowledgement from a new static node (before it reaches the second distance threshold) the mobile node enters into a handover mode to continue the transmission of data packets to the newly discovered node. MA-MAC introduces a new header in the payload part of the packet, depends on scheduling of nodes and network density, and relying on the two reported thresholds is fairly critical.

Mobility Aware RI-MAC (MARI-MAC) [DD12] is a contention based MAC protocol, an extension of the Receiver-Initiated RI-MAC [Sun+08] protocol. In MARI-MAC (like in MA-MAC) a distance threshold is defined to trigger and handle the handover procedure. MARI-MAC is based on strong assumptions such as the node mobility model that should known a priori, and the relationship between RSSI and distance can be stable. However, in real-word the received signal level used as a mobility indication does not provide fair accuracy to evaluate proximity, as it is reported in [HV09]. Moreover, the performance of receiver-initiated protocols is highly correlated to the density of sender nodes. Indeed, the more senders surround one receiver, is higher the probability of packet collision in the network and as a result latency may increase dramatically.

In In [DW14a], [DW14b], Dargie *et al.* present the MX-MAC protocol that follows the similar principles of the X-MAC protocol regarding to the medium access and low-power listening. MX-MAC, which allows a mobile node to transmit in burst once it gains access to the wireless medium. The protocol utilizes a Least Mean Square (LMS) filter that continuously evaluates the RSSI values of received ACKs from its temporary parent, and thus, if it detects a persisting deterioration in the link quality it initiates a handover procedure. Moreover, MX-MAC introduces three different types of MAC addresses: multicast, neighbor discovery, and unicast address. The MX-MAC protocol is particularly suitable for environments where the mobile nodes are few and its efficiency strongly depends on the network density. Indeed, if the number of neighboring static nodes is small, then the probability of discovering a new static relay node reduces significantly, (similarly to MA-MAC).

Kuntz *et al.* [Kun+13] propose X-Machiavel, a X-MAC-based solution. X-Machiavel has been designed to reduce the delay of accessing the wireless medium for the mobile nodes both under high and low contention scenarios. The authors consider that the data packets that originate from mobile nodes have higher value compared with the static nodes. Thus, X-Machiavel provides the mobile nodes with the privilege to “steal" the wireless medium from a static node that gained it earlier. X-Machiavel comes with two functionalities: *i)* in the first mode, where there is no traffic in the network (i.e., transmission between static nodes) *ii)* the second operation is based on the transmissions between static nodes. More specifically, when a mobile node expects to transmit a data packet, it first samples the medium. If it does not detect any signal, it follows the standard procedure, transmitting strobe packets during the preamble period prior the data packet. If it detects a preamble (i.e., second mode), it is allowed to take possession of the medium at the end of the ongoing preamble being sent by a static node. In order to do so, X-Machiavel specifies a delay (MIFS, MACHIAVEL Inter-Frame Space) that static nodes have to sample between the strobe and their data. The value of the MIFS delay may vary according to the time that a node should take to sample the channel. More specifically, a mobile node overhears the medium to detect a preamble's strobe from a transmission between two static nodes. Once it receives a strobe packet, it transmits its own data packet back to the transmitter. Afterwards, the transmitter static node, once it receives the data packet from the mobile node it continues sending its strobes till its receives an ACK from another static node and later it transmits its data packet. As a results, static nodes are forced to postpone their own data transmissions, which eventually leads to increase both 1-hop and end-to-end delays since X-Machiavel gives priority to the mobile nodes. X-Machiavel, is thus considered as a “non fair” contention based protocol for the nodes in a WSN.

MAC protocol	**Advantages**	**Drawbacks**
MA-MAC [ZD10], MX-MAC [DW14b]	traffic independent efficient handover mechanism	reactive protocol network density dependency designed for very small networks
MARI-MAC [DD12]	traffic independent	inaccurate proximity estimation network density dependency
X-Machiavel [Kun+13]	traffic independent hybrid protocol (reactive & proactive) overhead minimization (preamble-less)	underlying protocol dependency proportion of mobile to static nodes dependency non-fair contention-based protocol
MoX-MAC [Ba+14]	proactive protocol overhead minimization (preamble-less)	traffic dependent (passive protocol) unnecessarily consume energy (for static nodes)
MOBINET [Rot+11]	proactive protocol optimal next-hop selection	traffic dependent (passive protocol) increase of idle listening (energy consumption)

Table 2.3: Summary of state-of-the-art preamble-sampling based MAC layer contributions addressing mobility in WSNs

In [Ba+14], the authors present the MoX-MAC protocol, a mobile access scheme for X-MAC. The main concept of MoX-MAC is similar with X-Machiavel. Under MoX-MAC the mobile nodes do not transmit strobe packets during the preamble period in contrary to static nodes. In fact, when a mobile node expects to transmit a data packet, it samples the medium hoping to detect an ACK packet transmission, that originally is sent to a static node. Thus, if a mobile node receives the ACK packet, it waits until the end of the scheduled transmission, and afterwards, it transmits its data packet to the transmitter static node (that has transmitted the preambles previously). Otherwise, if no ACK packet is detected, it follows the default procedure of the X-MAC. Note that in the MoX-MAC the transmitter static node after the transmission of its data packet, keeps its radio turned *ON* for potential transmission of a packet from a mobile node. The efficiency of this approach strongly depends on the communication frequency between the static nodes. Moreover, if no mobile node transmits data packet, the transmitter static nodes unnecessarily consume energy by keeping their radio *ON* to potentially receive data packets from mobile nodes.

MOBINET [Rot+11] allows the mobile nodes to detect the surrounding static nodes (if there are any) in a passive listening mode. A mobile node when enters a static network, it builds a neighborhood table with destination addresses of the static nodes of its transmission range, by overhear the medium for transmitted packets from its temporary neighbors. Note that in order to delete the old entries corresponding to neighbors that could be out-of-range, each record of the table is associated with a Time To Live (TTL). Thus, once the TTL of a record expires, it is deleted from the table. Later, when the mobile node desires to transmit, it sends a data packet in unicast to one of the destination addresses listed in its neighborhood table. MOBINET comes with two methods, the random and selective method respectively. In the first approach, the next-hop selection is randomly selected among the ones available in the neighborhood, while on the other method the mobile node transmits to the "best" sensor located in its neighborhood in terms of number of hops for instance (depends from the routing protocol used by the static network).

2.3.4 Overview of Mobility-Aware Protocols

Most of the previously presented protocols may not satisfy our objectives of addressing bursty traffic in mobile environments, in a highly proactive manner and by attaining low 1-hop and end-to-end delay values under mobile environments.

Hence, as summarized in Table 2.3, MA-MAC and MX-MAC approaches are highly reactive solutions. Since MX-MAC requires a significant number of packet transmissions in order to estimate quality of the link before its establishment, its application induces potential delays or losses in the network. Furthermore, these solutions strongly depend on the network density and they are designed for small-scale networks. On the other hand, X-Machiavel even though being a traffic independent protocol, it strongly depends on features of the X-MAC protocol, such as strobe packets in the preamble, and moreover, it suffers under scenarios where we have more mobile over static nodes.

MoX-MAC and MOBINET being proactive protocols, appear as the most relevant to our targeted context. Moreover, the previously mentioned solutions are independent from the underlying MAC protocol. Indeed, they can be implemented both on top of strobe-based (e.g., X-MAC) and data-based (e.g., ContikiMAC) MAC protocols. We therefore have selected these contributions as best candidates for further comparison during our evaluation campaign, Chapter 5.

2.3.5 Context Description in Opportunistic Routing

After exhaustively studying the medium access schemes, we now focus on studying the impact of MAC with upper layers (since the PHY has hardware dependencies). More specifically, we aim at investigating the interactions between MAC and routing layer. Indeed, in typical WSNs, readings originated from sensor nodes are forwarded in a multi-hop fashion towards the sink. This process is usually performed through unicast communications, where packets are recursively sent to a designated next hop. In this approach, this next-hop is selected by the routing protocol, depending on various metrics such as link quality. We here rely on a different approach, where packets are sent opportunistically (i.e., anycast). Hence, a data packet is transmitted to the first potential forwarder (e.g., any neighbor closer to the sink in term of hops) acknowledging the corresponding message [Liu+09b].

In this manuscript, we investigate such opportunistic routing solutions operating with underlying LPL-based MAC mechanisms, intended to save energy through preamble transmissions (and acknowledgements) prior to any data communication [Can+11]. By increasing the number of potential forwarders, an opportunistic routing process has a straight impact on MAC layer performance. Indeed, several neighbors competing for the same packet transmission necessarily lead to faster (yet multiple) preamble acknowledgements. As a result, opportunistic routing allows for improved throughput, reduced end-to-end delays, balanced energy consumption between nodes in the network [Spa+12]. Finally, in a WSN context where properties of both wireless links (e.g., quality and symmetry) and of so constituted network (e.g., nodes mobility or failure) may greatly vary over time [Loh+13; Wan+12], the proposed approach is meant to increase the communications resilience, when compared to unicast-based routing protocols [Rot+11].

Originally designed to improve throughput performance in mesh networks [BM04; ZR03], opportunistic routing rapidly gained interest in other networking areas, due to the multiple benefits it provides. Indeed, by relying on a subset of potential next hop neighbors (i.e., forwarders) instead of a single one, this routing technique allows higher throughput and improved delay performance as well as increased resiliency. In the context of WSNs, opportunistic routing also gained a large interest from researchers [Lan+12], [Duq+13], as this approach addresses several critical issues, prominent in this field.

Network lifetime: Most WSNs are either powered by batteries or by energy-harvesting devices that can only provide limited power supplies. Energy resources are thus scarce and must be preserved at every layer of the protocol stack [Ana+09]. In this context, opportunistic routing protocols can help increasing network lifetime at two levels. First, by allowing several neighboring nodes to compete for the acknowledgement of a single message, the associated preamble length should be reduced. Indeed, the first eligible neighbor perceiving this preamble will send an acknowledgement. This will cause the interruption of the preamble transmission, and, thus, leading to reduced energy consumption at the transmitter size. Second, opportunistic routing techniques offer a wide range of possible paths toward the sink, while unicast-based routing protocols only rely on a single or a few one for each source node. This phenomenon leads to a better charge repartition and thus improved network lifetime, as has been shown in [Spa+12].

Communications performance: WSNs typically have limited throughput or delay performance requirements, as most of the transmitted data usually pertain sensors reading, represented as a couple of floating values. Yet, most WSN MAC protocols usually aim at reducing energy consumption through radio-duty cycling. One of the most significant effects of this approach is a drastic drop of throughput, in parallel with an increased average delay [Can+11]. Opportunistic routing techniques can help counterbalance this effect. Indeed, by utilizing a quicker preamble acknowledgement and a more efficient charge repartition, a higher throughput and lower delay performance is achieved [Duq+13]. Moreover, this aspect helps managing networks that operate high traffic loads (e.g., large-scale networks, high-frequency sensing, video streams).

Topology changes: Due to their low-power radio and the environment they commonly evolve in (e.g., outdoor, remote, hostile), WSNs are usually subject to link quality changes and node failure [Loh+13], thus leading to topology changes over time. Also, the network itself may evolve due to nodes mobility or failure [Wan+12]. Under such situations, unicast-based routing protocols have to automatically adapt themselves through routing path discovery and re-construction [AA09]. This step takes time and probably requires the exchange of additional control messages. On the other hand, by utilizing opportunistic routing techniques, the disappearance of a link is compensated by the remaining neighbors belonging to the same subset of potential forwarders [Was+07]. This approach is much more resilient than traditional routing techniques and thus well-suited for WSN specificities.

In this study, we introduce a mechanism that mitigates packet duplication inherent to opportunistic routings through smart MAC layer heterogeneous configuration. In order to perform it, information related to the number of neighboring potential forwarders is required. Many other mechanisms use this information [Bea+11; HV08], and in such context the learning process can thus be performed once for all mechanisms.

2.3.6 Overview of Packet Duplication Issue in Opportunistic Routing

Several research proposals have been developed to prevent multiple nodes to acknowledge a single packet simultaneously in opportunistic routing. As this phenomenon is not specific to WSNs, solutions have been proposed to cope with it in opportunistic networks relying on WiFi. Most of those proposed to face the present problem through an agreement between the sender and the subset of neighbors, to select a single forwarder (that will be valid for this single transmission only) [Cha+07]. This approach fits with WiFi specificities, but cannot be translated to WSNs. Indeed, it induces too high control traffic overhead for networks with such low throughput as WSNs. Especially, when considering duty-cycling techniques at the MAC layer, such agreements may undermine the expected saved energy.

In ExOR [BM05], all potential forwarders have their own transmission time slot sorted by routing progress, and forward only if they do not have overheard any transmission from other nodes. This process selects the best available next hop in a distributed way and avoids duplicate forwarding, but also implies increased control traffic for time-slot allocation. Also this approach relies on time synchronization of the nodes, which is not necessarily possible nor desired in WSNs, due to the corresponding overhead of time-synchronization mechanisms [SY04]. In addition, most WSNs rely on *N to 1* communications where most packets share the same destination. In this context, ExOR would induce all potential forwarders to share the same time-slots, leaving the problem unsolved.

We selected some of the most interesting approaches either for their performance or their realism in a WSN case of use.

In [Lan+12], Landsiedel *et al.* present ORW, a complete opportunistic routing solution for WSN. Concerning the unique forwarder problem, they proposed to solve it through a double acknowledgement mechanism. Indeed, as in classical opportunistic routing solutions, each potential forwarder catching a preamble will immediately acknowledge it. Under ORW, these nodes will then launch a random timer and after this timer expires they will send a second preamble acknowledgement. The selected unique forwarder will thus be the first among the potential forwarders to acknowledge the preamble, but also the one whose second acknowledgement was the faster.

In [Duq+13], duplicates are filtered out at the routing layer, when non-disjointed paths are used by several duplicates. This solution reduces unnecessary forwarding, but only partly solves the problem as disjoint paths may be used by packet duplicates. Also, duplicated packets are filtered only at common part of their routes. Thus, additional traffic is not avoided.

2.4 Conclusions

In this Chapter, we have set up the stage for the rest of this manuscript. To this aim, we carry out a thorough study over four major and representative conferences (i.e., ACM/IEEE IPSN, ACM SenSys, ACM MobiHoc and ACM MobiCom) in Ad-Hoc and WSNs in order to derive the current tendency of the validation methodology that authors follow, and especially to what extent experiments on testbeds have become a must for performance evaluation of new (ad-hoc and sensor) network algorithms and protocols. We therefore studied 674 articles in total of last six years from 2008 to 2013 where 596 are related to Ad-Hoc & WSN. Among the large set of statistics that we compiled our investigation shows that few experimental approaches through custom testbeds are indeed reproducible (i.e., 15.9%).

Moreover, we performed a thorough literature review related to medium access procedures, with a special focus on preamble-sampling MAC protocols. More specifically, we investigated the area of runtime traffic auto-adaptation and auto-configuration. We also presented the state of the art for MAC protocols related to mobility aware networks.

Preamble-sampling protocols have been evaluated in number of WSNs scenarios and utilized in number of real-world deployments [LL07], [Dyo+10], [Zha+04] [Chi+10a]. As presented in this Chapter many MAC protocols that have been proposed in the literature deal with auto-adaptation and mobility to some extent. However, to the best of our knowledge very few of them address the needs implied by the presence of variable and bursty traffic in a dynamic wireless sensor network. As a result, no successful WSN deployment with mobility handling and variable traffic has been proposed so far. Thus, new asynchronous MAC protocol should be defined to increase the network performance and minimize the energy consumption.

Furthermore, in this Chapter, we introduced the concept of opportunistic routing and the impact of MAC configurations to it. Indeed, we presented the packet duplication issue in opportunistic routing due to the anycast-based transmissions and some works from the literature on how to mitigate it.

As exposed earlier in this chapter, performance evaluation methods highly vary among existing works. Before detailing our contributions related to MAC layer in dynamic networks, we first describe our protocols evaluation methodology.

Chapter 3

Can experiments lead to scientific results? A thorough experimental study

As discussed in the previous Chapter, the recent tendency to have algorithmic and protocol proposals facing real environments, it is questionable whether the so obtained results should be considered as scientific or empirical ones. In this Chapter, we explore the role of testbeds in the development procedure of protocols or applications for WSNs and IoT. Indeed, we highlight to what extent the addition of experimentations can significantly improve the value of performance evaluation campaigns. Moreover, we investigate the complementarity between simulation and experimentation studies by evaluating latest features available among open testbeds (e.g., energy monitoring, mobility). In this context, we insist on how simulations and experimentations can be efficiently and successfully coupled with each other in order to obtain reproducible scientific results, rather than sole proofs of concept. Indeed, we especially highlight the main characteristics of such evaluation tools that allow to run multiple instances of a same experimental setup over stable and finely controlled components of hardware and real-world environment. Our results show that such open platforms, can guarantee a certain stability of hardware and environment components over time, thus, turning the unexpected failures and changing parameters into core experimental parameters and valuable inputs for enhanced performance evaluation.

Contribution

This chapter presents the following contributions:

1. We first introduce FIT IoT-LAB, a very large-scale open experimental testbed.
2. We then exhibit through a series of experiments led on FIT IoT-LAB open testbed, how to conduct meaningful experiments under real-world conditions, and moreover, we aim at demonstrating to what extent some of the simulation setup and conditions from reality could be emulated.
3. Finally, we describe how successful testbed experimentations could be translated into real-world deployments, by taking into consideration the number of unpredictable issues such as the link quality and stability, network density or drifts of mobile robots that arises during the experimental procedure.

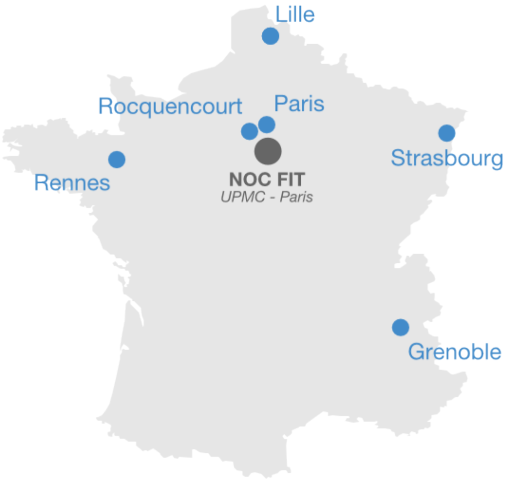


Figure 3.1: FIT IoT-LAB: A very large scale WSN and IoT testbed.

3.1 Facilities of FIT IoT-LAB Platform

FIT IoT-LAB platform is part of the FIT[§] experimental platform, a set of complementary components that enable experimentation on innovative services both for academics, industrial researchers and engineers. In fact, FIT IoT-LAB is the evolution and extension of the SensLAB [Roz+11] project (i.e., 2010-2013). It provides facilities suitable for evaluating small wireless sensor devices and heterogeneous communicating objects. FIT IoT-LAB platform offers to users to control the deployed sensor nodes and direct access to the gateways to which nodes are connected, and thus, allowing researchers to monitor nodes energy consumption and network-related metrics, e.g., end-to-end delay, throughput or overhead. Moreover, it provides quick experiments deployment, along with easy results collection, evaluation and consequently analysis.

In this section, we introduce FIT IoT-LAB platform, a large scale and an open access multiuser testbed. Indeed, we provide a detailed description of hardware (e.g., motes and radio), testbeds and platform.

FIT IoT-LAB is composed of 2728 wireless sensor nodes distributed over six different sites in France (see Figure 3.1), Inria Grenoble (928), Inria Lille (640), ICube Strasbourg (400), Inria Rocquencourt (344), Inria Rennes (256) and Mines-Telecom Paris (160). Table 3.1 provides a detailed overview of nodes distribution over six sites in France. The wireless devices are allocated within different topologies and environments (e.g., isolated or real-world) throughout all sites. Finally, some of the platforms offer mobile robots to evaluate mobility-based solutions.

[§] FIT - http://fit-equipex.fr/

Type of node	**Grenoble**	**Lille**	**Rocquencourt**	**Strasbourg**	**Rennes**	**Paris**	**Total**
WSN430 (800MhZ)	256	-	-	256	-	-	512
WSN430 (2.4GhZ)	-	256	120	-	256	-	632
Cortex M3	384	320	24	120	-	90	938
Cortex A8	256	-	200	24	-	70	550
Open Host	32	64	-	-	-	-	96
Total	928	640	344	400	256	160	**2728**

Table 3.1: Testbeds distribution of nodes

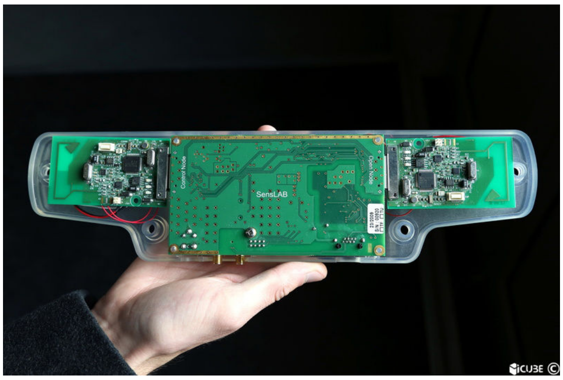


Figure 3.2: Detailed view of an IoT-LAB node and its gateway.

3.1.1 An IoT-LAB hardware

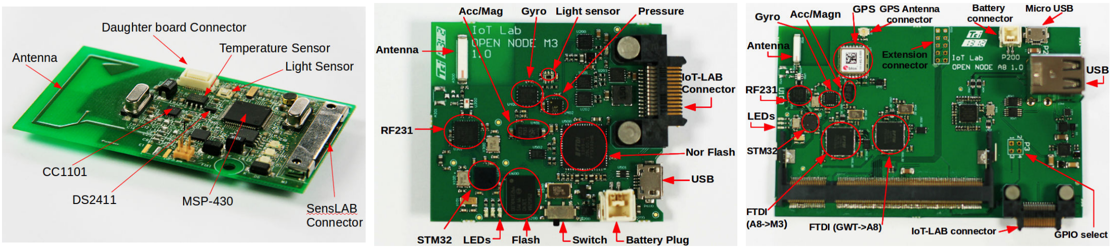
A global IPv4/IPv6 networking backbone provides power and connectivity to all FIT IoT-LAB nodes and guaranties the out of band signal network needed for command purposes and monitoring feedback. Furthermore, FIT IoT-LAB infrastructure comprises a set of nodes, in Figure 3.2 an overview of an FIT IoT-LAB node is depicted, where each node consists of three main components, the open wireless sensor node, gateway and control node respectively. More specifically, a FIT IoT-LAB node is composed of the following components:

- **Open wireless sensor node:** (able to retrieve environmental information such as sound, light or temperature) that is dedicated to users during their experimentation. The node is fully open and the user is granted a full access to the memory. This implies that variety of operating systems can be flashed and run on top of the open node (the operations are handled using a remote access to reboot and (re)load the firmware).

- **Gateway:** guarantees the connection to the global infrastructure of the IoT-LAB in order to flash, control, monitor the open node and eventually to retrieve the measurements. The gateway also handles the open node serial link if the node is set to be a sink node.

- **Control node:** is similar to the open node and is used to interact, passively or actively, with the Open wireless sensor node. Moreover it provides power supply, and monitors consumption and sensors values during experiments.

In order to meet with the researchers requirements, for instance accurate energy consumption monitoring or reproducibility, FIT IoT-LAB offers various boards, WSN430, Cortex M3 and A8 node (see Figure 3.3). More specifically, the sensor nodes are equipped with different processor architectures and radio chips (i.e., MSP430 with CC1101 or CC2420, and STM32 and Cortex A8 with AT86RF231 respectively). This makes FIT IoT-LAB compatible both with the IEEE 802.15.4 standard [Ieea] and with open Medium Access Control (MAC) [Can+11], [DD13] protocols, and moreover, suitable for real-world WSN and IoT deployments [Loh+13], [Dil+11]. Defining complementary and heterogeneous testbeds with different node types, topologies and environments allows for coverage of a wide range of real-life use-cases, and thus, researchers may expect different behaviors from each platform. Furthermore, FIT IoT-LAB platform comes with all the necessary tools (e.g., RSSI mapping, link stability and quality analyzer) to ensure that the environmental conditions of the facilities are fulfilled before lunching experiments. Hence, with IoT-LAB experimental facilities, researchers may perform real experiments in small or large scale platform in order to evaluate and analyze their solutions, by translate a real-world deployment situation in it.



(a) WSN430 board. (b) M3 board. (c) A8 board.

Figure 3.3: The IoT-LAB nodes are protected with a specific box designed on purpose.

Figure 3.4: A view of ICube's platform.

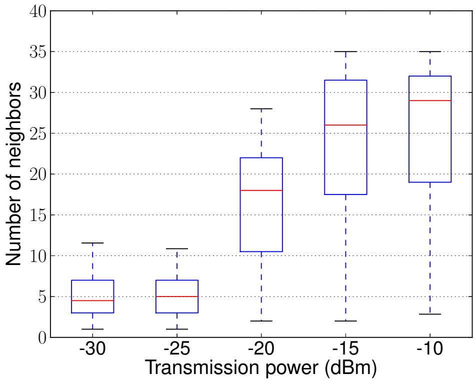
3.1.2 ICube's platform

In this investigation we focus on the site of ICube Strasbourg. ICube's platform offers two testbeds; the one is structured as a 3D grid with 10 lines and 8 columns, distributed in three layers of 80 nodes each, thus, in total 240 fixed WSN430 nodes, while the other testbed consists of 64 M3 and 14 A8 nodes respectively, distributed in two layer of 39 each. Furthermore, ICube offers 44 mobile nodes, 7 WSN430 and 37 M3 based-boards respectively (see Figure 3.4). Considering various technical constraints such as autonomous navigation with a certain minimal speed, carrying and powering an IoT-LAB node (with size of $32 \times 8 \times 6\ cm$), automatic docking, at least three hours of battery autonomy and realtime monitoring of the robot, drove the engineers of IoT-LAB to TurtleBot2[¶] robots where Robot Operating System (ROS[‖]) framework (e.g., navigation stack with the AMCL module) is used. TurtleBot2 is able to replay trajectories with a list of checkpoints. Indeed, the users may choose among the predefined trajectories (e.g., line, square, triangle) or design their own circuit to perform their experiments. The robots move with speed of $30 cm/sec$ (similar to human walk), and may communicate both with static nodes and other mobile nodes. Finally, each TurtleBot2 has a dedicated docking station for charging and reserving purposes.

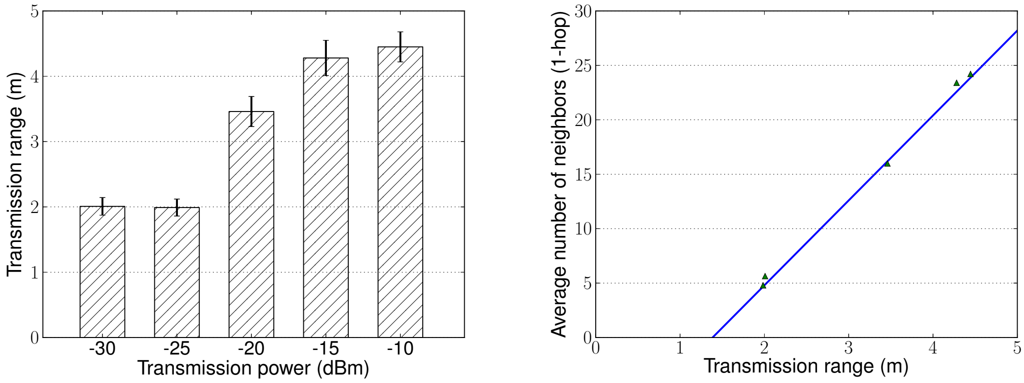
Even though the open testbed is very complete, still in the following section, we will present through a number of experimental campaigns that it misses some important guidelines and features, and to show how to retrieve scientific and meaningful results instead of proofs of concept only, by utilizing a not complex set of code.

[¶] http://www.turtlebot.com/

[‖] http://www.ros.org/



(a) Average number of 1-hop neighbors.



(b) Transmission range.

(c) Correlation evidence.

Figure 3.5: Neighborhood density (left), transmission range (center) and the correlation between both (right).

3.2 Thorough analysis of an IoT testbed

In this section we present our thorough experimental study over the testbed located in ICube Strasbourg. In our set of experiments, we chose the 80 nodes of the middle layer to work on our project. All 80 nodes are randomly selected as data sources and implement a time-driven application model by broadcasting in CBR mode one packet every 60 seconds. We chose a 10 *bytes* data size, which corresponds to the general information used by monitoring applications (e.g., node ID, packet sequence, sensed value) [Kdo+12]. We run a number of experiments by considering different transmission power values (i.e., -10 dBm to -30 dBm) in order to study the impact of the radio model in transmission range and link quality. The experiments lasted for 150 minutes while 7900 transmissions occurred in each round.

In this campaign, we aimed at removing assumptions one at a time. Thus, we perform our evaluation within an idealistic scenario where we keep our application as simple as possible and no routing protocol is running in the network. As a result, this scenario will allow us to focus on the performance of the open testbed by affecting the results at minimum. At the MAC layer, we employed X-MAC [Bue+06] protocol with sampling frequencies at 125 ms, 250 ms *and* 500 ms. For convenience in terms of complexity and time consumption, we implemented our project on top of the Contiki OS [Dun+04] since the code with COOJA simulator remains unchanged. However, there are various available choices for the users, such as TinyOS [Lev+04] or OpenWSN [Wat+12]. The details of the experimental setup are exposed in Table 3.2.

Topology parameters	Value
Testbed organization	Regular grid ($10\ m \times 8\ m \times 3\ m$)
Deployed nodes	80 fixed sensors
Number of sources	80
Node spacing	One meter
Experiment parameters	**Value**
Duration	150 *min*
Application model	Time-driven: CBR 1 *pkt/min*
Type of Transmission	Broadcast
Number of events	7900 *pkts*
Payload size	10 *bytes* (+ 6 *bytes* MAC header)
MAC model	X-MAC [Bue+06]
Sampling frequency	(125, 250, 500) *ms*
Hardware parameters	**Value**
Antenna model	Omnidirectional CC1101
Radio propagation	868 *MHz*
Modulation model	GFSK
Transmission power	$(-10, -15, -20, -25, -30)$ *dBm*
Battery	880 *mAh*, 3.7 V

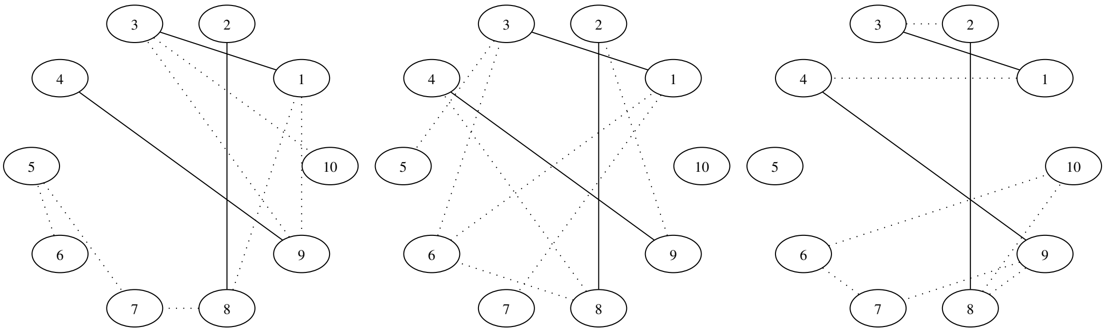
Table 3.2: Experimental setup

Hereafter, we present the results of our investigation over an open testbed in terms of the neighborhood density, stability and quality of the links as well as accurate energy consumption, and drifts over the predefined trajectories [Pap+13].

3.2.1 Selecting the transmission power

To estimate the density in the network, we count the average number of neighbors per node. To do so, we calculate the successful symmetrical packet transmissions among the nodes for various transmission power values (i.e., from $-10\ dBm\ to\ -30\ dBm$). In our campaign, we decided to keep high-quality symmetrical links only (i.e., over than 90% of successful receptions in both ways) and below 10% the confidence interval, in order to provide a fair analysis. In Figure 3.5a, the average number of neighbors per transmission power is depicted. As expected, the higher the transmission power, the more neighbors a node has, and consequently higher the density in the network. It is worth to point out that the pairs $(-30,\ -25)$ *dBm*, and $(-15,\ -10)$ *dBm* present similar results. In order to comprehend this behavior, we calculated the transmission range versus transmission power by employing the *Euclidian distance equation.* As can be observed from Figure 3.5b, the transmission range of the nodes follow similar trend with the density one. As a result, unlike most radio propagation models used in simulation, the increase of transmission power is not linearly reflected in the transmission range (and thus, in our testbed, nor in the neighbors density). In this context, using a realistic environment helps researchers getting closer to real deployments, in a way that cannot be obtained through simulations.

To better understand the previously obtained results, we study the correlation between the transmission range and the average number of 1-hop neighbors. First, we use a scatter plot (see Figure 3.5c) that clearly shows the linear correlation between the two distributions. Actually, through our experimental data, we obtained a linear correlation coefficient of 0.98. Thus, we simply use the *least squares* fitting method, and extract the following relation: $f(x) = -10.8 + 7.8x$ with x being the transmission range and $f(x)$ the resulting average of 1-hop neighbors (see at the blue curve in Figure 3.5c). Although the increase of transmission power is not directly linearly reflected in the transmission range (but rather by several linear stages), this simple relation seems to remain quite robust despite the experimental noises.



(a) Wednesday (Night). (b) Thursday (Morning). (c) Thursday (Night).

Figure 3.6: Links stability over time.

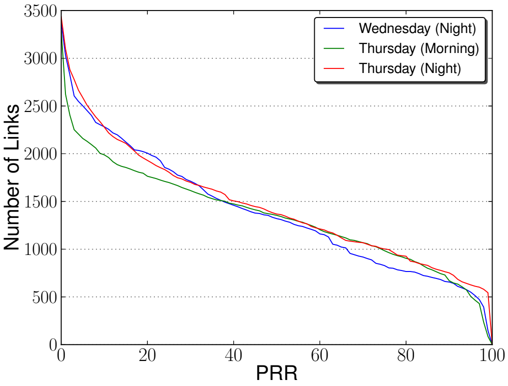


Figure 3.7: Links quality over time.

3.2.2 Assessing link stability & quality

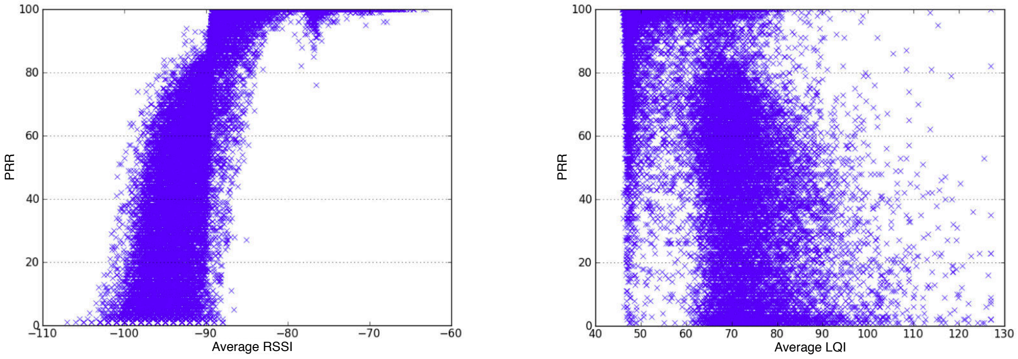
There are certain situations neither occur during the simulation nor simple experimental deployments using only few nodes. Among those, the presence of links with low quality induces packet loss and, therefore, may introduce a bias in the measurements. Hence, in realistic deployments researchers should take into account that most of the communication links are unstable, unreliable or even unidirectional. However, most of routing and MAC protocols require bidirectional links so that two nodes can exchange information (e.g., data, acknowledgements), thus, it is essential to evaluate wether this assumption can be preserved in testbeds.

Several criterions may be considered to represent and evaluate the quality of a link. Among those, we focus on Packet Reception Rate (PRR) and its symmetry, since it appears to be one of the most accurate and straightforward way to do so.

We, thus, measured the PRR of each link in the testbed by setting each node broadcasting in turn 100 messages and computing the number of subsequent receptions. Thus, we were able to precisely compute the PRR of each link as well as its bidirectionality. In this scenario, we considered a transmission power of $-20\ dBm$, and evaluated the stability of links over time. In particular, we repeated this experiment three times every 12 hours, once on Wednesday evening and twice on Thursday (morning and evening respectively). Then, we analyzed the links properties in order to study their stability over time. In Figure 3.6, the nodes are depicted in a circle for visualization purpose. Ten nodes are a subset of the total 80 nodes. As it can be observed, three connected components (little more than 30% of the links) remain stable over time (links 1-3, 2-8 and 4-9, plain arrows) while variations occur among the remaining links (dotted arrows).

Date	Neighbors	Confidence Interval
Wednesday (night)	13.95	2.16
Thursday (morning)	13.38	1.9
Thursday (night)	15.34	1.94

Table 3.3: Average number of 1-hop neighbors over time.



(a) Correlation between RSSI and PRR.

(b) Correlation between LQI and PRR.

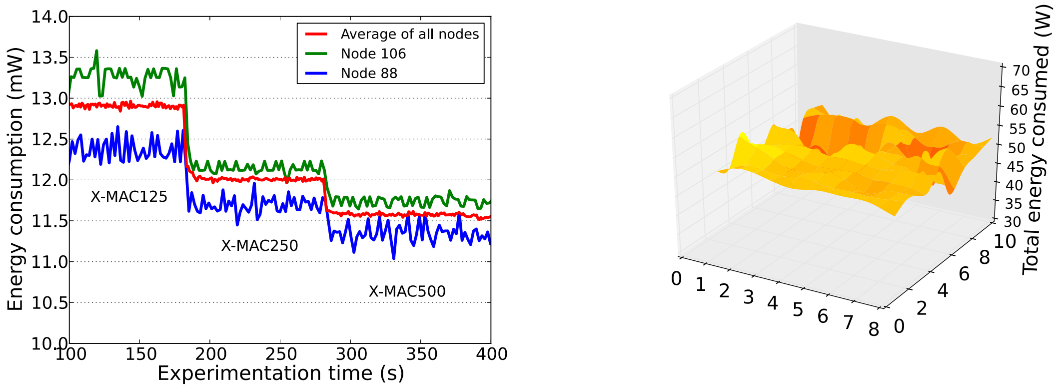
Figure 3.8: Interest of RSSI and LQI values to evaluate link quality and symmetry.

As previously mentioned, in real deployments the link quality and symmetry may significantly vary over time. Figure 3.7 displays a detailed representation of link quality throughout the whole experimental procedure, in particular the total number of bidirectional links per PRR is presented. The results show that all three experiments follow similar trend. In fact, more than 500 bidirectional links present PRR above 90%. However, this behavior is very hard to reproduce in simulators. Thus, researchers often need to remove this assumption, while preserving identical density over time (in order to perform a meaningful evaluation). As it is presented in Table 3.3, ICube's platform preserves certain stability in terms of network density, (i.e., approximately 14 neighbors) with PRR above 90% in all three experiments. Then, the density criterion remains stable while the links are changing. Thus, it allows researchers to remove assumptions one at a time, to better evaluate their protocols. It also shows that the testbed can be used either way. Indeed, it can be used as a scientific tool, producing scientific results that can be reproduced and slightly vary over time, even with a changing topology of identical density. When enlarging the set of considered links for experimentation, users would introduce more and more randomness due to the lower quality of those links, thus reaching real life conditions and meeting some of the requirements for the validation of a proof-of-concept and iterative prototyping.

3.2.3 Selecting the quality radio links

As abovementioned, many protocols at every layer of the communication stack require bidirectional links to operate. While this assumption can easily be guaranteed in simulation, it does not stand in experimental conditions. However, researchers often need to keep only a set of stable, high-quality and bidirectional links while testing their solutions.

In this context, we, thus, evaluated the connection between link quality as stated before (i.e., PRR and bidirectionnality) and common signal quality indicators provided by the radio. Indeed, most radio chipsets provide two indicators that could be used for this purpose, the Received Signal Strength Indication (RSSI, expressed in dBm) and the Link Quality Indicator (LQI). The RSSI value is an estimate of the signal power level in the radio channel while on our testbed the LQI estimates how easily a received signal can be demodulated.



(a) Real time energy consumption.

(b) Total energy consumption in 3D representation.

Figure 3.9: The average (left) and total (right) energy consumption in various sampling frequency configurations.

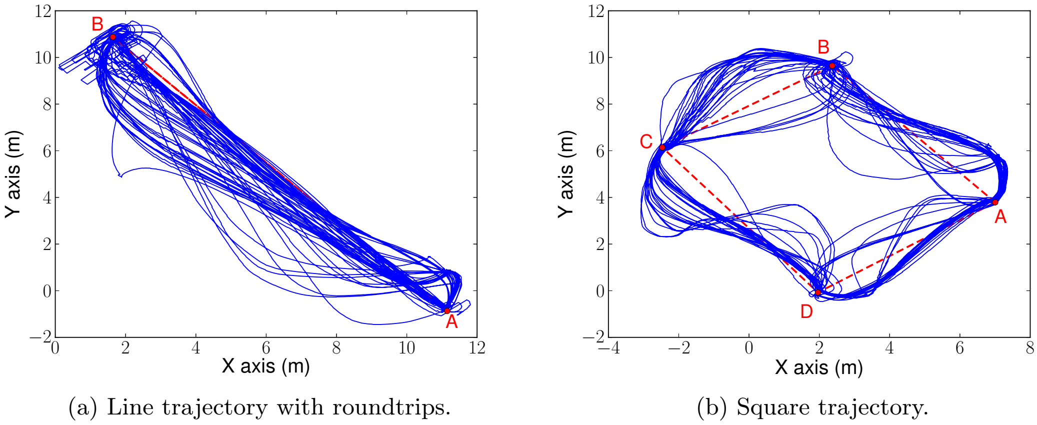
We conducted experiments in which each node broadcasts a hundred of messages in turn so that we can determine together average RSSI, LQI and PRR for each link. As depicted in Figure 3.8 , the RSSI can hardly help estimate the PRR of a link, as there is no strict relation between both parameters (e.g., a RSSI of -95 dBm can induce a link with a PRR spanning from 0% to 80%). A similar conclusion can be drawn for the LQI value. However, the RSSI can help to select a set of high-quality links. Indeed, links with a RSSI abobe -83 dBm all display a PRR greater than 90%. The LQI, however, can not ensure the same guarantee.

3.2.4 Evaluating the energy consumption

Simulators provide an estimation of the energy consumption of sensor nodes. Indeed, they often provide a linear model, where all nodes follow the same smooth and predictable energy consumption pattern, and consider a subset of energy consumption causes only (e.g., nor sensors or memory). On the contrary, in reality two identical nodes may follow different consumption patterns, and some minor changes in the protocol configuration may present unexpected impact. In fact, two distinct nodes would have various energy consumption profiles, either they embed different component (e.g., memory, CPU). Or, even though they could come with two identical hardware configurations, it can also happen due to the electric components manufacturing, their fixation over the silicium board, and so on.

In this manuscript, we utilized the Energest [Dun+07] module of Contiki OS to retrieve the energy consumption results. This energy estimation module maintains a table with entries for all components, the CPU, and the radio transceiver. Each table entry contains the total time that the corresponding component has been turned on, more specifically, it monitors in real-time the radio and CPU usage by saving the duration spent in each state (e.g., transmitting, receiving data, awaken, sleeping). This information is then combined with the energy values that are detailed in the component datasheet for each state in order to provide an accurate calculation of energy consumption per node.

In this experiment, we want to demonstrate how the consumed energy can be accurately and efficiently retrieved in open testbed. To this aim, we considered the example of X-MAC protocol with various sampling frequency configurations, 125 ms, 250 ms and 500 ms (i.e., X-MAC125, X-MAC250 and X-MAC500 respectively) and we evaluated their energy consumption impact. More specifically, we run an experiment where the nodes were constantly sampling the medium and decreasing their sampling frequency every 100 seconds, from $125ms$ to $500ms$. Figure 3.9a presents the accurate energy consumption for two distinct nodes and average for the



(a) Line trajectory with roundtrips.

(b) Square trajectory.

Figure 3.10: Robot's drifts over predefined trajectories.

whole network. Overall, there is little difference (i.e., close to 0.5 mW) among the nodes, which is due to the same embedded hardware components. It is worth to point out that in the former case, energy profiles can be first established. Then, the end user would choose nodes having more or less the same profile in order to get significant results. In the latter, heterogeneous nodes can be picked in order to reflect what will be encountered when facing reality through deployments.

We also performed a mapping of the energy depletion throughout the network, in order to identify its repartition and disparity among the nodes. To do so, we measured the total energy consumption of each node for the total duration of an experiment. The results of this evaluation are displayed in Figure 3.9b. Looking more closely, the nodes are having heterogeneous behavior, meaning that there are differences in energy consumption (i.e., from 35 W to 50 W) which can be seen as a drawback from testbeds. This difference should be due to the used hardware since, with the same setup in simulation the nodes would consume precisely the same. Hence, this anomaly brings us step towards real deployments where the researchers may have to deal with number of unexpected behaviors. As a result, it can be seen as one of the advantages of testbeds compared to simulators, as they allow us to get a flavor of real deployments without having to modify the code.

3.2.5 Assigning and planning trajectories for mobile robots

In this subsection, we investigate the ability of robots on how accurately they follow the predefined trajectories that are available to the users. More specifically, we aim on showing the drifts that the robots may present due to the potential miscalculation of the navigation system that the robots are employed with. Hence, for this campaign we utilized two TurtleBot2 robots. We set the robots to replay over a line (from point A to B and vice versa) and square (A, B, C and D coordinates) trajectories for 90 minutes to obtain a large set of data related to the coordinates of the robots (ten samples per second). To calculate the robot's drift, we first determine the equation of the line ε (i.e., $\overrightarrow{AB}$) (3.1), considering the given coordinates, $A_1(x_1, y_1)$ and $B_2(x_2, y_2)$ for instance. We then, calculate the distance $d(M_0, \varepsilon)$ (3.2), between position $M_0(x_0, y_0)$, position of the robot, and the line ε.

$$\varepsilon = Ax + By + \Gamma \tag{3.1}$$

$$d(M_0, \varepsilon) = \frac{|Ax_0 + By_0 + \Gamma|}{\sqrt{A^2 + B^2}} \tag{3.2}$$

Drift	**AVG**	**5 PCTL**	**Median**	**95 PCTL**	**CI**
Line (m)	0.576	0.018	0.342	1.856	0.005
Square (m)	0.469	0.013	0.248	1.406	0.004

Table 3.4: Trajectory drift measurements

In Figures 3.10a and 3.10b the drifts of the mobile robots with respect to the predefined circuits (i.e., line and square) are illustrated. As can be observed, the robots replay the line trajectory with a drift of 0.57 *m* while the square circuit with 0.46 *m* in average. The detailed statistical results are presented in the Table 3.4. This drift anomaly could be explained at two levels, the software and hardware respectively. On the one hand, the AMCL module, the navigation stack of the ROS framework, computes the robots path in order to smooth the trajectories, as a result, the recorded traces are not in line with the predefined checkpoints-based trajectories. On the other, as can observed the mobile robot present a slight drift among its loops over the very same circuit. This hardware-related drift is introduced due to the sensors odometry that employed by the navigation system of TurtleBot2. However, the 3D camera (with the range detector sensor) that should handle the odometry drift, lacks in open-space and large-scale environments where not enough landmarks exist to compute the path, as at ICube's platform. In fact, the maximum distance that the 3D camera may reach is 8 meters, thus, we observed that in certain positions the robot is too far from the wall (i.e., the landmark). We, thus, assume that the robots may present a better behavior in corridor-based scenarios. As a result, the miscomputation of the AMCL navigation in conjunctions with the sensors odometry navigation explains the drifts in our experiments.

This phenomenon appears to be to the detriment of mobile robots that carry wireless sensor nodes, thus, raising the question of cost versus accuracy versus quantity of robots in our community. Overall, we aimed at showing that various hardware and environment parameters can be controlled and kept stable over multiple instances of a same experimentation setup (e.g., node radio coverage, trajectories of mobile robots). Open testbeds therefore allow researchers to face real-world conditions while guaranteeing the scientific nature of their observations and results. On the contrary, the simulators and emulators by employing mobility plug-ins, such as BonnMotion [Asc+10], allow the mobile nodes to follow and reproduce accurately the predefined (or random) trajectories without imposing drifts in their traces.

3.3 Conclusions and Perspectives

Performance analysis of newly designed algorithms and protocols is extremely desirable for efficient WSNs deployment. Simulation and experimental evaluation methods are essential steps for the development process of protocols and applications. Nowadays, the new solutions can be tested at a very large scale over both simulators and testbeds. On the one hand, simulators and emulators allow researchers to isolate or simplify some assumptions (e.g., radio propagation) by tuning configurable parameters to serve proof-of-concept requirements. Therefore, simulators being more suitable for evaluating and comparing the solution with its competitors from the state-of-the-art. On the other, a testbed is a platform for experimentation which allows for rigorous and transparent evaluation of new protocols, and reflects some potential anomalies that their proposal may show later. Testbeds and simulators are two crucial and complementary design and validation tools for achieving a successful real-world deployment (see Table 3.5). Theoretically development process should start from the theoretical analysis by providing bounds and indication of its performance, be validated and verified by simulations and finally confirmed in open testbeds. Hence, once the entire procedure is successfully done and the performance results show coherence, then researchers could push their solution to engineers in order to proceed with real-world deployments.

	Simulation	**Experimentation**
Radio links	• high-level reproducibility • theoretical models	• low-level reproducibility overall (high-level for high-quality links only) • real-world radio environment
Network topology	• high-level reproducibility	• low-level reproducibility (most of links being of mid or low quality)
Node mobility	• high-level reproducibility	• mid-level reproducibility (drifts being induced)
Energy consumption	• mid-level accuracy	• high-level accuracy

Table 3.5: Simulations and experiments: complementary approaches for validation of WSN and IoT solutions.

We have demonstrated number of values of a testbed that can be added to WSN and IoT protocol evaluation, and moreover how it can be efficiently and successfully coupled with simulations. We investigated the node radio coverage based on the transmission power, as a result, allowing users to select the most suitable transmission power for their experiments. Moreover, we exposed how both real time and total energy consumption can be accurately monitored, and to what extent the link quality and stability assumptions can be removed or not, at the researchers choice. Furthermore, we presented the accuracy of the robots to replay the predefined trajectories. Hence, the evaluation campaign of WSN protocols can go one step further towards real-world deployment by removing the previously mentioned assumptions, at little time cost and with limited complexity.

Finally, we highlighted to what extent open testbeds can produce scientific results and not only proofs of concept. Indeed, such open facilities allow researchers to run multiple instances of a same experimental setup over stable and finely controlled components of hardware and real-world environment.

In following scientific works, we aimed at applying the previously presented methodology. Indeed, it is especially valuable to capture both simulation and real effects of runtime adaptations at the MAC layer such as in our proposed T-AAD protocol.

Chapter 4

Lightweight Traffic Auto-adaptations for Low-power MAC Protocols

In typical WSN deployments, the traffic load may vary greatly over time and from one node to another. Consequently to the traditional event or time-driven a priori known traffic patterns, those networks face occasional, bursty and unanticipated multi-hop data transmissions where multiple packets are transmitted in a row. These situations may occur when some nodes stores the data instead of directly sending it, for example due to network disconnections or radio channel unavailability. This specific case is rarely addressed by preamble-sampling MAC protocols. Indeed, most MAC protocols are designed to address constant traffic, by preserving the same configuration for all nodes in the network, throughout the deployment. Such homogeneously configured solutions avoid network disconnections and isolated nodes, yet lead to long periods of channel occupancy, increased delays and energy-consumption. We here present T-AAD, a Traffic Auto-ADaptive mechanism, specifically designed to address the previously reported phenomenon through the introduction of heterogeneous node configurations in the network. T-AAD dynamically and locally adapts the MAC configuration depending on the actual and expected traffic load, without endangering network connectivity nor overall network performances. T-AAD is therefore compliant with LPL based MAC protocols. In this Chapter, we evaluate our scheme and we demonstrate that it provides significant gains in terms of delay and energy consumption, when compared to respective research work available in the literature (e.g., MaxMAC, AADCC).

Contribution

This chapter presents the following contributions:

1. We first introduce T-AAD, a preamble-sampling compliant MAC protocol that automatically adapts the MAC parameters at runtime.
2. We then demonstrate through an analytical study of its behavior that it quickly adapts to traffic variations. Thus, we show to what extent it allows for reduced energy consumption at both receiver and sender sides, along with delay and channel occupancy, when compared to CSMA-based solutions;
3. Finally, we perform a thorough performance evaluation, both through simulation and experimental study over FIT IoT-LAB [Pap+13] testbed. In addition, we compare our mechanism both with a statically configured network using X-MAC [Bue+06] and an auto-adaptive mechanism such as AADCC [MH10b].

4.1 Introduction

In application such as wildlife monitoring [Dyo+10], [Tho+04], [Zha+04] or clinical medical and home-care [Chi+10a], energy-efficiency is one of the most important parameters, since nodes have to save energy to meet the lifetime requirements of typical applications. To the best of our knowledge, no deployment has taken place without considering the energy efficiency. Furthermore, WSNs have come to maturity, thus offering more complex applications and traffic patterns. In event-driven traffic models, nodes only transmit their measurements upon detection of an event determined by the application, otherwise when the channel is occupied or the network becomes unreachable, nodes may store their collected data and transmit them later at high-rate for short period of time (i.e., burst). This situation implies increased channel occupancy for limited periods of time, at the cost of deteriorating the average network performance.

The MAC layer is an essential component of the sensor network protocol stack. It manages the communications between nodes. In WSN, it is in charge for turning the radio device ON and OFF periodically. This duty-cycling functionality results in a fundamental trade-off between energy consumption and network performance (i.e., latency). Numerous MAC protocols have been devised [Bac+10], but to the best of our knowledge very few of them address the needs implied by the presence of variable traffic in the network (e.g., MaxMAC, AADCC). As a result, no successful WSN deployment with auto-adaptation to the traffic has been experienced.

In this study, we consider contention-based MAC protocols [Can+11] mainly because they are scalable (e.g., local decisions) and robust to network changes (e.g., mobility). In these protocols, nodes in the network asynchronously check the radio channel for incoming packets at regular intervals. In between, they turn OFF their radio to save energy. The period between two channel samplings is referred to as the Sleeping Time (ST). Then, a node expecting to transmit will have to send beforehand a preamble longer than the ST, to ensure that its target is awakened. While a longer ST allows nodes to save energy, it reduces the amount of traffic they can handle. Conversely, a shorter ST allows nodes to handle more traffic, at the cost of increased energy consumption.

Since two nodes with different MAC parameters may be unable to communicate, most of the WSN deployments consider homogeneous (i.e., identical for all nodes) and static (i.e., invariant over the time) configurations [Sua+08], [Bar+08a]. But avoiding any partition of the network is a situation far from ideal, since performance and energy gain could be increased by providing a dedicated configuration to each node, according to its current traffic load.

In this Chapter, we introduce T-AAD, a Traffic Auto-ADaptive mechanism [Pap+14b]. T-AAD anticipates traffic load variations by tuning the MAC parameters according to the upcoming traffic volume of the concerned nodes. Using T-AAD, a node decides autonomously from its own parameter setting, thereby aiming at both increasing the network lifetime (i.e., reduce to minimum the energy consumption) and decreasing the latency. To evaluated our solution, we perform an exhausted comparative performance evaluation, both through COOJA simulator [Ost+06], and experimental studies with TelosB motes and over FIT IoT-LAB [Pap+13] testbed.

4.2 T-AAD Design

In this Section, we detail the functionality of our scheme. T-AAD is a lightweight traffic auto-adaptive algorithm for WSNs facing variable and dynamic traffic loads. The idea behind T-AAD is to automatically adapt MAC parameters, in reaction to traffic load variations. In particular, this mechanism tunes the MAC parameters based on the amount of packets that a node expects to transmit.

We propose a receiving node to switch from long to short ST mode for the period of traffic variation, and then back to the long mode. Thus, the node adopts a short ST configuration for a certain time in order to quickly handle a high traffic load, and then it switches back to long ST during periods of sporadic traffic, thus maintaining a low-power MAC configuration. Figure 4.1 illustrates X-MAC protocol, without and with the T-AAD mechanism. Two nodes (A and B)

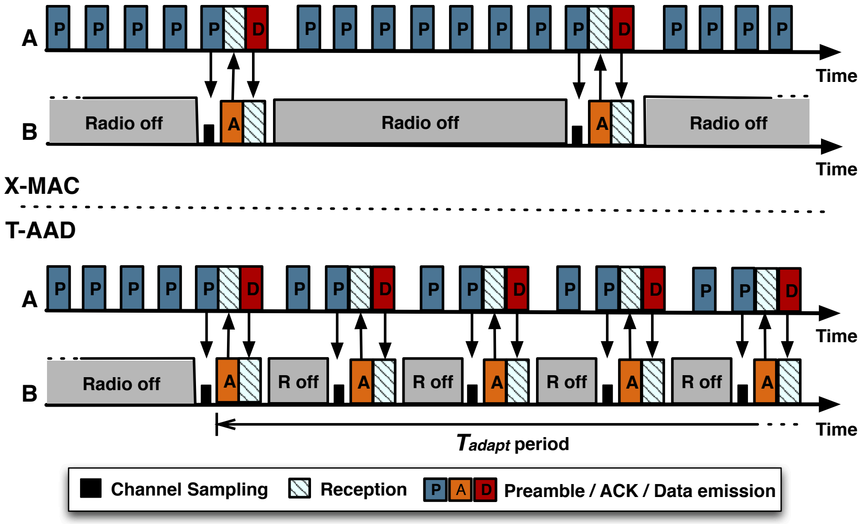


Figure 4.1: Improvement introduced by the T-AAD mechanism over X-MAC protocol.

are to handle a bursty traffic. Node A sends to B and should be allowed to reduce the preamble period. Node B is receiving from A and should sample the medium more often in order to cope with incoming bursty traffic. This way, both the A and B nodes will be able to handle the traffic variation quickly (i.e., delay decrease and throughput increase) and in an energy efficient fashion (i.e., extending the battery lifetime).

In this section, we detail how A and B can agree on both preamble and channel sampling periods, thus experiencing lower energy consumption (especially for A since it has a shorter preamble duration) and increasing reactivity thanks to shorter delays (especially for B that handles more received packets in the same time frame). We need to emphasize the fact that no centralized computation (i.e., no communication overhead for control traffic) is required while no connectivity loss can be experienced.

Our proposal is based on a mechanism where a receiver node smoothly and dynamically adjusts to the traffic demands for the time that the traffic load variation takes place. In order to achieve it, one piece of information is included in each packet. This allows the indication of the number of packets (i.e., through MAC or application queue) that a node expects to send in a row (e.g., queue storage). This value is used to calculate the time (here referred as T_{adapt}) which a node stays in the short ST mode.

All nodes start with identical long ST configurations, referred to as ST_{max} (e.g., 500 ms, 1 s), depending on the initial user deployment configuration to face sporadic traffic. Once a receiving node detects a traffic load variation, it will calculate the T_{adapt} according to the information at the first received packet.

Then, it will switch its ST mode to minimum, depending on the hardware, referred as ST_{min} (e.g., 32 ms, 64 ms), and will return to ST_{max} as soon as the traffic burst is expected to end, in order to cope with the nature of the transmission. Hence, a node with the T-AAD mechanism running on top of any static MAC protocol gains energy when using a long ST mode, and performs better in delay when switching to short ST. Thus, it is expected to profit from both situations. The detailed function of T-AAD is presented in Algorithm 1. Note that one node implementing this Algorithm regularly updates a timer according to T_{adapt} period. Upon the expiration of this timer, the node switches back to a long ST mode.

Algorithm 1: Functionality of the T-AAD mechanism

```
begin
    when reception of packet P then
        if $ST_{max}$ then
            if $P.Q_{len} > 1$ then
                calculate $T_{adapt}$;
                $set_{ST}(ST_{min})$ for $T_{adapt}$;
            else
                continue with $(ST_{max})$;
            end
        else
            if $P.Q_{len} > 1$ then
                calculate new $T_{adapt}$;
                if new $T_{adapt}$ > current $T_{adapt}$ then
                    $set_{ST}(ST_{min})$ for new $T_{adapt}$;
                else
                    continue with current $T_{adapt}$;
                end
            else
                continue with current $T_{adapt}$;
            end
        end
    end
end
```

4.3 High Traffic Load Period Estimation

T_{adapt} designates the actual time that a node will stay in the short ST mode. Let ST_{max} be the time period for long ST and ST_{min} for short ST respectively. Q_{len} indicates the total number of packets that a transmitter is expected to send, and finally let M_{err} be the margin of error. As we placed ourselves in a realistic context, we considered losses that inevitably occur in wireless technologies, due to either collisions or interferences in the network. Those losses induce packet retransmissions on a non-predictable basis. We consider M_{err} to cope with the potential packet retransmissions. Moroever, in order to ensure that the calculated period T_{adapt} is at least as long as the whole burst period (including those MAC retransmissions), a margin of error is added to reflect the probability of having packet retransmissions. This margin is depending on deployment conditions (e.g., loss-rate, perturbations), and could be updated dynamically during the deployment, based on past observations (T_{adapt} ending while more traffic remains for instance), thus engineers may adjustM_{err} by recording wrong/good calculations of T_{adapt} that have been made. Hence, this observation can be modeled and computed as follows:

$$T_{adapt} = ST_{max} + (Q_{len} - 2) \times ST_{min} \times (1 + M_{err}) \tag{4.1}$$

Let us assume that a node A has n packets to be transmitted towards the node B. Thus, the queue length (i.e., Q_{len}) is equal to n. There are $n-1$ sleeping periods among the n consecutive packets. The receiver is in the ST_{max} mode during the first period, and the rest $n-2$ switches in short ST (i.e., $n - 2 \times ST_{min}$), thus inducing highly reactive changes in the MAC configuration. As a result, T-AAD is able to handle varying traffic efficiently. Furthermore, in order to avoid any miscalculation at evaluation performance, we add a M_{err} based on the queue length and the traffic congestion in the network. Finally, since each packet contains Q_{len}, node B first will calculate the T_{adapt} before switching to ST_{min} for T_{adapt} period.

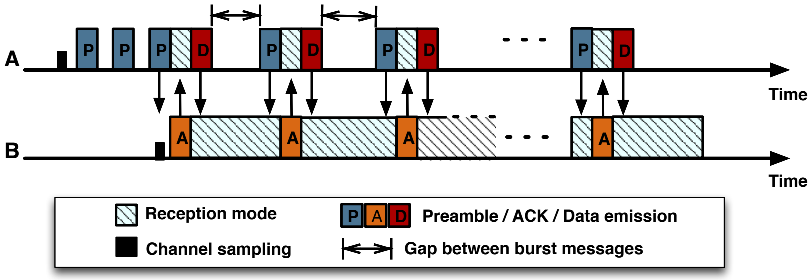


Figure 4.2: CSMA-based solutions (e.g., MaxMAC [HB10]).

4.4 Advantage of T-AAD over CSMA-based Solutions

In existing protocols such as BoXMACs [ML08], each node can be configured in order to keep its radio ON for a while, right after receiving each packet. As a result, they turn into the CSMA mode during traffic load increase. In MaxMAC [HB10], once the incoming packet rate reaches the last threshold, the nodes switch in a CSMA fashion as well (see Figure 4.2, the CSMA version of MaxMAC).

It is worth pointing out the reason why we aim to reduce the channel sampling frequency to minimum rather than eliminating it completely and switching into the basic CSMA mode and then recovering to the preamble-sampling mode again. In particular, we observe in both simulation and experimentation campaigns that on the transmitter side there is a time interval between two consecutive packet transmissions because of the necessity to have a Clear Channel Assessment (CCA) before each message transmission). Note that, one of the most important parameters of the radio component is the CCA threshold, which indicates if the radio channel is free or busy. In fact, while the CCA is enabled, packets will be transmitted only if the channel is free. Hence, keeping the radio active at the receiver side during this period has a significant impact on the energy performances. By taking into consideration the fact that most of the power consumption is spent during the reception or transmission mode, and even more after our mathematical analysis, we notice that switching the radio between each mode (i.e., ON and OFF) provides more energy gain than keeping it turned ON constantly. Below is the formula which provides us with the energy gain that we obtain compared with CSMA-based solutions.

$$E_{save} = (P_{rx} - P_{off}) \times (X - 1) \times (T_{int} - T_{Soff} - T_{Son}) \tag{4.2}$$

This equation is always true if and only if the following three concepts are fulfilled:

- The consumed reception power (i.e., P_{rx}) must be greater than the power consumed during the idle mode (i.e., P_{off}). According to most current hardware characteristics, this statement holds (see Table 4.1).
- X, represents the number of intervals between n consecutive packets. During high traffic loads, there are multiple packet transmissions, thus, X is greater than 1.
- Finally, the time between two consecutive packets (i.e., T_{int}) must be greater than the sum of times the radio switches ON and OFF (i.e., T_{Soff} and T_{Son}). This value can be retrieved from the radio component datasheet or determined after initial simulation and experimental campaigns. Indeed, during our evaluation, we experienced intervals ranging from 20 to 26 ms.

	Power RX	Power TX	Power Off	Startup time
CC1101	$19.7mA$	$17.4mA$	$426\mu A$	$240\mu s$
CC2420	$14.7mA$	$17mA$	$200nA$	$192\mu s$

Table 4.1: Characteristics of CC1101 and CC2420 (datasheets) radio chipsets

4.5 Performance Evaluation of T-AAD mechanism

The mechanism presented in this study can be applied to various preamble-sampling protocols (e.g., X-MAC, SpeckMAC, ContikiMAC). To evaluate our mechanism we picked the X-MAC [Bue+06] protocol since many real WSN and IoT deployments and recent scientific contributions rely on this protocol [Zim+12], [YH12]. Moreover, it was already pointed out that X-MAC is a serious alternative to the MAC layer defined in the 802.15.4 standard [Sua+08]. Hence, we believe that X-MAC is a generic protocol above which we show that our proposal can work. X-MAC optimizes the LPL mechanism by allowing intended receivers to reply to strobed preamble with an ACK, which triggers a data transmission. Thus, it stays *ON* until the data transmission is complete. This scheme mitigates both overhearing on non-recipient nodes and long preamble transmissions on senders.

T-AAD aims at managing dynamic traffic at best, by adapting MAC parameters locally and at runtime. In the previous section, we exposed the design of this mechanism, and detailed to what extent it may reduce energy consumption and delay when compared to major contributions from the literature. In this section, we present a thorough performance evaluation, both through the COOJA simulator [Ost+06] with Sky motes and experimental study over FIT IoT-LAB, a very large scale WSN testbed [Pap+13]. In addition, we implemented and compared our mechanism both with a statically configured network using X-MAC [Bue+06] and a state-of-the-art auto-adaptive mechanism such as AADCC [MH10b] on top of the Contiki OS [Dun+04], in order to have a fair and thorough comparative study.

4.5.1 Simulation Evaluation

Simulation Setup

In this study, we assigned 500 ms for ST_{max} and 32 ms for ST_{min} respectively as ST configurations for T-AAD and AADCC both for simulation and experimentation. On the other hand, for pre-configured X-MAC, we kept the standard format with ST of 32, 125, 250 and 500 ms (i.e., X-32, X-125, X-250 and X-500 respectively). The maximum number of retransmissions was 3 for all protocols. At the routing layer, we rely on a broadly used scalable gradient protocol which generates a routing tree rooted at the sink (i.e., using a number of hops as a metric). Note that the induced control message overhead is limited since a single message originates from the sink (including a rank equal to 0) [Wat+09]. All receiving nodes (i.e., in the communication area of the sink station) update their rank before forwarding the message. The process is kept running while new control messages are initiated only upon reception of a better rank information. This routing protocol was chosen for its performances under realistic conditions [Bar+08a].

Our simulation scenario involves 50 wireless sensors that are either randomly or uniformly distributed (i.e., grid) in an area of 50 × 50 meters. An example of the grid topology used in our simulations is depicted on Figure 4.3. The nodes use a time-driven application in which 10 packets are sent in a row, every 500 seconds, with a random initial backoff to avoid synchronous traffic bursts. We chose a 10 byte data size for both the simulation and the experiment, which corresponds to the general information used by monitoring applications (e.g., node ID, packet sequence, sensed value). For the sake of clarity and easier initial analysis, before running our tests over our real sensor network testbed, we used a radio model based on disks where each wireless sensor has a -10 dBm power transmission, thus imposing multi-hop communications to reach the sink station (i.e., up to four hops). The simulation lasts for 35 minutes.

	COOJA Simulator	**IoT-LAB testbed**
Platform parameters	**Value**	**Value**
Topology	Grid / Random ($50m \times 50m$)	Regular grid ($10m \times 8m \times 3m$)
Number of nodes	50 fixed sensors	80
Number of sources	49	79
Node spacing	7.14 *meters*	1 *meter*
Sim & Exp parameters	**Value**	**Value**
Duration	35 *minutes* / 2100 *seconds*	70 *minutes* / 4200 *seconds*
Application model	Burst 10 *pkts*/500 *s*	Burst 10 *pkts*/1000 *s*
Number of events	2000	2400
Payload size	10 *Bytes* (+6 *Bytes* MAC header)	
Routing model	Gradient [Wat+09]	
MAC model	T-AAD: ST_{min}=32 *ms*, ST_{max}=500 *ms*	
	AADCC [MH10b]: ST_{min}=32 *ms*, ST_{max}=500 *ms*	
	X-MAC [Bue+06]: ST = 32, 125, 250 or 500 *ms*	
Maximum retries	3	
Margin of error (M_{err})	15% (see Equation 4.1)	
Hardware parameters	**Value**	**Value**
Antenna model	Omnidirectional CC2420	Omnidirectional CC1101
Radio propagation	2.4 *GHz*	868 *MHz*
Modulation model	O-QPSK	GFSK
Transmission power	-10 *dBm*	-20 *dBm*

Table 4.2: Simulation & experimental setup

The results hereinafter present the gain of our proposal in terms of energy consumption and delay, compared to a state-of-the-art auto-adaptive protocol AADCC and to a homogeneously pre-configured X-MAC. More importantly they show that heterogeneous configurations can be determined independently at each node in a localized fashion, without risking any partitioning of the network. The details of the simulation setup are exposed below in Table 4.2.

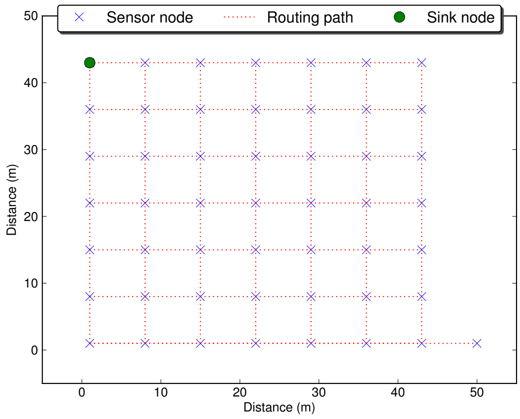
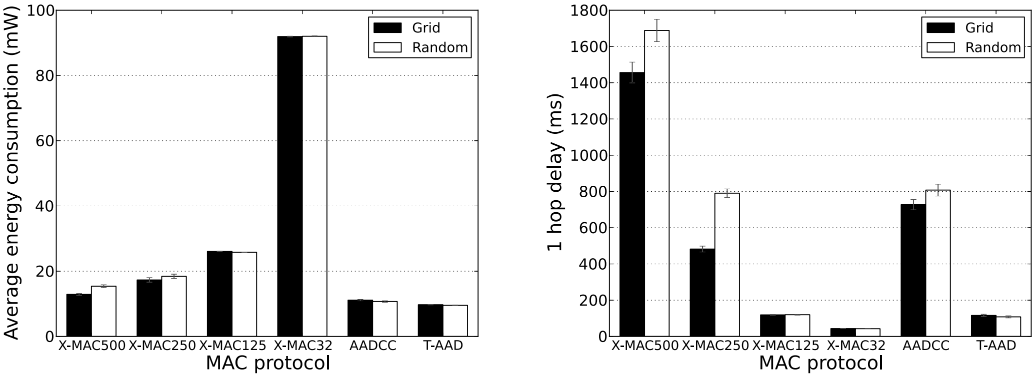


Figure 4.3: The topology used in our simulations.



(a) Average energy consumption

(b) Average one-hop delay

Figure 4.4: Performances of MAC protocols in a simulation-based network.

Simulation Results

a) Energy consumption: In Figure 4.4a, the average energy consumption per second for the whole network is presented for both grid and random topologies. T-AAD and AADCC consume less energy network-wide than any of the pre-configured X-MAC. In particular T-AAD reduces the energy depletion by about 37% when compared to X-125. The results show that the preamble length has a straightforward impact on energy dissemination. In case of low traffic, the total energy consumption is higher for short preamble lengths due to more periodic channel samplings. Conversely, in the case of heavy traffic loads, long preambles consume more due to the constant presence of transmission and reception of the preambles. As an example, the energy dissemination is higher for nodes that forward more (i.e., nodes located around the sink). Globally during the conducted experiments we observed that T-AAD performs slightly better than AADCC. This is mainly due to the long preambles that AADCC necessarily induces for each adaptation process.

b) Delay performance: Figure 4.4b depicts the average one-hop delay per packet for all nodes. The one-hop delay includes the initial back-off, the channel sampling period, potential congestion back-offs, potential retransmission delay and the preamble length. With an average one-hop delay of 108.08 ms, T-AAD displays well performance. Even though T-AAD and AADCC have similar energy consumption, in terms of delay T-AAD performs almost five times better than AADCC. Thus, according to these results, T-AAD anticipates the traffic load variations better. Overall, all LPL configurations perform identically or worse in random topologies. This is mainly due to the potential bottleneck links that are more prone to appear in random topologies, having as a result nodes to handle heavy traffic. X-500 has the worst performance (i.e., 1688.6 ms) due to the very high channel occupancy (i.e., high congestion) the radio medium is almost never available to transmit a packet. This reveals that there is a high probability of provoking congestion back-offs in the network, especially for the nodes that are located one-hop away from the sink, having as a result the increase of the one-hop delay around the sink. Furthermore, these higher delays are also due to MAC retransmissions. Indeed, due to radio channel competition and hidden terminals, larger proportions of sent packets require several retransmissions before being acknowledged.

4.5.2 Experimental Evaluation in Non Contention Environment

Setting up a complete WSN deployment is a very complex task [Bar+08a]. To further evaluate the performance of the T-AAD mechanism, we performed a number of experiments.

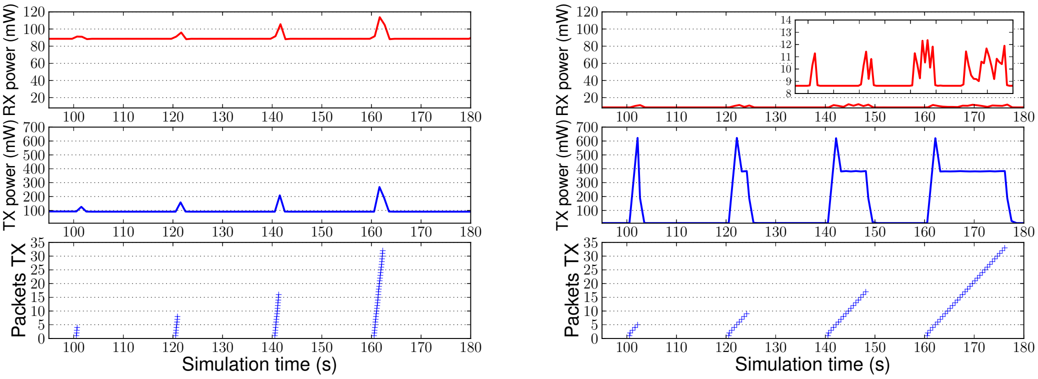
Experimental Setup

In this set of experiments, we have used two TelosB [Pol+05] motes (i.e., transmitter and receiver respectively), located one next to the other. We implemented our solutions on top of Contiki OS [Dun+04], to reproduce the non-contention scenario.

In this investigation, we evaluated our proposal within idealistic scenario, where a single sender periodically transmits in bursty mode from 4, 8, 16 to 32 packets to a single receiver in order to avoid contention and competition for the radio medium. This way, we aim at obtaining accurate results both for senders and receivers behavior. We run number of experiments with TelosB motes for pre-configured X-MAC both in 32 *ms* and 500 *ms* ST values and we compared them with T-AAD scheme.

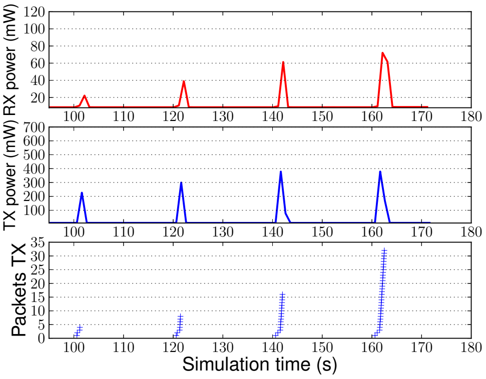
Behavior Analysis in Non Contention

An overview of the proceedings of this first experimental campaign is detailed in Figure 4.5. The first two subplots of each figure represent the energy consumption for receiver and for transmitter node respectively while bottom subplot is for the delay. According to the results, transmitter node configured at 500 *ms* consumes more energy, due to the long preambles, compared to other solutions. On the opposite, due to the frequent channel sampling in X-32 configuration



(a) X-MAC protocol with sampling frequency configured at 32 *ms*.

(b) X-MAC protocol with sampling frequency configured at 500 *ms*.



(c) T-AAD proposal.

Figure 4.5: The energy consumption and delay performance results for two TelosB motes testbed for T-AAD scheme and for X-MAC protocol with 32 *ms* and 500 *ms* sampling frequency respectively.

the receiver node consumes much more energy comparing with other setups while it performs better in terms of delay. As can be observed, there is a antagonism between the different X-MAC configurations, in a homogeneous setup. Indeed, a low LPL value allows the nodes to handle efficiently and rapidly a more intense traffic (as burst in this situation). This allows for a reduction of delay performance and channel occupancy (due to a faster acknowledgement of message preambles). Moreover, it induces a higher energy consumption in the absence of traffic. On the contrary, a higher LPL value allows for a reduced energy consumption (at the receiver side) under low or non-existent traffic load, but makes the communication under intense packet rates more difficult, increasing messages delays, energy consumption (at the transmitter side) an radio channel occupancy. Thus, every MAC configuration is adapted to a specific situation and traffic load. In consequence, keeping it homogeneous and static throughout the network prevents it from addressing the traffic load variations inherent to such dynamic networks.

Conversely, T-AAD manages every situation by providing an heterogeneous and dynamic MAC configuration to the nodes [Pap+15a]. In particular, both the transmitter and receiver nodes consume less or equal than any X-MAC setup. There are short peaks every constant period due to the very short LPL values during the burst periods (i.e., for data packet transmissions and receptions). Indeed, the high LPL value in the absence of traffic loads helps the node to reduce its average energy consumption in this situation, to the same level as X-500. Likewise, the rapid LPL reduction in case of high traffic load variations, allows T-AAD to achieve burst delay performance similar to X-32. It also induces a reduction of the average channel occupancy and energy consumption by at least half. This way, instead on focusing on a specific traffic load, T-AAD allows the nodes in the network to have it both ways at no cost. The performance of T-AAD it is well depicted in Figure 4.5c.

4.5.3 Experimental Evaluation in Contention Environment

Experimental Setup

We continued our thorough empirical analysis within a contention-based scenario, ensuing X-MAC, AADCC and T-AAD protocols, conducting them over the ICube's platform of the open large scale FIT IoT-LAB [Pap+13] testbed.

To perform our experimental evaluation, we chose to work with 80 nodes of the same layer, where 79 nodes are randomly selected as data sources and a single sink was located at the top left of the platform. The nodes implement a time-driven application model, and each of them transmits 10 packets in a row every 1000 seconds with an emission power at -20 dBm, thus imposing multi-hop transmissions to reach the sink station (i.e., up to four hops), see Figure 4.6). As per Equation 4.1, the margin of error was fixed at a realistic value of 15% of the total T_{adapt}. Finally, the experiments last for 70 minutes. The details of the experimental setup are exposed below in Table 4.2.

Experimental Results in High Contention

At first, we evaluated the general performances of the network, either configured homogeneously using X-MAC or implementing auto-adaptation mechanisms (AADCC and T-AAD). Figure 4.7a and 4.7b illustrate the network performances in terms of delay (i.e., average time for ten packets transmissions) and energy consumption, for each case of study. Both results outline the same aspects as for simulation. Indeed, we can observe that the delay increases respectively with ST, while energy consumption decreases, thus demonstrating that there is a strong antagonism between energy consumption and delays when considering homogeneously configured networks. Conversely, T-AAD allows the network to gain both in delay and energy consumption at the same time. In fact, T-AAD allows for delays similar to X-125 with an energy consumption lower than that obtained with AADCC or X-500. This phenomenon is due to both a better handling of messages during periods of high traffic load, and to lower sampling frequencies during calm periods in the network. Moreover, it is achieved in a localized fashion, without endangering network connectivity.

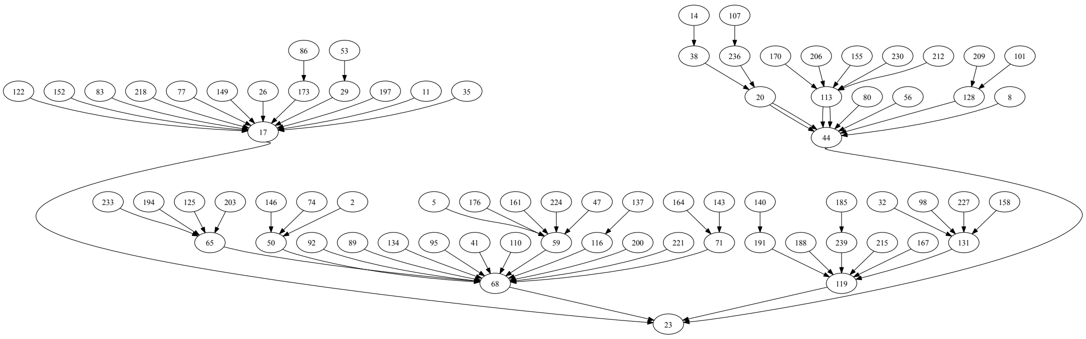
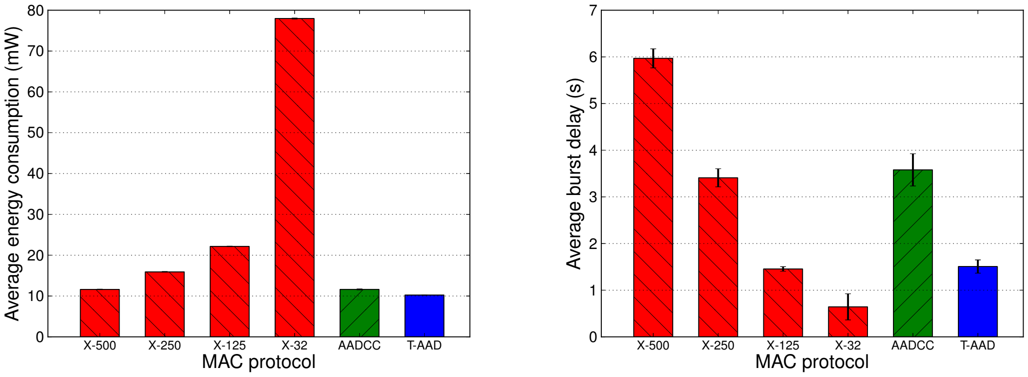


Figure 4.6: Routing topology of the testbed



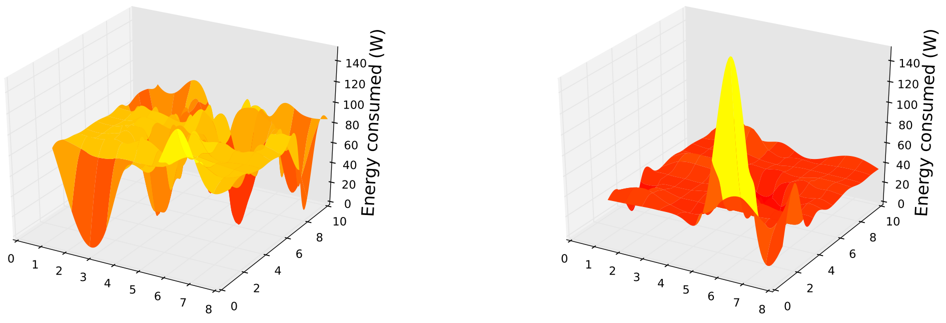
(a) Average energy consumption

(b) Average burst (transmission of 10 $pkts$) delay

Figure 4.7: Performances of MAC protocols under a complete and real network.

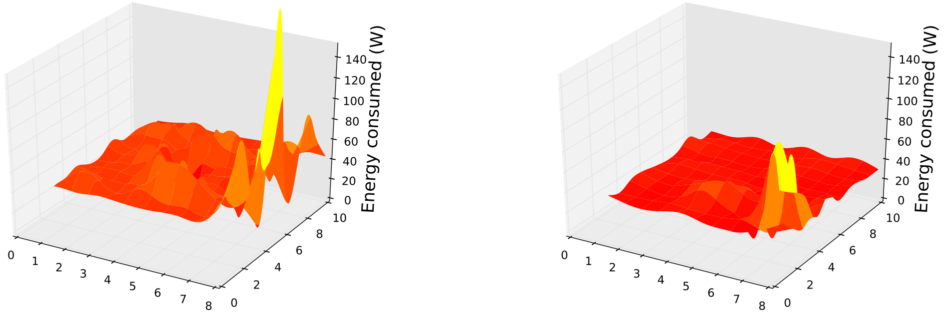
We also performed a mapping of the energy depletion throughout the network, in order to identify its repartition and disparity among the nodes. To do so, we measured the total energy consumption of each node for the whole duration of the experiments. It can be noted that it does not impact the protocol as energy monitoring is a service provided by the testbed through independent monitoring channels. The results of this evaluation are displayed in Figure 4.8 for T-AAD, AADCC and X-125. As can be observed, the energy consumptions of X-MAC and AADCC vary depending on the considered node. Some nodes, located at the extremities of the routing topologies consume only a little energy (about 8 W), while most nodes have to manage the traffic load and thus consume a greater amount of energy during the experimentation (130 W in average). On the contrary, with T-AAD, the energy consumption remains similar throughout the network (with the exception of the sink). Indeed, the nodes consume about $25 - 30$ W, independent of the position of the node in the routing topology (the sink, noticeable by its energy peak not being considered in our study).

The previously presented phenomenon is due to the fact that the traffic load bears a strong influence on energy consumption (long preamble transmissions). As a consequence, nodes which have to relay information from their neighbors will have to remain active for a larger proportion of time than the nodes that are activated, only to send their own data. T-AAD, allows for a quicker management of packets, and thus nodes can go back to sleep after a short period only, which explains the homogeneity of its energy depletion. Hence, T-AAD appears to be more stable and less dependent on the traffic load, as well as providing significant energy consumption reduction when compared to any network homogeneously configured with X-MAC or with an auto-adaptive AADCC mechanism.



(a) X-125

(b) X-500



(c) AADCC

(d) T-AAD

Figure 4.8: Total energy consumption for X-125, AADCC and T-AAD.

4.5.4 Summary of Evaluation

In this section, we evaluated to what extent our T-AAD mechanism can enhance the management of versatile traffic by auto-adapting dynamically and locally the MAC parameters. We demonstrated both through simulations and experimental campaigns that our solution allows for energy savings, and delay decrease as well as channel occupancy reduction, when compared to any homogeneously pre-configured X-MAC version. Moreover, T-AAD outperforms state-of-the-art auto-adaptive solutions such as AADCC, with a 12% reduction in energy wastage along with a 58% decrease in terms of delays.

4.6 Conclusions and Perspectives

In this Chapter, we intensively demonstrated the drawbacks of a network with homogenous and static MAC configurations. The results could show that addressing a scenario with traffic load variations in homogenous preamble-sampling configurations reveals a tradeoff. On the one hand when configured with ST of 500 *ms* less energy is consumed but poor network performance is induced (e.g., high latency) and vice versa with ST 125 *ms*. Hence, after a thorough study of the state of the art, we proposed a novel approach to address this problem. T-AAD is an auto-adaptive scheme that is compliant with preamble-sampling based MAC protocols. Our mechanism is based on local decisions at each node, made on the available information from data transmissions (e.g., acknowledgments, preambles), thus limiting overall communication overhead and being well suited for constrained devices and networks. T-AAD also exhibits the feasibility of being able to manage the heterogenous MAC configuration in a localized fashion, without endangering network connectivity and consequently to handle dynamic traffic (e.g., energy-efficiency, delay).

We have performed an exhaustive performance evaluation both through the COOJA simulator [Ost+06] and experimental studies with TelosB motes, and with FIT IoT-LAB [Pap+13] testbed in order to accurately analyze the performance of our proposal. We compared our mechanism both with a statically configured network using X-MAC [Bue+06] and an auto-adaptive mechanism such as AADCC [MH10b]. Our results show that T-AAD outperforms any preconfigured X-MAC setup since it automatically reaches the optimum MAC configuration for the time of the traffic load change. We also avoid transition phases, and thus, useless long preambles such as in AADCC, which allows T-AAD to outperform this solution. Therefore, T-AAD succeeds in addressing the energy/latency trade-off.

Our perspectives for future work consist of further exploring this lead, and in particular by investigating the automated learning of traffic patterns. Indeed, we first focused on auto-adaptation to address the dynamicity induced by varying traffic, and thus, we decided to investigate dynamics and instability ensued from mobility in the network and its impact on MAC protocols. We therefore maintain the burst or varying traffic assumption, while removing the hypothesis of static infrastructure in our following Chapter. Hence, our vision is to employ the T-AAD (or similar) scheme in mobile sensor nodes within a mobility-aware WSNs.

Chapter 5

Enhanced MAC under Mobility and Bursty Traffic Assumptions in WSNs

In this Chapter, we focus on designing MAC protocol under dynamic and bursty traffic for mobility-aware WSNs. During the development of MAC protocols, mobility may pose many communication challenges. These difficulties require first a link establishment between mobile and static nodes, and then an energy efficient and low delay burst handling mechanism. In this study, we investigate preamble-sampling solutions that allow asynchronous operation in the sensor network. We first introduce anycast transmission to ContikiMAC where a mobile node emits an anycast data packet whose first acknowledging node will serve as responsible to forward it towards the sink. Once this link is established, burst transmission can start, according to the respective burst handling mechanism of ContikiMAC. Although it is considered as negligible in the literature, such an anycast-based on-the-fly operation actually results in high packet duplication at the sink. Hence, we demonstrate that even a basic anycast-based M-ContikiMAC would fail to handle bursty traffic from mobile nodes mainly due to increased unnecessary traffic and channel occupancy (in dense networks). We then propose Mobility-Enhanced ContikiMAC (ME-ContikiMAC), a protocol that reduces packet duplications in the network by more than 90% comparing to M-ContikiMAC. Moreover, our results show that ME-ContikiMAC outperforms a number of state-of-the-art solutions, by terms of reducing both delay and energy consumption.

Contribution

This chapter presents the following contributions:

1. We first introduce M-ContikiMAC, an extension of the statically-oriented ContikiMAC protocol, to manage the communication in mobility-aware WSNs.
2. We then present the enhanced version of our packet duplication control mechanism in order to mitigate the duplications that M-ContikiMAC induces, in order to reduce the channel occupancy.
3. Furthermore, we discuss the remaining limitations of M-ContikiMAC and propose to overcome them by introducing some optimizations of ME-ContikiMAC. Hence, we show to what extent it allows reduced 1-hop and end-to-end delays.
4. Finally, we enhance our initial performance evaluation in order to compare ME-ContikiMAC proposal against M-ContikiMAC as well as against other state-of-the-art solutions (such as MoX-MAC [Ba+14] and MOBINET [Rot+11]).

5.1 Mobility in WSNs

As already discussed in Section 2.3.3, many existing MAC protocols deal with mobility to some extent (i.e., [DD13]). However, to the best of our knowledge very few of them address the needs implied by the presence of variable and bursty traffic in the network. As a result, no successful WSN deployment with mobility handling and dynamic traffic has been proposed so far.

In this Chapter, we introduce the basic M-ContikiMAC protocol [Pap+14a], which is compliant with any preamble-sampling MAC protocol. We illustrate this by embedding our proposed solutions as anycast-based extensions of ContikiMAC [Dun11], which presents some anomalies in the network due to its anycast-based transmissions. Furthermore, we discuss how to allow mobile nodes to co-exist and communicate with static nodes in the network. We also further detail our Mobility-Enhanced ContikiMAC (ME-ContikiMAC) protocol [Pap+15d] that handles mobility even under very dense WSN scenarios. This allows us to evaluate ME-ContikiMAC in a large-scale environment where we verify ME-ContikiMAC's efficiency in dense network environments. In addition to the original description of M-ContikiMAC and ME-ContikiMAC, we present MobiXplore [Pap+15b], a MAC scheme that allows a seamless transfer of communication (handover) to achieve low reconnection and handover delays.

5.2 Background & Overview

In this Section, we first provide the necessary background on ContikiMAC protocol and its drawbacks. We then perform a high-level description of our proposed approaches, M-ContikiMAC, ME-ContikiMAC schemes.

5.2.1 ContikiMAC protocol

We have chosen to develop our schemes over the ContikiMAC protocol due to its popularity in real WSN and IoT deployments as well as in recent scientific contributions [Duq+13], [Vuc+13]. However, our proposals are general enough to be used in any preamble-sampling oriented protocols. ContikiMAC, the default MAC layer protocol in Contiki OS, embeds most of innovative features of existing preamble-sampling protocols. In particular, periodic wake-ups have been suggested by X-MAC, the phase-lock optimization has been presented by WiseMAC [EH+03] and the use of data packet copies as a wake-up strobe has been previously introduced by the BoX-MAX [ML08], the default low-power MAC protocol in TinyOS. Furthermore, ContikiMAC comes with a bursty transmission handling mechanism [Duq+11]. In [MQ14], after a thorough performance evaluation, the results illustrate that ContikiMAC achieves a better delay performance and significantly lower energy consumption compared to X-MAC. Therefore, we consider ContikiMAC as the leading protocol in the preamble-sampling family of MAC protocols above which we demonstrate that our proposal can operate efficiently.

ContikiMAC originally provides two types of transmissions, the so-called unicast and broadcast. Under unicast mode, the sender repeatedly transmits its data packet that contains the payload and the destination address until it receives a link layer acknowledgment from the receiver. On the other side, the intended receiver periodically wakes-up to sample the medium for packet transmissions from its neighbors. Once a transmission is detected during a wake-up, the receiver keeps the radio *ON* to receive the packet that will follow. When a data packet transmission is successful, the receiver replies with an ACK packet. Under broadcast mode, the potential receivers do not acknowledge the received data packet. The sender actually repeatedly transmits the data packet during the entire preamble period to ensure that all its neighbors have received it. The concept of unicast and broadcast transmissions according to the ContikiMAC is illustrated in Figure 5.1.

Furthermore, ContikiMAC provides a burst handling mechanism to anticipate high traffic periods in the network [Duq+11]. In particular, under burst mode a transmitter expects to transmit multiple packets in a row. To do so, the sender modifies the header for each data packet of the queue (except the last). In fact, it sets a flag that notifies the receiver that another packet

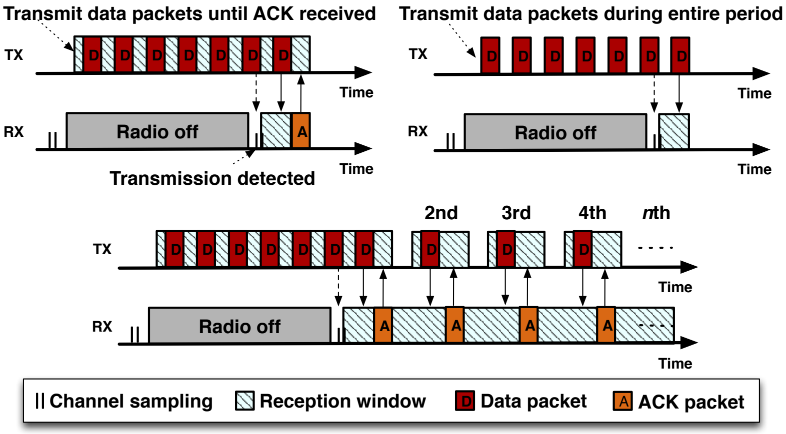


Figure 5.1: Transmission modes of ContikiMAC protocol, namely Unicast, Broadcast and Burst respectively.

follows. On the other side, the node receiving the flagged packet, appropriately adapts its radio duty cycle. Indeed, the receiver keeps its radio *ON* for a period of inter packet deadline to receive the following packets. As a result, the receiver node switches into Carrier Sense Multiple Access (CSMA) mode until it receives a data packet (conventionally the last packet of the queue) that is not labeled with the burst flag notification. Finally, the transmitter first waits for the acknowledgement of the ongoing transmission before transmitting the next data packet.

5.2.2 Challenge

Even though the ContikiMAC protocol is well designed for static networks, it does not perform effectively under environments where static and mobile nodes co-exist. In fact, since the mobile nodes do not participate in the construction of the routing tree, they are not aware of the next-hop address and this actually prevents them from utilizing unicast transmissions. Considering the default unicast and broadcast functions of ContikiMAC, a mobile node should transmit its packets by employing broadcast. Since broadcasting is a costly alternative, mobile nodes fail to access the medium to communicate with static nodes in an efficient manner (resulting in low energy consumption and delay performance). In the case of a burst transmission, a mobile node may either transmit all n packets in broadcast, or it transmits the first data packet in broadcast to discover a temporary parent and then switches to unicast mode to dequeue its buffer. As a result, the default transmission modes of ContikiMAC induce certain network deficiencies when it comes to mobile nodes.

5.2.3 ME-ContikiMAC in a Nutshell

Since the mobile nodes do not utilize any routing scheme, there is a need for an efficient parent discovery mechanism. We here, introduce the Mobile-ContikiMAC mechanism (M-ContikiMAC), an extension of the ContikiMAC protocol, and its enhanced version ME-ContikiMAC. In this study, we depart from the unicast design paradigm. Instead, we propose an additional transmission mode, which allows to any given node that is located in the transmission range of a mobile transmitter to be its receiver, (acting as a temporary parent). To implement this approach, we introduce an anycast transmission, where a packet is transmitted opportunistically to the first potential forwarder acknowledging the corresponding packet. Hence, a mobile node chooses the next-hop, static node that wakes-up the soonest. Note that in anycast mode, the potential receivers are all identified by the same destination address.

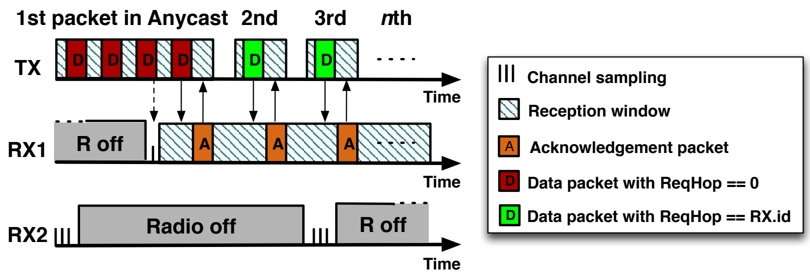


Figure 5.2: M-ContikiMAC in basic mode where the transmitter transmits first in anycast following by unicast.

5.3 Design of M-ContikiMAC & ME-ContikiMAC

This Section presents the core mechanisms of our proposed protocols. More specifically, we provide a detailed description about how M-ContikiMAC interacts with the underlying base MAC protocol, and the way that the mobile nodes co-exist with the static nodes in the network by utilizing M-ContikiMAC. We then propose two enhancements over the basic M-ContikiMAC: packet duplication control and delay optimization, and we further explain how these extensions improve network performance by also combining the minimization of energy consumption.

5.3.1 M-ContikiMAC Protocol

In this study, we consider scenarios where the mobile nodes tend to transmit packet bursts (the burst notification flag is activated) under dense network scenarios. Let us assume a mobile node (TX), that expects to send n packets in a burst. As previously stated, TX is not aware about the next-hop as well as about the surrounding nodes within its transmission range. Thus, TX sets an anycast destination address for its first data packet of the queue that allows its temporary neighbors to receive it. Furthermore, on the first data packet we set an additional one byte of information (besides the burst flag), called ReqHop (Request for a next-Hop) that gets the value zero when the transmitter searches for a next-hop. Hence, TX repeatedly transmits its first data packet in anycast until it receives a link layer acknowledgment from a potential forwarder (similar to the original version of the ContikiMAC) which in turn will be its new next-hop. Later, TX sends the remaining $n-1$ packets on its queue in unicast to its temporary parent, while ReqHop is switched to $RX.id$. Note that according to the current M-ContikiMAC parent discovery configuration, only static nodes are allowed to respond to anycast transmissions from a mobile node. The detailed process of parent discovery is illustrated in Figure 5.2.

On the other side, a static node (for instance RX1) that wakes-up to sample the medium for an upcoming packet performs the following procedure: *i)* it checks if the destination address of the data packet contains its own address *ii)* if not, it then checks whether is a broadcast address *iii)* finally, if neither of these two transmission types correspond to the destination address it then checks if it is an anycast address. If so, RX1 checks the value of ReqHop whether it is *zero* or equal to its unique identification ($RX1.id$) that practically allows RX1 to accept the packet (otherwise it rejects it). Once RX1 receives the first data packet, it responds with an ACK including its own $RX1.id$, while keeping the radio ON to receive the remaining packets, as originally ContikiMAC was designed. Later, once TX receives the ACK, it sets ReqHop to $RX1.id$ and transmits the rest of the packets to its new temporary parent. In case, another static node (e.g. RX2) wakes-up during the burst transmission, it will realize that the packets are not intended to itself. Since the destination address of the packet is neither anycast nor its own, and moreover ReqHop is equal to $RX1.id$, thus, RX2 will turn its radio OFF after the sampling procedure. The detailed function of M-ContikiMAC is presented in Algorithm 2.

Algorithm 2: Functionality of M-ContikiMAC

```
begin
    when Receiver (R) receives a packet (P) then
        if P.address is R.address (Unicast) then
            Accept the packet;
            Sent ACK;
            if Burst flag ON then
                Keep radio ON for the next packet;
            else
                Turn radio OFF;
            end
        else if P.address is Broadcast then
            Accept the packet;
            Turn radio OFF;
        else if P.address is Anycast && ReqHop == $\emptyset$ then
            Accept the packet;
            Sent ACK including R.id;
            if Burst flag ON then
                Keep radio ON for the next packet;
            else
                Turn radio OFF;
            end
        else if P.address is Anycast && ReqHop == R.id then
            Accept the packet;
            Sent ACK;
            if Burst flag ON then
                Keep radio ON for the next packet;
            else
                Turn radio OFF;
            end
        else
            Reject the packet;
        end
    end
end
```

5.3.2 Reconnection mechanism of M-ContikiMAC

When mobile nodes are present in WSNs, link disconnection between a mobile and a static node is a very frequent phenomenon, mainly due to bad link quality or mobility (mobile node moves away from the range of its temporary parent). During the burst transmission period, if a mobile node does not receive the expected ACK for its ongoing transmission, it assumes that it is disconnected from its temporary parent. In order to anticipate this situation, it enables a reconnection mechanism to discover a new forwarder. Indeed, it sets the ReqHop flag back to *zero* and retransmits the same data packet in anycast mode to find a new parent to continue transmitting the remaining packets of the queue as shown in Figure 5.3.

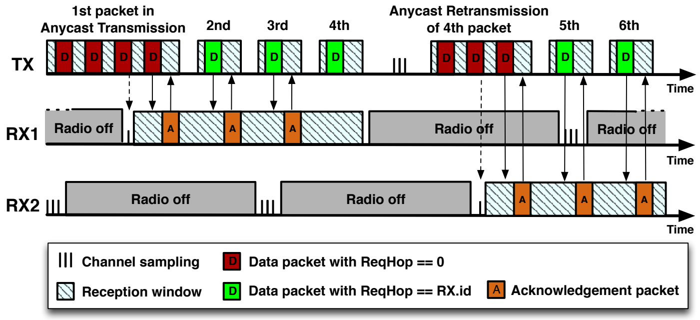


Figure 5.3: Connection recovering with M-ContikiMAC.

5.3.3 Drawbacks of M-ContikiMAC

The previously presented mechanism induces certain anomalies due to the nature of anycast transmission mode. In fact, if two or more nodes simultaneously sample the medium, they may receive the same transmitted packet, while their ACKs may collide. Indeed, the probability of two or more nodes simultaneously sampling the medium for incoming packet is strongly correlated to the total number of the forwarders. Thus, the probability of having duplicated packets at the sink node increases in dense networks, which in turn induces unnecessary traffic in the network (i.e., including both originally transmitted and forwarded packets). Hence, the traffic in the network, as well as the congestion, the channel occupancy and the competition for medium access increase, which in turn enlarges the probability of packet retransmissions due to potential collisions. As a result, network performance significantly degrades while energy consumption for the whole network attains higher values as shown in [Pap+14c].

Let us assume that TX transmits repeatedly the first of the total n packets of its queue by employing the anycast mode with ReqHop equal to zero. Thus, any static node in transmission range of TX is privileged to receive it. Hence, two (e.g., RX1 and RX2) or more static nodes having the same or almost the same sampling frequency phase may wake up and sample the medium simultaneously, as a result they will receive and consequently will acknowledge the data packet. Note that, there is high probability that the ACK packets may collide. As a result, according to basic M-ContikiMAC, if TX does not receive its expected ACK, it will postpone the burst transmission for the next preamble cycle and will retransmit in anycast the collided data packet (see Figure 5.3). However, since RX1 and RX2 are not aware about the collision of their acknowledgements, they will forward the previously received packet further to the sink and will keep their radio turned ON, since they consider that more packets will arrive due to the burst flag (Figure 5.4a). As a result, this situation generates duplicate packets in the network that lead to collisions as well as high interference energy consumption values.

5.3.4 Toward ME-ContikiMAC

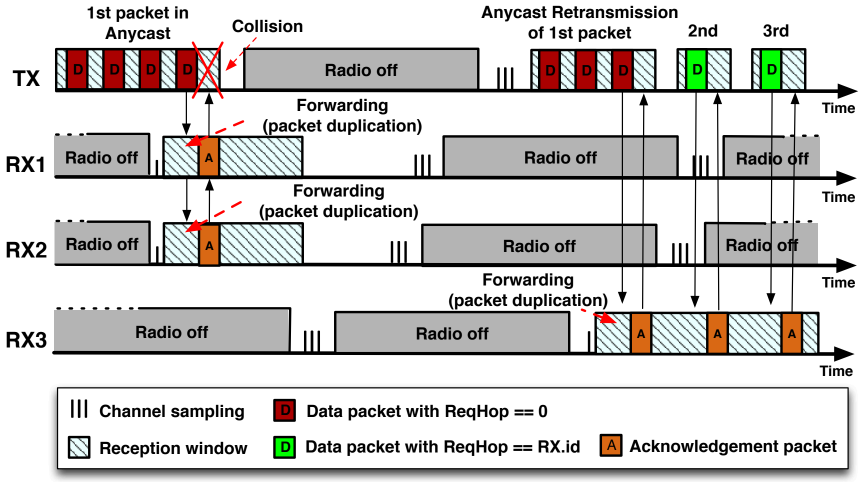
Hereafter, we present ME-ContikiMAC, the enhanced version of M-ContikiMAC, to handle mobility under dense networks by avoiding duplicates and resulting in a significant delay degradation.

Packet duplication control mechanisms

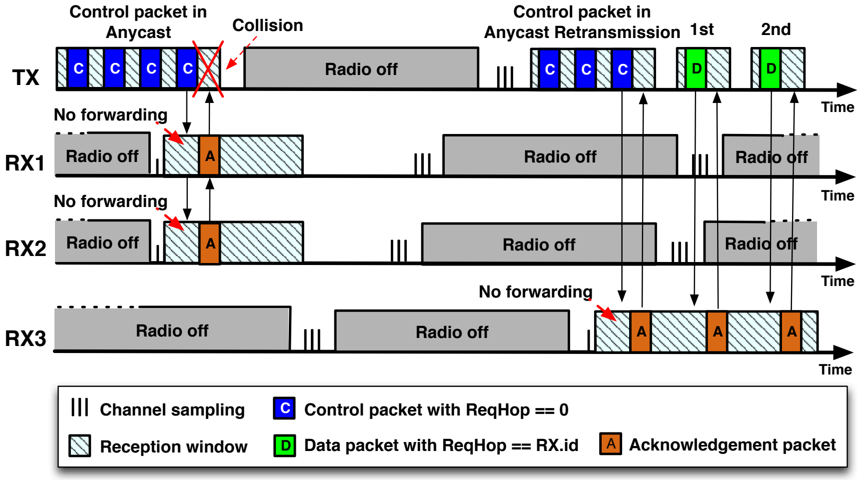
TX expecting to transmit n data packets in burst, will transmit one additional control packet upfront, $n + 1$ packets. In particular, it will repeatedly send a control packet in anycast which will be labeled not to be forwarded, while ReqHop equals to zero. If there will be an ACK collision for the control packet, the sender will retransmit it, while the receivers will not forward it further to the sink. Once the transmitter receives the corresponding ACK for the control packet, it will initiate the burst procedure (with data packets) to its newly temporary parent, as it is depicted in the Figure 5.4b. As a result, according to our simulation evaluation (see later in Section 5.4), we significantly reduce the unnecessary traffic in the network.

More specifically, two static nodes, RX1 and RX2, that sample the medium simultaneously, both will receive the control packet (recall that RX1 and RX2 will not forward the received control packet) and consequently will respond with an ACK which eventually will collide. Hence, TX will not be acknowledged for its control packet, thus, it will postpone the burst transmission of n data packets to the following preamble cycle. Once TX wakes-up, it will proceed again to the parent discovery mechanism by retransmitting the control packet until it receives an ACK from a static node (e.g., RX3). Afterwards, TX will initiate the burst process to the new discovered parent (Figure 5.4b).

We now further optimize the packet duplication control mechanism. Indeed, we observed that when a mobile node intends to establish a new connection, after a link disconnection, by utilizing the previously presented recovering mechanism of M-ContikiMAC (Figure 5.3), multiple packet reception issue arises again (see Figure 5.5a). To overcome this phenomenon, we configure

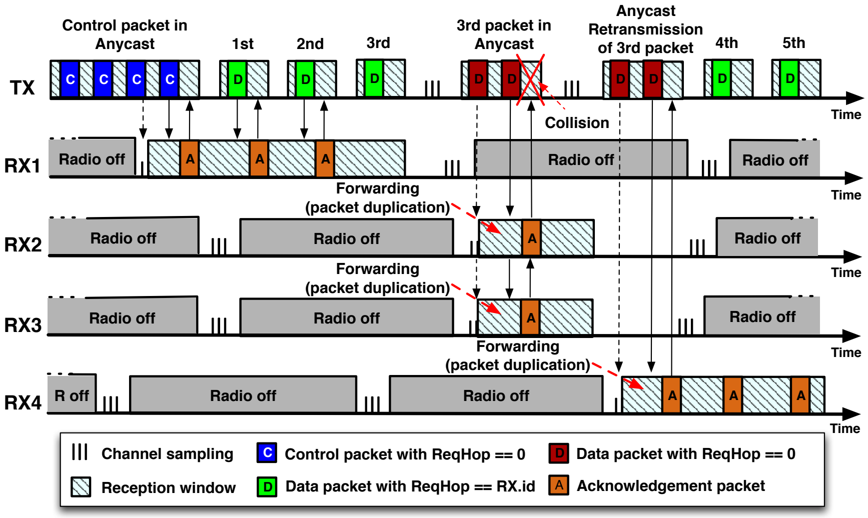


(a) When two or more nodes have similar sampling phase, then a packet may be received by two or more nodes, that will in turn forward it. This situation leads to packet duplication at the sink.

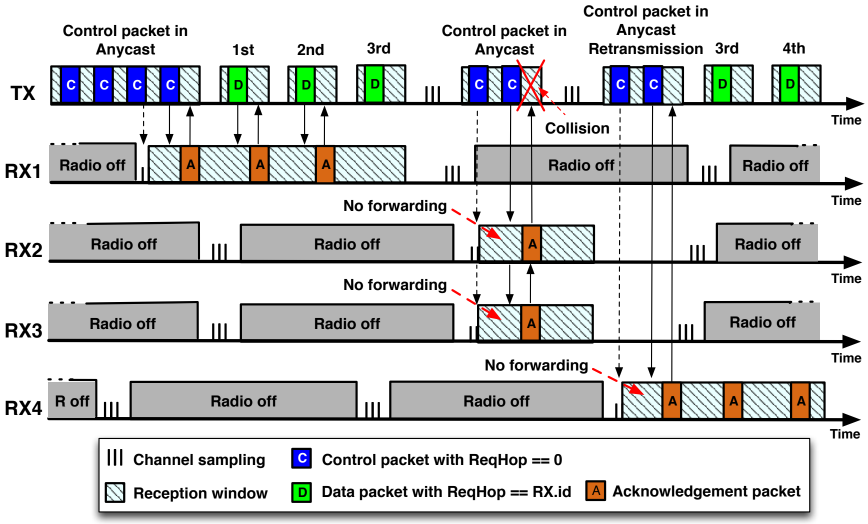


(b) Packet duplication control mechanism of ME-ContikiMAC: By introducing control packets before the burst transmission we achieve the reduction of the duplications in the network.

Figure 5.4: Case 1: Packet duplication issue arises at the beginning of the burst transmission (upper) and enhancement of ME-ContikiMAC with packet duplication control mechanism (bottom).



(a) To recover a connection by transmitting a data packet, we may end up with duplications if two or more nodes will sample the medium simultaneously.



(b) ME-ContikiMAC in enhanced connection recovering depiction: Upon a link disconnection the mobile node transmits a control packet upfront to avoid duplications.

Figure 5.5: Case 2: Packet duplication issue arises during the burst transmission (upper) and enhancement of ME-ContikiMAC with packet duplication control mechanism (bottom).

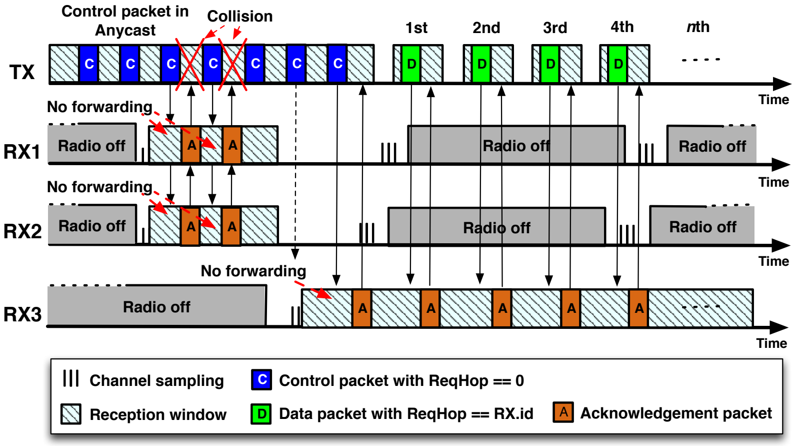


Figure 5.6: ME-ContikiMAC in delay-enhanced illustration: Mobile nodes in agressive state.

the TX during next-hop discovering procedure, instead of transmitting (by employing anycast) the data packet with ReqHop set to *zero*, to retransmit upfront a control packet (with ReqHop equals to 0) which will be labeled not to be forwarded. Thus, RX1 or RX2 will forward a data packet only when a link with TX is established. As a result, network performance is improved, congestion and collisions in the network are avoided while better energy efficiency is achieved. The detailed procedure of network reconnection of ME-ContikiMAC is illustrated in Figure 5.5b.

Delay enhancement

In this subsection, we will present the optimization of M-ContikiMAC in terms of delay. As can be observed from Figures 5.4a and 5.4b, according to the original version of ContikiMAC once the transmitted packet collides, the transmitter stops the ongoing preamble cycle and cancels the packet transmission. TX then retransmits either data or control packet in the following preamble cycle. Thus, TX waits for a complete preamble period (e.g., 125 ms, 500 ms) before retransmitting the packet. Apparently, this default operation of ContikiMAC induces high delays, especially when the nodes in the network are configured in long preamble-sampling frequencies such as 500 ms or 1 s. In order to overcome this barrier, we consider that the mobile nodes should have the priority to access the medium and, thus, we introduce aggressive nodes in the network. In particular, we configure the mobile nodes to behave more actively compared to the static, in order to receive privilege over the static nodes and gain the wireless medium. In order to do so, we appropriately modify the ME-ContikiMAC protocol to allow the mobile nodes to continue transmitting their packets during the preamble cycle even if their ACKs collide. By doing so, we achieve to significantly reduce the attained delay values. Figure 5.6 illustrates our proposed delay enhancement.

5.4 Performance Evaluation of ME-ContikiMAC

In the previous Section, we have presented the design of both M-ContikiMAC & ME-ContikiMAC, and discussed to which extent they may improve network performance when compared to major contributions in the related literature. Hereafter, we present a thorough performance evaluation of our proposals. In order to evaluate the efficiency of ME-ContikiMAC, we have run a set of simulations over COOJA [Ost+06] with Sky motes. Moreover, we utilized BonnMotion [Asc+10], a tool to generate mobility in the network. For comparison purposes, we also implemented on top of the Contiki OS and compared ME-ContikiMAC both against our previous work M-

Topology parameters	Value
Topology	Regular grid & random (50×40)
Number of nodes	40 fixed & 8 mobile sensors
Number of sources	47
Node spacing	$x = 6\ m\ /\ y = 8\ m$
Network degree	13.6
Mobility parameters	**Value**
Mobility model	Random waypoint
Velocity	Low speed: from 0.5 m/s *to* 2 m/s
	High speed: from 2 m/s *to* 8 m/s
Simulation parameters	**Value**
Duration	54 *minutes*
Application model	Mobile nodes: Burst: 16 $pkts/90\ s$
	Static nodes: CBR: 1 $pkt/30\ s$
Number of events	Mobile nodes: 4096 *pkts*
	Static nodes: 3990 *pkts*
Payload size	33 *Bytes*
Routing model	Static network: Gradient [Wat+09]
	Mobile nodes: Opportunistic
Number of hops	Multihop (5 hops maximum)
MAC model	Mobile nodes: MoX-MAC, MOBINET-S, MOBINET-R, M-ContikiMAC & ME-ContikiMAC
	Static nodes: ContikiMAC
Sampling frequency	125 *ms*
Maximum retries	3
Hardware parameters	**Value**
Antenna model	Omnidirectional CC2420
Radio propagation	2.4 *GHz*
Modulation model	O-QPSK
Transmission power	-10 *dBm*

Table 5.1: Simulation setup

ContikiMAC and state-of-the-art protocols such as MoX-MAC and MOBINET, both selective (i.e., MOBINET-S) and random (i.e., MOBINET-R) mode. Note that we deactivate the phase-lock optimization function from the default configuration of ContikiMAC, in order to provide fair analysis and thorough comparative study.

Our simulation scenario involves 40 fixed nodes (including the sink) that are uniformly (i.e., grid) or randomly distributed in an area of $50 \times 40\ m$, with network degree 13.6 in average, similarly to dense wireless lighting control networks [Dan+15]. Moreover, there are 8 mobile nodes that move by employing a random waypoint mobility model, with two different velocities. More specifically, the low speed (i.e., from 0.5 m/s *to* 2 m/s) that represents a human walk, and high speed (i.e., from 2 m/s *to* 8 m/s) that represents a typical jogging speed. In this study, we present application-dependent (i.e., time-driven) results where the mobile nodes transmit bursts of 16 packets every 90 *sec* while the static nodes transmit by utilizing a Constant Bit Rate (CBR) of 1 *pkt* per 30 *sec*, having as a result more than 8000 *pkts* transmissions in total. As far it concerns the MAC layer, we have set a maximum of three retransmissions and the sampling frequency to 125 *ms*. We choose the packet size to be equal to 33 *bytes* that corresponds to all necessary information for MAC, routing and application operations (e.g., node ID, packet sequence, burst and ReqHop flags, sensed values). Furthermore, we used a radio model based on disks for the sake of clarity, where each node emits at -10 *dBm* transmission power, imposing thus, multi-hop communications among the mobile nodes and the sink (up to five hops). At the routing layer, we rely on a broadly used scalable and under realistic conditions gradient

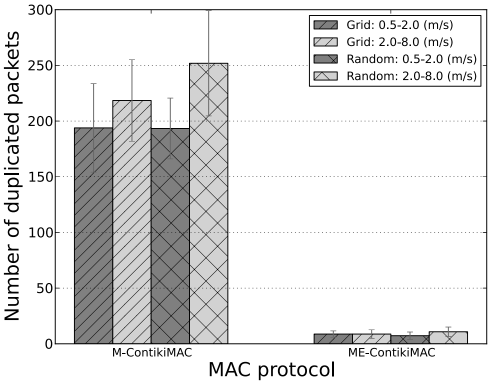


Figure 5.7: Total number of packet duplications at the sink.

protocol [Wat+09] that generates a routing tree rooted at the sink (i.e., by employing as a metric a number of hops towards to the sink) having low overhead. Finally, each simulation lasted 54 *min*. The details of the simulation setup are exposed in Table 5.1. The results hereinafter show the performance gain of our proposal in terms of delay (i.e., both 1-hop and end-to-end) and energy consumption. In fact, we demonstrate that two different MAC configurations (i.e., statically oriented and mobile oriented) can cooperate with each other, so that the mobile nodes can smoothly coexist within a static network, without causing inefficiencies in the network.

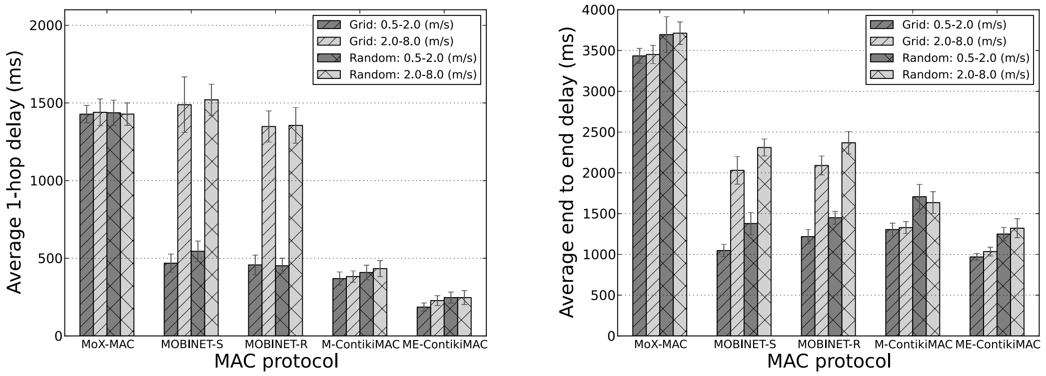
5.4.1 Packet duplication

Figure 5.7 illustrates the total number of packet duplications at the sink node both for M-ContikiMAC and ME-ContikiMAC. As can be observed, uncontrolled anycast transmissions may cause multiple packet receptions in the network. Indeed, the more dense is the network higher the probability of having packet duplications. As a result, network traffic congestion, channel occupancy and medium access competition increase, and, the probability of packet retransmission gets higher values due to the potential collisions in the network. The proposed ME-ContikiMAC succeeds in significantly reducing the multiple reception of a single packet at the sink node by employing the proposed packet duplication control mechanisms. More specifically, we succeed to reduce duplications by more than 90% comparing to our primary work, M-ContikiMAC. Consequently, ME-ContikiMAC significantly decreases the number of unnecessary packets in the network and, thus, potential collisions.

5.4.2 Delay performance

Figures 5.8a and 5.11c illustrate the average 1-hop (from mobile to any static node) and end-to-end (from mobile to sink node) delay per packet transmission. Both 1-hop and end-to-end delay include the channel sampling period, initial back-off, potential congestion back-off, potential retransmission delay and the transmission time of the preamble. Overall, the protocols within high velocity scenarios perform worse than in the low ones, mainly due to the difficulties of a link establishment between the mobile and static node (i.e., more frequent connections/disconnections). Furthermore, in the end-to-end delay, all protocols perform worse in random topologies. This phenomenon takes place due to the potential bottleneck links that are more prone to appear in random topologies, having as a result nodes to handle heavy traffic.

Furthermore, our simulation results show that both in 1-hop and end-to-end delay, MoX-MAC attains the worst performance. This could be explained by the phenomenon that in MoX-MAC a mobile node first overhears the whole transmission between the static nodes, and later it



(a) Average 1-hop delay, mobile to static node.

(b) Average end to end delay, mobile to sink.

Figure 5.8: A thorough delay performance evaluation of M-ContikiMAC and ME-ContikiMAC protocols

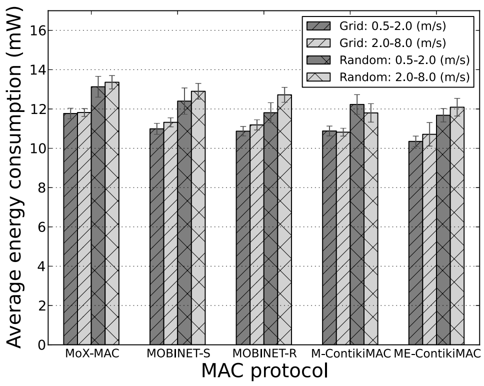


Figure 5.9: Average energy consumption.

transmits its data packet to the detected sender. On the other side, MOBINET shows promising results in end-to-end delay for low speed scenarios, and more specifically, for the selective version of MOBINET, since it selects the best parent among the list of the potential next hop nodes (see Table 5.2). As a result, it picks up the node closest to the sink in terms of hop. However, MOBINET presents poor performance in high speed scenarios, it may due to the insufficient time for a mobile node to overhear the transmissions from neighborhood nodes, before to transmit its data packets.

Finally, ME-ContikiMAC significantly improves both 1-hop and end-to-end delay for all the considered scenarios. Indeed, it reduces up to 60% the performance in high speed scenarios. These results are mainly due to the delay enhancement that we have presented in the previous section by introducing aggression to the mobile nodes. In addition, the reduction of the unnecessary transmissions in the network decreases potential collisions, and consequently retransmissions that have a major impact on the delay performance. Furthermore, as can be observed from the Table 5.2, all solutions have more or less the same amount of hops (the lower being for MOBINET because of its next-hop handling at MAC layer), thus meaning that the end-to-end delay reduction is independent to this metric. As a result, with ME-ContikiMAC we achieve a significant improved communication in mobile WSNs.

Scenario	MoX-MAC	Mobinet-S	Mobinet-R	M-ContikiMAC	ME-ContikiMAC
Grid: *slow*	3.34 (0.04)	2.53 (0.05)	2.99 (0.04)	3.46 (0.06)	3.26 (0.07)
Grid: *fast*	3.31 (0.05)	2.47 (0.05)	2.95 (0.06)	3.48 (0.09)	3.28 (0.04)
Random: *slow*	3.64 (0.09)	3.09 (0.13)	3.45 (0.09)	4.02 (0.29)	3.70 (0.22)
Random: *fast*	3.65 (0.09)	2.95 (0.15)	3.45 (0.09)	3.97 (0.22)	3.71 (0.10)

Table 5.2: Average (along with confidence interval) number of hops, from mobile to sink.

5.4.3 Energy consumption

In Figure 5.9 the average energy consumption per second for the whole network is presented for both grid and random topologies. The results show that the overhearing procedure has a straightforward impact on energy dissipation. As can be observed, ME-ContikiMAC consumes less energy (i.e., 1 mW in average) network-wide when compared to MoX-MAC, and both selective and random-based MOBINET protocols.

5.5 MobiXplore: Handover Delay Optimizing

Contrary to the statically-based deployments, applications in WSNs with mobile sensor nodes can cause frequent topology changes and the deterioration of established links. Thus, due to the mobility the quality of a communication link between a mobile transmitter and a stationary relay node significantly fluctuates, resulting in high end-to-end latency, and irregular packet arrival time (i.e., jitter).

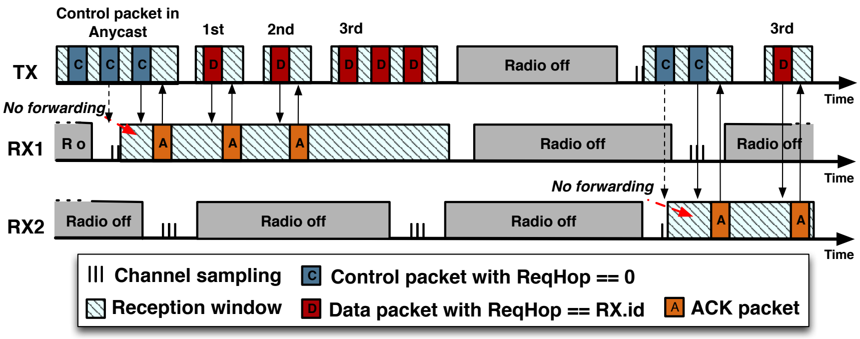
In consequence, in this Section, we introduce MobiXplore, a MAC scheme that allows a seamless transfer of communication (handover) to achieve low handover latency. More specifically, we highlight the remaining limitations, reconnection and handover issues, of ME-ContikiMAC and propose MobiXplore that overcomes them by introducing certain enhancements. Hence, by extending our previous work, we design and develop the MobiXplore, an enhanced medium access scheme for mobile sensor nodes that guarantees a better selection of the next-hop node, a temporary parent, while at the same time optimizes the handover procedure.

5.5.1 Motivation

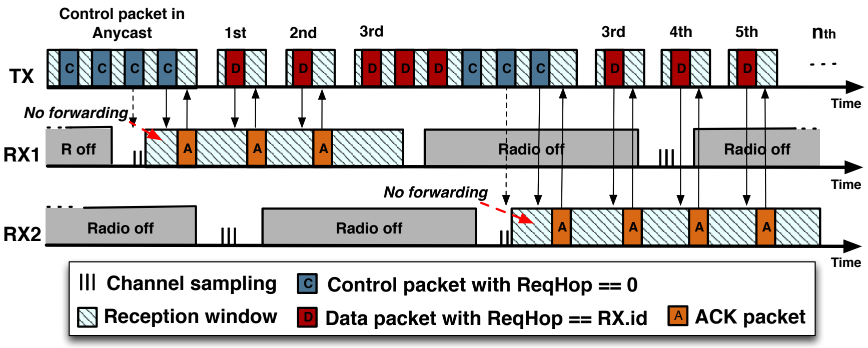
Our analysis and simulation experience have showed that M-ContikiMAC and ME-ContikiMAC do not perform effectively under intermittent link connections (e.g., due to mobility) between a mobile and its temporary parent (i.e., static node). As can be observed from Figures 5.10(a), according to the handover procedure of ME-ContikiMAC, upon a link disconnection during the burst transmission of n packets, the mobile transmitter (i.e., M) abandons the ongoing preamble cycle, and thus, cancels its scheduled transmissions. Indeed, it will postpone the burst transmission for the next preamble cycle. Such situations are tightly linked to next-hop selection and handover issues. So far, to the best of our knowledge, such issues have not been much investigated in the context of MAC layer in WSNs.

5.5.2 Design of MobiXplore

We consider scenarios where nodes transmit in burst, thus, burst notification flag of ContikiMAC is activated. Figures 5.10(a) confirms what was anticipated. Indeed, ME-ContikiMAC induces high reconnection and handover delays, especially when the mobile nodes are configured in long preamble-sampling frequencies such as 500 ms or 1 s. In fact, the mobile node will waste a whole preamble cycle (or more) before initiating the new-parent discovery procedure by retransmitting control packets in anycast. Due to the contention based network, upon successful CBR, the transmitter will start the process. Otherwise it will proceed to backoff.



(a) Handover delay issue due to the default setup of ME-ContikiMAC.



(b) MobiXplore: on enhament of the handover mechanism.

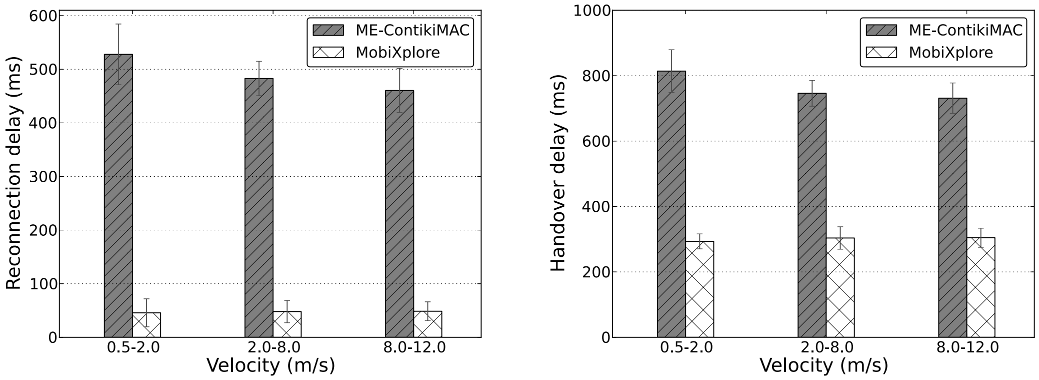
Figure 5.10: MobiXplore in handover enhancement illustration: Mobile nodes in aggressive behavior.

In order to improve the previously discussed reconnection and handover issues, we consider that the mobile nodes should have the priority to access the medium and, thus, we introduce aggressive mobile nodes in the network. In particular, we configure the mobile nodes to behave more actively compared to the static, in order to receive privilege over the static nodes and regain the wireless medium as soon as possible. In order to do so, once a mobile node detects the network disconnection, it immediately initiates the parent discovery procedure without the involvement of the CSMA layer. More specifically, by employing MobiXplore protocol we appropriately reconfigure MAC layer parameters in order to allow the mobile node to continue with transmission of control packets, during the same preamble cycle, once it does not receive acknowledgements for the data packets due to the transmission range. By doing so, we achieve to significantly reduce the attained delay values. Note that our proposal is compliant with any preamble-sampling family of MAC protocols. We illustrated this by embedding our proposed solutions as extensions of ME-ContikiMAC, see Figures 5.10(b).

5.6 Performance Evaluation of MobiXplore

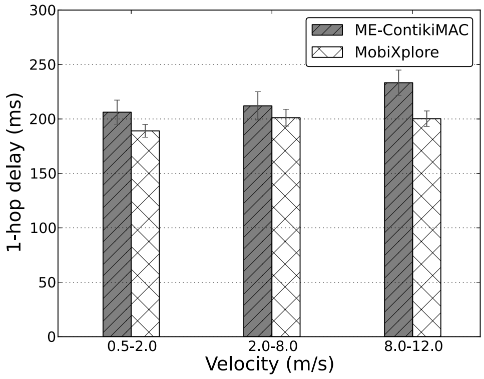
5.6.1 Simulation Setup

In order to evaluate the efficiency of MobiXplore, we have implemented it on top of ME-ContikiMAC and run a set of simulations over COOJA [Ost+06] with Sky motes (a simulator for the Contiki OS) and we have utilized BonnMotion [Asc+10] to generate mobility. Note that we kept activated the phase-lock optimization function. Our simulation scenario involves 8 mobile



(a) Average reconnection delay, from mobile to any static node.

(b) Average handover delay, from mobile to any static node.



(c) Average 1-hop delay, from mobile to any static node.

Figure 5.11: A thorough delay performance evaluation of MobiXplore in regular grid topology.

nodes with various speeds (i.e., 0.5-2, 2-8 and 8-12 m/s) utilizing a random waypoint mobility model and 40 fixed nodes (including the sink) that are uniformly distributed in an area of 50×40 m, with network degree at 6.15 in average. In this study, we demonstrate application-dependent results where the mobile nodes transmit bursts of 32 packets every 120 sec while the static nodes transmit by utilizing a CBR of 1 pkt per 30 sec, having as a result more than 13300 $pkts$ transmissions in total. As far as it concerns the MAC layer, we have set a maximum of three retransmissions and the sampling frequency to 125 ms. We choose a packet size equal to 38 bytes, which corresponds to all necessary information for MAC, routing and application. We used a radio model based on disks for the sake of clarity, where each node emits at -15 dBm transmission power. At the routing layer, we utilize the gradient protocol [Wat+09] and the number of hops as metric. For comparison purposes, we also implemented and compared MobiXplore against ME-ContikiMAC which outperforms a number of state-of-the-art solutions (including M-ContikiMAC, MoX-MAC and MOBINET), by terms of reducing both delay and energy consumption [Pap+15d]. Finally, each simulation lasted for 68 minutes.

5.6.2 Simulation Results

Figures 5.10 illustrates the average reconnection (i.e., time to establish a new link), handover and 1-hop delays per packet transmission from any mobile to any static node. The handover and 1-hop delays include the channel sampling period, initial backoff, potential congestion backoff, potential retransmission delay, propagation time and transmission time of the preamble. The results demonstrate that MobiXplore presents a very promising performance. More specifically, MobiXplore significantly improves both reconnection and handover delays for all the considered velocities by more than 90% and 55% respectively, which in turn reduces the 1-hop delay up to 14% when compared to ME-ContikiMAC, see Figure 5.11c. These results are mainly due to the MAC layer enhancements that we have detailed in the previous Section by introducing aggression to the mobile nodes (i.e., transmitting control packets in the same preamble cycle). As a result, by utilizing MobiXplore we are able to achieve a significantly improved communication for mobile WSNs.

5.7 Conclusions

In this Chapter, we have introduced ME-ContikiMAC, an enhanced version of M-ContikiMAC protocol from our previous work, for tackling mobility issues and provide reliable, low delay and energy efficient communication between mobile and static nodes for WSNs. Our investigation demonstrated that two different configurations (i.e., statical oriented and mobile oriented) of the same MAC protocol can be combined, so that the mobile nodes can smoothly co-exist within a static network, without causing performance degradation for the static nodes that reside in the network. Note that, the proposed mechanism in this study can be applied to various preamble-sampling protocols (e.g., X-MAC). We performed a thorough simulation performance evaluation over two topologies (uniform and random nodes distribution) on top of COOJA simulator that demonstrates promising results. In fact, according to our results ME-ContikiMAC significantly enhances the overall network performance by reducing packet duplications (up to 90%), channel occupancy and delay while keeping at low level the energy consumption, when compared to M-ContikiMAC and other state-of-the-art protocols.

Furthermore, we have presented MobiXplore, a MAC scheme that allows a seamless handover. Our initial simulation performance evaluation over COOJA simulator provides us promising results in terms of both reconnection and handover delays (i.e., more that 55% reduction when compared to ME-ContikiMAC) while allows for uninterrupted sensing.

5.8 Perspectives

Our ongoing and future work consists of further investigating this lead in mobile sensors. More specifically, we will continue our study of the energy consumption and handover schemes for mobile sensor nodes and will try to reduce it with optimized algorithms. Moreover, our vision is to further explore ME-ContikiMAC by performing a set of experimental studies over FIT IoT-LAB, a very large scale WSN testbed [Pap+13]. Thus, we plan to evaluate our mechanism under real-world scenarios and improve it by learning from the challenges that may arise from the experimental procedure.

In the long term, we would like to also investigate solutions that detect arrivals of mobile nodes in a new wireless sensor network, in order to anticipate the interaction of mobile and static sensor nodes is a promising approach that will allow real-world deployments, such as advanced surveillance systems [Fre+13].

In Chapters 4 and 5, we have investigated the MAC layer very thoroughly. In particular, we studied the impact of traffic variations to the network performance, and proposed T-AAD, a scheme that automatically tunes its MAC layer parameters at runtime to quickly adapt to the traffic load changes in the network. Moreover, we explored the lead of dynamic and bursty traffic in mobile sensor nodes within mobility-aware WSNs, we thus, enhanced the integration of mobile sensors in static network, without causing inefficiencies in the network. Therefore, we proposed M-ContikiMAC, ME-ContikiMAC and MobiXplore, new MAC layer schemes compliant with preamble-sampling MAC protocols that allow for low-power and low-delay mobile to static node communication.

Quite naturally, we then focused on examining to what extent simple routing schemes (e.g., opportunistic) using the service provided by our enhanced MAC layer would provide good performances and how to improve them when it's not the case.

Chapter 6

A Packet Duplication Control Mechanism for Opportunistic Routing

In traditional routing protocols designed for WSNs, each sensor node is related to one or more parents that will forward its data packet to the sink station. This technique performs well in static topologies with homogeneous configurations. However, it fails to cope with unstable networks such as mobility-aware networks or node (backbone) failures. Opportunistically oriented routing scheme is an approach to address the previously reported issue. In opportunistic routing, the packets are transmitted to a set of potential forwarders and then forwarded by the neighbor that first acknowledges the packet. Yet, several former studies demonstrated that in some cases, a single packet may be forwarded by multiple neighbors simultaneously. This situation leads to packet duplication and consequently to increased channel occupancy and energy consumption in the network. After intensively studying the MAC layer in WSN, we here continue on investigating the impact of MAC configurations and reconfigurations at each node (i.e., local decisions, local impact) on the overall network (i.e., whole structure, global impact). In this Chapter, we study to what extent the previously reported phenomenon depends on both the topology density and the nodes MAC configuration. We then introduce a mechanism that handles the potential deafness in the network through heterogeneous configuration among the nodes in the network. We do so through local, dynamic and automatic MAC parameters adaptation, in order to reduce unnecessary traffic, channel occupancy and energy consumption due to packet duplication in opportunistic networks. Finally, we provide both theoretical analysis and experimental campaign to detail the benefits of our approach.

Contribution

This chapter presents the following contributions:

1. We first study the probability that multiple nodes receive the same data packet, depending on the size of the potential forwarders and the sampling frequency;
2. We introduce heterogeneity among the nodes in the WSN in order to reduce the probability of having multiple receivers, for a single packet;
3. Finally, we evaluate our adaptive scheme through an experimental campaign over FIT IoT-LAB [Pap+13] testbed. In addition, we compare our mechanism with a statically pre-configured network by employing X-MAC [Bue+06] protocol.

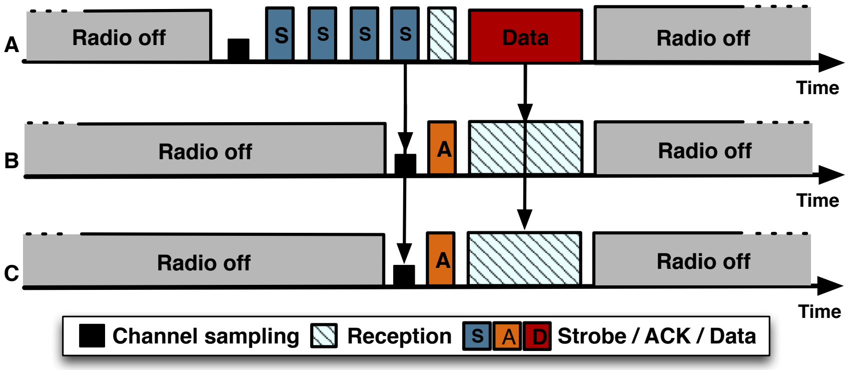


Figure 6.1: When multiple nodes share the same wake-up slot, a single packet may be received by two or more nodes.

6.1 Context Description & Motivation

6.1.1 Packet Duplication Issue

WSNs operating with preamble-sampling MAC protocols under homogenous configurations, a single packet could be received by multiple neighbors. Indeed, two or more potential forwarders may sample the radio channel simultaneously or almost simultaneously, thus detecting and acknowledging a strobe associated to the ensuing packet transmission. The transmitter will most probably receive one of those acknowledgements and directly transmit the data packet. However, multiple receptions will be triggered by all neighboring nodes that have acknowledged the preamble. This phenomenon leads to several distinct forwarded messages, recursively propagated at each hop, and eventually to packet duplication at the sink. In parallel, overall network performance is affected due to the corresponding traffic and channel occupancy increase.

In this Chapter, we introduce an auto-adaptive scheme to leverage the packet duplications in WSNs, operating in a cross-layer scheme (i.e., preamble-sampling MAC and opportunistic routing protocol). In such context, the probability of two or more nodes simultaneously sampling the medium for incoming packet is strongly correlated to the total number of potential forwarders along with their wake-up interval (further referred to as duty-cycle) [Lan+12]. In this investigation, we introduce heterogeneity among the nodes in the WSN in order to reduce the probability of having multiple receivers, for a single packet. Indeed, the nodes dynamically and automatically regulate their configurations in localized manner, without endangering network disconnection. As a result, our lightweight mechanism by enabling potential deafness (e.g., risk for node isolation) in the network improves the network performance and reduces the energy consumption.

6.1.2 Problem Statement: Packet Duplication

In WSNs relying on opportunistic routing, two factors need to be fulfilled for a single data packet to be received by several neighbors (Figure 6.1). First, two or more eligible nodes have to sample their radio channel while the packet is being transmitted. Depending on the channel sampling rate and network local density, the probability of this condition to be met varies. Indeed, the more frequently nodes sample their radio channel (and thus the higher its duty-cycle) and the more potential forwarders involved, the more likely several of those are to be awake during the packet transmission. In addition, among all these awaken neighbors during data transmission, more than one of these have to successfully catch the preamble and acknowledge it.

Parameters	**Value**
Total number of nodes	240 fixed sensors
Number of sources	1
Duration	10 $minutes$
Application model	CBR: 1 $pkt/5\ s$
Type of transmission	Broadcast & Anycast
Payload size	10 $Bytes$ (+6 byte MAC header)
Number of events	100
Antenna model	Omnidirectional CC1101 interface
Radio propagation	868 MHz
Modulation model	GFSK
Transmission power	$-10\ dBm$

Table 6.1: Experimental Setup

This problem can be formalized and quantified by adapting the birthday paradox, as follows. Each node wakes its radio up to sample the medium for incoming messages under regular intervals. This channel sampling procedure takes about 7 ms on typical radio chipsets. Let us consider that a node transmits packet in an anycast fashion. Among all the potential receivers, let one of those be the first to acknowledge this packet. In this case, the probability of any other potential forwarder to sample the medium at the same time slot (i.e., catch the preamble and acknowledge it) is given by Equation (6.1), with the channel sampling duration called $ONtime$ (e.g., 7 ms).

$$P_1(X) = \frac{1}{\frac{wake-up\ interval}{ONtime}} = \frac{ONtime}{wake-up\ interval} \tag{6.1}$$

Instead of looking for any pair of nodes sharing the same wake-up slot as with the birthday paradox, we here consider that a node already caught the strobe, and then evaluates how many other potential forwarders share the same wake-up slot. By utilizing the number n of potential forwarders, we formalize in Equation (6.2) the probability of having two or more of those catching the preamble (and thus getting the data) at the same time slot. To do so, we first formalized its complementary: considering one first acknowledging node, the probability that no other potential forwarder samples its channel at the same time slot is given as follows:

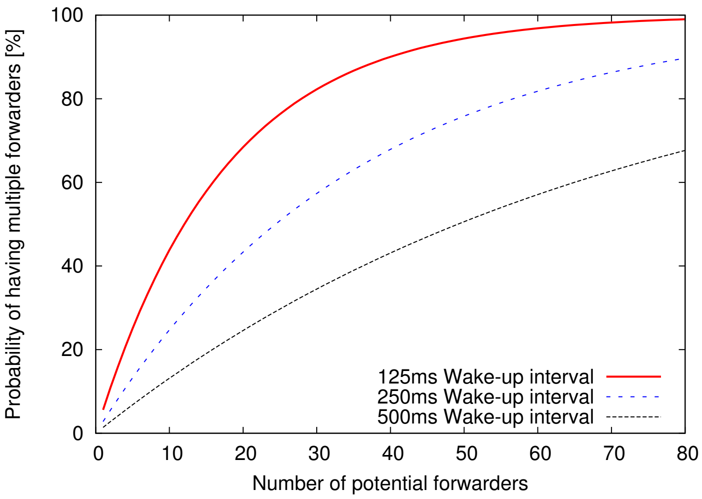


Figure 6.2: Probability that multiple nodes receive the same data packet, depending on the size of the potential forwarders and the wake-up interval.

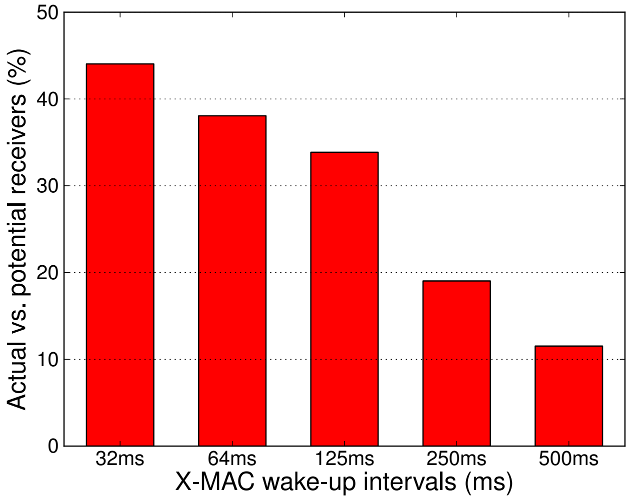


Figure 6.3: Experimental evaluation of the proportion of potential forwarders receiving a single message, depending on the nodes wake-up interval.

$$P_2(X) = 1 - \overline{P}(X) = 1 - (\prod^{n} P_1(X)) \tag{6.2}$$

Figure 6.2 uses equation (6.2) to represent the analytic probability of a single packet to be simultaneously received by multiple nodes. As shown here, this probability is strongly correlated to both the wake-up interval provided to the nodes in the network and the size of the potential receivers set. Indeed, both a low wake-up interval and a high number of potential forwarders increase the probability of a single packet to be received by multiple nodes. Also, this problem is critical, as most of the packet transmissions are affected, even at low density and high sampling interval. Note that, due to the multi-hop fashion of typical WSNs, this phenomenon may recursively affect forwarding at each hop.

We then provide an experimental evaluation over FIT IoT-LAB testbed [Pap+13] in order to validate our assumption. We run experiments with varying wake-up intervals, ranging from 32 ms to 500 ms. In this study, we follow a simple scenario in order to limit border effects. Therefore, we keep one sender and all other nodes of the platform as potential receivers. The sender transmits in CBR mode, one broadcast packet followed by an anycast packet (a communication of a single sender sending to a member in a subset group of potential receivers) every five seconds. In total, 200 packets are transmitted with transmission power at $-10dBm$. The details of the experimental setup are presented in Table 6.1.

Our intention is to evaluate the number of actual simultaneous receivers (i.e., anycast) among the maximum potential receivers (i.e., broadcast). Figure 6.3 presents the proportion of potential forwarders that receive a packet versus the nodes wake-up intervals. As can be observed, the results follow to the mathematical analysis. Indeed, the shorter the wake-up interval, the higher the probability of multiple receivers. Since, the unnecessary packets in the network increase, as a result the overall performance will degrade (e.g., delay, Packet Reception Ratio (PRR)).

6.2 Proposed Adaptive Scheme

In order to address the packet duplication problem, we introduce the potentiality of deafness in WSN. Through local adaptations, some node in the network may be deaf to others transmissions

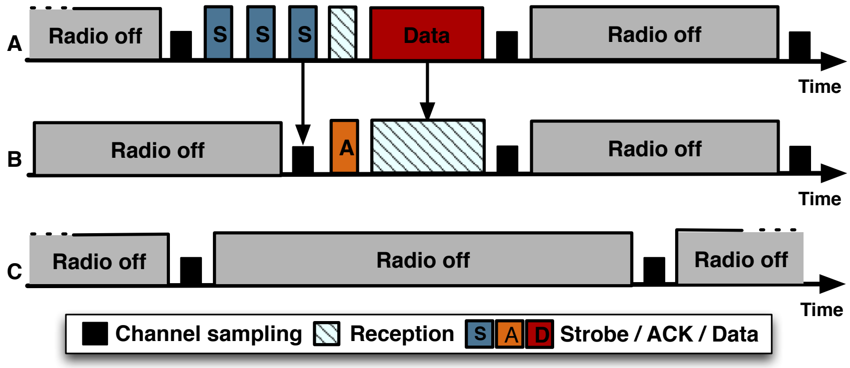


Figure 6.4: Here, node A tries to send a packet, but since C has longer sampling frequency, the transmission may fail.

(e.g., node C in Figure 6.4). In fact, we propose a mechanism for adapting the nodes wake-up interval (i.e., sampling frequencies) based on the local number of potential forwarders (i.e., designated by the routing layer as potential next-hops towards the sink). By doing so, we aim at reducing the probability that multiple nodes receive the same data packet without endangering network performance [Pap+14c].

This information can be obtained through various approaches. First, each node can overhear the messages broadcasted by its neighbors during the construction of the routing structure. This solution do not induce additional control traffic, but can only be performed when the routing tree is being built, and is thus best-suited for static networks. Another approach is to let each node broadcast HELLO messages at regular interval. Doing so requires additional control messages to be sent, but provides nodes with an exact view of their vicinity, in real-time. Another option is to let nodes overhear data messages transmitted by its neighbors over time during the deployment. This approach may only provide partial information (as some unicast messages will not be perceived due to radio duty-cycling) but is performed passively, at no energy cost. Each one of the above-mentioned approaches is well suited for.

As presented in Section 2.3.6, there are number of approaches in order to retrieve information about the neighboring potential forwarders. Each one of the approaches is well-suited for a specific scenario. In this study, we consider a static network relying on broadly used scalable gradient protocol which generates a routing tree rooted at the sink (i.e., using a number of hops as a metric). Thus, we obtained the local number of potential forwarders by overhearing messages broadcasted during the construction of the routing tree. However, our contribution can be implemented on top of many opportunistic routing protocols, and with any previously-mentioned method to retrieve the number of potential forwarders.

This value (i.e., the number of potential forwarders) can then be used in accordance with application-level performance parameters to affect each node in the network with a specific wake-up interval. Application-level performance parameters can range from expected lifetime of the deployment to Quality of Service (QoS) and resilience requirements (e.g., maximal delay or loss-rate), and depend on the application specificities and requirements [Mok+14], [BOY10]. For instance, patient remote monitoring systems usually require extra resilience (as every data is valuable), while long-term home-care systems require increased lifetime to function autonomously. Regarding our proposed solution, the first deployment example requires higher packet duplication rate for resilience while the latter needs the duplication probability to be minimal in order to maximize network lifetime.

Both application-level performance parameters and the size of the potential forwarders subset are provided as parameters of a Multi-objective Optimization Problem (MOP). The optimal solution to this MOP (or the *Pareto-optimal* solutions) represent the best-suited MAC parameters a node can be configured with to fulfill all application requirements. Thus, when the set of identified other potential forwarders (i.e., neighbors sharing the same opportunistic routing metric) is larger, the node will select a longer wake-up interval to fit with the lifetime requirement, without exceeding the reliability requirement. Conversely, a reduced set of identified other potential forwarders will imply that the node will select a shorter wake-up interval to ensure efficient management of incoming messages.

Thus, the proposed adaptive scheme addresses the unique forwarder problem without endangering the network performance. However, by having some nodes equipped with a shorter wake-up intervals than others, our approach may induce an inequity between them. Indeed, having heterogenous configurations among the nodes in the network leads to potential deafness of a node (Figure 6.4), which in turn may induce network partition. Moreover, nodes with shorter wake-up interval are more likely to catch incoming preambles, and will thus receive more traffic, resulting in a worse energy repartition between nodes in the network. Finally, network dynamics (e.g., mobility) can deeply affect the performance of our mechanism. These problems can be solved by a regular re-calculation of the set of other potential forwarders. Note that, while we did not consider this option in our campaign, the applicative parameters may change over time. In this context, the local MAC configuration of sensor nodes is updated in real-time, in accordance with the new solution to the multi-objective optimization problem.

6.3 Performance Evaluation

We now present a thorough performance evaluation for both the original static mechanism and our adaptive proposal. Our empirical analysis was conducted over the ICube's platform of the open large scale FIT IoT-LAB [Pap+13] testbed. In particular, we utilize 240 WSN430 nodes of ICube's platform where 239 nodes were randomly selected as data sources. The sink node is located at the top right of the testbed.

The enhancement presented in this work can be applied to various preamble-sampling protocols (e.g. X-MAC [Bue+06], ContikiMAC [Dun11] etc.). We implement our contribution in conjunction to the very well known X-MAC protocol due to its scientific contributions and popularity in real deployments [Dyo+10], [Zim+12], [YH12]. For the static version of opportunistic routing, we keep the preconfigured format of X-MAC with wake-up intervals ranging from 125 ms to 250 ms and 500 ms (i.e., S-125, S-250 and S-500 respectively). As the network behavior often heavily depends on the choice of the essential MAC protocol parameters (e.g., sampling frequency), we studied our mechanism with different configurations of those MAC parameters. Hence, in this study, we decided to set as applicative requirement a maximal duplication probability of 60% with no minimal lifetime, in order to keep the best trade-off between resiliency (i.e., packet duplication), QoS and energy consumption. Thus, according to the previously presented analysis (c.f. Section 6.1.2), nodes that have less than ten potential receivers will continue with their original MAC configuration (i.e., 125 ms), while the nodes that have more than twenty potential receivers will switch to 500 ms, finally the rest will switch to 250 ms.

The nodes use a time-driven application in CBR at 1 $pkt/100\ sec$, with maximum number of retransmissions was 3 for all protocols. We utilize a 10 byte data size, which corresponds to the general information used by monitoring applications (e.g., node ID, sequence number, sensed value). We set the transmission power at $-10\ dBm$ (i.e., transmission range $\simeq 4.45m$ [Pap+13]) to guaranty multiple communication hops. We implement our contributions using Contiki OS. Finally, the experiments last for 87 minutes (during this period almost twelve thousands transmissions occurred).

The results hereinafter illustrate the gain of our mechanism in terms of packet duplications, reliability, delay and energy consumption, compared to homogeneously and statically pre-configured opportunistic routing. In fact, the results show that our proposal enhance the packet duplication/energy dissemination trade-off. Moreover it shows that heterogeneous configurations can be determined independently at each node in a localized mode. The details of the experimental setup are exposed in Table 6.2.

Platform parameters	**Value**
Number of nodes	240 fixed sensors
Number of sources	239
Topology	($10\ m \times 8\ m \times 3\ m$) regular grid
Node spacement	One meter
Experimental parameters	**Value**
Duration	87 *minutes*
Application model	CBR: 1 $pkt/100\ s$
Payload size	10 *Bytes* (+6 bytes MAC header)
Number of events	11950
Routing model	Opportunistic
MAC model	X-MAC [Bue+06]:
	Sampling freq. (125, 250, 500 *ms*)
Maximum retries	3
Hardware parameters	**Value**
Antenna model	Omnidirectional CC1101
Radio propagation	868 *MHz*
Modulation model	GFSK
Transmission power	$-10\ dBm$
Battery	880 *mAh*, 3.7 *V*

Table 6.2: Experimental setup

Type of transmissions	**S-125**	**S-250**	**S-500**	**Adaptive**
Nr. of Original trans.	11950	11950	11950	11950
Nr. of Total trans.	76618	47362	32347	42726
Nr. of Unsuccessful trans.	28833	19008	13898	6182

Table 6.3: Traffic analysis

6.3.1 Packet Duplication

Figure 6.5 illustrates the average number of packet duplications at the sink node per transmission. The results present that the preamble length has a significant affect to multiple reception issue. In fact, the shorter is the preamble length higher the probability of having duplicated packet at the intermediate or at the sink node as we stated above in the third section. Hence, the total number of packets in the network (i.e., including both originally transmitted and forwarded packets) is increasing, while reducing the wake-up interval. Note that, multiple packet reception leads to higher traffic in the network, congestion, channel occupancy and the competition of the medium access, thus, it enlarges the probability of packet retransmissions due to the potential collisions in the network (when the receivers forward the same packet toward the sink).

Our proposed mechanism achieves to reduce the multiple receptions of a single packet at the sink. In particular, we reduce the packet duplication by more than 50% comparing to the S-125 (Figure 6.5). As a result, we decrease significantly the number of unnecessary packets in the network and consequently the collisions (see Table 6.3).

6.3.2 Reliability

We transmit almost twelve thousands messages, to estimate accurate PRR that is calculated as the total number of successfully received packets divided by the total number of transmitted packets. Figure 6.6a depicts the performance for the achieved PRR. Opportunistic routing that statically configured with wake-up interval at 500 *ms* achieves the worst results. As can be observed, the shorter is the sampling frequency the better performs the routing protocol. This

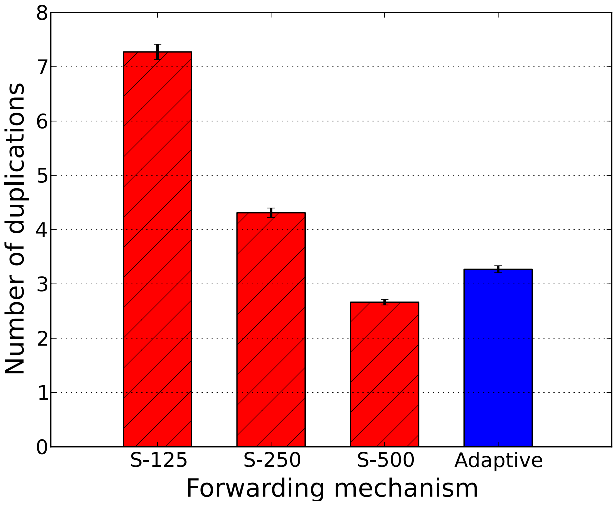


Figure 6.5: Average number of duplications at the sink

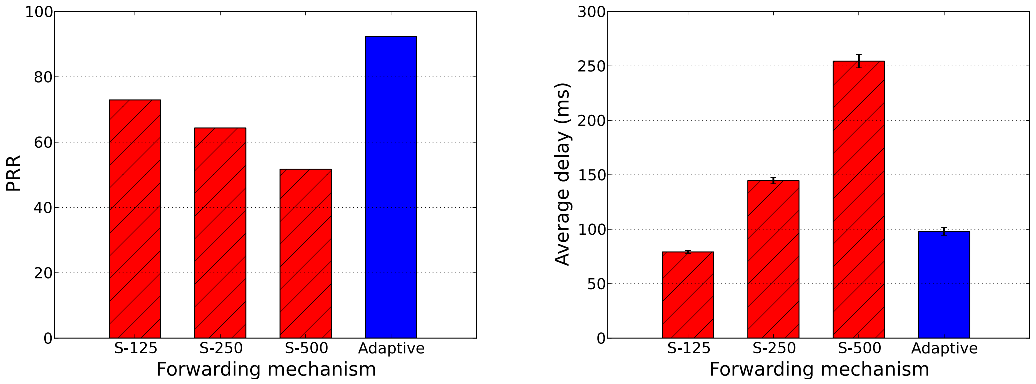
phenomenon is mainly due to the higher probability in a scenario with shorter wake-up intervals (i.e., 125 ms has 76618 transmissions), the packets to be received by the sink. Even though the collided transmissions increase when reducing the wake-up interval, yet the ratio unsuccessful to total transmissions remain at similar levels. As a result, opportunistic routing with shorter sampling frequencies perform better comparing to longer ones in terms of PRR.

Our proposal improves the performance of packet delivery ratio. In fact, the experimental results shows that it reaches more than 90% of PRR. By adapting the sampling frequency locally at each sensor node, we provide an harmony among the nodes. We achieve to reduce the unnecessary traffic which is the reason for the congestion and collisions in the network. Thus, the proportion of collisions to total number of transmissions reduced (Table 6.3), as a result, the PRR increased.

6.3.3 Delay

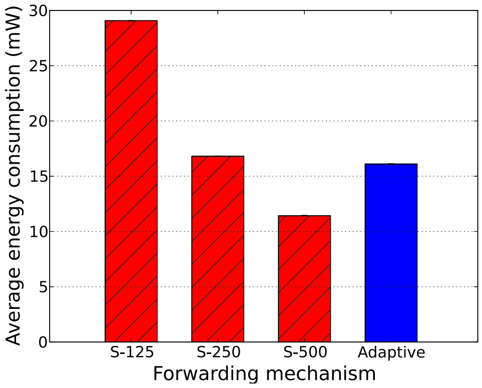
Figure 6.6b presents the average end-to-end delay per packet for all nodes. The multi-hop end-to-end delay includes the initial back-off, the channel sampling period, potential congestion back-off, potential retransmission delay and the transmission time of the preamble. It appears that, the sampling frequency has straightforward impact to the delay performance. Indeed, the results show that with wake-up interval at 500 ms we obtain the worst performance (i.e., 254.418 ms), since the long preambles of S-500 corresponds to high channel occupancy of the medium. Furthermore, the delay is even higher for the nodes located one-hop away from the sink due to the high probability of provoking congestion back-offs in the network, as well as due to MAC retransmissions. Indeed, due to radio channel competition and hidden terminals, large proportion of transmissions require several retransmissions before being acknowledged.

The adaptive scenario shows that even though a large proportion of nodes switch either to 250 ms or 500 ms, they still present competitive delay performance. This phenomenon is mainly due to the following parameters. Firstly, we reduce the unnecessary transmissions in the network, thus, we decrease the competition to the medium access and the collisions, consequently the retransmissions as well which has major impact on the delay. Secondly, the adaptive mechanism allows a certain portion of node operating at short wake-up intervals.



(a) Packet reception rate.

(b) Average end to end delay.



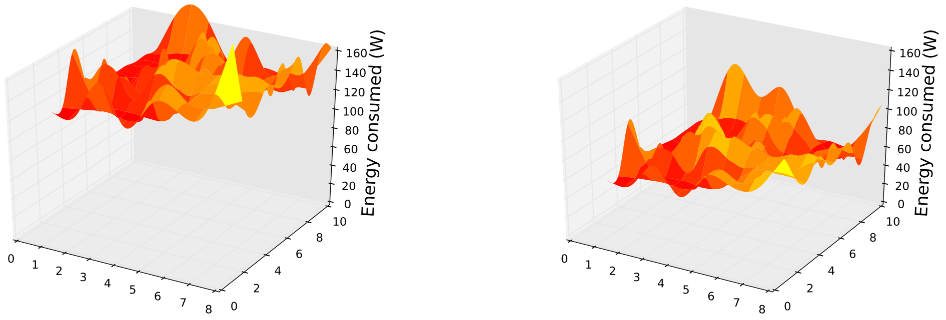
(c) Average energy consumption.

Figure 6.6: The impact of the adaptive mechanism to the network performance.

6.3.4 Energy Consumption

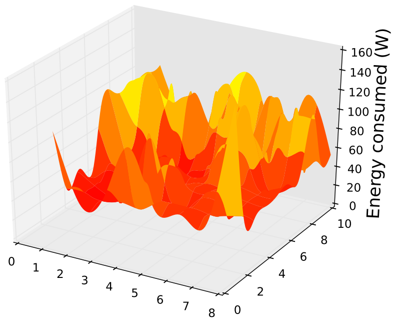
In Figure 6.6c, the average energy depletion per second for the whole network is presented both for homogeneously and statically configured, and for adaptive scenarios. The results show that the wake-up interval has a straightforward impact on energy dissipation. Indeed, S-125 performs better than S-250 and S-500 in terms of PRR and delay, while consuming almost three times more energy than S-500, due to the high sampling frequency, and even more the high traffic load in the network (Table 6.3). Our adaptive scheme consumes less energy network-wide. Indeed, with the adaptive mechanism, the energy consumption is reduced by about 45% when compared to S-125, due to the portion of nodes in the network that switched to lower sampling frequency (i.e., 250 or 500 ms). Moreover, among all operations assigned to sensor nodes, communication (i.e., transmission and reception) is the most energy-consuming. Thus, with our adaptive mechanism we achieved to reduce the unnecessary transmissions in the network, as a result, we manage to save a significant amount of energy.

Furthermore, we perform a mapping of the energy consumption throughout the second layer (i.e., 80 nodes) of the IoT-LAB platform, in order to analyze the disparity among the nodes. We calculated the total energy consumption of each node for the whole duration of the experiment (see Figure 6.7). As we can observe, the energy consumption in case of S-125 is very high while in the opposite nodes with configuration of S-500 consume much less. Globally, there is a homogenous behavior among the nodes in the network (i.e., S-125 and S-500). In fact, the nodes perform similarly with some exceptions (i.e., high peaks) that are more prone to appear either in the forwarding nodes (e.g., nodes 1-hop away from the sink mainly) or the sink node itself.



(a) S-125.

(b) S-500.



(c) Adaptive.

Figure 6.7: Total energy consumption for Adaptive, S-125 and S-500.

Conversely, in the case of adaptive, the results show a heterogenous behavior among the nodes. In particular, there are nodes that present very high consumption in contrary with others that consume less. This result is due to the adaptive phenomenon of our mechanism. In fact, the consumption results are varying over the network depending to the wake-up interval frequency of each node configuration. As a results, node with big set of neighbors switches either to 250 ms or to 500 ms of sampling frequency and consequently consume less energy while nodes that keep operating at 125 ms of wake-up interval come with higher consumption. However, in scenarios with heavy traffic loads, long preambles consume much more than the short ones due to the constant presence of transmission and reception of the preambles. As an example, the energy dissemination is much higher in places where nodes forward more. Typically these nodes are located around the sink.

6.4 Conclusions and Perspectives

In this Chapter, we examine to what extent our proposed auto-adaptive mechanism can mitigate packet duplication in opportunistic routing. Moreover, we present that in a scenario with opportunistic routing configured statically in homogenous preamble-sampling reveals a tradeoff between performance and energy consumption. On the one hand, when configured with long wake-up intervals (i.e., 500 ms), less energy is consumed but poor network performance is induced (e.g., high latency) and vice versa with sampling frequency of 125 ms. Hence, after studying the state of the art, we introduce an adaptive approach to address this problem. Our proposed mechanism is based on local configuration at each node, made on available information from control-packet transmissions (e.g., routing construction) and leverages the unnecessary overhead. We perform an exhausted performance evaluation of our mechanism through an experimental campaign over the FIT IoT-LAB [Pap+13] testbed. The derived results show that the adaptive mechanism achieves better performance in terms of delay (i.e., 100 ms) and PRR (i.e., $> 90\%$) since it reduces the channel occupancy, as well as energy consumption when compared to homogeneously and statically pre-configured opportunistic routing.

Our perspectives for future work consists of further exploring this lead, and in particular by investigating and introducing new metrics, such as link quality. Furthermore, our vision is to employ the proposed scheme within a mobility-aware WSNs.

Chapter 7

Conclusions and Perspectives

This Chapter concludes the thesis, reminding the addressed problems, highlighting the contributions, and opening up perspectives.

7.1 Conclusions

The goal of this dissertation was to solve certain key issues of the medium access schemes in WSNs within constrained environments, such as dynamic traffic and topology, as well as to enhance the overall performance of networks in which variable and bursty transmissions occur. Since the MAC layer is in charge for managing the communication between wireless sensor nodes, and even more, among all operations of a sensor node, transmission, reception, CPU and LPM, the communication is the most energy consumed. We therefore focused on improving the access to the wireless medium for low-delay communication in energy efficient manner.

In this manuscript, we considered the approach based on asynchronous instead of synchronized methods, mainly due to the efficiency and tolerance of preamble-sampling protocols on a large scale networks and dynamic topologies. Indeed, such an approach can cope with evolving network topology, scalability and node mobility, due to the efficient cooperation of sensor nodes in localized fashion within small groups of nodes.

All contributions, new MAC layer protocols for mobile sensor nodes and scheme to handle the traffic variations in the network, that we proposed aim to fulfill the previously presented purposes.

We started in Chapter 3, by presenting the FIT IoT-LAB platform, and demonstrating how conditions of real-deployments can be reproduced on it. We highlighted how an open testbed can be efficiently and successfully coupled with simulations, and moreover, we pointed out what thorough empirical campaign can bring to the evaluation and analysis of a protocol or an application. To this aim, we provided guidelines to translate simulation campaign to successful experimental deployments. In particular, we exposed for instance how both local and global energy consumption can be precisely monitored, and to what extent the link stability assumption can be removed or not, at the users choice. Thus, the evaluation campaign of WSN protocols can go one step further towards real deployment by removing the above mentioned assumptions, at little time cost and with limited complexity.

Then, in Chapter 4, we investigated the impact of traffic variations to the network performance. We confirmed that addressing a sensor network with unpredictable traffic in homogenous MAC configurations reveals a tradeoff between energy consumption and delay performance.

Hence, after a thorough study of the state of the art, we proposed T-AAD, a scheme that automatically reconfigures its MAC layer parameters at runtime to quickly adapt to the traffic load changes in the network. T-AAD, being compliant with most of the preamble-sampling based MAC protocols, and it allows for reduced energy consumption at both the receiver and sender sides, along with delay and channel occupancy reductions, when compared to the state-of-the-art solutions.

To evaluate the T-AAD scheme, we followed the methodology presented in Chapter, and we tested our auto-adaptive scheme both through simulation (i.e., COOJA simulator) and experimental studies over FIT IoT-LAB [Pap+13] testbed. Our performance evaluation campaigns showed that our approach manages to successfully reduce the energy consumption, while decreasing the latency.

Hence, once we proposed the traffic auto-adaptation scheme to address the dynamicity induced by varying traffic, we therefore decided to study dynamics and instability ensued from mobility in the network and its impact on MAC protocols. Consequently, we maintain the burst or varying traffic assumption, while removing the hypothesis of static infrastructure. Indeed, in Chapter 5, our work consisted of exploring the lead of dynamic and bursty traffic in mobile sensor nodes within mobility-aware WSNs. We thus first focused on improving the integration of mobile sensors in static network, without causing inefficiencies in the network.

In consequence, we proposed M-ContikiMAC, a new MAC layer protocol compliant with preamble-sampling MAC protocols. M-ContikiMAC extends the statically oriented ContikiMAC protocol to allow for mobile to static node communication. In particular, a mobile node by employing M-ContikiMAC protocol, will transmit in anycast (the first data packet of total n of burst) whose first acknowledging node will serve as responsible to forward it towards the sink.

We further discussed how to allow mobile nodes to co-exist and communicate with static nodes in the network, and highlighted the remaining limitations of M-ContikiMAC. The anycast-based packet transmissions present some anomalies in the network, such as high packet duplications. We therefore proposed ME-ContikiMAC to overcome them by introducing some optimizations to handle mobility even in very dense networks. In the enhanced version a delay optimization scheme and packet duplication control mechanism were introduced to reduce the 1-hop delay performance and mitigate the duplications respectively, which in turn reduces the channel occupancy, as well as energy consumption when compared to basic M-ContikiMAC as well as against other state-of-the-art solutions, such as MoX-MAC and MOBINET).

Finally, we investigating the reconnection and handover schemes in LPL-based MAC protocols. We came with MobiXplore, a MAC layer scheme that allows a seamless handover to address the previously mentioned issues. Our simulation results demonstrate that MobiXplore signicantly improves both reconnection and handover delays for all the considered velocities by more than 90% and 55% respectively, which in turn reduces the 1-hop delay up to 14% when compared to ME-ContikiMAC.

After thoroughly studying the medium access schemes both in case of varying traffic loads and dynamic network topologies, we focused on examining to what extent our enhancements at MAC layer can be used by upper layers. For instance, it was shown that opportunistic routing can achieve low-latency in duty-cycled (i.e., LPL) networks. However, several former studies demonstrated that in certain cases, a single packet may be forwarded by multiple neighbors simultaneously (due to the nature of anycast transmission mode). This situation leads to packet duplication and consequently to increased channel occupancy and energy consumption in the network.

In Chapter 6, we investigated to what extent the previously reported phenomenon depends on both the topology density and the nodes MAC configuration, and moreover, to what extent an auto-adaptive scheme can mitigate this issue in opportunistic routing. Our preliminary experimental analysis showed that an opportunistic routing configured statically in homogenous preamble-sampling reveals a tradeoff between performance and energy consumption.

We here proposed a mechanism based on local configuration at each node, made on available information from control-packet transmissions (e.g., routing construction) and leverages the unnecessary overhead. Our experimental results over the FIT IoT-LAB testbed showed that our proposed adaptive scheme considerably improved the performance in terms of delay and PRR since it reduces the channel occupancy, as well as energy consumption when compared to homogeneously and statically pre-configured opportunistic routing.

7.2 Perspectives

The contributions of this thesis can be extended in several directions. Let us now present some of them.

7.2.1 Experiments

Even though the majority of our contributions in this manuscript contain experimental evaluation and verification, we did not had the opportunity to test our proposed mobility-aware MAC protocols due to unavailability of the mobile robots. While we systematically utilized a realistic COOJA emulator that bridges the gap between simulation and experimentation, by remaining as close as possible to programming conditions of real embedded systems, experiments would allow to reveal more interesting details of this research [Pap+13].

Recently we received the TurtleBot2 robots and our engineer team is working to make them available for experimentation. Therefore, our vision consists of further exploring the proposed mobility oriented MAC protocols in real-world mobile robots. Indeed, our ongoing work is to investigate M-ContikiMAC, ME-ContikiMAC and MobiXplore schemes by performing a set of experimental studies over TurtleBot2 robots in FIT IoT-LAB testbed. Thus, we will evaluate the behavior of our protocols in a real-world and large-scale environment.

7.2.2 Integrating MobiXplore within Enhanced Opportunistic Routing

Environmental and wildlife monitoring, clinical medical and home-care monitoring, smart houses and cities are just some of the examples of WSN applications, where communication in such lossy links and low-power networks is challenging. In such deployments, energy-efficiency is one of the most important parameters, since nodes have to save energy to meet the lifetime requirements, and moreover, the deployments must be reliable and reactive, enabling interactive applications.

Furthermore, in wireless networks, link conditions vary at a fast time scale, and thus, the path at an instant may not be good at the next instant [Pap+13]. Hence, the optimal-path routing which is considered well suited for wired networks (i.e., due to the stable links), may not be an ideal approach for wireless communications such as WSN. Consequently, we rely on a different approach, where packets are sent opportunistically (i.e., anycast). Hence, a data packet is transmitted to the first potential forwarder (e.g., any neighbor closer to the sink in term of hops) acknowledging the corresponding message. Therefore, opportunistic routing schemes mitigate the impact of lossy links by exploiting the anycast nature of wireless transmissions [Liu+09b].

To conciliate the above goals, radio duty cycling is required at the MAC layer, while opportunistic routing is essential in order to leverage the latency. Several studies in the literature have shown that opportunistic routing is an efficient way to achieve low-latency and energy-efficiency in WSN [24, 26, 38, [Duq+13]]. For instance, in [Duq+13], the authors present ORPL, an extension of RPL that performs opportunistic routing. The authors claim that opportunistic nature of ORPL brings low latency, reliable communication in duty-cycled networks.

Consequently, our vision is to adapt our proposed mobility-aware MAC (i.e., ME-ContikiMAC, MobiXplore) services to better address the constraints imposed by the routing layer. By combining duty-cycle MAC protocols with an enhanced opportunistic routing protocol, we aim at achieving both low energy consumption and delay performance, without exceeding the reliability requirements.

7.2.3 Reliable Data Collection Schemes in Fault and Delay Tolerant Networks

By investigating the impact of our enhanced MAC solutions on the service that is provided to upper layers 7.2.2, naturally, we also focus on examining to what extent such an heterogeneity at MAC layer may support fault and delay-tolerance networks.

WSNs comprise of numerous sensor devices deployed for monitoring purposes. In many-to-one topologies collected data are typically relayed via intermediate nodes until they reach a sink station, that can store or processes the sensed information. For more than a decade there has been significant interest in designing and deploying WSNs, with applications ranging from monitoring patients and animals to large and costly structures [Opp+14], where low cost and easily deployed WSNs can provide significant benefits.

Such applications typically impose requirements for lossless communication with all measuring points to provide accurate information of the sensed measurements. However, constrained sensor nodes combined with the unstable wireless channel can pose significant challenges in meeting this need. The need for cost efficient sensing elements, limited in size, may commonly result to unexpected failures such as node crashing or network disconnection as experience of past real deployments has shown [Opp+14]. Furthermore, the nature of wireless communication (i.e., link instability and asymmetry [Pap+13]) and the largely unpredictable channel conditions (e.g., impact of weather on communications [Boa+10b]) over the deployment area may easily lead to poor design. Fault-tolerance has therefore gained much attention from researchers in the field of WSN, with reliability targeted at every layer of the communication stack (e.g., MAC, routing of sensed data towards the sink stations) [Liu+09a].

Our vision is to design a distributed rate adaptive scheme, for fault and delay tolerant WSN, and examine an array of options for improving the data packet collection over a multi-hop network by focusing at each node locally in a decentralized fashion. We will work on a queue management algorithm that adjusts the CBR to reduce the probability of having packet drops, once the queues are getting full. Based on predefined queue thresholds, our scheme will run when all communication paths to the sink are unavailable, independently the data collection approach (e.g., with or without data aggregation). Thus, the main goal of our adaptive mechanism is to avoid loss of data packet upon wireless link failures, while being independent from protocols that are embedded in the communication stack.

7.2.4 Age of Information in WSNs

Furthermore, our vision for future work would be to adapt our MAC solution into concept of Age of Information, in order to reduce the unnecessary traffic in the network and to avoid the potential congestions in the network.

In WSNs, sensing times may vary on each sensor node and so update packets can vary as well. In applications such as heart rate of a patient in the ER or the status of a F1 racecar tire, the timeliness with which a system presents its current status to a receiver is of critical importance. Assuming that every packet has a generation time stamp, the destination can calculate the age of the information it has for each of the sensor nodes. The Age of Information metric captures the freshness of each arriving status update. To quantify the information freshness, the concept of Age of Information has been recently introduced in [Kau+11]; capturing the time elapsed from the most recently generated status update till its reception.

Assuming the most recently update received at time t carries the timestamp of its generation $u(t)$, the source's status age, is the random process $\Delta(t) = t - u(t)$. Thus, keeping a low average $\Delta(t)$ can be a system requirement towards updating the receiver in a timely fashion [Kau+12].

Optimizing the timeliness of updates through the age of information metric is not equivalent to optimizing for throughput, nor minimizing delay. Consider that throughput can be maximized by making the source send updates as fast as possible. However, this may lead to the monitor receiving the information delayed because the sent messages get backlogged within the network, which we do not control. Conversely, the delay of status updates could be decreased by reducing the rate of updates, this may also lead to the receiver having unnecessarily outdated status information because of sparse updates.

Delay due to queuing at the transmitter can be happen in congested networks leading to increased age of information of a potentially critical sensor. To tackle such cases we investigate, through extensive test-bed experiments, the application of a simple queue management technique, in which we maintain a queue with only the latest status packet of each source, overwriting any previously queued update from that source. This simple technique drastically limits the need for buffering and can be applied in systems where the history of source status is not relevant. We show that this scheme results in significantly less transmissions compared to the standard M/M/1 queue model. Furthermore, the proposed technique reduces the per source age of information, especially in settings not using queue management with high status update generation rates.

Bibliography

[802a] "IEEE Standard for Local and metropolitan area networks–Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1: MAC sublayer". In: *IEEE Std 802.15.4e-2012 (Amendment to IEEE Std 802.15.4-2011)* (2012), pp. 1–225.

[802b] "IEEE std. 802.15.4, Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)". In: *Standard for Information Technology* (2011).

[AA09] H. Alwan and A. Agarwal. "A Survey on Fault Tolerant Routing Techniques in Wireless Sensor Networks". In: *Proceedings of the 3rd IARIA International Conference on Sensor Technologies and Applications (SENSORCOMM).* 2009, pp. 366–371.

[Ahn+06] Gahng-Seop Ahn, Se Gi Hong, Emiliano Miluzzo, Andrew T. Campbell, and Francesca Cuomo. "Funneling-MAC: A Localized, Sink-oriented MAC for Boosting Fidelity in Sensor Networks". In: *Proceedings of the 4th ACM Conference on Embedded Networked Sensor Systems (Sensys).* 2006, pp. 293–306.

[Aky+02a] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. "A Survey on Sensor Networks". In: *Communications Magazine, IEEE* 40.8 (2002), pp. 102–114.

[Aky+02b] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. "Wireless Sensor Networks: A Survey". In: *Elsevier Computer Networks* 38.4 (2002), pp. 393–422.

[Ana+09] G. Anastasi, M. Conti, M. Di Francesco, and A. Passarella. "Energy conservation in wireless sensor networks: A survey". In: *Elsevier Ad Hoc Networks Journal* 7.3 (2009), pp. 537–568.

[Anw+10] Markus Anwander, Gerald Wagenknecht, Torsten Braun, and Kirsten Dolfus. "BEAM: A Burst-aware Energy-efficient Adaptive MAC protocol for Wireless Sensor Networks". In: *Networked Sensing Systems (INSS), 2010 Seventh International Conference on.* 2010, pp. 195 –202.

[AP14] N. Accettura and G. Piro. "Optimal and Secure Protocols in the IETF 6TiSCH Communication Stack". In: *ISIE.* IEEE, 2014.

[Asc+10] Nils Aschenbruck, Raphael Ernst, Elmar Gerhards-Padilla, and Matthias Schwamborn. "BonnMotion: A Mobility Scenario Generation and Analysis Tool". In: *Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques (SIMUTools).* 2010.

[Ba+14] Papa Dame Ba, Ibrahima Niang, and Bamba Gueye. "An Optimized and Power Savings Protocol for Mobility Energy-aware in Wireless Sensor Networks". In: *Telecommunications Systems* 55.2 (2014), pp. 271–280.

[Bac+10] A. Bachir, M. Dohler, T. Watteyne, and K.K. Leung. "MAC Essentials for Wireless Sensor Networks". In: *Communications Surveys Tutorials, IEEE* 12.2 (2010), pp. 222–248.

[Bar+08a] Guillermo Barrenetxea, Francois Ingelrest, Gunnar Schaefer, Martin Vetterli, Olivier Couach, and Marc Parlange. "SensorScope: Out-of-the-Box Environmental Monitoring". In: *Proceedings of the International Conference on Information Processing in Sensor Networks (IPSN).* 2008, pp. 332–343.

[Bar+08b] Guillermo Barrenetxea, François Ingelrest, Gunnar Schaefer, and Martin Vetterli. "The Hitchhiker's Guide to Successful Wireless Sensor Network Deployments". In: *Proceedings of the 6th ACM conference on Embedded Networked Sensor Systems (Sensys).* 2008, pp. 43–56.

[Bea+11] Julien Beaudaux, Antoine Gallais, Julien Montavont, and Thomas Noel. "LIFT: Layer Independent Fault Tolerance Mechanism for Wireless Sensor Networks". In: *Proc. IEEE VTC.* 2011, pp. 1–5.

[Ber+07] Luis Bernardo, Rodolfo Oliveira, Miguel Pereira, Mário Macedo, Paulo Pinto, Nova De Lisboa, and Nova De Lisboa. "A Wireless Sensor MAC Protocol for bursty data traffic". In: *Proceedings of the 18th IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC).* 2007, pp. 1–5.

[BM04] S. Biswas and R. Morris. "Opportunistic Routing in Multi-hop Wireless Networks". In: *ACM SIGCOMM Computer Communication Review* 34.1 (2004), pp. 69 –74.

[BM05] Sanjit Biswas and Robert Morris. "ExOR: Opportunistic Multi-hop Routing for Wireless Networks". In: *Proc. ACM SIGCOMM.* 2005, pp. 133–144.

[Boa+10a] C.A. Boano, N. Tsiftes, T. Voigt, J. Brown, and U. Roedig. "The Impact of Temperature on Outdoor Industrial Sensornet Applications". In: *IEEE Transactions on Industrial Informatics* 6.3 (2010), pp. 451–459.

[Boa+10b] Carlo Alberto Boano, James Brown, Zhitao He, Utz Roedig, and Thiemo Voigt. "Low-Power Radio Communication in Industrial Outdoor Deployments: The Impact of Weather Conditions and ATEX-Compliance". In: *Sensor Applications, Experimentation, and Logistics* 29 (2010), pp. 159–176.

[BOY10] Jalel Ben-Othman and Bashir Yahya. "Energy Efficient and QoS Based Routing Protocol for Wireless Sensor Networks". In: *Journal of Parallel and Distributed Computing* 70.8 (2010), pp. 849–857.

[Bue+06] Michael Buettner, Gary V. Yee, Eric Anderson, and Richard Han. "X-MAC: A Short Preamble MAC Protocol for Duty-Cycled Wireless Sensor Networks". In: *Proceedings of the 4th ACM Conference on Embedded Networked Sensor Systems (Sensys).* 2006, pp. 307–320.

[Can+11] C. Cano, B. Bellalta, A. Sfairopoulou, and M. Oliver. "Low energy operation in WSNs: A survey of preamble sampling MAC protocols". In: *Computer Networks* 55.15 (2011), pp. 3351–3363.

[Cha+07] Szymon Chachulski, Michael Jennings, Sachin Katti, and Dina Katabi. "Trading Structure for Randomness in Wireless Opportunistic Routing". In: *Proc. ACM SIGCOMM.* 2007, pp. 169–180.

[Che+11] G. Chelius, C. Braillon, M. Pasquier, N. Horvais, R.P. Gibollet, B. Espiau, and C.A Coste. "A Wearable Sensor Network for Gait Analysis: A Six-Day Experiment of Running Through the Desert". In: *IEEE/ASME Transactions on Mechatronics* 16.5 (2011), pp. 878–883.

[Chi+07] C. F. Chiasserini, R. Gaeta, M. Garetto, M. Gribaudo, D. Manini, and M. Sereno. "Fluid Models for Large-scale Wireless Sensor Networks". In: *Elsevier Performance Evaluation* 64.7-8 (2007), pp. 715–736.

[Chi+10a] O. Chipara, C. Lu, T. Bailey, and G.C. Roman. "Reliable Clinical Monitoring using Wireless Sensor Networks: Experiences in a Step-down Hospital Unit". In: *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems (Sensys).* 2010, pp. 155–168.

[Chi+10b] O. Chipara, C. Lu, T. C. Bailey, and G.-C. Roman. "Reliable Clinical Monitoring using Wireless Sensor Networks: Experiences in a Step-down Hospital Unit". In: *SenSys*. ACM, 2010.

[Cis] *The Internet of Things Infographic*. URL: `http://blogs.cisco.com/news/the-internet-of-things-infographic/`.

[Cor+12] G. Corbellini, E.C. Strinati, and A. Duda. "LA-MAC: Low-latency asynchronous MAC for wireless sensor networks". In: *Personal Indoor and Mobile Radio Communications (PIMRC), 2012 IEEE 23rd International Symposium on*. 2012, pp. 380–386.

[Dan+15] C. Dandelski, B.-L. Wenning, D. Perez, D. Pesch, and J.-P. Linnartz. "Scalability of dense wireless lighting control networks". In: *IEEE Communications Magazine* 53.1 (2015), pp. 157–165.

[DD12] Qian Dong and Waltenegus Dargie. "Analysis of the Cost of Handover in a Mobile Wireless Sensor Network". In: *Proceedings of the 10th International Conference on Wired/Wireless Internet Communication (WWIC)*. 2012, pp. 315–322.

[DD13] Qian Dong and W. Dargie. "A Survey on Mobility and Mobility-Aware MAC Protocols in Wireless Sensor Networks". In: *IEEE Communications Surveys Tutorials* 15.1 (2013), pp. 88–100.

[Dil+11] Reza S. Dilmaghani, Hossein Bobarshad, M. Ghavami, Sabrieh Choobkar, and Charles Wolfe. "Wireless Sensor Networks for Monitoring Physiological Signals of Multiple Patients". In: *IEEE Transactions on Biomedical Circuits and Systems* 5.4 (2011), pp. 347–356.

[DL03] T. van Dam and K. Langendoen. "An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks". In: *SenSys*. ACM, 2003.

[Dun+04] Adam Dunkels, B. Gronvall, and Tiemo Voigt. "Contiki - a Lightweight and Flexible Operating System for Tiny Networked Sensors". In: *Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks (LCN)*. 2004, pp. 455–462.

[Dun+07] Adam Dunkels, Fredrik Osterlind, Nicolas Tsiftes, and Zhitao He. "Software-based On-line Energy Estimation for Sensor Nodes". In: *Proceedings of the 4th ACN Workshop on Embedded Networked Sensors (EmNets)*. 2007, pp. 28–32.

[Dun11] Adam Dunkels. *The ContikiMAC Radio Duty Cycling Protocol*. Technical Report T2011:13. Swedish Institute of Computer Science, 2011.

[Duq+11] Simon Duquennoy, Fredrik Österlind, and Adam Dunkels. "Lossy Links, Low Power, High Throughput". In: *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems (Sensys)*. 2011, pp. 12–25.

[Duq+13] Simon Duquennoy, Olaf Landsiedel, and Thiemo Voigt. "Let the Tree Bloom: Scalable Opportunistic Routing with ORPL". In: *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems (Sensys)*. 2013, pp. 1–14.

[DW14a] Waltenegus Dargie and Jianjun Wen. "A MAC Protocol for Medical Applications (Poster Abstract)". In: *Proceedings of the 13th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. 2014, pp. 287–288.

[DW14b] Waltenegus Dargie and Jianjun Wen. "A Seamless Handover for WSN Using LMS Filter". In: *Proceedings of the 39th IEEE Conference on Local Computer Networks (LCN)*. 2014, pp. 287–288. URL: `http://www.rn.inf.tu-dresden.de/uploads/Publikationen/LCN2014V2.pdf`.

[Dyo+10] Vladimir Dyo et al. "Evolution and Sustainability of a Wildlife Monitoring Sensor Network". In: *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems (Sensys)*. 2010, pp. 127–140.

[EH+03] A. El-Hoiydi, J.-D. Decotignie, C. Enz, and E. Le Roux. “WiseMAC, an Ultra Low Power MAC Protocol for the wiseNET Wireless Sensor Networks”. In: *Proceedings of the 1st ACM Conference on Embedded Networked Sensor Systems (Sensys)*. 2003, pp. 302–303.

[EH02] A. El-Hoiydi. “Aloha with preamble sampling for sporadic traffic in ad hoc wireless sensor networks”. In: *Proceedings of the IEEE International Conference on Communications (ICC)*. Vol. 5. 2002, pp. 3418–3423.

[EL+05] E. Egea-López, J. Vales-Alonso, A. S. Martínez-Sala, P. Pavón-Mariño, and J. García-Haro. “Elder Care Based on Cognitive Sensor Network”. In: *Summer Simulation Multiconference - SPECTS*. 2005.

[Eri+09] Joakim Eriksson, Fredrik Österlind, Niclas Finne, Adam Dunkels, Nicolas Tsiftes, and Thiemo Voigt. “Accurate Network-Scale Power Profiling for Sensor Network Simulators”. In: *Proceedings of the 6th European Conference on Wireless Sensor Networks (EWSN)*. 2009, pp. 312–326.

[EV06] S. C. Ergen and P. Varaiya. “PEDAMACS: Power Efficient and Delay Aware Medium Access Protocol for Sensor Networks”. In: *Transactions on Mobile Computing, IEEE* 5.7 (2006), pp. 920–930.

[Fre+13] Edison Pignaton de Freitas, Tales Heimfarth, Alexey Vinel, Flávio Rech Wagner, Carlos Eduardo Pereira, and Tony Larsson. “Cooperation among Wirelessly Connected Static and Mobile Sensor Nodes for Surveillance Applications”. In: *Sensors* 13.10 (2013), pp. 12903–12928.

[Glu+11] A. Gluhak, S. Krco, M. Nati, D. Pfisterer, N. Mitton, and T. Razafindralambo. “A Survey on Facilities for Experimental Internet of Things Research”. In: *IEEE Communications Magazine* 49.11 (2011), pp. 58–67.

[Hac+14] G. Hackmann, Weijun Guo, Guirong Yan, Zhuoxiong Sun, Chenyang Lu, and S. Dyke. “Cyber-Physical Codesign of Distributed Structural Health Monitoring with Wireless Sensor Networks”. In: *IEEE Transactions on Parallel and Distributed Systems* 25.1 (2014), pp. 63–72.

[HB10] Philipp Hurni and Torsten Braun. “MaxMAC: a maximally traffic-adaptive MAC protocol for wireless sensor networks”. In: *Proceedings of the 7th European conference on Wireless Sensor Networks*. EWSN’10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 289–305.

[HC02] J.L. Hill and D.E. Culler. “Mica: a wireless platform for deeply embedded networks”. In: *IEEE Micro* 22.6 (2002), pp. 12–24.

[Hir+13] D. Hiranandani, K. Obraczka, and J.J. Garcia-Luna-Aceves. “MANET protocol simulations considered harmful: the case for benchmarking”. In: *IEEE Wireless Communications* 20.4 (2013), pp. 82–90.

[HT11] J. Hui and P. Thubert. *Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks*. RFC 6282. IETF, 2011.

[HV08] Karel Heurtefeux and Fabrice Valois. “Distributed Qualitative Localization for Wireless Sensor Networks”. In: *Proc. ADHOC-NOW*. Vol. 5198. 2008, pp. 218–229.

[HV09] K. Heurtefeux and F. Valois. “Distributed Localization Protocol for Routing in a Noisy Wireless Sensor Network”. In: *Proceedings of the 5th International Conference on Mobile Ad-hoc and Sensor Networks (MSN)*. 2009, pp. 223–230.

[Ieea] In: *IEEE Std 802.15.4, Wireless MAC and PHY Specifications for Low-Rate Wireless Personal Area Networks WPANs* (2006).

[Ieeb] “Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, ANSI/IEEE Std 802.11.” In: *IEEE Standard 802.11* ().

[Iga+15] Mónica Aguilar Igartua, Francesca Cuomo, and Isabelle Guérin Lassous. “Special issue on Modeling and Performance Evaluation of Wireless Ad-Hoc Networks”. In: *Elsevier Ad Hoc Networks* 24, Part B (2015), pp. 1–2.

[Iov14] Oana Iova. *Standards Optimization and Network Lifetime Maximization for Wireless Sensor Networks in the Internet of Things.* Ph.D. Thesis. 2014.

[Jon+10] V. Jones, V. Gay, and P. Leijdekkers. “Body Sensor Networks for Mobile Health Monitoring: Experience in Europe and Australia”. In: *Proceedings of the 4th International Conference on Digital Society (ICDS).* 2010, pp. 204–209.

[Jur+07] Raja Jurdak, Pierre Baldi, and Cristina Videira Lopes. “Adaptive Low Power Listening for Wireless Sensor Networks”. In: *IEEE Transactions on Mobile Computing* 6.8 (Aug. 2007), pp. 988–1004.

[Kau+11] S. Kaul, M. Gruteser, V. Rai, and J. Kenney. “Minimizing age of information in vehicular networks”. In: *Proceedings of the 8th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON).* 2011, pp. 350–358.

[Kau+12] S. Kaul, R. Yates, and M. Gruteser. “Real-time status: How often should one update?” In: *Proceedings of the 31st IEEE International Conference on Computer Communications(INFOCOM).* 2012, pp. 2731–2735.

[Kdo+12] H. Kdouh, H. Farhat, G. Zaharia, C. Brousseau, G. Grunfelder, and G.E. Zein. “Performance analysis of a hierarchical shipboard Wireless Sensor Network”. In: *Proceedings of the 23rd IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC).* 2012, pp. 765–770.

[KL] Albert Ko and Henry Y. K. Lau. “Robot Assisted Emergency Search and Rescue System With a Wireless Sensor Network”. In: *International Journal of Advanced Science and Technology* (), pp. 69–78.

[KM07] Kurtis Kredo II and Prasant Mohapatra. “Medium Access Control in Wireless Sensor Networks”. In: *Computer Networks* 51.4 (2007), pp. 961–994.

[Kro+12] J. Kropf, L. Roedl, and A Hochgatterer. “A modular and flexible system for activity recognition and smart home control based on nonobtrusive sensors”. In: *Proceedings of the 6th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth).* 2012, pp. 245–251.

[Kun+11] Romain Kuntz, Antoine Gallais, and Thomas Noël. “Auto-adaptive MAC for Energy-Efficient Burst Transmissions in Wireless Sensor Networks”. In: *Proc. IEEE WCNC.* 2011, pp. 233–238.

[Kun+13] Romain Kuntz, Julien Montavont, and Thomas Noël. “Improving the medium access in highly mobile Wireless Sensor Networks”. In: *Telecommunication Systems* 52.4 (2013), pp. 2437–2458.

[Kur+05] Stuart Kurkowski, Tracy Camp, and Michael Colagrosso. “MANET Simulation Studies: The Incredibles”. In: *Mobile Computing and Communications Review* 9.4 (2005), pp. 50–61.

[Lan+06] Koen Langendoen, Aline Baggio, and Otto Visser. “Murphy Loves Potatoes: Experiences from a Pilot Sensor Network Deployment in Precision Agriculture”. In: *Proceedings of the 20th IEEE International Symposium on Parallel and Distributed Processing (IPDPS).* 2006, pp. 174–174.

[Lan+12] Olaf Landsiedel, Euhanna Ghadimi, Simon Duquennoy, and Mikael Johansson. “Low Power, Low Delay: Opportunistic Routing meets Duty Cycling”. In: *Proceedings of the 11th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN).* 2012, pp. 185–196.

[Lan07] Koen Langendoen. “Medium Access Control in Wireless Sensor Networks”. In: *Volume II: Practice and Standards, Nova Science Publishers* (2007).

[Lev+04] Philip Levis et al. “TinyOS: An operating system for sensor networks”. In: *in Ambient Intelligence*. Springer Verlag, 2004.

[Liu+09a] Hai Liu, Amiya Nayak, and Ivan Stojmenovic. “Fault Tolerant Algorithms/Protocols in Wireless Sensor Networks”. In: *Guide to Wireless Sensor Networks*. Computer Communications and Networks. Springer London, 2009, pp. 261–291.

[Liu+09b] Haitao Liu, Baoxian Zhang, Hussein T. Mouftah, Xiaojun Shen, and Jian Ma. “Opportunistic Routing for Wireless Ad Hoc and Sensor Networks: Present and Future Directions”. In: *IEEE Communications Magazine* 47.12 (2009), pp. 103–109.

[LL07] Mo Li and Yunhao Liu. “Underground Structure Monitoring with Wireless Sensor Networks”. In: *Proceedings of the 6th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. 2007, pp. 69–78.

[Loh+13] Stefan Lohs, Reinhardt Karnapke, and Jorg Nolte. “Link Stability in a Wireless Sensor Network - an Experimental Study”. In: *4th International ICST Conference on Sensor Systems and Software (S-CUBE)* (2013), pp. 146–161.

[ME+08] Razvan Musaloiu-E., Chieh-Jan Mike Liang, and Andreas Terzis. “Koala: Ultra-Low Power Data Retrieval in Wireless Sensor Networks”. In: *Proceedings of the 7th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. 2008, pp. 421–432.

[Mei+10] Andreas Meier, Matthias Woehrle, Marco Zimmerling, and Lothar Thiele. “ZeroCal: automatic MAC protocol calibration”. In: *Proceedings of the 6th IEEE international conference on Distributed Computing in Sensor Systems*. DCOSS’10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 31–44.

[MH10a] Christophe J. Merlin and Wendi B. Heinzelman. “Schedule Adaptation of Low-Power-Listening Protocols for Wireless Sensor Networks”. In: *IEEE Transactions on Mobile Computing* 9.5 (2010), pp. 672–685.

[MH10b] C.J. Merlin and W.B. Heinzelman. “Duty Cycle Control for Low-Power-Listening MAC Protocols”. In: *IEEE Transactions on Mobile Computing* 9.11 (2010), pp. 1508 –1521.

[ML08] D. Moss and P. Levis. “BoX-MACs: Exploiting Physical and Link Layer Boundaries in Low-Power Networking”. In: *Technical Report SING-08-00, Stanford University* (2008).

[Mok+14] L. Mokdad, J. Ben-Othman, B. Yahya, and S. Niagne. “Performance Evaluation Tools for QoS MAC Protocol for Wireless Sensor Networks”. In: *Ad Hoc Networks* 12 (2014), pp. 86–99.

[Mon+07] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler. *Transmission of IPv6 Packets over IEEE 802.15.4 Networks*. RFC 4944. IETF, 2007.

[MQ14] Mathieu Michel and Bruno Quoitin. “ContikiMAC vs X-MAC performance analysis”. In: *University of Mons, Technical Report* (2014).

[Nam+12] Nguyen Van Nam, Isabelle Guerin Lassous, Victor Moraru, and Tahiry Razafindralambo. “Characterisation and Application of Idle Period Durations in IEEE 802.11 DCF-based Multihop Wireless Networks”. In: *Proceedings of the 15th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM)*. 2012, pp. 277–284.

[Opp+14] Felix Jonathan Oppermann, Carlo Alberto Boano, and Kay Romer. “A Decade of Wireless Sensing Applications: Survey and Taxonomy”. In: *The Art of Wireless Sensor Networks*. Signals and Communication Technology. 2014, pp. 11–50.

[Ost+06] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt. “Cross-Level Sensor Network Simulation with COOJA”. In: *Proceedings of the 31st Annual IEEE International Conference on Local Computer Networks (LCN).* 2006.

[Pak+06] J. Pak, J. Son, and K. Han. “A MAC Protocol Using Separate Wakeup Slots for Sensor Network”. In: *Computational Science and Its Applications, Springer.* Vol. 3981. 2006, pp. 1159–1168.

[Pal+13] M.R. Palattella, N. Accettura, X. Vilajosana, T. Watteyne, L.A. Grieco, G. Boggia, and M. Dohler. “Standardized Protocol Stack for the Internet of (Important) Things”. In: *IEEE Communications Surveys Tutorials* 15.3 (2013), pp. 1389–1406.

[Pap+13] Georgios Z. Papadopoulos, Julien Beaudaux, Antoine Gallais, Thomas Noel, and Guillaume Schreiner. “Adding value to WSN simulation using the IoT-LAB experimental platform”. In: *Proceedings of the 9th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob).* 2013, pp. 485–490.

[Pap+14a] Georgios Z. Papadopoulos, Antoine Gallais, Thomas Noel, Vasileios Kotsiou, and Periklis Chatzimisios. “Enhancing ContikiMAC for Bursty Traffic in Mobile Sensor Networks”. In: *Proceedings of the IEEE Sensors.* 2014.

[Pap+14b] Georgios Z. Papadopoulos, Julien Beaudaux, Antoine Gallais, and Thomas Noel. “T-AAD: Lightweight Traffic Auto-ADaptations for Low-power MAC Protocols”. In: *Proceedings of the 13th IEEE IFIP Annual Mediterranean Ad Hoc Networking Workshop (MED-HOC-NET).* 2014, pp. 79–86.

[Pap+14c] Georgios Z. Papadopoulos, Julien Beaudaux, Antoine Gallais, Periklis Chatzimisios, and Thomas Noel. “Toward a Packet Duplication Control for Opportunistic Routing in WSNs”. In: *Proceedings of the IEEE Global Communications Conference (GLOBECOM).* 2014, pp. 94–99.

[Pap+15a] Georgios Z. Papadopoulos, Antoine Gallais, Guillaume Schreiner, and Thomas Noel. “Demo Abstract: Live Adaptations of Low-power MAC Protocols”. In: *Proceedings of the 21st International Conference on Mobile Computing and Networking (MobiCom).* 2015.

[Pap+15b] Georgios Z. Papadopoulos, Vasileios Kotsiou, Antoine Gallais, Periklis Chatzimisios, and Thomas Noel. “Optimizing the Handover Delay in Mobile WSNs”. In: *Proceedings of the 2nd IEEE World Forum on Internet of Things (WF-IoT).* 2015.

[Pap+15c] Georgios Z. Papadopoulos, Kosmas Kritsis, Antoine Gallais, Periklis Chatzimisios, and Thomas Noel. “Performance Evaluation Methods in Ad-Hoc and Wireless Sensor Networks: A Literature Study”. In: *IEEE Communications Magazine* (2015).

[Pap+15d] Georgios Z. Papadopoulos, Vasileios Kotsiou, Antoine Gallais, Periklis Chatzimisios, and Thomas Noel. “Wireless Medium Access Control under Mobility and Bursty Traffic Assumptions in WSNs”. In: *Springer Mobile Networks and Applications (MONET)* (2015), pp. 1–12.

[Pol+04] Joseph Polastre, Jason Hill, and David Culler. “Versatile Low Power Media Access for Wireless Sensor Networks”. In: *Proceedings of the 2nd ACM Conference on Embedded Networked Sensor Systems (Sensys).* 2004, pp. 95–107.

[Pol+05] J. Polastre, R. Szewczyk, and D. Culler. “Telos: enabling ultra-low power wireless research”. In: *Proceedings of the 4th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN).* 2005, pp. 364–369.

[Raj+05] V. Rajendran, J.J. Garcia-Luna-Aveces, and K. Obraczka. “Energy-efficient, application-aware medium access for sensor networks”. In: *Proceedings of the 2nd IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS).* 2005, 630–637.

[Raj+06] V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves. “Energy-efficient, Collision-free Medium Access Control for Wireless Sensor Network”. In: *Wireless Networks, ACM* 12.1 (2006), pp. 63–78.

[Rhe+08] Injong Rhee, A. Warrier, M. Aia, Jeongki Min, and M.L. Sichitiu. “Z-MAC: A Hybrid MAC for Wireless Sensor Networks”. In: *IEEE/ACM Transactions on Networking* 16.3 (2008), pp. 511–524.

[RI04] V. Raisinghani and S. Iyer. “Cross-layer Design Optimizations in Wireless Protocol Stacks”. In: *Computer Communications* 27.8 (2004), pp. 720 –724.

[RM04] K. Romer and F. Mattern. “The Design Space of Wireless Sensor Networks”. In: *IEEE Wireless Communications* 11.6 (2004), pp. 54–61.

[Rot+11] Damien Roth, Julien Montavont, and Thomas Noel. “MOBINET: Mobility management across different wireless sensor networks”. In: *IEEE Wireless Communications and Networking Conference (WCNC).* 2011.

[Roz+11] Clement Burin des Roziers, Guillaume Chelius, Tony Ducrocq, Eric Fleury, Antoine Fraboulet, Antoine Gallais, Nathalie Mitton, Thomas Noel, and Julien Vandaele. “Using SensLAB as a First Class Scientific Tool for Large Scale Wireless Sensor Network Experiments”. In: *Proceedings of the 10th IFIP International Conference on Networking.* 2011, pp. 147–159.

[Sfp] 2014. URL: `http://www.sfpark.org`.

[Sha+06] Kewei Sha, Weisong Shi, and O. Watkins. “Using Wireless Sensor Networks for Fire Rescue Applications: Requirements and Challenges”. In: *Proceedings of the 6th IEEE International Conference on Electro/Information Technology.* 2006, pp. 239–244.

[Sim+04] G. Simon, M. Maróti, A. Lédeczi, G. Balogh, B. Kusy, A. N., G. Pap, J. Sallai, and K. Frampton. “Sensor Network-Based Countersniper System”. In: *SenSys.* ACM, 2004.

[Sma] *Smart Gains.* URL: `http://www.smartgrains.com`.

[Soh+00] K. Sohrabi, J. Gao, V. Ailawadhi, and G. J. Pottie. “Protocols for Self-organization of a Wireless Sensor Network”. In: *Personal Communications, IEEE* 7.5 (2000), pp. 16–27.

[Spa+12] Petros Spachos, Periklis Chatzimisios, and Dimitrios Hatzinakos. “Energy Aware Opportunistic Routing in Wireless Sensor Networks”. In: *Proc. IEEE GLOBECOM.* 2012, pp. 405–409.

[Sto08] I. Stojmenovic. “Simulations in Wireless Sensor and Ad Hoc Networks: Matching and Advancing Models, Metrics, and Solutions”. In: *IEEE Communications Magazine* 46.12 (2008), pp. 102–107.

[Sua+08] Pablo Suarez, Carl-Gustav Renmarker, Adam Dunkels, and Thiemo Voigt. “Increasing ZigBee Network Lifetime with X-MAC”. In: *Proceedings of the ACM Workshop on Real-world Wireless Sensor Networks (REALWSN).* 2008, pp. 26–30.

[Sun+08] Yanjun Sun, Omer Gurewitz, and David B. Johnson. “RI-MAC: A Receiver-Initiated Asynchronous Duty Cycle MAC Protocol for Dynamic Traffic Loads in Wireless Sensor Networks”. In: *Proceedings of the 6th ACM conference on Embedded Networked Sensor Systems (Sensys).* 2008, pp. 1–14.

[SY04] F. Sivrikaya and B. Yener. “Time Synchronization in Sensor Networks: a Survey”. In: *IEEE Network* 18.4 (2004), pp. 45–50.

[Tho+04] Bjørn Thorstensen, Tore Syversen, Trond-Are Bjørnvold, and Tron Walseth. “Electronic Shepherd - A Low-Cost, Low-Bandwidth, Wireless Network System”. In: *Proceedings of the 2nd ACM International Conference on Mobile Systems, Applications, and Services (MobiSys).* 2004, pp. 245–255.

[Tol+05] G. Tolle et al. “A Macroscope in the Redwoods”. In: *SenSys.* ACM, 2005.

[Ton+14] A.-S. Tonneau, N. Mitton, and J. Vandaele. “A Survey on (mobile) Wireless Sensor Network Experimentation Testbeds”. In: *Proceedings of the IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS).* 2014.

[Tri] *Trilliant Communications Platform.* URL: `http://trilliantinc.com/`.

[Ver+08] R. Verdone, D. Dardari, G. Mazzini, and A. Conti. “Wireless Sensor and Actuator Networks”. In: *Technologies, Analysis and Design.* Academic Press, 2008, pp. 1–11.

[Vuc+13] M. Vucinic, B. Tourancheau, and A Duda. “Performance comparison of the RPL and LOADng routing protocols in a Home Automation scenario”. In: *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC).* 2013, pp. 1974–1979.

[WA+08] G. Werner-Allen, S. Dawson-Haggerty, and M. Welsh. “Lance: Optimizing High-Resolution Signal Collection in Wireless Sensor Networks”. In: *SenSys.* ACM, 2008.

[WA06] Kai-Juan Wong and D. K. Arvind. “SpeckMAC: Low-power Decentralised MAC Protocols for Low Data Rate Transmissions in Specknets”. In: *Proceedings of the 2nd International Workshop on Multi-hop Ad Hoc Networks: from Theory to Reality.* 2006, pp. 71–78.

[Wan+12] You-Chiun Wang, Fang-Jing Wu, and Yu-Chee Tseng. “Mobility management algorithms and applications for mobile sensor networks”. In: *Wireless Communications and Mobile Computing* 12.1 (2012), pp. 7–21.

[Wan+14] Q. Wang, X. Vilajosana, and T. Watteyne. *6TiSCH Operation Sublayer (6top) draft-wang-6tisch-6top-sublayer-01.* Internet-Draft. IETF, 2014.

[Was+07] Kamil Wasilewski, Joel W. Branch, Mark Lisee, and Boleslaw K. Szymanski. *Self-healing routing: a study in efficiency and resiliency of data delivery in wireless sensor networks.* 2007.

[Wat+09] Thomas Watteyne, Dominique Barthel Kris Pister, Mischa Dohler, and Isabelle Auge-Blum. “Implementation of Gradient Routing in Wireless Sensor Networks”. In: *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM).* 2009, pp. 1–6.

[Wat+12] Thomas Watteyne, Xavier Vilajosana, Branko Kerkez, Fabien Chraim, Kevin Weekly, Qin Wang, Steven D. Glaser, and Kris Pister. “OpenWSN: a standards-based low-power wireless development environment”. In: *Transactions Emerging Telecommunications Technologies* 23.5 (2012), pp. 480–493.

[Win+12] T. Winter et al. *RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks.* RFC 6550. IETF, 2012.

[Ye+02] W. Ye, J. Heidemann, and D. Estrin. “An energy-efficient MAC protocol for wireless sensor networks”. In: *INFOCOM.* IEEE, 2002.

[Ye+06] Wei Ye, Fabio Silva, and John Heidemann. “Ultra-low Duty Cycle MAC with Scheduled Channel Polling”. In: *Proceedings of the 4th ACM Conference on Embedded Networked Sensor Systems (Sensys).* 2006, pp. 321–334.

[YH12] Ou Yang and Wendi B. Heinzelman. “Modeling and Performance Analysis for Duty-Cycled MAC Protocols with Applications to S-MAC and X-MAC”. In: *IEEE Transactions on Mobile Computing* 11.6 (2012), pp. 905–921.

[Yic+08] J. Yick, B. Mukherjee, and D. Ghosal. “Wireless sensor network survey”. In: *Computer Networks (Elsevier)* 52.12 (2008), pp. 2292–2330.

[ZD10] Tang Zhiyong and Waltenegus Dargie. “A Mobility-Aware Medium Access Control Protocol for Wireless Sensor Networks”. In: *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM) Workshops.* 2010, pp. 109–114.

[Zha+04] Pei Zhang, Christopher M. Sadler, Stephen A. Lyon, and Margaret Martonosi. “Hardware design experiences in ZebraNet”. In: *Proceedings of the 2nd ACM Conference on Embedded Networked Sensor Systems (Sensys)*. 2004, pp. 227–238.

[Zho+06] Gang Zhou, Chengdu Huang, Ting Yan, Tian He, J.A. Stankovic, and T.F. Abdelzaher. “MMSN: Multi-Frequency Media Access Control for Wireless Sensor Networks”. In: *Proceedings of the 25th IEEE International Conference on Computer Communications (INFOCOM)*. 2006, pp. 1–13.

[Zim+12] Marco Zimmerling, Federico Ferrari, Luca Mottola, Thiemo Voigt, and Lothar Thiele. “pTunes: Runtime Parameter Adaptation for Low-power MAC Protocols”. In: *Proceedings of the 11th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. 2012, pp. 173–184.

[ZR03] M. Zorzi and Ramesh R. Rao. “Geographic Random Forwarding (GeRaF) for Ad Hoc and Sensor Networks: Energy and Latency Performance”. In: *IEEE Transactions on Mobile Computing* 2.4 (2003), pp. 349–365.

List of Figures

List of Tables

List of Abbreviations

6TOP	6TSCH Operation Sublayer.
6TiSCH	IPv6 over the TSCH mode of IEEE 802.15.4e.
ADC	Analog-to-Digital Converter.
CBR	Constant Bit Rate.
CCA	Clear Channel Assessment.
CPU	Central Processing Unit.
CSMA	Carrier Sense Multiple Access.
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance.
DCF	Distributed Coordination Function.
DLL	Data Link Layer.
FDMA	Frequency Division Multiple Access.
GUI	Graphical User Interface.
IEEE	Institute of Electrical and Electronics Engineers.
IETF	Internet Engineering Task Force.
IoT	Internet of Thing.
IP	Internet Protocol.
LLC	Logical Link Control.
LLN	Low-power and Lossy Network.
LPL	Low-Power Listening.
LPM	Low-Power Mode.
LPP	Low-Power Probing.
M2M	Machine-to-Machine.
MAC	Medium Access Control.
MANET	Mobile Ad-Hoc Network.
OSI	Open Systems Interconnection.
PC	Personal Computer.
PCF	Point Coordination Function.
PRR	Packet Reception Ratio.

QoS	Quality of Service.
RPL	IPv6 Routing Protocol for Low-power and Lossy Networks.
TDMA	Time Division Multiple Access.
TSCH	Timeslotted Channel Hopping.
WSN	Wireless Sensor Network.