UNIVERSITÀ DI PISA
Scuola di Dottorato in Ingegneria "Leonardo da Vinci"

Corso di Dottorato di Ricerca in
Ingegneria dell'Informazione

Tesi di Dottorato di Ricerca

Models and Protocols for Resource Optimization in Wireless Mesh Networks

Antonio Carmelo Pinizzotto

Anno 2011
SSD ING-INF/05

UNIVERSITÀ DI PISA

Scuola di Dottorato in Ingegneria "Leonardo da Vinci"

Corso di Dottorato di Ricerca in
Ingegneria dell'Informazione

Tesi di Dottorato di Ricerca

Models and Protocols for Resource Optimization in Wireless Mesh Networks

Autore:

Antonio Carmelo Pinizzotto _____

Relatori:

Prof. Giuseppe Anastasi _____

Dott. Marco Conti _____

Anno 2011
SSD ING-INF/05

Contents

Chapter 1

Introduction

1.1 Problem Statement

The past decade has witnessed the widespread and ever growing diffusion of wireless technologies, such as WiFi, fueled by their cost lowering and their increasing performances. This have been generating renewed and growing interest in research and development in the Mobile (multihop) Ad Hoc Networks (MANETs), targeted to civilian applications. MANETs are collections of mobile nodes connected together over a wireless medium. These nodes can freely and dynamically self-organize into arbitrary and temporary ad hoc network topologies, allowing people and devices to seamlessly internetwork in areas with no preexisting communication infrastructure (e.g., disaster recovery and battlefield environments). To build a connected network each node is both end user and forwarder for other users' packets. The ad hoc networking concept is not new, having been around in various forms for over 30 years, mainly for tactical military applications. The recent renewed interest in MANETs is also due to the standardization efforts of the Internet Engineering Task Force (IETF) MANET Working Group which is standardizing four routing protocols, and to the ubiquitous 802.11 wireless cards (an enabling technology for civilian MANETs). However, this type of network does not yet have an impact on our way of using wireless networks. Users seldom operate 802.11 in ad hoc mode and, except in laboratory testbeds, never use multihop ad hoc networks. This has opened a debate in the scientific community on why, after almost a decade

of research into ad hoc networking, MANET technology has not yet affected our way of using wireless networks. A common answer is emerging: most of the ongoing research on mobile ad hoc networks is driven by either Department of Defense (DoD) requirements (large-scale military applications with thousands of ad hoc nodes) or specialized civilian applications (disaster recovery, planetary exploration, etc). DoD generated a research agenda and requirements that are far from real users requirements. Indeed, military and specialized civilian applications require lack of infrastructure and instant deployment. They are tailored to very specialized missions, and their cost is typically not a main issue. On the other hand, from the users standpoint, scenarios consisting of a limited number of people wanting to form an ad hoc network for sharing some information or access to the Internet are much more interesting. In this case, users are looking for multipurpose networking platforms in which cost is an issue and Internet access is a must. To turn MANETs into a commodity some changes to the original MANET definition would seem to be required. By relaxing one of the main constraints of MANETs, "*the network is made of users devices only and no infrastructure exists*", we move to a more pragmatic "*opportunistic ad hoc networking*" in which multihop ad hoc networks are not isolate self-configured networks, but rather emerge as a flexible and low-cost extension of wired infrastructure networks, coexisting with them. Indeed, a new class of networks is emerging from this view: Wireless Mesh Networks (WMNs) [KSK04]. WMNs are built on a mix of fixed and mobile nodes interconnected via wireless links to form a multihop ad hoc network. As in MANETs, users devices are an active part of the mesh. They dynamically join the network, acting as both user terminals and routers for other devices, consequently further extending network coverage. Mesh networks thus inherit many results from MANET research but have civilian applications as the main target. Furthermore, while the MANET development approach was mainly simulation-based, from the beginning mesh networks have been associated with real testbeds. By designing/implementing "good enough" solutions it has been possible to verify the suitability of this technology for civilian applications and stimulate users interest in adopting it. Even though wireless mesh networks are quite recent, they have already shown great potential in the wireless market, even if the full potential of mesh networking in supporting new applications is still not fully unleashed. Indeed, we can subdivide wireless mesh networks into two main classes: off-the-shelf and

proprietary solutions. An example of the first class are so-called community networks built (mainly) on 802.11 technology and aimed at providing Internet access to a community of users that can share the same Internet access link. Some examples of this are Seattle Wireless, Champaign-Urbana Community Wireless Network (CUWiN), San Francisco BAWUG, and the Roofnet system at MIT (MIT Roofnet). On the other hand, several companies are now selling interesting solutions that exploit the mesh network potential for indoor and/or outdoor applications (e.g., MeshNetworks, Tropos Networks, Radiant Networks, Firetide, BelAir Networks, Strix Systems). For example, indoor mesh networks can be set up by wireless interconnected access points that, by exploiting routing algorithms developed for MANETs, can create extended WLANs without a wired infrastructure. Outside buildings, mesh networks can be used to provide wireless access across wide geographic areas by minimizing the number of wired ingress/egress points toward the Internet. Outdoor networks might be used, for example, by municipalities to extend their wired networks wirelessly.

A wireless mesh network is a fully wireless network that employs multihop communications to forward traffic en route to and from wired Internet entry points. Different from flat ad hoc networks, a mesh network introduces a hierarchy in the network architecture with the implementation of dedicated nodes (called wireless routers) communicating among each other and providing wireless transport services to data traveling from users to either other users or access points (access points are special wireless routers with a high-bandwidth wired connection to the Internet backbone). The network of wireless routers forms a wireless backbone (tightly integrated into the mesh network), which provides multihop connectivity between nomadic users and wired gateways. The meshing among wireless routers and access points creates a wireless backhaul communication system, which provides each mobile user with a low-cost, high-bandwidth, and seamless multihop interconnection service with a limited number of Internet entry points and with other wireless mobile users. Roughly and generally speaking, backhaul is used to indicate the service of forwarding traffic from the originator node to an access point from which it can be distributed over an external network. Specifically in the mesh case, the traffic is originated in the users devices, traverses the wireless backbone, and is distributed over the Internet network. To summarize, Figure 1.1 illustrates the mesh net-

work architecture, highlighting the different components and system layers. A WMN is dynamically self-organized and self-configured, with the nodes in the network automatically establishing and maintaining mesh connectivity among themselves (creating, in effect, an ad hoc network). This feature brings many advantages to WMNs such as low up-front cost, easy network maintenance, robustness, and reliable service coverage. More specifically, 802.11-based wireless mesh networks are emerging as a key technology to provide cost-effective ubiquitous access to the Internet [KSK04].

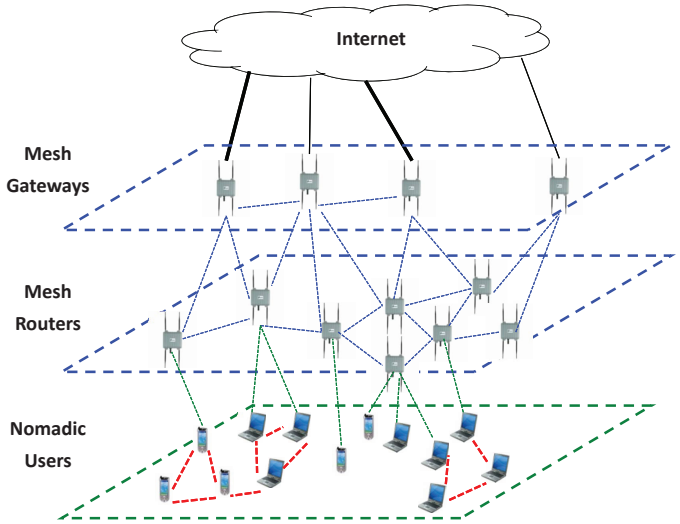


Figure 1.1: A three-tier architecture for wireless mesh networks.

Normally, in mesh networks only a subset of routers, referred to as *gateways*, has a high-speed Internet connection, while Internet access is shared among all the other mesh nodes by exploiting the ad hoc routing capabilities of the mesh routers [BCG05, AWW05]. However, this vision is rapidly changing. Real-world mesh networks have been recently deployed, which are used to share a potentially large number of low-speed Internet connections (i.e., DSL fixed lines) available at the customers' premises. Examples of such networks are

Meraki-based deployments in urban areas [Mer], or the Ozone's network in Paris, which is composed of 400 mesh routers, most of them using standard DSL links as Internet backhaul, while only ten gateways are provided with an ISP-owned fiber link [Ozo]. In a broader sense, wireless mesh networks are evolving into a *converged infrastructure* used to share the Internet connectivity of sparsely deployed fixed lines with *heterogeneous capacity*, ranging from ISP-owned broadband links to subscriber-owned low-speed connections [SACB08] (Figure 1.2).

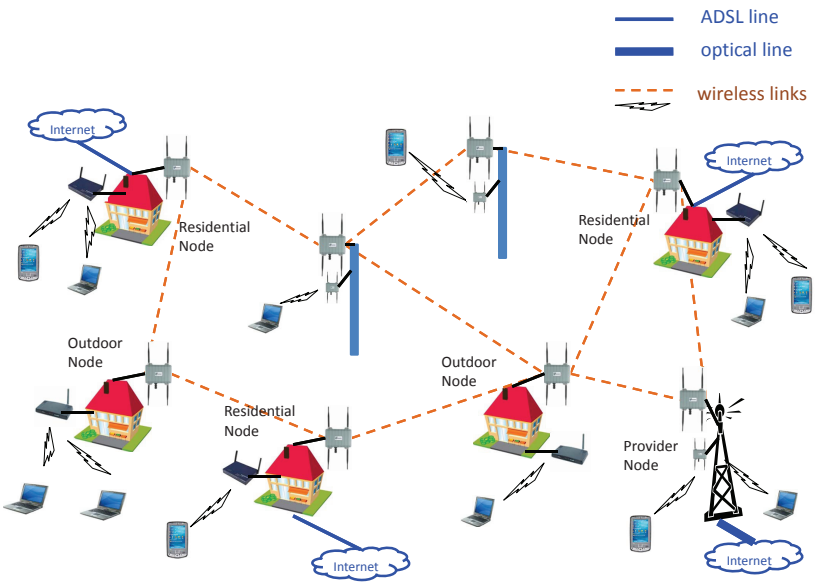


Figure 1.2: Internet access sharing realized with a heterogeneous wireless mesh network.

Being mesh networks primarily used for Internet access, both traffic routing and Internet gateway selection play a crucial role in determining the overall network performance, and in ensuring the optimal utilization of the mesh infrastructure [LLT03, ZWR08]. For instance, if too many mesh nodes select the same gateway as egress point to the Internet, congestion may increase excessively on the wireless channel, or the Internet connection of the gateway

can get overloaded. This is especially important in the heterogeneous mesh networks we consider in the work of this thesis, because low-speed Internet gateways may easily become a *bottleneck*, limiting the achievable capacity of the entire network. In addition a load-unaware gateway selection can lead to an unbalanced utilization of the gateways' backhaul links, and, eventually, to the underutilization of network resources.

To improve load balancing and increase capacity of WMNs, previous studies suggested to use balanced tree structures rooted at the gateways, and to route the traffic along the tree paths. For instance, an heuristic for calculating load-balanced shortest path trees taking into account flow load is proposed in [HCC05]. In [BHK07], approximated solutions are defined for calculating load-balanced trees that allocate the same bandwidth to all the nodes, using both single-path and multi-path approaches. An alternative strategy is proposed in [MBLD07], where the complexity of finding optimal routes is mitigated by considering only delay optimal routing forests, i.e., unions of disjoint trees routed at the gateway nodes. However, tree-based routing structures are less reliable to link failures than mesh-based structures. Furthermore, the admission of a new flow usually triggers complex reconfiguration procedures for the entire tree.

A simpler approach to improve network performance is to define routing metrics for traditional shortest-path first routing protocol capable of discovering high-throughput paths and/or facilitating load balancing. Initially proposed metrics (e.g., ETX [DCABM03] and ETT [DPZ04]), focused only on link characteristics (e.g., frame loss and transmission rates), and they do not balance the load. Recent studies proposed to introduce in the metric computation estimates of inter-flow and intra-flow interference (e.g., IRU [YWK06]), location-dependent contention (e.g., CATT [GS08], ETP [MBLD07]) or load-dependent cost (such as the queue length in WCETT-LB [MD07], or the number of per-link admitted flows in LAETT [ALCR08]). Although these metrics have been demonstrated to work quite well in mesh networks, and to provide higher throughput performance than simple hop count, they are completely unaware of the available resources at the gateways. On the contrary, significant performance improvements might be obtained by considering residual capacity of gateways' Internet connections, as well as load distributions, when routing traffic flows. However, there is a complex interdependence between the way

traffic flows are routed in the network and the utilization of network resources, which makes quite difficult to define simple heuristics to estimate the remaining capacity of a network path or a gateway.

If the capacity prediction is important in providing QoS to the customers, guaranteeing the correct behavior of applications such as file transfer, in many other real time applications, such as voice and video communications, it is actually crucial to predict the end-to-end packet delay. This task, in a wireless environment, is even more challenging than the capacity management because of the interdependencies among packet loss rates at the physical layer, random access MAC protocol, traffic routing and load distribution, difficult to analytically model.

Another important issue to deal with, in WMNs, is the radio coverage. Like in a WLAN, also in a WMN the user access points (both in mesh gateways and routers) are placed in fixed position, determining and limiting the coverage area for the mobile users. To extend the range of a WLAN, two approaches are traditionally followed in real practice. On the one hand, it would be possible to increase the transmission power of an access point in order to reach farther nodes. However, the main shortcoming of this solution is that it may lead to a poor channel reuse because a larger number of users should access the network through the same base station. Consequently, the contention level within each cell increases, thus degrading the per-client throughput. Moreover, the effectiveness of this technique is limited by the fact that the IEEE 802.11 technology operates in an unlicensed frequency spectrum (i.e., the ISM band) [The99], and national regulations usually set stringent limits to the maximum transmission-power levels in unlicensed bands. Alternatively, we may opt for deploying more access points at a closer spacing, increasing the network capacity. However, a number of reasons, including co-channel interference between nearby access points, availability of a limited number of orthogonal non-interfering frequency channels, as well as cost and management overheads, limit the effectiveness of this alternative solution.

To overcome the limitations of the above-discussed approaches, several authors have recently advocated a new architecture for WLANs (naturally applicable to WMNs), which integrates ad hoc networking technologies in the network infrastructure [LBB04, KSK04, NLP05, ABC+07]. Traditionally, mobile ad hoc networks (MANETs) are conceived as an isolated collection of mobile

nodes connected together over a wireless medium, which self-organize into an autonomous multi-hop wireless network [CG07b]. However, it is now recognized that the ad hoc networking paradigm can also be applied to infrastructure-based wireless networks, building an *hybrid ad hoc network*, and providing a flexible, robust and cost-effective increase of network coverage. Specifically, we envisage an *extended WLAN* in which static and mobile clients transparently communicate using traditional wired technologies or ad hoc networking technologies. The same concept can be applied to extend a WMN through a MANET so to have a hybrid WMN. Thus, the client traffic can be forwarded to the access points through multi-hop wireless paths established by using an ad hoc routing protocol [ABC+07]. It is important to underline that other classes of hybrid ad hoc networks have emerged from this vision, such as: Multihop Cellular Networks (MCN), which combine the features of cellular systems and ad hoc networks [LH00], and mesh networks, which employ a multi-hop wireless backbone to provide Internet access to mobile users [BCG05].

Several technical challenges have to be faced in order to construct such an hybrid ad hoc network because the characteristics of the ad hoc networking (e.g., multi-hop relaying, lack of a centralized administration, etc.) differ significantly from the conventional IP architecture. For instance, the address autoconfiguration protocols commonly used in infrastructure WLANs, such as the Dynamic Host Configuration Protocol (DHCP) [Dro97] or the Zeroconf protocol [CAG05], are not directly applicable in multi-hop wireless networks. However, a mobile device cannot participate in unicast communications until it has been assigned a free IP address and the corresponding subnet mask. It is evident that pre-configuration is impractical in mobile environments, as well as a violation of the self-organizing paradigm. Thus, an address autoconfiguration protocol is crucial to allow the dynamic and automatic allocation of unique IP addresses to mobile clients.

1.2 Contributions

To address the issues described in Section 1.1, in this thesis we provide the following main contributions.

1. We consider wireless mesh networks (WMNs) used to share the Internet connectivity of sparsely deployed fixed lines with heterogeneous capacity,

ranging from ISP-owned high-speed links to subscriber-owned low-speed connections. If traffic is routed in the mesh without considering the load distribution and the bandwidth of Internet connections, some gateways may rapidly get overloaded because they are selected by too many mesh nodes. This may cause a significant reduction of the overall network capacity. To address this issue, we firstly develop a queuing network model that predicts the residual capacity of network paths, and identifies network bottlenecks. By taking advantage of this model, we design a novel Load-Aware Route Selection algorithm, named LARS, which improves the network capacity by allocating network paths to upstream Internet flows so as to ensure a more balanced utilization of wireless network resources and gateways' Internet connections. Using simulations and a prototype implementation, we show that the LARS scheme significantly outperforms the shortest-path first routing protocol using a contention-aware routing metric, providing a high throughput improvement in various network scenarios.

2. In practical wireless mesh networks if traffic is routed without considering wireless and wired constraints, in terms of bandwidth and queues, as well as the traffic distribution, some gateways or intermediate mesh routers may rapidly get overloaded, leading not only to throughput bottleneck in some critical nodes but also increasing the queuing delay which could become unacceptable for real-time applications such as voice or video. To address these problems, we developed a multi-class queuing network model to analyze feasible throughput allocations, which is able to predict the averagee end-to-end packet delay, in heterogeneous WMNs, for both upload and download traffic.

3. We focus on hybrid wireless mesh networks addressing the problem of auto-configuration. The IP address auto-configuration of wireless mobile nodes in hybrid mesh network is a crucial issue because the ad hoc networking (e.g., multi-hop relaying, lack of a centralized administration, etc.) differ significantly from the conventional IP architecture. To address this issue, in this thesis we propose extensions to DHCP to enable the dynamic allocation of globally routable IPv4 addresses to mobile stations in hybrid ad hoc networks, which transparently integrate conventional

wired technologies with wireless ad hoc networking technologies. Some of the attractive features of our solution are its ability to cope with node mobility, the introduction of negligible protocol overheads, and the use of legacy DHCP servers. We have implemented a prototype of our scheme, and tested its functionalities considering various topology layouts, network loads and mobility conditions. The experimental results show that our solution ensures short address configuration delays and low protocol overheads.

4. Communications infrastructures are a critical asset in today's Information society. However, legacy telecommunication systems easily collapse in case of disruptions that may occur due to security incidents or crises. In this thesis, we firstly elaborate on the major shortcomings of the current communications networks for security applications to identify the key missing requirements for such networks. Then, we show that the ad hoc networking technologies, coupled with disruptive-tolerant techniques, are the best suited paradigm to build the next generation of dependable, secure and rapidly deployable communications infrastructures. In particular, we focus on mesh, opportunistic, vehicular, and sensor networks, giving an overview of the most recent advances and summarizing the challenges facing the design and the deployment of these networks. Finally, we conclude presenting the open research issues to realize the vision of a dependable communications infrastructure, with special attention to aspects such as interoperability among multiple heterogeneous networks, autonomic network management and QoS protection.

1.3 Publications Related to the Thesis

The work presented in this thesis has resulted in the following publications in international journals, conferences and books, as specified in the following.

1. The content of Chapter 2 ("A Framework for Load Aware Routing in Wireless Mesh Networks") and Chapter 4 ("Experimental Performance Evaluation") is based on the following journal and conference papers:

   (a) E. Ancillotti, R. Bruno, M. Conti, E. Gregori, and A. Pinizzotto. Load-Aware Routing in Mesh Networks: Models, Algorithms and

Experimentation. *Computer Communications*, 34(8):948-961, June 1 2011. DOI:10.1016/j.comcom.2010.03.004., in press.

(b) R. Bruno, M. Conti, and A. Pinizzotto. A Queuing Modeling Approach for Load-Aware Route Selection in Heterogenous Mesh Networks. In *Proc. of IEEE WoWMoM09*, Kos , Greece, June 15-19 2009.

2. The content of Chapter 3 ("Theoretical Performance Evaluation") is based on the following journal and conference papers:

   (a) R. Bruno, M. Conti, and A. Pinizzotto. Routing Internet Traffic in Heterogeneous Mesh Networks: Analysis and Algorithms. *Performance Evaluation*, 2011. DOI:10.1016/j.peva.2011.01.006, in press.

   (b) R. Bruno, M. Conti, and A. Pinizzotto. Capacity-Aware Routing in Heterogeneous Mesh Networks: An Analytical Approach. In *Proc. of IEEE MsWiM09*, Tenerife , Canary Islands, Spain, October 26-30 2009.

3. The content of Chapter 5 ("Hybrid Mesh Networks") is based on the following journal and conference papers:

   (a) E. Ancillotti, R. Bruno, M. Conti, E. Gregori, and A. Pinizzotto. Dynamic address autoconfiguration in hybrid ad hoc networks. *Pervasive and Mobile Computing*, 5(4):300-317, August 2009.

   (b) R. Bruno, M. Conti, and A. Pinizzotto. Enhancing DHCP for Address Autoconfiguration in Multi-hop WLANs. In *ICDCN 2008*, volume 4904 of *Lecture Notes in Computer Science*, pages 528 539, Kolkata, India, January 5-8 2008. Springer.

4. Finally, the content of Chapter 6 ("Mesh Networks: An Application Scenario") is based on the following book chapter:

   (a) R. Bruno, M. Conti, and A. Pinizzotto. Mobile Ad Hoc Sensor Systems for Global and Homeland Security Applications. In S. Msra, I. Woungang, and S. Misra, editors, *Guide to Wireless Sensor Networks*, pages 687708. Springer London Publisher, May 2009.

1.4 Outline of the Thesis

The thesis is organized as follows. In Chapter 2, we introduce the framework of the heterogeneous wireless mesh networks (WMN) and present a Load Aware Routing Selection algorithm (LARS) designed to provide a QoS optimizing the traffic distribution in terms of overall throughput. The algorithm is based on a novel networking queuing model. In Chapter 3 we extend to model to a multi-class queueing network analytical model, which is also able to predict the average end-to-end packet delay. Chapter 4 describes a prototype implementation of the LARS solution as a proof-of-concept, on small-scale experiments conducted in our trial mesh network. In Chapter 5 we propose a solution to the address auto-configuration of a multi-hop ad hoc network connected to an extended fixed networking infrastructure such as a legacy wired network connected to a WMN. The proposed protocol is implemented ed evaluated on an experimental test-bed. Chapter 6 gives an overview of the ad hoc networking technologies suitable to replace the legacy telecommunication systems as a survivable communications system in disaster scenarios. At the end, in Chapter 7, we summarize the main findings and contributions of this thesis along with some future research directions.

Chapter 2

A Framework for Load Aware Routing in Wireless Mesh Networks

2.1 Introduction

802.11-based wireless mesh networks are emerging as a key technology to provide cost-effective ubiquitous access to the Internet [KSK04]. Normally, in mesh networks only a subset of routers, referred to as *gateways*, has a high-speed Internet connection, while Internet access is shared among all the other mesh nodes by exploiting the ad hoc routing capabilities of the mesh routers [BCG05, AWW05]. However, this vision is rapidly changing. Real-world mesh networks have been recently deployed, which are used to share a potentially large number of low-speed Internet connections (i.e., DSL fixed lines) available at the customers' premises. Examples of such networks are Meraki-based deployments in urban areas [Mer], or the Ozone's network in Paris, which is composed of 400 mesh routers, most of them using standard DSL links as Internet backhaul, while only ten gateways are provided with an ISP-owned fiber link [Ozo]. In a broader sense, wireless mesh networks are evolving into a *converged infrastructure* used to share the Internet connectivity of sparsely deployed fixed lines with *heterogeneous capacity*, ranging from ISP-

owned broadband links to subscriber-owned low-speed connections [SACB08].

Being mesh networks primarily used for Internet access, both traffic routing and Internet gateway selection play a crucial role in determining the overall network performance, and in ensuring the optimal utilization of the mesh infrastructure [LLT03, ZWR08]. For instance, if too many mesh nodes select the same gateway as egress point to the Internet, congestion may increase excessively on the wireless channel, or the Internet connection of the gateway can get overloaded. This is especially important in the heterogeneous mesh networks we consider in this work, because low-speed Internet gateways may easily become a *bottleneck*, limiting the achievable capacity of the entire network. In addition a load-unaware gateway selection can lead to an unbalanced utilization of the gateways' backhaul links, and, eventually, to the underutilization of network resources.

To improve load balancing and increase capacity of WMNs, previous studies suggested to use balanced tree structures rooted at the gateways, and to route the traffic along the tree paths. For instance, an heuristic for calculating load-balanced shortest path trees taking into account flow load is proposed in [HCC05]. In [BHK07], approximated solutions are defined for calculating load-balanced trees that allocate the same bandwidth to all the nodes, using both single-path and multi-path approaches. An alternative strategy is proposed in [MBLD07], where the complexity of finding optimal routes is mitigated by considering only delay optimal routing forests, i.e., unions of disjoint trees routed at the gateway nodes. However, tree-based routing structures are less reliable to link failures than mesh-based structures. Furthermore, the admission of a new flow usually triggers complex reconfiguration procedures for the entire tree.

A simpler approach to improve network performance is to define routing metrics for traditional shortest-path first routing protocol capable of discovering high-throughput paths and/or facilitating load balancing. Initially proposed metrics (e.g., ETX [DCABM03] and ETT [DPZ04]), focused only on link characteristics (e.g., frame loss and transmission rates), and they do not balance the load. Recent studies proposed to introduce in the metric computation estimates of inter-flow and intra-flow interference (e.g., IRU [YWK06]), location-dependent contention (e.g., CATT [GS08], ETP [MBLD07]) or load-dependent cost (such as the queue length in WCETT-LB [MD07], or the num-

ber of per-link admitted flows in LAETT [ALCR08]). Although these metrics have been demonstrated to work quite well in mesh networks, and to provide higher throughput performance than simple hop count, they are completely unaware of the available resources at the gateways. On the contrary, significant performance improvements might be obtained by considering residual capacity of gateways' Internet connections, as well as load distributions, when routing traffic flows. However, there is a complex interdependence between the way traffic flows are routed in the network and the utilization of network resources, which makes quite difficult to define simple heuristics to estimate the remaining capacity of a network path or a gateway.

To address this problem, in this chapter we make the following two main contributions. First of all, we develop a queuing-based model of an heterogeneous mesh network, which incorporates the interdependencies between packet loss rates at the physical layer, random access MAC protocol, traffic routing and load distribution. This model is used to estimate the network capacity, and to identify network bottlenecks, due to either congestion on the wireless channels or overloading of Internet fixed lines. Then, we propose a novel *Load-Aware Route Selection* algorithm, named *LARS*, which integrates traffic routing with gateway selection. The goal of LARS is to improve network capacity, and to avoid underutilization of gateways' resources. The idea behind the design of the LARS algorithm is to allow each mesh node to distribute the traffic load among multiple gateways to ensure evenly utilization of Internet connections. To this end, mesh nodes select the routes towards the gateways taking into account the residual capacity of the paths, and the utilization of the gateways' fixed lines. We exploit the proposed queuing model to predict the residual capacity of each network path, and to discard paths or gateways that cannot accept additional demands ([ABCP11]).

It is important to point out that previous studies have proposed to use queuing models to investigate system performance of CSMA-based ad hoc networks. However most of these studies have applied queuing theory to the analysis of *single-hop* ad hoc networks [ASS03, OM04, TS08]. To the best of our knowledge, in literature a few examples exist which deal with the multi-hop case. In [BA09], the authors model random access multi-hop wireless networks as open $GI/G/1$ queuing networks to analyze the average end-to-end delay and maximum achievable per-node throughput. However, the formulation proposed

in [BA09] can be applied only to random networks, and it does not incorporate flow-level behaviors. Our objective is different from [BA09], because we consider arbitrary topologies and routing strategies, and we focus on per-flow performance. Previous papers [BCP09a, BCP09b] have also developed queuing models to analyze the network capacity of heterogeneous WMNs. This chapter extends those analytical studies to incorporate packet losses in the channel modeling.

We evaluate the performance gains provided by the LARS scheme over the shortest-path first routing algorithm using a contention-aware routing metric performing both simulations under large-scale network scenarios. Our simulations demonstrate the accuracy of the proposed modeling framework over a wide range of network settings. Furthermore, the numerical results show that proposed route and gateway selection algorithms significantly outperforms the shortest-path first routing using a contention-aware routing metric, providing up to 240% throughput improvement in some network scenarios.

The remaining of this chapter is organized as follows. Section 2.2 introduces the overall system architecture. Section 2.3 develops the capacity analysis for a multi-hop heterogeneous mesh network. In Section 2.4, we describes the LARS algorithm. Section 2.5 validate the analysis and evaluate the performance gains of proposed algorithms using simulation. Finally, conclusions and future extensions are discussed in Section 2.6.

2.2 System architecture overview

Figure 2.1 illustrates the reference network architecture adopted in this work. As shown in the diagram, in the WMN there are *stationary wireless routers*, termed mesh routers, which form a multi-hop wireless backbone network. Connected to the mesh routers there are *local access points* (APs) that aggregate and forward traffic from mobile clients associated to them. In addition, a small subset of the deployed mesh routers is composed of *gateways* with a fixed connection to the Internet, providing wide- area connectivity for all mesh clients. Thus, the mesh nodes do not generate traffic flows but they only forward the user traffic to the gateways through wireless multi-hop transmissions.

Differently from traditional mesh architectures, which generally assume a limited number of mesh gateways connected to the Internet through unlimited-

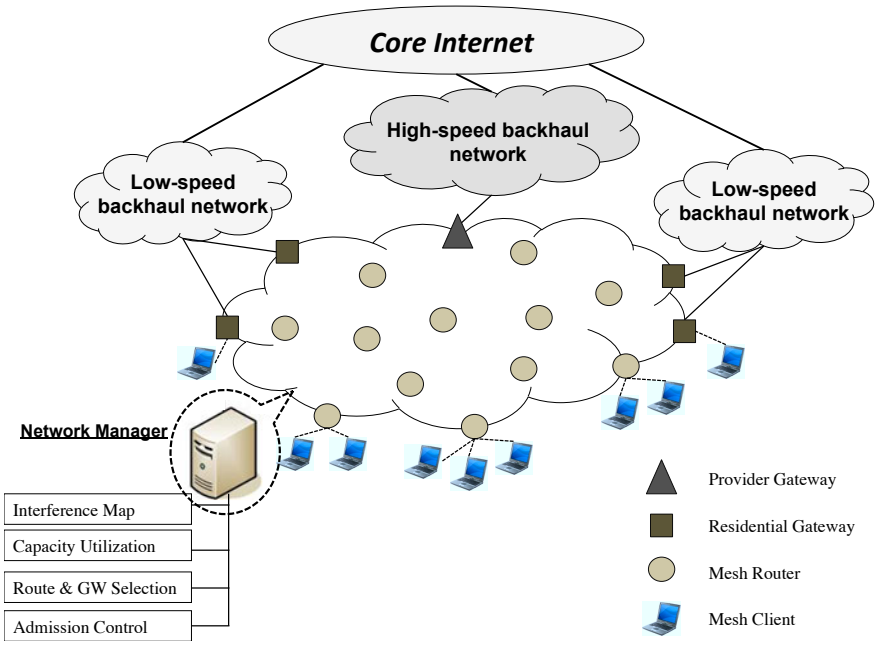


Figure 2.1: Reference network architecture.

bandwidth fixed links, we recognize that mesh gateways can have connections to the speed with highly heterogeneous capacity. Specifically, there are a few gateways that have an high-speed backhaul connection (e.g., optical fiber or fixed broadband wireless) to the Internet. Since such high-capacity links are usually installed by network providers, we refer to this category of gateways as *provider gateways*. However, there might be a number of mesh routers directly owned by mesh network subscribers, which can share their low-speed Internet connection (e.g., ADSL cable) with other mesh users. Thus, we refer to this second category of mesh gateways as *residential gateways*. On the user side, wireless clients have a direct wireless connection to one mesh router, which acts as aggregator point for the user-generated traffic.

In this architecture we assume that QoS provisioning is managed by a centralized entity, hereafter called *network manager*. This entity is responsible for the admission of a new arriving traffic flow, and for the efficient selection of routes satisfying the QoS demands of that flow. The network manager implements a set of components to perform its tasks. First of all, it generates a connectivity map and interference characterization of the mesh network based

on the statistics collected from each mesh node. Moreover, this manager node maintains a list of the admitted traffic flows and the network paths used to route their traffic. Interference map and load information are then used to construct the capacity utilization model, and to drive the route and gateway selection process for a newly arriving flow. Although a distributed approach for network management would provide better network resilience, there are many advantages in adopting such an architecture that *centralizes* certain functions of the routing protocol. Firstly, it facilitates the deployment of intelligent algorithms that exploit a global knowledge of the network status (e.g., connectivity and interference maps, offered loads, resource utilization, etc.) to provide network optimizations (e.g., topology and channel management, or QoS policies). In addition, several commercial solutions of mesh networking have already a centralized controller where CAC and special routing functionalities can be deployed. Finally, this architecture is also compatible with one of the deployment scenarios recently proposed by the CAPWAP working group [CMS09] for the management, monitoring, and control of large number of interconnected wireless access devices. Note that typical scalability issues of a centralized scheme in large-scale WMNs could be addressed by a hybrid approach, where the WMN is divided into clusters, and network managers for each cluster coordinate resource control decisions in a distributed manner.

In the following two sections we detail the operations performed by the capacity utilization model, and the gateway and route computation module.

2.3 Modeling network capacity

In this section we develop the capacity utilization model of the heterogeneous WMN described in Section 2.2. This model will be used to estimate every node's available capacity, which is needed to compute a feasible route for a new arriving traffic flow. For the sake of clarity, Table 2.1 summarizes the notation used in the following analysis.

2.3.1 Network model

The key idea behind the proposed modeling framework is to convert the physical network model into an equivalent queuing network model. Both these network models can be represented as graphs. In the physical network model

variable	definition
$\mu_{i,l}$	mean service rate of packets at the queue l of the station i
λ_i^e	arrival rate of packets from outside (i.e., mesh clients) to station i
$\lambda_{i,l}$	overall arrival rate of packets at the queue l of the station i
$\lambda_{i;j}^f$	mean rate of packets transferred from station i to station j
$\lambda_{i;j}^t$	mean rate of packets served at station i and heading towards station j
$\lambda_{i;j}^r$	mean rate of packets that are corrupted when transmitted on the wireless link from station i to station j
λ_i^r	overall rate of packets to be retransmitted that are inserted in the wireless queue of station i
$p_{i,l;j,s}$	*routing probability*: the probability that a job is transferred to queue s of node j after service completion at queue l of node i.
$r_{i;j}$	*retransmission probability*: the probability that a job served at the wireless queue of node i is corrupted when transferred to node j. It holds that $\lambda_{i;j}^r = r_{i;j} \cdot \lambda_{i;j}^t$.

Table 2.1: Model notation

the vertexes of the network graph are the mesh nodes, and the edges characterize the physical network connectivity relationships (e.g., in terms of channel bandwidth and packet transmission error rate of each link) that exist between the communicating entities. On the other hand, in the equivalent queuing network model each mesh node is represented using an equivalent queuing station, and the edges characterize the packet forwarding process between two queues.

Formally, let $\mathcal{G}_r$, $\mathcal{G}_p$ and $\mathcal{M}$ be the set of residential gateways, provider gateways and mesh routers, as defined in Section 2.2. Let n_w, n_r and n_p be the cardinality of the $\mathcal{M}$, $\mathcal{G}_r$, and $\mathcal{G}_p$ sets, respectively, with $n = n_w + n_r + n_p$. Then, the physical network model can be described using a mixed graph $G(V \cup \{a\}, E_w, E_g)$, where the graph vertexes V ($|V| = n$) represent the mesh nodes (i.e., $V = \mathcal{G}_r \cup \mathcal{G}_p \cup \mathcal{M}$) and a is a virtual vertex that corresponds to the fixed infrastructure (i.e., the Internet). We denote by E_w the set of edges representing the wireless links between mesh nodes, while E_g is the set of edges representing the backhaul links between the gateways and the infrastructure. The neighborhood of node $v \in V$, denoted by $N(v)$, is the set of nodes to which node v is physically connected. If node v is a gateway, the virtual node a is included in the neighborhood of v.

To construct the link-layer connectivity map of the WMN, we assume that the *transmission range* of each wireless transmitter is fixed and equal to R_{tx}. Each edge $e_{i,j}$ between node $i, j \in G$ is labelled with a pair of values, $C_{i,j}$ and

$r_{i,j}$. The former parameter represents the nominal capacity associated to that link, while the latter represents the probability that a packet transmitted over that link is corrupted by channel errors. Moreover, we assume that probability $r_{i,j}$ is time-invariant, which is equivalent to assume that the packet errors at the PHY layer can be modeled using a stationary random process. In general, the $r_{i,j}$ process can be derived by integrating a specific PHY layer model into the channel characterization. Alternatively, actual link-layer measurements can be used to extract the statistics of the packet transmissions error rates and to allow a trace-driven emulation of the physical network. For the sake of generality, in the following we do not model the statistics of the $r_{i,j}$ process, but we only assume that it is stationary.

For simplicity, we assume that each $e_{i,j} \in E_w$ has a fixed and constant transmission rate C_w, i.e., no rate adaptation is used on the wireless links. Moreover, the low-speed backhaul link from a residential gateway $i \in \mathcal{G}_r$ to the wired infrastructure a has fixed capacity C_r, while the high-speed backhaul link between a provider gateway $i \in \mathcal{G}_p$ and the wired infrastructure a has fixed capacity C_p. Generally, it holds that $C_r \ll C_p$.

From the graph representation $G(V \cup \{a\}, E_w, E_g)$ of the heterogeneous WMN, we are able to derive an equivalent queuing network model $G'(Q, L)$, where Q indicates the set of queuing systems in the network, for brevity *stations*, and L is the set of connections between stations. Intuitively, jobs in the queuing network represent packets in the physical network[1]. Owing to the analogy between the physical network and an equivalent queuing network, each mesh node $i \in V$ is modeled through a service station $k \in Q$. In general, this equivalent queuing station may include several queues. This allows us to model mesh nodes with multiple interfaces such as the gateways, which are equipped with wired and wireless interfaces. Furthermore, the internal queuing structure of the queuing station is also exploited to provide the model with the flexibility to characterize different routing strategies. Hereafter, $q(j)$ indicates the number of internal queues at station j. For simplicity, in this study we assume that all queues have infinite size and serve packets according to a FCFS discipline.

It is intuitive to note that, being the WMN composed of two classes of mesh nodes, gateway and mesh routers, at least two different queuing station models should be specified for the analysis. For ease of explanation, Figure 2.2

[1] In the following, the terms job and packet are used equivalently.

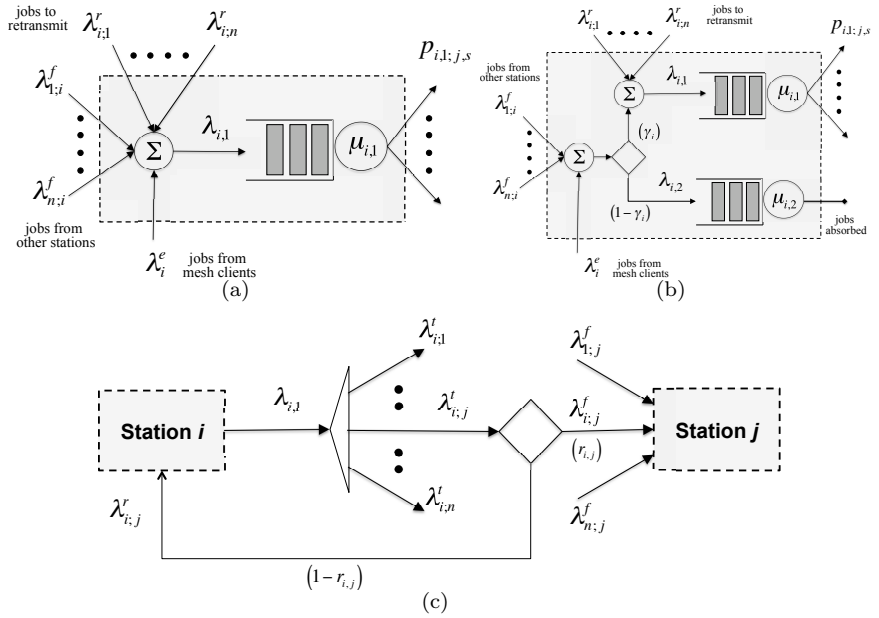


Figure 2.2: Components of the queuing network model: *a*) structure of a non-gateway station, *b*) structure of a gateway station, and *c*) wireless channel model.

exemplifies the structure of the queuing stations used to model mesh nodes. Specifically, Figure 2.2(a) illustrates a station modeling a wireless mesh router i ($i \in \mathcal{M}$) not connected to the wired infrastructure. This station consists of a singe queue (i.e., $q(i) = 1$), say $q_{i,1}$, which models the transmissions on the wireless channel[2]. Let us denote with $\mu_{i,1}$ the mean service rate of queue $q_{i,1}$. As observed in Section 2.2, each mesh node i aggregates the traffic flows originated from the mesh clients associated with it. We model this aggregated traffic through its average packet arrival rate, say λ_i^e. In addition to locally generated jobs, station i can receive jobs from each of its neighboring stations, with average arrival rate $\lambda_{j;i}^f$ ($j \in N(i)$). On the other hand, station i transfers jobs to its neighboring stations. Let $\lambda_{i;j}^t$ be the overall rate of jobs served at queue $q_{i,1}$ that are headed towards stations j. A fraction $r_{i;j}$ of these jobs will be retransmitted because corrupted by channel errors. These retransmissions

[2]For simplicity, we consider a single wireless interface. The extension to multiple wireless interfaces is straightforward, but it would require the incorporation in the model of a channel assignment algorithm, which is out of the scope of this study.

can be modeled as an additional ingress traffic for queue $q_{i,1}$, with mean rate $\lambda^r_{i;j}$ equal to $\lambda^t_{i;j} \cdot r_{i;j}$. Consequently, the overall rate of packets that are re-inserted in queue $q_{i,1}$ after been served is $\lambda^r_i = \sum_k \lambda^r_{i;k}$. For clarity, the described channel model is also illustrated in Figure 2.2(c). Owing to the previous considerations, it holds that the total arrival rate at queue $q_{i,1}$ can be expressed as $\lambda_{i,1} = \sum_j \lambda^f_{j;i} + \sum_k \lambda^r_{i;k}$, while the mean rate of jobs that are transferred from station i to station j can be computed as $\lambda^f_{i;j} = (1 - r_{i,j}) \cdot \lambda^t_{i;j}$.

Figure 2.2(b) describes the internal structure of a station modeling a gateway node i ($i \in \mathcal{G}_r \cup \mathcal{G}_p$). In this case, the queuing station structure consists of two queues: $q_{i,1}$, which models wireless transmissions, and $q_{i,2}$, which models the transmissions on the gateway's backhaul link. Similarly to non-gateway stations, each gateway station can receive packets forwarded by neighboring stations, with average arrival rate $\lambda^f_{j;i}$ ($j \in N(i)$), as well as packets generated by associated mesh clients, with average arrival rate λ^e_i. Then, the logic implemented internally to the gateway station determines what fraction of the arriving jobs is transferred to $q_{i,1}$ or $q_{i,2}$. It is important to note that in this work we are primarily concerned with *upstream Internet traffic*. In other words, we assume that traffic flows are originated from mesh nodes and destined for the Internet (i.e., the virtual node a). Hence, when a job reaches a gateway station, it could be directly transferred to queue $q_{i,2}$, and then leave the network after being served. However, a residential gateway may have a low-speed upstream connection to the Internet, which rapidly becomes a bottleneck as the traffic received on the wireless interface builds up, limiting the achievable capacity of the whole mesh network. To make this limitation less severe, the residential gateway may take advantage of the available wireless bandwidth to behave as a relay node, and further forwarding the traffic to one of its neighbors, which may be less congested, or closer to a provider gateway. To model this capability we introduce the *re-forwarding* probability γ_i. Specifically, a job received by gateway i is routed through the wireless queue $q_{i,1}$ with probability γ_i, or directly through the upstream wired queue $q_{i,2}$ with probability $(1-\gamma_i)$. The design of the γ_i function depends on the routing and resource allocation strategies implemented in the mesh network. Finally, similarly to non-gateway stations, jobs transmitted by queue $q_{i,1}$ can be corrupted by channel errors and they will be retransmitted. These retransmitted jobs will contribute to the overall arrival rate at queue $q_{i,1}$ with an additional flow of jobs having mean

arrival rate equal to λ_i^r.

Before concluding this section, it is important to describe the wireless channel access-control mechanisms we adopt for the MAC layer, because they significantly affect the estimate of the $\mu_{i,1}$ values. In this study, we assume that a simplified CSMA-based MAC protocol is used by the wireless transmitters to coordinate simultaneous transmissions of interfering nodes. This basic MAC scheme implements an idealized collision avoidance mechanism. More precisely, we assume that each node has an instantaneous knowledge of the communication state (i.e., idle, receiving or transmitting) of other interfering nodes, so as to ensure that it starts transmitting only when its transmitted packets does not cause a collision. This is somehow equivalent to determine a collision-free random transmission schedule among contending nodes. To model the interference relationships between contending mesh nodes we use the Protocol Model as in [BA09, BCP09a]. In other words, a transmission from mesh station i to mesh station j, with $i, j \in Q$, is *admissible* if the following conditions are satisfied: 1) $dist_{i,j} \leq R_{tx}$ and 2) for every other transmitting node k, $dist_{k,j} \geq (1+\Delta) \cdot R_{tx}$, where $dist_{i,j}$ is the Euclidean distance between node i and node j, and Δ is a positive constant that represents a guard zone in the Protocol Model. Note that not all the admissible transmissions are successful, because we incorporate in the analysis a retransmission probability $r_{i,j}$. The collision-less MAC protocol used in our study might be considered somehow restrictive, especially because we neglect the detailed protocol implementation of collision avoidance and resolution mechanisms, such as 802.11-like backoff schemes. However, though in a simplified form, this MAC scheme captures the fundamental aspects of location-dependent contention inherent to multi-hop environments, which is due to differences in the number of contending nodes at both endpoints of each communication link. In other words, in this study we are more concerned on modeling the link capacity degradation due to location-dependent contention, rather than precisely incorporating in the analysis all the features of the IEEE 802.11 MAC protocol.

2.3.2 Node utilization

In this section we develop the analysis to determine if a given throughput allocation in $G(V \cup \{a\}, E_w, E_g)$ is *feasible*. Before formally defining when a throughput allocation is feasible, and describing our analytical methodology, it

is useful to introduce some notation.

Let us denote with λ^e the overall arrival rate of a job from *outside* (i.e., from mesh clients associated to mesh nodes) to the mesh network. Furthermore, let $p^e_{i,l}$ be the probability that a job from outside the network enters queue l of station i. This implies that the job arrival rate from outside to queue l of station i is $\lambda^e_{i,l} = \lambda^e \cdot p^e_{i,l}$. For brevity, we introduce the *probability matrix of external arrivals* defined as $\mathbf{P^e} = \{p^e_{i,l}, i \in Q, l \in [1, q(i)]\}$. Note that this notation conforms to the network model formulated in Section 2.3.1.

Definition 1. Throughput allocation. *A throughput allocation for* $G(V \cup \{a\}, E_w, E_g)$ *is any assignment for the rate* λ^e *and the probability matrix* $\mathbf{P^e}$.

A fundamental parameter to compute the arrival rate at every queue $q_{j,s}$ is the *routing probability* $p_{i,l;j,s}$ defined as the probability that a job is transferred from the queue l of station i (i.e., $q_{i,l}$) to queue s of station j (i.e., $q_{j,s}$). Following the equivalency between the real WMN and the queuing network, the $p_{i,l;j,s}$ value expresses the probability that mesh node i selects mesh node j as next-hop to reach the Internet. The queue indexes are used to specify if the packet is transmitted using the wireless links or the wired fixed lines. For the sake of brevity, we introduce the network *routing matrix* defined as $\mathbf{R_{fwd}} = \{p_{i,l;j,s}, i, j \in Q, l \in [1, q(i)], s \in [1, q(j)]\}$, which is the probabilistic representation of the underlying routing process. Now, we can define the feasibility of a throughput allocation as follows.

Definition 2. Feasible throughput allocation. *A throughput allocation is feasible for a given routing matrix* $\mathbf{R_{fwd}}$ *if every queue* $q_{i,l}$ *(*$i \in Q$ *and* $l \in [1, q(i)]$*) has a bounded time-average number of packets. This is equivalent to state that arrival process at queue* $q_{i,l}$ *is admissible with rate* $\lambda_{i,l}$.

From a mathematical point of view, Definition 2 implies that a throughput allocation is feasible if all the queues in the system are stable, i.e., the number of jobs waiting in queue does not grow indefinitely. From a more practical perspective, to determine if a throughput allocation is feasible is equivalent to verify that the allocation of a given set of flows on a given set of network paths does not violate the network capacity constraints. To verify the queue stability we have to compute the queue's *utilization factor* [BGdMT06]. From elementary queuing theory this requires the evaluation of the first moments of the packet arrival and service processes at each queue of the network. Specifically,

the utilization $\rho_{i,l}$ of the queue l at station i (i.e., $q_{i,l}$) is $\rho_{i,l} = \lambda_{i,l}/\mu_{i,l}$. By definition, an infinite-size queue is stable if and only if $\rho_{i,l} < 1$.

Note that in statistical equilibrium the rate of departure from a queue is equal to the rate of arrival, and the overall arrival rate at queue $q_{i,l}$ can be written as:

$$\lambda_{i,l} = \lambda^e \cdot p^e_{i,l} + \sum_{\substack{j=1 \\ j\neq i}}^{n} \sum_{s=1}^{q(j)} \lambda_{j,s} \cdot p_{j,s;i,l} + \begin{cases} \sum\limits_{\substack{k=1 \\ k\neq i}}^{n} \lambda^t_{i;k} \cdot r_{j,k} & l = 1 \\ 0 & l = 2 \end{cases}, \text{ for } i \in Q, l \in [1, q(i)] \; . \tag{2.1}$$

It is intuitive to observe that the specific formulation of the routing matrix depends on several factors including the routing algorithm used in the WMN, the network topology, the throughput allocation and the retransmission probabilities.

The following lemma provides a methodology to compute the routing matrix $\mathbf{R_{fwd}}$ without solving the system defined in Equation 2.1, but considering a simplified queuing network.

Lemma 1. *Let $\overline{p}_{i,l;j,s}$ the routing probability of a simplified queuing network $\overline{G'}(Q, L)$ obtained from $G'(Q, L)$ by setting $r_{i,j} = 0$ for $i, j \in Q$, while leaving unmodified all the other characteristics, i.e., λ^e, $\mathbf{P^e}$, and how the jobs are routed between the stations. Then, it holds that*

$$p_{i,l;j,s} = \begin{cases} \frac{\overline{p}_{i,l;j,s}}{1 + \sum\limits_{\substack{k=1 \\ k\neq i}}^{n} \frac{r_{i,k}}{1-r_{i,k}} \cdot \overline{p}_{i,l;j,s}} & i \neq j \\ 1 - \sum\limits_{\substack{k=1 \\ k\neq i}}^{n} p_{i,l;k,s} & i = j \end{cases} \tag{2.2}$$

Proof. Without loss of generality, we prove expression 2.2 for $i, j \in \mathcal{M}$, i.e., when the communication endpoints are two mesh routers. In this case, $q(i) = q(j) = 1$. The other cases can be easily derived following the same line of reasoning.

By definition, it holds that

$$p_{i,1;j,1} = \frac{\lambda^f_{i;j}}{\lambda_{i,1}} \quad , \quad \overline{p}_{i,1;j,1} = \frac{\overline{\lambda}^f_{i;j}}{\overline{\lambda}_{i,1}} \; ,$$

where $\overline{\lambda}^f_{i;j}$ and $\overline{\lambda}_{i,1}$ are the mean rate of jobs transferred from station i to station j in $\overline{G'}(Q,L)$, and the overall rate of jobs entering queue $q_{i,1}$ in $\overline{G'}(Q,L)$, respectively. Since corrupted packets transmitted over the wireless link between station i and station j do not enter into queue $q_{j,1}$, it is intuitive to observe that $\lambda^f_{i;j} = \overline{\lambda}^f_{i;j}$. In other words the same rate of packets are transferred from station i to station j in both $\overline{G'}(Q,L)$ and $G'(Q,L)$. This implies $\lambda^f_{i;j} = \overline{p}_{i,1;j,1} \cdot \overline{\lambda}_{i;j}$, and that

$$\lambda_{i,1} = \overline{\lambda}_{i,1} + \sum_{\substack{k=1 \\ k \neq i}}^{n} \lambda^r_{i;j} \ . \tag{2.3}$$

Now, considering the channel model illustrated in Figure 2.2(c) we can write the following two equalities:

$$\lambda^r_{i;j} = r_{i;j} \cdot \lambda^t_{i;j} \qquad , \qquad \lambda^r_{i;j} = \lambda^t_{i;j} - \lambda^f_{i;j} \ .$$

With simple algebraic transformations, the above expressions can be written as

$$\lambda^r_{i;j} = \frac{r_{i;j}}{1 - r_{i;j}} \lambda^f_{i;j} = \frac{r_{i;j}}{1 - r_{i;j}} \overline{p}_{i,1;j,1} \overline{\lambda}_{i;j} \ . \tag{2.4}$$

By substituting expression 2.4 in formula 2.1, after simple manipulations we obtain equation 2.2, and this concludes the proof. □

Now, the average arrival rate $\lambda_{i,l}$ can be computed from λ^e, $\mathbf{P^e}$ and $\mathbf{R_{fwd}}$ by solving a system of linear equations obtained by writing the flow balance condition at each queue of the system as in equation 2.1. The mean service rates for the queues modeling transmissions on wired links can be easily derived by observing that in switched communication technologies there is no contention. Hence, average service times depend only on the nominal link capacity and the packet size. Then, under the assumption that the packet size is constant and equal to P bits, it holds that $\mu_{i,2} = P/C_r$ if $i \in \mathcal{G}_r$, and $\mu_{i,2} = P/C_p$ if $i \in \mathcal{G}_p$. On the other hand, the derivation of the average service rate for the queues modeling transmissions on the wireless channel is more involved because it is necessary to take into account the location-dependent contention, the distributions of active queues (i.e., queues with at least a packet to serve) and the channel access coordination procedures implemented by the MAC protocol. Several stochastic models have been developed to analyze the access delays of CSMA-based MAC protocols used in multi-hop environments. Recall from Sec-

tion 2.3.1 that in this work we consider a basic collision-free CSMA-based MAC protocol, and we assume that each mesh node has an instantaneous knowledge of the communication state of other interfering nodes. Then, following the footprints of [GCL06] and our previous work [BCP09a], we can model the impact on the channel access of location-dependent contention by employing an *average value analysis*, and considering only the long-term fraction of time each mesh node spends in one of three potential states: transmission state, receiving state, and idle state. This modeling approach will lead to a mathematically manageable and reasonably accurate analysis.

To compute the $\mu_{i,1}$ parameter we analyze the channel events during the $X_{i,1}$ period, defined as the interval from the time instant a job reaches the head of queue $q_{i,1}$ to the time instant in which its service is completed. Under the assumption that the transmission events are identically and independently distributed (i.i.d.), it holds that $\mu_{i,1}=1/E[X_{i,1}]$, where $E[\cdot]$ is the expectation operator. To simplify the derivation of the $E[X_{i,1}]$ expression we condition to the possible destinations of a job served at queue $q_{i,1}$. Specifically, owing to the conditional expectation theory we can write that

$$E[X_{i,1}] = \sum_{j=1}^{n} \sum_{s=1}^{q(j)} E[X_{i,1;j,s}] \cdot p_{i,1;j,s} \ , \tag{2.5}$$

where $X_{i,1;j,s}$ is the time needed by queue $q_{i,1}$ to complete the service of a job heading to queue $q_{j,1}$. This time will mainly depend on the level of contention around the transmitting station i and the receiving station j, i.e., on the distribution of interfering nodes in the network, as well as on their activity level, i.e., the fraction of time these nodes contend for the channel access. More precisely, due to the random access scheme the this packet transmission can be preceded by a number $z_{i,1;j,s}$ of transmissions performed by other contenting stations[3]. Let us denote with $E[B_{i,1;j,s}]$ the average period of channel time occupied by other stations' packet transmissions, which precedes the service of the packet at the head of queue $q_{i,1}$, given that this packet is heading towards queue $q_{j,s}$. Then, under the assumption of fixed packet size, it is straightforward to derive that

$$E[B_{i,1;j,s}] = P \cdot E[z_{i,1;j,s}]/C_w. \tag{2.6}$$

[3]In our idealized MAC scheme transmission attempts are not preceded by backoff delays.

This yields to the following expression for the $E[X_{i,1;j,s}]$ parameter.

$$E[X_{i,1;j,s}] = P \cdot (1 + E[z_{i,1;j,s}])/C_w \ . \tag{2.7}$$

To derive a closed expression for the $E[z_{i,1;j,s}]$ parameter, the key approximation of our analysis is to assume that station i attempts to transmit a packet to station j immediately after the channel becomes idle again with a constant (state independent) probability equal to $\tau_{i,1;j,s}$. This approximation is commonly adopted when modeling CSMA-based random access schemes, and it also known as *decoupling* approximation [KAMG07]. While in single-hop networks it is generally assumed that all nodes have the same *transmission probability*, in our study the location-dependent contention is modeled by admitting different values of the $\tau_{i,1;j,s}$ probabilities. The decoupling approximation yields that $z_{i,1;j,s}$ is geometrically distributed with parameter $\tau_{i,1;j,s}$, that is

$$Pr\{z_{i,1;j,s} = h\} = (1 - \tau_{i,1;j,s})^h \tau_{i,1;j,s} \ . \tag{2.8}$$

Now, it is straightforward to derive that

$$E[B_{i,1;j,s}] = \frac{(1 - \tau_{i,1;j,s})}{\tau_{i,1;j,s}} \cdot S \ , \tag{2.9}$$

and formula (2.7) can be rewritten as $E[X^r_{i,1;j,s}] = P/(C_w \cdot \tau_{i,1;j,s})$.

The following lemma provides an explicit expression for the transmission probability $\tau_{i,1;j,s}$.

Lemma 2. *Under the assumption that reception and transmission events in $G'(Q, L)$ are mutually independent, it holds that*

$$\tau_{i,1;j,s} = \prod_{h \in \mathcal{E}_i} \prod_{u=1}^{q(h)} (1 - \phi_{h,u} \cdot \omega_{h;j}) \cdot \prod_{k \in \mathcal{E}_j \cup \{j\}} (1 - \psi_{k,1}) \ , \tag{2.10}$$

where

- *$\phi_{h,u}$ is the long-term fraction of time spent by queue $q_{h,u}$ receiving packets;*
- *$\psi_{k,1}$ is the long-term fraction of time spent by queue $q_{k,1}$ transmitting packets;*
- *$\omega_{h,u;j}$ is the fraction of wireless queues that are neighbors of station h, but*

they are not interferers for station j, and which have a not-null routing probability towards queue $q_{h,u}$;

- *$\mathcal{E}_i$ is the set of mesh nodes in the interference region of node i (formally, $\mathcal{E}_i = \{h : dist_{h,j} \leq (1+\Delta) \cdot r \ , \ h \in G)$.*

Proof. The proof follows the same line of reasoning of Lemma 1 in [BCP09a]. □

In summary, the analytical methodology we adopt to determine the feasibility of a throughput allocation consists of the following steps. First of all, from the WMN topology $G(V \cup \{a\}, E_w, E_g)$ we extract the equivalent queuing network $G'(Q, L)$. Then, given the throughput allocation (λ_o and $\mathbf{P_o}$), and the routing matrix $\mathbf{R_{fwd}}$, we can determine the overall arrival rate at each queue solving the linear system defined with equation (2.1). From the $\lambda_{i,l}$ values we compute the $\phi_{i,l}$ and $\psi_{i,1}$ parameters, and the $\tau_{i,1;j,s}$ probabilities using Lemma 2. This allows us to derive the average service times of each queue in the network, and to check the feasibility of the throughput allocation.

2.4 Load-aware algorithms for route and gateway selection

In the previous section we have developed a analytical framework to determine if a given routing matrix leads to an unfeasible throughput allocation. In this section we develop a practical *Load-Aware Route Selection* (*LARS*) algorithm, which exploits this framework to construct a routing matrix that avoids unevenly utilization of gateways' backhaul links and network bottlenecks, while ensuring that the resulting throughput allocation is feasible. A key feature of our solution, is to implement a simple and efficient strategy to discover and select feasible paths (i.e., paths with sufficient remaining capacity to accommodate the bandwidth demands of new flows). As a matter of fact, it is unrealistic to perform an exhaustive search because there are exponentially many paths between a source/destination pair, and a brute force strategy does not scale. For these reasons, in the literature various solutions have been proposed for reducing the complexity of this problem. A popular approach is to consider only disjoint and braided paths [WB06], but it is still computationally intensive to construct multiple disjoint paths. An alternative strategy is proposed

in [MBLD07], where the complexity of finding optimal routes is mitigated by considering only routing forests, i.e., unions of disjoint trees rooted at the gateway nodes. However, tree-based structures are less reliable to link failures than mesh-based structures. The authors in [KGDB07] propose to transform the original network graph into an edge graph, where multiple links are aggregated into segments. This approach results into a reduction in the number of possible paths to check for feasibility, depending on the adopted segment size.

In our routing scheme we adopt a simpler approach by constructing a *routing mesh* from each mesh node to the available gateways. More precisely, for each mesh node i we compute the minimum cost paths towards each gateway j (with $j \in \mathcal{G}_r \cup \mathcal{G}_p$). Note that minimum cost path can be efficiently computed in a loop-free manner using Dijkstra and Bellman-Ford algorithms if the routing metric is *isotonic* [YWK05]. Thus, the number of paths to check for feasibility grows linearly with the number of gateways and mesh nodes. The penalty we pay for this simplicity is that occasionally the routing process may not find a feasible route although it exists.

To facilitate the description of the LARS solution, Algorithm 1 shows the pseudo-code that it is used to determine a feasible route for a newly arrived traffic flow. Let us assume that at time t a set $\mathcal{F}^{(k)}$ of k upstream Internet flows, $f^{(1)}, f^{(2)}, \ldots, f^{(k)}$ has been already admitted in the network. Each of these flow has demanded a certain average bandwidth to satisfy its QoS requirements. Formally, let $b^{(i)}$ denote the mean arrival rate of packet generated by the i-th flow in the set $\mathcal{F}^{(k)}$. Thus, the overall offered load $\lambda^{e^{(k)}}$ is equal to $\lambda^{e^{(k)}} = \sum_{i=1}^{k} b^{(k)}$, and $\mathbf{P^e}^{(k)}$ is the related throughput allocation. Finally, let $\mathbf{R}_{\mathbf{fwd}}^{(k)}$ be the equivalent routing matrix used to forward these flows.

Now, let as assume that at time $t+1$ arrives a new flow $f^{(k+1)}$, originated at mesh node $s \in V$, which demands a bandwidth equal to $b^{(k+1)}$. We denote with $\mathcal{Q}_s$ the set of gateways that node s can use to access the Internet. Without loss of generality, we can assume that $\mathcal{Q}_s$ contains all the gateways deployed in the network. Then, we update the throughout allocation vector to include the additional demand of this flow. The core of LARS scheme is the selection of the *best* gateway for s in the set $\mathcal{Q}_s$ of potential Internet gateways. To this end, LARS algorithm finds the gateway g in the set $\mathcal{Q}_s$ that is at the least distance from node s. The minimum cost path between s and g, denoted with $path_{s,g}$, is computed using the $\mathsf{MinimumCostPath}(s, g, G(Q, L))$. Then, the routing matrix

Algorithm 1: Pseudo-code of the LARS algorithm.

Input: $G(Q, L)$, $\mathcal{F}^{(k)}$, $\mathbf{P}^{e^{(k)}}$, $\mathbf{R}_{\mathbf{fwd}}^{(k)}$, and $f^{(k+1)}$.
Output: stable, $\mathbf{P}^{e^{(k+1)}}$, $\mathbf{R}_{\mathbf{fwd}}^{(k+1)}$.

```
1  stable ← true ;
2  admitted ← false ;
3  λ̄^e(k+1) ← λ^e(k) + b^(k+1) ;
4  P̄^e(k+1) ← Update(P^e(k), f^(k+1)) ;
5  Q_s ← G_r ∪ G_p ;
6  while (Q_s ≠ ∅)  or  !admitted  do
7      g ← ExtractClosest(v, Q_s) ;
8      path_{s,g} ← MinimumCostPath(s, g, G(Q, L)) ;
9      R̄_fwd^(k+1) ← Update(path_{s,g}, R_fwd^(k)) ;
10     stable ← IsStable(λ̄^e(k+1), P̄^e(k+1), R̄_fwd^(k+1)) ;
11     if stable then
12         Consolidate(λ̄^e(k+1), P̄^e(k+1), R̄_fwd^(k+1)) ;
13         F^(k+1) ← F^(k) ∪ f^(k+1) ;
14         admitted ← true ;
15     else
16         Q_s ← Q_s \ {g} ;
17     end
18 end
```

is updated assuming the new flow is routed on $path_{s,g}$. Finally, the algorithm checks if the *tentative* forwarding matrix $\mathbf{R}_{\mathbf{fwd}}^{(k+1)}$ obtained after adding this new flow from s to g generates a bottleneck in the mesh network. To this end, the analytical framework developed in Section 2.3 is used to perform a *stability test* that checks the feasibility of the throughput allocation. If the stability test is positive, the algorithm confirms the flow allocation using the `Consolidate` function. On the other hand, if gateway g is a bottleneck it can not be used as an egress point to the Internet for this new flow and it is removed from the set $\mathcal{Q}_s$. If there are still available gateways in the set $\mathcal{Q}_s$, LARS will repeat the steps in the *while* cycle, testing the feasibility of a new gateway g.

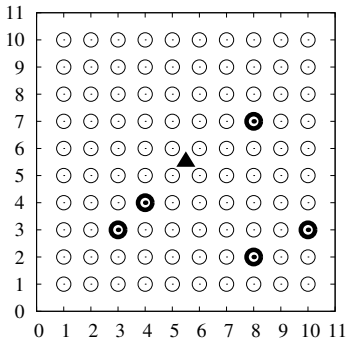
2.5 Performance evaluation

We compare the performance of the LARS scheme an the shortest-path first routing protocol using a contention-aware routing metric using both numerical simulations and preliminary test-bed experiments, as described later in Chapter 4. We take advantage of the different evaluation environments to investigate the proposed approach from various point of views, including practicality and feasibility.

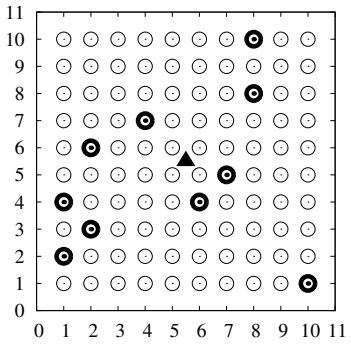
2.5.1 Numerical simulations

The goal of this section is twofold. First of all we compare the analytical results of our model with the outcome of an event-based simulator to validate our analysis. Then, we compare the throughput performance gains of the LARS scheme over a shortest-path first routing algorithm using the IRU metric. For brevity, in the following we refer to this second scheme as *SPF-IRU*. For the sake of clarity, before describing the simulation environment and set-up, we present a brief description of the IRU metric as reported in [YWK06]. Specifically, to capture inter-flow interference, the IRU metric for link l is defined as, $IRU_l = ETT_l \times N_l$, where N_l denotes the number of mesh nodes with which the transmission on link l interferences, while ETT_l [DPZ04] is the expected transmission time on link l. Hence, the IRU cost captures the aggregated channel time that transmissions on link l consume on neighboring nodes, which essentially represents the inter-flow interference. Note that results in [YWK06] show that the IRU metric is able to substantially improve total network throughput in mesh networks by balancing network load.

In the following experiments, the nodes are deployed in a square area of size 1 Km. In the center of the simulated area we place as single provider gateway (i.e., $n_p = 1$), which has a symmetric high-speed Internet connection with $C_p = 1$ Gbps. Then, other 100 nodes (mesh routers and residential gateways) are deployed on a grid layout, with grid points separated by 100m. More precisely, we randomly pick up n_w grid points where we place mesh routers, and in the remaining n_r grid points we place residential gateways ($n = n_w + n_r = 100$). This ensures a sufficient degree of randomness in the locations of the gateways. To simulate residential gateways with low-speed backhaul links, we set $C_r = 5$ Mbps. Finally, we model diverse levels of network heterogeneity by



(a) $n_r/n = 5\%$



(b) $n_r/n = 10\%$

Figure 2.3: Illustrative network topologies. Bold circles represent residential gateways, while filled triangles are provider gateways.

varying the percentage n_r/n of residential gateways over mesh routers. For the sake of clarity, two illustrative network topologies are plotted in Figure 2.3.

The interference in the network is simulated using the Protocol Model, and the transmission range and interference range of each node are fixed and equal to 100m and 200m, respectively. Regarding the MAC protocol, we have implemented the collision-free CSMA-based access scheme described in Section 2.3.1. A more realistic MAC protocol, using practical collision avoidance mechanisms (e.g., 802.11-based backoff algorithm) will be considered in future work. However, to take into account that we abstract away the MAC-layer signaling issue, i.e., a node is instantly informed about the success of its transmissions, we set

Parameter	Values
Area of deployment	1×1 Km^2
Number of mesh routers	95,90,80
Number of provider gateways	1
Number of residential gateways	5,10,20
Wireless transmission rate	30 Mbps
Wired transmission rate	1 Gbps (provider), 5 Mbps (residential)
Transport protocol	UDP
Per-flow offered load	uniform in [100kbps, 200kbps]

Table 2.2: Overview of simulation parameters

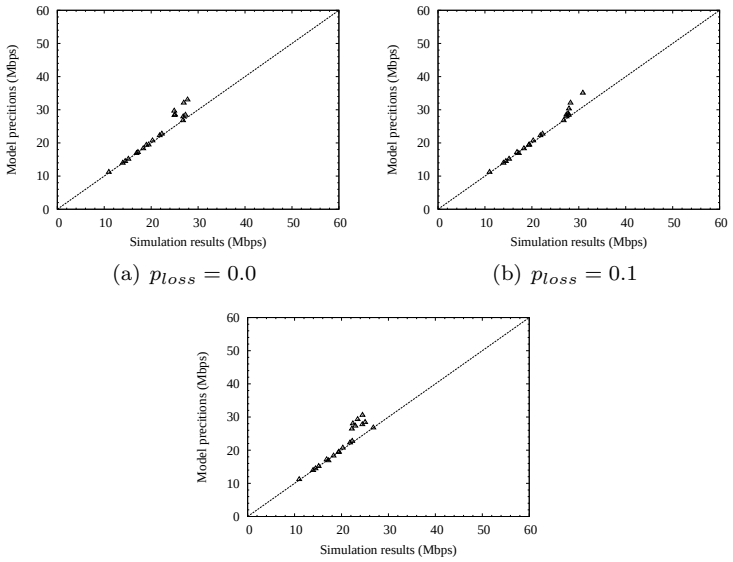
the effective wireless channel bandwidth to $C_w = 30$ Mbps. Finally, to model channel errors to each wireless link is assigned a constant packet loss rate, in accordance with the physical layer model used in Section 2.3.

Regarding the traffic model, in this study we use UDP as the transport protocol for generating data traffic. We consider *upstream* Internet flows established from randomly selected mesh nodes towards the wired infrastructure. Following the notation introduced in Section 2.4, $b^{(k)}$ is the average uplink bandwidth demand of flow $f^{(k)}$. If not otherwise specified, in the following tests $b^{(k)}$is a random value uniformly selected in the range [100kbps, 200kbps], while the inter-packet arrival time is exponentially distributed.

The most important simulation parameters are summarized in Table 2.2.

Capacity estimation

In this section we validate the analysis by comparing the network capacity predicted using our model, and the network capacity measured through simulations in different network scenarios. Without loss of generality, in the following experiments we assume that all the wireless links have the same packet loss rate, say p_{loss}. Following the Definition 2, to compute the network capacity we use randomly generated traffic traces. More precisely, a traffic trace is composed of a large number of independent traffic flows, and the originator of each flow is randomly selected among the mesh nodes. During each simulation run, flows are sequentially injected into the network, and the maximum network capacity is obtained when a new flow cannot be accepted without saturating one of the queues in the network. In the following graphs, we report the results related to the network capacity estimation using the LARS algorithm. The LARS scheme



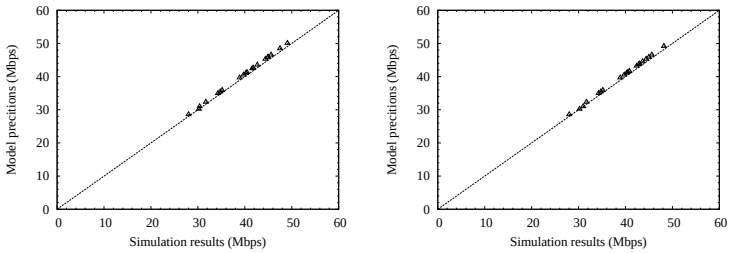
(a) $p_{loss} = 0.0$ (b) $p_{loss} = 0.1$

(c) $p_{loss} = 0.2$

Figure 2.4: Comparison between predicted and measured network capacity using the LARS scheme with $n_r/n = 0.05$.

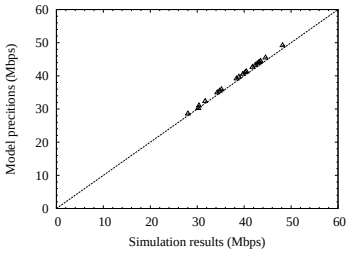
permits to inject in the network a larger number of flows than SPF-IRU (results presented later in Section 2.5.1). This allows us to validate the analysis in conditions of higher contention on both the wired and the wireless channel, which is very helpful to check the accuracy of the analysis in a larger set of network conditions.

Figures 2.4 and Figures 2.5 show a set of scatter plots comparing the network capacity predicted by our model and the one measured through simulations for different numbers of residential gateways and packet loss rates. Twenty different traffic traces are tested per each network topology. The plots show that the correspondence between analytical and simulation results is quite good in most of the considered scenarios. By inspecting the results we discovered that the slight discrepancies between the model predictions and the simulation results occurs primarily when the saturation of a wireless queue is responsible for the limitation of network capacity. This suggests that the model accuracy can be improved by further refining the average value analysis developed to characterize the service times of wireless queues (see Lemma 2).



(a) $p_{loss} = 0.0$

(b) $p_{loss} = 0.1$



(c) $p_{loss} = 0.2$

Figure 2.5: Comparison between predicted and measured network capacity using the LARS scheme with $n_r/n = 0.1$.

A number of additional important observations can be derived from the shown results. First, network capacity is greatly dependent on the location of residential gateways, and the traffic pattern. This is even more evident with SPF-IRU[4] since it uses the gateways' resources in a less efficient way than LARS. In general, the higher the n_r/n value, the higher the network capacity. This is expected because adding more gateways increases the aggregate bandwidth available to access the Internet. Moreover, we can observe that increasing the packet loss rate the network capacity may decrease because the retransmitted packets consume more wireless bandwidth. However, many network scenarios are negligibly affected by an increase of packet loss rates. This is especially true for scenarios characterized by poor network capacity because, in this case, it is likely that the network bottleneck is a particularly disadvantaged gateway. Thus, the effect of a small increase in the number of retransmitted frames is dominated by the inefficiency in the use of the gateways' low-speed backhaul links.

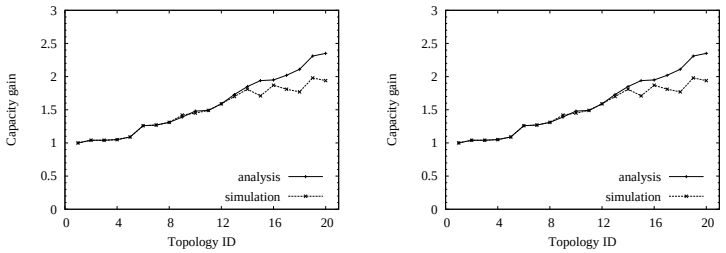
[4]Simulation results related to SPF-IRU are not reported here due to space limitations

Both the above observations motivate the LARS design in which the route selection algorithm takes into account the locations of gateways, as well as the remaining capacity of fixed lines and wireless links.

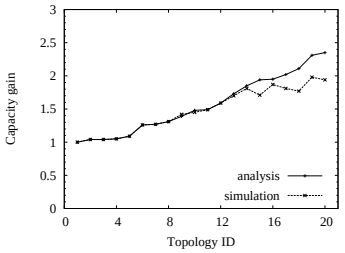
Capacity gain

In this section, we evaluate the efficiency of the LARS solutions in terms of the provided *throughput gain* G, i.e., the ratio between the maximum network capacity they obtain and the one achieved by SPF-IRU routing algorithm. Since network capacity measurements have a high dispersion over different topologies and traffic traces, rather than using mean or standard deviation as comparison metric, we show the throughput gain obtained in each network scenario. More precisely, Figure 2.6 and Figure 2.7 show the performance gain provided by LARS over SPF-IRU for each of the topologies considered in Figure 2.4 and Figure 2.5, respectively. For the sake of clarity, in the graphs we sort the topologies from the maximum network capacity obtained by the shortest-path routing algorithm to the minimum one. Moreover, for each topology we plot the performance gain measured using simulations and the one predicted by the analysis. For these scenarios, LARS significantly outperforms shortest path routing providing a throughput improvement that range from 10% up to 240%. By analyzing more in depth these results we have found out that the performance gain is higher for the most disadvantaged topologies, i.e., for the topologies where the SPF-IRU scheme obtained the lowest performance. This further underlines that the key property of the LARS solution is to anticipate the emergence of network bottleneck and to avoid paths passing though such bottleneck. It is also interesting to observe that higher performance gains are achieved for $n_r/n = 0.05$ than $n_r/n = 0.1$. This can be explained by observing that the higher the density of residential gateways, and the higher the probability that a mesh client has a close gateway. Thus, the inefficiency of a shortest-path first routing algorithm may diminish.

The remarkable throughput improvements provided by LARS can be explained by considering the ability of this algorithm to evenly distribute the network load among all the available gateways. This observation will be further discussed in the following sections, where we use the prototype LARS implementation in a small-scale realistic mesh network to perform a more fine grained study of gateways' resource utilization.



(a) $p_{loss} = 0.0$ (b) $p_{loss} = 0.1$



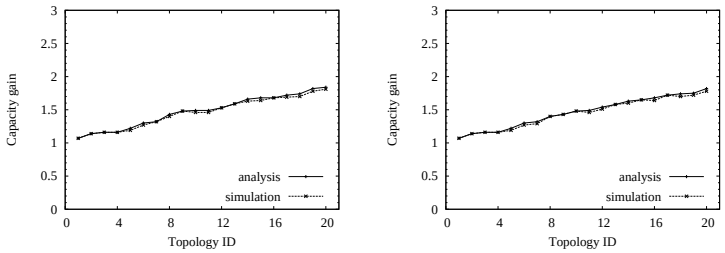
(c) $p_{loss} = 0.2$

Figure 2.6: Capacity gain of LARS over SPF-IRU for $n_r/n = 0.05$. The topologies in each graph are ordered from the one with maximum network capacity to the one with minimum network capacity.

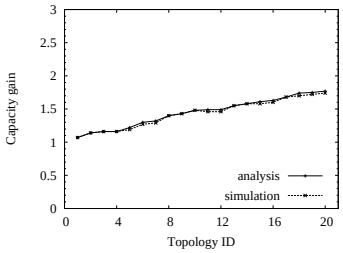
2.6 Conclusions

Differently from other studies on WMNs, in this chapter we have considered heterogeneous WMNs where gateways' backhaul links may have various speeds. Focusing on this scenario, we have developed a queuing network model to analyze the network capacity as a function of several system parameters, including locations of gateways, traffic patterns, link bandwidths and packet loss rates. By exploiting this predictive tool, we have designed LARS, a load-aware route and gateway selection algorithm that improves the network capacity by ensuring a more balanced utilization of the network and gateways' resources. Using simulations and a prototype implementation in a realistic small-scale mesh network, we have shown that the LARS scheme significantly outperforms the shortest path routing using a contention-aware routing metric, providing up to 240% throughput improvement in some network scenarios.

Although our analysis considers packet losses due to channel errors, we have used an idealized CSMA-based MAC protocol, which primarily captures



(a) $p_{loss} = 0.0$ (b) $p_{loss} = 0.1$



(c) $p_{loss} = 0.2$

Figure 2.7: Capacity gain of LARS over SPF-IRU for $n_r/n = 0.1$. The topologies in each graph are ordered from the one with maximum network capacity to the one with minimum network capacity.

location-dependent contention issues due to differences in the number of contending nodes at both endpoints of each communication link. Although this basic CSMA model can provide accurate expressions, the extension of our analysis to a real MAC protocol implementing practical collision avoidance mechanisms is a challenge that needs to be addressed. Furthermore, traffic flows can express their QoS demands using various metrics. For instance, end-to-end delay may be a more important metric to use for real-time traffic. However, jointly considering capacity and end-to-end delay constraints in the routing process is a complex issue. Finally, to integrate specific fairness models in the gateway selection is also an interesting research direction.

Chapter 3

Theoretical Performance Evaluation

3.1 Introduction

Wireless mesh networks (WMNs) are increasingly deployed to provide cost-effective ubiquitous access to the Internet [KSK04]. Normally, in WMNs a set of stationary wireless mesh routers form a multi-hop wireless backbone, where a small subset of these routers act as *gateways* being connected to the Internet through high-speed fixed lines [BCG05]. Typically, it is also assumed that the link capacity in the Internet is much larger than the wireless channel capacity. However, this vision is rapidly changing. Real-world mesh networks are frequently used to share a potentially large number of low-speed Internet connections (i.e., DSL fixed lines) available at the customers' premises. Examples of such networks are Meraki-based deployments in urban areas [Mer], or the Ozone's network in Paris, which is composed of 400 mesh routers, most of them using standard DSL links as Internet backhaul, while only ten gateways are provided with an ISP-owned fiber link [Ozo]. In a broader sense, wireless mesh networks are evolving into a *converged infrastructure* used to share the Internet connectivity of sparsely deployed fixed lines with *heterogeneous capacity*, ranging from ISP-owned broadband links to subscriber-owned low-speed connections [SACB08].

WMNsMesh being primarily used for Internet access. Therefore, both traf-

fic routing and Internet gateway selection play a crucial role in determining the overall network performance and in ensuring optimal utilization of the mesh infrastructure [ZWR08]. Indeed, depending on the location of the mesh nodes and the gateways, some of the mesh nodes may obtain substantially lower throughput than others. Similarly, if many mesh nodes select the same gateway as egress (ingress) point to (from) the Internet, congestion may increase excessively on the wireless channel, or the Internet connection of the gateway can get overloaded. This problem is particularly relevant in the heterogeneous WMNs considered in this study, because low-speed Internet gateways may easily become a *bottleneck*, limiting the achievable capacity of the entire network, while most of the available studies have assumed that bottlenecks appear only in the wireless network. Finally, a load-unaware gateway selection can lead to an unbalanced utilization of network resources.

To improve load balancing and increase capacity of WMNs, previous studies suggested to use balanced tree structures rooted at the gateways, and to route the traffic along the tree paths. For instance, a heuristic algorithm for calculating load-balanced shortest path trees taking into account flow load is proposed in [HCC05]. In [BHK07], approximated solutions are defined for calculating load-balanced trees that allocate the same bandwidth to all the nodes, using both single-path and multi-path approaches. An alternative strategy is proposed in [MBLD07], where the complexity of finding optimal routes is mitigated by considering only delay optimal routing forests, i.e., unions of disjoint trees routed at the gateway nodes. However, tree-based routing structures are less reliable to link failures than mesh-based structures. Furthermore, the admission of a new flow usually triggers complex reconfiguration procedures for the entire tree in order to maintain the load balancing properties.

A simpler approach to improve network performance is to define routing metrics for shortest-path first (*SPF*) routing that determine high-throughput paths and/or facilitate load balancing. Initially proposed metrics (e.g., ETX [DCABM03] and ETT [DPZ04]), focused only on link characteristics (e.g., frame loss and transmission rates), thus they cannot balance the load. Recent studies proposed to introduce in the metric computation estimates of inter-flow and intra-flow interference (e.g., IRU [YWK06]), location-dependent contention (e.g., CATT [GS08], ETP [MBLD07]), or load-dependent cost (e.g., the queue length in WCETT-LB [MD07], or the number of per-link admitted flows in

LAETT [ALCR08]). Although these metrics have been demonstrated to work quite well in mesh networks, and to provide higher throughput performance than simple hop count, they are completely unaware of the available resources at the gateways. On the contrary, significant performance improvements may be obtained by considering residual capacity of the links between the gateways and the Internet, as well as the load distribution, when routing traffic flows. However, there is a complex interdependence between the way traffic flows are routed in the network and the utilization of network resources, which makes defining simple heuristics to estimate the remaining capacity of a network path or a gateway quite difficult.

To address the above problems, in this chapter we make the following contributions. We develop a *multi-class queuing network model* for heterogeneous WMNs and used it to determine if a given allocation of flows on a set of network paths is feasible. More precisely, our model characterizes the network performance as a function of the traffic pattern, the distribution of gateways and mesh routers in the WMN, the heterogeneity of link capacities, as well as the location-dependent contention on the wireless channel; then, given the routing strategy used to allocate the flow demands on the network paths, we exploit our model to establish if the resulting flow allocation does not violate the network capacity constraints. Moreover, we also mathematically characterize the average packet end-to-end delay, defined as the average time taken by a packet to reach the Internet after it is generated. To validate our modeling methodology, in this study we consider a basic CSMA-based MAC protocol, which implements an idealized collision avoidance mechanism that can always detect if the medium is busy or free before a transmission attempt. The primary goal of this study is not to accurately model the performance of specific standard MAC protocols, but to investigate the impact on system performance of the location-dependent contention inherent to multi-hop environments, due to differences in the number of contending nodes at both endpoints of each communication link ([BCP11]).

It is important to point out that several previous studies have proposed to use queuing models to investigate system performance of CSMA-based ad hoc networks. However most of these studies have applied queuing theory to the analysis of *single-hop* ad hoc networks [ASS03, OM04, TS08]. To the best of our knowledge, in the literature a few examples exist which deal with

the multi-hop case. In [BA09], the authors model random access multi-hop wireless networks as open $GI/G/1$ queuing networks to analyze the average end-to-end delay and maximum achievable per-node throughput. However, the formulation proposed in [BA09] can be applied only to random networks, and it does not incorporate flow-level behaviors. Our objective is different from [BA09], because we consider arbitrary topologies and routing strategies, and we focus on per-flow performance. In our paper [BCP09a] we have developed a single-class queuing model to analyze the network capacity of heterogeneous WMNs, however, the analysis in [BCP09a] is valid only for upstream Internet traffic, which is a somehow unrealistic traffic model for typical WMNs. To go further, in this chapter we extend our previous analysis to incorporate generic traffic distributions, which motivates the use of a novel modeling methodology based on multi-class queuing networks.

Guided by our analysis, in this chapter we propose a *Capacity-Aware Route Selection* algorithm (*CARS*), which integrates traffic routing with gateway selection. Instead of using SPF routing, CARS scheme determines the set of optimal routes from the mesh node that originates a new flow, and the available gateways. It is important to note that any cost function can be used to determine the initial set of optimal network paths. However, isotonic routing metrics are preferable because they permit efficient and loop-free computation of minimum cost paths [YWK05]. Then, CARS allocates the new flow to the best network path that has enough residual capacity (as predicted by our model) to satisfy its bandwidth demands. As a result, a mesh node can discard paths or gateways that cannot accept additional demands. This facilitates load balancing in the network by avoiding the rapid exhaustion of the link capacity of disadvantaged mesh nodes or gateways, leading to a more efficient utilization of both wireless and wired network resources. Through simulations performed in network scenarios with different numbers of gateways and link capacities, we show that CARS scheme results in significant throughput improvements over SPF routing using IRU metric [YWK06], which captures only inter-flow interference (i.e., mutual interference between adjacent flows). Furthermore, the simulation results confirm the accuracy of the proposed modeling methodology.

The remaining of this chapter is organized as follows. Section 3.2 introduces the network model. In Section 3.3 we develop the capacity analysis. Section 3.4 describes the proposed CARS algorithm. In Section 3.5 we present simulation

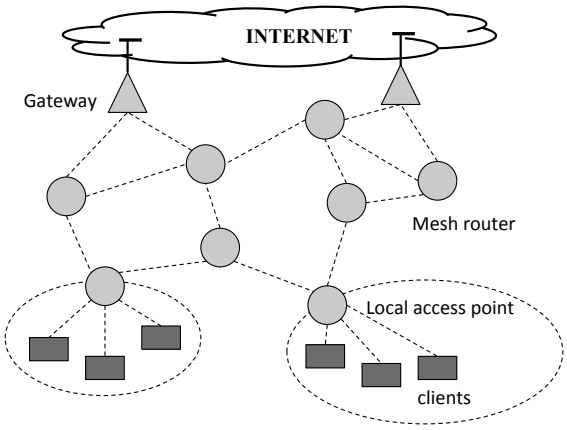


Figure 3.1: General architecture of WMNs.

results to validate the analysis, and to compare CARS performance against two other routing algorithms. Finally, conclusions and future work are discussed in Section 3.6.

3.2 Network Model

In this work we are concerned with *heterogeneous* wireless mesh networks (WMNs) as illustrated in Figure 3.1., which consist of fixed wireless routers, and mobile or semi-static end-user stations, also named *mesh clients*. Wireless mesh routers are equipped with local access points, which aggregate the traffic from mesh clients that are associated with them. Thus, mesh routers constitute a wireless mesh backbone providing a wireless infrastructure for mesh clients. Some of the mesh routers have also a physical link to a wired backhaul network, and they serve as *gateways* between the WMN and the global Internet. All the resources residing on the wired network (e.g., files or application servers) can be accessed through any of the available gateways. Henceforth, mesh routers and mesh gateways are collectively termed *mesh nodes*.

Differently from previous studies, which generally assume a limited number of mesh gateways in the network, as well as no capacity constraints on the gateways, we consider different classes of mesh gateways. More precisely, we consider mesh gateways connected to the wired network using low-speed links,

and mesh gateways connected to the wired network using very high-speed links. As discussed in Section 3.1, the former category of gateways can model mesh routers deployed at the customers' premises, which are generally connected to the Internet through residential access lines (e.g., DSL or cable lines), whereas the latter category can model mesh gateways located at the provider's premises, which have a high-speed connection to the Internet (e.g., fiber or point-to-point high-capacity wireless links). Hereafter, we will refer to the first type of gateways as *residential gateways*, and to the second one as *provider gateways*.

A second relevant difference between the network scenario targeted in this work and the architecture of WMNs generally adopted in previous studies is that we relax the assumption on symmetric capacity for fixed access lines. More precisely, the distinguishing characteristic of DSL-based technologies is that the upload speed is generally lower than the download speed. This asymmetry in the bandwidth for the two transmission directions may have a significant impact on the overall network capacity if not properly taken into account during the routing process. To the best of our knowledge, previous studies have only considered the case of wired communications technologies with symmetric bandwidth for both directions.

To represent the above network model, let us introduce $\mathcal{G}_r$ as the set of residential gateways, $\mathcal{G}_p$ as the set of provider gateways, and $\mathcal{M}$ as the set of mesh routers without a physical connection to the wired infrastructure. Let n_w, n_r and n_p be the cardinality of $\mathcal{M}$, $\mathcal{G}_r$, and $\mathcal{G}_p$ sets, respectively, with $n = n_w + n_r + n_p$. Then, the network is modeled using a mixed graph $G = (V \cup \{a\}, E_w, E_g)$, where the graph vertices V ($|V| = n$) represent the mesh nodes (i.e., $V = \mathcal{G}_r \cup \mathcal{G}_p \cup \mathcal{M}$), and a is a virtual node that corresponds to the fixed infrastructure. We denote by E_w the set of undirected edges representing the wireless links between mesh nodes, while E_g is the set of directed edges representing the network links between the gateways and the infrastructure. The neighborhood of node $i \in V$, denoted by $N(i)$, is the set of nodes to which node i is physically connected. If node i is a gateway, the virtual node a is included in the neighborhood of i.

Each link $e \in E_w$ has a capacity (bit rate) C_w for both directions, whereas each link $e \in E_g$ has a capacity that depends on the direction of the communication, as well as on the gateway class. More precisely, we assume that a link $e \in E_g$ from a residential gateway $i \in \mathcal{G}_r$ to the wired infrastructure a has

capacity C_r^u, and from a to gateway i has capacity C_r^d, respectively. Similarly, we assume that a link $e \in E_g$ from a provider gateway $i \in \mathcal{G}_p$ to the wired infrastructure a has capacity C_p^u, and from a to gateway i has capacity C_p^d, respectively. It is important to note that in this work we are primarily concerned with *Internet traffic*; in other words, we assume that user traffic is either originated from or is destined to the fixed infrastructure.

Concerning the physical layer model of the wireless communication channel, we assume that the *transmission range* of each station is fixed and equal to r. Moreover, a pair of mesh nodes that are within each other's *interference range* may interfere with each other's transmissions, even if they cannot directly communicate. To model the interference relationships between contending mesh nodes, we use the Protocol Model as in [BA09, BCP09a]. In other words, a transmission from mesh node i to mesh node j, with $i, j \in V$, is successful if the following conditions are satisfied: 1) $|i–j| \leq r$, i.e., the euclidean distance between nodes i and j is lower or equal to r; and 2) for every other transmitting node k, $|k-i| \geq (1+\Delta)\cdot r$, where Δ is a fixed positive constant that represents a guard zone in the Protocol Model. Alternatively, $(1+\Delta)\cdot r$ can be interpreted as the *interference range*, i.e., the largest distance at which a sender can interfere with an ongoing transmission. For brevity, let us denote with $I(i)$ the set of interferers for node i, i.e., $j \in I(i)$ if $|j-i| \leq (1+\Delta)\cdot r$.

To coordinate simultaneous transmissions of interfering nodes we employ a basic CSMA-based MAC protocol, which implements an idealized collision-avoidance mechanism. More precisely, we assume that each node has an instantaneous knowledge of the communication state (i.e., idle, receiving or transmitting) of other interfering nodes, so as to ensure that it starts transmitting only when both above conditions can be satisfied. This is somehow equivalent to determine a collision-free random transmission schedule among contending nodes. This assumption might be considered restrictive, especially because we neglect the detailed protocol implementation of collision avoidance and resolution mechanisms, such as 802.11-like backoff schemes, as well as the impact of *hidden nodes* on the link capacity. However, though in a simplified form, the considered MAC scheme captures some of most important aspects of location-dependent contention inherent to multi-hop environments, which is due to differences in the number of contending nodes at both endpoints of each communication link. In other words, in this study we are more concerned on

variable	definition
$\mu^r_{i,l}$	The service rate of the queue l of the station i for packets of the rth class
$\lambda^r_{e;i}$	The arrival rate from mesh clients to station i for packets of the rth class
$\lambda^r_{a;i}$	The arrival rate from the wired infrastructure to station i for packets of the rth class
$\lambda^r_{o;i,l}$	The arrival rate from *outside* (i.e., mesh clients and wired infrastructure) to queue l of station i for packets of the rth class
λ_o	The *overall* arrival rate from outside to the mesh network
$p^r_{o;i,l}$	The probability that a packet from outside the mesh network enters queue l of station i as a job of the rth class
$\lambda^r_{f;i}$	The arrival rate from node i's neighboring nodes to station i for packets of the rth class
$\lambda^r_{i,l}$	The overall arrival rate of packet of the rth class r to queue l of station i
$p^r_{i,l;j,s}$	The probability that a packet of the rth class at queue l of node i is transferred at queue s of node j
$\rho^r_{i,l}$	The utilization of the queue l of the node i with respect to jobs of the rth class

Table 3.1: Model notation

modeling the link capacity degradation due to location-dependent contention, rather than precisely incorporating in the analysis all the features of the IEEE 802.11 MAC protocol. The extension of our modeling framework to take into account hidden nodes is part of our future work.

3.3 Queuing Analysis

In this section we develop a queuing-based analysis of the WMN architecture described above. We use this mathematical framework to derive expressions for the network capacity and the average end-to-end packet delay. Fro the sake of clarity, Table 3.1 summarizes the key notations used throughout the analysis.

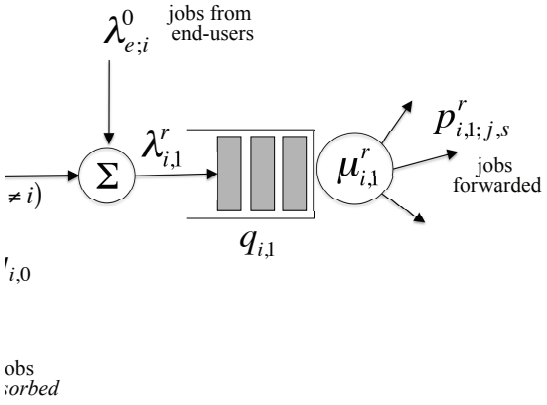
3.3.1 Queuing Network Model

The first step of our analysis is to define a representation of an heterogeneous WMN, i.e. $G(V \cup \{a\}, E_w, E_g)$, through an equivalent *multi-class* queuing network $G'(Q, L)$, where Q is the set of queuing systems in the network, for brevity *stations*, which model the real mesh nodes, and L is the set of connections between stations. First of all, it is important to discuss the reasoning behind using a multi-class queuing network. Intuitively, jobs in the queuing network

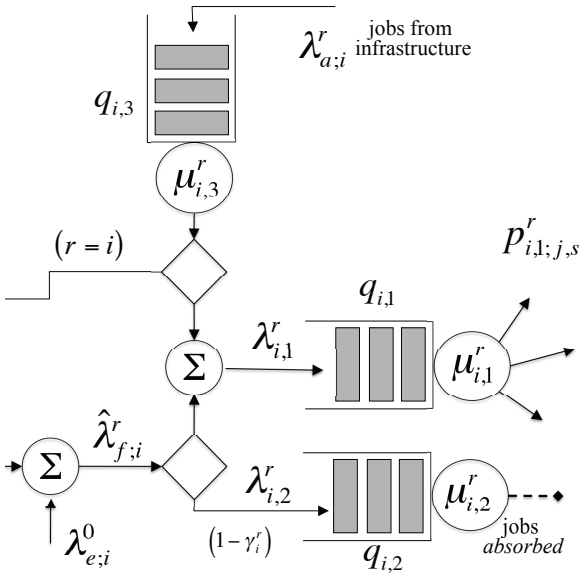
represent packets in the physical network[1]. However, each packet may have a different destination (either a mesh client associated to a mesh node or a device located in the global Internet), and may belong to traffic flows with different bandwidth demands. Hence, multiple job classes is a flexible technique to separately characterize the properties of different traffic flows. More precisely, the packets of a flow originated from a device in the wired infrastructure, which enters the WMN from gateway i ($i \in \mathcal{G}_r \cup \mathcal{G}_p$), and have a client associate to mesh node r ($r \in V$) as destination, are modeled as jobs of class r. We define the average arrival rate of such flow as $\lambda^r_{a;i}$. On the other hand, the packets of a flow originated from a mesh client associated to mesh node i ($i \in V$), and heading to a device in the wired infrastructure (i.e., to the virtual node a), are modeled as jobs of class 0. The average packet arrival rate of such flows is defined as $\lambda^r_{e;i}$. Being n the number of mesh nodes, the number of classes needed to model all the possible traffic flow destinations is $R = n+1$. It is also worth pointing out that job classes can differ in their service times and in their routing probabilities, which ensures a high modeling flexibility.

Based on our analogy between the physical network and an equivalent queuing network, each mesh node $i \in V$ is modeled through an *equivalent* queuing system $i \in Q$. In general, this equivalent queuing station may include several queues to capture the most important features of the multiple network interfaces (for both wired and wireless technologies), which a mesh node is equipped with. For brevity, let $q(i)$ be the number of queues in station i. It is intuitive to note that, being the WMN composed of two classes of nodes, gateway and non-gateway nodes, at least two different queuing station models should be specified for the analysis. For ease of explanation, Figure 3.2 exemplifies the structure of the queuing stations used to model mesh nodes. Specifically, Figure 3.2(a) illustrates a station modeling a wireless mesh router i ($i \in \mathcal{M}$) not connected to the wired infrastructure. This mesh node aggregates the traffic originated from the mesh clients associated with it. Since we do not consider communications between mesh clients but only between mesh clients and the Internet, it holds that $\lambda^r_{e;i} = 0$ for $r \neq 0$. In addition to locally generated jobs, station i receives jobs forwarded by neighboring mesh nodes, with average arrival rate $\lambda^r_{f;i}$. In case $r = i$, namely, the jobs are intended for a mesh client associated to mesh node i, the traffic is redirected to a queue, say $q_{i,0}$, modeling the transmissions

[1] In the following, the terms job and packet are used equivalently.



(a) Non-gateway mesh node



(b) Gateway mesh node

Figure 3.2: Equivalent queuing stations used to model mesh nodes.

from the local access point to the mesh clients. After completing the service at queue $q_{i,0}$, the jobs are absorbed and leave the network. Note that we assume there are no capacity constraints at the local access point, which implies that jobs are instantaneously served at queue $q_{i,0}$. On the contrary, if $r \neq i$, the jobs should be forwarded to another node, and they are redirected to the queue, say $q_{i,1}$, which models the transmissions on the wireless channel. Hence, the total arrival rate at queue $q_{i,1}$ is $\lambda^r_{i,1} = \lambda^r_{f;i} + \lambda^r_{e;i}$ (for $r \neq i$). Finally, after being served from queue $q_{i,1}$, a job of class r will be transferred to the s-th queue of the j-th station with probability $p^r_{i,1;j,s}$. In other words, $p^r_{i,1;j,s}$ is the *routing probability*, which models the packet forwarding process implemented in the physical network as a stochastic process.

Figure 3.2(b) illustrates a station modeling a gateway node i ($i \in \mathcal{G}_r \cup \mathcal{G}_p$). In this case, the internal structure of the queuing station is more complicated because we need four queues, say $q_{i,0}$, $q_{i,1}$, $q_{i,2}$ and $q_{i,3}$ (i.e., $q(i) = 4$), to model wireless transmissions to associated mesh clients, wireless transmissions to neighboring mesh nodes, uplink wired transmissions and downlink wired transmissions, respectively. Specifically, each gateway i may receive packets from the wired infrastructure, which have mesh node r as destination. These packets will be routed through queue $q_{i,3}$, and we denote with $\lambda^r_{a;i}$ their average arrival rate. Note that if $r = i$, then the mesh clients associated to this gateway are the packet destination, and after being served at queue $q_{i,3}$ the jobs are redirected to queue $q_{i,0}$. On the other hand, gateway i may receive packets forwarded by neighboring mesh nodes, with average arrival rate $\lambda^r_{f;i}$, as well as packets generated by associated mesh clients, with average arrival rate $\lambda^r_{e;i}$. In case $r = i$, the received jobs are intended for the mesh clients associated to mesh gateway i, and the traffic is redirected to queue $q_{i,0}$. On the other hand, if $r = 0$, the packet destination is the wired infrastructure (i.e., the virtual node a) and the packet should be routed through queue $q_{i,2}$, which models the wired access line in the uplink direction. However, a residential gateway may have a low-speed upstream connection to the Internet, which rapidly becomes a bottleneck as the traffic received on the wireless interface builds up, limiting the achievable capacity of the whole mesh network. To make this limitation less severe, the residential gateway may take advantage of the available wireless bandwidth to behave as a relay node, and further forwarding the traffic to one of its neighbors, which may be less congested, or closer to a provider

gateway. To model this capability we introduce the *re-forwarding* probability γ_i^r. Specifically, a job of class r received by gateway i is routed through the wireless queue $q_{i,1}$ with probability γ_i^r, or directly through the upstream wired queue $q_{i,2}$ with probability $(1-\gamma_i^r)$. The design of the γ_i^r function depends on the routing and resource allocation strategies implemented in the mesh network. Note that in our model $\gamma_i^r = 1$ for $r \neq 0$ because communications between mesh gateways through the wired infrastructure are not permitted. In other words queue $q_{i,2}$ can be used only by upstream flows to access the Internet (i.e., virtual node a).

For simplicity, in this study we assume that all queues have infinite size and serve packets according to a FCFS discipline.

3.3.2 Feasible Network Throughput

In this section we develop the analysis to determine if a given throughput allocation in G is *feasible*. Before formally defining when a throughput allocation is feasible, and describing our analytical methodology, it is useful to introduce some notation.

Let us denote with λ_o the overall arrival rate of jobs from *outside* to the mesh network. Furthermore, let $p^r_{o;i,l}$ be the probability that a job from outside the network enters the l-th queue of the i-th station as a job of the r-th class. This yields that the arrival rate from outside to queue l of station i (i.e., $q_{i,l}$) for class r jobs is $\lambda^r_{o;i,l} = \lambda_o \cdot p^r_{o;i,l}$. Then, we define the *probability matrix of external arrivals* as $\mathbf{P_o} = \{p^r_{o;i,l}, i \in Q, l \in [1, q(i)], r \in [1, R]\}$. Note that this notation conforms to the network model formulated in Section 3.3.1. For instance, for gateway i it holds that $\lambda^r_{o;i,3} = \lambda_o \cdot p^r_{o;i,3} = \lambda^r_{a;i,3}$. Thus, from a given value of λ_o it immediately follows that $p^r_{o;i,3} = \lambda^r_{a;i,3}/\lambda_o$. Similar reasoning can be applied to derive the probability $p^r_{o;i,l}$ for the internal queues of each mesh node.

Definition 3. Throughput allocation. *A throughput allocation for G is any assignment for the rate λ_o and the probability matrix $\mathbf{P_o}$.*

In Section 3.3.1 we have introduced the routing probability $p^r_{i,l;j,s}$ defined as the probability that a job of class r is transferred from the queue l of station i (i.e., $q_{i,l}$) to queue s of station j (i.e., $q_{j,s}$). Following the equivalency between the real WMN and the queuing network, the $p^r_{i,l;j,s}$ value expresses the probability that mesh node i selects mesh node j as next-hop to reach destination

r. The queue indexes are used to specify if the packet is transmitted using the wireless links or the wired fixed lines. Then, we define the network *routing matrix* as $\mathbf{R_{fwd}} = \{p^r_{i,l;j,s}, i, j \in Q, l \in [1, q(i)], s \in [1, q(j)], r \in [1, R]\}$, which is the probabilistic representation of the underlying routing process. It is intuitive to observe that the specific formulation of the routing matrix depends on the routing algorithm used in the WMN, as well as the network topology.

The following definition specifies when a throughput allocation is feasible.

Definition 4. Feasible throughput allocation. *Given a routing matrix* $\mathbf{R_{fwd}}$*, a throughput allocation is feasible if every queue* $q_{i,l}$ *(*$i \in Q$ *and* $l \in [1, q(i)]$*) has a bounded time-average number of packets. This is equivalent to state that arrival process at queue* $q_{i,l}$ *is admissible with rate* $\lambda^r_{o;i,l}$*.*

From a mathematical point of view, Definition 4 implies that a throughput allocation is feasible if all the queues in the system are stable, i.e., the number of jobs waiting in queue does not grow indefinitely. From a more practical perspective, to determine if a throughput allocation is feasible is equivalent to verify that the allocation of a given set of flows on a given set of network paths does not violate the network capacity constraints.

To verify the queue stability we have to compute the queue's *utilization factor* [BGdMT06]. From elementary queuing theory this requires the evaluation of the first moments of the packet arrival and service processes at each queue of the network. Specifically, under the assumption that service rates are independent of the queue load, the utilization $\rho^r_{i,l}$ of the queue l at node i (i.e., $q_{i,l}$) with respect to jobs of the r-th class is

$$\rho^r_{i,l} = \frac{\lambda^r_{i,l}}{\mu^r_{i,l}} \,, \tag{3.1}$$

where $\lambda^r_{i,l}$ is the average packet arrival rate of class r jobs at queue $q_{i,l}$, and $\mu^r_{i,l}$ is the corresponding average service rate. Then, the overall utilization of queue $q_{i,l}$ can be computed as:

$$\rho_{i,l} = \sum_{r=0}^{R} \rho^r_{i,l} \,. \tag{3.2}$$

By definition, an infinite-size queue is stable if and only if $\rho_{i,l} < 1$.

The average rate $\lambda^r_{i,l}$ can be computed from λ_o, $\mathbf{P_o}$ and $\mathbf{R_{fwd}}$ by solving

the following system of linear equations:

$$\lambda_{i,l}^r = \lambda_o \cdot p_{o;i,l}^r + \sum_{j=1}^{n} \sum_{s=1}^{q(j)} \lambda_{j,s}^r p_{j,s;i,l}^r \text{ for } i \in Q, l \in [1, q(i)] \,, \tag{3.3}$$

obtained by writing the flow balance condition at each queue of the system.

The average service rates for the queues modeling transmissions on either uplink or downlink wired links can be easily derived by observing that in switched communication technologies there is no contention. Hence, average service times depend only on the nominal link capacity and the packet size. Then, under the assumption that the packet size is constant and equal to P bits, it holds that

$$\mu_{i,l}^r = \begin{cases} P/C_r^u & i \in \mathcal{G}_r, l = 2 \\ P/C_r^d & i \in \mathcal{G}_r, l = 3 \\ P/C_p^u & i \in \mathcal{G}_p, l = 2 \\ P/C_p^d & i \in \mathcal{G}_p, l = 3 \end{cases} . \tag{3.4}$$

On the other hand, the derivation of the average service rate for the queues modeling transmissions on the wireless channel is more involved because it is necessary to take into account the location-dependent contention, the distributions of active queues (i.e., queues with at least a packet to serve) and the channel access coordination procedures implemented by the MAC protocol. Several stochastic models have been developed to analyze the access delays of CSMA-based MAC protocols used in multi-hop environments. Recall from Section 3.2 that in this work we consider a basic collision-free CSMA-based MAC protocol, and we assume that each mesh node has an instantaneous knowledge of the communication state of other interfering nodes. Then, following the footprints of [GCL06] and our previous work [BCP09a], we can model the impact on the channel access of location-dependent contention by employing an *average value analysis*, and considering only the long-term fraction of time each mesh node spends in one of three potential states: transmission state, receiving state, and idle state. This modeling approach will lead to a mathematically manageable, but still reasonable accurate, analysis.

To compute the $\mu_{i,1}^r$ parameter[2] we analyze the channel events during the

[2]Recall that $q_{i,1}$ refers to the queue at station i that models transmissions on the wireless

$X^r_{i,1}$ period, defined as the interval from the time instant a class r job reaches the head of queue $q_{i,1}$ to the time instant in which it is transferred to the next-hop station. Then, it holds that $\mu^r_{i,1}=1/E[X^r_{i,1}]$, where $E[\cdot]$ is the expectation operator. To simplify the derivation of the $E[X^r_{i,1}]$ expression we condition to the possible destinations of a class r job served at queue $q_{i,1}$. Specifically, owing to the conditional expectation theory we can write that

$$E[X^r_{i,1}] = \sum_{j=1}^{n} \sum_{s=0}^{\min\{2,q(j)\}} E[X^r_{i,1;j,s}] \cdot p^r_{i,1;j,s} \ , \tag{3.5}$$

where $X^r_{i,1;j,s}$ is the time needed to transfer a job of class r from queue $q_{i,1}$ to queue $q_{j,s}$. This time will mainly depend on the level of contention around the transmitting station i and the receiving station j, i.e., on the distribution of interfering nodes in the network, as well as on their activity level, i.e., the fraction of time these nodes contend for the channel access.

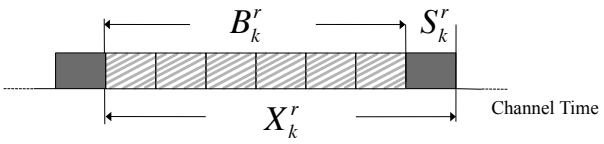


Figure 3.3: Illustrative example of the channel events during the transmission of a class r packet from $q_{i,1}$ to queue $q_{j,s}$.

For ease of explanation Figure 3.3 shows the evolution of channel events during a generic $X^r_{i,1;j,s}$ interval. As illustrated in the diagram, due to the random access scheme the transmission of a packet from queue $q_{i,1}$ to queue $q_{j,1}$ may be preceded by a number $z_{i,1;j,s}$ of transmissions by other contenting stations, which does not depend on the packet class[3]. Let us denote with $E[B_{i,1;j,s}]$ the average period of channel time occupied by other stations' packet transmissions, which precedes the service of the packet at the head of queue $q_{i,1}$, given that this packet is heading to queue $q_{j,s}$. Then, under the assumption of

channel.

[3]In the consider idealized CSMA-based MAC protocol transmission attempts are not preceded by backoff delays.

fixed packet size, it is straightforward to derive that

$$E[B_{i,1;j,s}] = P \cdot E[z_{i,1;j,s}]/C_w. \tag{3.6}$$

This yields to the following expression for the $E[X^r_{i,1;j,s}]$ parameter.

$$E[X^r_{i,1;j,s}] = P \cdot (1 + E[z_{i,1;j,s}])/C_w \ . \tag{3.7}$$

An interesting result of expression (3.7) is that the value of $E[X^r_{i,1;j,s}]$ value does not depend on class r. This can be explained by noting that the impact of class r on the service time is taken into account in formula (3.5) through the per-class routing probabilities. However, it is also intuitive to note that the time needed to transfer a packet on a link from $q_{i,1}$ to $q_{j,s}$ using a random access scheme should not depend on the packet class but only on the contention level around station i and station j.

To derive a closed expression for the $E[z_{i,1;j,s}]$ parameter, the key approximation of our analysis is to assume that station i attempts to transmit a packet to station j immediately after the channel becomes idle again with a constant (state independent) probability equal to $\tau_{i,1;j,s}$. This approximation is commonly adopted when modeling CSMA-based random access schemes in single-cell WLANs, and it also known as *decoupling* approximation [KAMG07]. We observe that it is reasonable to extend this approximation also to the case of multi-hop environments give that we assume a perfectly synchronized MAC. It is important to note that, while in single-hop networks it is generally assumed that all nodes have the same *transmission probability*, in our study the location-dependent contention is modeled by admitting different values of the $\tau_{i,1;j,s}$ probabilities. The decoupling approximation yields that $z_{i,1;j,s}$ is geometrically distributed with parameter $\tau_{i,1;j,s}$, that is

$$Pr\{z_{i,1;j,s} = h\} = (1 - \tau_{i,1;j,s})^h \tau_{i,1;j,s} \ . \tag{3.8}$$

Now, it is straightforward to derive that

$$E[B_{i,1;j,s}] = \frac{(1 - \tau_{i,1;j,s})}{\tau_{i,1;j,s}} \cdot S \ , \tag{3.9}$$

and formula (3.7) can be rewritten as $E[X^r_{i,1;j,s}] = P/(C_w \cdot \tau_{i,1;j,s})$.

The following lemma provides an explicit expression for the transmission probability $\tau_{i,1;j,s}$.

Lemma 3. *For the random access MAC model, under the assumption that reception and transmission events in $G'(Q, L)$ are mutually independent, it holds that*

$$\tau_{i,1;j,s} = \prod_{k \in I(j) \cup \{j\}} [1 - p_{tx}(i)] \cdot \prod_{h \in I(i) \cap I(j)} \left[1 - \frac{p_{rx}(h)}{1 - p_{tx}(h)} \cdot \omega_{h;j}\right] \cdot \prod_{h \in I(i) \setminus (I(i) \cap I(j))} [1 - p_{rx}(h) \cdot \omega_{h;j}] \ , \tag{3.10}$$

where

- *$p_{tx}(i)$ is the long-term fraction of time spent by station i transmitting packets on the wireless channel;*
- *$p_{rx}(i)$ is the long-term fraction of time spent by station i receiving packets on the wireless channel;*
- *$\omega_{h;j}$ is the ratio of packets transmitted on the wireless channel from neighbors of station h, which are not interferers for station j, and which have a not-null routing probability towards h, and the overall amount of packets transmitted on the wireless channel from neighbors of station h;*

Proof:

In summary, the analytical methodology we adopt to determine the feasibility of a throughput allocation consists of the following steps. First of all, from the WMN topology $G(V \cup \{a\}, E_w, E_g)$ we extract the equivalent queuing network $G'(Q, L)$. Then, given the throughput allocation (λ_o and $\mathbf{P_o}$), and the routing matrix $\mathbf{R_{fwd}}$, we can determine the overall arrival rate at each queue solving the linear system defined in (3.3). From the $\lambda_{i,l}$ values we compute the $p_{tx}(i)$ and $p_{rx}(i)$ parameters, and the $\tau_{i,1;j,s}$ probabilities using Lemma 3. This allows us to derive the average service times of each queue in the network, and to check the feasibility of the throughput allocation.

3.3.3 End-to-End Delay

In this section we describe closed expressions to determine the end-to-end delay for uplink traffic in a WMN. In the framework presented in Section 3.3.2 this

case is obtained by setting $\lambda^r_{a;i}=0, \forall i \in Q, r \neq 0$.

First of all, it is intuitive to note that, even assuming a Poisson model for the external packet generation process at each station of the queuing network, both service times and overall packet inter-arrival times are generally not exponentially distributed. Unfortunately, the problem of deriving closed-form expressions for the state probabilities of a $G/G/1$ queuing network is generally mathematically intractable, and there are analytical techniques based on iterative procedures to approximate the parameters of the queuing network model. In this study, we make two simplifying assumptions. First of all, we approximate a non-product form network as a product form network. In other words, the state probability is approximated as the product of the marginal probabilities that in the queue $q_{i,l}$ there are $k_{i,l}=k$ jobs. Furthermore, we neglect the interactions between queuing stations, which are modeled as individual $M/G/1$ queuing systems. The advantage of such approach is that it allows us to obtain closed-form expressions for the average end-to-end delay, as described in the following.

From the well-known closed-form expressions for $M/G/1$ queues [BGdMT06], we can compute the mean number of jobs in queue $q_{i,l}$ as follows

$$\overline{K}_{i,l} = \rho_{i,l} \left[1 + \frac{\rho_{i,l}}{1-\rho_{i,l}} \cdot \frac{1+c^2_{B_{i,l}}}{2} \right] . \tag{3.11}$$

Then, the average total time a job spends in the queue $q_{i,l}$ either waiting to be served or in service, also known as the sojourn time, is computed using the Little's theorem as

$$\overline{T}_{i,l} = \overline{K}_{i,l} / \lambda_{i,l} . \tag{3.12}$$

In (3.11), the key parameter is $c^2_{B_{i,l}}$, namely the squared coefficient of variation of service times at queue $q_{i,l}$. For the queues modeling transmissions on wired links (i.e., $q_{i,2}$ and $q_{i,3}$), it holds that $c^2_{B_{i,l}} = 0$ because the service time is constant. For the queues modeling transmissions on the wireless channel, recall that $\mu_{i,1} = 1/E[X_{i,1}]$; hence, $c^2_{B_{i,l}} = (E[X^2_{i,l}] - E[X_{i,l}]^2) \cdot \mu^2_{i,l}$. Under the simplifying assumption that $X_{i,1}$ is a geometric random variable, and using standard probabilistic arguments, it is straightforward to derive that $c^2_{B_{i,l}} = C_w(1-\mu_{i,1})/P$.

With the knowledge of the value of $\overline{T}_{i,l}$ and the routing matrix $\mathbf{R_{fwd}}$, the

following lemma provides the end-to-end delay $\overline{E}_{i,l}$, defined as the average time for a job that enters the l-th queue of the i-th station from outside the network, to reach the wired infrastructure.

Lemma 4. *Under the assumption that reception and transmission events in $G'(Q, L)$ are mutually independent, it holds that*

$$\overline{E}_{i,l} = \overline{T}_{i,l} + \sum_{j=1}^{n} \sum_{s=1}^{q(j)} p_{i,l;j,s} \overline{E}_{j,s} \quad \text{for} \quad i \in Q, l \in [1, q(i)] \; . \tag{3.13}$$

Proof: By definition, the average end-to-end delay of a job is equal to the average sojourn delay at each queue traversed by that job over the route from the source to the destination (in our case, the destination is always the wired infrastructure). When a new job is created it enters queue $q_{i,l}$ with a probability $p^r_{o;i,l}$. Thus, $\overline{T}_{i,l}$ is the first contribution for the computation of the $\overline{E}_{i,l}$. After completing the service at queue $q_{i,l}$, the job is routed to queue $q_{j,s}$ with probability $p_{i,l;j,s}$. Owing to the assumption of independence between queuing stations, this job will require a time $\overline{E}_{j,s}$ to reach its intended destination from queue $q_{j,s}$. By writing a similar expression for each queue of the system, we obtain the linear system shown in (3.13), and this concludes the proof. □

Finally, with the knowledge of the $\overline{E}_{i,l}$ values, it is straightforward to compute the average end-to-end delay $\overline{E}$ over all the network queues as follows

$$\overline{E} = \sum_{i=1}^{n} \sum_{l=1}^{q(i)} \overline{E}_{i,l} \cdot p_{o;i,l} \; . \tag{3.14}$$

3.4 Capacity-Aware Route Selection

The most important outcome of the modeling methodology described in Section 3.3, is the development of a *predictive tool* that allow us to determine if a given routing matrix leads to an unfeasible throughput allocation. In this section we address a somehow opposite problem: given a set of flow demands, how to construct the routing matrix that makes the resulting throughput allocation feasible? Our goal is to design a fast and efficient strategy to discover feasible paths in an heterogeneous WMN. As a matter of fact, it is unrealistic to perform an exhaustive search because there are exponentially many paths

between a source/destination pair, and a brute force strategy does not scale. For these reasons, in the literature various solutions have been proposed for reducing the complexity of this problem. A popular approach is to consider only disjoint and braided paths [WB06], but it is still computationally intensive to construct multiple disjoint paths. An alternative strategy is proposed in [MBLD07], where the complexity of finding optimal routes is mitigated by considering only routing forests, i.e., unions of disjoint trees rooted at the gateway nodes. However, tree-based structures are less reliable to link failures than mesh-based structures. The authors in [KGDB07] propose to transform the original network graph into an edge graph, where multiple links are aggregated into segments. This approach results into a reduction in the number of possible paths to check for feasibility, depending on the adopted segment size.

In this work we adopt a simpler approach by constructing a *routing tree* from each mesh node to the available gateways. More precisely, for each mesh node i we compute the minimum cost paths towards each gateway j (with $j \in \mathcal{G}_r \cup \mathcal{G}_p$). Note that minimum cost path can be efficiently computed in a loop-free manner using Dijkstra and Bellman-Ford algorithms if the routing metric is *isotonic* [YWK05]. Thus, the number of paths to check for feasibility grows linearly with the number of gateways and mesh nodes. The penalty we pay for this simplicity is that occasionally the routing process may not find a feasible route although it exists.

To explain the operations of the proposed *Capacity-Aware Route Selection* (*CARS*) algorithm, we adopt the following traffic model for the Internet flows. Specifically, in this study we assume that a bidirectional flow is established between the mesh node $v \in V$ and the wired infrastructure (represented by the virtual node a). This flow needs a certain bit rate to satisfy its QoS requirements. In general, the packet arrival rate can follow a generic distribution thus we express the bandwidth demands in terms of the average packet arrival rate. Now, let us assume that at time t the CARS algorithm has already admitted a set $\mathcal{F}^{(k)}$ of k Internet flows, $f^{(1)}, f^{(2)}, \ldots, f^{(k)}$, and that the i-th flow requested an uplink bandwidth and downlink bandwidth equal to $b_u^{(i)}$ and $b_d^{(i)}$, respectively. The *asymmetry* of flow demands is represented through the ratio $\eta^{(i)} = b_u^{(i)}/b_d^{(i)}$. For instance, If $\eta^{(i)} = 1$ then Internet flow $f^{(i)}$ is symmetric, $\eta^{(i)} = 0$ indicates a downlink Internet flow, while an uplink Internet flow is obtained when $\eta^{(i)} \to \infty$. Finally, let $\mathcal{P}^{(k)}$ be the set of k network paths cho-

sen by the CARS algorithm to route these k flows so as to ensure a feasible throughput allocation in the network. From $\mathcal{F}^{(k)}$ and $\mathcal{P}^{(k)}$ it is straightforward to derive the throughout allocation $\lambda_o^{(k)}$ (e.g., $\lambda_o^{(k)} = \sum_{i=1}^{k}(b_u^{(i)} + b_d^{(i)})$), $\mathbf{P_o^k}$ and the routing matrix $\mathbf{R_{fwd}^{(k)}}$.

Now, let as assume that at time $t+1$ arrives a new flow $f^{(k+1)}$ originated at mesh node $v \in V$ with uplink and downlink bandwidth demands equal to $b_u^{(k+1)}$ and $b_d^{(k+1)}$, respectively. Then, CARS performs the following steps searching for a new routing matrix that permits to admit this new flow:

1. Update the throughout allocation by adding the new flow. Thus, the modified throughput allocation is $\lambda_o^* = \lambda_o^{(k)} + b_u^{(k+1)} + b_d^{(k+1)}$ and $\mathbf{P_o^*}$.

2. Construct two *optimal routing trees* $\mathcal{Q}_u^{(k+1)}$ and $\mathcal{Q}_d^{(k+1)}$. The first one consists of the minimum cost paths from mesh node v to the available gateways, whereas the second one consists of the minimum cost paths from the available gateways to mesh node v. The paths in these sets are ordered from the one with the minimum path cost to the one with the largest one. Note that these path sets may be different depending on the formulation of the routing metric function. Moreover, heterogeneity of fixed line capacities may also lead to a different route selection for upstream and downstream flows.

3. Extract the minimum cost path in set $\mathcal{Q}_u^{(k+1)}$ and set $\mathcal{Q}_d^{(k+1)}$, say P_u^i and P_d^i, respectively.

4. Update the routing matrix by adding P_u^i and P_d^i to $\mathbf{R_{fwd}^{(k)}}$. Let denote with $\mathbf{R_{fwd}^*}$ the modified routing matrix.

5. *Check the feasibility* of throughput allocation $\lambda_o^{(*)}$ and $\mathbf{P_o^*}$ given the routing matrix $\mathbf{R_{fwd}^*}$. If the feasibility check is *positive* then **goto 7**, else **goto 6**.

6. Remove P_u^i and P_d^i from $\mathcal{Q}_u^{(k+1)}$ and $\mathcal{Q}_d^{(k+1)}$, respectively. If either one or both these sets are empty than *reject flow* $f^{(k+1)}$ and *exit(failure)*, else **goto 3**.

7. *Accept flow* $f^{(k+1)}$, and set $\lambda_o^{(k+1)} = \lambda_o^*$, $\mathbf{P_o^{(k+1)}} = \mathbf{P_o^*}$, and $\mathbf{R_{fwd}^{(k+1)}} = \mathbf{R_{fwd}^*}$. Then, *exit(success)*.

Before evaluating the performance of CARS scheme and investigating its load-balancing properties, it is useful to briefly discuss possible refinements of the CARS specification. First of all, we can observe that our model, in addition to check feasibility of throughput allocation, is able to identify which are the queues that get overloaded for a given throughput allocation. A possible enhancement for CARS would be to eliminate from the network topology the mesh routers or gateway nodes that are overloaded, and to re-compute the route sets for the modified topology. This would permit to consider longer paths able to route around congested network regions. Furthermore, in the CARS design the modified routing matrix $\mathbf{R}^{*}_{\mathbf{fwd}}$, which is checked for feasibility, is computed starting from the previous routing matrix $\mathbf{R}^{(k)}_{\mathbf{fwd}}$ without changing the paths used for the previously admitted flows. A possible alternative would be to adopt an approach similar to the one proposed in [MBLD07], and to accept partial reconfigurations of the selected network paths. However, the penalty for an improved adaptability of the routing process would be the increase of computational complexity, and a longer transient phase for network adaptation.

3.5 Performance Evaluation

In this section we use computer-based simulations to validate our analysis, and to evaluate the performance gains obtained by CARS over other two load-balanced routing protocols.

3.5.1 Simulation set-up

We have developed a customized C++ simulator for the performance evaluation of routing protocols in heterogeneous mesh networks. Specifically, in our simulator environment the interference is simulated using the Protocol Model [BCP09a], and the transmission range and interference range of each node are fixed and equal to 100m and 200m, respectively. Regarding the MAC protocol, we have implemented the collision-free CSMA-based access scheme described in Section 3.2. Recall that in our perfectly synchronized MAC protocol the nodes have complete and instantaneous information on other nodes state, i.e., if a node is idle, receiving, transmitting of backing off. Finally, the

wireless channel bandwidth is fixed and set to $C_w = 50$ Mbps, and the channel is assumed noiseless.

In the following experiments, the nodes are deployed in a square area of size 1 Km. In the center of the simulated area we place as single provider gateway (i.e., $n_p = 1$), which has a symmetric high-speed Internet connection with $C_p^u = C_p^d = 1$ Gbps. Then, other 100 nodes (mesh routers and residential gateways) are deployed on a grid layout, with grid points separated by 100m. More precisely, we randomly pick up n_w grid points where we place mesh routers, and in the remaining n_r grid points we place residential gateways ($n = n_w + n_r = 100$). This ensures a sufficient degree of randomness in the locations of gateways. Furthermore, in our tests we have investigated different values for the number n_r of residential gateways.

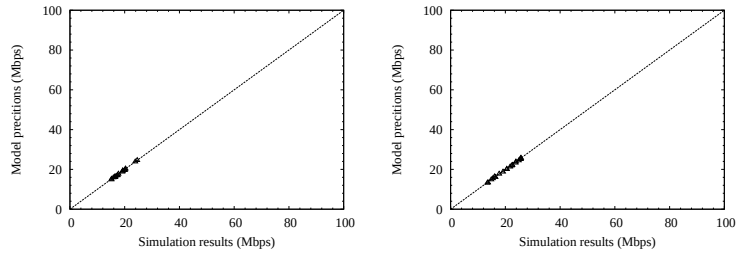
Regarding the traffic model, in this study we use UDP as the transport protocol for generating data traffic. We consider Internet flows established between the wired infrastructure and randomly selected mesh nodes. Each flow is *bidirectional* because a mesh node can both download and upload traffic from/to the Internet. Following the notation introduced in Section 3.4, the $b_u^{(k)}$ and $b_d^{(k)}$ parameters are the average uplink and downlink bandwidth, respectively, demanded by each flow $f^{(k)}$. If not otherwise specified, in the following tests both $b_u^{(k)}$ and $b_d^{(k)}$ are random values uniformly selected in the range $[50\text{kbps}, 150\text{kbps}]$, while the inter-packet arrival time is exponentially distributed.

3.5.2 Model Validation

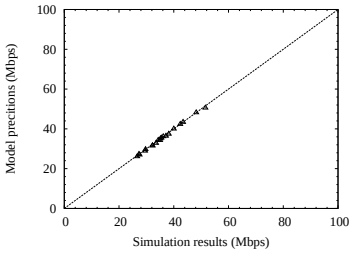
In this section, we verify the validity of the assumptions made in the analysis and the accuracy of model predictions for the maximum feasible throughput and the end-to-end delays. The following results are obtained considering a practical scenario where the packets are always routed to the closest gateway over the shortest path. To compute the shortest path we have used the IRU routing metric [YWK06]; we we refer to this scheme in the sequel as *SPF-IRU*. Note that results in [YWK06] show that IRU metric is able to substantially improve the total network throughput over simple hop count metric by better balancing network loads.

Maximum Feasible Throughput

Following Definition 4, to compute the maximum feasible throughput we use randomly generated traffic traces. More precisely, each flow of the traffic trace is sequentially injected into the network, and the maximum network capacity is obtained when a new flow cannot be accepted without saturating one of the queues in the network.



(a) $n_r/n = 0.05$ (b) $n_r/n = 0.1$

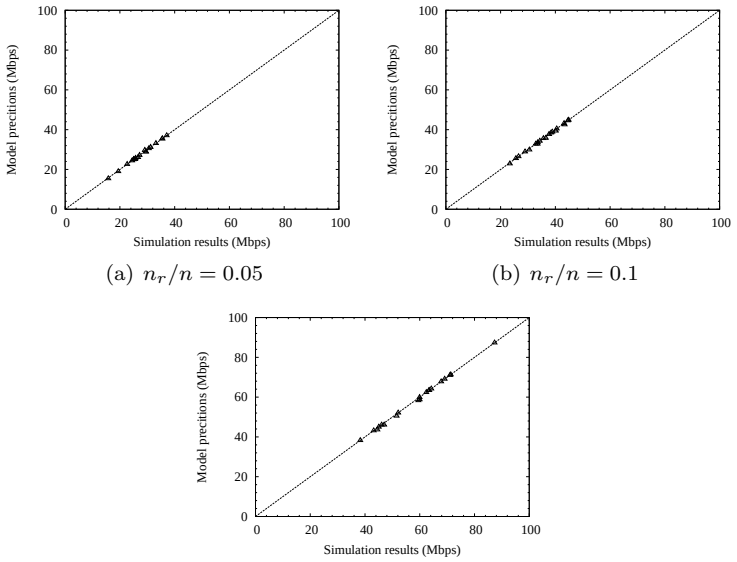


(c) $n_r/n = 0.2$

Figure 3.4: *ASYM GW* Case: comparison between predicted and measured network capacity for symmetric Internet flows ($b_u^{(k)} = b_d^{(k)}, \forall f^{(k)}$).

We have investigated two representative cases. In the first one, referred to as *ASYM GW*, we assume that residential gateways have an asymmetric low-speed Internet connection with $C_r^d = 5$ Mbps and $C_r^u = 2$ Mbps. This is compatible with the characteristics of current ADSL technologies. In the second case, referred to as *SYM GW*, we assume that residential gateways have a symmetric low-speed Internet connection with $C_r^d = 3.5$ Mbps and $C_r^u = 3.5$ Mbps[4]. This allows us to explore the impact on the overall network capacity of the bandwidth asymmetry of Internet fixed lines. Note that diverse levels

[4]Note that the aggregated bandwidth of downlink and uplink wired lines is the same in both cases to facilitate fair comparison of capacity results.



(a) $n_r/n = 0.05$ (b) $n_r/n = 0.1$

(c) $n_r/n = 0.2$

Figure 3.5: *SYM GW* Case: comparison between predicted and measured network capacity for symmetric Internet flows ($b_u^{(k)} = b_d^{(k)}, \forall f^{(k)}$).

of network heterogeneity can be also simulated by varying the percentage of residential gateways over mesh routers. Due to space limitations, we report only plots related to the *SPF-IRU* scheme, but similar results have been also obtained with hop count metric and with CARS algorithm.

For the sake of clarity, we present a brief description of IRU definition as reported in [YWK06]. Specifically, to capture inter-flow interference, the IRU metric for link l is defined as, $IRU_l = ETT_l \times N_l$, where N_l denotes the number of mesh nodes with which the transmission on link l interferences, while ETT_l [DPZ04] is the expected transmission time on link l. Hence, the IRU cost captures the aggregated channel time that transmissions on link l consume on neighboring nodes, which essentially represents the inter-flow interference.

Figures 3.4 and Figures 3.5 show a set of scatter plots comparing the network capacity predicted by our model and the one measured through simulations for symmetric Internet flows (i.e., $b_u^{(k)} = b_d^{(k)}, \forall f^{(k)}$), and for different numbers of residential gateways in the mesh network. Results shown in Figures 3.4 are obtained in the *ASYM GW* case, while Figures 3.5 refer to the

SYM GW case. Each network scenario consists of twenty topologies, and each topology instance is averaged over five different traffic traces[5]. The plots show that the correspondence between theory and simulation is good in all the considered scenarios.

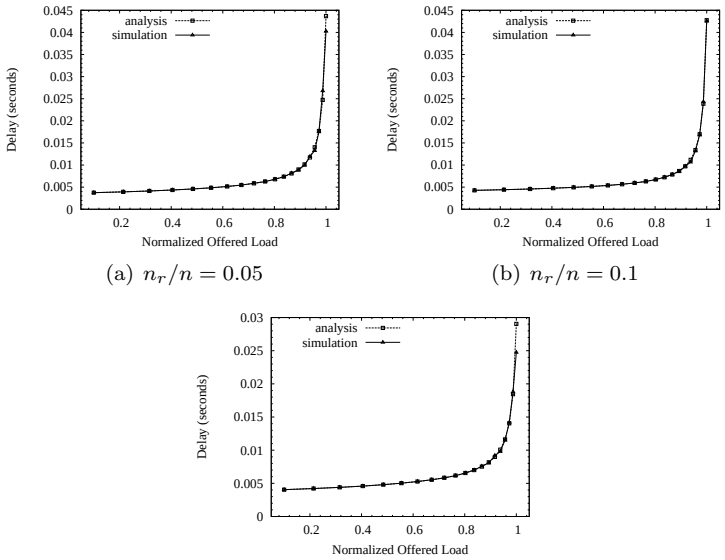
A number of important observations can be derived from the shown results. First, network capacity is greatly dependent on the specific mesh topology and locations of residential gateways, and in general, the higher the n_r/n value, the higher the network capacity. This is expected because adding more gateways increases the aggregate bandwidth available to access the Internet. However, comparing results in Figure 3.4(a) and 3.4(b) we observe that increasing the percentage of residential gateways from 5% to 10% of total nodes has a limited effect on the network capacity, while a substantial throughput improvement is obtained when $n_r/n = 0.2$. The second observation is that the overall network throughput significantly depends on the ratio between uplink and downlink capacity of Internet fixed lines. Comparing the results shown in Figures 3.4 and Figures 3.5 we can conclude that for symmetric Internet flows network capacity is almost twice in the *SYM GW* case than in the *ASYM GW* case. Both these observations motivate the CARS design in which the route selection algorithm takes into account the locations of gateways, as well as the capacity of fixed lines.

End-to-End Delay

To investigate end-to-end delays of uplink traffic we use the following traffic model. First of all, each mesh node is the originator of a single uplink flow, i.e., $b_d^{(k)} = 0$ and $k = 1, 2, \ldots, n$. Furthermore, we consider a uniform traffic distribution, namely $b_u^{(k)} = b, \forall k$. Then, we vary the parameter b to span from lightly-loaded networks up to saturated conditions.

Figures 3.6 compare the end-to-end delay averaged over all the flows as predicted by our model using formula (3.14), and the one measured through simulations for different numbers of residential gateways in the mesh network. Results shown in the figures are obtained in the *ASYM GW* case. As shown in the Figures 3.4, the network capacity significantly changes with the network topology. For this reason, we plot the end-to-end delays versus the normalized offered load to facilitate the comparison between delays observed in different

[5]Confidence intervals are very tight and are not reported.



(a) $n_r/n = 0.05$ (b) $n_r/n = 0.1$

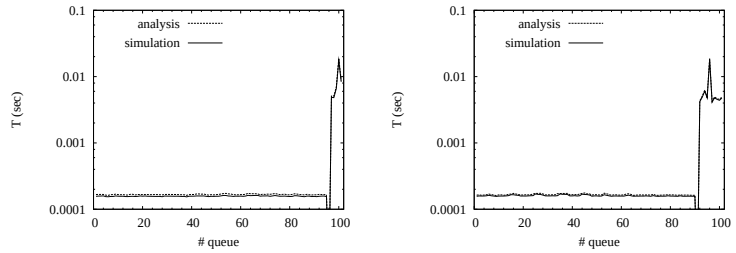
(c) $n_r/n = 0.2$

Figure 3.6: Comparison between predicted and measured average end-to-end delays vs. the normalized network load).

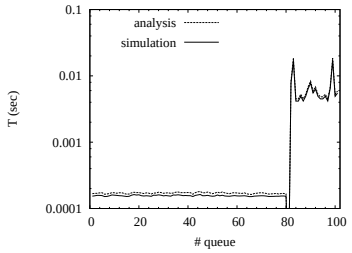
network scenarios. Note that simulation curves are obtained by repeating the tests with five different traffic traces and averaging the obtained results[6]. It is observed that the simulation results agree closely with the theoretical values. From the shown results we can also observe that the delay curves versus the normalized load present very similar behaviors. This can be explained by noting that the SPF scheme tends to quickly overload isolated gateways because they are used by a potentially larger number of mesh nodes. The queuing delay T of such gateway rapidly increases and become dominant over the delays introduced by the other queues.

To better illustrate this behavior Figures 3.7 show on a logarithmic scale the queuing delay of all the queues in the network in the same network scenarios of Figures 3.6 for a network load that is 90% of the maximum achievable throughput. As shown by these results, even close to the saturation, the network resources are unevenly utilized, because the wireless queues $q_{i,1}$ (i.e., $i < n - n_r$ in the graphs) are lightly loaded, while the residential gateways' up-

[6]Simulations are sufficiently long to obtain very stable results. Thus, confidence intervals are very tight and are not reported.



(a) $n_r/n = 0.05$ (b) $n_r/n = 0.1$



(c) $n_r/n = 0.2$

Figure 3.7: Queuing delays with a network load equal to 90% of the maximum achievable throughput.

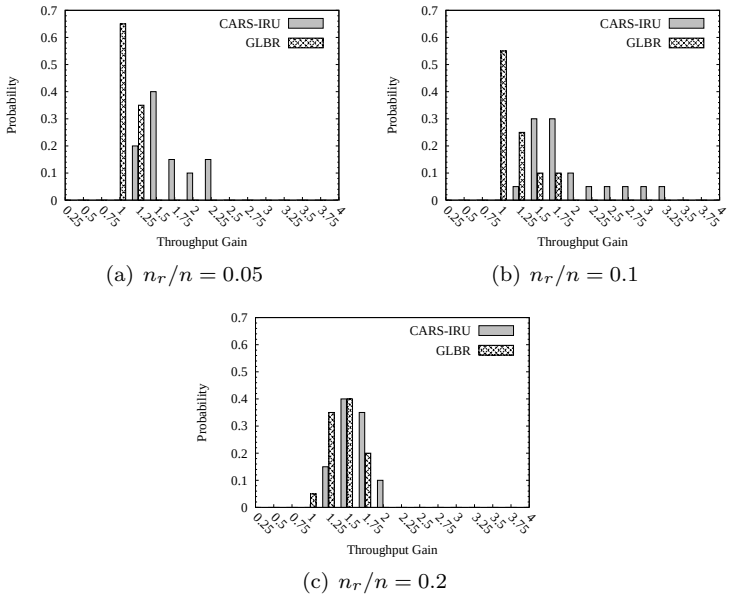
link wired queues $q_{i,2}$ (i.e., $i > n - n_r$ in the graphs) have queuing delay order of magnitude higher. The drop in the queuing delay curve for $i = n - n_r$ is explained by noting that this point corresponds to the providers' uplink wired queue, which has a much higher service rate than other residential gateways.

3.5.3 CARS Performance

To evaluate and compare the performance of different routing strategies, in addition to the proposed CARS scheme we consider two other routing algorithms. Specifically, we consider SPF-IRU [YWK06], which represents a routing strategy that tries to balance the network load under the assumption that the gateways have no capacity constraints, and GLBR [TSHN09], which selects network paths for Internet flows so that the variance of the load on gateway nodes becomes as small as possible, and the increase in path lengths is limited[7]. Note that different variants of the CARS scheme can be devised depending on

[7]The reader is referred to [TSHN09] for a more detailed description of the GLBR algorithm.

the cost function used to compute the all-pairs shortest path matrix. An intuitive option is to construct the set of optimal network paths using the IRU metric, so to ensure a fair comparison with SPF-IRU scheme. For brevity, we refer to this solution as *CARS-IRU*.



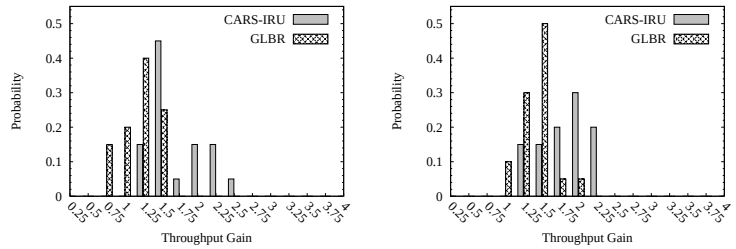
(a) $n_r/n = 0.05$ (b) $n_r/n = 0.1$

(c) $n_r/n = 0.2$

Figure 3.8: *ASYM GW* Case: Histogram of throughput gains of *CARS-IRU* and *GLBR* over *SPF-IRU* for symmetric Internet flows (i.e., $b_u^{(k)} = b_d^{(k)}, \forall f^{(k)}$).

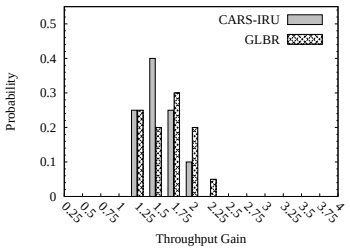
We evaluate the efficiency of a routing algorithm in terms of its *throughput gain* G over the conventional *SPF-IRU* scheme, i.e., the ratio between the maximum network capacity it obtains and the one achieved by the *SPF-IRU* scheme. Since network capacity measurements have a high dispersion over different topologies, rather than using mean or standard deviation as comparison metric, we analyze the *probability distribution* of throughput gains, which provides a deeper insight on system behaviors. To this end, Figures 3.8 and Figures 3.9 show the normalized frequency of throughout gains for *CARS-IRU* and *GLBR* for the same topologies and parameter settings considered in Figures 3.4 and Figures 3.5, respectively. Note that the specific shape of the histograms depends on the set of 20 topologies we used during both analysis and simulations. It is intuitive to realize that different sets would generate dif-

ferent pdfs[8]. More precisely, let us denote with g_i the i-th value of throughput gain reported on the x axis of Figures 3.8 and Figures 3.9 (e.g., $g_i = 1.5$ for $i = 6$). Then, the height of the bar centered on g_i provides the probability that the throughput gain measured in the tested topologies fall in the range $[g_{i-1}, g_i]$.



(a) $n_r/n = 0.05$

(b) $n_r/n = 0.1$



(c) $n_r/n = 0.2$

Figure 3.9: *SYM GW* Case: Histogram of throughput gains of *CARS-IRU* and *GLBR* over *SPF-IRU* for symmetric Internet flows (i.e., $b_u^{(k)} = b_d^{(k)}, \forall f^{(k)}$).

First of all, let us consider the *ASYM GW* scenarios. In this case, the plotted curves show that the *CARS-IRU* scheme ensures an overall throughput improvement over *SPF-IRU* between 25% and 100% in most of the considered topologies, with maximum gains up to 200% for some particularly disadvantaged topologies where almost all residential gateways happened to be close to each other. On the contrary, *GLBR* algorithm provides, on average, much lower performance gains over *SPF-IRU* scheme. More precisely, for $n_r = 5$, *GLBR* algorithm obtains the same maximum achievable throughput as *SPF-IRU* in 65% of the considered topologies. The efficiency of the *GLBR* algorithm improves as more gateways are deployed in the network because this generates more

[8] For the sake of figure readability we do not report analytical results, which are very close to the simulation results.

opportunities for distributing the traffic over the gateways so as to minimize the variance of the network load at the gateways. As shown in Figure 3.8(c), there are only 5% of the considered topologies where *GLBR* algorithm obtains the same maximum achievable throughput as *SPF-IRU*, and there are a few topologies where *GLBR* algorithm outperforms *CARS-IRU*.

Comparing the results in Figures 3.8 and Figures 3.9 it is observed that *GLBR* performs better in the *SYM GW* case than *ASYM GW* case. This can be explained by noting that the *GLBR* algorithm is designed to ensure load balancing primarily for the uplink direction. Thus, if both the uplink and downlink wired links at the gateways have the same capacity, and the traffic flows are symmetric, then the best route in the uplink direction is probably the best also for the downlink direction. Moreover, the results in Figures 3.9 confirm that increasing the number of gateways is beneficial for the throughput performance of the *GLBR* scheme because there are more opportunities for distributing the traffic over the gateways. Nevertheless, *CARS-IRU* outperforms *GLBR* in almost all the considered topologies, and only for $n_r = 20$ the maximum throughput gain for *GLBR* is greater than the one for *CARS-IRU*.

3.6 Conclusions

In this chapter we have shown that a multi-class queuing network model can be effectively used to characterize the maximum achievable throughput, as well as the average end-to-end delay, for heterogeneous WMNs. An important outcome of our analysis is that some mesh nodes may obtain substantially lower throughput than others depending on several factors, including locations of gateways, traffic patterns and link capacities. Hence, network performance could be significantly improved by taking into account the residual capacity of network paths and gateways' connection to the Internet in the route and gateway selection processes. To this end, we have proposed CARS, a capacity-aware routing selection algorithm that takes advantage of model predictions to evenly distribute the network load among available gateways. We have shown through simulations that CARS significantly outperforms conventional shortest path routing, as well as an alternative routing method that distributes the traffic load on the gateway nodes to minimize its variance.

The results and framework presented in this chapter may lead to several

venues for future research. One directions include the delay analysis and characterization of the maximum achievable throughput for network using other MAC protocols, such as TDMA-based access schemes. Furthermore, various strategies can be devised to select the initial subset of optimal routes between the mesh nodes and the available gateways. For instance, delay may be a more significant metric to use for real-time traffic. However, jointly considering capacity and end-to-end delay constraints in the routing process is still an open research area. Finally, the extension of our routing method to deal with intra-mesh traffic in addition to Internet traffic is an ongoing activity.

3.7 Appendix: Proof of Lemma 3

Owing to the analogy between real nodes and equivalent queuing stations, it is more intuitive to represent the interference relationships between nodes in the physical network, rather than between queues. To this end, Figure 3.10 shows the *interference regions* for a wireless transmission from mesh node i to mesh node j. As described in Section 3.2 the interference region of mesh node i, say $I(i)$, consists of the set of mesh nodes such that if $h \in I(i) \subset V$, then $|h-i| \leq (1+\Delta) \cdot r$. Note that the transmission event on wireless channel is modeled using queue $q_{i,1}$, while next-hop station j may route the received packet to either its wireless queue $q_{j,1}$ (for further relaying), its upstream wired queue (if $j \in \mathcal{G}_r \cup \mathcal{G}_p$), or its local access point (if $j \in \mathcal{G}_r \cup \mathcal{G}_p$) depending on chosen destination r and the routing process (i.e., $p^r_{i,1;j,s}$).

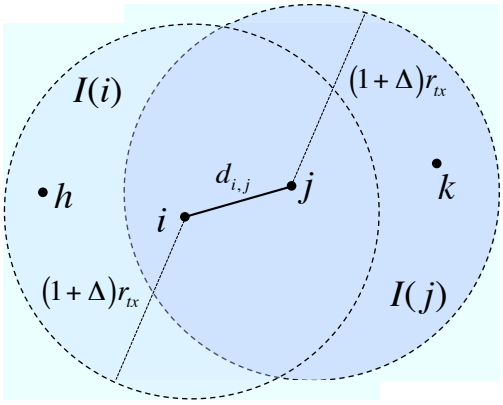


Figure 3.10: Representation of interference areas for a transmission from queue $q_{i,1}$ to queue $q_{j,s}$.

According to the Protocol Model a station i can successfully transmit a packet on the wireless channel to its neighboring station j only if none of the interfering neighbors of station j are transmitting simultaneously, and its transmission will not interfere with nearby receivers. More formally, let us denote with $\mathcal{S}_i$ the event that station i is not transmitting a packet on the wireless channel, and with $\mathcal{R}_j$ the event that station j is not receiving a packet from the wireless channel. Then, the probability $\tau_{i;j}$[9] that a wireless transmission from

[9]Note that the probability of successful transmission $\tau_{i,1;j,s}$ does not depend on the des-

station i to station j is successful, under the Protocol Model approximation, can be written as

$$\begin{aligned}\tau_{i;j} &= \Pr\left\{\left(\bigcap_{k\in I(j)\cup\{j\}} \mathcal{S}_k\right)\bigcap\left(\bigcap_{h\in I(i)} \mathcal{R}_h\right)\right\} \\ &= \Pr\{\beta_{i;j}\cap\alpha_{i;j}\}\ ,\end{aligned} \tag{3.15}$$

where the first term (shortly $\beta_{i;j}$) is the event that every mesh node k, which could interfere with a wireless packet reception at station j, is not transmitting, including j. The second term (shortly $\alpha_{i;j}$) is the event that every mesh node h, which could be interfered by a packet transmitted by queue $q_{i,1}$ (i.e., $h\in I(i)$), is not receiving wireless packets from other mesh nodes. By applying the Bayes' theorem on conditional probability, we can rewrite $\tau_{i;j}$ as follows

$$\begin{aligned}\tau_{i;j} &= \Pr\left\{\bigcap_{k\in I(j)\cup\{j\}} \mathcal{S}_k\right\}\cdot\Pr\left\{\bigcap_{h\in I(i)} \mathcal{R}_h \mid \bigcap_{k\in I(j)\cup\{j\}} \mathcal{S}_k\right\} \\ &= \Pr\{\beta_{i;j}\}\cdot\Pr\{\alpha_{i;j}|\beta_{i;j}\}\ .\end{aligned} \tag{3.16}$$

To compute the $\Pr\{\beta_{i;j}\}$ probability we introduce an auxiliary parameter $p_{tx}(k)$ defined as the long-term fraction of time spent by station k transmitting packets on the wireless channel. By noting that $\Lambda = C_w/L$ is the maximum feasible service rate of a wireless queue, on average it holds that

$$p_{tx}(k) = \sum_{r=0}^{R}\frac{\lambda_{k,1}^r}{\Lambda}\ . \tag{3.17}$$

Then, under the assumption that reception and transmission events at each queue are mutually independent, it is straightforward to derive that

$$\beta_{i;j} = \prod_{k\in I(j)\cup\{j\}}\left[1-p_{tx}(k)\right]\ , \tag{3.18}$$

The derivation of the $\Pr\{\alpha_{i;j}|\beta_{i;j}\}$ expression follows the same line of reasoning used for $\Pr\{\beta_{i;j}\}$. More precisely, we introduce an auxiliary parameter $p_{rx}(h)$ defined as the long-term fraction of time spent by station h receiving

tination queue s. Thus, for brevity we simply write $\tau_{i;j}$

packets on the wireless channel. By analogy with formula (3.17), and using the notation illustrated in Figure 3.2 we can write

$$p_{rx}(h) = \left(\sum_{r=0}^{R} \lambda_{f;h}^{r} - \lambda_{e;h}^{0} \right) / \Lambda \ . \tag{3.19}$$

Remind from Section 3.2 that mesh clients associated to a mesh node send their traffic to a dedicated access point co-located with the mesh node itself. Thus, the $\lambda_{e;i}^{0}$ rate does not initially contribute to the contention on the links of the wireless mesh backbone.

To complete the derivation of the $\Pr\{\alpha_{i;j}|\beta_{i;j}\}$ probability we have to compute the impact on the probability of event $\alpha_{i;j}$ of the conditioning with event $\beta_{i;j}$. Since our modeling strategy is based on an average value analysis, this conditioning is removed by introducing a *weighting factor* $\omega_{h;j}$ for the $p_{rx}(h)$ value, which takes into account the fact that not all the neighbors of station h can transmit wireless packet to station h. Indeed, stations that are neighbors of station h, as well as interferes of station j are blocked (this is due to the conditioning with event $\beta_{i;j}$). Thus, we approximate $\omega_{h;j}$ as the ratio of packets transmitted on the wireless channel from neighbors of station h, which are not interferers for station j, and which have a not-null routing probability towards h, and the overall amount of packets transmitted on the wireless channel from neighbors of station h. More formally, $\omega_{h;j}$ can be written as:

$$\omega_{h;j} = \frac{\sum_{r=1}^{R} \sum_{u \in [N(h) \setminus I(j)]} \sum_{s=1}^{q(h)} p_{u,1;h,s} \cdot \lambda_{u,1}^{r}}{\sum_{r=1}^{R} \sum_{u \in N(h)} \sum_{s=1}^{q(h)} p_{u,1;h,s} \cdot \lambda_{u,1}^{r}} \ . \tag{3.20}$$

Finally, under the assumption that reception and transmission events at each queue are mutually independent, we can write that

$$\Pr\{\alpha_{i;j}|\beta_{i;j}\} = \prod_{h \in I(i) \cap I(j)} \left[1 - \frac{p_{rx}(h)}{1 - p_{tx}(h)} \cdot \omega_{h;j} \right] \cdot \prod_{h \in I(i) \setminus (I(i) \cap I(j))} \left[1 - p_{rx}(h) \cdot \omega_{h;j} \right] \ , \tag{3.21}$$

where the term $p_{rx}(h)/(1-p_{tx}(h))$ takes into account that, due to the conditioning with $\beta_{i;j}$, station h cannot transmit if $h \in I(j)$.

This concludes the proof. □

Chapter 4

Experimental Performance Evaluation

4.1 Introduction

In this chapter we describe a proof-of-concept prototype of a mesh network developed to implement the LARS solution proposed in Chapter 2. The small-scale experiments conducted in our trial realistic 5-node outdoor mesh network is aims to confirm the performance gains with respect to the shortest-path first routing protocol, and anctually they demonstrate the practicality and feasibility of using load-aware route and gateway selection in WMNs.

In Section 4.2 we discuss the approach we adopted to implement such architecture in an experimental mesh network and show the performance results. Then we conclude in Section 4.3.

4.2 Preliminary test-bed evaluation

We have developed a proof-of-concept prototype implementing the proposed LARS solution. The goal of the following small-scale experiments is to confirm the performance gains obtainable with the proposed approach, and to demonstrate the practicality and feasibility of using load-aware route and gateway selection in WMNs.

4.2.1 Test-bed description

The following experimental results have been collected in a trial outdoor mesh network deployed in the CNR's campus area in Pisa, Italy. This mesh test-bed consists of five Soekris-based mesh routers deployed on the rooftops of various buildings, which are equipped wit both directional and omni-directional antennas[1]. Figure 4.1 illustrates the network topology and connectivity graph of our test-bed. Solid lines indicate links between directional antennas, while dashed lines are used to represent links between omni-directional antennas. As shown in the diagram, all mesh nodes except node $GW1$ are equipped with omni-directional antennas of various gains (8 dBi for nodes $GW3$ and A, and 15 dBi for nodes $GW2$ and B). Mesh node $GW1$ is equipped with one 15 dBi Yagi directional antenna pointing to node $GW2$, and one 19 dBi Grid directional antenna pointing to node $GW3$, while nodes $GW3$ and $GW2$ are equipped with one 19 dBi Grid directional antenna and one 15 dBi Yagi directional antenna, respectively, both pointing to node $GW1$. The shortest link in our network is from A to B, which is 80-meter long, while the longest link is from $GW1$ to $GW3$, which extends over 280 meters. These differences in link distances and antenna characteristics ensure a reasonable variability of link qualities. In our mesh test-bed, nodes $GW1$, $GW2$ and $GW3$ are connected to a high-speed wired infrastructure, thus they function as gateways for the other mesh nodes. In order to emulate backhaul links with various bandwidths, we have used *netem* [Fon09], a linux tool that provides network emulation functionalities for testing protocols, including rate control to limit the input/output transmission speed of a network interface. Using netem, we have set up the transmission speed of the wired link at $GW1$ to 2 Mbps, while the wired links at both $GW2$ and $GW3$ are set up to a lower transmission speed equal to 0.5 Mbps. Concerning the wireless interfaces, autorate capabilities are disabled and the data rate is fixed to 11 Mbps. Note that the bandwidth of gateways' fixed links has been set to a lower value than the wireless interfaces to investigate the case that a gateway is the bottleneck node limiting the capacity of the entire network.

[1] A more detailed description of the hardware architecture of our mesh routers is reported in [ABC09]

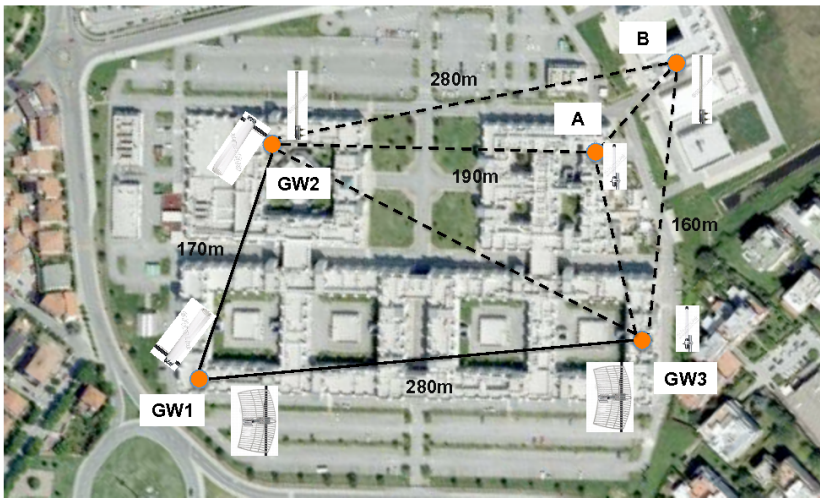


Figure 4.1: Connectivity graph of our experimental outdoor mesh network. Solid lines are directional links, while dashed lines are omni-directional links.

4.2.2 LARS software architecture

In order to gain a more clear insight on the issues related to jointly perform admission control, gateway and route selection using flow-based routing in a real mesh network, it is useful to briefly describe the software architecture we have adopted to implement the LARS scheme. Our reference software architecture is an open source implementation of the ad hoc routing protocol OLSR [CJ03]. The OLSR protocol is pro-active, table driven and utilizes an optimized technique called multipoint relaying for efficient message flooding. The current OLSR daemon also implements an optional link quality extension to compute the link costs according to the ETX metric [DCABM03].

To implement the LARS scheme, we have developed a set of additional software components which have been integrated into the OLSR daemon. For the sake of clarity, the diagram reported in Figure 4.2 illustrates the overall software architecture on both the mesh-node side and the network-manager side, as well as the communications between the different modules we have developed. As shown in the figure, the LARS implementation consists of two separate system components, one running in the kernel space and the other running in the user space, which communicates using the *Generic Netlink* communications channel. The kernel component has been developed using the *netfilter* framework,

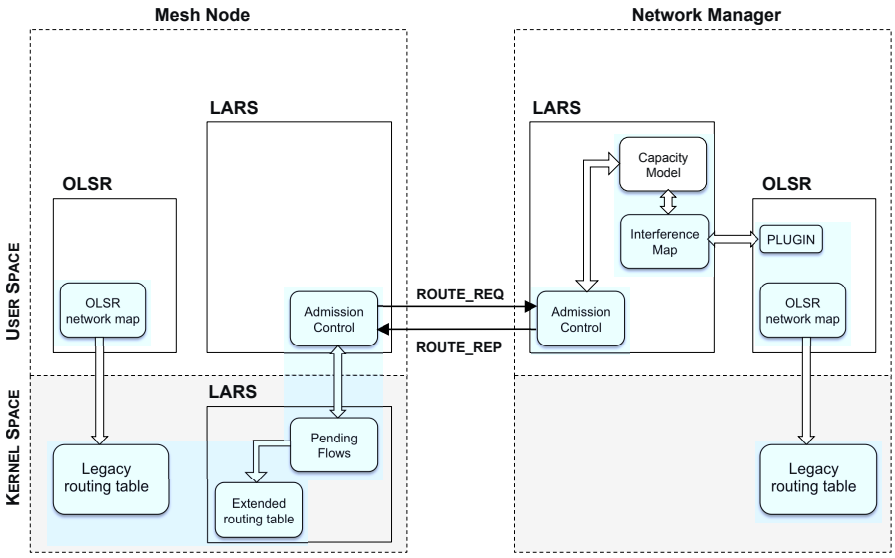


Figure 4.2: Software architecture of the LARS scheme.

which permits to easily implement both stateless and stateful packet filtering. More precisely, each packet received by a mesh node on the wireless interface used to aggregate the traffic generated by mesh clients associated to that mesh node, is intercepted by the LARS module in the kernel *prerouting* chain. If this packet belongs to a new arriving traffic flows, this flow is classified by the LARS module[2], and, if it is an upstream Internet ow, it is added to the list of *pending flows* waiting for approval by the admission control module. More precisely, the LARS component implemented in the user space periodically (every 4 seconds in our implementation), polls the kernel component to check if a pending flow exists. In this case, the mesh node sends a `ROUTE_REQ` message to the network manager to discover a feasible path for this new flow. The network manager replies with a `ROUTE_REP` message, which either contains the route the flow must follow when forwarded in the network[3]. The key issue here is that the legacy kernel routing table does not provide a native support for flow-based routing but only for hop-by-hop routing (i.e., routing based on the

[2]Traffic flows are classified using source and destination IP addresses and transport ports.

[3]Various options are possible for specifying this route. For instance, we can specify the complete route, which is equivalent to support classical flow-based routing. A simpler design choice would be to provide only the identity of the egress router that must be used to exit the mesh network, delegating to OLSR the routing decisions. In our prototype, we have implemented the first option.

knowledge of the next hop for each IP destination). For these reasons we have implemented our own extended kernel routing process supporting flow-based routing, which runs in parallel to the legacy kernel routing process. Note that if a flow is not classified as an upstream Internet flow it passes through the LARS kernel module without any processing, and it is handled in a standard way using the unmodified kernel routing tables.

The LARS component deployed at the network manager is responsible for the computation of the feasible routes, if any, to be allocated to the newly arriving flows. Thus, it implements an admission control module for signaling exchange with the correspondent module collocated at every mesh node. However, to perform the LARS decision process as specified in Section 2.4, it is necessary to know the packet loss rates for each wireless link of the mesh network. To collect this information, we have developed a new *OLSR plugin*, which is a library that can be dynamically linked to the OLSR daemon using a standard API, enabling the generation/processing of OLSR messages, as well as the access to internal functionalities of the OLSR daemon. More precisely, our plugin access the internal OLSR routing table to read the link quality measurements and to build a complete interference map of the wireless mesh network. This map, along with the link loads due to previously admitted flows, is used to execute the LARS algorithm and to determine the feasible route, if any, for the flow that originated the initial `ROUTE_REQ` message. Note that in our prototype we do not reject flows whose bandwidth demands cannot be fulfilled because this would require more sophisticated CAC mechanisms to communicate with the application. On the contrary, the flow is admitted but routed using the legacy OLSR protocol. Moreover, we implicitly assume that the traffic demands are static and known a priori to the network manager. In principle this would be true only if the clients established strict SLAs with the mesh operator. In a more practical case, the network manager should *infer the traffic demands* based on the traffic traces collected at each mesh router/gateway, as well as the most recent traffic demand history. Recently, various schemes have been proposed that integrate traffic estimation with routing and gateway selection [Dai08].

Finally, it is worthwhile to discuss how forwarding is implemented in case of flow-based routing. In principle, the network manager should instruct all the mesh nodes traversed by a flow about the path that flow should use. Then,

a local routing cache should be used to store per-flow routing decisions at intermediate mesh nodes. However, to make easier the implementation of our prototype, we decided to use a simpler approach taking advantage of the IP options fields. Specifically, at the source mesh node, the IP address of each mesh node the packet should traverse is added to the packet IP-header as an additional IP source-route option. The disadvantage of such solution is the typical one of any source-routing based scheme, i.e., an additional protocol overhead is added to the routing process, which is directly proportional to the path length. However, in our small scale network this routing overhead is almost negligible compared to the performance gains ensured by the LARS solution.

4.2.3 Experiments

To gain a better understanding of the advantages and disadvantages of our LARS prototype, and to evaluate the performance limits of the proposed algorithm, we have conducted two distinct sets of experiments.

The first set of experiments aims at validating the correct implementation of the designed mechanisms, and to evaluate the impact of gateways' locations on the system performance. In these tests, every 40 seconds we inject in the network a new traffic flow. A traffic flow is an UDP connection generating packets with a constant rate equal to 100 Kbps and fixed payload equal to 1400 Bytes. For these set of experiments *all the flows are originated at mesh node* B, while the flow destination is a server located in the external Internet. As shown in Figure 4.1, both low-speed gateways $GW2$ and $GW3$ are one-hop distant from mesh node B, while it is necessary to traverse at least two wireless hops to reach the high-speed gateway $GW1$ from node B. Thus, the OLSR algorithm will select one of the two closest gateways as default gateway for the egress traffic generated by mesh node B. The gateway choice if not fixed, but it depends on the current measurements of link qualities for link $GW2 \leftrightarrow B$ and $GW3 \leftrightarrow B$. However, it is straightforward to observe that, in this case, the upstream throughput for mesh node B is mainly limited by the bandwidth of a single slow-speed fixed line (in our case 0.5 Mbps). Thus, it is reasonable to expect that a performance improvement could be easily achieved by using multiple gateways' upstream links in parallel. This configuration is achievable in a linux-based device, because the Linux kernel permits to easily set up a

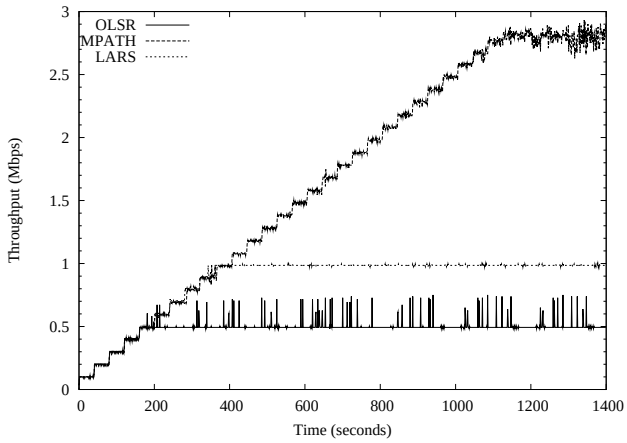


Figure 4.3: Experimental results when all the traffic flows have node B as source.

default route as a multipath route, and to balance the traffic over multiple upstream links. In the considered scenario, we can configure mesh node B to equally use the two gateways $GW2$ and $GW3$. Note that it is not possible to also add a default route to $GW1$ because this default route would necessarily share wireless links with the other default routes generating a cycle in a routing decision process, which uses only information on the next hop when forwarding packets.

Figure 4.3 shows that instantaneous throughout obtained by mesh node B when thirty-five flows are progressively added to the network for three routing strategies: LARS, standard OLSR and static routing with multiple default routes, for brevity MPATH. Note that the overall capacity of the gateways' upstream links is 3 Mbps, which should permit to fulfil the bandwidth demands of at most thirty flows without packet losses. The experimental results clearly indicate that LARS scheme is significantly more efficient than the other two tested algorithms because it ensure the maximum utilization of the resources available at the three gateways. MAPTH also ensures better performance than OLSR because it permits to fully utilize the bandwidth of $GW2$ and $GW3$, but not the bandwidth of gateway $GW1$. Finally, OLSR is the worst among the tested algorithms because it limits mesh node B to use the closest gateway for all the flows it originates. It is interesting to note that the throughput of mesh node B is not bounded to 0.5 Mbps, although that is the maximum speed of the

fixed line at both $GW2$ and $GW3$, but sporadic higher peaks can be observed. These apparently surprising results can be explained by observing that OLSR suffers from route oscillations due to well-known instability of its link quality measurements [RSBA07]. Due to these oscillations, the transmission queues at both gateways $GW2$ and $GW3$ can contain packets generated by mesh node B. This may cause a sort of multiplexing effect on links $GW2 \leftrightarrow B$ and $GW3 \leftrightarrow B$, because there might be short period of times during which both gateways are simultaneously using their wired links to serve node B's traffic.

We have carried out a second set of experiments to extend the previous results to a more general scenario in which all the mesh nodes can be originators of traffic flows. More precisely, a traffic flow is an UDP connection generating packets with a constant rate equal to 100 Kbps, which is injected into the network every 40 seconds, as for the first set of experiments. However, differently from the previous experiments, now the originator of each new flow is randomly selected between the five mesh nodes (i.e., we assume that also the gateways can have mesh clients directly associated to them). The performance metric used to compare alternatives schemes is the overall network throughput computed over all the mesh nodes. Since the number of active flows in the network changes during the experiment, we have computed the mean aggregate network throughput by averaging the instantaneous throughput values measured between two consecutive flow arrivals. In this way, we can univocally associate a throughput measure to a given network offered load. Finally, to have statistical meaningful results, we have used five different traffic traces, each one composed of forty traffic flows.

Figure 4.4 shows the measured average aggregate throughputs, and their 95% confidence intervals, as a function of the network load expressed in terms of number of active flows. We have conducted experiments using both the LARS prototype and the standard OLSR, but not the MAPTH scheme. As described previously, with multiple default routes it is difficult to avoid route cycles in a multi-hop wireless network with multiple flow originators. From the shown experimental results, we can observe that the LARS scheme is able to maximize the network capacity and to fully utilize the network resources. More precisely, given the bandwidth limitations of gateways' fixed lines at most thirty upstream Internet flows could be supported without introducing packet losses on the gateways' transmission buffers. The LARS curve reported in

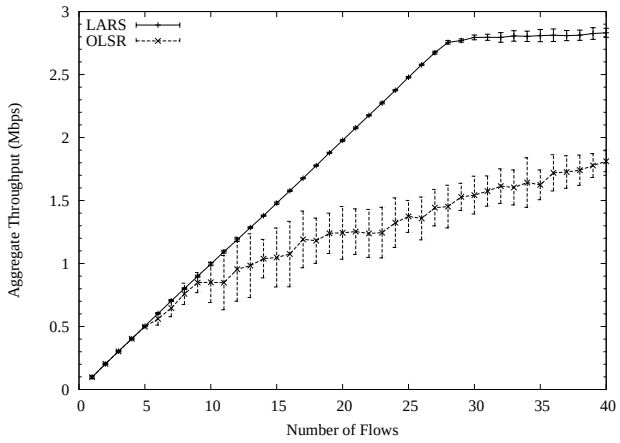


Figure 4.4: Experimental results when the source of each traffic flow is randomly selected.

Figure 4.4 confirms that, in the considered network scenario, our route and gateway selection algorithm is able to satisfy the bandwidth demands of the first thirty flows injected into the network, almost independently of the specific traffic pattern. On the contrary, the standard OLSR performs a blind gateway selection, which quickly introduces inefficiency and significant packet losses. Moreover, with OLSR the network capacity is noticeably dependent on the traffic patterns and gateways' locations. This explains the large confidence intervals that affects the throughput measurements for OLSR.

To better explain the essential reasons why LARS outperforms so significantly the standard OLSR protocols, at least in the considered network scenarios, Figure 4.5 reports the utilization of the gateways' fixed line when in the mesh network there are thirty upstream Internet flows with randomly selected source mesh nodes. Since this number of flows generates an offered load equal to the overall bandwidth of gateways' uplink connections, it is intuitive to acknowledge that at least one of the gateways must be fully used. However, LARS attempts to distribute the load in an uniform way over all the available gateways, while OLSR always selects the closest gateway, leading to a very unbalanced and inefficient use of the network resources.

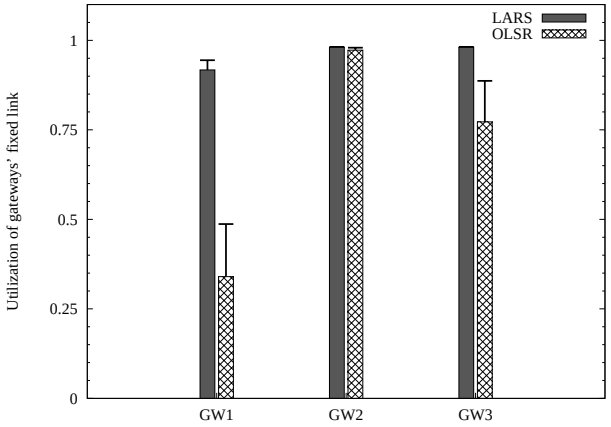


Figure 4.5: Utilization of gateways' backhaul links when in the network there are 30 flows.

4.3 Conclusions

In this chapter we have considered heterogeneous WMNs where gateways' backhaul links may have various speeds. Focusing on this scenario, we have developed a queuing network model to analyze the network capacity as a function of several system parameters, including locations of gateways, traffic patterns, link bandwidths and packet loss rates. By exploiting this predictive tool, we have designed LARS, a load-aware route and gateway selection algorithm that improves the network capacity by ensuring a more balanced utilization of the network and gateways' resources. Using a prototype implementation in a realistic small-scale mesh network, we have shown that the LARS scheme significantly outperforms the shortest path routing using a contention-aware routing metric.

Chapter 5

Hybrid Mesh Networks

5.1 Introduction

In recent years, we have witnessed to an exceptional growth of the number of deployed wireless local area networks (WLANs) as a result of the commercial success of the IEEE 802.11 technology [The99], and the consequent increase in the number of wireless users. A typical 802.11-based WLAN consists of two different entities: access points (APs), also called base stations, which are connected to the network infrastructure, and mobile clients (or stations), which are associated with an AP that is reachable through single-hop wireless transmissions. However, due to radio signal attenuation, the coverage area of a single WLAN is quite limited. In addition, several factors such as electromagnetic interference, fading, obstacles, etc., may impair the radio transmissions. For these reasons, ensuring truly seamless network coverage to a mobile user can be a challenging task.

To extend the range of WLAN systems, two approaches are traditionally followed in real practice. On the one hand, it would be possible to increase the transmission power of an access point in order to reach farther nodes. However, the main shortcoming of this solution is that it may lead to a poor channel reuse because a larger number of users should access the network through the same base station. Consequently, the contention level within each cell increases, thus degrading the per-client throughput. Moreover, the effectiveness of this technique is limited by the fact that the IEEE 802.11 technology operates in

an unlicensed frequency spectrum (i.e., the ISM band) [The99], and national regulations usually set stringent limits to the maximum transmission-power levels in unlicensed bands. Alternatively, we may opt for deploying more access points at a closer spacing, increasing the network capacity. However, a number of reasons, including co-channel interference between nearby access points, availability of a limited number of orthogonal non-interfering frequency channels, as well as cost and management overheads, limit the effectiveness of this alternative solution.

To overcome the limitations of the above-discussed approaches, several authors have recently advocated a new architecture for WLANs, which integrates ad hoc networking technologies in the network infrastructure [LBB04, KSK04, NLP05, ABC+07]. Traditionally, mobile ad hoc networks (MANETs) are conceived as an isolated collection of mobile nodes connected together over a wireless medium, which self-organize into an autonomous multi-hop wireless network [CG07b]. However, it is now recognized that the ad hoc networking paradigm can also be applied to infrastructure-based wireless networks, building an *hybrid ad hoc network*, and providing a flexible, robust and cost-effective increase of network coverage. Specifically, we envisage an *extended WLAN* in which static and mobile clients transparently communicate using traditional wired technologies or ad hoc networking technologies. Thus, the client traffic can be forwarded to the access points through multi-hop wireless paths established by using an ad hoc routing protocol [ABC+07]. It is important to underline that other classes of hybrid ad hoc networks have emerged from this vision, such as: Multihop Cellular Networks (MCN), which combine the features of cellular systems and ad hoc networks [LH00], and mesh networks, which employ a multi-hop wireless backbone to provide Internet access to mobile users [BCG05].

Several technical challenges have to be faced in order to construct such an hybrid ad hoc network because the characteristics of the ad hoc networking (e.g., multi-hop relaying, lack of a centralized administration, etc.) differ significantly from the conventional IP architecture. For instance, the address autoconfiguration protocols commonly used in infrastructure WLANs, such as the Dynamic Host Configuration Protocol (DHCP) [Dro97] or the Zeroconf protocol [CAG05], are not directly applicable in multi-hop wireless networks. However, a mobile device cannot participate in unicast communications until

it has been assigned a free IP address and the corresponding subnet mask. It is evident that pre-configuration is impractical in mobile environments, as well as a violation of the self-organizing paradigm. Thus, an address autoconfiguration protocol is crucial to allow the dynamic and automatic allocation of unique IP addresses to mobile clients. To tackle this problem we propose extensions to DHCP to enable the automatic allocation of globally routable IPv4 addresses to mobile stations in the envisaged extended WLAN[1]. Important features of our proposed solution are the following: *i*) it is a fully distributed and automatic scheme that does not maintain state information in the already configured nodes, *ii*) it does not assume that the address allocation space is known a priori by the new nodes, *iii*) it does not require changes of the legacy DHCP-server implementation, *iv*) no DHCP servers are deployed in the ad hoc component of the extended WLAN (see Section 5.3 for a detailed description of the network architecture), *v*) it is designed to efficiently cope with node mobility, and *vi*) it generates negligible and controlled protocol overheads. Note that DHCP is usually considered not applicable to MANETs since, in case the DHCP server is running on a mobile node, the DHCP server might not be permanently reachable by all nodes. However, our solution is not affected by this problem, since new nodes communicate directly with the DHCP servers deployed on the wired part of the extended WLAN by exploiting the relay capabilities of already configured nodes.

In principle, it may be argued that any other autoconfiguration protocol proposed for ad hoc networks might be also employed to assign a unique network-layer identifier to mobile stations in the envisaged extended WLAN. However, autoconfiguration protocols for MANETs are generally designed to select an identifier with a scope limited to the ad hoc network [WZ04]. This approach is reasonable for *stand-alone* MANETs, which are not connected to external networks, but it introduces additional complexities once we permit the interconnection between ad hoc networks and the Internet. Specifically, if private IP addresses are used within the MANET, a network address translator (NAT) has to be implemented on each gateway to enable IP communications. Then, the NAT-based gateway translates the source private IP address of outgoing traffic with a globally valid IP address, which is routable on the Internet. However, recent studies have clearly demonstrated that NAT-

[1] An initial version of our proposal, as well as preliminary experimental results, were presented in [BCP08].

based gateways are very inefficient when multi-homing (i.e., more than one gateway in the same MANET) is allowed and the network topology is highly dynamic [EE04, ETHE04, ABC+07]. On the contrary, in our paper [ABC+07] we have shown that the use of globally routable IP addresses in the ad hoc network permits to implement very efficient gateways that support transparent IP communications, even in highly mobile conditions. These observations motivate our efforts to use DHCP for assigning globally valid IP addresses also to ad hoc nodes. Note that an alternative approach to configure globally routable IP address would be to use an hardware-based addressing. In other words, a global network prefix may be assigned a priori to the ad hoc network, and the IP address is then completed using the node's unique hardware interface identifier. However, this approach requires additional features that are only available in IPv6. In addition, it is not always true that network interfaces have globally unique addresses, but violations of this assumption are possible.

To verify if our scheme guarantees satisfactory configuration delays and an acceptable efficiency in terms of protocol overheads, we have implemented a fully operational prototype and we have tested its functionalities, taking into consideration various topology layouts, network loads and mobility conditions ([ABCP09]). Our experimental results show that: *i*) even if the new client is several hops far from the DHCP server, and asymptotic TCP flows saturate the wireless links, the configuration delays are acceptable, and *ii*) the protocol overheads are negligible even if node mobility interferes with the operations of the autoconfiguration protocol.

The remaining of this chapter is organized as follows. Section 5.2 outlines the related work on address autoconfiguration protocols for MANETs. In Section 5.3 we define the architecture of an extended WLAN. Section 5.4 briefly reviews the DHCP specification. The basic idea of the proposed solution is presented in Section 5.5, while the protocol details are described in Section 5.6. Section 5.7 presents the experimental evaluation, and Section 5.8 concludes the chapter with final remarks.

5.2 Related Work

Various address autoconfiguration protocols for MANETs have been proposed in the literature, and it is out of the scope of this chapter to present a complete

review. Rather, we focus on outlining the various approaches that have been adopted and the features of representative solutions. The reader is referred to [WZ04] for an exhaustive survey.

Generally speaking, autoconfiguration protocols for ad hoc networks can be classified as *stateless*, *stateful* or *hybrid* solutions. Protocols following a stateful approach are very structured schemes, because every node has to maintain detailed state information about the utilization of the MANET address space. This state information is usually represented by an address allocation table that contains the addresses currently in use within the ad hoc network. The main challenge of this class of solutions is the maintenance of the allocation table consistency, especially in the presence of packet losses and network merging. One of the first schemes employing a stateful approach with a distributed allocation table is the MANETconf [NP02] protocol. With MANETconf an unconfigured node selects a reachable MANET node as the *initiator* of the address allocation procedures. The initiator selects an address that has not been used yet (at least according to its local address table), and it broadcasts a request containing this address to all the nodes in the MANET. An allocation is assumed to be successful only if the initiator receives a positive reply from all the nodes in the MANET. Note that, due to message unreliability, inconsistencies in the allocation tables are still possible, and this may lead to unnecessary address changes or undetectable conflicts. To ensure reliable global synchronization of the allocation tables, it is fundamental to implement reliable broadcast mechanisms, which are generally complex and resource-consuming protocols. To avoid maintaining complete allocation tables in each node, which may not scale in large MANETs, the Prophet protocol [ZNM03] follows a different approach. Specifically, each node in the MANET maintains a function $f(n)$ and a state value, called *seed*, to generate a sequence of integers. Function $f(n)$ is chosen in such a way that the probability to select the same integer when different *seeds* are used is extremely low. When a new node, say B, wants to join the MANET it broadcasts an address request to one of its neighbors, say A, which selects a new *seed* and generates an integer applying this *seed* to $f(n)$. Then, node B will use the generated value as its IP address, and the state value obtained from A as the seed to assign IP addresses to other new nodes. Note that this protocol may generate duplicate addresses. Thus, additional mechanisms are needed to detect and solve these conflicts.

In principle, stateless protocols are less complex solutions than stateful schemes, because each node selects autonomously its own address and performs a Duplicate Address Detection (DAD) procedure to verify its uniqueness and resolve conflicts. However, Perkins et al. [PMW+02] proposed one of the first schemes by adapting the IETF Zeroconf protocol to the MANET case. The basic idea is that each new node selects a random address from a *preconfigured address space*. This means that the IP address block from which nodes have to choose their IP addresses is known in advance to each node. After self-assigning an IP address from this allocation address space, the new node queries all other nodes in the ad hoc network to verify if one of them is already using this address. If the new node does not receive any negative reply within a given timeout and after multiple tries, it will assume that the chosen address is not currently used in the MANET. Two drawbacks can be identified in this scheme. The first one is unreliability caused by the exclusive use of timeouts to stop the DAD procedure, because message delays may be unbounded in an ad hoc network. The second one is the protocol overhead generated by the flooding of the network with address request messages. To increase the protocol efficiency, a different strategy is described in [Vai02], called *weak* DAD, which integrates the DAD mechanism with the routing protocol. More precisely, each node generates a key at initialization time (either randomly or based on an unique hardware ID) and distributes this key in the routing messages. Duplicate addresses are detected by receiving packets with an address that corresponds to multiple keys. Nevertheless, conflicts cannot be detected if two nodes select the same key and the same address. This event is unlikely if the key length is sufficiently large. However, increasing the key length also increases the routing protocol overheads. An optimization of this approach is proposed in the PACMAN (Passive Autoconfiguration for Mobile Ad Hoc Networks) protocol [Wen05], where no additional information (i.e., keys) is sent in the routing node messages, but every node analyzes the routing traffic to identify anomalies. Thus, this protocol implement a *passive* DAD mechanism because conflicts are detected by passively awaiting for routing events that would not have occurred with unique addresses. PACMAN can be classified as a hybrid scheme because every node maintains also an address allocation table. However, these tables are not synchronized, and an unconfigured node can request the allocation table from neighboring nodes only to expedite the con-

figuration process. A shortcoming of this approach is that the DAD procedure depends on the specific routing protocol used in the MANET.

Before concluding this review of related work, it is also useful to outline the activities of the IETF AUTOCONF working group [IET07], which is studying the standardization of mechanisms for configuring unique local and/or globally routable IPv6 addresses. In principle, IPv6 should make the autoconfiguration of unique addresses easier than IPv4 because the size of the IPv6 address space permits each node to build its own globally routable IPv6 address by embedding a globally unique hardware ID (e.g., the 48 bit IEEE MAC address). This is the basic idea of the original IPv6 stateless address autoconfiguration protocol [TN98], and its extension to the MANET case [Fan03]. However, no hardware ID can be considered really globally unique. For instance, interface drivers permit to dynamically change the MAC address. For these reasons, the proposals that have received more attention in the research community are the schemes that use gateway nodes to distribute within the MANET a network prefix that can be used for configuring a (typically globally) routable IPv6 address. One solution is described in [WMP+06], which defines both proactive and reactive strategies to discover the gateways within the ad hoc network. An alternative solution is described in [JNF04]. This scheme introduces the concept of "prefix continuity". More precisely, multiple subnets (i.e., network prefixes) can be used in the same MANET. However, network identifiers should be assigned to visiting nodes in such a way that any node has at least one neighbor using the same prefix. In other words, the MANET should be organized in clusters of hosts sharing the same network prefixes. This network organization reduces the overheads introduced by flooding gateway advertisements.

5.3 Network Model

Before describing the details of the proposed extensions to DHCP, it would be useful to illustrate the complete network architecture we consider for building hybrid ad hoc networks interconnected to the Internet. The application scenario we envisage for this system consists in providing a cost-effective, seamless and robust wireless Internet access for nomadic users in small-scale areas, such as campuses or enterprise buildings. The design goal is to ensure transparent communications between static hosts, which use traditional wired technologies,

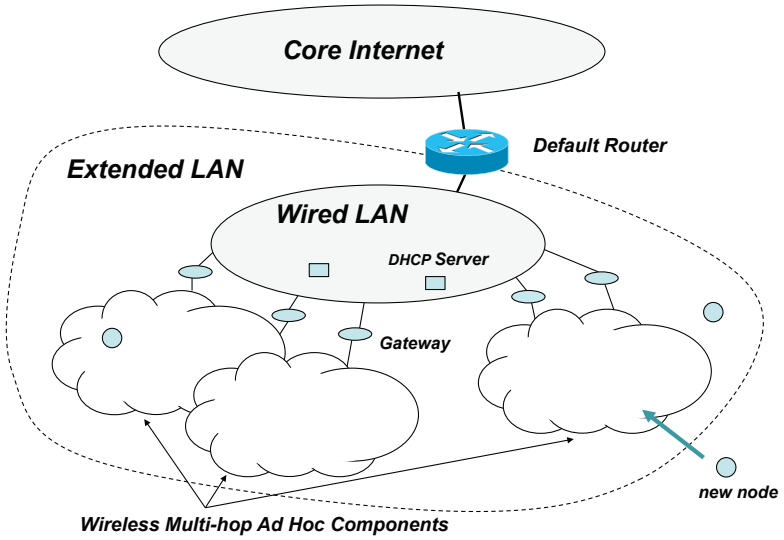


Figure 5.1: Reference network architecture.

and mobile clients, which use more advanced ad hoc networking technologies [ABC+07]. For the sake of clarity, in Figure 5.1 we depict the reference network architecture we consider in our study.

As illustrated in the figure, we envision an extended WLAN, hereafter also indicated as *multi-hop WLAN*, composed of a conventional LAN (the wired component) and several ad hoc components. In this network mobile clients not in close proximity to the fixed networking infrastructure establish multi-hop wireless paths to communicate with each other using an ad hoc routing protocol. Special devices, named gateways, interconnect the wired LAN with the ad hoc components. These gateways are static devices with multiple interfaces. One fixed interface is used to connect the gateway to the wired LAN, while the other wireless interface operates in ad hoc mode. Thus, a gateway can be seen as an enhanced access point supporting ad hoc networking, rather than infrastructure-based wireless communications. Finally, standard IP routing is used to connect the extended WLAN to the core Internet.

In our architecture multi-homing is permitted, i.e., multiple gateways can

be located within the same ad hoc component, and the ad hoc routing manages the network-layer handoff of mobile nodes between gateways[2]. We assume that DHCP servers are deployed in the wired component to administer the dynamic assignment of unique IP addresses to both wired host and mobile clients temporarily associated to the network. This ensures that the *the extended WLAN is a single address space*, where both ad hoc and static hosts have an IP address with the same network identifier. In our previous work [ABC+07] we have shown that this architectural design allows transparent support for node mobility and facilitates Intranet communications. In the following sections we describe how an unconfigured mobile host that wants to join the multi-hop WLAN, for brevity denoted as *new node*, can query the DHCP servers to obtain its IP configuration parameters.

5.4 DHCP Standard

In this section we outline the DHCP specification for IPv4, i.e., DHCPv4 [Dro97]. Note that the modifications introduced with IPv6 to the original IP addressing architecture required a complete redesign of the DHCP standard [DBV+03]. Thus, the extensions to DHCP proposed in this chapter are applicable only to DHCPv4.

DHCPv4 (hereafter simply DHCP) is designed exploiting the client/server model, and DHCP clients and servers interact through a series of client-initiated request-response transactions. Obviously, the DHCP server plays a central role in DHCP because it provides the configuration parameters to the Internet hosts (clients) that communicate with it. In small networks it can be sufficient a single server to support many clients, while large networks may require multiple DHCP servers. The DHCP servers are the owners of the addresses used by all DHCP clients and manage their use, keeping track of both the allocated addresses and the available ones. The most efficient mechanism used by DHCP servers for assigning IP addresses is the *dynamic allocation mode*, which provides a time-limited address allocation. Specifically, DHCP servers assign IP addresses to clients on a lease, and, before the lease expires, DHCP clients should request the renewal of the lease. In this way DHCP servers can imme-

[2]Note that in ad hoc mode there is not link-layer handoff because mobile nodes does not have to *associate* to the gateways.

diately reuse IP addresses that have not been renewed.

Generally speaking, the DHCP communication protocol consists of responses issued by one or more DHCP servers in reply to different types of requests from clients. To describe the client-server interactions it is useful to give an example of a typical dynamic address allocation. First, when a DHCP client boots up, it sends a DHCP_DISCOVER packet to its local physical subnet to locate available servers. This message is a layer-2 broadcast, i.e., the destination IP address is 255.255.255.255. Each DHCP server receiving this broadcast should respond with a DHCP_OFFER sent to the client's MAC address. The DHCP_OFFER includes a tentative IP address for the client, the IP address of the DHCP server sending the response, and the lease duration. A DHCP client may receive multiple DHCP_OFFER messages from different DHCP servers, and the client must choose one of the servers that replied. Then, the DHCP client broadcast a DHCP_REQUEST message to inform all the DHCP servers that the offer has been accepted. To this end, the DHCP_REQUEST message contains the IP address of the selected DHCP server. Only that DHCP server is allowed to respond to the request message with a DHCP_ACK, which contains the rest of the information needed by the client to start conventional IP-based communications, including the location of a DNS server and a default Internet gateway. The DHCP servers that made the offers that were not accepted will return the offered IP address to their range of assignable addresses.

Since DHCP uses the unreliable User Datagram Protocol (UDP) for encapsulating messages, it defines a retransmission strategy to cope with message losses. Note that DHCP clients are responsible for detecting message losses and for all message retransmissions. Specifically, the clients adopt a retransmission strategy that incorporates a randomized exponential backoff algorithm to determine the delay between retransmissions. In general, the delay between retransmissions is doubled up to a maximum of 64 seconds, while the delay before the first retransmission should be 4 seconds plus a random value uniformly selected in the range $[-1, 1]$ [Dro97].

One of the most important limitations of DCHP, which is common to several host configuration protocols, is the reliance on broadcasts for communication. For performance reasons, broadcasts are normally propagated only within a local network segment, and this means that DHCP clients and DHCP servers on different physical network segments cannot communicate directly. To elim-

inate the necessity of having a DHCP sever on every single physical subnet, a router (or a normal Internet host) can be configured as a *DHCP Relay Agent*. A DHCP relay agent will intercept DHCP_DISCOVER and DHCP_REQUEST packets from clients. Then, the DHCP relay can either rebroadcast the clients' DHCP messages to other networks, or send them directly to specific DHCP servers it was configured to contact. The DHCP server responds back to the relay agent that, in turn, forward the servers' replies directly to the original client's MAC address.

5.5 Outline of the Idea

The goal of our autoconfiguration scheme, called *Ad-Hoc DHCP* (*AH-DHCP*), is to assign a globally routable IPv4 address to the mobile nodes of a multi-hop WLAN using the DHCP-based mechanisms already implemented in the wired part of the network, without requiring any change of the standard DHCP server implementation. In this way we can assign globally routable IP addresses to ad hoc nodes without requiring that a pre-configured IP address space is reserved to the ad hoc components.

To enable a new node to deliver its address request to the available DHCP servers, we exploit the DHCP relay capability. More precisely, a new node should execute a preliminary discovery procedure to identify other wireless nodes already associated with the multi-hop WLAN and reachable through one-hop wireless transmissions. Then, the unconfigured node elects one of the discovered neighbors to act as DHCP relay agent, which will forward all the client's DHCP messages to the known DHCP servers. The DHCP standard does not define any specific mechanism to discover the available DHCP relay agents, but client-originated DHCP packets are implicitly forwarded by the relay agents located on the same physical network segment of the client. This behavior is acceptable in wired networks because they are controlled environments, and both the location and number of DHCP relay agents are carefully planned. Typically, DHCP relay agents are enabled only on the interfaces of routers interconnecting different subnets. On the contrary, in a multi-hop WLAN each wireless node is a potential DHCP relay agent that may act as a proxy during the configuration process of a new node. Therefore, if multiple DHCP relay agents are used concurrently to pass client's messages to

DHCP servers, the DHCP servers may be overloaded by the simultaneous requests. Moreover, multiple copies of the same DHCP messages will travel in the multi-hop WLAN increasing the protocol overheads[3]. In conclusion, introducing a DHCP relay agent discovery mechanism can introduce a twofold benefit. Firstly, it reduces the number of messages generated during the configuration process. Secondly, it guarantees that the DHCP servers receive a single address request from each new node joining the multi-hop WLAN.

There is another shortcoming in the original design of DHCP that prevents its efficient use in multi-hop WLANs. Specifically, DHCP standard assumes that nodes are static during a client-server transaction, and message losses are infrequent. For these reasons, DHCP clients adopt a simple retransmission strategy that relies on timeouts to detect messages losses [Dro97]. However, a multi-hop WLAN is a dynamic environment where nodes are free to move almost arbitrarily. Thus, the selected DHCP relay and the unconfigured node may move out of their respective transmission ranges and become unreachable before the address assignment is completed. This may lead to unacceptable delays in the address allocation. Moreover, external interference or routing protocol inconsistencies can produce not negligible packet errors. Consequently, efficient procedures should be devised to cope with node mobility, and unexpected communication problems. To this end, our scheme incorporates a mechanism to allow a timely detection of nodes' movements and/or failures in order to ensure a prompt re-selection of a new valid DHCP relay agent.

After the completion of the initial configuration procedure, each wireless node has to periodically interact with the DHCP server to renew its address. Some authors [NP02, WZ04] observed that it might be difficult to guarantee a continuous access to DHCP servers since ad hoc networks can become partitioned due to node mobility. However, in the considered network scenarios this limitation does not appear problematic. First of all, the multi-hop WLAN we envision will be mostly used as a flexible and cost-effective extension of the fixed networking infrastructure in enterprise buildings or campus facilities. In these contexts, users are semi-static or nomadic and are interested in having a continuous access to Internet and its centralized services (e.g., web browsing, access to centralized data repositories, etc.). In addition, DHCP servers are located only in the wired part of the network. Thus, until the wireless node

[3] An analysis of the use of the overhead associated to the use of multiple relays is reported in Section 5.7.3.

is able to reach an access point through a multi-hop path, it will be able to contact the DHCP server for address renewals.

5.6 AH-DHCP Description

We assume that the gateways are the first nodes to join the multi-hop WLAN. Note that the gateways can interact with the DHCP servers using their wired interfaces. For this reason, AH-DHCP does not need an initialization procedure, which, on the contrary, is an important task of autoconfiguration protocols for stand-alone MANETs [NP02]. Thus, in the following we only describe the AH-DHCP operations when a new node (other that the access point) wants to join the multi-hop WLAN. For brevity, and whenever ambiguity does not occur, we refer to AH-DHCP clients and AH-DHCP relay agents simply as clients and relays. For the sake of clarity, in Figure 5.2 and Figure 5.3 we illustrate the protocol state machines of a client and a relay agent, respectively. In these diagrams we represent the events that initiate a transition in brackets (e.g., the expiration of a timeout, the reception of a specific message, etc.). If a message is generated at the end of a transition, it is represented with a box at the end of the transition arch. Furthermore, Table 5.1 and Table 5.2 list the messages and parameters specific to AH-DHCP.

5.6.1 DHCP Relay Discovery Phase

Let node C be a new mobile node that wants to join the multi-hop WLAN. To this end, it has to query a DHCP server for receiving the necessary IP configuration parameters. Thus, node C starts its AH-DHCP client module entering into the "DHCP relay discovery" state. Then, node C periodically broadcasts special messages, called RELAY_DISCOVER messages (see Figure 5.2), with period T_R. Every wireless node that is already part of the multi-hop WLAN, and is running a relay agent, after receiving a RELAY_DISCOVER message, should reply with a RELAY_ACK message (see Figure 5.3). This RELAY_ACK message expresses the willingness of the relay agent to act as *initiator* of the address configuration process for node C. Note that RELAY_DISCOVER messages are broadcast frames that can be received only if two nodes are in radio visibility, while RELAY_ACK messages are unicast frames sent directly to node C's MAC address.

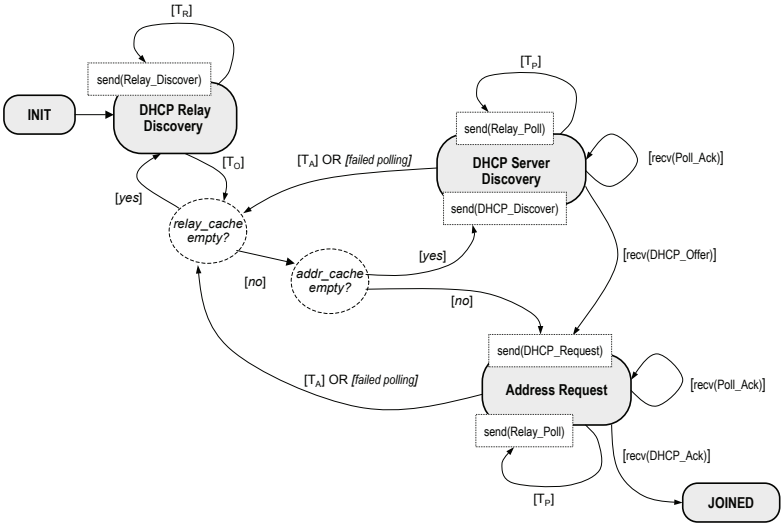


Figure 5.2: State machine of the AH-DHCP client's behavior.

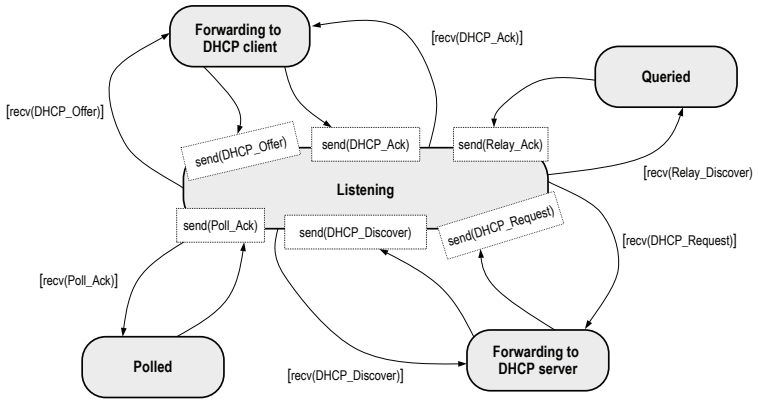


Figure 5.3: State machine of the AH-DHCP relay agent's behavior.

Table 5.1: AH-DHCP message notation.

Message Type	*Message Description*
RELAY_DISCOVER	hello-like message sent by a new node during the DHCP relay discovery phase
RELAY_ACK	reply of DCHP relays to RELAY_DISCOVER messages
RELAY_POLL	poll message sent by a new node to the selected DHCP relay agent
POLL_ACK	reply of DHCP relays to RELAY_POLL messages

Each RELAY_ACK message transports a list of attributes characterizing the DHCP relay capabilities, such as the remaining battery energy, the distance (in terms of hops) between the relay and its closest gateway, if the relay is already involved in a configuration procedure for another wireless node, etc. The identity (MAC and IP address) and the attributes of each relay that replied to a RELAY_DISCOVER message are stored in a temporary cache, called *relay_cache*. Node C allocates a fixed time, say T_O, to collect the neighbors' responses. After the expiration of this timer, node C selects the "best" relay according to a pre-defined policy applied to the attributes of discovered relays. In our prototype we have implemented the following strategy: the relay that is at the minimum distance from an access point should be selected as the forwarder of DHCP messages. Note that other mechanisms could be devised to select a single DHCP relay agent. For instance, it could be possible to implement a timer-based response mechanism where the timer is set based on some preferences (e.g. DHCP relays having a shorter distance from the gateways have smaller timeout values, and may respond first). However, a DHCP relay discovery scheme based on period broadcast messages can be easily integrated into classical hello-like neighbor discovery schemes implemented in popular ad hoc routing algorithms (e.g., AODV or OLSR). After selecting a DHCP relay agent, say R_A, node C can begin a conventional DHCP transaction by sending a unicast DHCP_DISCOVER message to R_A. Note that legacy DHCP clients transmit broadcast DHCP_DISCOVER messages because they are not aware of the available DHCP relays. On the contrary, since AH-DHCP clients scan their neighborhood to discover available AH-DHCP relays, they can use a single relay as unique initiator of the address allocation process. This avoids sending multiple copies of the same allocation request to the DHCP servers.

Table 5.2: AH-DHCP parameters.

Parameter Type	*Parameter Description*	*Default Value*
T_R	repetition period of RELAY_DISCOVER messages	20 msec
T_O	maximum duration of DHCP relay discovery phase	100 msec
T_P	repetition period of RELAY_POLL messages	50 msec
max_{miss}	Relay_Poll	4
T_A	timeout for a DCHP transaction	3 sec

As described above, to increase the probability of receiving at least a response from neighboring wireless nodes, node C periodically broadcasts new RELAY_DISCOVER messages with period T_R. However, to avoid synchronization with other AH-DHCP clients in radio visibility of node C and transmitting RELAY_DISCOVER messages, the generation of these packets should be randomized. Several randomization schemes have been proposed in literature for wireless environments, especially for multi-hop broadcasting [NTCS99]. However, during the DHCP relay discovery phase we use only local broadcasts to discover one-hop neighbors. Thus, we may avoid sophisticated mechanisms to reduce collision probability. For the RELAY_DISCOVER messages, we simply add a variable jitter to the time instant at which a new message should be transmitted. More precisely, if t_k is the time instant at which node C should transmit the k-th RELAY_DISCOVER message, the real transmission is scheduled at time $t'_k = t_k + jitter$, where $jitter$ is a random value selected in the interval $[-MAX_j, MAX_j]$. In our prototype implementation we selected $MAX_j = 0.1 \cdot T_R$. Note that this randomization strategy is similar to the one adopted in the OLSR specification [CJ03] to avoid synchronization of routing control messages. Similarly, it is possible to have collisions involving the RELAY_ACK replays, because node C may have a large number of neighboring nodes with DHCP relaying capabilities. Again, we adopt as collision avoidance strategy the randomization of RELAY_ACK transmissions, but we provide to these packets a higher level of spreading by selecting a maximum jitter value equal to 50% of T_R. Finally, it is possible that after the T_O expiration, node C has not received any response. In this case, node C re-initializes the T_O timer and continues transmitting RELAY_DISCOVER messages.

5.6.2 DHCP Transaction

After sending the unicast DHCP_DISCOVER message to the selected relay R_A, node C waits in "DHCP server discovery" state for receiving a DHCP_OFFER message from the DHCP sever, which the relay agent has forwarded the message to (see Figure 5.2). As explained in the Section 5.4, each DHCP_OFFER message contains the tentative configuration parameters offered by the replying DHCP server. Thus, node C, after receiving a DHCP_OFFER message, extracts these configuration parameters and store them in a temporary cache, called *addr_cache*. Then, node C sends an unicast DHCP_REQUEST (note that in standard DHCP, DHCP_REQUEST messages are broadcast messages) to the selected relay, and it waits in the "Address request" state for receiving a final DHCP_ACK message from the DHCP sever, which would complete the configuration process. When node C has received from the DHCP server the confirmation for using the requested configuration parameters, it can start the ad hoc routing agent and get associated to the multi-hop WLAN. It also activates its internal DHCP relay agent to intercept the requests of future nodes that want to join the multi-hop WLAN.

It is important to note that in our solution the number of DHCP messages received by the DHCP server is constant, and independent of the network topology. Furthermore, the number of DHCP messages transmitted in the network during a DHCP transaction depends only on the number of hops of the shortest path between node C and its closest gateway. For instance, let us assume that node C has n neighboring DHCP relays, and that the closest gateway is $d+1$ hops far from node C. Under the hypotheses that no DHCP messages are lost during a DHCP transaction, it is straightforward to derive that the number of DHCP messages transmitted in the ad hoc network is equal to $4d$ (a DCHP transaction is composed of four DHCP messages, which are replicated on each of the d links between the DHCP relay and its closest gateway), while the DHCP server receives only two DHCP messages and generates two replies (see Section 5.7.3 for experimental results confirming these observations). On the contrary, activating all the n available relays generates *uncontrolled* overheads, and an excessive number of messages per DHCP transaction. More specifically, the number of DHCP messages transmitted in the ad hoc network is at least $4d \cdot n$. Note that this is a lower bound for the protocol overhead because some of the DHCP relays may have their closest gateway further than d hops.

Moreover, the DHCP server will receive $2n$ DHCP messages, generating $2n$ replies. In other words, the protocol overheads increases at least linearly with the number of neighbors of node C.

As noted in Section 5.4 each node has to periodically renew its DHCP lease with the DHCP server. However, it may happen that the DHCP relay agent is not able to contact the DHCP server (e.g., due to inconsistencies of routing table, poor link qualities, etc.) and to renew its IP network parameters. In this case the node cannot participate to the routing because its IP information have to be considered stale. Thus, this node has to repeat the address autoconfiguration process described in Section 5.6.1 to acquire new IP network parameters. Nevertheless, it is reasonable to believe that the failure of a renewal attempt will be a rare event, with no appreciable impact on the configuration latencies of new nodes joining the network.

5.6.3 Message Losses and Local Node Mobility

In the previous section we have implicitly assumed that there are no DHCP message losses. However, in real environments DHCP messages can be lost for several reasons. For instance, it can occur that between the selected relay and the access point there are persistent communication problems (e.g., overloaded channels, link breakages, etc.) that make the transmission delays unlimited. In addition, frames can be lost due to channel interference or unexpected node crashes. Finally, being mobile, node C and the selected relay R_A can move during the DHCP transaction without remaining in radio visibility. As explained in Section 5.4, legacy DHCP clients implement a retransmission strategy using a randomized exponential backoff algorithm, with a maximum retransmission delay of 64 seconds [Dro97]. Such a delay is acceptable only because DHCP message losses are assumed extremely rare in wired networks. However, this strategy is not adequate to cope with an highly dynamic system. To ensure that node C is able to promptly discover a topology change, we implement a *proactive pollling* mechanism in our AH-DHCP client. Specifically, during a DHCP transaction the new node C sends periodic unicast RELAY_POLL messages, with period T_P, to the selected DHCP relay R_A, which mandatorily replies with a POLL_ACK message. If R_A does not reply to max_{miss} consecutive polls, node C can assume that relay R is not reachable anymore and it removes that relay from the *relay_cache*. Note that R_A stops replying to node

C's RELAY_POLL messages also if it looses its connection to the gateway. The generation period of RELAY_POLL messages and the max_{miss} value should be chosen as a tradeoff between the promptness in detecting topology changes, protocol overheads and the tolerance to poll message losses. As shown in Section 5.7.2, with a proper setting of the polling mechanisms, the increase of address configuration latency due to node mobility can be of the order of a few tens of milliseconds in some configurations.

After a failed polling, node C should search an alternative relay in its *relay_cache* (see Figure 5.2). However, if no alternative relays are already known, the only choice for node C is to start a new DHCP relay discovery phase. On the other hand, if an alternative DHCP relay is known, say R_B, node C can resume the DHCP transaction using this new relay. In this case, two possibilities can occur. One possibility is that the configuration process was interrupted before node C received a DHCP_OFFER message from a DHCP server. Then, node C has to send a new unicast DHCP_DISCOVER message to R_B. The other possibility is that node C has already received a DHCP_OFFER message from a DHCP server. Then, it can retrieve the offered IP parameters from the *addr_cache* and send a new DHCP_REQUEST message through DHCP relay R_B for the same IP parameters.

Note that the above-described polling mechanism is effective to quickly detect communication problems between the new node and the selected relay agent. However, if the DHCP transaction fails due to communication problems between the selected relay agent and its gateway, the polling mechanism is ineffective because the client will continue to receive the POLL_ACK messages. For this reason it is still necessary to implement a timeout to detect possible losses of DHCP messages. However, we substitute the legacy exponentially backoff algorithm used by DCHP clients to set retransmission timeouts with a fixed timeout T_A. We believe that the use of a fixed timeout is more suitable for a highly dynamic, and potentially lossy, environment, because it allows more prompt detection of failed DHCP transactions.

5.7 Experimental Evaluation

To verify if our proposed scheme guarantees satisfactory address configuration delays and an acceptable efficiency in terms of protocol overheads, we

have implemented a fully operational prototype of AH-DHCP, and we have tested it in a multi-hop WLAN, composed of two access points and five mobile nodes. To the best of our knowledge, stateful address autoconfiguration protocols (e.g., MANETconf), which are the schemes most similar to our approach, have been only validated via simulations, and no implementations are available. Note that publicly available solutions for address configurations in hybrid ad hoc networks and mesh networks are generally based on private addressing rather than routable Internet addresses, require NAT-based gateways, and use portions of the MAC addresses to build the internal IP address (see for instance, the addressing scheme used in MIT Roofnet [BABM05] or in the Microsoft Mesh Connectivity Layer [DPZ04]). Thus, their functionalities are not comparable with our proposal. On the other hand, most experimental mesh networks use static addressing, while commercial mesh networks employ proprietary schemes.

For the sake of flexibility, we did not use commercial access points in our testbed, but computers equipped with both a wired and wireless interface, and implementing the gateway functionalities described in [ABC+07]. To develop the AH-DHCP prototype we adopted as reference implementation the DHCP client and relay agent public source code provided by the Internet System Consortium (ISC), which is one of the most popular DHCP distributions for POSIX-compliant operating systems [Int06]. Then, we made the necessary modifications to the DHCP software modules to implement the mechanisms described in Section 5.5. Concerning the DHCP server, we used the legacy DHCP server deployed on our campus wired network, which the gateways were attached to.

Regarding the hardware configuration, our testbed consists of seven *Acer Aspire 5633WLMi* laptops with *Intel Pro-Wireless 3945* as integrated wireless card. All nodes use a Linux 2.6.22 kernel and run the OLSR_Unik implementation in version 0.4.10, which is fully compliant with the RFC 3626 [CJ03]. The ad hoc nodes are connected via IEEE 802.11*b* wireless links, transmitting at the maximum fixed rate of 11 Mbps. All nodes were located in the same room, and the *IP-tables* feature of Linux was used to emulate the multi-hop topologies. In our experiments, the background traffic is represented by persistent TCP flows, i.e., long-lived TCP connections transferring infinite-size files, and we used the *iperf* tool [NLA05] to generate these flows.

It is worth pointing out that we conducted the performance tests in an area of CNR building covered by other uncoordinated WLANs, which introduced uncontrollable radio interference. However, we believe that the randomness due to the external interference is well representing the characteristics of real radio environments and it is useful to attain more realistic results. To measure steady-state performance we have replicated each test two hundred times. The following graphs report both the average values and the 95% confidence intervals, which are generally very tight and not always easily appreciable from the graphs.

5.7.1 IP Address Configuration Delay in Static Configurations

First, we carried out a set of experiments to select the most appropriate parameter setting for the DHCP relay discovery phase. Following the notation introduced in Section 5.6 and listed in Table 5.2, let T_R be the repetition period of RELAY_DISCOVER messages, and T_O the observation interval during which the new node collects the RELAY_ACK messages sent by the neighboring relay agents. In general, the new node can have several neighboring nodes already part of the multi-hop WLAN. Hence, it is important for the client to discover all the possible relays in order to select the best one (e.g., the relay at a shortest distance from an access point). It is obvious that the efficiency of the DHCP relay discovery phase depends on how frequently the new node generates RELAY_DISCOVER messages, and for how long it collects the relays' replies. In principle, the shorter the T_R period, the faster should be the discovery process. However, the closer two consecutive RELAY_DISCOVER messages are, the higher the probability that RELAY_ACK messages generated by different relays collide. To investigate this effect we used the network layouts illustrated in Figure 5.4. More precisely, we considered a single client C with n neighboring AH-DHCP relays. All these potential relays are in radio visibility with the same access point A. Thus, the distance between the client C and the access point A is two hops. In the experiments we varied the T_R parameter and we forced the client to execute a continuous DHCP relay discovery procedure. Then, we measured the minimum time needed to receive a RELAY_ACK message from all the available relays. We initially performed our test without background traffic, i.e., when OLSR routing messages and AH-DHCP messages are the only packets

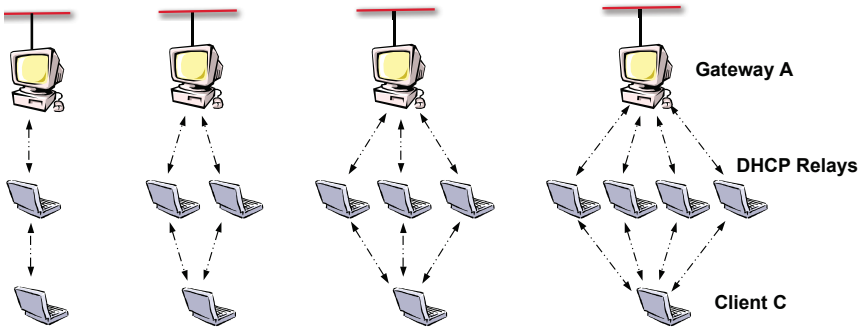
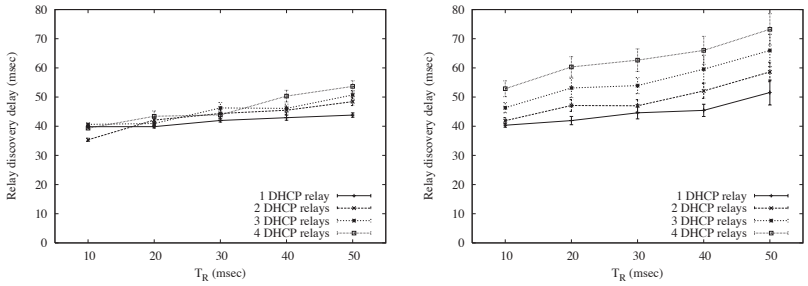


Figure 5.4: Network layouts used for measuring the efficiency of DHCP relay discovery phase.

transmitted over the wireless links. Then, we replicated the test introducing background traffic consisting of asymptotic TCP uplink flows opened between each relay and the gateway. If not otherwise stated, the TCP payload size is 1024 bytes.

Figure 5.5(a) shows the minimum T_O interval needed to discover all the available relays in a network without background traffic as a function of T_R and for various n values. From the experimental results we observe that the time needed to complete the DHCP relay discovery procedure slightly increases by increasing the T_R period and the number of relays to discover. In addition, even for $T_R = 10$ msec (that is the shortest repetition period of RELAY_DISCOVER considered in our tests), the minimum time needed to discover a single relay is about 40 msec. By inspecting the packets traces we found out that this is mainly due to two reasons. Firstly, RELAY_DISCOVER messages are broadcast frames that are not protected by layer-2 retransmissions. Thus, the transmission of these messages is unreliable and they can get lost in the wireless channel. Secondly, the generation of RELAY_ACK packets may be subject to a non-negligible delay because the AH-DHCP relay module has to read the node's routing table to fill in the list of attributes, which is delivered within each RELAY _ACK massage. The user-space function we adopted to access the internal routing table introduces up to 10 msec of delay.

We replicated the same tests adding TCP background traffic saturating the wireless links, and Figure 5.5(b) reports the measured minimum T_O interval. As expected, the minimum time needed to discover all the neighboring relays



(a) without background traffic (b) with background traffic

Figure 5.5: Minimum duration of DHCP relay discovery phase.

increases by introducing background traffic because both collision probability and queuing delays increase. However, with $T_R = 20$ msec, it is still possible to discover four DHCP relays in less than 60 msec. Thus, according to our results $T_R = 20$ msec is a reasonable trade-off between the promptness of the DHCP relay discovery phase and the protocol efficiency (a detailed analysis of the AH-DHCP overhead is reported in Section 5.7.3). Thus, the experimental results shown in the rest of this chapter have been obtained by fixing $T_R = 20$ msec. Regarding the T_O interval, we express its value as a function of the T_R value as follows:

$$T_O = m \cdot T_R + \Delta \ ,$$

where m is the maximum number of RELAY_DISCOVER messages a new node can send during a single observation period, and Δ is a guard time introduced to absorb jitter effects. In the following tests, we set $\Delta = T_R/2$, if not otherwise stated.

The second set of experiments we carried out aims at evaluating the total IP address configuration delay, say D_{conf}, which is defined as the time interval from the instant when the new node sends the first RELAY_DISCOVER message, and the instant at which it receives the DHCP_ACK message with the committed IP configuration parameters. The D_{conf} delay can be divided into two main components: D_{disc} and D_{assign}. The first component D_{disc} expresses the time between the first RELAY_DISCOVER message sent by the AH-DHCP client running on the new node and the election (through the unicast DHCP_DISCOVER message sent by the AH-DHCP client to the selected relay) of the DHCP relay agent acting as unique initiator of the address configuration process. It is

intuitive to note that $D_{disc} \geq T_O$. In general, D_{disc} will be longer than T_O only if node C has not received any RELAY_ACK message during the initial observation period, and it has to repeat the DHCP relay discovery procedure. The second component D_{assign} expresses the time between the DHCP relay activation and the reception of the DHCP_ACK message that concludes the IP address assignment. In other words, the D_{assign} value represents the duration of the DHCP transaction established between the AH-DHCP client and the legacy DHCP server. Several factors can affect this delay, including the processing delays introduced by relay agents and DHCP servers [PKLK04]. However, in a multi-hop WLAN system also the distance of the DHCP server from the new node plays a crucial role in determining the D_{assign} value. To estimate this component of the D_{conf} delay we performed several tests in the network scenarios illustrated in Figure 5.6. More precisely, we considered a single client C that is n wireless hops far from the access point A. Thus, at least $n-1$ relays are needed to establish this n-hop path between C and A. Obviously, each wireless hop adds its own medium access delay, processing delay and queuing delay. Similarly to the results shown in Figure 5.5 we performed our tests both without background traffic and with background traffic. In this case, the background traffic consists of $n-1$ asymptotic TCP flows opened from each relay to the gateway.

Figure 5.7(a) and Figure 5.7(b) show the IP address configuration delay without and with background traffic, respectively, as a function of the T_O value and for different n values. As expected, there is a clear dependence of the total configuration delay on the duration of the observation period, because D_{disc} increases almost linearly with T_O (graphs are omitted due to space limitations). Moreover, the D_{conf} value increases by increasing the number of hops needed to reach the gateway. This delay increase is not significant in the experiments without background traffic, while it is considerable with background traffic. This is due to the increment of queuing delays caused by the TCP packets that are buffered in the transmission queues of DHCP relay nodes. In fact, without background traffic, the network contention induced by control messages (i.e., OLSR and DHCP packets) is negligible and the transmission buffers are empty most of the time. Consequently, most of the delay accumulated along the path is due to the processing delays introduced by DHCP relay agents. On the contrary, with background traffic, the transmission buffers may store several

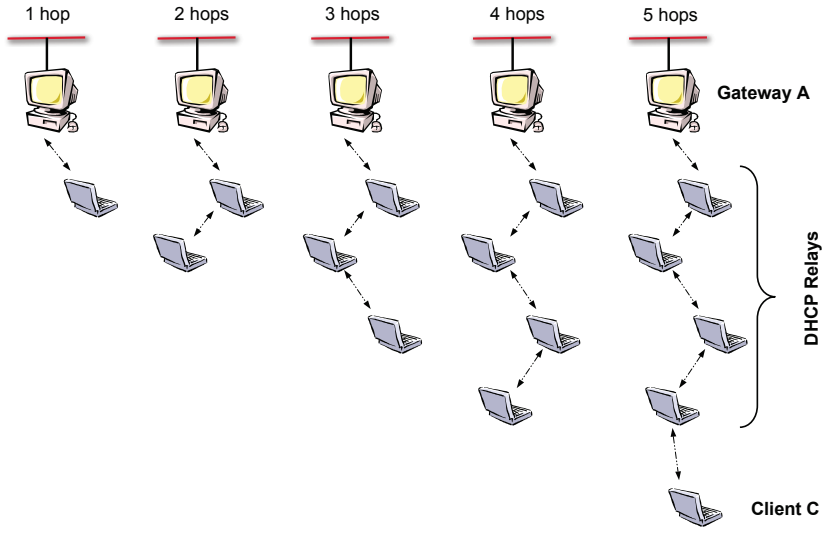
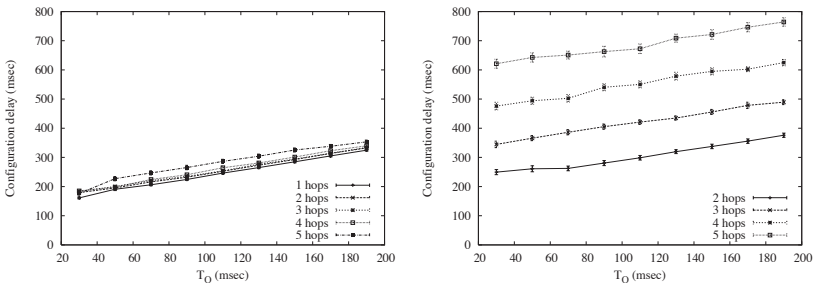


Figure 5.6: Network layouts used for the measuring address configuration delays in static configurations.

TCP packets. However, the experimental results show that the proposed autoconfiguration protocol ensures reasonably small address configuration delays (shorter than 0.8 sec) even when the new joining node is distant five hops from the gateway, and the network is fully loaded.

5.7.2 IP Address Configuration Delay in Mobile Configurations

In this section we evaluate the impact of node mobility on the total address configuration delay. To this end, we consider three different network scenarios, which are illustrated in Figure 5.8(a), Figure 5.8(b) and Figure 5.8(c). Specifically, Figure 5.8(a) represents the case of a new node C with a single neighboring AH-DHCP relay, say R_A, which is two hops far from the closest gateway G_A. Thus, after the DHCP relay discovery phase node C necessarily selects R_A as the unique initiator of the address configuration process. However, before completing the IP address assignment, node C moves out of node R_A's radio range. In this case, the polling mechanism allows a prompt detection of this event because node C stops receiving POLL_ACK messages from R_A. However,

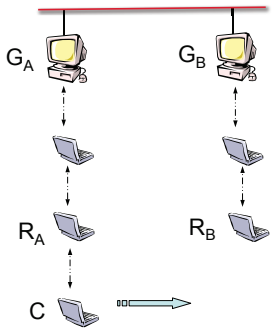


(a) without background traffic (b) with background traffic

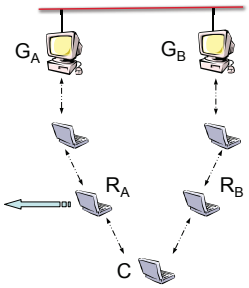
Figure 5.7: IP address configuration delay in static chain topologies.

node C's *relay_cache* is empty and it has to trigger a new DHCP discovery phase do find another neighboring relay (i.e., node R_B). In Figure 5.8(b) we illustrate a different case, because node C has now two neighboring AH-DHCP relay agents, both two hops away from a gateway. Therefore, after the DHCP relay discovery phase, node C will select randomly one of the two equivalent relays (R_A in our example) to start the address configuration process. Before completing the IP address assignment, node R_A moves out of node C's radio range. However, node C's *relay_cache* is not empty (it contains also the identity of relay R_B). Thus, node C can immediately start a new address configuration procedure. Finally, Figure 5.8(c) illustrates a network scenario identical to Figure 5.8(b), but in this case the mobile node is the intermediate node between R_A (the relay selected by node C in our example) and the gateway G_A. Since node R_A has lost its connectivity with the gateway, it stops replying to node C's polls. Note that R_A becomes aware of the topology change only when its link to the intermediate node expires[4]. After loosing its relay, node C will behave exactly as in the case illustrated in Figure 5.8(b). For the sake of brevity, hereafter we denote the first scenario as *Scenario A*, the second one as *Scenario B*, and the last one as *Scenario C*. In the following we report experimental results obtained by setting the period of RELAY_POLL messages equal to 50 msec, and the maximum number of consecutively missed POLL_ACK messages needed to declare a failed polling equal to four. This means that about 200 msec are needed by the polling scheme to declare lost a DHCP relay. Note

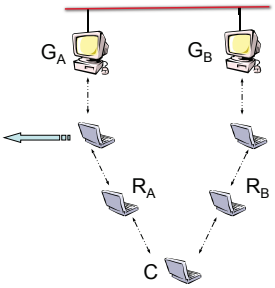
[4]In our experiments, we configured OLSR to declare lost a link after 500 ms passed without receiving any OLSR message.



(a) Scenario A: new node C moves during DHCP transaction



(b) Scenario B: AH-DHCP relay agent R_A moves during DHCP transaction



(c) Scenario C: Intermediate node moves during DHCP transaction

Figure 5.8: Network layouts used for the measuring address configuration delays in mobile configurations.

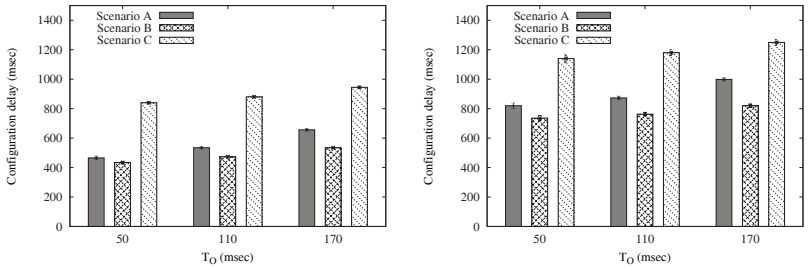
that the T_A timeout is set to 3 seconds in our experiments, but the T_A value does not affect the system performance for the considered mobility scenarios, where either the selected relay or the new node move while the other relays in the network are static.

Figure 5.9(a) and Figure 5.9(b) show the IP address configuration delays for all the three scenarios, without and with background traffic, respectively. Background traffic consists of two asymptotic TCP flows, one from node R_A to gateway G_A, and one from node R_B to gateway G_B. We set $T_R = 20$ms, as in Section 5.7.1, and we investigated three representative values for the T_O parameter. In our tests, each mobile node (i.e., node C in Scenario A, node R_A

in Scenario B, and the intermediate node between R_A and G_A in Scenario C) is configured to start moving after the completion of the DHCP relay discovery phase. Randomness is introduced in our experiments by inserting a random delay (uniformly selected in the range $[50, 100]$ msec) between the completion of the DHCP relay discovery phase and the beginning of node's movement. As a result of this randomization, the DHCP transaction can be interrupted either before node C receives a DHCP_OFFER message or before it receives the final DHCP_ACK message.

The experimental results shown in Figure 5.9 indicate that, in the considered network scenarios, the address configuration delays are acceptable (always less than one second in Scenario A and Scenario B) and the polling mechanism ensures a prompt detection of relay unavailability. We can observe that in Scenario A the configuration delay is longer than the one measured in Scenario B. To explain this behavior we should note that, in the former case, node C has to perform at least two DHCP discovery phases, while in the latter case one DHCP discovery phase may be sufficient, because in Scenario B the *relay_cache* contains the identity of both node R_A and node R_B. Consequently, the longer the T_O interval, the more significant the delay difference between Scenario A and Scenario B. Regarding Scenario C, we can observe that the configuration delays are significant higher than in the other two cases. The reason is that R_A becomes aware of the movement of the node that it is using as next-hop towards the gateway only after the OLSR link timeout. In our tests, this timeout is set to 500 ms, which corresponds to the difference in configuration delays between Scenario B and Scenario C. However, this additional delay is independent of our address configuration process, and it is only related to the dynamics of the ad hoc routing protocol.

Our experimental measurements show that background traffic negatively affects the address autoconfiguration process, which is an expected result that reproduces the behaviors observed also in static configurations (see Figure 5.7). Finally it is worth pointing out that the use of a temporary *addr_cache* helps to reduce the configuration delays in case of mobility, especially for Scenario B. More precisely, if the DHCP transaction is interrupted after node C has received a DHCP_OFFER message from a DHCP server, node C can resume the DHCP transaction by sending a new DHCP_REQUEST message for the same IP parameters (which are stored in the *addr_cache*) through a new relay (see



(a) without background traffic (b) with background traffic

Figure 5.9: IP address configuration delay measured in mobile scenarios.

Figure 5.2). This optimization avoids replicating the entire DHCP transaction after each topology change.

5.7.3 AH-DHCP protocol overheads

In previous sections we focused on estimating the address configuration delays. However, another performance figure particularly relevant for an autoconfiguration protocol is the amount of protocol overheads generated. To evaluate this aspect we analyzed the size of all the packets transmitted and received from the new node when joining the multi-hop WLAN. Then, we classified the protocol overheads into three categories representing the packets generated and received during the DHCP relay discovery phase, the DHCP transaction and the DHCP relay polling. Figure 5.10 and Figure 5.11 show the protocol overheads in terms of bytes generated by/received from the end host for a few representative cases chosen from the network scenarios illustrated in Figure 5.6 and in Figure 5.8, respectively. On the other hand, Table 5.3 and Table 5.4 report the protocol overheads in terms of number of packets generated by/received from the end host.

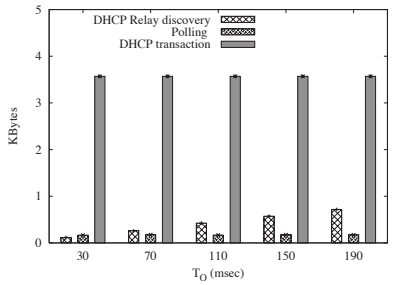
Our results indicate that, when analyzing the overheads in terms of bytes, the DHCP messages exchanged during the DHCP transaction are the dominant protocol overheads, and that the overall AH-DHCP overheads are practically negligible (less than 1 Kbyte). This can be explained by observing that the payload of AH-DHCP control packets (i.e., RELAY_DISCOVER, RELAY_ACK, RELAY_POLL and POLL_ACK messages) are 44-byte long (28 bytes for the IP and UDP headers plus 16 bytes for the payload listing the node's at-

tributes), while DHCP packets, in our DHCP version, are either 1472-byte long (DHCP_DISCOVER and DHCP_REQUEST messages) or 300-byte long (DHCP_OFFER and DHCP_ACK messages). On the contrary, if we analyze the protocol overheads in terms of packets, we can observe that the overhead associated to the DHCP transaction is the smallest one, only four packets, and it is independent of both the specific setting for AH-DHCP parameters, and the network topology. However, it is worth pointing out that DHCP messages are replicated on each wireless hop they traverse on the path between the selected relay and the closest gateway. This means that to compute the overall protocol overheads due to DHCP messages, the overheads reported in Table 5.3 should be multiplied by n, where n is the hop distance between the selected relay and the closest gateway. In any case, the DHCP server will receive only one copy of each DHCP message.

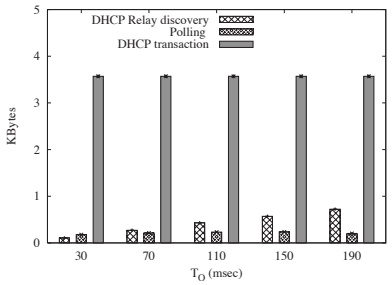
As expected, the AH-DHCP overheads generated during the DHCP relay discovery phase depend on the T_O value. Specifically, the longer the T_O value, the more RELAY_DISCOVER messages are generated, and the more RELAY_ACK messages are received by node C. For instance, let us consider the case $T_O =$ 70 ms in Table 5.3. Since $T_R = 20$ ms, node C sends three RELAY_DISCOVER messages and, in principle, it should receive three RELAY_ACK messages[5]. In our test conditions, the link quality is good and messages are rarely lost due to channel noise. Thus, the measured overhead is very close to the expected value of six packets. Note that, with background traffic the overhead increases rather than decreasing. The explanation of this behavior is that a single DHCP relay discovery phase is not always sufficient to node C to discover its DHCP relay.

As shown in Figure 5.10 and Figure 5.11, the AH-DHCP overheads generated during the DHCP relay polling are independent of the T_O value, but are affected by the presence of background traffic and the number of hops between the new node and the gateway. This is easily explained by noting that the duration of the DHCP transaction (i.e., D_{assign}) increases when the distance between the new node and the gateway increases, especially if background traffic disturbs the DHCP transaction (see Figure 5.7). Thus, the longer D_{assign}, the more RELAY_POLL messages are generated, and the more POLL_ACK messages are received by node C. From the shown results it is evident that AH-DHCP

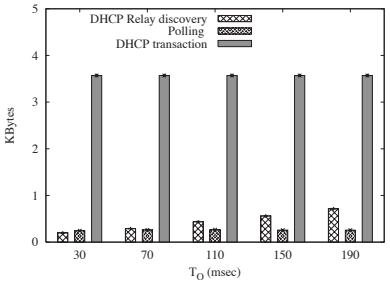
[5]Node C may receive less than three RELAY_ACK messages either because the RELAY_ACK messages are lost due to channel noise/ contention, or because the relay did not receive the RELAY_DISCOVER message.



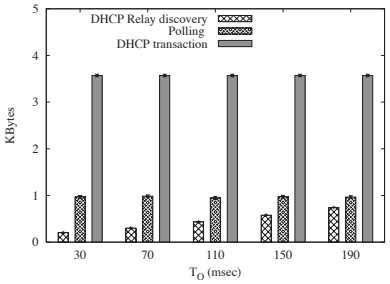
(a) 2-hop chain without background traffic



(b) 5-hop chain without background traffic



(c) 2-hop chain with background traffic



(d) 5-hop chain with background traffic

Figure 5.10: AH-DHCP protocol overheads (in bytes) for the network scenarios illustrated in Figure 5.6.

overheads in terms of packets are significantly higher than DHCP overheads. However, this overhead can be reduced by adjusting the repetition periods of RELAY_DISCOVER and RELAY_POLL messages. In addition, the number of generated messages is quite low and it is reasonable to believe that it has no negative impact on the access delay of data packets.

Similar considerations can be derived by analyzing Figure 5.11. The main difference we can notice is that the protocol overheads generated by the DHCP transactions in Scenario A are higher than the ones generated in Scenario B and Scenario C. This can be explained by observing that in Scenario B and Scenario C, if the mobile node moves after node C has received a DHCP_OFFER message, then node C can resume the DHCP transaction by sending a new DHCP_REQUEST message directly to relay R_B. On the contrary, in Scenario A node C has always to restart a completely new DHCP transaction after losing the radio visibility with relay R_A. Therefore, a higher number of DHCP mes-

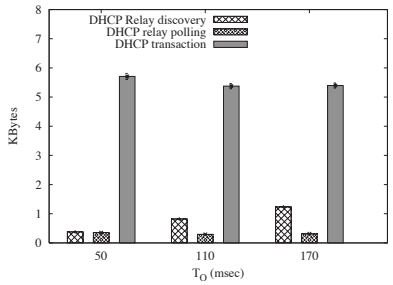
Table 5.3: AH-DHCP protocol overheads (in packets) for the network scenarios illustrated in Figure 5.9.

T_O		2-hop chain		5-hop chain	
		w/o bck. traffic	with ack. traffic	w/o bck. traffic	with ack. traffic
70 ms	DHCP relay discovery	5.95	6.72	5.96	7
	Polling	4.08	6.11	4.95	18.2
	DHCP transaction	4	4	4	4
150 ms	DHCP relay discovery	13.02	13.29	13.26	13.40
	Polling	4.02	6	5.47	20.4
	DHCPtransaction	4	4	4	4

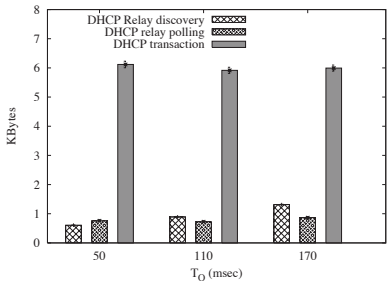
Table 5.4: AH-DHCP protocol overheads (in packets) for the network scenarios illustrated in Figure 5.11 for $T_O = 70$ ms.

		Scenario A	Scenario B	Scenario C
DHCP relay discovery	w/o back. traffic	19.1	13.52	13.6
	with back. traffic	20.6	14.22	14.27
Polling	w/o back. traffic	6.8	5.95	16
	with back. traffic	16.8	11.8	21.86
DHCP transaction	w/o back. traffic	6.02	5.64	5.75
	with back. traffic	6.62	5.76	5.90

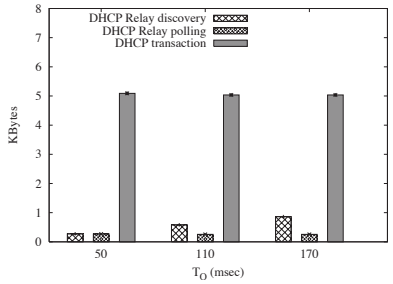
sages are generated in Scenario A than Scenario B, resulting in higher protocol overheads. However, as reported in Table 5.4, less than two DHCP messages have to be retransmitted, on average, to complete the DHCP transaction. In addition, we can observe that the polling overhead is maximum for Scenario C because relay R_A keeps replying to the RELAY_POLL messages until the ad hoc routing protocol does not declare lost its connection to the gateway. On the other hand, Scenario A has the highest overhead for the DHCP relay discovery phase because at least two separate discovery procedures are necessary to discover relay R_A and relay R_B, while in both Scenario B and Scenario C the two relay are discovered during the first initial DHCP discovery phase.



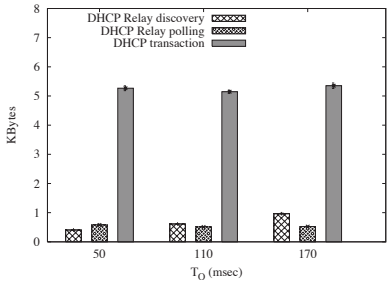
(a) Scenario A, without background traffic



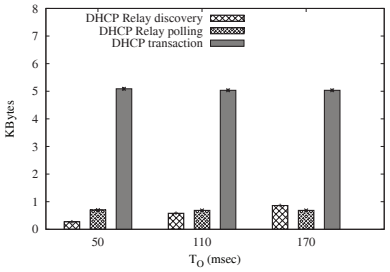
(b) Scenario A, with background traffic



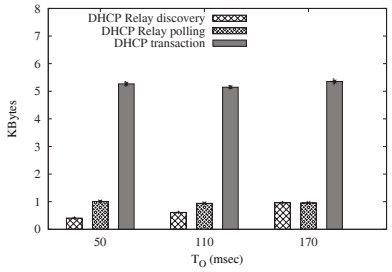
(c) Scenario B, without background traffic



(d) Scenario B, with background traffic



(e) Scenario C, without background traffic



(f) Scenario C, with background traffic

Figure 5.11: AH-DHCP protocol overheads (in bytes) for the network scenarios illustrated in Figure 5.9.

5.8 Conclusions

In this chapter we described AH-DHCP, an address autoconfiguration protocol for multi-hop WLAN. The main goal of our work was to prove the applicability of DHCP, originally designed to provide configuration parameters to hosts in a

fixed network, also when traditional WLANs integrate ad hoc networking technologies to discover and maintain multi-hop wireless path within the network. The basic idea was to take advantage of DHCP relay capabilities available in already configured nodes. To this end, we proposed extensions to DHCP to enable a new node to dynamically choose a reachable relay agent as the unique initiator of the configuration procedure. Then, this relay transparently passes all the client-originated messages to the DHCP servers located in the wired part of the network. Our proposed solution can tolerate messages losses and node mobility because it implements appropriate mechanisms to promptly react to persistent communication problems and topology changes.

Experiments conducted with a prototype implementation of AH-DHCP have shown that our solution ensures short address configuration delays and low protocol overheads, even when node mobility or background traffic interferes with the operations of the autoconfiguration protocol. For future work, we intend to investigate mechanisms to reduce the impact of multi-hop forwarding on address assignment delays in large-scale multi-hop WLANs, e.g., by introducing a hierarchy of DHCP relay agents. Another possible research direction is the extension of our solution to IPv6.

Chapter 6

Mesh Networks: An Application Scenario

6.1 Introduction

Today's modern society is considered an Information society because the creation, circulation and manipulation of information are activities that pervade many aspects of our cultural, economical and social life. Consequently, governments, economy and society in general, are becoming increasingly dependent on *Information & Communication Technologies* (ICT), which are the means of providing information. For these reasons, the communications infrastructures used to transport information are considered a critical asset of our society, such as the transportation and power supply infrastructures, and they should be protected and secured. The need to ensure resiliency, security and dependability of our communications systems is made more compelling by the tight interdependence between the information infrastructure and other critical infrastructures. For instance, security problems, breakdowns and failures in the information systems may create widespread damage in transportation or energy infrastructures. In addition, the nature and extent of the threats jeopardizing our communications infrastructures are considerable higher today than in earlier times. As well explained by the European Security Research Advisory Board in its 2006 report "*modern crises are progressively changing their character from 'predictable' emergencies... to unpredictable catastrophic*

events" [ESA06], and current communications networks are not designed to withstand unplanned and unexpected disruptive events such as natural or man-made disasters. In fact, in assessing the communication breakdowns that have taken place in the aftermath of events of the magnitude of 9/11, Katrina hurricane or London bombings, when many mission-critical networks were down and unavailable, it has been observed that "*telecommunications was the greatest single area of concern*" [Lon06, UK 06]. It is also important to highlight that, during a crisis or an emergency situation, the availability of a reliable and dependable communications system is also fundamental to allow first responders, rescue teams and public safety agencies operating in the disaster area to carry out disaster relief operations. In fact, all the disaster and crisis management activities rely on the exchange of information between government entities, operators of critical infrastructures, and rescue teams, as well as on the interaction of first responders with citizens and victims. In the following discussion, we will primarily concentrate our attention on this communication scenario, i.e., the provision of resilient and flexible communications services in a disaster zone for Public Protection Disaster Relief (PPDR) missions.

The experiences gathered after the most recent large disasters (e.g., Indian Ocean tsunami in 2004) or massive terrorist attacks (e.g. 9/11 airplane crashes in 2001 or Madrid train bombings in 2004) have permitted the clear identification of the missing capabilities of existing communications systems to provide the necessary support for PPDR applications. Among the most important shortcomings that have been identified by various forums and committees [Lon06, UK 06, US 06, US 05, Hat05] it is useful to note: the lack of sufficient robustness and resiliency to disruptive events, the limitations in the interoperability between private networks operated by public safety agencies, the difficulties for integrating private networks with the core communications infrastructures, the lack of flexibility and versatility in the communications services, and the limited support of priority communications in public networks. To effectively address the above issues, we advocate the use of self-organizing architectures exploiting the ad hoc networking paradigm to realize a resilient and versatile communications system meeting the requirements of a disaster response system. Traditionally, mobile multi-hop ad hoc networks (also MANETs) are conceived as groups of devices that self-organize into peer-to-peer networks by establishing multi-hop wireless connections [Ahm07, CCL03].

Therefore, it is intuitive that first responders may use the ad hoc networking technologies to quickly set up on-demand communications services between their handheld devices, enabling a reliable dissemination of vital information, as well as an effective collaboration in time-critical relief operations. However, in the recent years, the MANET research has achieved important results in successfully exploiting the multi-hop ad hoc networking to build various types of specialized networks, such as mesh networks, vehicular networks, sensor networks and opportunistic networks, which have been designed to support well-defined application requirements [CG07a]. For instance, mesh networks provide rapidly deployable wireless extension to legacy communications infrastructures; vehicular networks apply the MANET technology to the inter-vehicles communications; sensor networks are designed to support monitoring applications in general; and opportunistic networks are an extension of MANET technology to cope with intermittently connected networks. We expect that these emerging technologies will provide most of the missing communications capabilities needed to develop a dependable, secure and rapidly deployable communications system for mission-critical scenarios and emergency response.

In this chapter we present the main characteristics and properties of these emerging technologies with special emphasis on mesh, vehicular, sensor and opportunistic networks ([Anc07]). The focus of our discussion is to explain how these networking solutions will facilitate the development of flexible and easily deployable communications systems that would be resilient to disruptive and unplanned events. While the maturity of these technologies is sufficient to predict the readily deployment in all the typical situations characterizing PPDR scenarios, there are still several open research and technical challenges that have to be addressed to realize an information sharing system for disaster response fully integrated with the existing communications infrastructures. In particular, in our discussion we will give special attention to aspects such as interoperability among multiple heterogeneous networks, autonomic network management, and QoS protection.

The remaining of this chapter is organized as follows. Section 6.2 illustrates the reference disaster scenarios that exemplify the communications challenges that characterize first responders' emergency response operations. In Section 6.3 we analyze the missing technological capabilities necessary to develop the next-generation of resilient, rapidly deployable and secure commu-

nications systems for PPDR applications. In Section 6.4 we outline the most consolidated international initiatives aiming at promoting the security research in the PPDR area. Section 6.5 reviews the most recent advances in the deployment of mesh, opportunistic, vehicular, and sensor networks. In Section 6.6 we discusses some of the most important research challenges. Finally, Section 6.7 draws concluding remarks.

6.2 Background

To identify the communications challenges that emerge after a security incident, and to highlight the communications capabilities needed during disaster relief operations, we consider a reference scenario, where a natural or manmade disaster devastates the communications infrastructures and first responders are involved in the emergency response.

First of all, we observe that the today public telecommunications networks are characterized by the considerable heterogeneity of the technologies and architectures adopted to provide communications services, either at the local or geographical scale. At one extreme, these networks are based on wired and wireless narrowband technologies (e.g. leased telephone lines, cellular and satellite systems, etc.), and they are mainly used to provide voice communications and a limited support of data transmissions. On the other extreme, these networks employ broadband wired and wireless technologies (e.g., WiFi, Wi-MAX, optical networks, etc.) to support more complex multimedia communications. However, these systems have common characteristics such as the dependence on dedicated infrastructures, the adoption of a centralized management for the communications resources, and the use of point-to-point links to interconnect the devices to other devices or control units. In case of an incident that causes partial damages to the network infrastructures (either turning some point-to-point links down or making some devices non functioning), large portions of these communications systems may stop working properly. To reduce the risk of suffering communications-service interruptions during a disruptive event, the most critical components of large-scale telecommunications networks are usually replicated. However, the experiences gathered from the most recent security incidents and disasters (e.g. 9/11 attacks or Katrina hurricane) have highlighted that this approach is not effective to ensure communications

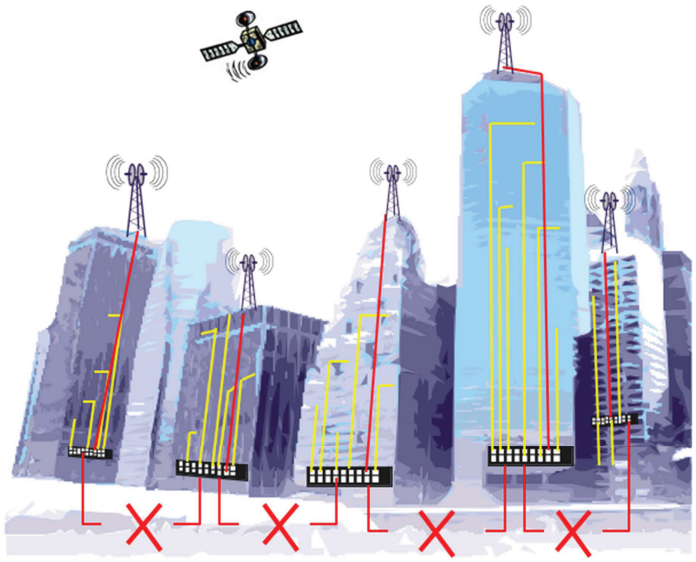


Figure 6.1: Communications infrastructure partially damaged: backup wireless links are established to activate *mesh-mode* communications.

system resiliency because these backup systems are generally unable to handle the huge traffic volumes generated in the wake of a crisis situation. The solution we envisage for dealing with the damages that an incident may cause to the legacy communications systems is to reuse what remains available of the infrastructure by establishing additional wireless backup links if possible (e.g., satellite links), and substituting point-to-point links with multi-hop wireless connections to form a more reliable wireless mesh backbone. This scenario is illustrated in Figure 1, which exemplifies an urban environment where an incident has interrupted wired links (represented by red crosses in the picture), and communicating devices establish alternative wireless links using satellites or terrestrial antennas.

In addition to re-establish the public communications systems in a disaster area, it is fundamental to rapidly deploy a communications platform that may guarantee an acceptable level of communication to first responders, rescue

workers, and any other Public Safety user operating in the disaster area. This temporary on-demand communications network may be created by establishing multi-hop ad hoc communications between the handheld devices carried by first responders and/or communicating devices (i.e., wireless routers) transported by rescue land vehicles or helicopters deployed on the disaster area. These specialized networks may be operated in parallel to the legacy networks or tightly integrated with them as an extension or replacement of a too seriously damaged communications infrastructure (see Figure 2). Note that, for first responders, it is necessary to have also access to the legacy wireless infrastructure networks to stay in contact with remote command and control centers.

In addition to deploying powerful wireless communications devices, the emergency response personnel may spread out across the disaster area tiny sensing devices. These sensing devices will form a sensor network that may provide a useful tool to remotely monitor a location or situation in real time, assisting first responders in the decision process and coordination activities during emergency response and security operations, as well as to detect and predict threats (e.g. the presence of toxic substances after a chemical plant explosion, or the imminent collapse of a building after and earthquake).

In extreme cases, a disruptive event may produce so extensive damages to bring down almost all the existing network infrastructures. Moreover, because of the prohibitive environmental conditions, it might be impractical to spread around a sufficient number of rescue vehicles so as to create well-connected ad hoc networks. In this context it is more likely to envisage the case of "*clouds*" of connected handheld devices (e.g., palmtops carried by first responders) that will be just sporadically connected to each other, and, possibly, to the surviving part of the infrastructure. These communication clouds will be extremely dynamic, as the rescue teams will move, and wireless links will appear and disappear. In the extreme case, a single, disconnected, user can form a communication cloud. Traditional networking approaches will fail to preserve the communications services in such scenario because they require a continuous end-to-end path between communicating endpoints, computed by a routing protocol, while such continuous paths will seldom be available in a security incident area. On the contrary, opportunistic networking techniques enable end-to-end paths even when communication endpoints are not connected at the same time to the same network by exploiting the *store-carry-and-forward*

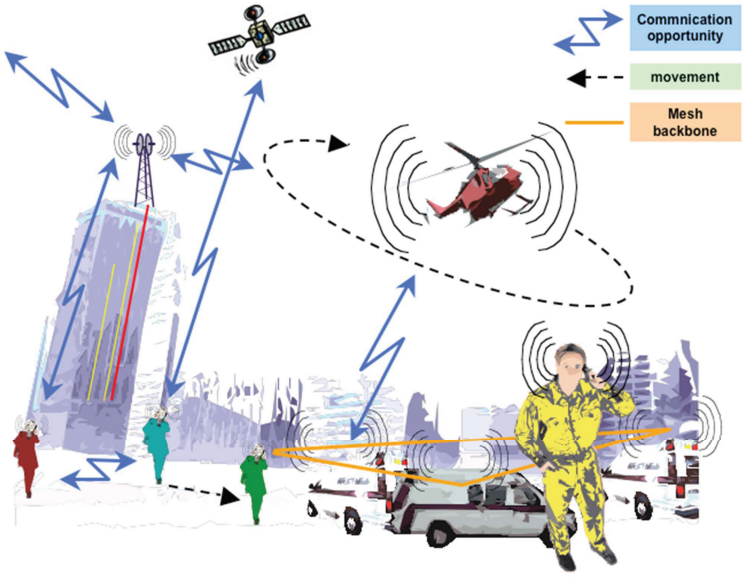


Figure 6.2: Communications infrastructure heavily compromised: heterogeneous and interoperable self-organizing wireless networks are deployed.

approach. It is evident that devices should have highly versatile communications capabilities to efficiently operate in a network that would be extremely dynamic, heterogeneous and mainly disconnected, formed by possibly isolated devices. In addition, in this disaster scenario, where communications will be extremely challenged as a consequence of infrastructure disruptions, communications opportunities will be a scarce resource to be sparingly managed. It is then critical to ensure that critical data are made available to the right set of users, by avoiding congestion and data unavailability.

6.3 Thoughts for practitioners

From the analysis of the previous reference scenario, as well as the analysis of other global and homeland security scenarios, we can identify the user requirements associated to typical public safety, emergency and disaster ap-

plications. These user requirements will be the basis to derive the technical requirements for the design of resilient, rapidly deployable and secure communications systems for PPDR applications [MHB07]. The most important technical requirements we have identified are the following:

1. *Ubiquitous access* – Public safety mobile radio networks must function in all areas served by first responders and involving disaster victims. This should include underground places, rural areas, remote or under-served areas, and challenged environments that were subjected to devastations. In addition, the seamless support of user mobility should be an integral part of the system design.

2. *Resiliency* – Natural and manmade disasters may cause partial, or even extensive, disruptions of the terrestrial communications infrastructures. However, a resilient communications system must be designed to survive to damages and failures, and to ensure the continuity of communication services, at least for critical applications. To this end, centralized architectures should be avoided because more prone to failures and clearly less re-configurable.

3. *Fast deployment* – To effectively deal with emergency situations a communications system for PPDR applications should be easily and rapidly deployable, and the communications services should be operational very quickly.

4. *Self-organization* – It is crucial that public safety networks implement advanced self-management capabilities in order to limit as much as possible human operations and maintenance, guaranteeing that the network properly operates despite unplanned and unexpected events. Self-organization is also a prerequisite to provide fast and dynamic deployment of temporary, on demand, communications network in disaster areas.

5. *Interoperability* – Emergency operations require the involvement of several groups of first responders operating for different agencies and authorities. Seamless communications between different units do not require only common procedures, but also interoperable equipments and communication protocols. In addition, private networks owned by public

safety agencies should be easily integrable with the public networks used by citizens to favor the information collection and distribution.

6. *QoS* – Emergency response management and disaster relief operations very often rely on the timely exchange of critical information (e.g. via voice or images/video) between first responders, and on providing correct and updated information to people. Therefore, the communications system used by first responders should provide QoS support to meet the stringent requirements of real-time flows. In addition, priority schemes should be integrated in the public communications networks to ensure that vital communications for first responders are not hindered by legacy data transmissions during emergency situations.

7. *Security* – Standard security properties should be assured also in a disruptive environment. However, in addition to protecting the privacy of the communications, in emergency scenarios it is also important to provide a reliable establishment of trust relationships among users in order to guarantee the secure identification of devices and users.

Although the technical requirements for reliable communications infrastructures to be used in PPDR operations are well defined, the recent disaster experiences have revealed that the existing solutions are unable to provide an adequate support for these situations. Traditionally, public safety agencies have relied on dedicated wireless systems to support communications between teams of first responders. In particular, it was generally believed that the reliability and security of the public Internet is inadequate for mission-critical functions. On the contrary, the allocation of dedicated spectrum for public safety applications, as well as the adoption of more stringent reliability and security requirements than the ones considered in commercial networks, should make dedicated systems sufficiently robust to operate also during emergency situations. For these reasons, industry standards for implementing narrowband private mobile radio systems, e.g. ETSI standard TETRA in Europe or APCO25 in the USA, have been developed in last decade, facilitating the deployment of these networks. However, a central lesson underscored by recent disruptive events (e.g. Katrina hurricane or London bombings) is that private mobile radio systems maintained by public safety agencies were outdated and incompatible [Lon06, UK 06, US 06, US 05, Hat05]. Specifically, these ag-

ing technologies were too limited to meet the growing demands of emergency communication services, because they were designed primarily for voice communications and lack other important capabilities such as high-speed data communications. Moreover, teams of first responders from different agencies were not able to communicate due to lack of interoperability between their private networks. This severely hindered the capability of first responders to acquire, process, and disseminate vital information. In addition, the wireless communications systems used by the first responders and law enforcement communities were unable to support seamless and interoperable communications with the legacy telecommunications networks used by citizens. This made impossible to distribute early warnings and updated information to people at disaster areas.

The inefficiencies in the design or deployment of their private networks led first responders and emergency managers to switch to public mobile networks to provide emergency services during large-scale disasters. However, terrestrial communications infrastructures (also called Land Mobile Radio systems, or LMR), such as traditional 3G cellular systems or emerging metro-scale broadband wireless access technologies, are generally based on centralized architectures where central units have full control over each cell. Thus, fundamental system functionalities, as access control, connection establishment, support of mobility, etc. relay on the existence and the availability of the network infrastructure itself. Consequently, centralized architectures suffer the main drawback of collapsing when the centralized infrastructure is out of order, and when unplanned or unexpected disruptive events occur. For example, disasters as New Orleans flooding destroyed all available network infrastructures. Nowadays, the only practical solution to deal with partial or total unavailability of LMR systems is to use satellite communications. However, satellite systems are seen as a fallback technology, suitable only for outdoor communications and subject to the availability of a satellite to act as a relay station between earth terminals. The lack of radio communications ability within buildings represented a notable failing of public safety LMRs and one that has led to tragic results during emergency situations such as 9/11 [UK 06, US 05].

Even if available, commercial telecommunications systems often were severely overloaded during emergencies. All the reports from governments and experts investigating the causes of the communication failure during recent natural or manmade disasters highlighted that commercial systems are often the most un-

reliable during critical incidents when public demand overwhelms the system [Hat05]. Unfortunately, prioritization schemes to reserve dedicate resources to emergency calls or to limit resource usage by low priority users are rarely implemented in commercial systems, or have not appropriate objectives. In fact, if congestion occurs in normal conditions, network operators assign greater importance to flows that have greater revenue-generating capability. On the contrary, during exceptional conditions like emergencies or disasters, network operators should consider more valuable the traffic generated by users involved in disaster relief operations.

The above analysis of the shortcomings of the existing, either public or private, communications systems for PPDR applications points out that the development of new networking technologies capable of providing the needed degree of reliability and dependability is fundamental. This need, as well as the growing threat perception, has boosted both private and public investments in researching novel security solutions. As explained in the following, these research initiatives have rapidly converged to an increasing consensus about the fact that the most mature and best suited networking paradigm fulfilling the requirements of PPDR applications is the ad hoc networking paradigm [ESA06]. In fact, being peer-to-peer networks formed by mobile devices with self-organizing capabilities, multi-hop ad hoc networks represent a key technological driver to deploy more resilient communications systems. To support this claim, in the following sections we firstly outline the most important national and international research programs that have been established in the sector of national and civil security, with special attention to the communications concerns. Then, we discuss how the recent advances in ad hoc networking may be successfully applied to realize a practical communications system for PPDR applications.

6.4 International Initiatives

A series of national and international initiatives have been established to bring together national governments, international organizations, industrial stakeholders, academia and emergency response communities and to set up the agenda of long-term security research. All these initiatives have identified the development of novel IT solutions to deploy dependable, versatile and secure

communications infrastructures, as a key investment area.

One of the first examples of this new approach to address global security challenges is represented by the establishment in the USA of the Department of Homeland Security (DHS), whose primary aim is to define a high-level strategic plan to coordinate all the organizations and institutions involved in the security missions and emergency response. To accomplish this ambitious goal, the DHS has created, among the others, the Directorate for Science and Technology (S&T Directorate) that aims at driving the development of technologies and capabilities in support of the homeland security. To this end a variety of agencies and programs have been established to promote the research on the security challenges identified by the DHS strategic plan. In particular, the SAFECOM program has been activated to improve interoperable communications nationwide through the definition of non-proprietary standards, open architectures, common operational procedures and communications systems ensuring interoperable voice and data capabilities for emergency response. In addition, the Homeland Security Advanced Research Projects Agency (HSARPA) is launching new solicitations and funding programs on a broad range of topics to promote the research and development efforts of innovative security solutions. In particular, HSARPA is now promoting the development of novel communications and information systems supporting more effective and coordinated decision-making processes and crisis management through reliable information acquisition and assessment. In this context, the development of more robust and flexible sensor networks is considered of paramount importance, because much of the security mission involves the monitoring of various environments, and the prediction and detection of threats to these environments.

Collaborative programs have been also established between Europe and USA for the coordination and development of joint specification of standards for Public Safety and Emergency (PS&E) scenarios. The most important example of these joint initiatives is the Project MESA (Mobility for Emergency and Safety Applications). Specifically, MESA is a standardization Partnership Project established between the European Telecommunications Standards Institute (ETSI) and the Telecommunications Industry Association (TIA) in the USA, whose original purpose was to elaborate a joint specification of next-generation mobile broadband technology to be deployed for the PS&E. Since 2002, the vision has evolved towards the definition of a set of interconnection

standards between heterogeneous systems, i.e., following the so-called 'systems of systems' approach.

Another relevant European initiative jointly funded by the European Commission and the European Space Agency, is GMES (Global Monitoring for Environment and Security). Since 2001, the GMES group is working on the implementation of European-level policies and information services dealing with environmental monitoring and security needs. The GMES approach is based on the observation and the understanding of the phenomena of the terrestrial environment through satellite and ground systems. This information is then provided to all the organizations involved in environmental management and security enforcement.

Although these programs have obtained important results, the European states felt the need to develop a longer-term perspective in the field of security research. For these reasons, in April 2005 it was created the European Security Research Advisory Board (ESRAB) to draw the strategic lines for European security research and to recommend the most adequate instruments to implement it. The key findings of ESARB [ESA06], and the experience formed with the Preparatory Action for Security Research (PASR, 2004-2006), have been taken into account in the definition of the Security theme in the 7th Framework Programme (FP7). Specifically, four priority missions have been identified: protection against terrorism and organized crime, border security, critical infrastructure protection, and restoring security in case of crisis [Eur06]. Then, from the analysis of the requirements of these security missions, the technology capabilities needed to meet these requirements have been identified, such as robust communications capabilities, improved situation awareness and interoperable command and control capabilities. For these reasons, ad hoc networking technologies, by providing decentralization, flexibility, reliability and adaptability as intrinsic features, should be key components of future communications systems for PPDR applications.

6.5 MASS Solutions for Public Safety Applications

The ad-hoc networking concept is not new, having been around in various forms for over 30 years. The initial development of ad hoc wireless communications

for military and tactical purposes can be dated back to 1972, when the DARPA agency initiated the Packet radio Network (PRN) program. The initial concept was then expanded in follow-up programs such as the Survivable Radio Network (SURAN) initiative in 1983 and the Global Mobile (GloMo) Information program in 1994. However, a real boost in the ad hoc networking research was given by the creation in 1997 of an IETF (Internet Engineering Task Force) working group, called MANET WG. The mission of this working group was to "standardize IP routing? protocol functionality suitable for wireless routing application" in multi-hop dynamic network topologies. A decade of intensive research in this filed has generated a considerable number of different routing algorithms, although only a few of them have been successfully deployed in real ad hoc networks. In parallel, several research projects in the area of mobile ad hoc networks had been lunched by academia. The extensive research activities conducted in the ad hoc networking field have developed both the theoretical and technical background for the deployment of multi-hop ad hoc networks [CCL03, CG07b]. However, despite the massive research efforts that have been dedicated to this field in the last two decades, it is quite recent the successful application of the ad hoc networking paradigm in real world applications that are appearing on the mass market. The explanation of this apparent contradiction is that, initially the research on MANETs adopted quite unrealistic assumptions: large-scale and totally decentralized networks capable of supporting any type of legacy TCP/IP applications. On the contrary, as discussed in [CG07a], the recent success of the ad hoc networking technologies is due to the adoption of a more pragmatic approach and the exploitation of ad hoc networking paradigm to extend the Internet and to support well-defined application requirements. Among the various classes of ad hoc networks that are under deployment, we believe that mesh, vehicular, sensor, and opportunistic networks are of particular interest and importance for PPDR scenarios, because they can be considered fundamental building blocks of the next-generation of dependable, and rapidly deployable communications systems for mission-critical scenarios. In the following we present an overview of the most recent advances in the design and the deployment of these emerging networks, and we discuss their relevance to the PPDR scenario.

6.5.1 Mesh networks

Mesh networks are hybrid MANETs, where dedicated nodes, namely mesh routers, communicating wirelessly through multi-hop paths construct a wireless backbone. The wireless backbone may have a (limited) number of connections with the existing wired infrastructure to provide a flexible and "low cost" extension of the Internet [BCG05]. Mobile/nomadic users obtain a multi-hop connectivity through the wireless backbone to communicate directly to each other, or to access the Internet via the closest mesh router. The use of multiple independent paths increases the availability and dependability of the wireless backbone through resilience to operational anomalies or security attacks. Therefore, the mesh technology can be used to rapidly deploy a high-capacity backbone in an area where the terrestrial infrastructures are partially collapsed, as shown in Figure 1.

The growing interest in mesh applications has boosted the industrial efforts to offer diverse wireless mesh solutions. Some vendors have focused on standard wireless technologies, such as IEEE 802.11 (aka WiFi) and IEEE 802.16 (aka WiMax) [EMSW02]. However, on top of the standard 802-based wireless connectivity, they adopted proprietary networking software solutions that cannot interoperate. For these reasons, various IEEE standardization groups are also actively working on including wireless mesh networking techniques in the specifications of wireless technologies. The most mature example of these standardization activities is the IEEE 802.11s working group, that is working to introduce advanced meshing capabilities in the WiFi technology [IEE06]. Another limitation of the existing solutions for building mesh network, as we will extensively discuss in Section 5, is the lack of reliable self-configuration procedures that can dynamically adapt to varying network conditions. Nevertheless the ability to use traditional wireless technologies, e.g. 802.11, for mesh networking, makes their development easier and less expensive. The RoofNet project at MIT [Bic05] demonstrated that it is possible to provide a city such as Boston, with broadband access with an 802.11b-based wireless network backbone infrastructure. Specifically, RoofNet consists of a limited number of nodes, positioned on roofs operated on a volunteer basis, which dynamically create the backbone and support mesh networking. Another example of real mesh application, which is relevant for our reference scenario, is the Quail Ridge Reserve Wireless Mesh Network project [CG07a], an effort to provide a wireless com-

munications infrastructure to a wildlife reserve. Aim of the project is to benefit on-site ecological research and to provide continuous and real-time monitoring of the environment. Finally, CalMesh, which is deployed on the UCSD campus and the San Diego County, is a specific example of an experimental mesh network for emergency and crisis scenarios, which provides first responders with a local network to communicate to each other and, in case, to the Internet [Dil07].

6.5.2 Vehicular Ad Hoc Networks

Vehicular Ad hoc NETworks (VANETs), are emerging as one of the most successful specializations of (pure) MANETs, which is expected to rapidly penetrate the market. Traditional VANETs use ad hoc communications for performing efficient driver assistance and car safety. In this sense, VANETs can be viewed as fundamental components of any Intelligent Transportation System (ITS) [You06, Tor05]. However, a vehicular network may be also used to perform efficient data distribution between vehicles and users during emergency situations as shown in Figure 2. Note that VANETs have a relevant advantage compared to traditional MANETs, as they rarely have constraints related to the devices' capacities (in terms of space, computation and power). VANETs necessitate a minimum of equipped vehicles for efficiently work, thus they did not fully exploited their potential benefits for civilian applications yet. However, VANETs research is pushed by both industrial and government organizations. Thus, VANET systems are one of the fields where MANET research can achieve its full potential. Examples of this effort can be found in projects such as the European FleetNet. In FleetNet, vehicles exchange short messages with local information. The messages inform the drivers about obstacles or traffic jams ahead, beyond the view of the driver's vision or of the vehicle sensors. Additional projects, such as the European Project CarTALK 2000 exploited the development of cooperative driver assistance systems and the development of self-organizing ad hoc radio network as a communication basis with the aim of preparing a future standard. CarTALK uses both direct and multi-hop communications for the data transfer, empowered with position and spatial awareness. Similarly, in the US, several projects are involved in this area, in some cases integrating VANET into a broader view, including mesh or grid networking as in VMesh/VGrid or the PORTAL project. There

is also a large involvement from the military and, since 2004, DARPA sponsors the Urban Challenge, where fully autonomous ground vehicles must conduct simulated military supply missions in an urban area. It is evident that all the knowledge developed in these projects will be useful also for the development of VANETs in emergency and crisis scenarios, when the equipments forming the vehicular network are transported by rescue land (e.g. trucks) or flying (e.g., helicopters) vehicles. For instance, in [Roc07] an inter-vehicular communication system is described, which is able to quickly discover and transmit real time multimedia information from around a crisis area to approaching first responders' vehicles.

6.5.3 Sensor networks

Among ad hoc networks, wireless sensor networks have a special role. The aim of a sensor network is to collect information about events occurring in the sensor field. To this end, sensor nodes, which are tiny, low-power and low-resources communicating devices with sensing capabilities, are deployed in the monitoring area, and the information collected by sensor nodes is generally delivered to collecting centers, also called sinks, by exploiting a wireless multi-hop ad hoc network. In some applications the retrieval of sensors' readings can be implemented in a more efficient way by introducing mobile nodes inside the network (e.g., robots) that move inside the sensors field collecting the information from sensor nodes via ad hoc wireless communications and then move close to the collecting center for delivering the sensed data. Alternatively, the sink node can move in the sensor filed (e.g., unmanned helicopters flying over the sensor) collecting data from each sensor node. In addition, the robots (actuators) can be used not only to collect data but can also be able to perform actions on the sensor field depending on the detected events. For example, a robot can be used to remove explosives. Therefore, sensor and actuators networks can be successfully applied in several security scenarios. In military and tactical contexts, one of the major applications of sensor network is considered the target localization and target tracking. To this end, a variety of different physical measurements have been developed to detect the target presence and its position [LH07]. In parallel, many sensor networks have been developed for civilian applications, mainly for habitat and environmental monitoring. A very famous example of this type of applications is the Great Duck Island Habitat Monitoring project, a

collaborative project between Intel and the University of California at Berkeley to deploy a sensor network on Great Duck Island, Maine, for monitoring migratory seabirds and the microclimates in and around nesting burrows. Another more recent example is the CitySense project, which is deploying an urban scale sensor network for monitoring weather conditions and air pollutants in the city of Cambridge, Massachusetts. Note that the technologies and protocols developed to deploy these real-world sensor networks for environmental monitoring represent also the basis for sensor networks targeting mission-critical application scenarios, such as surveillance, intruders' reconnaissance and tracking, tracking of goods and vehicles, detection of nuclear, biological and chemical attack, underwater surveillance for harbor control, etc. [Lop07].

6.5.4 Opportunistic Networks

Opportunistic networks constitute a medium-term application of general-purpose MANETs for providing connectivity opportunities to pervasive devices when no direct access to the Internet is available. One of the main limitations of legacy MANETs is the fact that partitioning causes the failure of ongoing communications, and/or nodes that are temporarily disconnected from the network cannot communicate. In opportunistic networks the information delivery is still multihop, but intermediate nodes store the messages when no forwarding opportunity towards the final destination(s) exists and exploit any contact opportunity with other mobile devices to forward information. In other words, this evolution of MANETs opportunistically exploits mobility, which resulted "hostile" for legacy ad hoc networks, and local forwarding in order to take advantage of the temporary wireless links when distributing information. Therefore, this networking paradigm has a huge potential for significantly improving the capability of first responders to reestablish effective communications in a crisis area, as shown in Figure 2 and discussed in [Lil07]. Note that, the opportunistic networking has several application scenarios beyond the PPDR scenarios, especially for pervasive computing and autonomic environments [PPC06]. For instance, the IRTF Delay Tolerant Networking (DTN) Research Group is working to standardize architecture and protocols for enabling Internet services in networks with intermittent connectivity where continuous end-to-end connectivity cannot be assumed. The DTN architecture is suitable to interconnect systems of different scales, ranging from small-size networks formed by single

mobile devices sparsely deployed in the environment, to interplanetary networks bringing together Internet-like network trunks sporadically connected through satellite links. DakNet or Saami network connectivity (SNC) are good examples of the potential applications of opportunistic and delay tolerant networks. DakNet aims at providing low-cost connectivity to rural villages in India, by exploiting mobile relays (i.e. access points mounted on buses, motorcycles, and even bicycles) passing by the village kiosks and exchanging data with them wirelessly. SNC use DTN architecture to provide network connectivity to the nomadic Saami population. The KiosNet Project is another example of opportunistic network application in developing countries to provide a variety of services such as birth, marriage, and death certificates, land records, and consulting on medical and agricultural problems.

6.6 Directions for future research

In the last two decades, the research on MANET technologies has laid the foundations to understand the intrinsic limitations and constraints introduced by multi-hop wireless communications and the absence of an authority managing and controlling the network. As discussed in Section 4, these extensive research activities not only have generated a considerable amount of technical papers, but they have also contributed to the development of several classes of real ad hoc networks, namely mesh networks, VANETs, WSN and delay tolerant networks, which will have a key role in the deployment of disaster-response communications systems. However, the specific requirements of safety applications pose new technical challenges that have not been adequately addressed so far. In the following we elaborate on the research issues that still need to be solved to realize practical and efficient systems.

6.6.1 Autonomic network management

The development of self-organizing capabilities is a fundamental prerequisite of any resilient communications system, because the communications devices should be able to react to the variations in the operating conditions without human intervention. In a sense, wireless multi-hop networks, being infrastructureless peer-to-peer networks, represent an excellent example of self-organized networks, because computing devices must coordinate with each other to per-

form all the networking functions. However, most of the research efforts in the MANET community have been dedicated to the development of routing protocols for mobile multi-hop ad hoc networks, producing an incredible number of algorithms. On the contrary, the self-organization property is a multifaceted concept that incorporates a variety of capabilities. Specifically, self-organization includes self-healing, which refers to the ability of the network to detect, localize and repair failures automatically; self-configuration, which is the capacity of automatically generating the set of appropriate configurations parameters to operate in the current environment; and self-optimization, which is the capability to adapt the network in order to achieve relevant objectives (e.g., desired QoS levels). Consequently, the deployment of a truly self-organized network requires the adoption of a holistic approach that takes into account the interplay between all the various self-capabilities.

The ultimate objective of an autonomic network-management module should be to design an autonomic network management architecture, where the network itself helps to detect, diagnose and repair failures, as well as to adapt its configuration and optimize its performance. However, the management of wireless networks in general is by far more complex than the management of wired networks, because wireless communications are affected by the irregularity and instability of the channel conditions that cause non-uniform and variable radio coverage areas. In addition, radio interference may lead to unpredictable behaviors and dramatically performance degradations. Moreover, in a disaster scenario, additional complexities arise because the parts of the communications network are deployed on demand in an unplanned manner. Thus, nodes may malfunction, be incorrectly configured or isolated. Individual link and node failures can easily cause network partitions. Network monitoring is a key tool to build the knowledge of the current status of the network and to discover the operating environment characteristics. Each device should not only collect local information, but also cooperate with other devices to build a representation of the entire network status. The collected information is the fundamental basis to detect anomalies and to trigger alerts to neighboring nodes or control units. The diagnostic tool responsible for the interpretation of the network state may adopt various policies such as a rule-based (i.e., the normal network state is codified though a set of admissible behaviors) or traffic-based (i.e., a set of normal traffic signatures characterizes the proper behavior of the network)

analysis engine. After an alert, additional diagnostic tests should be executed to verify the root cause of the problem and to automatically trigger the most appropriate countermeasure, such as to isolate trouble links and nodes, to reallocate channels, to find alternative multi-hop paths or to balance network loads.

Since the research on the self-management of ad hoc networks is in a very preliminary phase, a few solutions can be identified, which are usually tailored for mesh networks. One example is the Distributed Ad hoc Monitoring (DAMON) [Ram04] system, which uses agents to monitor network behaviors and send collected measurements to central data repositories. However, the use of centralized analysis does not make this system suitable for challenged environments. A more recent proposal is described in [QBRZ06], which describes a diagnostic system that employs trace-driven simulations to detect faults and perform root cause analysis in mesh networks. While a simulation-based approach may be useful to model the complex interaction between the several factors that affect the network behavior, the time required to simulate a large-scale network impedes the utilization of this solution for real-time network management.

6.6.2 Network interoperability

Ensuring interoperable wireless communications among the devices belonging to first responders is a key requirement to effectively respond to manmade and natural disasters. The harmonization of the various standards employed by public safety agencies, as well as the shift towards open architectures and non-proprietary standards will both be crucial factors in favor of the device interoperability. However, due to the different national and international regulations on spectrum allocation, it is extremely difficult to predict a global harmonization of radio systems in the short/medium term. For instance, US and other developed countries are planning to allocate parts of the frequency bands now used for analog TV for public safety purposes [Peh06], while those bands will continue to be used in many developing countries for broadcasting analog TV signals. A promising technological approach to overcome these constraints is to promote the use of cognitive radios and software-defined radios (i.e. software reconfigurable radios, or SDR) in the devices used by first responders. Specifically, cognitive radios are special SDRs that can adjust their

transmission and reception parameters and algorithms according to multiple factors, such as radio spectrum occupancy or current state of the environment. This radio concept opens the way to more efficient radio resource management, but it also represents a potential solution for frequency coordination issues, limitations of available spectrum and problems of incompatible equipments. For these reasons, the design of cognitive radios for public safety applications is emerging as a very active research area [Ron05, Paw05], and two major research directions can be identified. On the one hand, there are still technological obstacles to build cheap and highly flexible SDR equipments supporting different modulation schemes and operating on large spectrum. On the other hand, the development of efficient spectrum sensing capabilities and the design of conflict resolution algorithms are still open issues where insufficient results have been obtained. For instance, in [Gan05] a cooperative spectrum sensing framework is proposed, where cognitive radios can exchange local sensing results to obtain an accurate estimate of unused frequency bands, and even the locations of the other radios, as well as to reduce detection times. In [NC06], a game theoretic framework is developed to model the efficiency of adaptive and distributed channel allocation for cognitive radios. However, it is not clear the tradeoff between the overheads needed to coordinate the frequency allocations and the network performance improvement.

6.6.3 QoS protection

Until recently, the design of mechanisms and policies to support QoS levels and the design of a resilient communications infrastructure appeared as two separated and uncorrelated research domain areas. However, after the analysis of communication breakdowns during recent disasters it is clearly emerged that the survivability of the communications infrastructure and end-to-end connectivity is not sufficient to guarantee the survivability of the communications services. For instance, in the final report of the 9-11 Commission it was pointed out that, although the cellular telecommunications networks were not destroyed by the terrorist attacks, the first responders where unable to use them because severely congested by the huge number of simultaneous connection attempts. In other words, in crisis response the network workloads can overwhelm the available network capacity such that the minimum application requirements of real-time traffic (e.g., voice communications) cannot be met. On the contrary,

in emergency situations it is fundamental to ensure that critical data are made available to the right set of users, avoiding congestion and data unavailability [BF04]. For these reasons, novel mechanisms are needed to support QoS in ad hoc networks to guarantee different QoS levels, which are appropriate to the information criticality and the network mission. It is evident that a system-wide QoS notion requires that the QoS support be implemented in each MANET protocol. However, it is also true that a QoS-aware routing protocol is the basis of any QoS solution for MANETs, because the ad hoc routing protocol is responsible for finding the relaying nodes that can meet the applications' requirements. For these reasons, especially in the last years, the MANET research focus has shifted from routing protocols maintaining best-effort end-to-end connectivity between mobile devices to the provision of diverse and more complex QoS attributes. These research activities have produced a considerable number of solutions, and the major contributions are outlined in [HIT07]. However, most of these potential solutions have neglected the importance of QoS robustness, namely the capacity of maintaining with high probability the QoS guarantees regardless of network variations such as individual link or node failures. Thus, the design of policies and mechanisms to obtain reliable and adaptive QoS support is still an open issue. An interesting direction for future research in the area of reliable QoS is the use of preemptive strategies. For instance, in [Ayy06] the authors proposed to use preemptive selection of routes according to predictive stability measures. Admission control strategies and segregation of dedicated network resources are also promising areas of investigation. As an example, [BF04] described an architecture composed of geographically distributed ticket servers to identify the priority that should be given to a blow in stressed networks, and to limit resource usage by low priority users.

6.7 Conclusions

In this chapter we have advocated the adoption of ad hoc networking technologies to address the fragility of our communications infrastructure, which has been dramatically exposed in the aftermath of recent natural and man-made disasters. In fact, in the recent years the significant advances in ad hoc networking technologies have led to the development of various types of spe-

cialized networks, such as mesh networks, vehicular networks, sensor networks and opportunistic networks, which are of particular interest and importance for PPDR scenarios. In addition, the ad hoc networking paradigm intrinsically provides flexibility, self-configurability and fully decentralized operations, which are necessary requirements to deploy the future generation of dependable, versatile and secure communications systems for PPDR applications. However, there are several open technical challenges that have to be addressed to realize this vision of a survivable communications system in disaster scenarios. For instance, it is unacceptable to have a communications network that partially stops working correctly during a crisis. Therefore, the focus is on providing continuous communication services, even with degraded performance. In other words, for modern disaster scenarios the focus should move from traditional QoS provision to QoS protection, with a native support of prioritization of emergency network traffic. Second, interoperability between devices, communication paradigms and network architectures is a prerequisite for an effective implementation of PPDR operations. However, the design of very specialized MANET-based networks has largely neglected the interoperability concerns. Finally, in disaster scenarios the human intervention for the bootstrap, configuration, maintenance and adaptation of the communications infrastructures is impossible. Therefore, self-management capabilities should be native functionalities and an integral part of the network design, so that the network itself may help to detect, diagnose and repair failures, as well as to adapt its configuration and optimize its performance.

Chapter 7

Conclusions

In this thesis we have addressed different key problems for wireless mesh networks (WMN). First, we have proposed an analytical predictive tool, developing a queuing network model capable to predict the network capacity and used it in a load aware routing protocol in order to provide, to the end users, a quality of service based on the throughput. We have then extended the queuing network model and introduced a multi-class queuing network model to predict analytically the average end-to-end packet delay of the traffic flows among the mobile end users and the Internet. Second, we have proposed a auto-configuration solution to extend the coverage of a WMN by interconnecting it to a Mobile Ad Hoc Network (MANET) in a transparent way for the infrastructure network (i.e., legacy Internet interconnected to the wireless mesh network). Third, we have implemented two real testbed prototypes of the proposed solutions as a proof-of-concept, both for the load aware routing protocol and the auto-configuration protocol. Finally we have discussed the issues related to the adoption of ad hoc networking technologies to address the fragility of our communications infrastructure and to build the next generation of dependable, secure and rapidly deployable communications infrastructures.

Going in more details, differently from other studies on WMNs, in Chapter 2 we have considered heterogeneous WMNs, which are wireless mesh networks where gateways' backhaul links may have various speeds. A practical example can be a wireless mesh networks evolved into a *converged infrastructure* used to share the Internet connectivity of sparsely deployed fixed lines with *heteroge-*

neous capacity, ranging from ISP-owned broadband links to subscriber-owned low-speed connections. Focusing on this scenario, we have developed a queuing network model to analyze the network capacity as a function of several system parameters, including locations of gateways, traffic patterns, link bandwidths and packet loss rates. By exploiting this predictive tool, we have designed LARS, a load-aware route and gateway selection algorithm that improves the network capacity by ensuring a more balanced utilization of the network and gateways' resources. Using simulations and a prototype implementation in a realistic small-scale mesh network, we have shown that our scheme significantly outperforms the shortest path routing using a contention-aware routing metric, providing up to 240% throughput improvement in relevant network scenarios. An important outcome of our analysis is that some mesh nodes may obtain substantially lower throughput, or on the contrary can get more overloaded, than others depending on several factors, including locations of gateways, traffic patterns and link capacities. Hence, exploiting the analytical prediction of these events, the overall network performance is significantly improved by LARS by taking into account the residual capacity of network paths and gateways' connection to the Internet in the route and gateway selection processes, while traditional routing protocols for ad hoc networks, both proactive (e.g. OLSR or TBRPF) and reactive (e.g. DSR), are not able to exploit these information. In Chapter 4 we have validated our solution and shown its feasibility in a real small scale testbed.

In Chapter 3 we have shown that extending the model to a multi-class queuing network model it can be effectively used not only to characterize the maximum achievable throughput, including both the download and upload traffic among the mobile end users and the Internet, but it can also be an analytical tool to predict the average end-to-end delay. The prediction is based on the analytical estimation of the queuing model statistical moments and using them to predict the average delay introduced by each node queuing system in the network. The validation of the model against a simulation shows a good prediction in relevant scenarios.

In Chapter 5 we described AH-DHCP, an address autoconfiguration protocol for multi-hop WLAN, such as WMNs. The main goal of our work was to prove the applicability of DHCP, originally designed to provide configuration parameters to hosts in a fixed network, also when traditional WLANs

integrate ad hoc networking technologies to discover and maintain multi-hop wireless path within the network. The basic idea was to take advantage of DHCP relay capabilities available in already configured nodes. To this end, we proposed extensions to DHCP to enable a new node to dynamically choose a reachable relay agent as the unique initiator of the configuration procedure. Then, this relay transparently passes all the client-originated messages to the DHCP servers located in the wired part of the network. Our proposed solution can tolerate messages losses and node mobility because it implements appropriate mechanisms to promptly react to persistent communication problems and topology changes. Experiments conducted with a prototype implementation of AH-DHCP have shown that our solution ensures short address configuration delays and low protocol overheads, even when node mobility or background traffic interferes with the operations of the autoconfiguration protocol.

In Chapter 6 we have advocated the adoption of ad hoc networking technologies to address the fragility of our communications infrastructure, which has been dramatically exposed in the aftermath of recent natural and man-made disasters. In fact, in the recent years the significant advances in ad hoc networking technologies have led to the development of various types of specialized networks, such as mesh networks, vehicular networks, sensor networks and opportunistic networks, which are of particular interest and importance for PPDR scenarios. In addition, the ad hoc networking paradigm intrinsically provides flexibility, self-configurability and fully decentralized operations, which are necessary requirements to deploy the future generation of dependable, versatile and secure communications systems for PPDR applications. However, there are several open technical challenges that have to be addressed to realize this vision of a survivable communications system in disaster scenarios. For instance, it is unacceptable to have a communications network that partially stops working correctly during a crisis. Therefore, the focus is on providing continuous communication services, even with degraded performance. In other words, for modern disaster scenarios the focus should move from traditional QoS provision to QoS protection, with a native support of prioritization of emergency network traffic. Second, interoperability between devices, communication paradigms and network architectures is a prerequisite for an effective implementation of PPDR operations. However, the design of very specialized MANET-based networks has largely neglected the interoperability concerns.

Finally, in disaster scenarios the human intervention for the bootstrap, configuration, maintenance and adaptation of the communications infrastructures is impossible. Therefore, self-management capabilities should be native functionalities and an integral part of the network design, so that the network itself may help to detect, diagnose and repair failures, as well as to adapt its configuration and optimize its performance.

We believe there are several related aspects that are worth being further investigated in future works. Although our queuing network model based analysis considers packet losses due to channel errors, we have used an idealized CSMA-based MAC protocol, which primarily captures location-dependent contention issues due to differences in the number of contending nodes at both endpoints of each communication link. Even if this basic CSMA model can provide accurate expressions, the extension of our analysis to a real MAC protocol implementing practical collision avoidance mechanisms is a challenge that needs to be addressed.

On hybrid WMN auto-configuration, we intend to investigate mechanisms to reduce the impact of multi-hop forwarding on address assignment delays in large-scale multi-hop WLANs, e.g., by introducing a hierarchy of DHCP relay agents. Another possible research direction is the extension of our solution to IPv6.

Other directions for future works include the delay analysis and characterization of the maximum achievable throughput for network using other MAC protocols, such as TDMA-based access schemes. Furthermore, various strategies can be devised to select the initial subset of optimal routes between the mesh nodes and the available gateways. For instance, delay may be a more significant metric to use for real-time traffic. However, jointly considering capacity and end-to-end delay constraints in the routing process is still an open research area. Finally, the extension of our routing method to deal with intra-mesh traffic in addition to Internet traffic is an ongoing activity.

Bibliography

[ABC+07] E. Ancillotti, R. Bruno, M. Conti, E. Gregori, and A. Pinizzotto. A Layer-2 Framework for Interconnecting Ad Hoc Networks to Fixed Internet: Test-bed Implementation and Experimental Evaluation. *The Computer Journal*, 50(4):478–499, May 2007.

[ABC09] E. Ancillotti, R. Bruno, and M. Conti. Design and Performance Evaluation of Throughput-Aware Rate Adaptation Protocols for IEEE 802.11 Wireless Networks. *Performance Evaluation*, 66(12):811–825, 2009.

[ABCP09] E. Ancillotti, R. Bruno, M. Conti, and A. Pinizzotto. Dynamic address autoconfiguration in hybrid ad hoc networks. *Pervasive and Mobile Computing*, 5(4):300–317, August 2009.

[ABCP11] E. Ancillotti, R. Bruno, M. Conti, and A. Pinizzotto. Load-Aware Routing in Mesh Networks: Models, Algorithms and Experimentation. *Computer Communications*, 34(8):948–961, June 1 2011. DOI:10.1016/j.comcom.2010.03.004, in press.

[Ahm07] Ahmed, N. and Jamshaid, K. and Khan, O. Z. SAFIRE: A Self-Organizing Architecture for Information Exchange between First Responders. In *Proceedings of IEEE Workshop on Networking Technologies for SDR Networks*, San Diego, CA, USA, June 8 2007.

[ALCR08] H. Aiache, L. Lebrun, V. Conan, and S. Rousseau. A load dependent metric for balancing Internet traffic in Wireless Mesh Networks. In *Proc. IEEE MeshTech'08*, pages 629–634, Atlanta, GA, USA, September 29, 2008.

[Anc07] Ancillotti, E. and Bruno, R. and Conti, M. and Gregori, E. and Pinizzotto, A. Implementation and Experimentation of a layer-2 Architecture for Interconnecting Heterogeneous Ad Hoc Networks to the Internet. In M. Conti, J. Crowcroft, and A. Passarella, editors, *Mobile Ad Hoc Networks: from Theory to Reality*. Nova Science Publisher, 2007.

[ASS03] F. Alizadeh-Shabdiz and S. Subramaniam. A Finite Load Analytical Model for IEEE 802.11 Distributed Coordination Function MAC. In *Proc. ACM WiOpt'03*, volume 3, pages 3–5, Sophia-Antipolis, France, March 3–5 2003.

[AWW05] I. Akyildiz, X. Wang, and W. Wang. Wireless mesh networks: a survey. *Computer Networks*, 47(4):445–487, March 2005.

[Ayy06] Ayyash, M. and Alzoubi, K. and Alsbou, Y. Preemptive quality of service infrastructure for wireless mobile ad hoc networks. In *Proceeding of IWCMC'06*, pages 707–712, Vancouver, British Columbia, Canada, July 3–6 2006.

[BA09] N. Bisnik and A. Abouzeid. Queuing network models for delay analysis of multihop wireless ad hoc networks. *Ad Hoc Networks*, 7(1):79–97, January 2009.

[BABM05] J. Bicket, D. Aguayo, S. Biswas, and R. Morris. Architecture and Evaluation of an Unplanned 802.11b Mesh Network. In *Proc. of ACM MobiCom*, pages 31–42, Cologne, Germany, Aug. 28–Sept. 2 2005.

[BCG05] R. Bruno, M. Conti, and E. Gregori. Mesh Networks: Commodity Multihop Ad Hoc Networks. *IEEE Commun. Mag.*, 43(3):123–131, March 2005.

[BCP08] R. Bruno, M. Conti, and A. Pinizzotto. Enhancing DHCP for Address Autoconfiguration in Multi-hop WLANs. In *ICDCN 2008*, volume 4904 of *Lecture Notes in Computer Science*, pages 528–539, Kolkata, India, January 5–8 2008. Springer.

[BCP09a] R. Bruno, M. Conti, and A. Pinizzotto. A Queuing Modeling Approach for Load-Aware Route Selection in Heterogenous Mesh

Networks. In *Proc. of IEEE WoWMoM'09*, Kos , Greece, June 15–19 2009.

[BCP09b] R. Bruno, M. Conti, and A. Pinizzotto. Capacity-Aware Routing in Heterogeneous Mesh Networks: An Analytical Approach. In *Proc. of IEEE MsWiM'09*, Tenerife , Canary Islands, Spain, October 26–30 2009.

[BCP11] R. Bruno, M. Conti, and A. Pinizzotto. Routing Internet Traffic in Heterogeneous Mesh Networks: Analysis and Algorithms. *Performance Evaluation*, 2011. DOI:10.1016/j.peva.2011.01.006, in press.

[BF04] C. Beard and V. Frost. Prioritization of Emergency Network Traffic using Ticket Servers: A Performance Analysis. *Simulation: Trans-actions of the Society for Modeling and Simulation*, 80(6):289–299, June 2004.

[BGdMT06] G. Bolch, S. Greiner, H. de Meer, and K. S. Trivedi. *Queueing Networks and Markov Chains: Modeling and Performance Evaluation with Computer Science Applications*. Wiley-Blackwell, May 2006.

[BHK07] Y. Bejerano, S. Han, and A. Kumar. Efficient load-balancing routing for wireless mesh networks. *Computer Networks*, 51(10):2450–2466, 2007.

[Bic05] Bicket, J. and Biswas, S. and Aguayo, D. and Morris, R. Architecture and evaluation of an unplanned 802.11b mesh network. In *Proceedings of ACM MobiCom 2005*, pages 31–42, Cologne, Germany, August 28–September 2 2005.

[CAG05] S. Cheshire, B. Aboba, and E. Guttman. Dynamic Configuration of IPv4 Link-Local Addresses. RFC 3927, May 2005.

[CCL03] I. Chlamtac, M. Conti, and J. Liu. Mobile Ad Hoc Networking: Imperatives and Challenges. *Elsevier Ad Hoc Networks Journal*, 1(1):13–64, July 2003.

[CG07a] M. Conti and S. Giordano. Multihop Ad Hoc Networking: the Reality. *IEEE Commun. Mag.*, 45(4):88–95, April 2007.

[CG07b] M. Conti and S. Giordano. Multihop Ad Hoc Networking: The Theory. *IEEE Commun. Mag.*, 45(4):78–86, April 2007.

[CJ03] T. Clausen and P. Jaquet. Optimized Link State Routing Protocol (OLSR). RFC 3626, October 2003.

[CMS09] P. Calhoun, M. Montemurro, and D. Stanley. Control And Provisioning of Wireless Access Points (CAPWAP) Protocol Specification. RFC 5415, March 2009.

[Dai08] Dai, L. and Xue, Y. and Cao, Y. and Cui, Y. Integrating Traffic Estimation and Routing Optimization for Multi-Radio Multi-Channel Wireless Mesh Networks. In *Proc. IEEE INFOCOM'08*, pages 71–75, Phoenix, AZ, April 13–18 2008.

[DBV+03] R. Droms, J. Bound, B. Volz, T. Lemon, C. Perkins, and M. Carney. Dynamic Host Configuration Protocol for IPv6 (DHCPv6). RFC 3315, July 2003.

[DCABM03] D. De Couto, D. Aguayo, J. Bicket, and R. Morris. A High-Throughput Path Metric for Multi-Hop Wireless Routing. In *Proc. of ACM MobiCom*, pages 134–146, San Diego, CA, USA, September, 14–19 2003.

[Dil07] Dilmaghani, R.B. and Rao, R.R. Future Wireless Communication Infrastructure with Application to Emergency Scenarios. In *Proceedings of IEEE WoWMoM 2007*, Helsinki, Finland, June 18–21 2007.

[DPZ04] R. Draves, J. Padhye, and B. Zill. Routing in Multi-Radio, Multi-Hop Wireless Mesh Networks. In *Proc. of ACM MobiCom'04*, pages 114–128, Sept. 26–Oct. 1 2004.

[Dro97] R. Droms. Dynamic Host Configuration Protocol. RFC 2131, March 1997.

[EE04] P. Engelstad and G. Egeland. NAT-based Internet Connectivity for On Demand MANETs. In *Proc. of WONS 2004*, pages 4050–4056, Madonna di Campiglio, Italy, January, 18–23 2004.

[EMSW02] C. Eklund, R. B. Marks, K. L. Stanwood, and S. Wang. IEEE standard 802.16: A technical overview of the WirelessMAN air interface for broadband wireless access. *IEEE Commun. Mag.*, pages 98–107, June 2002.

[ESA06] ESARB. Meeting the Challenge: the European Security Research Agenda - A report from the European Security Research Advisory Board, September 2006.

[ETHE04] P. Engelstad, A. Tønnesen, A. Hafslund, and G. Egeland. Internet Connectivity for Multi-Homed Proactive Ad Hoc Networks. In *Proc. of IEEE ICC'2004*, volume 7, pages 4050–4056, Paris, France, June, 20–24 2004.

[Eur06] European Commission. FP7 Cooperation Work Programme - Theme 10: Security (Call 1), December 2006.

[Fan03] Z. Fan. IPv6 stateless address autoconfiguration in ad hoc networks. In *Proc. of PWC'03*, pages 665–678, Venice, Italy, September 23-25 2003.

[Fon09] T. L. Fondation. Netem Tools, October 2009.

[Gan05] Ganesan, G. and Li, Y. Cooperative Spectrum Sensing in Cognitive Radio Networks. In *Proceeding. of DySPAN 2005*, pages 137–143, Baltimora, MR, USA, November 8–11 2005.

[GCL06] Y. Gao, D.-M. Chiu, and J. C. Lui. Determining the end-to-end throughput capacity in multi-hop networks: methodology and applications. *SIGMETRICS Perform. Eval. Rev.*, 34(1):39–50, 2006.

[GS08] M. Genetzakis and V. Siris. A Contention-Aware Routing Metric for Multi-Rate Multi-Radio Mesh Networks. In *Proc. IEEE SECON'08*, pages 242–250, San Francisco, CA, USA, June 16–20, 2008.

[Hat05] Hatfield, D. and Weiser, P. Toward a A Next Generation Strategy – Learning from Katrina and Taking Advantage of New Technologies, 2005.

[HCC05] J. He, J. Chen, and S.-H. Chan. Extending WLAN coverage using infrastructureless access points. In *Proc. of IEEE HPSR 2005*, pages 162–166, May 12–14 2005.

[HIT07] L. Hanzo II and R. Tafazolli. A Survey of QoS Routing Solutions for Mobile Ad Hoc Networks. *IEEE Communications Surveys and Tutorials*, 9(2):50–70, 2nd Quarter 2007.

[IEE06] IEEE TGs. Joint SEE-Mesh/Wi-Mesh Proposal to 802.11 TGs, March 2006. http://grouper.ieee.org/groups/802/11/.

[IET07] IETF. Ad-Hoc Network Autoconfiguration (autoconf) WG, December 2007.

[Int06] Internet Systems Consortium. ISC DHCP Version 3.0.5, November 5 2006.

[JNF04] C. Jelger, T. Noel, and A. Frey. Gateway and Address Autoconfiguration for IPv6 Ad Hoc Networks. Internet Draft, April 2004.

[KAMG07] A. Kumar, E. Altman, D. Miorandi, and M. Goyal. New insights from a fixed-point analysis of single cell IEEE 802.11 WLANs. *IEEE/ACM Trans. Networking*, 15(3):588–601, January 2007.

[KGDB07] A. Kashyap, S. Ganguly, A. Das, and S. Banerjee. VoIP on Wireless Meshes: Models, Algorithms and Evaluation. In *Proc. IEEE INFOCOM'07*, pages 2036–2044, Anchorage, USA, May 6–12 2007.

[KSK04] R. Karrer, A. Sabharwal, and E. Knightly. Enabling Large-Scale Wireless Broadband: The Case for TAPs. *ACM SIGMOBILE Comp. Comm. Review*, 34(1):27–34, 2004.

[LBB04] S. Lee, S. Banerjee, and B. Bhattacharjee. The Case for a Multi-hop Wireless Local Area Network. In *Proc. of IEEE INFOCOM*, volume 2, pages 894–905, Hong Kong, China, March, 7–11 2004.

[LH00] Y.-D. Lin and Y.-C. Hsu. Multihop cellular: a new architecture for wireless communications. In *Proc. of IEEE INFOCOM 2000*, volume 3, pages 1273–1282, Tel Aviv, Israel, March 26–30 2000.

[LH07] S. Liang and D. Hatzinakos. A Cross-Layer Architecture of Wireless Sensor Networks for Target Tracking. *IEEE/ACM Transactions on Networking*, 15(1):145–158, 2007 February 2007.

[Lil07] Lilien, L. and Gupta, A. and Yang, Z. Opportunistic Networks for Emergency Applications and Their Standard Implementation Framework. In *Proceedings of IEEE IPCCC 2007*, pages 588–593, New Orleans, LA, USA, April 11–13 2007.

[LLT03] B. Liu, Z. Liu, and D. Towsley. On the Capacity of Hybrid Wireless Networks. In *Proc. of IEEE INFOCOM'03*, volume 2, pages 1543–1552, Mar. 30–Apr. 3 2003.

[Lon06] London Regional Resilience Forum. Looking Back, Moving Forward – The Multi-Agency Debrief, September 2006.

[Lop07] Lopez-Ramos, M. and Leguay, J. and Conan, V. Designing a novel SOA architecture for security and surveillance WSNs with COTS. In *Proceedings of IEEE MASS-GHS'07*, Pisa, Italy, October 8 2007.

[MBLD07] V. Mhatre, F. Baccelli, H. Lundgren, and C. Diot. Joint MAC-aware routing and load balancing in mesh networks. In *Proc. ACM CoNEXT'07*, pages 1–12, New York, USA, December 10–13 2007.

[MD07] L. Ma and M. Denko. A Routing Metric for Load-Balancing in Wireless Mesh Networks. In *Proc. of IEEE AINAW '07*, volume 2, pages 21–23, May 21–23 2007.

[Mer] Meraki Networks Inc. Residential & MDU Case Studies. http://meraki.com/.

[MHB07] B. Manoj and A. Hubenko-Baker. Communication challenges in emergency response. *Communications of the ACM*, 50(3):51–53, 2007.

[NC06] N. Nie and C. Comaniciu. Adaptive Channel Allocation Spectrum etiquette for Cognitive Radio Networks. *Mobile Networks and Applications*, 11(6):779–797, December 2006.

[NLA05] NLANR/DAST. Iperf Version 2.0.2, May 3 2005.

[NLP05] S. Narayanan, P. Liu, and S. Panwar. On the Advantages of Multi-hop Extensions to the IEEE 802.11 Infrastructure Mode. In *Proc. of IEEE WCNC 2005*, volume 1, pages 132–138, New Orleans, LA, USA, March 13–17 2005.

[NP02] S. Nesargi and R. Prakash. MANETconf: Configuration of Hosts in a Mobile Ad Hoc Network. In *Proc. of IEEE INFOCOM 2002*, volume 2, pages 1059–1068, New York, NY, USA, June, 23–27 2002.

[NTCS99] S. Ni, Y. Tseng, Y. Chen, and J. Sheu. The broadcast storm problem in a mobile ad hoc network. In *Proc. of ACM MobiCom'99*, pages 151–162, Seattle, WA, USA, August 15–20 1999.

[OM04] M. Özdemir and A. McDonald. An M/MMGI/1/K queuing model for IEEE 802.11 ad hoc networks. In *Proc. IEEE PE-WASUN'04*, pages 107–111, Venice, Italy, 2004.

[Ozo] Ozone. French Wireless ISP. http://www.ozone.net/.

[Paw05] Pawelczak, P. and Prasad, R.V. and Liang, X. and Niemegeers, I.G. Cognitive radio emergency networks - requirements and design. In *Proceeding. of DySPAN 2005*, pages 601–606, Baltimora, MR, USA, November 8–11 2005.

[Peh06] J. Peha. The digital TV transition: A chance to enhance public safety and improve spectrum auctions. *IEEE Commun. Mag.*, 44(6):22–23, June 2006.

[PKLK04] A. Park, P. Kim, M. Lee, and Y. Kim. Fast Address Configuration for WLAN. In *Parallel and Distributed Computing: Applications and Technologies*, Lecture Notes in Computer Science, pages 396–400. Springer, December 2004.

[PMW+02] C. Perkins, J. Malinen, R. Wakikawa, E. Belding-Royer, and Y. Sun. IP Address Autoconfiguration for Ad Hoc Networks. Internet Draft, July 2002.

[PPC06] L. Pelusi, A. Passarella, and M. Conti. Opportunistic Networking: Data Forwarding in Disconnected Mobile Ad Hoc Networks. *IEEE Commun. Mag.*, 44(11):134–141, November 2006.

[QBRZ06] L. Qiu, P. Bahl, A. Rao, and L. Zhou. Troubleshooting Wireless Mesh Networks. *Computer Communications Review*, 36(5):19–28, October 2006.

[Ram04] Ramachandran, K.N. and Belding-Royer, E.M. and Aimeroth, K.C. DAMON: a Distributed Architecture for Monitoring Multihop Mobile Networks. In *Proceeding of IEEE SECON 2004*, pages 601–609, Santa Clara, CA, USA, October 4–7 2004.

[Roc07] Roccetti, M. and Gerla, M. and Palazzi, C.E. and Ferretti, S. and Pau, G. First Responders' Crystal Ball: How to Scry the Emergency from a Remote Vehicle. In *Proceedings of IEEE IPCCC 2007*, pages 556–561, New Orleans, LA, USA, April 11–13 2007.

[Ron05] Rondeau, T. W. and Bostian, C. W. and Maldonado, D. and Ferguson, A. and Ball, S. and Midkiff, S. F. and Le, B. Cognitive Radios in Public Safety and Spectrum Management. In *Proceeding of 33rd Research Conference on Communication, Information and Internet Policy*, Arlington, VA, USA, September 23–25 2005.

[RSBA07] K. Ramachandran, I. Sheriff, E. Belding, and K. Almeroth. Routing Stability in Static Wireless Mesh Networks. In *Proc. PAM'07*, pages 73–82, Louvain-la-neuve, Belgium, April 5–6 2007.

[SACB08] V. Siris, I. Askoxylakis, M. Conti, and R. Bruno. Enhanced, Ubiquitous and Dependable Broadband Access using MESH Networks. *ERCIM News*, 73:50–51, April 2008.

[The99] The Institute of Electrical and Electronics Engineer, Piscataway, NJ. *Local and Metropolitan Area Network – Specific Requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, August 1999.

[TN98] S. Thomson and T. Narten. IPv6 stateless address autoconfiguration. RFC 2462, December 1998.

[Tor05] Torrent-Moreno, M. and Killat, M. and Hartenstein, H. The challenges of robust inter-vehicle communications. In *Proceedings of IEEE VTC-Fall 2005*, pages 319–323, Dallas, TX, USA, September 28–25 2005.

[TS08] O. Tickoo and B. Sikdar. Modeling Queueing and Channel Access Delay in Unsaturated IEEE 802.11 Random Access MAC Based Wireless Networks. *IEEE/ACM Trans. Networking*, 16(4):878–891, August 2008.

[TSHN09] H. Tokito, M. Sasabe, G. Hasegawa, and H. Nakano. Routing Method for Gateway Load Balancing in Wireless Mesh Networks. In *Proc. IEEE ICN '09*, pages 127–132, Dresden, Germany, June 14–18 2009.

[UK 06] UK Government (J. Reid, T. Jowell). Addressing Lessons from the Emergency Response to the 7 July 2005 London Bombings, September 2006.

[US 05] US National Task Force on Interoperability. Why Can't We Talk?, February 2005.

[US 06] US Homeland Security. Hurricane Katrina: A Nation Still Unprepared, May 2006.

[Vai02] N. Vaidya. Weak Duplicate Address Detection in Mobile Ad Hoc Networks. In *Proc. of ACM MobiHoc 2002*, pages 206–216, Lausanne, Switzerland, June, 9–11 2002.

[WB06] S. Waharte and R. Boutaba. Totally Disjoint Multipath Routing in Multihop Wireless Networks. In *Proc. IEEE ICC'06*, volume 12, pages 5576–5581, Istanbul, Turkey, June 2006. IEEE.

[Wen05] K. Weniger. PACMAN: Passive Autoconfiguration for Mobile Ad Hoc Networks. *IEEE J. Select. Areas Commun.*, 23(3):507–519, March 2005.

[WMP+06] R. Wakikawa, J. Malinen, C. Perkins, A. Nilsson, and A. Tuominen. Global Connectvity for IPv6 Mobile Ad Hoc Networks. Internet Draft, March 2006.

[WZ04] K. Weniger and M. Zitterbart. Address Autoconfiguration on Mobile Ad Hoc Networks: Current Approaches and Future Directions. *IEEE Network*, 18(4):6–11, July/August 2004.

[You06] Yousefi, S. and Mousavi, M.S. and Fathy, M. Vehicular Ad Hoc Networks (VANETs): Challenges and Perspectives. In *Proceedings of 6th International Conference on ITS Telecommunications*, pages 761–766, Chengdu, China, June 21–23 2006.

[YWK05] Y. Yang, J. Wang, and R. Kravets. Designing Routing Metrics for Mesh Networks. In *Proc. of IEEE WiMesh, 2005*, Santa Clara, CA, USA, September, 26 2005.

[YWK06] Y. Yang, J. Wang, and R. Kravets. Load-balanced routing for mesh networks. *ACM SIGMOBILE Mobile Computing and Communications Review*, 10(4):3–5, 2006.

[ZNM03] H. Zhou, L. Ni, and M. Mutka. Prophet Address Allocation for Large Scale MANETs. *Ad Hoc Networks Journal*, 1(4):423–434, November 2003.

[ZWR08] P. Zou, X. Wang, and R. Rao. Asymptotic Capacity of Infrastructure Wireless Mesh Networks. *IEEE Trans. Mob. Comp.*, 7(8):1011–1024, August 2008.