Cooperation Strategies for
Enhanced Connectivity at Home

POLITECNICO DI TORINO

SCUOLA DI DOTTORATO
Dottorato in Ingegneria Elettronica e delle Comunicazioni – XXIV ciclo
Settore disciplinare: ING-INF/03

Tesi di Dottorato

Cooperation Strategies for Enhanced Connectivity at Home

Claudio ROSSI
Mtr. 178823

Tutori
prof. Claudio CASETTI

Coordinatore del corso di dottorato
Prof. Ivo Montrosset

5 Dicembre 2013

Summary

WHILE AT HOME, USERS MAY EXPERIENCE A POOR INTERNET SERVICE while being connected to their 802.11 Access Points (APs). The AP is just one component of the Internet Gateway (GW) that generally includes a backhaul connection (ADSL, fiber, etc..) and a router providing a LAN. The root cause of performance degradation may be poor/congested wireless channel between the user and the GW or congested/bandwidth limited backhaul connection. The latter is a serious issue for DSL users that are located far from the central office because the greater the distance the lesser the achievable physical datarate. Furthermore, the GW is one of the few devices in the home that is left always on, resulting in energy waste and electromagnetic pollution increase. This thesis proposes two strategies to enhance Internet connectivity at home by (i) creating a wireless resource sharing scheme through the federation and the coordination of neighboring GWs in order to achieve energy efficiency while avoiding congestion, (ii) exploiting different king of connectivities, i.e., the wired plus the cellular (3G/4G) connections, through the aggregation of the available bandwidth across multiple access technologies.

In order to achieve the aforementioned strategies we study and develop:

- A viable interference estimation technique for 802.11 BSSes that can be implemented on commodity hardware at the MAC layer, without requiring active measurements, changes in the 802.11 standard, cooperation from the wireless stations (WSs). We extend previous theoretical results on the saturation throughput in order to quantify the impact in term of throughput loss of any kind of interferer. We implement and extensively evaluate our estimation technique with a real testbed and with different kind of interferer, achieving always good accuracy.

- Two available bandwidth estimation algorithms for 802.11 BSSes that rely only on passive measurements and that account for different kind of interferers on the ISM band. This algorithms can be implemented on commodity hardware, as they require only software modifications. The first algorithm applies to intra-GW while the second one applies to inter-GW available bandwidth estimation. Indeed, we use the first algorithm to compute the metric for assessing the Wi-Fi load of a GW and the second one to compute the metric to decide whether accept incoming WSs from neighboring GWs or not. Note that in the latter case it is assumed that one or more

WSs with known traffic profile are requested to relocate from one GW to another one. We evaluate both algorithms with simulation as well as with a real test-bed for different traffic patterns, achieving high precision.

- A fully distributed and decentralized inter-access point protocol for federated GWs that allows to dynamically manage the associations of the wireless stations (WSs) in the federated network in order to achieve energy efficiency and offloading congested GWs, i.e, we keep a minimum number of GWs ON while avoiding to create congestion and real-time throughput loss. We evaluate this protocol in a federated scenario, using both simulation and a real test-bed, achieving up to 65% of energy saving in the simulated setting. We compare the energy saving achieved by our protocol against a centralized optimal scheme, obtaining close to optimal results.

- An application level solution that accelerates slow ADSL connections with the parallel use of cellular (3G/4G) connections. We study the feasibility and the potential performance of this scheme at scale using both extensive throughput measurement of the cellular network and trace driven analysis. We validate our solution by implementing a real test bed and evaluating it "in the wild, at several residential locations of a major European city. We test two applications: Video-on-Demand (VoD) and picture upload, obtaining remarkable throughput increase for both applications at all locations. Our implementation features a multipath scheduler which we compare to other scheduling policies as well as to transport level solution like MTCP, obtaining always better results.

Acknowledgements

DURING MY PH.D. I HAVE RECEIVED support and encouragement from many people, which I would like to thank. Thus I sincerely thank:

- my advisor Claudio Casetti for his great guidance and support.
- Prof. Carla-Fabiana Chiasserini for the precious contribution she has given to my publications.
- all the PhD students I had the pleasure to meet during these years, especially Carlo Borgiattino for his kindness and his support in the wireless lab.
- all the telecommunication research group with its staff, that have provided always a good support for both technical and organizational issues.
- the Telefonica Research team for the great professional experience I had during my stay at Barcelona.

I am grateful to all my family, that has always supported me in difficult moments.

I dedicate this thesis to my father, for his solid moral guide and for teaching me the real values of life, including the importance and the pleasure of knowledge.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Enhancing Internet connectivity in residential environments poses several challenges. We propose two novel strategies to improve the energy efficiency of the home-access network and to overcome performance bottlenecks of the wired backhaul.

The first strategy foresee the creation of a federation among Internet Gateways (GWs), implementing the concept of *federated homes*. The latter are neighborhoods where network resources are shared and networked devices belonging to different users cooperate. Federated homes have the potential to optimize resources by incorporating APs in smart *Gateways* that handle all inward and outward network traffic. GWs are advanced home devices capable of providing wireless Internet access and of offering storage and multimedia services, including audio and video real-time streaming. Additionally, they can control a IEEE 802.11 Basic Service Set (BSS), providing wireless Internet access. The federated houses model provides an appealing backdrop to implement cooperation among GWs. We outline a framework that allows such cooperation through the monitoring of local wireless resources and wireless station offloading toward other federated GWs. Thus, we propose wireless resource sharing among federated GWs in order to implement enhanced connectivity and energy saving by (i) allowing under-used devices to hand over their wireless stations to nearby GWs and temporarily switch off, (ii) relocating one or more WSs in case of an overloaded GW, thus offloading congestion. Our simulation results show that, in realistic residential settings, the proposed framework yields an energy saving between 45% and 65%, while providing congestion avoidance and meeting the user expectations in terms of real-time throughput. Thus, with this scheme we enhance the connectivity at home by providing an energy efficient wireless access to the Internet.

The second scheme exploits the availability of different access technologies within the home, i.e., the co-existence in a household of a wired connection (ADLS, cable, fiber) and one or multiple cellular connections (3G/4G). Often, Internet experience in a household can be seriously limited by the wired backhaul. For example, technologies based on DSL features a capacity that is inversely proportional to the distance between the user (home) and the nearest central office. As a consequence, many users experience low throughput

throughout the entire day, being unable to fully enjoy bandwidth-hungry applications such as high definition video streaming. In order to overcome the aforementioned limits, we propose to aggregate the available capacity of the cellular network with the capacity of the wired network, by implementing a scheme that selectively Onloads traffic from the wired onto the cellular network. We consider the technical challenges pertaining to this concept and we propose a network-integrated solution that can boost applications within the home without harming the operations on the cellular network. We name such technology 3G Onloading (3GOL) and we evaluate its performance with both extensive trace analysis, and a real test-bed. We implement a 3GOL prototype that we evaluate at residential locations of a European metropolitan city obtaining up to 4x speedups on the downlink (Video-On-Demand), and up 6x speedups on the uplink (picture upload).

1.1 Enhancing connectivity in a Federated Network

The growing popularity of appliances and consumer devices embedding a WiFi interface has led to the proliferation of Access Points (APs) in public areas and private homes alike. In the latter case, however, the deployment occurs in an uncoordinated fashion, leading to overlapping coverage and spectrum conflicts. Also, APs in private homes are usually underloaded and are left on around the clock, both an energy waste and an unnecessary increase in electromagnetic pollution. This situation can be addressed by a new paradigm for home networking which is garnering widespread attention, based on the concept of *federated homes*. This concepts envisions that the Internet is moving from a core/technology centric architecture to an edge/user centric network where content will be created and delivered to/from hundreds of millions households connected at the edge of the Internet. To address the aforementioned challenges, we consider an evolvable Future Internet architecture based on a GAteway-oriented federation of Residential netwOrks (FIGARO [1]).

Gateway-centric Networking: the residential gateway has a central role in our vision of the Future Internet. It interconnects the residential network with the Internet and it is responsible for aggregating a multitude of devices and services within the residential network. The gateway is the only invariant and indispensable element of the residential network and a natural control point where Internet-based services pass through. Most Internet-enabled end-user devices are connected to it while at home, and will in the future also access the gateway remotely over the Internet. Furthermore, a residential gateway today is nearly as powerful as a PC, and is capable of supporting the increasing requirements originating from the Future Internet challenges. With the convergence of networking technologies, the home gateway has become a critical infrastructure at home and is one of the few always on devices. A gateway-centric approach enables efficient network management in terms of monitoring of network, application, and services, as

well as automatic troubleshooting and network optimizations. Furthermore, it also enables efficient management of digital content. In FIGARO, we have therefore chosen the residential gateway as the primary platform to implement and demonstrate the strength of our proposed Future Internet architecture.

Federation of Residential Networks: FIGARO proposes an evolvable Future Internet architecture based on gateway-based federation of residential networks. We define federation of networks as follows: two or more independent networks that are interconnected and operate, at least partly, in a coordinated fashion compose a federation of networks. The networks in a federation can be independent and heterogeneous in terms of their ownership, characteristics, and technologies, as well as in their targeted services and applications. However, these networks ideally share a set of common objectives, such as, providing a certain set of services to one or more users, exchange of information, sharing and optimization of network resources. Hence, in a federation of networks the sum is greater than its parts since the federation can offer functionality and services beyond what an individual network can provide. In FIGARO, we extend the above concept and define federation of residential networks as follows. We consider the residential gateways to undertake the role of federators. However, residential network federation also presents some challenges that must be addressed. Network federations require the definition of key system requirements for successful operation. For example, common interfaces for querying data, content, services and resources must exist. The federation should also have an agreement regarding common representation of information, such as network configuration and network monitoring data.

Overall FIGARO System Architecture: in FIGARO, residential gateways undertake the federator role. Figure 1.1 shows residential networks connected at the edge of the Internet and illustrates a simplified view including two types of residential network federations. The upper part illustrates external federation interconnecting multiple gateways to form a cooperative overlay across residential networks. This federation enables further collaboration to offer added value in terms of, for example, access and sharing of content, storage and network capacity. The right-most residential network illustrates an example of an internal network federation consisting of the regular IP-based network (e.g., quadruple play) on one hand, and other types of networks, possibly non-IP, specific to services in other sectors (e.g., home automation and e-health) on the other hand. The federation enables features such as communication, resource, and content sharing among the involved networks as well as a common interface to these networks through the gateway.

In Chapter 2 we describe the basic tools needed in order to implement a wireless resource sharing protocol among federated GWs. In Section 2.1 we propose an interference estimation technique that we evaluate in a real testbed with different kind of interferers, while in Section 2.2 we discuss our available bandwidth monitoring algorithm that we evaluate with extensive simulation using different kind of traffic. We describe a wireless resource sharing protocol along with its load estimation algorithms in Chapter 3, and we evaluate them both with simulation and with a real testbed.

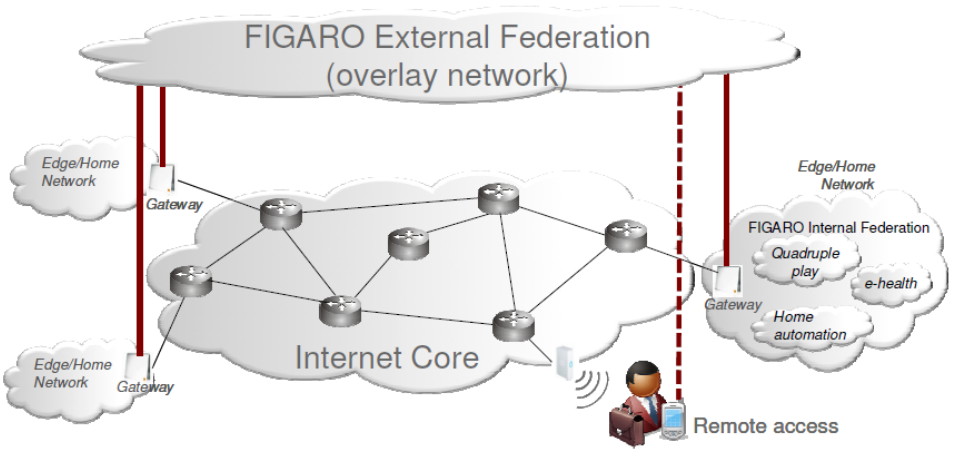


Figure 1.1. FIGARO system architecture.

Enhancing connectivity with energy efficiency: today ICT is accountable for 2-4% of the worldwide carbon emissions and this number is projected to double by 2020 [2]. Telecoms infrastructure and devices account for 34% of the total ICT consumption, and the 95% of this share is due to home and access networks [3]. Within the EU FP7 FIGARO project [1], we address the emerging need for green technologies by proposing a cooperation scheme among federated residential Gateways (GWs).

The cooperation scheme enables GWs within radio range of each other to manage their associated devices, or Wireless Stations (WSs): (i) under low traffic conditions, a GW can hand over the associated WSs to nearby GWs and temporarily switch off; (ii) under high traffic conditions, a GW can selectively hand over one WS at a time in order to lower the congestion. Thus, in order to optimize the usage of the wireless medium and save energy, federated Gateways with overlapping coverages should identify and optimally relocate the Wireless Stations (WSs) among themselves, and, possibly, turn themselves off if a subset of nearby Gateways can adequately support the current load requested by the WSs. Also, an underloaded (or temporarily switched off) Gateway should be called upon for help by Gateways that experience a congested wireless medium, and asked to associate some of their WSs. Such operations require that Gateways have self-load assessment capabilities and run inter-Gateway procedures for WS relocation. In order to address the first aspect, in our work we focus on passive techniques, i.e., solutions that aim at estimating the traffic load by observing some meaningful metrics. Unlike active solutions, passive techniques do not inject probing packets into the network, hence they do not yield additional overhead.

However, existing passive approaches do not fully support multi-rate WLANs with

variable traffic patterns. Metrics based on either the number of associated WSs [4], the channel busy (or, equivalently, idle) time [5], or the aggregated BSS throughput [6], are affected by the payload size and the data rate of the transmitted packets. It follows that such metrics may indicate the availability of bandwidth when the saturation throughput has been already reached, or, conversely, they may detect saturation in presence of available bandwidth. Other techniques, e.g., [7, 8], either apply only to self-estimation of the downlink bandwidth availability, or require changes in the WSs. The model based technique presented in [9] estimates the achievable throughput considering the different nature of traffic and a multirate scenario. However, we cannot use this approach to compute the gateway load metric as it does not compare against the wireless channel capacity.

As for the relocation of WSs, this can be done in either a centralized or a distributed fashion. Centralized solutions are suitable for coverages resulting from controlled placement of the Gateways, as is the case of big enterprises and college campuses, but they are hardly fitting for a residential scenario where each Gateway is independently placed within a household. Examples of centralized relocation schemes that enable Gateways to switch themselves off have been proposed in [10–12]. Also, WS relocation can be either Gateway- or WS-initiated, as in [13]. These approaches have different impact in terms of hardware/software modification to the devices, as well as of the degree of signalling involved. Other solutions have been designed to overcome capacity limitations of single APs. In particular, [14] has suggested the use of TDMA techniques to let WSs access multiple APs at a time, while [8, 15] aggregate the bandwidth available at different APs and balance their load. Such approaches, differing in scope from our work, require modification in the WSs.

1.2 Enhancing connectivity with 3G OnLoading

The adoption of bandwidth hungry[1] mobile devices in the last few years has posed serious provisioning challenges for carriers worldwide. The key idea underpinning our system is to face such challenges by sharing available capacity in existing cellular connection(s).

Sharing unused cellular bandwidth between mobile connections was proposed in Shair [17], where the authors propose a DTN-based approach built on bluetooth to share bandwidth. Our goal is different, we augment existing wired connections with one or more cellular connections, while being agnostic of the underlying technology. Recent work on exploiting different interfaces to increase overall performance includes [18, 19]. Sivakumar et al. [18] describe a method to utilize all available interfaces on the same mobile device. This idea of channel bonding is not new and has been proposed for Ethernet, DSL and Wi-Fi as a means to increase throughput by combining available network interfaces. Badam et al. [19] use a reliable but slow wired connection to exchange control information about

[1] Monthly global mobile data traffic will surpass 10 exabytes in 2016 [16].

a lossy but fast wireless connection in order to obtain a reliable and high speed network. Our solution is different, as we focus on augmenting the bandwidth of wired connections with that of cellular connections. Our work shares specific commonalities in exploiting off-peak bandwidth [20], specifically on how to transfer data while keeping bandwidth costs low. 3GOL utilizes cellular bandwidth whenever available and, by performing active measurements as well as large datasets analysis, we show that it is feasible to onload data onto the cellular network without overloading it.

Conversely, a fair amount of research has gone into devising mobile data offloading [21, 22] solutions that offload data *from* resource constrained cellular networks *onto* (relatively) resource rich wired networks using Wi-Fi and Femtocells. Such cellular network offloading strategies assume that the cellular network is constantly under strain while the wired network has abundant resources to assist. The reality, however, is that wired networks themselves have bottlenecks, first and foremost of which is the access link [23]. For example, ADSL is often constrained by the distance between the customers and the telephone exchange and has much less bandwidth on the uplink than on the downlink, thus effectively limiting the adoption of applications, specially the ones that source content from within the home. The closest work to ours, is the work by Rodriguez et al. [24]. We extend this work in significant ways: we explicitly deal with economic constraints like volume caps, we perform extensive measurements, we implement a fully working prototype that we test in the wild.

A simple back of the envelope calculation (Sec. 4.1) shows that a cellular network has smaller total aggregate capacity by roughly 1-2 orders of magnitude than the corresponding wired network that covers the same geographic area. Still, at a specific location (e.g., a household) and when there is free capacity, a cellular network can provide a comparable, if not greater, amount of bandwidth than the overlapping wired network. This points to opportunities when the resource-constrained cellular network can augment the typically assumed resource-rich wired network. We present a system that "*OnLoads*" data from a wired network onto a cellular network for residential users with access bottlenecks, using existing mobile devices with cellular data connectivity. We present our case for two applications, i.e., augmenting video-on-demand in the initial playout phase (also known as 'Powerboosting') and speed-up the upload of a set of pictures. We propose a network-integrated solution, where there is a single operator that provides both wired and cellular broadband connectivity. While such a solution simplifies economic constraints towards deployment, we still need to quantify if there is enough free capacity in the cellular network *during certain times* to offer opportunities for traffic OnLoading. We perform active measurements of available capacity over different locations and over time in a large European city to answer these questions 4.2. We develop an architecture for 3GOL that can be easily deployed today 4.1. We describe our implementation of 3GOL that does not require modification to either the residential gateway or the cellular infrastructure (Sec. 4.3) and evaluate its performances in the wild (Sec. 4.4). We also compare our greedy multipath scheduler against other multipath schemes as well as with transport level solution like

MTCP [77]. We then consider the case when the operator providing the wired and the cellular network service can be different. In this case, economic constraints and volume caps on cellular data plans make the problem of OnLoading harder: a cell tower might have capacity to offer, but the volume caps on individual 3G data contracts might not permit end users to use 3GOL. We use a large trace to understand typical usage patterns of users with monthly cellular data plans, giving us an idea on the amount of leftover volume that can be used for 3GOL. We show that users typically use less than 20% of their monthly allowed volume; the other 80% is *already-paid-for* volume that we can exploit with 3GOL (Sec. 4.5). In order to avoid cap overruns, we develop a simple estimator that given the past data usage estimates the daily safe volume that can be devoted to 3GOL (Sec. 4.5). We also study the amount of traffic onloaded onto the cellular network (Sec. 4.5) using our traces. A prototype of 3GOL is being tested in 30 households of a large European city, with the intention of a larger scale deployment later.

Furthermore, we also present arguments (Sec. 4.1.4) about the future viability of 3GOL, that can be seen as an appealing long-term technology.

Chapter 2

Tools for available bandwidth estimation and monitoring in 802.11 WLANs

In this chapter we present the tools needed to implement a scheme that, exploiting the concept of Network Federation, achieves energy-efficiency in residential networks. Since users access the Internet through the local WLAN provided by the Internet Gateway while a home, assessing the performance of the WLAN, thus evaluating the available bandwidth on the wireless link, is the first step towards the implementation of such energy-efficient scheme.

Estimating the available bandwidth in a 802.11 WLAN is very challenging because of the shared nature of the wireless medium and due to the unlicensed ISM band used. Indeed, other devices operating on such frequencies create interference, that can affect performances of 802.11 WLAN. Such devices cause electromagnetic interference that can potentially undermine the throughput of residential and enterprise WiFi networks. In Section 2.1.1, we propose an interference estimation and monitoring technique for 802.11 Access Points that can be implemented without specialized hardware and without any modification to wireless stations. We validate our technique with commercial hardware, evaluating its accuracy with different types of interferers. In Section 2.2, we present a lightweight algorithm for the estimation of the achievable throughput of a new wireless station and the available bandwidth monitoring in IEEE 802.11 networks. We consider a multirate WLAN with access point (AP), where there may be both elastic and inelastic traffic flows. Through our algorithm and leveraging previous theoretical results, the AP can estimate: (i) the available bandwidth that a new station wishing to associate with the AP can use, (ii) the impact on the system performance of admitting the new station, (iii) the bandwidth still available (if any) for inelastic traffic. The above quantities can be effectively used for admission control in WLANs and load balancing among APs with overlapping coverages. Indeed, simulation results show that the estimates yielded by our

algorithm accurately reflect the system throughput behavior when there are both elastic and inelastic flows, in the uplink and downlink directions.

2.1 Interference identification in the ISM band

In today's overcrowded, arbitrary deployment of home WLANs, interference is likely coming from a neighboring Access Point (AP), operating on either the same or a different frequency channel. In addition, there are many non-802.11 devices working on ISM bands and representing possible sources of interference. It follows that, in spite of the increasing availability of planning, deploying and managing tools, radio interference remains a key performance bottleneck for home and enterprise WLANs alike.

Very few tools are really helpful to understand how much interference affects the operation of a given wireless network, and how interference patterns evolve over time. To further compound this problem, whatever tools are available require expert usage and only operate as spectrum scanners, often providing little insight on the nature, causes and effects of interference. Among commercial solutions, Airmagnet Spectrum XT [25] and AirMaestro [26] are examples of custom hardware systems that integrate spectrum analyzer functionality to facilitate non-WiFi device detection. Examples of interference estimation techniques based on available bandwidth testing [27, 28] require traffic injection among wireless links, either to build the interference map in absence of real traffic [27], or to perform the actual interference estimation [28]. Furthermore, these techniques relies on strong assumptions, like having fixed data rate for all wireless nodes. Another example of interference estimation tools based on available bandwidth testing [29] requires traffic injection. The downside of these approaches is that they affect normal network operations and require certain traffic patterns to test interfering links, which may be incompatible with realistic traffic scenarios.

A different approach is based on trace collection and subsequent analysis. Proposed solutions aim at analyzing specific aspects of a 802.11 wireless network, ranging from physical and link-level behavior [30–32], wireless station (WS) location and coverage [33], to transport and network layer performance [34]. In [35], traces are captured using several sniffers in a WLAN and a state machine-based learning approach is proposed to identify interference. Similarly, the authors of [36] exploit a large wireless monitoring infrastructure to monitor a production WLAN and perform a cross-layer analysis to diagnose performance problems. While some of these approaches appear to be effective, they only have offline applicability, as they require postprocessing of wireless traces to identify interfering signals. They fail to evaluate the accuracy and agility of interference estimation mechanisms, especially in presence of WS mobility and sophisticated bit rate adaption mechanisms. Also, they do not discuss the integration of their interference estimation mechanisms with applications like power control and channel assignment.

An example of online, passive interference estimation is given in [37], which presents

a methodology to dynamically generate fine-grained interference estimates across an entire WLAN. However, the solution in [37] requires both a second wireless card on the APs and to compute the real-time graph of all interfering nodes. The latter implies the presence of a centralized controller for the entire WLAN, which may not be always available, especially in residential networks. Similarly, [38] uses a specific functionality provided by a recent WiFi chipset to perform online detection of multiple non-WiFi devices including fixed frequency devices (e.g., ZigBee), frequency hoppers (e.g., Bluetooth) and broadband interferers (e.g., microwave ovens).

In this work, we propose a MAC-layer approach to interference estimation by adopting passive measuring techniques. Our solution is implemented at the AP and accounts for all possible causes of interference, specifically: transmissions originated within neighboring Basic Service Sets (BSSs), either operating on the same or on a different frequency channel, and transmissions from non-802.11 devices. The key point of the proposed technique resides in the comparison that the AP performs, for each of the data frames it sends, between the expected time required to successfully transmit the frame and the actual time measured by the AP. In order to understand the impact of interference on the BSS throughput performance, we then extend the computation of saturation throughput [39, 40] by accounting for the interference effects. We implement our solution in a testbed and validate it via experimental results. Unlike previous solutions, ours can be implemented on commercial APs with any WiFi chipset, without requiring either specialized hardware or modifications to the WS.

In the following sections, we will present an approach for estimating the amount of channel occupation due to interferents and a methodology for the computation of the Saturation Throughput that includes such interference.

2.1.1 Inferring Interference

We consider an IEEE 802.11 BSS managed by an AP capable of monitoring local wireless resources. The AP and its associated WSs may transmit frames with different payload size and their data rate may vary according to the experienced channel propagation conditions. The AP collects statistics within its BSS and it carries out such measurements periodically over a time interval, hereinafter referred to as measurement period.

Inferring interference requires that the AP estimates whether the channel is sensed busy because of "legitimate" ongoing transmissions or because of interference. This procedure consists in computing, for each frame k sent by the AP, what is the *expected* time interval $T_e(k)$ that a transmission (either successful at the first attempt or subject to collisions/errors, hence repeated one or more times) should take, were it not affected by interference. This time computation should then be compared against actual *measurements* of transmission interval $T_m(k)$, taken by the AP driver, in order to infer whether unaccounted-for signals (interferers) are cluttering the channel. The latter unduly lengthen

the transmission because the channel is sensed busy even if no legitimate BSS node is actively sending data. The estimation of the fraction of time taken by interferers, I, can be computed as follows:

$$I = \frac{\sum_{k=1}^{K} \left(\frac{T_m(k) - T_e(k)}{T_e(k)} \right)}{K}, \tag{2.1}$$

where K is the total number of frames transmitted by the AP within the current measurement period. $T_m(k) = t_a(k) - t_x(k)$, where $t_a(k)$ corresponds to the ACK notification time, while $t_x(k)$ is the time at which frame k is handed over to the driver for transmission. To cope with the case where the driver manages multiple queues with different priorities, as in 802.11e, when an ACK is received, $t_i(k)$ is updated for all packets already handed over to the driver but not ACKed yet. As for $T_e(k)$, for the sake of clarity, we describe how it is computed by referring to Fig. 2.1.

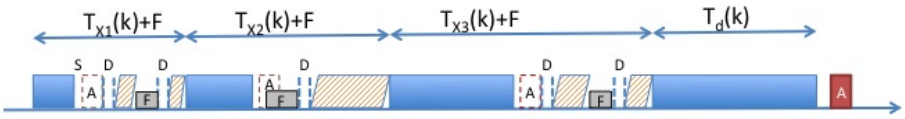


Figure 2.1. Timeline of a repeated transmission by the AP. "S", "A", "D" and "F" stand for, respectively, SIFS, ACK, DIFS, and Freeze.

The figure portrays the case where the AP repeats the transmission of a frame four times before success. Each attempt is renewed after the mandatory random backoff period. It is to be underlined that each retransmission attempt could occur at a lower bit rate than the previous one. Indeed, the MAC-layer rate control procedure implemented in most 802.11 drivers mandates for each MAC frame to be associated with a retry vector. This vector specifies the number of retries to be performed at decreasing bit rates, successively attempted in case a transmission fails (i.e., no ACK is received). These rates are known by the driver. Thus, the duration of each frame transmission can be easily computed by the AP itself. We denote by $T_d(k)$ the duration of the successful transmission, which depends on the data rate and the payload size of the frame and is computed as specified in the 802.11 standard [41]. As for the failed transmissions, we indicate the i-th transmission attempt of frame k as $T_{Xi}(k)$. Such a quantity is composed by: the transmission duration of the data frame, the retransmission timeout T_o (which is set equal to SIFS plus the ACK duration), the backoff time associated with the i-th attempt, and the DIFS time intervals needed to declare the channel idle. The contribution due to the backoff time is set to half the contention window used at the i-th attempt. However, it is important to recall that the backoff counter is frozen by an 802.11 interface whenever the channel is sensed busy. Carrier sensing may be triggered by: (i) transmissions from other WSs in the same BSS, (ii) transmissions from other BSSs operating on the same channel, (iii) transmissions from neighboring BSSs using a different channel or from non-802.11

devices. Note that the first case is a "legitimate" interruption and, as such, is not classified as interference. As for data frames transmitted within other BSSs operating on the same channel, they can be received by the AP of the tagged BSS through a virtual interface operating in monitoring mode. In our computation, we separately account for such contribution and denote it by $\delta(k)$.

Based on the description and definitions above, the expected transmission interval $T_e(k)$ can be obtained as:

$$T_e(k) = \sum_{i=1}^{A} T_{Xi}(k) + T_d(k) + \text{SIFS} + \text{ACK} + \sum_{j=1}^{N} (T_{RXj}(k) + T_{NRj}(k)) + \delta(k) + \epsilon, \tag{2.2}$$

where:

- A is the number of failed transmissions for frame k;
- SIFS and ACK are, respectively, SIFS and ACK durations. The ACK duration is computed by considering its actual transmission rate;
- N is the number of WSs in the tagged BSS;
- $T_{RXj}(k)$ and $T_{NRj}(k)$ are the duration of, respectively, successful and failed transmission cycles by other WSs within the tagged BSS (during which the AP has to freeze its own backoff while trying to transmit frame k);
- $\delta(k)$ is the airtime taken by non-colliding transmissions from neighboring BSSs operating on the same channel, while the AP tries to transmit frame k;
- ϵ is the approximation error due to the granularity with which time intervals are detected by the AP's driver.

We remark that the AP has no knowledge of failed transmissions by WSs. However, assuming a symmetrical channel between AP and WSs, the same packet error rate (PER) may apply to any transmission, thus yielding a rough estimate of the percentage of WS transmissions that ultimately fail.

In conclusion, by computing $T_e(k)$ and measuring $T_m(k)$ for each data frame, the AP can obtain an estimate of I at each measurement interval. *This method can be effectively implemented at run time and it does not require any knowledge of the past.* Since per-frame processing and statistics are already included in any WiFi driver, our method adds only a negligible complexity. The technology we refer to can be any among a, b, and g; also, the proposed technique can be easily extended to DCF with handshake as well as to the 802.11e/n EDCA.

2.1.2 Computation of the Saturation Throughput

For a practical understanding of the impact of interference, we now introduce a methodology to compute the theoretical saturation throughput that would be achieved if the BSS operated in an unhindered scenario. We will then estimate the contribution of interferers and derive the *theoretical saturation throughput in presence of interference.* In Sec. 2.1.4, the latter quantity will be compared against actual live measurements.

Again, we consider an IEEE 802.11 BSS managed by an AP capable of monitoring local wireless resources at the MAC layer. During each measurement period, the AP computes: average size of the frame payload (P), maximum payload size (P_{max}), average data rate for data frames and for ACKs, average PER (p_e), and number of active WSs within the BSS (N). The AP considers a node in the BSS (either itself or a WS) to be active if the node has successfully transmitted at least one data frame within the last measurement period. The average PER could be computed based on the modulations used for the transmissions in the measurement period, their associated signal-to-noise (SNR) ratio, and assuming independent bit errors on the channel. Since this method gives poor results due to multipath effects and inaccurate SNR measurements by the hardware, we estimate p_e as the ratio of the number erroneously received frames to the number of transmitted frames. To compute the numerator, at the receiver we count the CRC errors (at the PHY and MAC layer), while at the transmitter we count all unsuccessful transmission attempts at the physical layer. The latter results in a worst case PER estimation, as collisions are also included in the count[1]. Conversely, the calculus at the receiver underestimates the actual number as the physical layer cannot always decode a corrupted frame and hand it to the MAC layer.

We now introduce the theoretical saturation throughput S_{th}, defined as the value of maximum achievable throughput in the BSS given the current traffic load. The saturation throughput is given in [40], which extends the original model in [39] to account for errors due to channel propagation conditions:

$$S_{th} = \frac{N\tau(1-\tau)^{N-1}P(1-p_e)}{E[T]}. \tag{2.3}$$

In (2.3), τ is the probability that a node (either a WS or the AP) accesses the medium at a generic time slot[2] and $E[T]$ is the average duration of a time interval in which an event occurs, namely, an empty slot, a successful transmission, a transmission failed due

[1]Collisions cannot be discriminated from errors caused by harsh channel conditions without changing the WS software or the 802.11 protocol.

[2]Considering a slotted time is the main approximation of this model.

to channel errors, or a collision. $E[T]$ can be computed as:

$$\begin{aligned} E[T] = (1-\tau)^N \sigma + \\ [N\tau(1-\tau)^{N-1}(1-p_e)]T_s + \\ [1-(1-\tau)^N - N\tau(1-\tau)^{N-1}]T_c + \\ [N\tau(1-\tau)^{N-1}p_e]T_{err} \end{aligned} \tag{2.4}$$

where σ is the slot time duration as defined in the 802.11 standard. By assuming that the retransmission timeout is equal to SIFS plus ACK, the average duration of a successful transmission, T_s, and of an erroneous transmission, T_{err}, are equal and given by:

$$T_s = T_{err} = T_d + \text{ACK} + \text{SIFS} + \text{DIFS} \tag{2.5}$$

where T_d is the average frame duration computed as specified by the standard, according to the BSS type, and using the average payload size and average rate measured by the AP at the BSS level.

As far as the average collision duration is concerned, its exact computation would require the AP to be aware of the number of nodes that are hidden with respect to each other. The work in [39,40] does not account for hidden WSs and the approaches proposed in the literature, e.g., [42], are not viable in our set up, as we do not require the AP to have knowledge of the users distribution within its coverage area. Thus, we approximate the average collision duration by making a worst-case assumption. Each collision involves a data frame of maximum payload size P_{max} among those observed by the AP during the measurement period. Clearly, the average collision time is overestimated in absence of hidden WSs, leading to underestimating the saturation throughput. It follows that T_c is computed as T_{err} but using P_{max} instead of P. We also observe that the AP can easily compute τ using the following equations [40]:

$$\begin{aligned} p &= 1-[(1-\tau)^{N-1}(1-p_e)] \\ \tau &= \frac{2(1-2p)(1-p^{m+1})}{W(1-(2p)^{m+1})(1-p)+(1-2p)(1-p^{m+1})} \end{aligned}$$

where p is the the conditional probability that a transmitted data frame encounters a collision or is received in error in saturation conditions; W is the minimum contention window; m is the retransmission limit. While deriving our results, we set $W = 31$ and $m = 5$. Note that p and τ have to be obtained through numerical methods, as described in [39,40]. We can pre-compute all values of τ as a function of N and p_e, and perform at each measurement period a simple look-up. For instance, if we consider N varying from 1 to 30, τ from 0.05 to 1 with 0.05 resolution, and a half precision floating point representation (16-bit) for τ, we would need only 1.2 MB memory space to store all values. As τ ranges from 0 to 1, this requirement can be further reduced by using an ad hoc code.

We stress that, although S_{th} represents the saturation throughput considering the node average behavior, it accounts for the different air time that WSs take to transmit their frames. Indeed, the average payload size P and $E[T]$ in (2.3) depend on the payload, data rate and access rate of each single WS.

In order to reflect the effects of all types of interferers, we let the AP keep track of I, computed as in (2.1), and of the average δ, computed as $\left(\sum_{k=1}^{K} \delta(k)\right)/M$ where M is the measurement period duration. We then discount from S_{th} the portion of throughput that cannot be achieved due to the two components above and, finally, compute the saturation throughput in presence of interference, S_{in}:

$$\begin{aligned} S &= (1-\delta)S_{th} \\ S_{in} &= (1-I)S\,. \end{aligned}$$

2.1.3 System Implementation: A MAC-layer Approach

As mentioned, our solution has the following desirable properties: (i) it allows online interference detection, from both WiFi and non-WiFi devices; (ii) it does not need specialized hardware; (iii) it runs on an AP without additional software (or hardware) modification to WSs.

We implement the estimation procedure described in Sec. 2.1.1 at the MAC layer, specifically within the mac80211 module of the Linux wireless driver *compact-wireless 2011-21-01* [43]. We select the MAC layer because it is the highest layer in the stack from which we can retrieve the data rate for every data transmission attempt as well as for ACKs.

The estimation procedure implies the implementation in the AP driver of: (i) the additional passive measurements described in Secs. 2.1.1 and 2.1.2, and (ii) the estimation of I. All required measurements, along with I, are made available to the application layer by using debugfs at each measurement period. Note that, since we modify only the mac80211 module and not the AP hardware or physical driver, such measurements can work on any wireless chipset. We then implement the computation of S and S_{in} with a simple user-space program that reads the measurements from file system and process them as described in Sec. 2.1.2. Given its low complexity, such computation can be implemented on any device.

2.1.4 Experimental Evaluation

We evaluate the validity of our approach through a testbed deployed in a university laboratory at Politecnico di Torino. In the laboratory, there are 18 detectable APs, which are part of 6 different SSID, whose signal is received at an average strength of -83 dBm. We use two stations, WS1 and WS2, each associated to a 802.11g AP. We name the access

points, AP1 and AP2, and refer to their BSSs as BSS1 and BSS2, respectively. Each AP runs the modified driver for the computation of I and the application for the computation of S and S_{in}. We connect the APs to a switch, which, in its turn, is connected to a desktop PC that we use as a traffic sink. All equipment is placed on a desk, at approximately 1 m height. AP1, AP2, WS2 and WS1 are placed at the vertices of a square of 30 cm side, in clockwise order starting from the right-top vertex (see Fig. 2.2).

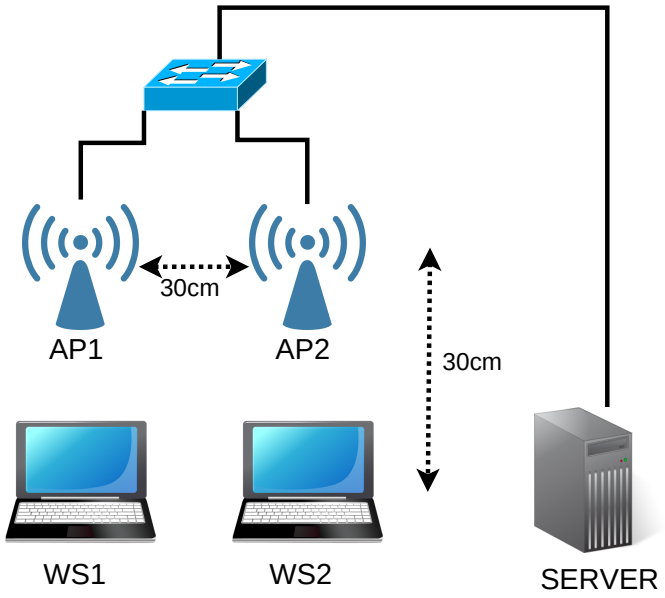


Figure 2.2. Testbed deployment used for validation.

APs are implemented in embedded wireless nodes featuring an Alix PC Engines motherboard, equipped with an AMD Geode 500 MHz processor and a IEEE 802.11 b/g compliant Wistron DCMA-82 Atheros wireless card. Each Alix runs OpenWrt Backfire, a Linux distribution for embedded devices. WSs are represented by ASUS notebooks, model P52F, with Ubuntu 12.04. Both APs are powered through PoE (Power over Ethernet). BSS1 operates on channel 6 and is our tagged BSS, while BSS2 acts as an interferer. In particular, BSS2 operates either on the same channel as BSS1, or on another among the 802.11 standard ones (in the 2.4 GHz band). We set the transmit power of both APs and WSs to 20 dBm.

Since we are interested in evaluating the throughput loss caused by interferers and the accuracy of our solution in estimating I and S_{in}, we saturate BSS1's capacity with a downlink UDP flow from the server to WS1 (through AP1), at a rate of 30 Mbps. The choice of UDP, rather than TCP, traffic allows us to precisely control the load without unpredictable effects due to congestion control mechanisms.

We start by setting BSS2 on channel 11 so as to assess the performance of our solution

in presence of 802.11 interferers operating on a different frequency channel. Specifically, we consider a dynamic traffic scenario where no traffic is generated within BSS2 in the interval [0, 25 s], then the server starts transmitting a UDP downlink flow at 30 Mbps to WS2 (through AP2). The results are shown in the top plot of Fig. 2.3, which compares the throughput achieved by AP1 (Thr) to the saturation throughput estimations S and S_{in}. Recall that S_{in} differs from S as it accounts for the interference term I. The bottom plot of Fig. 2.3 instead depicts (i) the estimated time fraction I during which the channel within BSS1 is sensed as busy due to BSS2, (ii) the average PER (p_e), and (iii) the time fraction δ during which AP1 senses the medium as busy due to other BSSs operating on the same channel. Note that δ is determined by those BSSs, out of the 18 that are present, that operate on the same channel as BSS1 (channel 6). All these quantities are expressed as percentages.

We observe that S_{in} closely matches the throughput measured by AP1 (Thr) before and after the interfering flow is enabled within BSS2. This clearly indicates that I correctly reflects the negative effect of a flow activated within a BSS operating on a different frequency channel (with BSS2 achieving an average aggregate throughput of 15.7 Mbps). It is also important to remark that the quantitative impact of interferers operating on non-overlapping channels may be severe, especially in the case of devices in close proximity (as in our case). Furthermore, when the interferer is enabled, not only I but also p_e increases. This suggests that transmissions within BSS2 may cause collisions at AP1, beside making AP1 detect the channel as busy.

We now extend our evaluation by varying both the channel used by the interferer (BSS2) and its offered load. We set AP2 to operate on a different channel at each run, namely, 6, 7, 9 and 11. For each channel, we carry out several experiments (each lasting 50 s) so as to vary the generation rate of the interfering traffic (i.e., the rate of the UDP downlink traffic flowing from the server to WS2). In all cases, we obtain an excellent match between the throughput achieved by AP1 and the estimated saturation throughput S_{in}. Due to lack of space, we only plot in Fig. 2.4 the comparison between the throughput measured by AP1 (Thr), S and S_{in} for channel 6 (top) and 11 (bottom). The results are presented as functions of the offered load in BSS2 (i.e., the interfering traffic generation rate). Each point plotted in the figure is the average over time of the values obtained during a 50-s experiment. Clearly, Thr decreases as the interferer traffic load increases. Again, S accurately estimates the throughput of BSS1 only in case of co-channel interference, i.e., when BSS2 operates on channel 6. Indeed, in this case the majority of the frames transmitted by AP2 are received by AP1's monitoring interface, which can correctly account for it through the quantity δ. Conversely, when the interferer operates on channel 11, S cannot reveal its presence, while S_{in} perfectly reflects its effects.

Figs. 2.5 (a) and (b) show the value of the time fractions I and δ, as the channel on which BSS2 operates and BSS2's traffic load vary. In case of co-channel interference (i.e., BSS2 on ch6), I is low as this quantity does not account for it; conversely, δ well captures such interference. Furthermore, we observe that the traffic load within the interfering BSS

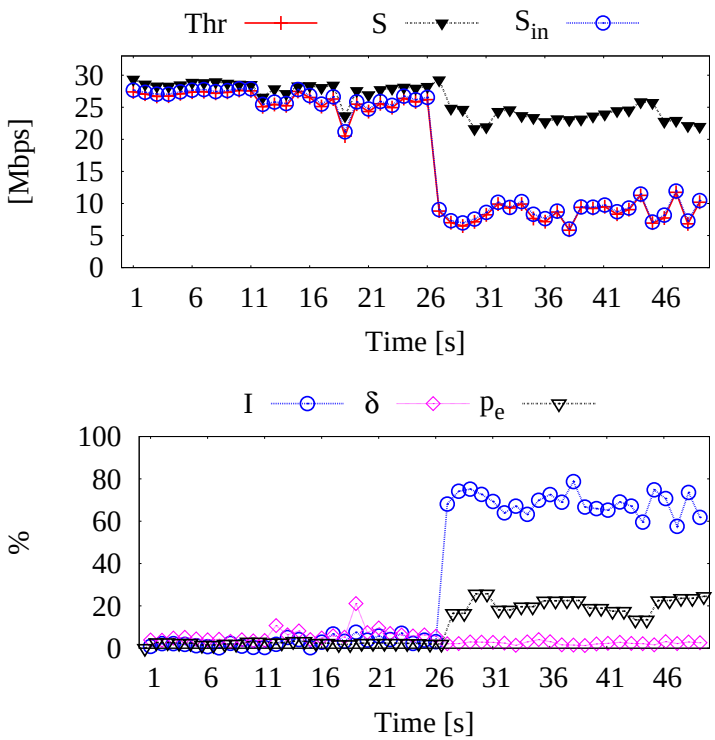
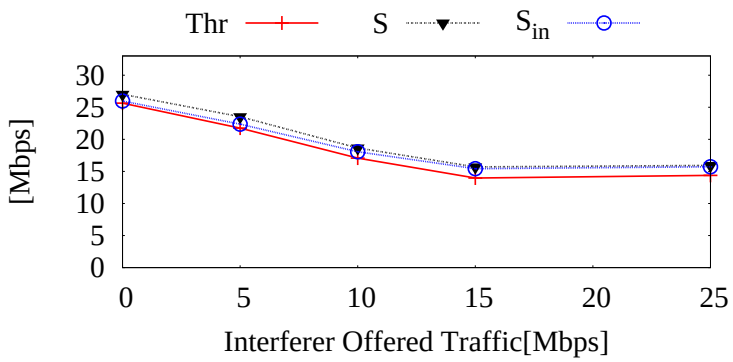


Figure 2.3. 802.11 interferer enabled at second 26 on a non-overlapping channel (ch11). Top: comparison between BSS1 throughput (Thr), the estimated saturation throughput S and that accounting for interference (S_{in}). Bottom: average PER (p_e) and time fractions during which the channel is sensed as busy due to co-channel interference (δ) and to BSS2 operating on ch11 (I).

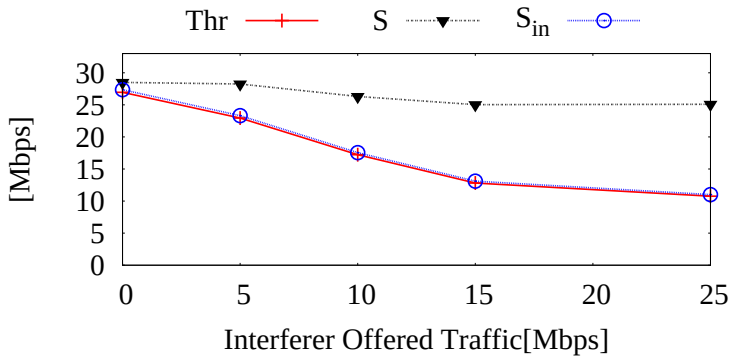
has a significant impact and this is correctly represented by both I (for channels other than 6) and δ (for channel 6).

Fig. 2.5(c) presents the results obtained with BSS2 operating also on channels 1, 3, and 5 and setting the rate of its traffic flow to 25 Mbps. We note that the behavior of S_{in} closely matches that of the measured throughput Thr on all channels. The values of Thr however change significantly depending on the considered channel: this is due to the different multipath conditions affecting the channels. We repeated the experiments placing the devices at different locations and we obtained similar results, which we omit for brevity. The fact that S_{in} and Thr consistently match suggests that our methodology provides an accurate estimation no matter the working environment that is considered.

Next, we focus on non-802.11 interferers and employ a pair of Bluetooth nodes and an analog video sender. First, we start a file transfer between the Bluetooth nodes, with the sender and the receiver placed, respectively, at the left and right side of WS1 and equally spaced from WS1 by 30 cm. As before, the file transfer starts at second 26 and the experiment lasts 50 s. As shown in Fig. 2.6, initially the interfering traffic flow causes



(a) Achieved vs. saturation throughput on ch6



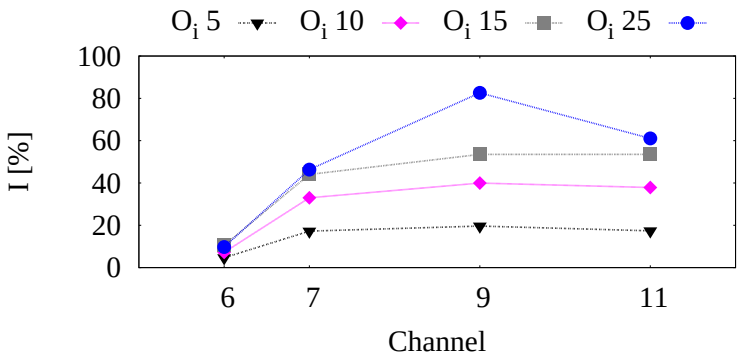
(b) Achieved vs. saturation throughput on ch11

Figure 2.4. Achieved vs. saturation throughput in BSS1 when BSS2 operates on the same channel (ch6) (a) and on a different channel (ch11) (b).

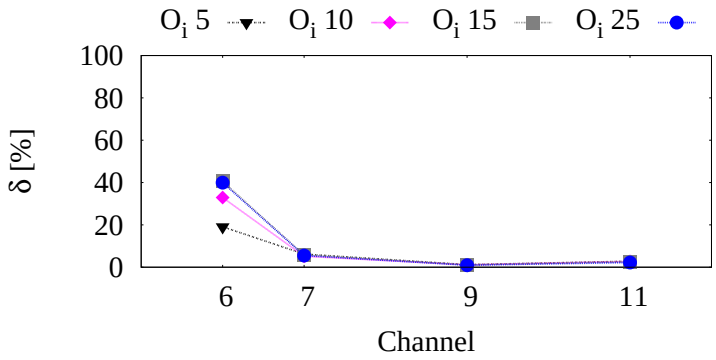
a noticeable throughput degradation at AP1, but after a couple of seconds the performance improves again. This effect is due to the Bluetooth adaptive frequency hopping (AFH) scheme, which tends to avoid channels characterized by high PER. Again, S_{in} closely follows the behavior of Thr in all phases of the experiments, confirming the validity of our technique.

We then let the video sender act as interferer. It is placed on the left of WS1, at 30 cm distance, turned on at second 26. The interference generated is so strong that the throughput drops almost to zero after a couple of seconds. Again, S_{in} is able to quickly reflect the behavior of the measured throughput, as highlighted by the results in Fig. 2.7.

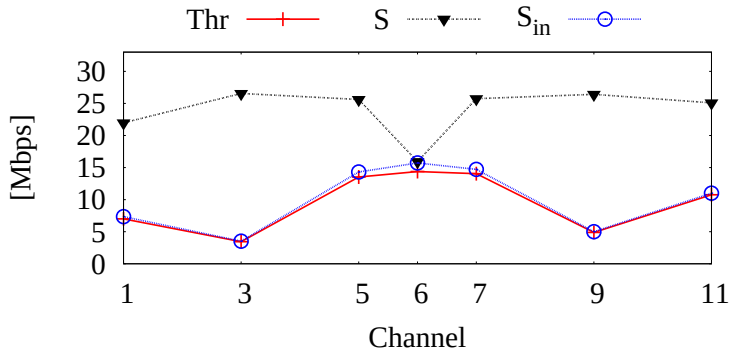
We now try the same video sender shifting its central frequency on channel 6. As before, the video sender is turned on at second 26. We observe a higher disturbance on the throughput Thr, which almost drop to zero. Surprisingly, after some seconds Thr grows again but it becomes more unstable. This behavior suggests a certain degree of adaptability of the wireless card, that is able to mitigate the negative impact of interferers. However, the presence of the interferer is detected by a sudden increase of p_e and by



(a) Interferer channel occupation, I



(b) δ



(c) Achieved vs. estimated saturation throughput

Figure 2.5. (a) Interference estimation I and (b) time fraction during which AP1 detects the channel busy due to co-channel interference (δ). The results are plotted as the channel used by BSS2 and its traffic load vary. (c) BSS1 throughput (Thr) vs. estimated saturation throughput S and S_{in}, as the channel used by BSS2 varies.

the oscillation of I, that accurately reflects the interferer disturbance into S_{in} as show in Fig. 2.8.

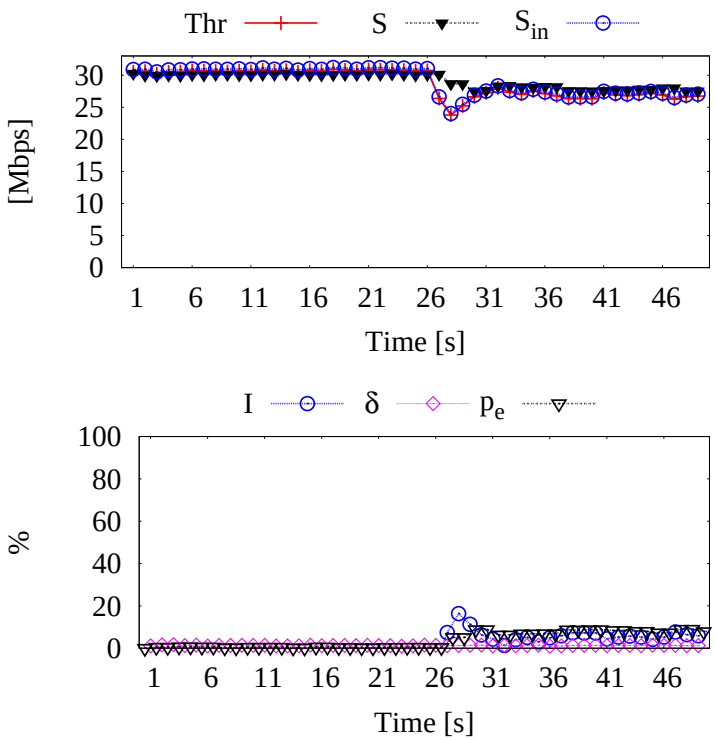


Figure 2.6. Bluetooth file transfer enabled at second 26. Top: comparison between BSS1 throughput (Thr), estimated saturation throughput (S) and that accounting for the interference effect (S_{in}). Bottom: average PER (p_e), and time fractions during which AP1 senses the channel as busy due to co-channel interference (δ) and to the Bluetooth interferer (I).

Finally, we evaluate a Narrow-Band (NB) interferer, i.e., an interferer that occupy a bandwidth of 200 kHz, that models the presence of devices like RF movement detectors used in residential security alarms. We place the NB interferer at the central frequency of BSS1 (CH 11) and then we shift it to the right at +500 kHz, +2 MHz and +3 MHz. In all cases we observe (i) a major disturbance on Thr and a reduced stability of the same after the interferer is turned on, (ii) a clear reflection of the interferer presence on both p_e and I which translate to a close match between S_{in} and Thr. We can not identify a clear trend between the throughput evolution and the distance in frequency between the interferer and CH 11 central frequency. We note that when enabled at CH 11 central frequency, the NB interferer produces a throughput drop that is almost fully recovered after 6-8 seconds. Conversely, the throughput almost drops to zero with the NB interferer at +500 kHz and at +2 MHz, while at +3 MHz it brings a less disruptive impact. The results are reported in Fig. 2.9.

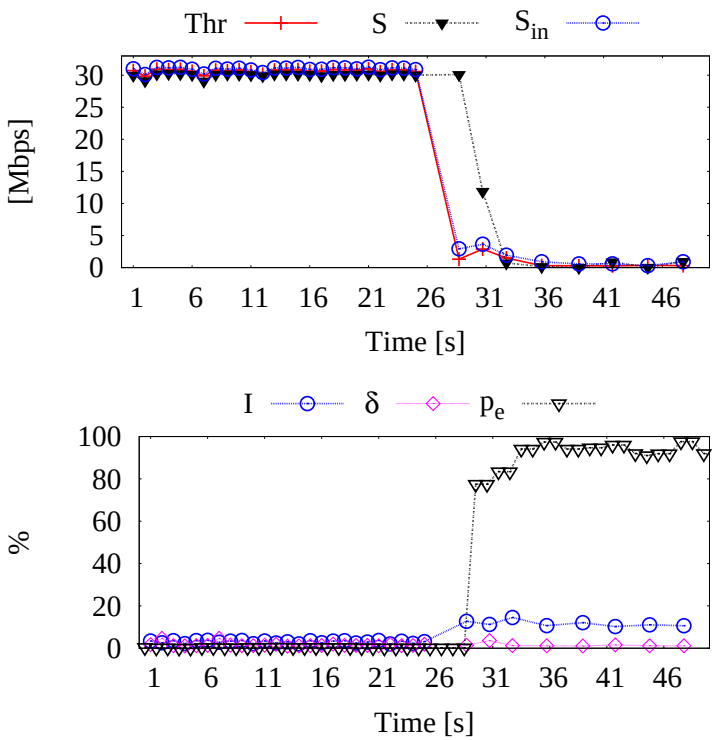


Figure 2.7. Analogue video sender enabled at second 26. Top: comparison between BSS1 throughput (Thr), estimated saturation throughput (S), and that accounting for the interference effect (S_{in}). Bottom: average PER (p_e), and time fractions during which AP1 senses the channel as busy due to co-channel interference (δ) and to the interference caused by the video sender (I).

2.1.5 Final remarks on interference estimation

We have designed and implemented a technique for interference estimation in 802.11 WLANs, which accounts for all possible sources of interference. It can be implemented at the access point and does not require any specialized hardware, changes in the 802.11 standard or in the wireless stations. We have validated our technique in a 802.11g network with different types of interferers. Experimental results show that our solution can estimate the impact of interference with excellent accuracy, under different scenarios. In the following sections and chapter of this thesis, we will use S instead of S_{in} as the symbol for the saturation throughput including the contribution of interferers.

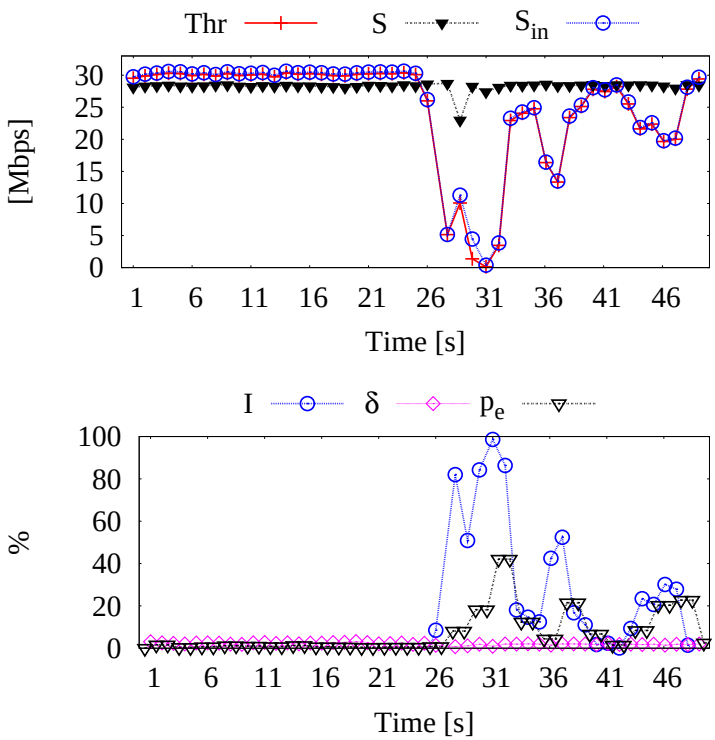
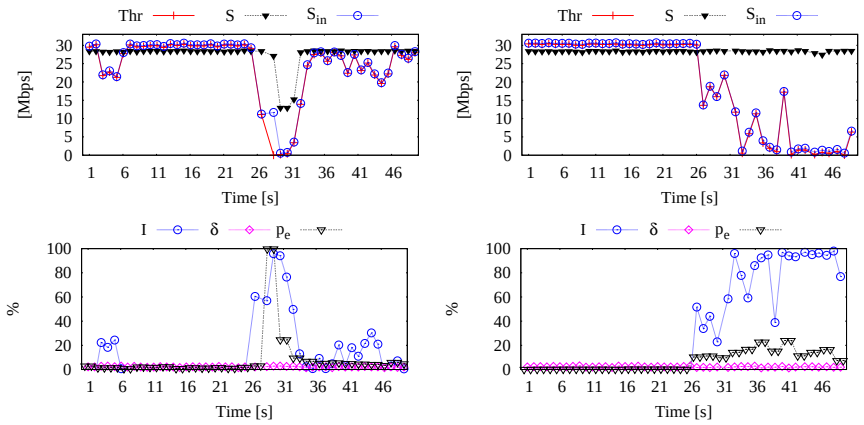


Figure 2.8. Analogue video sender enabled at second 26 on channel 6. Top: comparison between BSS1 throughput (Thr), estimated saturation throughput (S), and that accounting for the interference effect (S_{in}). Bottom: average PER (p_e), and time fractions during which AP1 senses the channel as busy due to co-channel interference (δ) and to the interference caused by the video sender (I).

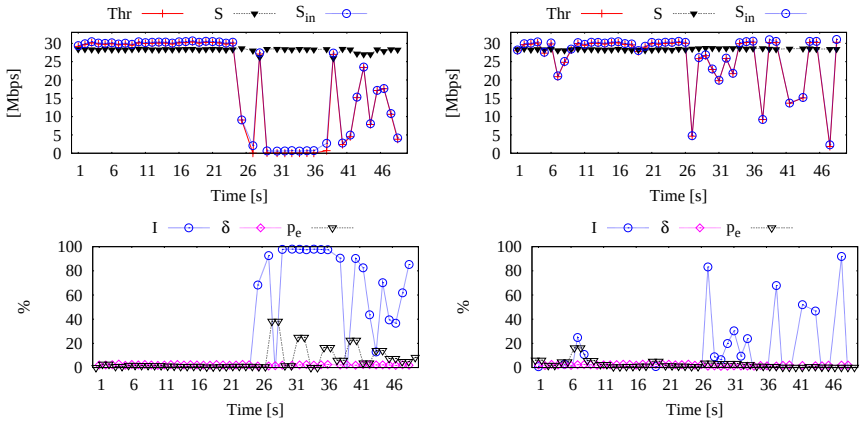
2.2 Available Bandwidth Estimation (ABE)

The current trend of providing wireless users with ubiquitous connectivity to the Internet has determined a wide deployment of access points (APs) using the IEEE 802.11 technology. Often, especially in highly populated areas, one location is under the coverage of more than one AP, even though a wireless station (WS) that happens to be in that location is associated to a single AP. The selection of the AP is usually dictated by users' preferences or subscription plans, and, as such, it disregards congestion levels or channel quality issues. However, if we abstract from the current scenario and address a broader picture carrying concerns such as energy-efficiency and electromagnetic pollution, a solution that minimizes the overlapping coverages of APs without degrading the performance expected by the WSs is desirable. In order to achieve such a goal, APs with overlapping coverages should identify and optimally split the WSs among themselves, and, possibly, turn themselves off if a subset of APs can adequately support the current load requested by WSs.



(a) At CH 11 central frequency

(b) At +500 kHz from CH 11 central frequency



(c) At +2 MHz from CH 11 central frequency

(d) At +3 MHz from CH 11 central frequency

Figure 2.9. Narrow-band interferer enabled at second 26 on channel 11 with different central frequencies. Top: comparison between BSS1 throughput (Thr), estimated saturation throughput (S), and that accounting for the interference effect (S_{in}). Bottom: average PER (p_e), and time fractions during which AP1 senses the channel as busy due to co-channel interference (δ) and to the interference caused by the narrow-band interferer (I).

To realize the above vision, discounting the obvious constraints currently imposed by contractual obligations, it is imperative that the APs should gauge the impact of one WS associating to a specific AP. Indeed, it is well known that users within the coverage of the same 802.11 AP share the available bandwidth by using the DCF at the MAC layer, which grants users an equal long-term channel access probability.

However, when different data rates are used and all other parameters are equal, a WS transmitting at low data rate will keep the channel busy for a longer time with respect

to a WS using a high data rate, thus degrading the achievable throughput of the whole network.

We aim at evaluating the impact of the admission of a new WS (or, equivalently, of a new traffic flow) in a multirate WLAN scenario. Unlike previous work (see Sec. 2.2.1), we present an algorithm that an AP can run to sensibly predict two important quantities. First, the available bandwidth, which is measured through a parameter, called b-metric, that we introduce. Second, the throughput decrease that, given the current load of the AP, the association of an additional WS will cause to the currently associated WSs.

The algorithm is based on online measurements by the AP integrated with previous theoretical results, and does not require explicit signaling between AP and WSs. Also, it accounts for the presence of both *elastic* and *inelastic* traffic, namely TCP and UDP traffic flows, in view of estimating the throughput achievable by the WSs as well as their losses. As a matter of fact, the user experience is severely affected by losses while running real-time audio/video applications supported by UDP, such as VoIP. On the contrary, losses on applications supported by TCP, such as HTTP or FTP, result in a throughput degradation which could be annoying for the user but not impairing. Hence, we need to take into account the nature of the existing traffic flows and their different impact on the user satisfaction level.

We point out that our algorithm can be seen as a stepping stone toward the definition of a more comprehensive framework, where traffic admission control and dynamic load balancing among APs with overlapping coverages can be achieved.

2.2.1 ABE related work and motivation

Here, we briefly recall the most popular approaches to load estimation in WLANs, and explain why they cannot be applied to a multirate Basic Service Set (BSS) in presence of different types of traffic.

We focus on passive techniques since they estimate the traffic load without inject probing packets into the network, hence without additional overhead. As an example, in [44] the load is related to the number of WSs associated with the AP, which is a valid approach when all WSs have the same behavior (e.g., same traffic pattern, payload size, data rate). Under more general conditions, other metrics have been proposed, based either on the channel idle (or, equivalently, busy) time [45–47] or on the aggregated BSS throughput [48]. However, when the BSS nodes always have data to transmit, the idle time introduced by interframe spaces and backoff periods, as well as the value of saturation throughput, depends on the payload size and on the data rate of the transmitted packets. It follows that such metrics may indicate saturation in the presence of available bandwidth, or, conversely, availability of bandwidth when saturation has been already reached. This latter effect can be observed in Fig. 2.10, where the time evolutions of the normalized idle time and available bandwidth (computed as one minus the normalized aggregate throughput) are compared to the behavior of the throughput itself. The results

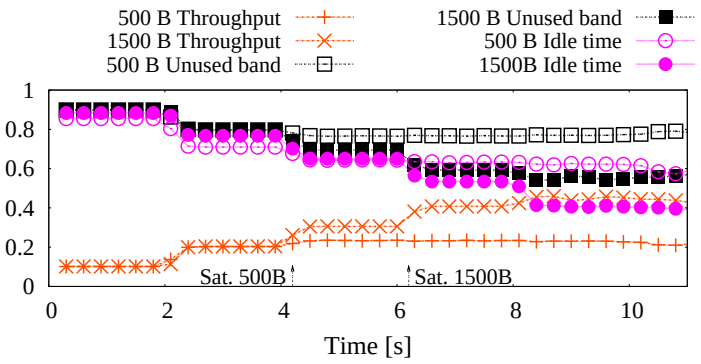


Figure 2.10. Time evolution of the normalized aggregate throughput, unused bandwidth and idle time, when 6 WSs become active at different time instants.

refer to an IEEE 802.11g BSS with one AP and 6 WSs; all nodes initially transmit at 54 Mbps and then vary their data rate through the AARF scheme [49] . Every two seconds between 0 s and 10 s, a WS starts generating a CBR traffic flow at 5.5 Mbps. Two different values of payload are considered, namely, 500 and 1500 bytes. We can clearly see that when the throughput reaches saturation, both the idle time and the available bandwidth metrics stabilize to positive values, which are different and vary according to the considered scenario. It is well known that users within the coverage of the same 802.11 AP share the available bandwidth by using the Distributed Coordination Function (DCF) at the MAC layer, which aims at granting users an equal long-term channel access probability. However, when different data rates are used and all other parameters are equal, a WS transmitting at low data rate will keep the channel busy for a longer time with respect to a WS using a high data rate, thus degrading the achievable throughput of the whole network. This behavior is also known as the "performance anomaly" of 802.11, described by Heausse et. al in [50].

In order to address these shortcomings, we propose an algorithm for bandwidth monitoring and a new metric that account for varying payload size and data rate, and for the presence of both elastic and inelastic traffic [60].

2.2.2 System scenario and preliminaries

We consider an IEEE 802.11 BSS with AP. The technology we refer to can be any among a, b, and g; also, the solution is described for the standard 802.11 MAC without RTS/CTS, but it can be extended to the DCF with handshake as well as to the 802.11e EDCA.

The WSs within the BSS can be sources or destinations of elastic or inelastic traffic flows, i.e, flows that use either TCP or UDP at the transport layer. At the MAC layer, the AP and the WSs may transmit frames with different payload size and their data rate may vary according to the experienced channel propagation conditions.

We assume that the AP can access the "protocol type" field in the IP packets, and collects statistics on the transmissions observed over the medium, such as the average payload size and data rate used for packet transmission within the BSS and the throughput experienced by the nodes.

More specifically, as in [59] the AP carries out such measurements over time intervals, named *cycles*. A cycle is defined as the minimum between a time T_{max} and the period needed to let (1) each active WS successfully send at least one data frame carrying inelastic traffic, and (2) the AP successfully transmits at least one data frame carrying inelastic traffic to every WS for which it has data to send. The AP considers a WS to be active in the generic cycle j if it successfully receives from the WS at least one data frame within a time T_{max} since the current cycle starting time. Likewise, the AP is active in cycle j if it has sent at least one frame within the cycle. In the following, we denote by $C(j)$ the duration of cycle j and by $\mathcal{N}(j)$ the set of nodes (WSs and AP) that were active in the cycle ($\mathsf{N(j)} = N(j)$). Fig. 2.11 shows an example of three transmissions sequences in a BSS including three WSs (A, B, C) and of the cycle instances that occur. Dashed boxes represent a transmission from the AP to a WS, while boxes with solid outline represent a transmission from a WS to the AP. Note that in the second case depicted in the figure, C is active in the $(j-1)$-th cycle and becomes inactive in the next one; In the third example, we highlight that only successful transmissions are taken into account to determine a cycle.

The AP keeps track of the number of active WSs and, for each cycle j, computes the duration $C(j)$. Then, at each cycle j and for each active WS k, the AP computes the uplink throughput for elastic and inelastic traffic of k, denoted by $\eta_k(j)$ and $\nu_k(j)$, respectively, as the ratio of the amount of data successfully transmitted by the WS in cycle j to the cycle duration. Similarly, considering the amount of data successfully received by the WSs in cycle j, the AP computes its own throughput for elastic and inelastic traffic, denoted by $\eta_{AP}(j)$ and $\nu_{AP}(j)$, respectively. Then, for each of these quantities, the AP computes a running average, $\overline{\eta}_k(j)$ and $\overline{\nu}_k(j)$ with $k \in \mathcal{N}(j)$, using the well-known exponential smoothing filtering. E.g., for the elastic traffic throughput of node k, we have $\overline{\eta}_k(0) = \eta_k(0)$ and $\overline{\eta}_k(j) = \alpha\eta_k(j) + (1-\alpha)\overline{\eta}_k(j-1)$, where $0 < \alpha < 1$.

Likewise, for each frame successfully transmitted by a WS or by the AP itself, the AP observes the payload size for elastic/inelastic traffic and the used data rates, and it computes the corresponding running averages: $\overline{P}_{k,e}(j)$, $\overline{P}_{k,i}(j)$, and $\overline{R}_k(j)$ ($k \in \mathcal{N}(j)$). For the data rate, the AP stores only one value because rate adaptation algorithms do not distinguish between elastic and inelastic flows. Also, the AP computes the running average of the data rate, $\overline{R}(j)$, and of the payload size, $\overline{P}(j)$, over all data frames, carrying either elastic or inelastic traffic, that it successfully sends or receives.

Next, we introduce a fundamental quantity for our bandwidth monitoring algorithm. Let us consider cycle j. At the end of the cycle, the AP computes the (aggregate) saturation throughput $S(j)$, as defined in 2.3. in [40], which extends the original Bianchi's

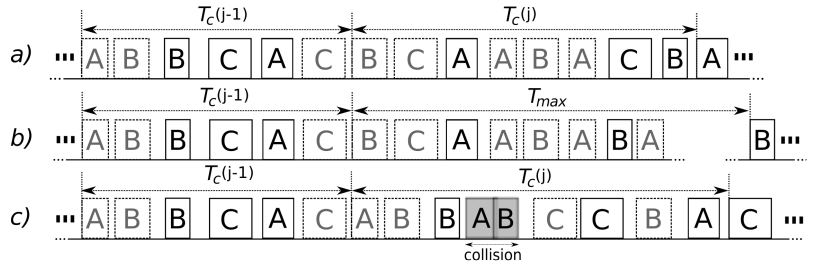


Figure 2.11. Example of transmission cycles, in the case of three WSs (A, B, C) belonging to the BSS.

model [39] in presence of errors due to channel propagation conditions. For sake of completeness, we report the formula of $S(j)$ with our new notation:

$$S(j) = \frac{N(j)\tau(j)[1-\tau(j)]^{N(j)-1}\overline{P}(j)(1-\overline{p}_e(j))}{E[T(j)]}. \tag{2.6}$$

The expressions of $\tau(j)$ and $E[T(j)]$ can be derived following [40] and have been reported in 2.1.2 together with the explanation on how to estimate $\overline{p}_e(j)$. In order to separately evaluate the bandwidth estimation algorithm, we assume to have no interferers and no neighboring BSSes. We remark that we have already evaluated the impact of both interferers and neighboring BSSes in 2.1.4. In order to account for such effects, the expression of $S(j)$ must be modified using 2.6.

Using (2.6), the AP computes the average per-node throughput under saturation conditions, as $S_n(j) = S(j)/N(j)$. Note that $S_n(j)$ represents the saturation throughput for a node with average behavior, i.e., a node using a payload size $\overline{P}(j)$ and a data rate $\overline{R}(j)$.

2.2.3 ABE monitoring algorithm

As mentioned earlier, our aim is threefold: (i) estimating the throughput that either elastic or inelastic traffic flows newly originated within a BSS can achieve, (ii) gauging the impact that the newly generated traffic will have on the performance of inelastic traffic flows already present within the BSS, and (iii) monitoring the available bandwidth for inelastic traffic (if there is any left). We focus on the effects of the new traffic flows on inelastic traffic only, since this type of traffic has more stringent quality of service requirements.

Without loss of generality, in the following we describe our estimation algorithm in the case where a new WS, which may be source or destination of elastic or inelastic traffic, wishes to join the BSS. The extension to the case where a WS already associated to the BSS wishes to start a new traffic flow and notifies the AP about it (as foreseen in 802.11e BSSs) is straightforward.

Let j identify the last cycle. We consider that for an incoming WS, x, roaming from another BSS, the AP can acquire through signaling exchange between the APs the same

statistics that it has been collecting for the active WSs, plus the downlink throughput that x would like to receive, hereinafter denoted by $\overline{\nu}_{AP}^{(x)} + \overline{\eta}_{AP}^{(x)}$. If, instead, such information is unavailable, in order to monitor the bandwidth availability, the AP makes the following assumptions: (i) both uplink and downlink data frames from/to x have a payload size equal to the average value $\overline{P}(j)$; (ii) x transmits at the average data rate $\overline{R}(j)$; (iii) x's desired throughput is equal to $\overline{\nu}_x(j) = S(j)$ and $\overline{\eta}_x(j) = 0$ in uplink, and $\overline{\nu}_{AP}^{(x)} = S(j)$, $\overline{\eta}_{AP}^{(x)} = 0$ in downlink. Then, the AP updates its desired throughput as $\overline{\nu}_{AP}(j) = \overline{\nu}_{AP}(j) + \overline{\nu}_{AP}^{(x)}$ and $\overline{\eta}_{AP}(j) = \overline{\eta}_{AP}(j) + \overline{\eta}_{AP}^{(x)}$; also, it updates the set $\mathcal{N}(j)$ by adding x.

Considering that the 802.11 access scheme provides per-packet fairness, it is clear that any node $k \in \mathcal{N}(j)$, such that $\overline{\eta}_k(j) + \overline{\nu}_k(j) \leq S_n(j)$, will be able to transmit all its uplink traffic, both elastic and inelastic, while the others will reach $S_n(j)$ and then will share the remaining bandwidth, if any. Thus, in order to evaluate the throughput that x would achieve and its impact on the performance of the other nodes, we have to estimate the throughput that each active node can obtain with respect to the value it has experienced. To do so, we adopt the procedure reported in Algorithm 1.

According to the proposed algorithm, we first compute the remaining bandwidth $B(j)$ as the difference between the available bandwidth, set equal to the saturation throughput $S(j)$, and the sum of the shares of the nodes (line 3). Each node share is computed as the minimum between $S_n(j)$ and its total (elastic and inelastic) throughput, as measured by the AP in cycle j. Then, lines 4–5 in Algorithm 1 report the amount of inelastic and elastic node throughput that can be accommodated within the $S_n(j)$ share.

We identify the set of nodes $\mathcal{N}_o$ whose total (elastic and inelastic) throughput exceeds $S_n(j)$ (line 7). Considering one of these nodes at a time, we assume that it will get a fraction of the remaining bandwidth so as to transmit one additional packet of average size. While doing this, the node will give priority to inelastic traffic. This occurs while (i) $B(j) > 0$ and (ii) there is at least one node for which the throughput experienced in cycle j has not been reached yet (lines 8–22). As $S_n(j)$ has been computed considering the average node behavior, we weigh the bandwidth consumed by the generic node k to transmit a packet by $\frac{\overline{R}(j)}{\overline{R}_k(j)}$, thus accounting for the actual data rate used by the node (lines 11 and 15). Also, we consider the worst case in which nodes with the lowest data rate seize the channel first. Indeed, the lower the data rate, the larger the consumed bandwidth (line 7).

At the end of this procedure, we obtain the estimated throughput that the nodes, including the AP and the incoming WS, can achieve for elastic and inelastic traffic ($\hat{\nu}_k(j)$ and $\hat{\eta}_k(j)$, $k \in \mathcal{N}(j)$), as well as the bandwidth ($B(j)$) still available (if any) for inelastic traffic.

Algorithm 1 Bandwidth monitoring

$B(j) := S(j)$
for $k \in \mathcal{N}(j)$
  $B(j) := B(j) - \min\{\overline{\nu}_k(j) + \overline{\eta}_k(j), S_n(j)\}$
  $\hat{\nu}_k(j) := \min\{\overline{\nu}_k(j), S_n(j)\}$
  $\hat{\eta}_k(j) := \min\{\overline{\eta}_k(j), S_n(j) - \hat{\nu}_k(j)\}$
end
$\mathcal{N}_o \leftarrow Sort\Big(k \in \mathcal{N}(j)\, s.t.\, \overline{\nu}_k(j) + \overline{\eta}_k(j) > S_n(j), \overline{R}_k(j)\Big)$
while $B(j) > 0 \ \wedge \ \mathcal{N}_o \neq \emptyset$
  for $k \in \mathcal{N}_o \ \wedge \ B(j) > 0$
    if $\hat{\nu}_k(j) < \overline{\nu}_k(j)$
      $\delta := \min\left\{\frac{\overline{P}_{k,i}(j)\overline{R}(j)}{C(j)\overline{R}_k(j)}, B(j)\right\}$
      $\hat{\nu}_k(j) := \hat{\nu}_k(j) + \delta$
      $B(j) := B(j) - \delta$
    elsif $\hat{\eta}_k(j) < \overline{\eta}_k(j)$
      $\delta := \min\left\{\frac{\overline{P}_{k,e}(j)\overline{R}(j)}{C(j)\overline{R}_k(j)}, B(j)\right\}$
      $\hat{\eta}_k(j) := \hat{\eta}_k(j) + \delta$
      $B(j) := B(j) - \delta$
    else $\mathcal{N}_o \leftarrow \mathcal{N}_o \setminus k$ fi
  end
end

Computation of the b-metric

We now gauge the impact on the entire BSS resulting from the association of the new WS x.

In order to do so, on the one hand, we evaluate the degradation of inelastic traffic performance perceived by the nodes already associated with the AP. We neglect elastic traffic losses, as they can be recovered by TCP. We compute the estimated total throughput loss as:

$$\begin{aligned} L(j) \ &= \ \max\left\{\left[\overline{\nu}_{AP}(j) - \overline{\nu}_{AP}^{(x)}\right] - \hat{\nu}_{AP}(j), 0\right\} \\ &\quad + \sum_{k \in \mathcal{N}(j)\setminus\{x, AP\}} \left[\overline{\nu}_k(j) - \hat{\nu}_k(j)\right] \end{aligned} \tag{2.7}$$

where $\left[\overline{\nu}_{AP}(j) - \overline{\nu}_{AP}^{(x)}\right]$ is the throughput of the AP measured during cycle j, i.e., without considering the downlink traffic request of x. On the other hand, the gain $G(j)$ brought by WS x to the aggregate BSS throughput is given by

$$G(j) = \hat{\nu}_x(j) + \max\left\{\hat{\nu}_{AP}(j) - \left[\overline{\nu}_{AP}(j) - \overline{\nu}_{AP}^{(x)}\right], 0\right\} \tag{2.8}$$

where the first term on the right hand side is the (uplink) estimated throughput of x, while the second term is the estimated increase in the throughput of the AP due to the traffic it will deliver to x.

Then, to evaluate if the admission of x is beneficial in terms of aggregate throughput to the BSS, we define the b-metric as:

$$\mathrm{b}(j) = G(j) - L(j) + B(j) \tag{2.9}$$

where $B(j) > 0$ only if $L(j) = 0$.

Note that such a metric accounts for the beneficial contribution to the aggregate throughput due to x, the possible loss experienced by the other nodes, as well as the bandwidth still available after the association of x. It follows that the b-metric clearly indicates whether the association of x will increase the total BSS throughput ($\mathrm{b}(j) > 0$) or not ($\mathrm{b}(j) \leq 0$); it can therefore be used as a parameter for admission control as well as for load balancing among neighboring APs. Finally, we highlight that, in view of designing an admission control scheme, the satisfaction of the throughput requirements of x, if known, could be taken into account. To this end, we could compute $\hat{\nu}_x(j)/\overline{\nu}_x(j)$ and $\hat{\nu}_{AP}(j)/\overline{\nu}_{AP}(j)$ for the uplink and downlink traffic, respectively, and verify that such quantities are above a given threshold.

2.2.4 ABE performance evaluation

We implemented our algorithm as well as the data rate adaptation scheme AARF in the Omnet++ v4.1 simulator [64]. starting from the IEEE 802.11g module included in the INETMANET extension [65]. Our scenario consists of an IEEE 802.11g BSS, where all nodes can initially transmit at 54 Mbps. The channel representation is a refinement of the ITU indoor model, obtained using the experimental measurements in [66]. As for the algorithm parameters, we set $\alpha = 0.4$ and $T_{max} = 0.1$ s. For all experiments we run 5 different simulations, changing the seed set of the random number generators, and we present the averaged results. For sake of simplicity, we first evaluate our algorithm using only UDP flows.

We start by considering the same settings used to highlight the inaccuracy of the idle time and of the throughput-based metrics in Fig. 2.10, i.e., 6 WSs transmitting packets with payload of 500 or 1500 bytes. The comparison between the plots in Figs. 2.12 and 2.10 highlights that, unlike the other metrics, our b-metric correctly reflects the behavior of the BSS aggregate throughput at the MAC layer and detects the saturation condition in all cases, approaching zero when the aggregate throughput reaches its maximum possible value.

We now consider six WSs, which are placed on a circle whose center is at the AP and with radius equal to 20 m. Each WS generates a CBR traffic flow at 5.5 Mbps. The six WSs become active at two-second intervals between 0 s and 10 s, and are turned

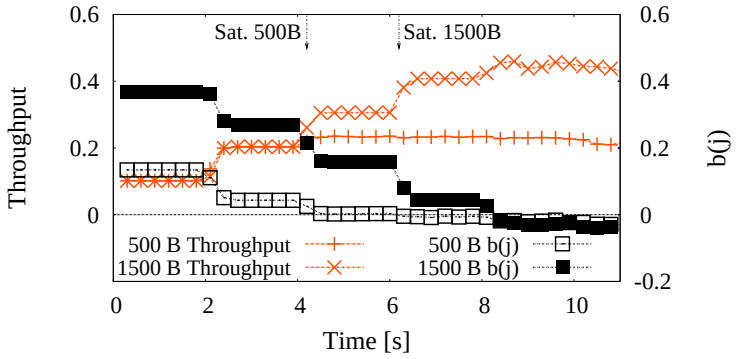


Figure 2.12. Time evolution of the aggregate throughput and of the b-metric, when 6 WSs transmit packets with payload of 500 B and 1500 B.

off in reverse order from 12 s onwards. The temporal evolution of the BSS aggregate throughput and of the b-metric (β) are presented in Fig. 2.13. The results clearly show that the b-metric (β) closely reflects the throughput behavior, indicating the bandwidth still available in the BSS and taking on negative values as the BSS reaches saturation. For instance, from 6 s to 8 s β reports a value of available bandwidth of 2 Mbps, but, as WS_5 becomes active at 8 s, the traffic load increases by 5.5 Mbps. Consequently, the BSS reaches saturation and β falls below zero. Similarly, when WS_6 becomes active within the BSS at 10 s, the network congestion, hence the collision probability, increases, leading to an evident performance degradation. Then, as the WSs start becoming inactive, the b-metric (β) promptly increases reflecting the new bandwidth availability.

Such results are confirmed by Fig. 2.14, which shows the temporal evolution of the data rate used by each WS in the same scenario as the one described above. In the scenario under study, the channel error probability is extremely low (due to the short AP-WS distance), hence the WS data rate is mainly determined by collision events, according to the AARF mechanism. By looking at the plot, we can see that, when the BSS reaches the saturation, i.e., the b-metric (β) becomes negative, the collision probability grows and, as a consequence, the data rate used by each of the WSs decreases.

Next, we consider a BSS with five WSs, $WS_1, \ldots, WS_5$, which become active at different time instants (namely, 0 s, 2 s, 4 s, 6 s, 8 s) and generate CBR traffic at different rates (namely, 10 Mbps, 8 Mbps, 4 Mbps, 2 Mbps, and 1 Mbps). Fig. 2.15 depicts the temporal evolution of the BSS aggregate throughput and of the β-metric. Again, we note that the β-metric accurately represents the difference between the saturation throughput and the current aggregate throughput. In particular, even in this scenario where the last stations becoming active generate traffic at a quite low rate and they erode a relatively small amount of bandwidth, the behavior of the β-metric correctly reflects such small changes (e.g., transitions in throughput and in β between 4 s and 8 s).

Finally, we evaluate the effects of mobility on the performance. We consider four

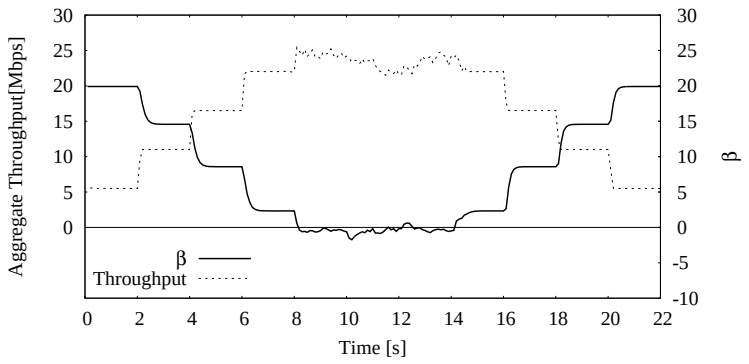


Figure 2.13. Temporal evolution of the aggregate throughput and of the b-metric (β), with six WSs becoming active at different time instants.

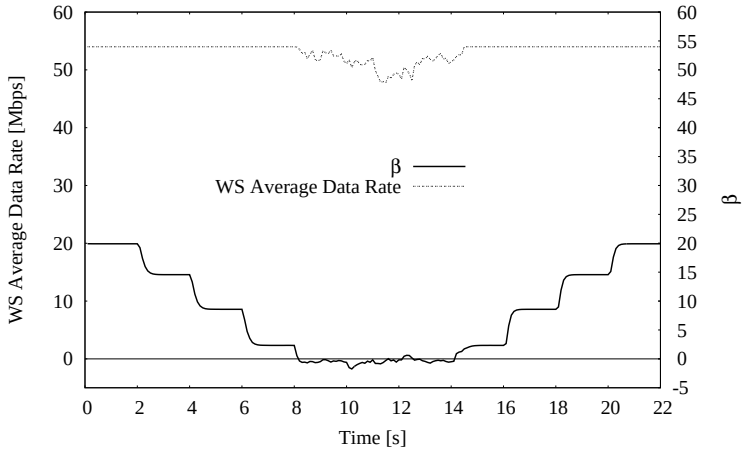


Figure 2.14. Temporal evolution of the average data rate used by the six WSs.

WSs, WS_1,..., WS_4, three of which are placed 20 m away from the AP, while WS_4 is located at a distance of 20 m from the AP at 0 s and moves away from it at a constant speed of 8 m/s. Clearly, as WS_4 moves away from the AP, its data rate decreases due to worsening channel propagation conditions, reaching a data rate as low as 18 Mbps at time instant 10 s. Also, the WSs are active for different time intervals, specifically: WS_1 in [0,11] s, WS_2 in [2,4] s, WS_3 in [1,6] s, and WS_4 in [0,10] s, and when active, they generate CBR traffic at 7 Mbps. Fig. 2.16 presents the temporal evolution of the BSS aggregate throughput, the b-metric (β), and the data rate of WS_4. From the plot, it is evident that the b-metric (β) accurately represents the bandwidth that becomes available when WS_2 and WS_3 stop generating traffic at time 4 s and 6 s, respectively. Also, observe that the farther WS_4 moves away from the AP (i.e., the lower its data rate), the smaller the available bandwidth, hence the value of the b-metric (β).

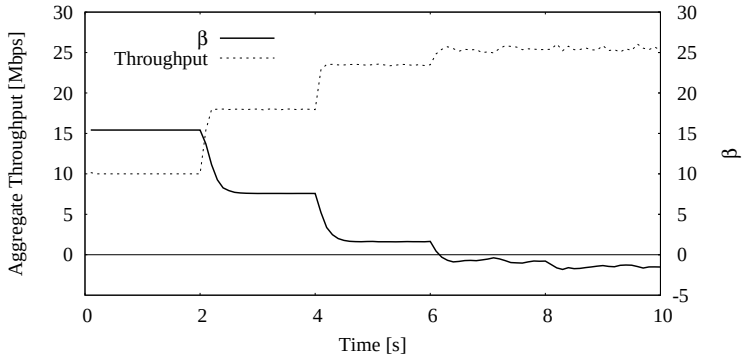


Figure 2.15. Temporal evolution of the aggregate throughput and of the β-metric in presence of five WSs with different traffic rates and activation times.

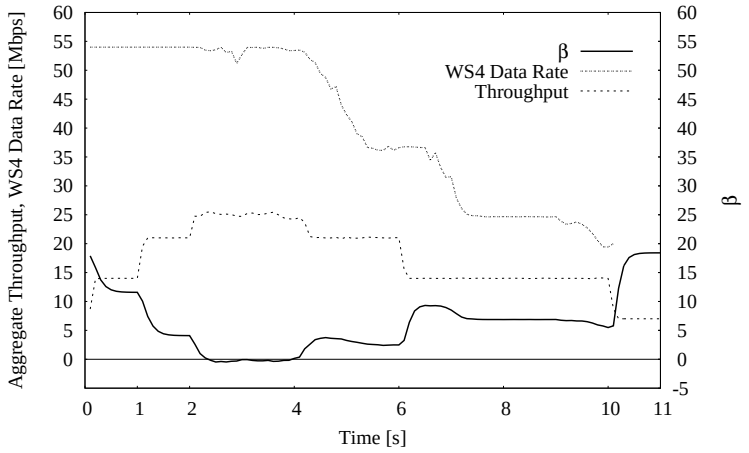


Figure 2.16. Temporal evolution of the BSS aggregate throughput, the b-metric (β), and the data rate of WS_4, with three static and one mobile WSs as well as different on-off traffic patterns.

We conclude that our metric accurately reflects the available bandwidth in the BSS, as load and station data rates vary.

Next, we evaluate the effectiveness of our algorithm in predicting the throughput that a WS starting one or more traffic flows can achieve, when both elastic (TCP) and inelastic (UDP) traffic are present. All subsequent results are averaged over 5 different simulation instances and use TCP SACK. UDP traffic is modeled as CBR traffic with an offered load of 8 Mbps. We fix the payload size to 1500 bytes and, for clarity of presentation, we limit our study to 3 WSs, assuming that the WS joining the BSS is not in roaming mode (thus, exact information on its traffic requirements is not available). Also, the depicted throughput is computed at the MAC layer and, for TCP traffic, it includes both data and

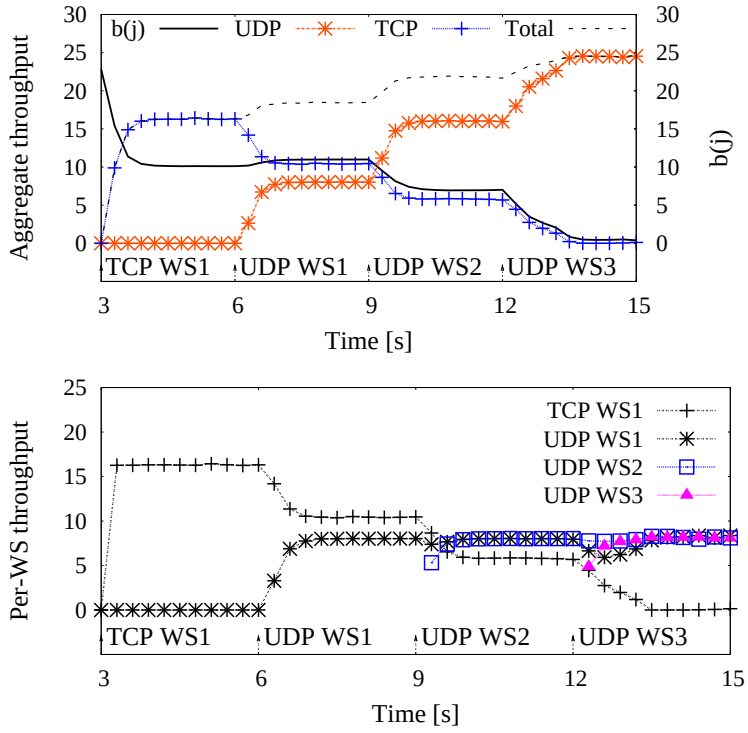


Figure 2.17. WS 1 originates one TCP and one UDP flow, while WS 2 and 3 originate one UDP stream each.

TCP ACK packets.

We first focus on the following scenario: WS 1 starts a TCP connection at $t = 3$ s and, subsequently, a UDP flow at $t = 6$ s. The other two stations, WS 2 and WS 3, start a UDP flow at $t = 9$ s and $t = 12$ s, respectively. Fig. 2.17 shows the temporal evolution of the BSS aggregate throughput and of our b-metric, as well as the throughput achieved by each WS. In spite of the saturation condition caused by the TCP session started by WS 1 at $t = 3$ s, the b-metric correctly reflects that some bandwidth is available for the newly originated flow. As the UDP stream starts at 6 s, TCP adjusts its throughput and lets UDP take the desired bandwidth. Interestingly, we note that the b-metric is not significantly affected by this new condition. This is due to two reasons: (i) the UDP stream is originated by the same WS that started the TCP flow and (ii) the UDP demand is less than the estimated remaining bandwidth. The slight change that we observe in the b-metric results from the smaller number of TCP ACKs within the cycle, hence from a greater observed average payload size. Conversely, when the UDP flow of WS 2 becomes active at $t = 9$ s, the b-metric drops to 8 Mbps. The available bandwidth, though, is enough to accommodate the flow by WS 3, which starts at $t = 12$ s and brings the system to saturation, hence the b-metric drops to 0. Also, as expected, the TCP flow almost dies

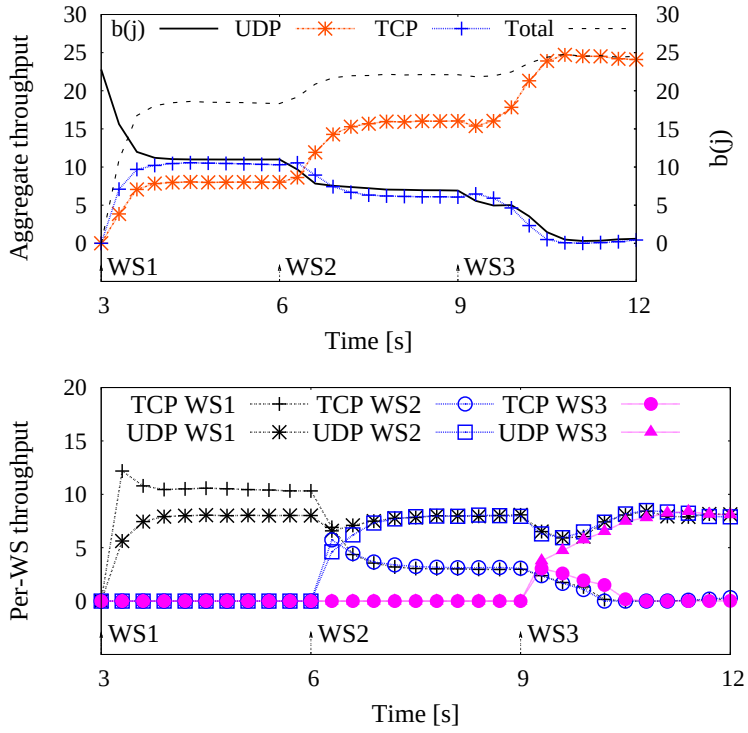


Figure 2.18. Three WSs originate one TCP and one UDP flow each.

out after $t = 12$ s. It follows that, if the b-metric were used for admission control, the AP would admit both WS 2 and WS 3. By looking at Fig. 2.17, it is evident that this would be a good choice, as all WSs are able to meet their UDP demand.

We then consider that all WSs originate one UDP and one TCP flow, and that WS 1, 2 and 3 become active at $t = 3$, 6 and 9 s, respectively. Due to the competition between elastic and inelastic traffic within the same WS, all TCP flows are expected to die out as the UDP streams accommodate their demand. Fig. 2.18 confirms such a guess showing that the time evolution of the aggregate TCP throughput matches that of the bandwidth available for inelastic traffic; again, the b-metric reflects such a behavior very closely.

At last, we consider the same settings but for the TCP flows direction: all WSs are now destinations of TCP traffic. Fig. 2.19 shows that in this case the UDP streams do not experience losses only for $t \in [3,6]$ s, i.e., when only WS 1 and the AP are active. In this time interval, the b-metric correctly detects enough bandwidth to accommodate an 8 Mbps-flow. Then, from Fig. 2.19, we note that, after $t = 9$ s, both WS 1 and 2 suffer a loss with respect to their demand, due to the new UDP flow started by WS 3. Consistently, the b-metric indicates that no bandwidth was available for inelastic traffic. We point out that the AP throughput share, which is used for TCP traffic, erodes some of the resources available for the WSs, due to the per-packet fairness provided by the DCF.

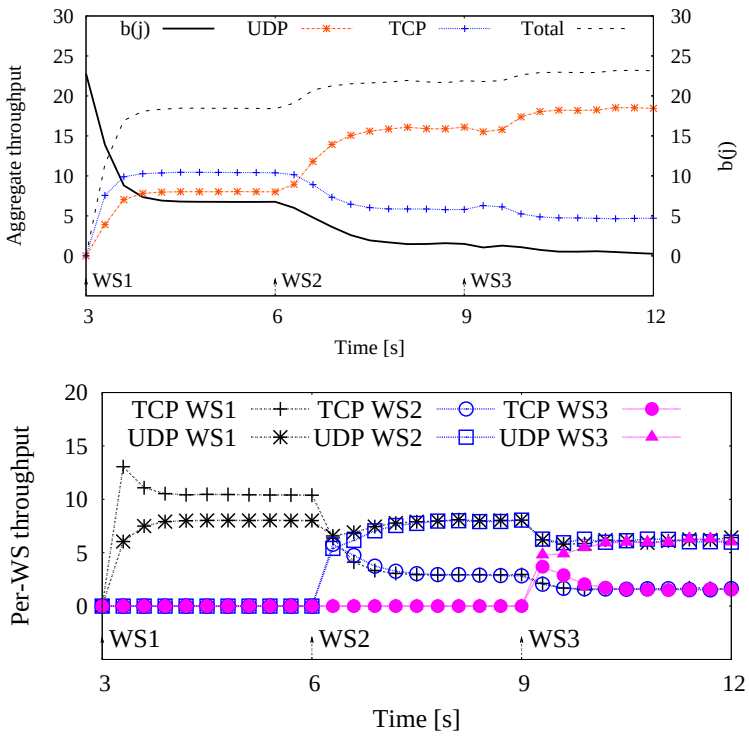


Figure 2.19. WS 1, 2 and 3 originate one UDP flow and are destinations of one TCP flow each.

2.2.5 Final remarks on ABE

We designed an algorithm and a metric for bandwidth monitoring in a multirate IEEE 802.11 BSS with AP, in presence of elastic and inelastic traffic. The metric we propose can be autonomously computed by the AP, without requiring any cooperation from the wireless stations. Also, the results we derived show that our solution can accurately estimate the bandwidth available for inelastic traffic, as well as the (both elastic and inelastic) throughput performance of all BSS nodes.

We remark that the definition and the computation of our metric is the basic constituent for the design of a distributed load balancing algorithm for 802.11 networks that maximizes the total system throughput while minimizing the number of active APs.

Chapter 3

Enhancing Connectivity in a Federated Network

The implementation of the energy-efficient cooperation scheme for federated GWs consists of three main steps as detailed in [51].

The first one is the estimation of the current load of the wireless channel and of its saturation throughput, through passive traffic measurements. By comparing the current load to the saturation throughput S, a GW classifies its status as either *Light*, *Regular*, or *Heavy*. In the Light status, traffic likely comes from background communications to/from the WSs, prompting the GW to try and relocate them, switch itself off and save energy. The Heavy status, instead, characterizes an overloaded BSS, where some WSs should associate to other BSSs to benefit from load balancing. In the *Heavy* status a GW tries to relocate one WS at a time, starting from the one having the lowest bit rate, until the *Regular* status is reached. A GW in *Regular* status is too busy to switch itself off while it does not need to be relieved of some of its WSs. It might, however, accommodate relocated WSs within its BSS.

Passive traffic measurements account for the second step too, in which a GW trying to relocate one or more of its WS compiles a traffic profile of each WS, detailing the throughput of its active (downlink/uplink) traffic flows.

The third and final step amounts to an inter-GW communication in which a *requesting* GW, aiming at relocating one or more of its WSs, sends a handover request, along with the traffic profiles of relocatable WSs, to *candidate* GWs. A computation of the projected load (current estimated load and expected load from the incoming WS profile), and its comparison against the estimated saturation throughput S, allow a candidate GW to assess its suitability to provide help to others (i.e., if the additional WS does not plunge the GW in Heavy status). Responses returned by candidate GWs let the requesting GW identify a feasible relocation strategy. Among the feasible solutions that allow a GW to relocate its WSs, the allocation maximizing the average data rate of the WSs is selected. For a *Heavy* initiated request, if no viable relocation is found the requesting GW needs to wake

up a neighboring "off" GW and repeat the requesting procedure. Further details about the protocol can be found in [51].

In the next section, we outline how we address the above issues, following a *distributed, Gateway-initiated* paradigm. Our solution does not require changes to the standard medium access control (MAC) protocol, nor to the WSs, which may be even unaware of its adoption. The details of the network architecture we propose are introduced in Section 3.2. In Section 3.3, we describe the algorithms that let Gateways assess their current load, hence whether they need help from federated Gateways or not, as well as their suitability to provide help to others. The inter-Gateway communication protocol is presented in Section 3.4 and evaluated in Section 3.5, along with the aforementioned algorithms, in a realistic residential scenario. Finally, Section 3.8 outlines the conclusions.

3.1 Outline and Motivating Examples

With respect to the different management techniques introduced above, we choose to pursue a *distributed, Gateway-initiated* system that does not require WSs to be modified, or even to be aware of its presence. As will be argued in the following, such system requires that Gateways (i) have self-load assessment capabilities; (ii) run inter-Gateway communication protocols for WS relocation; (iii) rely on robust authentication and authorization procedures provided by the federated network and (iv) employ reliable Wake on Wireless LAN procedures [52].

Firstly, self-load assessment through traffic measurement allows Gateways to classify their status as either *Light*, *Regular*, or *Heavy*. The assessment depends on the available airtime on the medium and on the number of associated WSs within the BSS that each Gateway controls. In particular, we compute the former by leveraging theoretical results on the BSS saturation throughput [39, 40] thus accounting for the collision probability as well as for the different data rates and payload sizes used within the BSS. Status evaluation guides Gateway policy in either seeking relocation of their WSs or in accepting WSs handed over by nearby Gateways. In the Light status traffic likely comes from background communications to/from the WSs, prompting the Gateway to try and relocate them, switch itself off and save energy. The Heavy status, instead, characterizes an overloaded BSS, where some WSs should associate to other BSSs to benefit from load balancing. A Gateway in Regular status is considered too busy to switch itself off while it does not need to be relieved of some of its WSs. It might however accommodate relocated WSs within its BSS. As a side remark, the Gateway status does not affect the spontaneous association of a WS within one's household, thus it does not interfere with normal operations. Also, for simplicity, we will present the self-load assessment procedure considering that the wireless access network, rather than the backhaul, is the traffic bottleneck. The procedure, however, can be extended to a more general setting as well.

Secondly, the relocation of WSs within the federated network involves communication

and coordination among Gateways. A Gateway receiving a help request needs to evaluate its own suitability to give help, i.e., the impact of accommodating a new WS within the BSS it controls. Under evaluation are: (i) the load parameters most recently exhibited by the WS within its current BSS; (ii) the transmission rate at which the WS would likely operate in the new BSS. While the load can be estimated by the requesting Gateway, and its parameters included in the relocation request, the transmission rate cannot be easily predicted and will require remote sampling, as described later. The Gateway requesting help then chooses to offload the WS to the most suitable candidate Gateway.

Lastly, while our performance evaluation mainly focuses on the effectiveness of the load assessment and of WS offloading, we will introduce architectural solutions that address both Gateway authentication and its wake-on process.

An example of the procedures just outlined is presented in Fig. 3.1, where a 3-house neighborhood is displayed. The first example (top part of Fig. 3.1) shows a case of energy saving through the switching off of some Gateways. With reference to the first row of the figure, in household 1, we assume that one WS is "whispering", i.e., occasionally sending low background traffic (mainly, status update for some applications and other signalling). The other WS completes the download of a software update and starts whispering as well. As a result, the Gateway in household 1 (GW 1, for short) becomes underloaded and goes in Light status (shaded in light green). Next door, one of the WSs in household 2 is engaged in peer-to-peer downloading, while the other WS is browsing Wikipedia. The WS running the peer-to-peer application shuts down, hence also the status of GW 2 shifts to Light. Finally, in household 3, we assume one WS listening to music streamed over the Internet, while the other two are browsing. Their Gateway is in Regular status (shaded in blue). Upon switching to Light, GW 1 and GW 2 will start vying for the chance to offload their WSs and turn themselves off to save energy. Through a protocol exchange over the backhaul, we assume that GW 3 rejects the help request by either neighbors since it establishes that accepting any of their WSs would force it into Heavy status. GW 2, instead, "wins" the competition thanks to its lower traffic load compared to GW 1: thus it hands its only active WS to GW 1 and goes "off". Upon accepting the next-door WS, GW 1 switches to Regular status and the equilibrium shown in the second row of Fig. 3.1 is reached.

Our second example (bottom part of Fig. 3.1) shows instead a case of congestion relief through load balancing. At the beginning, GW 1 is "off" following its offloading of two WSs to GW 2. The latter, however, suddenly finds itself in Heavy status (shaded in red) when two local WSs become actively engaged, respectively, in a video conference over the Internet, and in a data backup to a cloud service. GW 3 is in Regular status, as in the previous example. In order to decrease its load, GW 2 initiates an offload request toward its neighbors. As before, GW 3 rejects it. Left with no alternatives, GW 2 uses a wake-on WLAN technique to awaken GW 1, followed by an offload request. GW 1 accepts the request and the equilibrium depicted in the last row of Fig. 3.1 (i.e., all Gateways in Regular status) is reached.

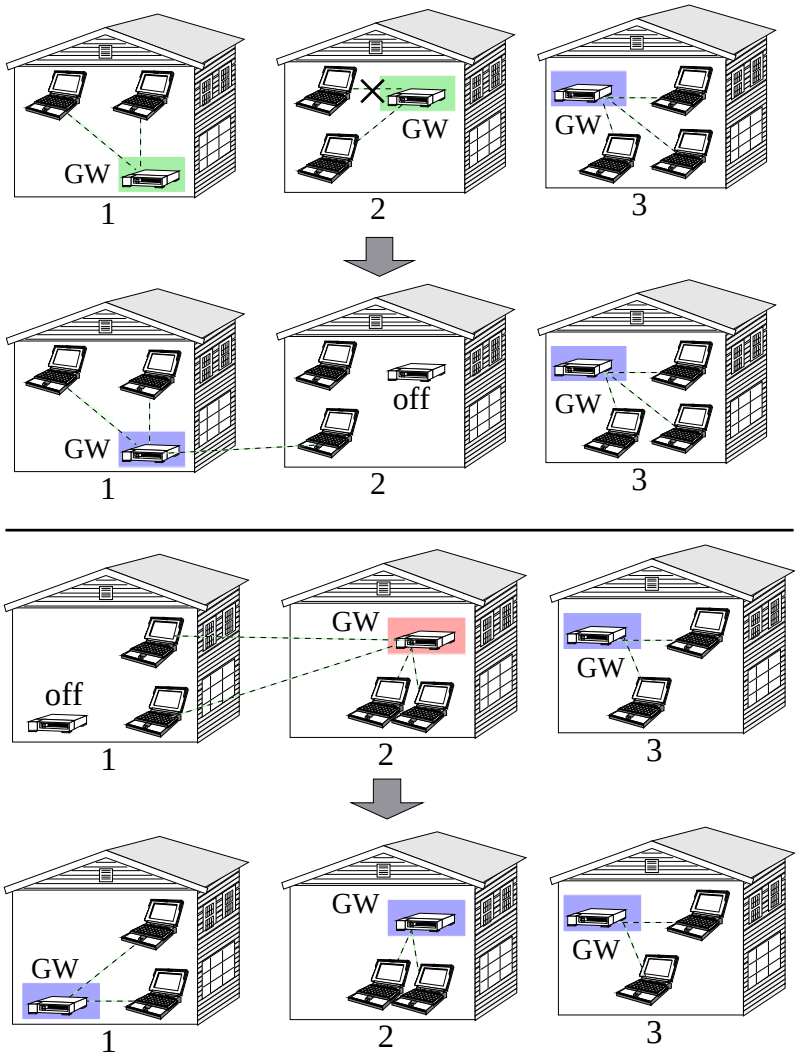


Figure 3.1. Energy saving (top) and load balancing (bottom) by WS offloading.

3.2 Network Architecture

In our study, we consider a set of residential units (e.g., houses or apartments), each of them equipped with a Gateway, with external and internal connectivity functions. We apply to this scenario the concept of "Federation", i.e., a logical overlay relationship among trusted home gateways for the purpose of content exchange and resource sharing [53].

When "on", Federated Gateways can communicate and coordinate with each other using an out-of-band channel, which runs through their backhaul Internet connection. Each "on" Gateway offers local wireless access through the 802.11 a/b/g/n technology over independently-managed (but possibly coordinated) frequency channels. The WSs

that operate within a BSS controlled by a Gateway can be sources or destinations of elastic or inelastic traffic, i.e, flows that use either TCP or UDP at the transport layer. At the MAC layer, the Gateway and the WSs transmit frames at a data rate that may vary according to the experienced channel propagation conditions.

When Gateways are "off", they no longer have wired, nor 802.11 radio, connectivity and only run a low-cost, low-power radio interface, e.g., a IEEE 802.15.4 card, that can be used as wake-on WLAN interface [52]. We define as Radio Federated Network (RFN) within a federation, a subset of Gateways that can reach each other, either directly or via multihop communication, through their low-power interface. The discovery procedure of RFN neighbors, which is out of the scope of this thesis, can occur through the periodic issuing/listening of hello messages on the low-power interface.

Inter-Gateway, out-of-band communication, Gateway wake-up procedures as well as WS relocation within the RFN require both authentication with a centralized AAA server and the creation of a group key, called Federated Group Key (FGK). As required by current secure multicast applications, the FGK must be refreshed periodically. The new FGK must thus be distributed to all members of the RFN including "off" Gateways. To this end, "off" Gateways maintain a loose synchronization, previously acquired through an NTP server, using a standalone clock. Upon a scheduled key expiration, an "off" Gateway will switch on in order to update its FGK. Additionally, in order to allow WSs to seamlessly associate to a new Gateway, there is the need to implement a reauthentication procedure at the WS, reusing the current keying material for the handover. These reassociation requirements can be fulfilled by exploiting already existing protocols, such as HOKEY [54], and will not be discussed further in this thesis.

The handover at the MAC layer can be seamlessly implemented by using one virtual access point (VAP) for each WSs, as proposed in [55]. With VAPs, every WS receives a unique BSSID to connect to, i.e., every WS has its own AP that never changes in time. A GW hosts as many VAPs as the number of associated WSs. Removing a VAP from on GW to another one achieves, at the MAC layer, a seamless handover of the WS corresponding to the shifted VAP, without requiring re-association messages nor specialized software or hardware at the client. In order to achieve seamless handover for ongoing applications, the network must support mobile IP or a similar technology. Recent studies [56, 57] have shown the suitability of optimized protocols, namely fast MIPv6, hierarchical MIPv6, as well as network-based mobility management solutions like Proxy Mobile IPv6 (PMIPv6), to support also real-time applications. However, this result can be achieved when the handover is performed within the same IPv6 domain, i.e., for local mobility. It is worth noting that mobile IP can also deal with Network Address Translation (NAT), which is widely used in home GWs, by performing IP-in-UDP tunneling [58]. We assume that the federated network is managed by a single ISP which supports multicast and implements also PMIPV6 or similar technology in order to guarantee a seamless handover for applications. We note that applications capable of restarting their flow after a disconnection,

e.g., HTTP downloads, Dropbox, etc.., may not need Mobile IP. The design of a next-generation global mobility protocol capable of guaranteeing seamless handover across multiple ISPs with little or no disruption time is outside the scope of this paper.

3.3 BSS Load Assessment and Management

As mentioned, the main idea at the basis of the two algorithms is to leverage the theoretical results on the saturation throughput derived in [39, 40], for the computation of the BSS capacity. We therefore start by introducing our notation and the computation of some fundamental quantities. In the following, we only consider the case where the traffic bottleneck is represented by the wireless access network and there are no endogenous flows. The extension to a more general setting can be done at the price of additional, burdening notation.

3.3.1 Preliminaries

A Gateway accesses the "protocol type" field in the IP packets and collects statistics on elastic and inelastic traffic within its BSS. The Gateway carries out such measurements periodically over a time interval, hereinafter referred to as measurement period. The Gateway considers a node in the BSS (either itself or a WS) to be active if the node has successfully transmitted at least one data frame within the last measurement period. We denote by $\mathcal{N}$ the set of currently active nodes and by N its cardinality.

Similar to the mechanism we described in [59], every measurement period, for each active WS k the Gateway computes a *running average* of the uplink (UL) throughput for all elastic and inelastic flows, denoted by η_k^u and ν_k^u, respectively. Likewise, the Gateway computes a running average of its own downlink (DL) throughput for both the elastic and inelastic traffic it handles for each WS k, denoted by η_k^d and ν_k^d, respectively. In addition, for each frame successfully transmitted from/to the generic WS k, the Gateway observes the payload size and the used data rate, and it computes the corresponding running averages. As in [60], the Gateway computes the running average of the data rate, R, and of the payload size, P, over all data frames that it correctly sends or receives. We will refer to all the above measurements the Gateway performs for a WS as the WS *traffic profile*.

Another fundamental quantity for our assessment algorithms is the (aggregate) saturation throughput S, which we take as value of ***BSS capacity***, defined as in 2.3, and reported below with the new notation.

$$S = \frac{N\tau[1-\tau]^{N-1}P(1-p_e)}{E[T]}. \tag{3.1}$$

All the details about the computation of S have been reported in 2.1.2 for completeness, together with the procedure for estimating p_e.

Algorithm 2 Gateway status assessment

Compute the saturation throughput S and initialize the load L to 0
for every active WS k do
  Set the minimum elastic throughput expected in UL and DL
  $\underline{\eta}_k^u := \min\{\eta_k^u, \alpha S\}$; $\underline{\eta}_k^d := \min\{\eta_k^d, \alpha S\}$
  Add to the load the measured inelastic throughput and
  the minimum elastic throughput
  $L := L + \nu_k^u + \nu_k^d + \underline{\eta}_k^u + \underline{\eta}_k^d$
end
Assess status by comparing the normalized load to thresholds
if $\frac{L}{S} \leq T_L \ \wedge \ N < N_L \rightarrow Light$
  else if $\frac{L}{S} > T_H \rightarrow Heavy$
    else $Regular$

Algorithm 3 Assessing the Gateway suitability to provide help

Compute the estimated saturation throughput $S^\star$, including x
Set the minimum elastic throughput expected by x in UL and DL
$\underline{\eta}_x^u := \min\{\eta_x^u, \alpha S\}$; $\underline{\eta}_x^d := \min\{\eta_x^d, \alpha S\}$
Compute estimated load $L^\star$ by adding to current actual load L
the total throughput expected by x
$L^\star := L + \nu_x^u + \nu_x^d + \underline{\eta}_x^u + \underline{\eta}_x^d$
Compute room-metric $\rho = 1 - \frac{L^\star}{S^\star}$
if $\rho < 1 - T_H$ associating x would shift the status to Heavy
  then GW cannot provide help
  else GW can associate x

We stress that, although S represents the saturation throughput considering the node average behavior, it accounts for the different air time that the WSs take. Indeed, the average payload size P, data rate R, and $E[T]$ in (3.1) depend on the payload, data rate and access rate of each single WS.

3.3.2 Does the Gateway need help?

Every measurement period and through running averages, the Gateway computes the capacity of the BSS it controls using the expression of the saturation throughput in (3.1). Then, it computes the traffic load L within the BSS and compares it to the saturation throughput S, so as to gauge its own status.

As reported in Alg. 2, to assess the load L, the Gateway leverages the throughput measurements it has carried out and computes L as the sum of all contributions due to the

existing traffic. Specifically, the contribution due to inelastic flows is set to their measured throughput. For the elastic flows of each WS, instead, the Gateway mitigates the effect of their greedy behavior by associating them a contribution that is at most equal to a fraction of the saturation throughput, namely, αS.

If the traffic load, normalized to the saturation throughput, is below a threshold T_L and the number of stations associated to the BSS is less than N_L, the Gateway is in Light status. In this case, the Gateway will ask for help so as to relocate its WSs and switch itself off. If instead the normalized load is above T_H, a Heavy status is detected and the Gateway will try to relocate one or more of its WSs so as to avoid overload conditions. Otherwise, a Regular status is assessed, in which case no help from the federated Gateways is required.

3.3.3 Who can help the Gateway?

Upon the reception of a help request asking for WS relocation, a federated Gateway needs to reliably evaluate the impact on its BSS of associating additional WSs, i.e., its suitability to give help. To do so, the Gateway computes the bandwidth available within its BSS, as if the WSs to be relocated were actually associated. We name such a quantity room-metric and use it as a suitability index: the greater the room-metric, the more suitable the Gateway to accommodate the WSs.

For simplicity, the room-metric computation is outlined below and in Alg. 3 in the case where a single WS has to be relocated; the extension to the case of more WSs is straightforward.

Let GW i be the Gateway that has to assess its suitability to provide help, and x the WS that another Gateway would like to relocate. As detailed later, through signaling exchange between Gateways, GW i can acquire the uplink and downlink throughput that x expects to receive for inelastic and elastic traffic, as well as the average payload of the frames x transmits and receives. GW i computes the saturation throughput $S^\star$ as if x had been already associated. More precisely, it adds x to the active nodes set and recomputes the average payload size and data rate in the BSS considering also the traffic profile of x, then it uses such values in (2.6). Next, in order to evaluate the impact that the association of x would have on the performance of the existing flows, the Gateway estimates what the load of the BSS would be if the throughput demand of all WSs were fulfilled. To this end, it adds to the current load the throughput that x expects for its elastic and inelastic traffic, in uplink and downlink. As in the procedure for the Gateway status assessment, the effect of the greedy behavior of the elastic flows, involving either the existing WSs or x, is mitigated by associating them at most a fraction of the saturation throughput.

The room-metric ρ is set to the estimated fraction of bandwidth that would be available in the BSS if x were associated. If the association of x drives the Gateway in Heavy status (i.e., the estimated normalized load exceeds T_H), then x is rejected; otherwise, the Gateway reputes that the WS can be relocated into its BSS.

3.4 Wireless Resource Sharing Protocol

We now introduce the protocol that lets federated Gateways share their radio resources. As mentioned, our objective is twofold: (i) to minimize the number of switched-on Gateways, and (ii) to avoid overloading traffic conditions for the "on" Gateways. To achieve these goals, a Gateway periodically assesses its status through Alg. 2, and, if in Light or Heavy status, it carries out an offload procedure, as summarized in Fig. 3.2. The procedure aims at relocating one or more WSs at other Gateways. The federated Gateways estimate which WSs they could associate, based on the value of their room-metric computed through Alg. 3, and reply accordingly. Upon finding a valid WS relocation, the Gateway that started the procedure can turn itself off if it was in Light status, while it experiences a load relief if it was in Heavy status.

We remark that the presence of a central controller is not required, and the implementation of the proposed protocol implies changes only in the Gateways, not in the WSs. Also, the Gateway status does not affect the spontaneous association of new WSs within one's household, thus it does not interfere with normal operations.

The offload procedure for a Gateway in Light or Heavy status is detailed below.

1) Consider a Gateway, GW l, that finds itself in **Light status**. Then, GW l starts an offload procedure by multicasting an OFFLOAD_REQUEST message to the federated Gateways. This message, as all of those exchanged between Gateways, is transmitted through the out-of-band channel and it includes the following information: (i) the status of the requesting Gateway, along with its room-metric (computed as $1 - L/S$), (ii) the frequency channel used in the BSS, and (iii) for each WS in the BSS, a hash of the association ID (AID), the MAC address and the measured traffic profile. After the OFFLOAD_REQUEST is issued, GW l sets a timer to the timeout value τ_r.

An OFFLOAD_REQUEST is processed only by the Gateways in the RFN that are currently "on" and not in Heavy status. Since the request comes from a Gateway in Light status, the federated Gateways first check if their room-metric is greater than the value advertized by GW l. If so, they discard the request since they are less loaded than GW l. Otherwise, they need to assess which of the WSs to be relocated are in their radio range and which data rate they could use to communicate with them. To do so, a Gateway can leverage information previously collected about these WSs, if they have been associated to the Gateway in the past. Alternatively, the Gateway sends a CTS so that all of its WSs will be frozen for a time τ_p while it can tune its 802.11 interface to the channel used by GW l. Then, we let GW l probe each WS in its BSS with an RTS message. As the probed WS will reply with a CTS, the Gateway monitoring the frequency channel can estimate the signal-to-noise ratio (SNR), hence the data rate to communicate with the WS. GW l will set the RTS duration field so that the corresponding field in the CTS will be the hash function of the WS's AID[1]. Such a procedure allows a Gateway, not in radio proximity

[1]The RTS duration field is set to the sum of the SIFS time, CTS transmission time and the hash of the

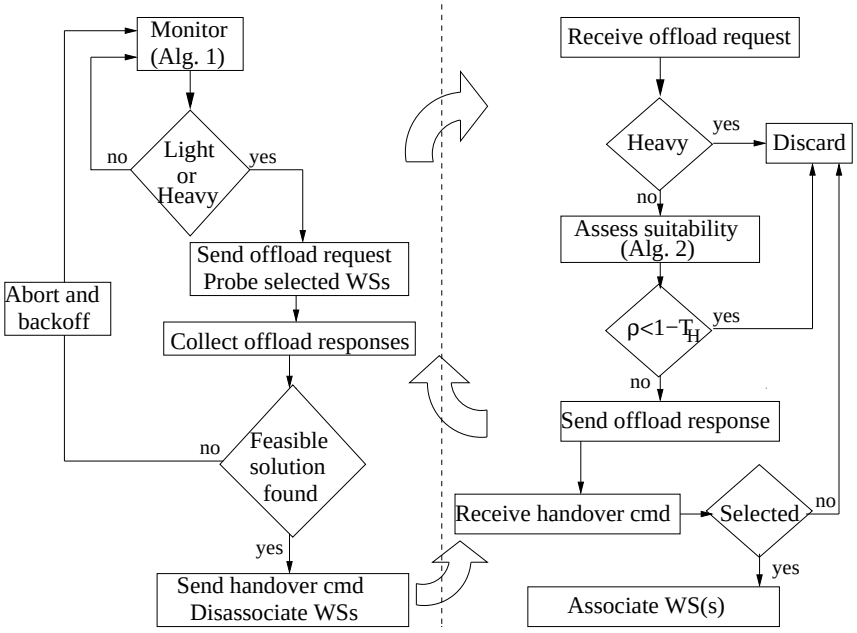


Figure 3.2. Flow chart of the offload procedure: Gateway asking for help (left) and Gateway receiving the help request (right).

of GW l (i.e., unable to hear the RTS), to identify the WS sending the CTS. Clearly, it introduces some overhead, but, since GW l is underloaded, we expect the number of WSs in its BSS to be small.

Each federated Gateway then considers the WSs from which it has heard a CTS and verifies which of them (if any) could be associated to its BSS. To do so, the Gateway evaluates through Alg. 3 the room-metric for the possible combinations of candidate WSs. Finally, it unicasts an OFFLOAD_RESPONSE message to GW l, including the combinations with a positive outcome (i.e., such that $\rho \geq 1 - T_H$), as well as the corresponding value of the room-metric and the data rates that could be used to communicate with the candidate WSs.

Upon the expiration of the timeout τ_r, GW l evaluates all received replies by scanning them and by assigning each WSs to the GW that will provide the maximum data rate. A random selection is used in order to solve possible ties. The rational is that WSs should be handed over to the Gateways that will be able to communicate with them at the highest data rate, so as to guarantee an efficient traffic transfer. This allocation procedure is scalable as it requires only one pass among the list of replies, which are limited by the size of the RFN [2]. Note that an allocation is valid only if *all* the WSs are assigned to another GW.

WS's AID. The value of the hash should be upper bounded by $2 \cdot$ SIFS plus the ACK duration so that probe CTS cannot be mistaken with regular CTS.

[2] 50% of the residential GWs have less than 10 neighbors [61].

If a valid allocation is found, GW l multicasts a HANDOVER_COMMAND, including the MAC address of the WSs assigned to the Gateways that offered their help. The message also contains a flag notifying that GW l is switching off. Otherwise, it multicasts to all Gateways an ABORT message. We remark that, by receiving the HANDOVER_COMMAND, all "on" Gateways in the RFN can keep track of those that switch themselves off.

Upon the reception of a HANDOVER_COMMAND, each selected (resp. non-selected) Gateway can include the assigned WS(s) in its authorized (resp. non-authorized) stations list, so that, when GW l switches itself off, each WS will necessarily associate with the right Gateway. The reassociation of a WS to a target Gateway can also be performed through the 802.11v BSS transition management procedure [62].

2) When a Gateway, GW h, finds itself in **Heavy status**, it starts an offload procedure similar to the one described above. A few differences, however, exist.

Firstly, GW h tries to hand over only one WS at a time, till its status changes into Regular. Specifically, it lists the WSs in decreasing order according to their offered load weighted by the inverse of their data rate, and it attempts to relocate the top WS first. Thus, the handover of each WS results into a different offload procedure.

Secondly, upon receiving an OFFLOAD_REQUEST from GW h, an "on" Gateway not in Heavy status will always reply. Again, the whole inter-Gateway communication takes place on the out-of band channel. A successful relocation is confirmed by a HANDOVER_COMMAND. If no viable relocation is found, GW h needs to wake up a neighboring "off" Gateway and ask for help. The Gateway to be turned on, GW w, can be selected based on the WS relocation history, if available, or it can be randomly chosen among the neighboring Gateways. However, the exchange of landline signalling could be unfeasible due to the "off" state of the Gateway. Thus, to accomplish this task in an energy-efficient manner, GW h unicasts through its low-power interface a wake-up sequence $\mathbb{W}$, to the selected "off" Gateway. The low-cost interface of GW w can detect the sequence and turn the rest of the Gateway circuitry on. However, in order to avoid attacks aiming at unnecessarily waking up "off" Gateways, $\mathbb{W}$ should be followed by an encrypted Message Authentication Code (MAC), $\mathbb{M}$, which can be decoded only using the FGK. Since all Gateways are synchronized, $\mathbb{M}$ can be calculated as $\mathbb{M} = \text{MAC}(\text{FGK}\|T\|\text{ID})$, where T is the current time expressed in seconds and ID is the unique identifier of GW w assigned within the federation. Thus, upon detecting $\mathbb{W}$, the low-power device at GW w will compute its own message authentication code, $\mathbb{W}'$, and it will turn the whole circuitry on only if $\mathbb{W}' = \mathbb{W}$.

At this point, GW h unicasts to the woken-up GW w an OFFLOAD_REQUEST and tries to relocate its WS to it, so as to decrease its load below the Heavy threshold. In the unlikely case where none of its WSs can be relocated, GW h will wake up another Gateway while GW w, having not associated any WS, will switch off again after a timeout. In this way, we let "off" Gateways turn themselves on if needed, while limiting the number of Gateways that wake up.

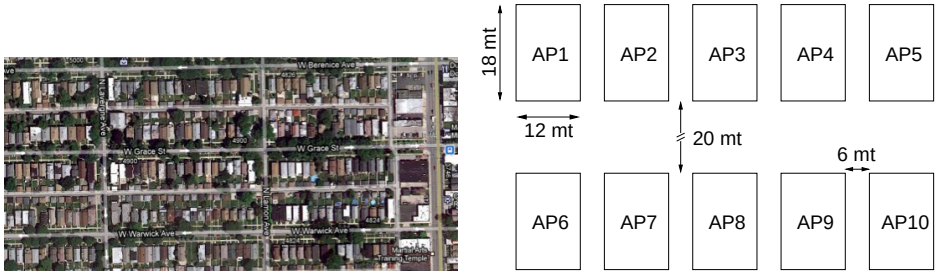


Figure 3.3. Federated detached houses scenario: Google view of the area (left) and abstract representation (right).

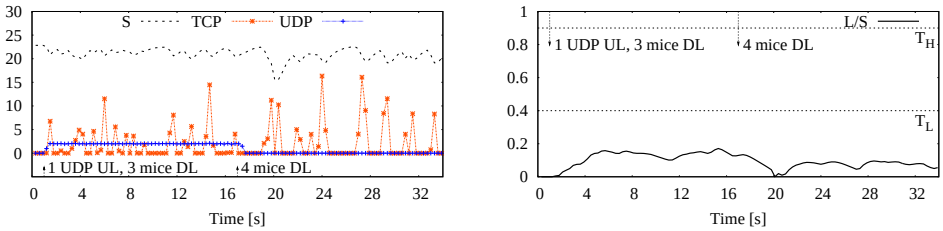
Finally, we remark that, upon receiving an OFFLOAD_REQUEST, a Gateway wishing to start an offload procedure defers its request till it receives a HANDOVER_COMMAND or an ABORT, and then perform a backoff. This ensures that in the RFN there is only one active offload procedure at the time. Also neighboring gateways answering to an OFFLOAD_REQUEST experience a very short freezing time, due to channel switching and RTS/CTS overhearing. In our evaluation, we upper-bound such a freezing time to one inter-beacon interval, i.e., 100 ms.

3.5 Evaluation in a Residential Scenario

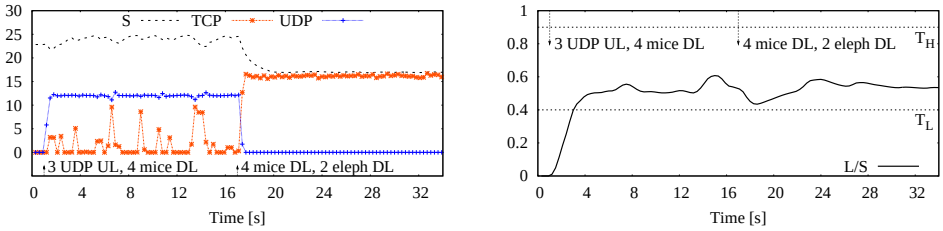
We implemented our algorithms and protocol in the Omnet++ v4.1 [64] simulator and starting from the IEEE 802.11g module included in the INETMANET extension. [65] extension. evaluated its performance in a realistic scenario referring to a neighborhood located in the suburbs of Chicago, IL. The RFN scenario, depicted in Fig. 3.3, includes 10 federated detached houses, each equipped with an IEEE 802.11g Gateway. To represent the propagation conditions over the wireless channel, we resorted to a refinement of the ITU indoor channel model, obtained using the experimental measurements presented in [66]; also, we implemented the automatic data rate adaptation scheme AARF [49]. The average fraction of Gateways in radio visibility of a WS, when a data rate of 6 Mbps is used, is 0.8. A handover delay of 300 ms accounts for the set-up of both an optimized mobile IP protocol and of the the link-layer reassociation procedure. Elastic traffic is simulated using TCP SACK, while inelastic traffic is represented by UDP flows. Both elephant and mice TCP flows are considered: the former represent bulk FTP transfers, while the latter correspond to an occasional, http-like file transfer, whose size is an instance of a random variable with negative exponential distribution and mean equal to 2 Mbytes. The payload size of TCP and UDP data packets is 1400 bytes. As for the algorithm and protocol parameters, we set the duration of the measurement period to 3 s, $\alpha = 0.2$, $N_L = 10$, $T_L = 0.4$ and $T_H = 0.9$, $\tau_r = 0.3$ s, and $\tau_p = 0.1$ s.

Below, we first evaluate the accuracy of the proposed algorithms in assessing the Gateway status and its suitability to accommodate additional WSs in its BSS. Then, we focus on the energy savings that our resource-sharing protocol can bring, while providing load balancing and meeting the user expectations in terms of throughput.

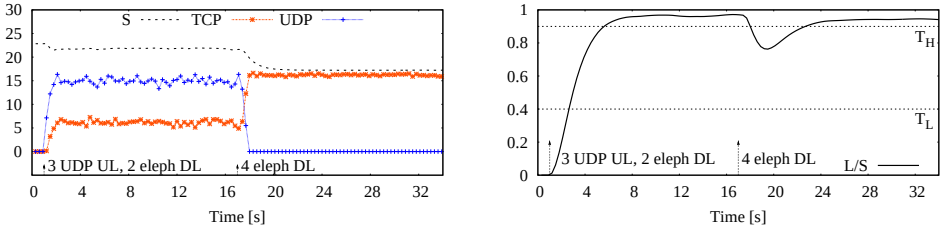
3.5.1 Gateway status and suitability assessment



(a) Light: aggregate throughput in Mbps (top) and normalized load, L/S (bottom)



(b) Regular: aggregate throughput in Mbps (top) and normalized load, L/S (bottom)



(c) Heavy: aggregate throughput in Mbps (top) and normalized load, L/S (bottom)

Figure 3.4. Detection of the Gateway status. Saturation and aggregate (elastic and inelastic) throughput ((a), (b), (c)); normalized load and status detection with respect to the thresholds T_L and T_H ((d), (e), (f)). The Light, Regular and Heavy status are always correctly detected.

For clarity, we start by considering only one Gateway; the case of several Gateways with overlapping coverages follows.

The first scenario we study corresponds to an underloaded BSS, which includes a WS originating an uplink 2-Mbps UDP flow, and three other WSs that are the destinations of one mouse TCP flow each. The aggregate throughput for elastic and inelastic traffic

is depicted in the top plot of Fig. 3.4(a), along with the saturation throughput S. The load estimate carried out by the Gateway is shown in the bottom plot, from which it can be seen that the Gateway correctly detects a Light status. At t=17 s, the UDP flow ends and the WS becomes the destination of one mouse TCP flow. Again, the Gateway correctly estimates to be in Light status thus showing that the saturation throughput is a good representation of the BSS capacity, and that our algorithm can accurately detect the BSS load level.

Fig. 3.4(b) presents the results for a medium-loaded BSS. Specifically, now there are three WSs that originate UDP traffic at 4 Mbps and are the destinations of a mouse TCP flow; a fourth WS is the destination of one mouse TCP flow. At t=17 s, the UDP streams end and two WSs become the destinations of an elephant TCP flow. The bottom plot in Fig. 3.4(b) shows that the Regular status is always detected.

Finally, Fig. 3.4(c) refers to an overloaded BSS with three WSs, each of which generates UDP traffic at 5 Mbps. Two of them are also the destinations of an elephant TCP flow. At t=17 s, the UDP streams end and two WSs become the destinations of an elephant TCP flow. The plots highlight the effectiveness of our algorithm in evaluating the BSS load under heavy traffic conditions too, and even for T_H as high as 0.9.

The next set of results, shown in Fig. 3.5, depicts the accuracy of our algorithm in estimating the suitability of a Gateway to accommodate additional WSs. In this case, a tagged Gateway receives offload requests from its federated Gateways and needs to assess how much room (if any) there is in its BSS. We refer to a scenario that includes four Gateways, three of which would like to relocate a WS to the tagged Gateway. To present different network conditions, we deploy the WSs so that, due to path loss, the initial data rate is 6 Mbps and consider a case where there is a mix of uplink and downlink flows. In particular, initially a WS originates one 1-Mbps UDP stream and one elephant TCP flow.

At t=8.2 s, a first relocation request is received for a WS that originates an elephant TCP flow and is the destination of a 0.5-Mbps UDP stream. The tagged Gateway computes its value of room-metric, which shows bandwidth availability (right plot), then, in our example, the WS is relocated to the tagged Gateway. At t=12.5 s, a second relocation request is received for a WS that originates a 0.5-Mbps UDP stream and is the destination of an elephant TCP flow. The room-metric correctly reflects the Gateway suitability to accommodate the WS, which is relocated to it. Finally, at t=16.1 s a third relocation request arrives, for a WS that is the destination of one 0.5-Mbps UDP stream and one elephant TCP flow. This time the requested bandwidth is higher than the current availability (i.e., the estimated load would exceed the Heavy status threshold), and ρ correctly drops below $(1 - T_H)$. The WS is therefore rejected.

3.5.2 Effectiveness of the resource-sharing protocol

We now evaluate the benefits brought by our protocol in terms of energy saving, along with its performance in terms of load balancing and traffic throughput.

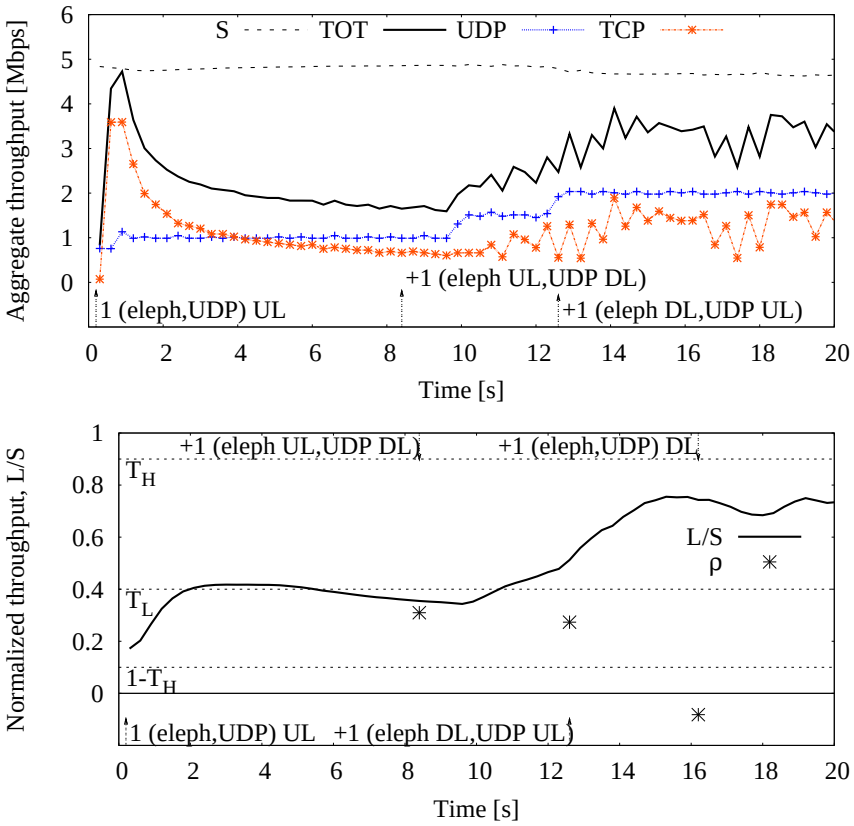


Figure 3.5. Relocation requests arrive at the Gateway for WSs that are sources and destinations of elastic and inelastic traffic. The room-metric ρ (denoted by the star marker in the right plot) correctly reflects the bandwidth availability within the BSS.

More specifically, in each scenario we compute the load level of every BSS, the number of WSs associated to each Gateway, and the difference between the MAC-layer throughput experienced by the WS with and without our energy-saving framework. As for the energy savings, we show the number of Gateways that the framework can switch off, as well as the energy consumed by the RFN with respect to the case where all Gateways are always "on". In order to compute the latter metric, we model the Gateway energy consumption as follows.

By relying on the specifications of available products, we consider that the power consumption of the 802.11 radio interface is equal to P_i=150 mW in idle mode, P_r=1.2 W in receive mode, and P_t=1.6 W in transmit mode [67], while the consumption of the low-cost, low-power interface (assumed to be an 802.15.4 radio) is p_s=186 μW in sleep mode and p_a =165 mW in receive/transmit mode [68]. As for the rest of the Gateway device, previous studies [69] have observed that the power consumption of a home Gateway, or, equivalently, of commercial modem/routers, is about $P_G = 4$ W and does not vary

significantly with the traffic load. Indeed, the idle state requires a large amount of energy because of the operations that have to be performed periodically to monitor the network state (e.g., DSL heartbeat, PPPoE link quality report). Thus, over a given observation period T, we compute the energy consumption of a Gateway by considering that in the "on" state its power consumption is P_G, plus that of the 802.11 radio and that of the 802.15.4 device in sleep mode. In the "off" state, instead, the only contribution is due to the 802.15.4 interface. The resulting value is given by,

$$T \cdot [t_{on} \left(P_G + P_i t_{on,i} + P_r t_{on,r} + P_t t_{on,t} + p_s\right) + t_{off} p_a]$$

where t_{on} (t_{off}) is the time fraction during which the Gateway is "on" ("off"), and $t_{on,i}$, $t_{on,r}$ and $t_{on,t}$ are the time fractions, during the "on" period, in which the 802.11 radio is in idle, receive and transmit mode, respectively. Given the energy model of the gateway, the energy saved by a switched-off GW (P_G) is always greater than the increased energy used by neighboring GWs and by the relocated WSs, whose maximum transmit power reaches 200 mW for recent MIMO products [70].

To better study the protocol dynamics, we first evaluate our scheme with scenarios featuring only uplink UDP traffic.

In order to evaluate the behavior of our scheme in Light and Heavy status, we consider a dynamic traffic scenario. Initially, all Gateways are "on" and they have 3 associated WSs each. At time t=0 s, every WS starts generating an uplink UDP stream at 1 Mbps (see Fig. 3.6(a)); since the per-Gateway load is 3 Mbps, all Gateways are in Light status. Then, between 60 and 68 s, every WS doubles its offered load (see Fig. 3.6(b)), driving the "on" Gateways into Heavy status.

The temporal evolution of the Gateways throughput, when all Gateways are initially in Light status, is shown in Fig. 3.6(a), where different marker/color combinations are used to represent the behavior of single Gateways. The Gateways that successfully carry out an offload procedure and become "off" correspond to downward curves, while Gateways that associate relocated WSs see their throughput grow. A sample of a successful offload can be observed in the interval $[3,4]$ s where a Gateway, upon switching itself off, relocates its three WSs to two other Gateways whose throughput therefore increases. Eventually (at t=8.5 s), the federated network settles at 3 "on" Gateways out of 10. Each "on" Gateway serves 10 WSs (see Fig. 3.6(c)) and is in Regular status.

Then, Fig. 3.6(b) shows the temporal evolution of the Gateways throughput when a sudden traffic increase drives the three "on" Gateways into Heavy status. As the WSs progressively double their offered load (between 60 and 68 s), two additional Gateways turn themselves on and come to the aid of the overloaded ones. We remark that the proposed algorithm always tries to minimize the number of "on" Gateways, thus the second one is switched on only when the first can no longer associate WSs without moving into Heavy status itself. When all Gateways are in Regular status (t=73 s), no further relocations occur and the network stabilizes at 5 "on" Gateways. The three Gateways that were "on"

at the end of the period depicted in Fig. 3.6(a) now have 7 associated WSs, while the first and the second Gateway that came in aid accepted 6 and 3 WSs, respectively, as shown in Fig. 3.6(c).

Next, we consider a different traffic scenario where initially all 10 Gateways serve the same number of WSs (namely, 2, 4, 6). Each WS generates a UDP flow with the same offered load, which is a varying parameter in different test runs. Fig. 3.7 shows the percentage of "off" Gateways, as well as the average number of WSs associated to a Gateway, upon reaching steady state. As expected, the number of switched off Gateways decreases as both the offered load and the number of WSs in the federated network increase. These results suggest that, for widely different load conditions, the configuration yielded by our solution well adapts to the system dynamics.

Next, we consider two sample traffic scenarios including TCP that allow us to evaluate the benefits of the offloading procedure in realistic settings. We consider that initially all Gateways are "on" and have three associated WSs each. Also, the nodes start transmitting at 54 Mbps; as already mentioned, the data rate is then adapted according to the AARF mechanism.

The first scenario includes a mix of TCP and UDP (uplink/downlink) traffic flows. We set up all BSSs to initially feature the same mix of traffic. Specifically, out of the three initially associated to every Gateway, two WSs originate one 0.5-Mbps UDP stream each and are the destinations of, respectively, one elephant and one mouse TCP flow, while the third WS originates an elephant TCP flow. Elephant TCP flows share a 10-Mbps link in the wired section of the network.

The top plot in Fig. 3.8 depicts the total throughput within each BSS controlled by a Gateway and highlights that, being in Light status, the Gateways try to relocate their WSs and turn themselves off. Around t=45 s, the RFN stabilizes with three Gateways that remain "on", two of which in Regular status (GW 3 and GW 4) and one in Light status (GW 8), as shown in the bottom plot. The WS distribution over the three "on" Gateways is as follows: 12 WSs are associated to GW 3 and GW 4, and 6 to GW 8. Note that, while GW 3 and GW 4 end up having a similar load, the load of GW 8 is much lower. However, GW 8 cannot relocate its WSs to either GW 3 or GW 4, as the additional load would drive the two Gateways into the Heavy status (see Fig. 3.8, bottom plot). As for the energy efficiency, the left plot in Fig. 3.9 depicts the saving achieved by each Gateway in the RFN, with respect to the case where all Gateways are "on". Though GW 3, 4 and 8 remain always "on" and have to serve all WSs in the RFN for most of the time, the overall energy saving exceeds 60%.

At this point, one may wonder about the degradation in performance that the WSs experience. Fig. 3.10 shows the difference between the throughput of the traffic flows in the initial configuration (i.e., all Gateways "on" and three WSs per Gateway) and the one experienced when the resource sharing protocol is applied (i.e., only three "on" Gateways). For clarity, in the case of TCP we show only the results for the flows experiencing the worst and the best performance. Observe that UDP streams practically experience no

losses and the variation in the throughput of TCP flows is marginal.

The second scenario features a similar combination of flows as the previous one, but we introduce two important changes. First, all elephant TCP flows now share a 100-Mbps link in the wired part of the network, thus allowing more breathing room for TCP congestion control, hence higher nominal throughput. Second, we removed the elephant TCP downlink flows from the WS associated to five out of ten Gateways, essentially earmarking those Gateways as candidates to start a successful offloading procedure. The per-Gateway throughput and the normalized load are displayed in Fig. 3.11. As expected, the five less-loaded Gateways are those that manage to offload their WSs to nearby Gateways and turn themselves off, as shown by the downward curves in the bottom plot. The WS distribution over the five "on" Gateways turns out to be the following: 4 WSs are associated to GW 2, GW 4 and GW 8, while 3 and 5 WSs are associated, respectively, to GW 3 and GW 9. In the first scenario, the reassociation of WSs to nearby Gateways caused the latter to see a throughput increase since local TCP flows were throttled on the wired link, and the newcomers could easily fill the available room. Now, instead, the availability of a larger backhaul capacity allows all TCP flows to greedily fill the available room with elastic traffic *prior* to receiving offloaded WSs (top plot of Fig. 3.11). As a result, when offload requests are dispatched, the five Gateways where downlink TCP flows are still active are chosen after establishing that the additional WSs do not cause their status to become Heavy. Looking at the total throughput that each active Gateway exhibits (top plot of Fig. 3.11), the changes are minimal. However, as depicted in Fig. 3.12, single elephant TCP flows on those Gateways incur throughput losses ranging from a few percentage points up to 60% in the worst case. The energy savings in this second scenario, highlighted in the right plot of Fig. 3.9, are quite remarkable (almost 50%). Additionally, not having their total throughput affected, "on" Gateways experience a negligible increase in their energy consumption.

Summary: The above results show that our approach exhibits high accuracy in estimating the network load conditions and the Gateway capability to accommodate additional WSs. Thanks to such accuracy, our framework can provide high energy savings, without significantly degrading the performance experienced by the users. In particular, in the case of inelastic traffic, users always obtain their expected throughput, independently of the BSS to which they are associated. We also mention that results (omitted for lack of space) with varying thresholds T_L and T_H have shown that these parameters can be effectively tuned so as to let the framework yield either higher energy savings or higher throughput for elastic traffic.

3.6 Comparison with a centralized optimal allocation

Assuming the feasibility of a centralized optimal scheme, we formulate the pertaining optimization problem in order to compare its solution with the allocation given by our

resource sharing protocol. Given a set of Gateways $\mathcal{G}$, and given a set of active WS ($\mathcal{N}$), we want to minimize the number of "on" Gateways ($\mathcal{G}$). Thus we formulate the following optimization problem:

$$obj : min \sum_{j=1}^{\mathcal{G}} x_j \tag{3.2}$$

$$\sum_{k=1}^{\mathcal{N}} T_k y_{kj} < S_j \qquad \forall j \in \mathcal{G} \tag{3.3}$$

$$\sum_{j=1}^{\mathcal{G}} y_{kj} = 1 \qquad \forall k \in \mathcal{N} \tag{3.4}$$

$$y_{kj} \leq R_{kj} \quad \forall k \in \mathcal{N}, j \in \mathcal{G} \tag{3.5}$$

$$y_{kj} \leq x_j \quad \forall k \in \mathcal{N}, j \in \mathcal{G} \tag{3.6}$$

Where

$$N_j = \sum_{k=1}^{\mathcal{N}} y_{kj}$$

$$R_j = \frac{\sum_{k=1}^{\mathcal{N}} y_{kj} R_{kj}}{\sum_{k=1}^{\mathcal{N}} y_{kj}}$$

$$P_j = \frac{\sum_{k=1}^{\mathcal{N}} y_{kj} P_k}{\sum_{k=1}^{\mathcal{N}} y_{kj}}$$

The objective function (3.2) minimizes the number of "on" Gateways. $x_j \in \{0 = off, 1 = on\}$ represents the Gateway status, $y_{kj} \in \{0,1\}$ indicates whether WS_k is associated with Gj ($y_{kj} = 1$) or not ($y_{kj} = 0$), Tk is the total throughput of station k, S_j is the saturation throughput of the BSS GW_j, and R_{kj} is the rate achievable by station k on Gateway j. Note that S_j depends on N_j, R_j and P_j, which are the number of allocated WS, the average rate, and the average payload size of Gateway j, respectively. The limit on the maximum achievable load on a given Gateway is enforced by condition (3.3), that bounds the total throughput generated by the assigned WS to the saturation throughput S_j. Condition (3.4) guarantees that a given WS is assigned to one and only one Gateway; condition (3.5) allows a WS to associate only to Gateways that are within coverage; condition (3.6) ensures that a Gateway is "on" if at least one WS is associated to it.

Since the saturation throughput S_j has a non linear formulation, the optimization problem becomes a Mixed Integer Non Linear Problem (MINLP), that cannot be solved to the optimum. However, we can reduce the problem to a Mixed Integer Quadratic Constrained Problem (MIQCP) by considering all WSs to transmit the same traffic, namely an UDP

uplink flow ν^u, and by linearizing the saturation throughput (2.6) in function of N_j, R_j and P_j:

$$\hat{S}_j = \alpha_1 N_j + \alpha_2 R_j + \alpha_3 P_j + c_u \tag{3.7}$$

where α_1, α_2, α_3 are the linear coefficients, while c_u is set so as $\hat{S}_j > S_j$. Since this approximation considers a saturation throughput always greater that the achievable one, it results that the optimal solution is a lower bound with respect to the number of "on" Gateways.

Thus, we can rewrite the maximum load constraint (3.3) as:

$$\begin{aligned} \nu^u \sum_{k=1}^{\mathcal{N}} y_{kj} &< \alpha_1 N_j + \alpha_2 R_j + \alpha_3 P_j + c_u \\ \nu^u Y_S &< \alpha_1 Y_S + \alpha_2 \frac{\sum_{k=1}^{\mathcal{N}} y_{kj} R_{kj}}{Y_S} + \alpha_3 \frac{\sum_{k=1}^{\mathcal{N}} y_{kj} P_k}{Y_S} + c_u \\ \nu^u Y_S^2 &< \alpha_1 Y_S^2 + \alpha_2 \sum_{k=1}^{\mathcal{N}} y_{kj} R_{kj} + \alpha_3 \sum_{k=1}^{\mathcal{N}} y_{kj} P_k + c_u Y_S \end{aligned} \tag{3.8}$$

where

$$Y_S = \sum_{k=1}^{\mathcal{N}} y_{kj}$$

By substituting condition (3.3) with (3.8), we obtain a MIQCP that can be solved to the optimum with commercial solvers like cplex.

We compare the percentage of the switched off gateways given by our resource sharing protocol to what achieved by the optimization problem. We initially assign to each gateway 2 WSs and we let each WS generate an UDP uplink flow at different offered traffic, starting from 0.5 Mbps and up to 6 Mbps. As shown in Fig. 3.13, our protocol achieve close to optimal results, switching at most one gateway less than the optimal solution for throughputs up to 5 Mbps. This difference is mainly due to the linear approximation of S and to the guard parameters T_H and T_L, which are not included in the mathematical formulation.

3.7 Testbed

We also demonstrate the effectiveness and the benefits of our energy-efficient cooperative protocol through a real deployment emulating a residential scenario. The testbed can be shown as an highly interactive demo, where users can generate traffic within a BSS through a wireless station, like a smartphone or a notebook, and observe, through a web

interface, the protocol behavior and the network topology changes caused by the new traffic scenario.

3.7.1 Architecture

As shown in Figure 3.14, the testbed is a federated network composed by three GWs, six WSs, a web server and a router. Each GW acts as AP of a 802.11g network operating on a different channel in the 2.4 GHz band, and secured with WPA-PSK, which is part of the 802.11i standard and it is widely adopted in residential networks. Every WS knows all access keys in the federated network, hence it can associate to any GW. We included four laptops and two smart-phones as WSs, so as to create the heterogeneity of a real-life residential environment.

When "on", GWs notify the web server about the WSs associated to them, along with the WSs traffic profile. The web server has two functions: (i) it provides contents to WSs and (ii) it graphically shows the testbed status over time in a web page. Specifically, such testbed monitoring interface shows the actual association of the WSs, the GWs current load (L), the aggregated throughput, the saturation throughput (S), and L/S, which is used to determine the GWs status. On top it displays the total energy consumption within the federation and, when the protocol is started, the total energy saving with respect to the case where all GWs are always "on". The layout of the monitoring interface is given in Figure 3.15.

3.7.2 Hardware and software description

GWs feature an Alix PC Engines motherboard, equipped with an AMD Geode 500 MHz processor, one IEEE 802.11 b/g compliant Wistron DCMA-82 Atheros wireless card and one omnidirectional antenna with a gain of 5 dBi. Each Alix runs OpenWrt Backfire, a Linux distribution for embedded devices, while the WSs can run any operating system.

The passive traffic measurements needed by the protocol are implemented *on the GW* within the mac80211 module of the Linux wireless driver *compact-wireless 2011-21-01* [43]. Note that, since we modified only the mac80211 module and not the GW hardware, such measurements work on any device. Specifically, at the BSS level we keep track of: average size of the packet payload, average SNR, average data rate, average packet error rate (PER), number of associated WSs. For each WS, at the MAC layer we measure: all throughput contributions (elastic/inelastic, downlink/uplink traffic) with the corresponding number of handled packets, the average packet payload, the average data rate, the average SNR. The PER is computed as the estimated fraction of the erroneously received/transmitted packets. For received packets we count the CRC errors (at the PHY and MAC layer), while for transmitted packets we count all unsuccessful transmission attempts at the physical layer. This results in a worst case PER estimation, as collisions

are also included in the count [3]. All measurements are made available to the application by the mac80211 module every 2 s.

3.7.3 The handover problem

Despite the broad spectrum of schemes and protocols aimed at providing seamless or faster handover in 802.11 networks, none of them has reached wide adoption. Solutions based on Mobile IP are complex because they rely on many dependencies, requiring both new hardware and new software to be deployed. This would imply additional costs for both hardware vendors and network operators, with no practical business models that justify these additional expenses [72]. The Inter Access Point Protocol (IAPP), standardized as 802.11f [73], cannot be used to provide faster re-authentication with 802.11i-based security standards, as the latter does not allow security context transfer between GWs. 802.11i provides some alternatives to reduce handover delays, namely Pairwise Master Key Caching (PMK Caching) and pre-authentication over the distribution system. The main problem of these schemes is that they are WS initiated, which makes them unsuitable for a gateway-centric solution like the one we propose. The recent 802.11v [62] IEEE standard includes a BSS transition management that enables a GW to request WSs to handover to another GW, or to indicate a set of preferred GWs to a WS, based on network load balancing needs. However, the GWs must be part of the same Extended Service Set (ESS) and the WSs should support this scheme. In conclusion, nowadays there is no inter-domain network-managed seamless handover solution able to be deployed without changing both GWs and WSs, and without requiring new network element such as the home agent of Mobile IP. As a result, current GWs and WSs come with no fast handover technology ready to use.

We modify the ODIN framework in order to use the VAP concept and to trigger the handover and specify the destination Gateway from our protocol. We use the dynamic channel switching procedure as proposed by the 802.11h [63] standard and implemented in the latest version of the linux wireless driver [43]. We did not implement any layer 3 handover management and we put all GWs as well as the WSs on the same network. As we use commodity gateway, the switch off/on is emulated and the energy consumption is estimated as explained in Section 3.5.2.

3.7.4 Testbed results

We complement the evaluation of Section 3.5.2 by studying the protocol behavior in presence of sleeping GW. Thus, we start with two “on” GWs, namely GW1 and GW2 and one “off” GW, GW3. We generate UDP downlink traffic using iperf. Then, we show

[3] Collisions cannot be discriminated from errors due to harsh channel conditions without changing the WS software or the 802.11 protocol.

the behavior and performance of the offload procedure in the following cases: (1) one or more GWs are in Light state and try to get rid of their WSs in order to switch off without waking up additional GWs; (2) one GWs is in Heavy state and try to relocate one WSs without waking up additional GWs. In the light scenario, initially each "on" GW has two WSs each one having a downlink flow at 2.5 Mbps. After 30 s GW1 switches "off" an hands over its WSs to GW2, that takes the additional throughput almost instantly. Consequently, the energy consumption drops from 10.5 W/h down to 6.2 W/h. The throughput and the energy evolution are shown in Figure 3.16. As expected, GW3 does not wake up to help GW1. In the heavy scenario, reported in Figure 3.17, GW1 has 4 WSs that totals 20 Mbps (5 Mbps each) of aggregated throughput. It is thus in the Heavy state, while GW2 is in Light state with 2 WSs totaling 5 Mbps of aggregated throughput. GW1 hands over one WSs to GW2 and reliefs from congestion, while GW3 stays "off". Both GW1 and GW2 ends up in Regular status and the energy consumption does not change, as expected. We also report the cumulated packet loss evolution as reported by the iperf server in Figure 3.18, noting that it is limited in both scenarios and that it is mainly due to the time the application takes to restart the flow.

3.7.5 Demo scenario

We now describe the scenario that we shown during the demonstration of our testbed. Among all realistic traffic patterns, we consider three kinds of traffic flows: *mice TCP*, *elephant TCP* and *UDP*. A *mice TCP* flow represents a whispering WS, e.g., a user browsing the web without any kind of background traffic. We emulate whispering stations by implementing the traffic model for mobile web browsing proposed in [71], where the mean web page dimension is 4 KB and the mean reading time (interval time between two consecutive web page requests) is 15 s. An *elephant TCP* flow consists of a bulk HTTP download of a content locally stored at the web server. Finally, we introduce *UDP* flows with a bit rate similar to widely used audio/video peer-to-peer real-time applications, like Skype. All flows are established between WSs and the web server.

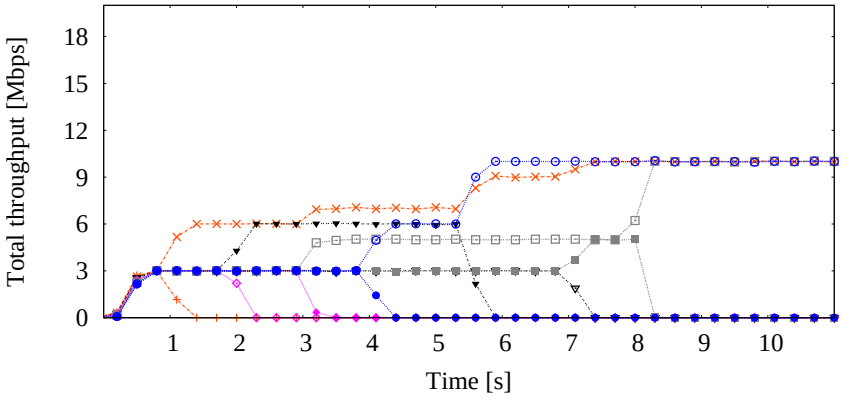
Initially we consider two WSs associated to each GW. Then, we show the behavior and performance of the offload procedure in the following cases: (1) one or more GWs are in Light state and try to get rid of their WSs in order to switch off; (2) one or more GWs are in Heavy state and try to relocate their WSs without waking up additional GWs, (3) one or more GWs are in Heavy state and a GW in "off" state has to be woken up to accommodate for the relocation. During the demo users are allowed to interact with one client, either a mobile phone or a notebook, adding traffic to the BSS to test the reaction of our scheme, e.g., a user could surf the web or start a bulk TCP-based transfer. The system behavior is shown by the testbed monitoring interface (see Figure 3.15).

3.8 Final remarks on resource sharing protocol

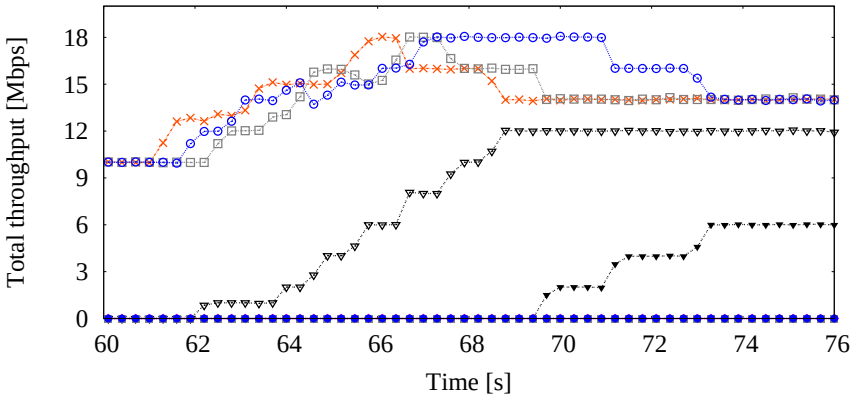
We designed a set of procedures aimed at managing underload and overload conditions in wireless Gateways of federated households. Exploiting load monitoring of uplink/downlink elastic and inelastic traffic, we introduced offload procedures that allow (i) an underloaded Gateway to relocate all of its WSs and thus switch off; (ii) an overloaded Gateway to relocate some of its WSs and alleviate its status. By simulation, we showed the effectiveness of the procedures in realistic federated neighborhood scenarios.

Energy saving introduced by algorithms such as ours have the potential to positively impact the global effort to achieve green networking, if implemented on a large scale. As expected, energy savings, though remarkable, do not come for free and at times could result in a somewhat downgraded experience for ongoing elastic flows. However, our algorithm allows home users to tinker with performance knobs (such as the α parameter) in order to tradeoff "greenness" and QoS according to their wishes.

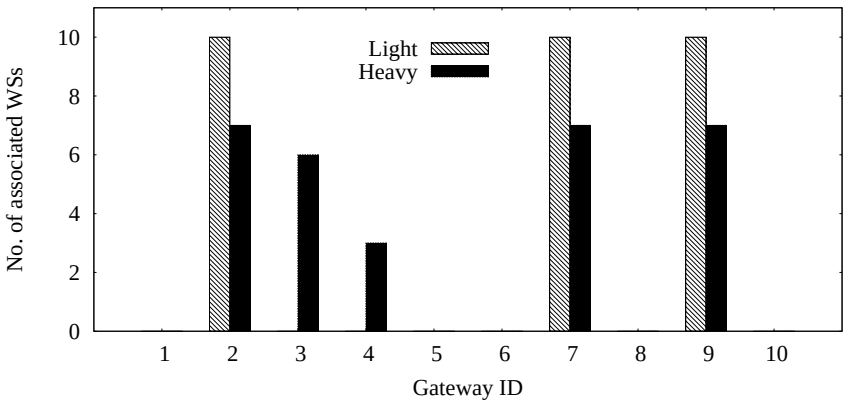
Our testbed showed a practical approach for real networks aiming at providing considerable energy savings without (i) significantly affecting user experience nor (ii) requiring changes in the WS hardware or software. Savings can be extended to ISPs by adopting DSLAM line aggregation as shown in [13]. Furthermore, our framework provides guidelines for the design of the GW hardware so as to benefit from potential energy savings.



(a) Gateways throughput (Light status)

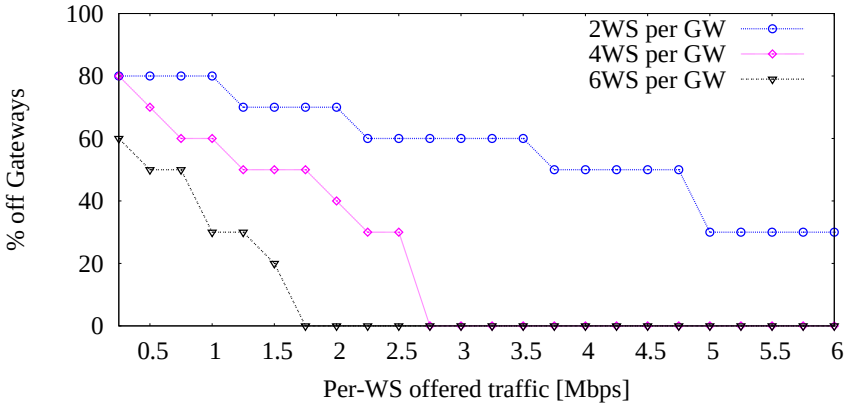


(b) Gateways throughput (Heavy status)

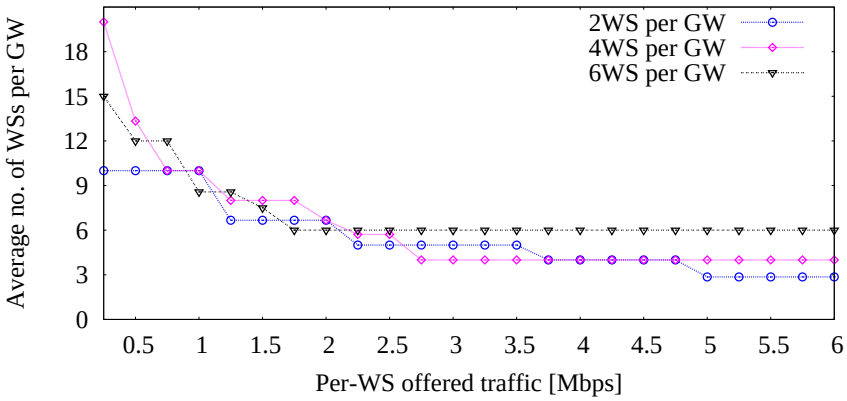


(c) WS distribution over the Gateways

Figure 3.6. Temporal evolution of the Gateways throughput and WS distribution, under Light and Heavy conditions.



(a) Percentage of "off" Gateways



(b) Average number of WSs/Gateway

Figure 3.7. Percentage of "off" Gateways and average number of associated WSs per Gateway, as the WS offered load varies and for a different initial number of WSs per Gateway.

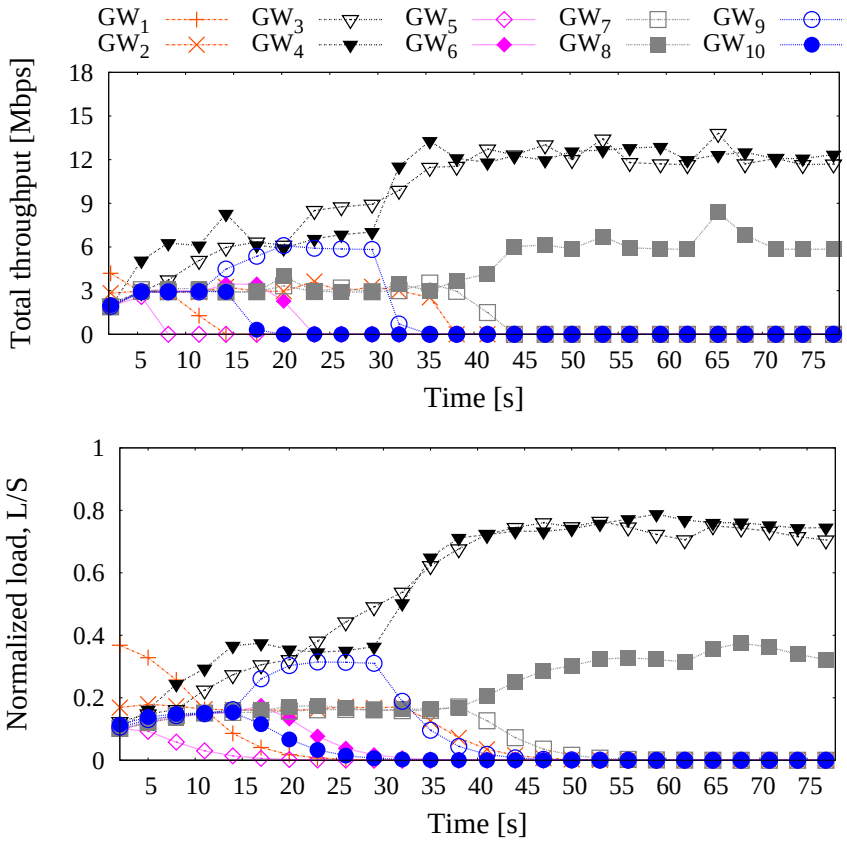


Figure 3.8. First scenario: time evolution of the BSS throughput (top) and of the normalized load (bottom).

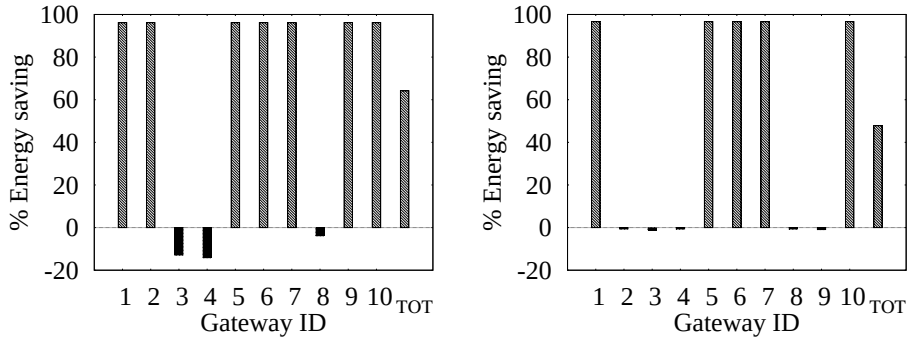


Figure 3.9. Energy saving in the first (left) and second (right) scenario.

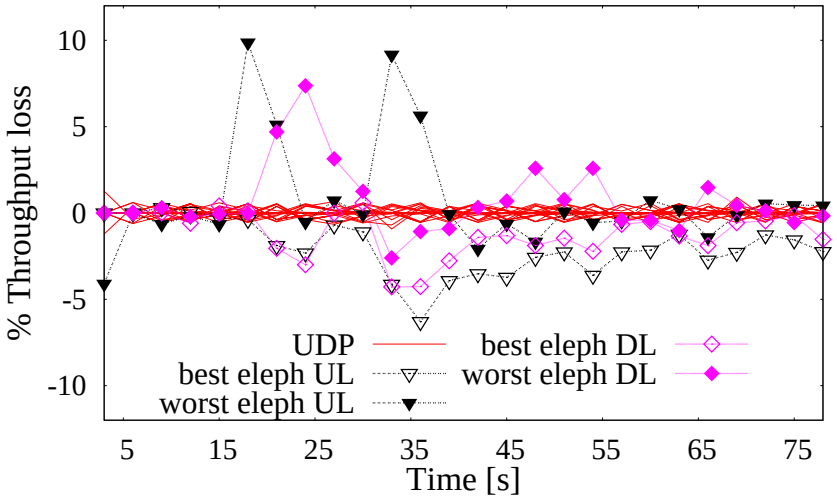


Figure 3.10. First scenario: time evolution of the BSS throughput loss.

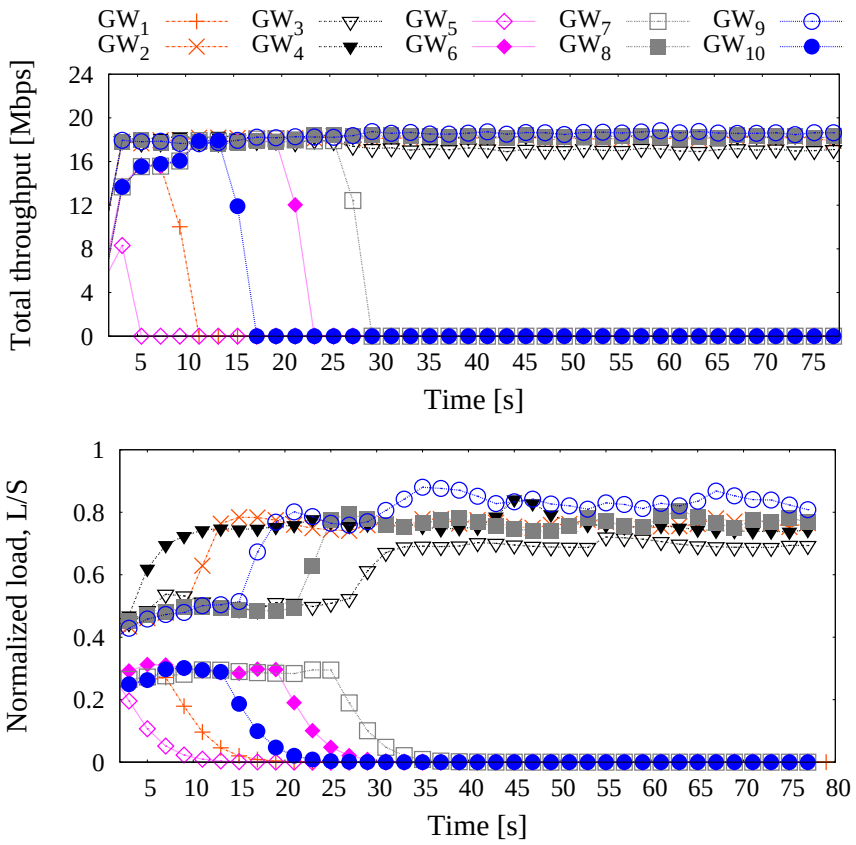


Figure 3.11. Second scenario: time evolution of the BSS throughput (top) and of the normalized load (bottom).

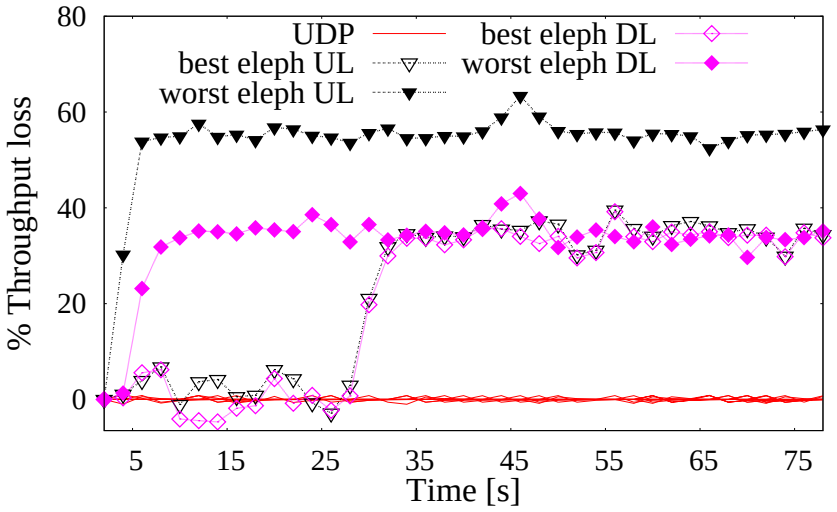


Figure 3.12. Second scenario: time evolution of the BSS throughput loss.

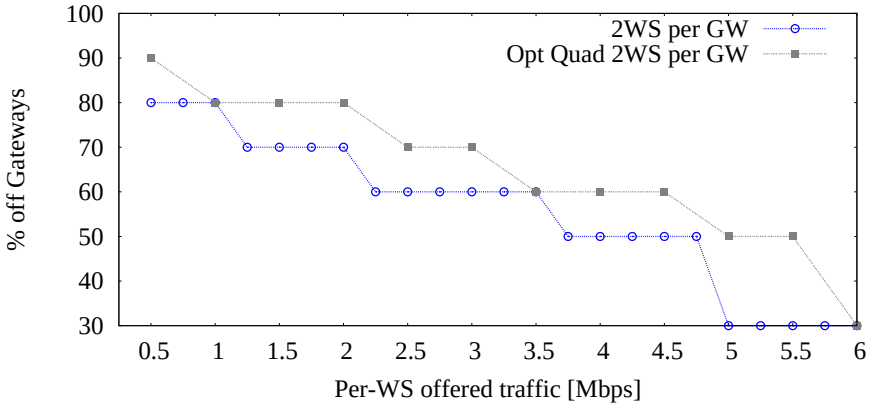


Figure 3.13. Comparison between the wireless resource sharing protocol and the optimal solution

Figure 3.14. Testbed architecture.

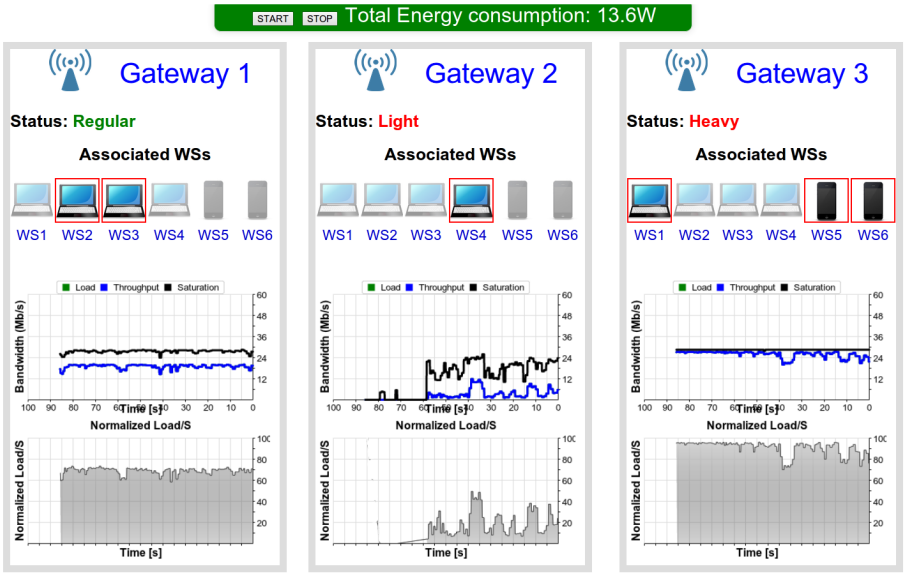
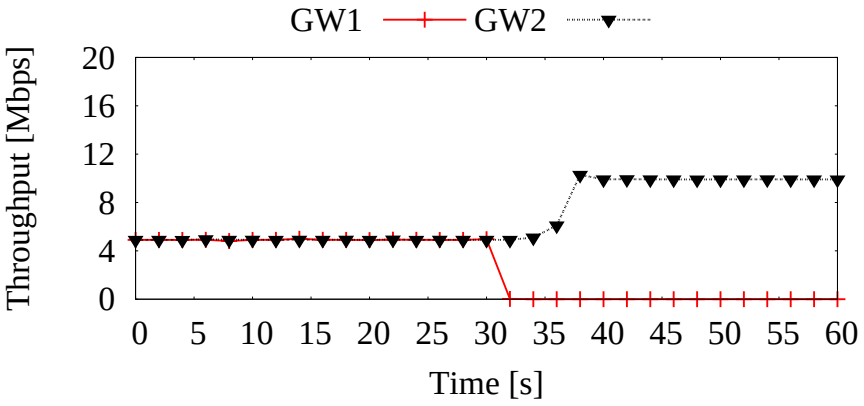
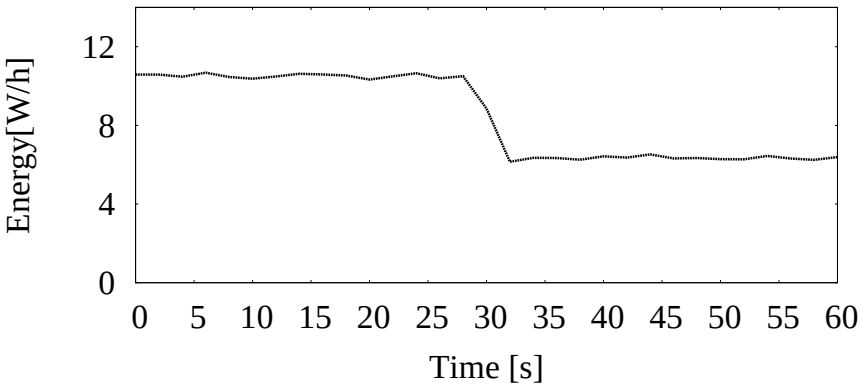


Figure 3.15. Testbed monitoring interface.

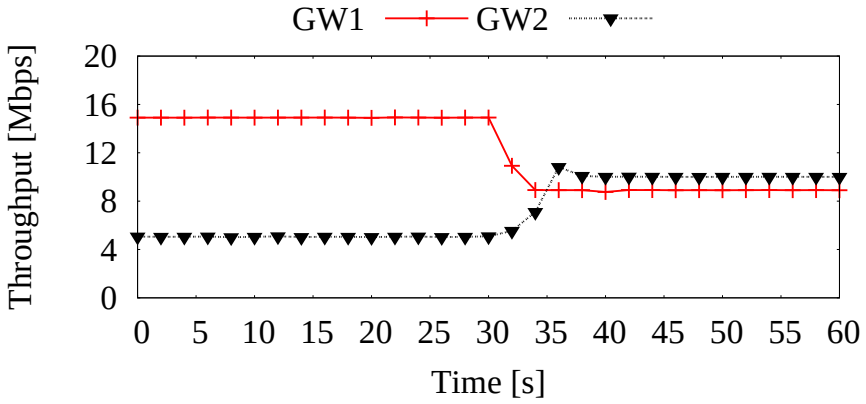


(a) Per-GW throughput temporal evolution case

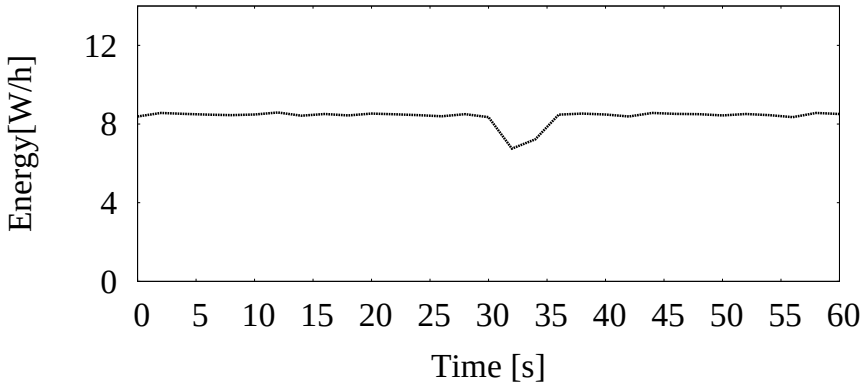


(b) Total energy consumption temporal evolution

Figure 3.16. Light scenario: temporal evolution of the per-GW throughput (a) and of the total energy consumption (b).



(a) Per-GW throughput temporal evolution case



(b) Total energy consumption temporal evolution

Figure 3.17. Heavy scenario: temporal evolution of the per-GW throughput (a) and of the total energy consumption (b).

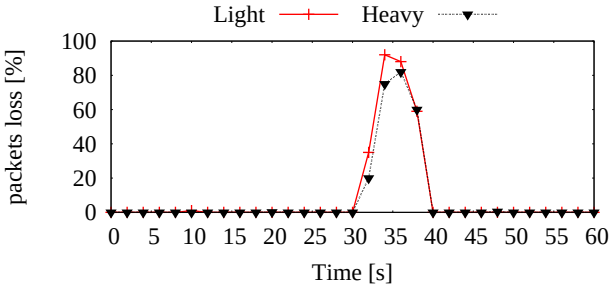


Figure 3.18. Packet loss in the light and heavy case.

Chapter 4

Enhancing connectivity with 3G OnLoading

The co-existence of cellular and wired networks has been exploited almost exclusively in the direction of OffLoading traffic from the former onto the latter. We claim that there exist cases that call for the exact opposite, i.e, use the cellular network to assist a fixed wired network. In particular, we show that by "OnLoading" traffic from the wired broadband network onto the cellular network we can usefully speedup wired connections, on the downlink or the uplink. We consider the technological challenges pertaining to this idea and implement a prototype 3G OnLoading service that we call *3GOL*, that can be deployed by an operator providing both the wired and cellular network services. By strategically OnLoading a fraction of the data transfers to the 3G network, one can significantly enhance the performance of particular applications. In particular we demonstrate non-trivial performance benefits of *3GOL* to two widely used applications: Video-on-Demand (VoD) and multimedia upload. We also consider the case when the operator that provides wired and cellular services is different, adding the analysis on economic constraints and volume cap on cellular data plans that need to be respected. Simulating *3GOL* over a DSLAM trace we show that *3GOL* can reduce video pre-buffering time and we design a simple estimator to compute the daily allowance that can be used towards *3GOL* while respecting caps. Our prototype is currently being piloted in 30 households in a large European city by a large network provider.

Main Results

- Using active measurements we find that 3GOL can increase downlink throughput of ADSL connections by $\times 2.6$ and uplink capacity by $\times 12.9$, while using 3 devices for 3G OnLoading. We see variations across locations, where rural areas seem to experience greater speedup but urban areas also have non-negligible benefits 4.2. Similar findings have been previously reported [75, 76].
- We implement a 3GOL prototype and quantify performance gains in the wild for

two applications: video-on-demand and uploading photos. These gains tend to be consistent across the day and lead to a maximum performance improvement of $\times 3.8$, $\times 4$ and $\times 6$ for pre-buffering, downlink and uplink respectively. We find that performance gains are present even when we use 3GOL to assist a 20 Mbps downlink residential connection (Sec. 4.4). We also test Multipath TCP (MPTCP) [77] against our implementation and show that MPTCP is outperformed by as much as $\times 2.3$.

- Our trace driven analysis shows that we can decrease latency for video delivery by at least 20% for 50% of the users, while respecting volume caps (Sec. 4.5).
- We investigate the load on the cellular network when 3GOL operates under volume caps and we find that the cellular network can cope with the onloaded traffic. We further quantify the additional load in function of the percentage of users adopting 3GOL (Sec. 4.5).

4.1 Context for 3GOL

In this section, we present a case for the viability for 3GOL as well as the high level description of the architecture. We first perform a back-of-the-envelope calculation to understand the capacities of the different networks (ADSL, cellular), then we study the leftover bandwidth pattern of cellular networks as well as the opportunities to use the same. Then, we discuss 3GOL related economic constraints both for the users and for the network. Then we address the future evolution of connectivity to assess the long term viability of 3GOL. We finally end this section with a high-level description of the architecture for 3GOL.

4.1.1 Comparison of capacity

If we assume that one cellular tower provides coverage to an area of 200 meters radius, and a typical population density of 35000 inhabitants per Km^2, as in a typical downtown dense metropolitan area [1], then each cell offers services to 4375 subscribers. If we assume that each household has 4 people and that we have 80% penetration of ADSL connectivity, then each cell covers 875 ADSL connections. We do not consider VDSL or fiber relevant as VDSL benefits are limited in range (2km), whereas cable/fiber deployment is planned based on demographic parameters such as population density. Consequently, fiber is mainly deployed in urban areas. We focus therefore on generic ADSL; and with an average downlink speed of 6.7 Mbps[2], the overall ADSL downlink capacity for the cell

[1] http://www.demographia.com/db-citydenshist.htm

[2] 6.7 Mbps is the average downlink throughput reported by Netanalyzr [78].

area would be 5.863 Gbps. Given that wired networks tend to be oversubscribed at the access, this number would be slightly lower but still on the order of a couple Gbps. The same area is covered by a cell tower with a typical $40 - 50$ Mbps backhaul connectivity to the rest of the Internet. Therefore the cellular network is 1-2 orders of magnitude smaller in terms of capacity than its wired counterpart serving the same geographical region. In the uplink direction, the difference can be smaller since ADSL might involve 1/10 asymmetry in capacity in uplink/downlink.

4.1.2 Cellular has leftover bandwidth at certain times

From the above analysis, it results that we cannot use the cellular network at all times. We now investigate if there are some times that offer opportunities to onload traffic to the cellular network. As with ideas already proposed for backbone networks [79], we note that it is possible to use the cellular network only during *off-peak* hours and by doing so avoid triggering upgrades or congestion. This is possible very often because cellular networks are provisioned for peak usage, while actual user traffic follows strong diurnal patterns, thereby leaving unused capacity for 3GOL.

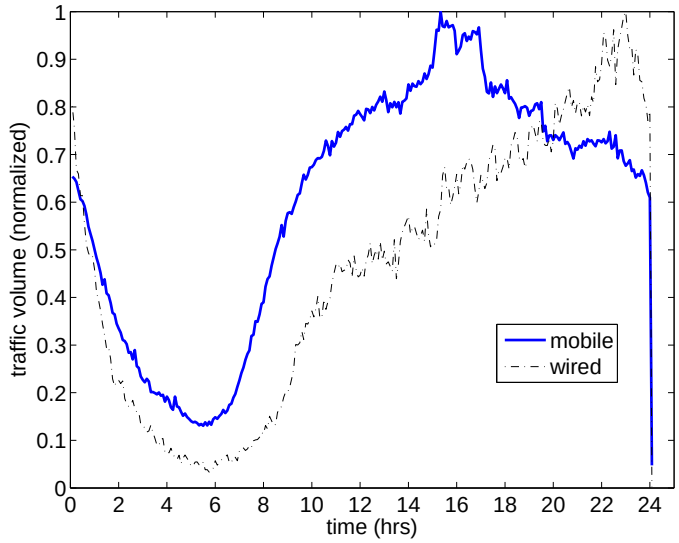


Figure 4.1. Traffic pattern over a day on cellular and wired network. Note the different peak timing on the two networks and the diurnal pattern in the cellular network

In Fig. 4.1, we plot normalized traffic from a DSLAM (Table 4.1) and a 3G network (data traffic, Table 4.1) from a single country. We do not have access to similar datasets from other countries but are in the process of acquiring them. Our first observation is that the cellular data follows a diurnal pattern as also reported by other studies [76]. This points to the fact that the cellular network is not constantly loaded, and hence data from

the wired network can be onloaded to the cellular. Our second observation from Fig. 4.1 is that the peaks for wired and the cellular are not perfectly aligned, which is confirmed in [76]. This is not too surprising – most users at home prefer to use their wired connection (via Wi-Fi or other technologies), as opposed to the cellular network. Furthermore, previous work [75] has shown that there is high spatial diversity in terms of load between cells, and some cells have left over capacity even during peak hours.

4.1.3 Using leftover capacity: Powerboosting

Although leftover capacity exists in cellular networks, we should carefully select the wired applications that would be allowed to use it. For example, we would not want to onload long running bulk (e.g., P2P) downloads. Applications that need onloading only for a short period of time seem ideal. Towards this direction, we develop a service that resembles the offering of PowerBoost by Comcast; delivering files at a much higher speeds for a short amount of time. Implementing PowerBoost is possible in cable networks since they have a shared-medium whose capacity can be allocated and released at short time-scales. ADSL networks, in contrast, have a dedicated twisted pair and need to adhere to long synchronization delays of the modems at the two ends. Therefore, they cannot implement PowerBoost natively. One of the main contributions of our work is the observation that by assigning excess cell network capacity to an individual customer for a limited amount of time we can effectively implement PowerBoost for ADSL with the help of the cellular network. Such temporary assistance could: a) assist the downlink of ADSL to reduce the pre-buffering time for more responsive video playout. and b) assist the uplink of ADSL to source content from home, e.g., uploading photos.

If 4G is available, the concept of 3GOL is even more compelling. With the reduced latency, and the large increase of bandwidth, the period of powerboosting time might be extremely short, reducing the overhead added on the cellular network. In some countries, where the deployment of 4G is faster than fiber (like France, or India), it is largely plausible that 3GOL might be a solution to improve home connectivity.

4.1.4 Future of connectivity

Next we put 3GOL in context with respect to network evolution. Fixed broadband access speeds have been growing by nearly an order of magnitude every 4 years since 1982 (Fig. 4.2). EU plans for fast broadband foresee 100% coverage of 20 Mbps (or more) for EU citizens by 2020. However, the majority of member states are still on the way to support 100% basic broadband (512 kbps to 4 Mbps) coverage [80]. In the US, progress of new wired broadband deployments in rural areas is slowing down, and there are no concrete plans to reverse this trend [81]. FTTH (with speeds upto 100 Mbps) is continuing to grow but its penetration is still limited (e.g., France at 13.5%) and operators carefully deploy these networks because per customer revenue is still low with respect to

investment [82], while mobile broadband is taking off and will soon pass wireline in total number of conenctions [83]. On the other hand, mobile broadband is taking off and will soon pass wireline in total number of connections. In the EU, the highest growth rate is in mobile broadband, where take-up more than doubled in 2011 [83]. Also, according to [83], 85% of the world's population will be covered by WCDMA/HSPA networks by 2017 and the number of mobile broadband subscriptions will reach 5 B from the actual 1.1 B. With the advent of LTE (already deployed in some countries around the world), mobile broadband access speeds are close to 100 Mbps, and LTE-Advanced, expected in 2013, promises speeds on the order of 1 Gbps.

As we look into the future, the cost evolution of wired and wireless technologies remain favorable to onloading as well. A recent report from A.T. Kearney [84] estimated the cost of bringing fiber to all households in Europe to 300 billion euro and the corresponding cost for bringing LTE to all mobile subscribers to 118 billion. Assuming similar SLAs as today[3], we conclude that the per Mbps cost of forthcoming wired and cellular technologies is comparable. Therefore, onloading remains feasible, not only in terms of capacity, but also cost.

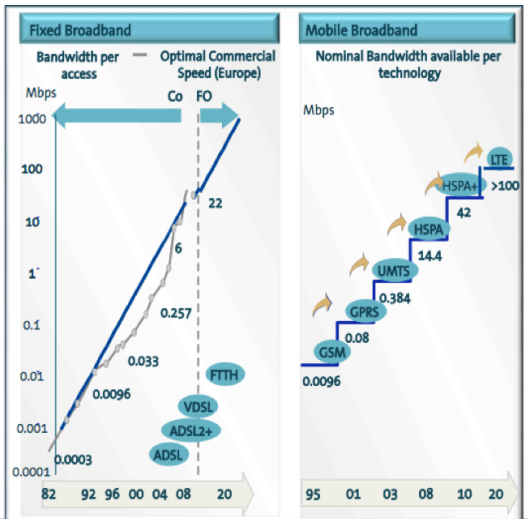


Figure 4.2. Evolution of wireless vs wired networks

4.1.5 Architecture sketch

We want 3GOL to be easily deployable and transparent for the applications. Hence, we propose to implement it as an Over The Top (OTT) solution, where a typical service like

[3]The oversubscription ratios of the future wired and mobile broadband networks are likely to remain similar to what they are today.

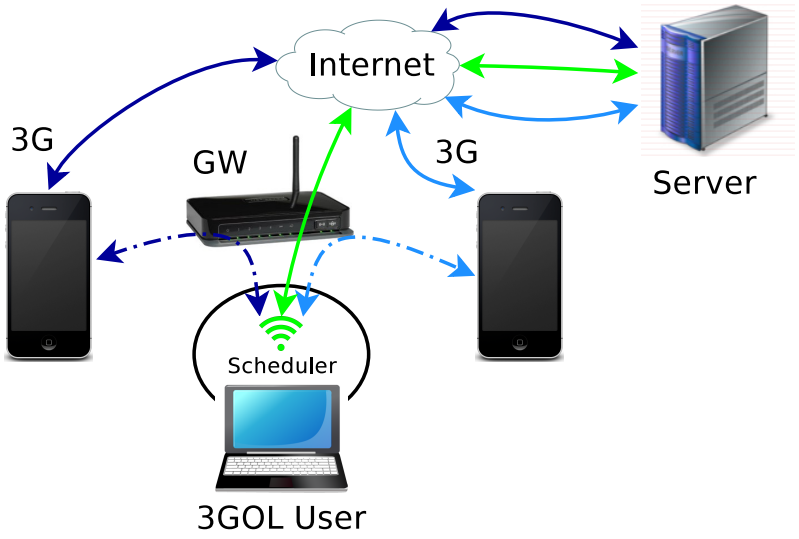


Figure 4.3. 3GOL architecture with one client (3GOL User), one residential gateway (GW) plus two 3G enabled devices.

voice or messaging is provided over the network, without using the native functionality of the network. In the OTT architecture (Fig. 4.3), every device participating in 3GOL is connected to the LAN managed by the residential gateway (GW) using either wireless or wired connections. First, we detail a network-integrated solution, where 3GOL is provided by a network operator that provides both wired and cellular service. Later, we describe how the architecture changes when the two providers are different.

In the network-integrated case, the single operator could provide the 3GOL traffic for free to its mobile subscribers during non-peak hours, i.e, avoid metering and subtracting it from their mobile data plans. This way it would benefit its ADSL business while using available resources in the corresponding 3G network. Network integration refers primarily to the communication between the billing elements inside the wired and wireless network, such that the respective pricing policies can be applied. Each device receives the permission to transmit from the 3GOL backend server, which is revoked by the same when congestion is detected. The backend server interfaces with the 3G network monitoring system and checks whether utilization in the affected area is below an acceptance threshold. If it is, the transmission is authorized and a permit is cached for a certain duration (few minutes). Else, the transmission is denied, and the cellular device does not advertise its availability on the Wi-Fi network.

We introduce two new software components. The first one runs on the cellular device and performs the following tasks: i) implements a proxy that pipes incoming connections through the 3G network, and ii) advertises the device availability through a discovery protocol like Bonjour only if the device has an active permission by the cellular network. The second one runs on the device that needs augmentation; referred to as the “client”,

and i) implements a proxy that waits for incoming connections from the applications to be augmented, ii) builds the set of admissible cellular devices (denoted by Φ) by discovering them on the Wi-Fi network, (iii) uses the GW plus the set of admissible cellular devices to transfer required contents using a multipath scheduler. The job of the scheduler is to handle multiple downloads and uploads of generic items over multiple paths. We consider the following setting: we have N available paths, with $N-1$ corresponding to Φ, and M items to download/upload, from/to a given server. We refer to the action of downloading/uploading the set of M items a transaction. The scheduler goal is to transfer the full set of M items as fast as possible, i.e., to minimize the total transfer time. In Sec. 4.5 we describe how we change the software components to handle volume caps, in the case where the network operators providing wired and cellular services are different.

4.2 3GOL: where and when

In the previous section, we described the context for 3GOL and outlined a high-level description of the system architecture, where 3GOL is provided as a network integrated solution. Even if 3GOL is offered by the operator that provides both wired and cellular services, we still need to understand where and when 3GOL can be used. We conduct active measurements to answer these questions.

Name	Description
3G web traffic	HTTP traffic logs for one large cellular network operated in western European country for 24 hr period, Oct 2011, millions of users
MNO	Per-user monthly data demand generated by nearly one million users in a Mobile Network Operator.
DSLAM	Flow level information for all subscribers connected to one DSLAM in a major European city. The coverage area includes 18,000 DSL connections
Handset experiments	Downlink and Uplink throughput experiments with 10 high end smartphones in six different locations in one European city across five days

Table 4.1. Data sources used.

Where? Location matters: The most common cellular technology deployed in Europe is UMTS/HSPA. Both the HS-DSCH downlink channel (HSDPA) and the E-DCH uplink channel (HSUPA) are best-effort technologies. Their capacity is shared between users through scheduling policies aimed at optimizing the channel allocation given the available radio conditions. Hence, the per user throughput at the base-station level depends on the highly variable radio conditions as well as on the number of users simultaneously accessing the channels. To glean the available bandwidth under real operating

conditions in a metropolitan city, we program 10 high-end terminal devices[4] to download and upload 2 MB files from/to a server via the cellular interface. The description is in Table 4.1, where we also describe the other datasets we use. We perform throughput measurements in a western European city at six different locations with these devices assisting the existing residential broadband connection. One would expect that the degree of augmentation will be higher for residential/rural areas and during the night. We select locations and timing to test this assumption.

Given that cellular networks have middleboxes [85] and dynamic IP address assignment, we use `wget` to estimate downlink bandwidth using a random 2 MB file to avoid biases added by proxies with compression capabilities. Uplink bandwidth is tested with `iperf` [86], although we also use `netperf` [87] for validation, obtaining similar results. We first activate one 3G device and perform uplink/downlink throughput measurements. We repeat such measurements four times in sequence. Every 20 minutes we introduce a new device and run the same measurements for all active devices in parallel. All devices are synchronized using NTP. We carefully select the size of the file and the number of repetitions to collect measurements in all locations without exceeding the data plan cap (10 GB/month) to avoid throttling. For each experiment, we also record the base station serving each smartphone as well as the DSL speed, that we repeatedly measure using `http://www.speedtest.com`.

Location	Time	DSL Mbps (d/u)	3G Mbps (d/u)	3GOL/DSL (d/u)
1. Densely populated residential area (city center)	1 a.m	3.44/0.30	5.73/3.58	2.67/12.93
2. Office area at rush hour	4 p.m	4.51/0.47	2.94/1.52	1.65/4.23
3. Residential area in tourist hotspot	10 p.m	6.72/0.84	2.08/1.29	1.31/2.54
4. Sparsely populated residential area (suburbs)	1 a.m	2.84/0.45	4.55/2.17	2.60/5.82
5. Densely populated residential area (city center)	n/a	8.57/0.63	3.88/2.63	1.45/5.17
6. Densely populated residential area (city center)	n/a	55.48/11.35	2.32/1.52	1.04/1.14

Table 4.2. Description of the locations we measure with comparison between the DSL and the 3GOL (DSL + 3G) throughput when using three devices.

Fig. 4.4 shows the aggregate 3G throughput as a function of the number of active devices for the first four locations at a given time of the day (Table 4.2). This analysis extends previous studies of cellular networks capacity based on individual measurements [76] as it overloads the base stations serving a specific location with multiple devices accessing the channel simultaneously. *Wired capacity can be augmented by up to* 14 Mbps *in the downlink and* 10 Mbps *in the uplink for some locations*. In some of them, even when the measurements are taken at peak hour (e.g. Location 2 at 4 pm), the cellular network seems to be well provisioned, leading to a linear increase in downlink throughput up to ten devices. For two devices we obtain an augmentation around 4.8 Mbps, median. However, the behavior for the uplink is significantly different. For three out of the four

[4]Samsung Galaxy S II with MIMO HSDPA Category 20 and HSUPA Category 6.

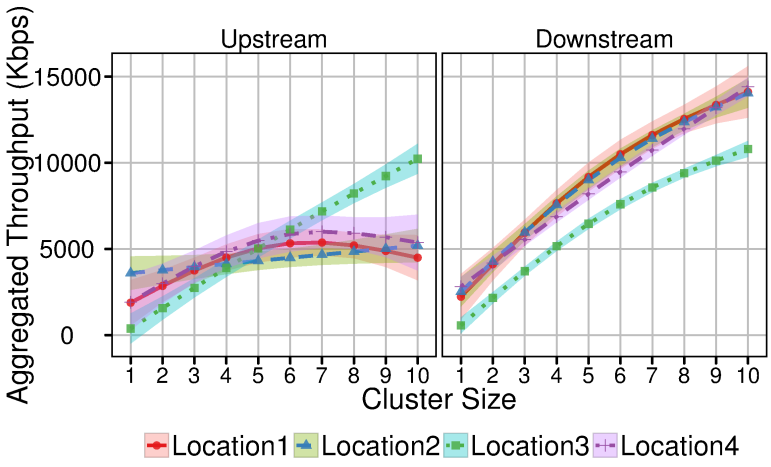


Figure 4.4. Aggregated throughput (uplink and downlink) for up to ten devices.

locations, we observe a clear plateau in aggregation at five devices, nearly equal to 5 Mbps which is the maximum capacity for HSUPA (5.76 Mbps). However we note that Location 3 exceeds 5 Mbps in the uplink as shown in Fig. 4.4 also when the devices are connected to the same base station. It is possible that the devices are connected to different sectors of the same base station, allowing a cluster of devices to load balance naturally on a cell basis. [5]. Such a scenario is plausible since Location 3 is a hub for tourists with a large density of cellular infrastructure.

When? For the network: To understand the temporal evolution of the cellular capacity, we programmed five of the devices to download and upload the same file in groups of five, three and one device at every hour over five days. We start downloads and uploads at ten past and twenty past each hour respectively. The results, illustrated in Fig. 4.5, indicate that *the throughput that can be provided to a single device can be up to* 2.5 Mbps *both for the downlink and for uplink depending on the hour of the day*. However, as the number of devices sharing the channel increases, we observe a higher per device throughput variability. With five devices, and across all locations, the uplink and downlink per device throughput can vary from 0.65 to 1.12 Mbps and from 0.77 to 1.42 Mbps at 2 pm and 2 am respectively. While the diurnal throughput variations do exist, they are rather small, and indicate low congestion levels in most of the studied locations, even in the peak-hours.

Load balancing: The violin plots[6] in Fig. 4.6 show the distribution of the capacity provided to one device by the different base stations covering the areas under study. The

[5]Unfortunately, our devices can only report the base station they connect to but not the sector which is a known information on the provider side.

[6]The violin plot is similar to box plots, except that they also show the probability density of the data at different values (in the simplest case this could be a histogram).

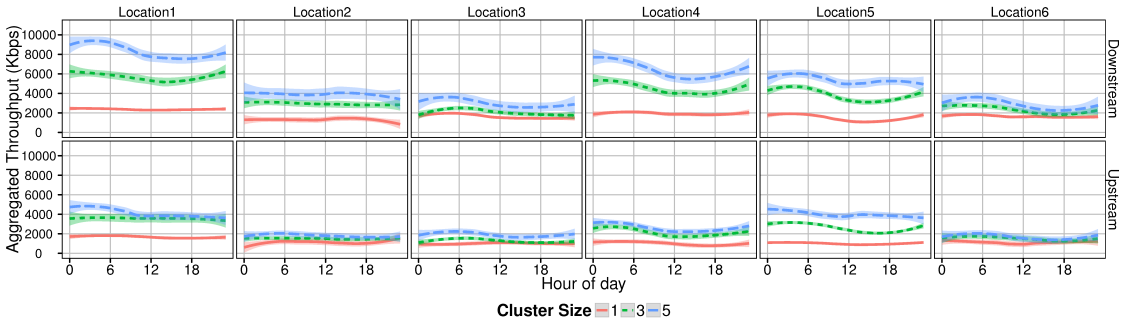


Figure 4.5. Aggregated throughput (downlink and uplink) in six different locations.

solid lines indicate the usual average capacity of dedicated UMTS channels under good radio conditions[7]: 360 kbps and 64 kbps in the downlink and the uplink, respectively. Any additional capacity above these thresholds comes from the shared HSDPA or HSUPA channels. From the violin plots we observe that a base station can provide throughput that varies from 0.7 to 2.5 Mbps both in the uplink and downlink. In Table 4.3 we report the average, maximum and the standard deviation of the per device throughput obtained by a base station, noting that it decreases with the number of grouped devices both in the downlink and in the uplink.

We note that devices are associated with at least two different base stations at all locations. This suggests that when there is a denser 3G deployment, *load can be balanced across different base stations in the network, hence enabling to obtain a throughput beyond the capacity limit of a single base station*.

Cluster Size	Uplink (mean/max/sd)	Downlink (mean/max/sd)
1	1.09/2.32/0.72 Mbps	1.61/2.65/0.57 Mbps
3	0.90/2.47/0.60 Mbps	1.33/2.32/0.51 Mbps
5	0.65/2.44/0.50 Mbps	1.16/3.44/0.56 Mbps

Table 4.3. Average, maximum, and standard deviation of the per device throughput of a HSPA base station for the different groupings.

Main Takeaways: Our results indicate that it is possible to achieve significant benefits both in the uplink and downlink even with a small number of 3G devices, at various geographical locations. The majority of uplink throughput augmentation can be obtained with 2-3 devices during off-peak times, whereas downlink throughput seems to scale up better. We also find that such benefits are pronounced in rural areas, but not negligible in urban areas. In fact, *when we contrast the throughput of 3GOL with that of wired broadband, we find a speedup of* $\times 1 - 3$ *downlink and* $\times 1 - 13$ *uplink even when using*

[7]UMTS capacity can be lower in case the device is having poor coverage.

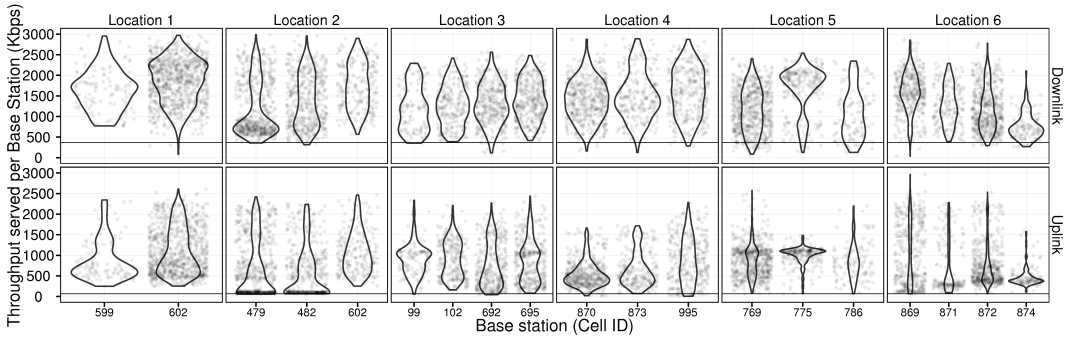


Figure 4.6. Throughput served per base station (downlink and uplink) in six different locations over five days.

just 3 devices (Table 4.2). Furthermore, these gains are observed not only for rural but also urban areas, and even during peak hours. Even at overloaded locations at peak hour, *it is possible to augment by 30% the downlink capacity and double the uplink one*.

4.3 3GOL implementation

In the previous sections, we have analyzed the feasibility of 3GOL using active measurements. In this section, we describe the implementation of the architecture for 3GOL (Sec. 4.3).

4.3.1 System Implementation

We implement the architecture shown in Fig. 4.3 under the worst case scenario: we consider all devices connected to the gateway (GW) through the local Wi-Fi. In this way we expose our system to packet losses due to the wireless channel, which is also affected by interference generated by co-located overlapping Wi-Fi networks. In the best case, clients could be wired and cellular devices could be docked to the GW. Hence we upper bound the backhaul aggregation to the local Wi-Fi maximum goodput, which for TCP flows is around 24 Mbps and 110 Mbps for 802.11g and 802.11n respectively[8]. We do not examine the energy related issues of 3GOL. 3GOL is not likely to be continuously used over long periods of time, and also 3GOL devices are often connected for recharging while at home, hence energy consumption is not a primary concern in the 3GOL design space.

Given that most residential traffic today is HTTP [88], we select two HTTP applications to augment via 3GOL: Video-on-Demand (VoD) for the downlink and picture upload for the uplink case. We choose a VoD application that uses Apple's HTTP Live Streaming

[8]The maximum goodput may decrease according to hardware quality, channel conditions, number of transmitting stations, average bitrate, average payload size used in the BSS

(HLS) protocol [89] that is supported by all Apple devices, including the Apple TV, and is also implemented in other video players (Android $\geq$ 4.0, VLC $\geq$ 2.0).

HLS is similar to Dynamic Adaptive Streaming over HTTP (DASH) [90] and divides the video in segments of short size that are separately requested by the player with one HTTP GET request for each segment. The list of segments is retrieved through an extended M3U (m3u8) playlist and this is the first downloaded element. Next, the player sequentially requests the segments, one at a time, in the same order in which they will be required by the decoder. The video starts after a pre-buffering phase which is application dependent. We study the amount of time that 3GOL takes to fill the player pre-buffer as this gives the startup waiting time for the user.

For the uplink case we select to upload a set of pictures because it is a common use-case in social networks like Facebook, as well as in specific photo sharing applications such as Flickr and Picasa. All native clients of the aforementioned applications use multi-part HTTP POST request to upload the pictures and they perform sequential uploads, one file at a time.

We implement the mobile component as an Android application that includes a basic HTTP proxy to serve the requests coming from the Wi-Fi using the 3G interface. Hence, the device must be connected both to the Wi-Fi network and to the 3G network as a client. This operation mode, that is different from the commonly implemented tethering, is not natively supported by Android and we had to root the phones to enable it. Instead, the VPN API introduced from Android version 4, could be used to perform route manipulation on the flight. We implement an HLS aware HTTP Proxy, an HTTP uploader, and a multipath-scheduler in the client component that we detail next. The client component intercepts the extended M3U (m3u8) playlist, and using the scheduler it pre-fetches the segments by performing parallel downloads. Also the HTTP uploader uses the scheduler to perform parallel multi-part POST requests to upload a set of selected picture on a web server. Both client components can easily be installed by the user as they do not require the server to change neither specific drivers or protocols. Furthermore, this implementation is completely transparent to the residential gateway.

Scheduler

Since we are dealing with devices having different capacities, bandwidth variability over time, TCP flows over 802.11 and HSPA channel allocation schemes, the application level goodput of each path could vary in time, especially for paths over the cellular network. A first option is to use a round-robin scheduling policy. However, the peak capacity of the ADSL link is generally very different from the peak capacity of HSPA and hence round-robin cannot be expected to maximize gains. One can use history to predict future conditions and to schedule. However, estimating available capacity under rapidly changing network conditions can result in inaccurate estimates [91].

Hence we design a simple greedy scheduling algorithm that can be described as follows. First, an item is assigned to each path. Then, if there are any remaining items ($M \geq N$), they are scheduled by order, on the first available path. It is easy to see that this simple greedy scheduling policy leads to all paths being busy, achieving maximum throughput and minimum download/upload time. Optimal performance will be achieved if all paths end at the same time, which is very unlikely to happen in our case. Hence, when all items have been already scheduled and a path becomes idle before the transaction is completed, we reassign the oldest scheduled item among the ones being transfered by the other $N - 1$ paths. We keep doing this until the transaction ends, i.e., until all the items have been transferred. The amount of bytes wasted is upper bounded by $(N-1)S_m$, where S_m is the maximum item size. However, this waste is generally much smaller than this upper bound as when a rescheduled item completes, all other ongoing transfers of that item are aborted.

We could modify the scheduler to cover also the playout phase, but given the wide amount of proposals in this area [92], we leave this extension as future work.

4.4 Evaluation

We present results of evaluating a prototype implementation of 3GOL as described in Sec. 4.3.1. We first look at the general performance of the scheduling algorithm, and then look more specifically on the applications described in Sec. 4.3.1 by evaluating them "in the wild" at different residential locations. The 3GOL setup we use in all evaluations consists of a notebook client running Ubuntu 10.04 LTS, and up to two Android 2.3 smartphones[9] and we do not introduce any change in the residential gateway. We use 802.11n compliant devices. Furthermore, we use a dedicated well provisioned web server, featuring a stable bandwidth of 100 Mbps in download and 40 Mbps in upload, and we disable caching on the content we transfer.

4.4.1 Scheduler Performance

We compare our greedy scheduler (GRD) with two common schedulers: round robin (RR) and minimum time scheduler (MIN). The round robin scheduler cyclically assigns one item to each path, while the minimum time scheduler assigns the items to the path that minimizes the estimated transfer time, computed by using the estimated available bandwidth of each path. For the MIN scheduler we assign the first N items in a round robin fashion to initialize and then estimate the bandwidth using exponential smoothing filtering. We set the filter parameter to 0.75 to maintain a high level of agility.

First we setup a 3GOL test environment using an ADSL line at 2 Mbps downlink and 0.512 Mbps uplink. We measure the average download time of an HLS video with

[9] Same devices, Galaxy S2, used in Sec. 4.2.

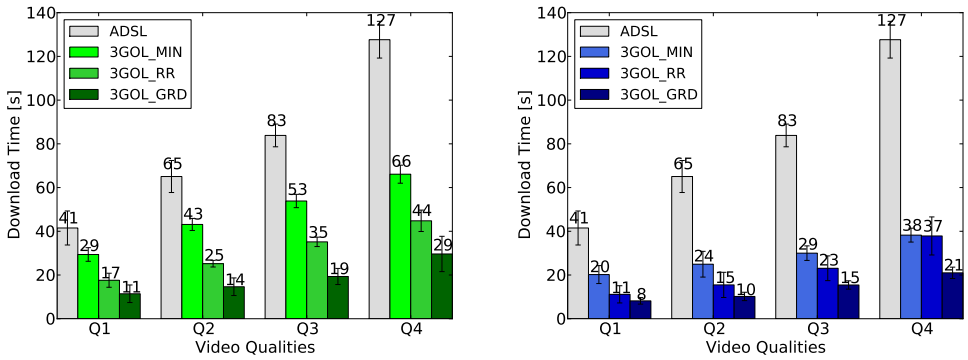
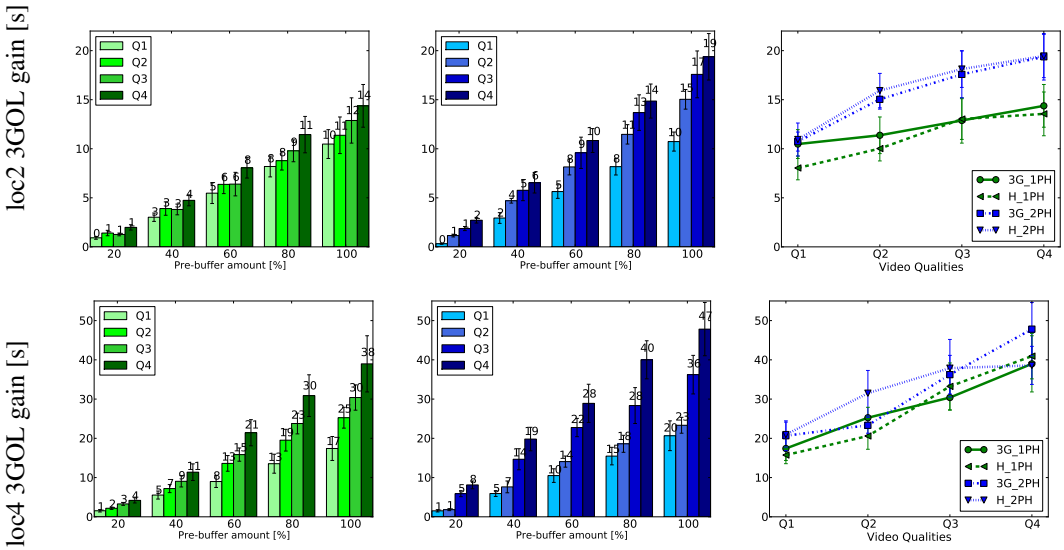


Figure 4.7. Scheduler comparison in downloading an HLS video lasting 200 s with a 2 Mbps ADSL using one device (left) and two devices (right).



(a) 1PH 3G (b) 2PH 3G (c) 100% download

Figure 4.8. Video 3GOL performance comparison between location with fastest (loc2) and slowest (loc4) ADSL, using one (1PH) or two (2PH) phones, and starting from IDLE (3G) or from a connected mode (H).

different resolutions. The video sequence is the sample HLS video provided by Apple (bipbop). We do not change the segmentation that was set to 10 seconds of video for each segment but we set the video duration to 200 s, which is the median video length of a You-Tube video [93]. We also keep the original qualities of the video (Q1=200 kbps, Q2=311,kbps, Q3=484,kbps, Q4=738,kbps) as they reflect commonly used bitrates according to the same study. Since the aforementioned qualities are lower than the ADSL speed, we are able to play the video without pauses. For each video quality we run 30

repetitions using first the ADSL alone, and then enabling 3GOL with each scheduler, one at a time. We repeat this cycle using both one and two phones. In order to minimize bandwidth fluctuations both in the Wi-Fi and in the 3G network, we start the scheduler comparison at night (1:00 am) In Fig. 4.7 we plot the results obtained, reporting both the average and the standard deviation for each case. For all video qualities we observe the same trend: the greedy scheduler is the best one, followed by the RR and the MIN scheduler. The MIN scheduler performs the worst due to the high variability of the available bandwidth of phones. The high variability of channel conditions results in poor estimates, leading to suboptimal decisions. Changing filter and/or sampling criteria was not helpful in improving the performance of the MIN scheduler. As expected, the benefit does not linearly scale with the number of phones used for the same reasons mentioned in Sec. 4.2

Location	DSL (d/u)	3G Signal (dBm/ASU)
loc1	6.48/0.83 Mbps	-81/16
loc2	21.64/2.77 Mbps	-95/9
loc3	8.67/0.62 Mbps	-97/8
loc4	6.20/0.65 Mbps	-89/12
loc5	6.82/0.58 Mbps	-89/12

Table 4.4. Measured ADSL speed (downlink/uplink) and 3G signal strength of all locations evaluated with 3GOL.

4.4.2 3GOL evaluation in the wild

We perform extensive evaluations of 3GOL at five different residential locations, both for video download and for photo uploads. Since we want to extend our sample points, we consider a different set of locations from the ones reported in Sec. 4.2 in the same city. We report in Table 4.4 the ADSL speeds, repeatedly measured as in Sec. 4.2 before starting each evaluation, as well as the signal strength measured for each location. We measure total upload/download time achieved with and without 3GOL, and for VoD we also study the pre-buffering time, i.e., the measured delay from the initial request of the video to the first frame displayed by the player. As the amount of pre-buffering is application dependent, we vary it throughout the experiments.

We first run the experiment with one cellular device then with two devices, each time starting with devices from idle (3G) first and then from a connected mode (H)[10]. For VoD, we vary the video quality from Q1 to Q4 in each case and for each quality we vary the pre-buffer amount from 20% up to 100% of the video length, which is the inner most loop. For each setting we repeat the experiment 30 times, first running it without 3GOL and then enabling it using the greedy scheduler. On average, the duration of both the

[10]To bring the phones into a connected mode (FACH or DCH) we issued a train of ICMP packets spaced at 0.1 s before starting the 3GOL transaction, interrupting them right afterwards.

download experiment and of upload one was approximately 12 h, depending on both ADSL and device speeds. In all locations, we run both download and upload during weekdays at daytimes, starting each one around 9.00 am. Note that we evaluate 3GOL under worst case conditions according to the findings reported in Sec. 4.2. Fig. 4.8 shows 3GOL gains, computed as the reduction in seconds with respect to the ADSL alone, at the fastest and the slowest location. The former corresponds to the location with the fastest ADSL (loc 2 with 21.64 Mbps), the latter to the location having the slowest one (loc4 with 6.2 Mbps downlink).

We observe that 3GOL gain increases with video quality as well as with pre-buffer amount. Indeed, both the number of segments and the size of each segment amplify the benefits brought by 3GOL, when increased. As we test 3GOL starting from very few pre-buffering segments (min. 4, max. 20) as well as from small segment size (min. 0.2 MB, max 0.95 MB), the observed trend suggests that 3GOL could always be beneficial for HLS and similar applications. From Fig. 4.8 it is evident that the addition of a second device does improve the overall performances, up to +35% of the best gain with 1 phone for location 4 and up to +26% for location 2. However, the obtained gains are high with even one device, leading to a reduction of 14 s for location 4 and 38 s for location 2.

In Fig. 7 we show the performance improvement when devices start from a connected state. The expected performance boost, due to the absence of the channel acquisition delay, does not clearly appear. This is partly due to the higher variability of environmental conditions. However, it can be noted that when present this gain is limited and it shrinks as the transaction duration increases. We can infer then, that the channel acquisition delay has little or no bearing on the performance of 3GOL for this application. Considering all pre-buffer setting from 20% to 80% across all locations, corresponding to an average and maximum speedup of $\times 2.1$ and $\times 3.8$ respectively, we obtain an average transaction time reduction of 47%.

Fig. 4.9 shows the reduction achieved by 3GOL in downloading the entire video for all locations, where each point is the average of the percentage reduction across all video qualities. Reductions span from a minimum of 38% up to a maximum of 72%, corresponding to a speedup ranging from a minimum of $\times 1.5$ up to a maximum of $\times 4.1$. We can see the benefit of adding the second device and of starting with devices on a connected mode. The second device always aids in increasing download time reductions, ranging from +5.9% up to +26% with respect to the case where only one device was used, while starting from a connected mode mostly brings marginal gains.

Upload:

We evaluate upload at the same five locations and using the same methodology. We repeatedly upload a set of 30 pictures with average size of 2.5 MB and standard deviation of 0.74 MB. We obtain these values from a set of 200 pictures taken with iPhone 5 and iPhone 4S, that are the most used devices for uploading photos to Flickr. Compared to the download case, we increase the number of items (from 20 to 30) we transfer, as well as their size (from a maximum of 0.9 MB up to a maximum of 3.9 MB).

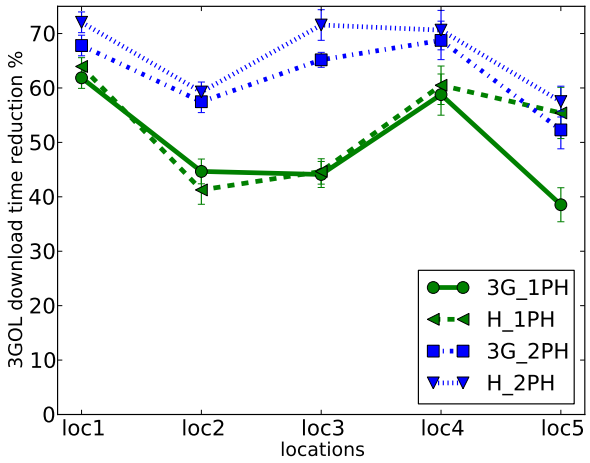


Figure 4.9. 3GOL % total video download time reduction in using one (1PH) and two phones (2PH) starting from idle (3G) and from a connected mode (H).

The limited ADSL uplink bandwidth in our locations range between 0.58 Mbps and 2.77 Mbps resulting in a huge 3GOL gain. Using one device the total upload time is reduced from 31% up to 75%, corresponding to a speedup between $\times 1.5$ and $\times 4.0$, while using two devices the upload times decrease from 54% up to 84%, corresponding to a speedup between $\times 2.2$ and $\times 6.2$.

Also in this case, gains are not directly proportional to the number of devices used, and one device is enough to achieve high gains. Fig. 4.10 shows the average total upload time obtained with ADSL and with 3GOL, both using one and two device starting from idle.

Clientless solution

3GOL can achieve high gains by using only one device, specially when applied to uploads. This is due to the shared nature of the wireless medium that has implications on both the in-home Wi-Fi and on the 3G network, as well as the asymmetric capacity of the ADSL that results in a very limited uplink throughput. If we restrict the use of 3GOL to upload applications, we can imagine a use case in which users exploit their own devices in order to boost upload of contents by installing a single application on the phone. This over the top "clientless solution" can ease the deployment of 3GOL because it requires the user to install only one new application. Such application must be accessible from the client (CL) after authentication, i.e., with the client browser, and boost user uploads using the 3GOL multipath-scheduler. As clearly shown in Fig. 4.11, despite the advantages of this solution, there is an inefficiency: the items must be transfered to the device first and then scheduled from there. As the transfer from the client to the device is not limited by any backhaul capacity, it will last only for a first portion of the transaction, the duration of which depends on Wi-Fi performance. Similarly with previous

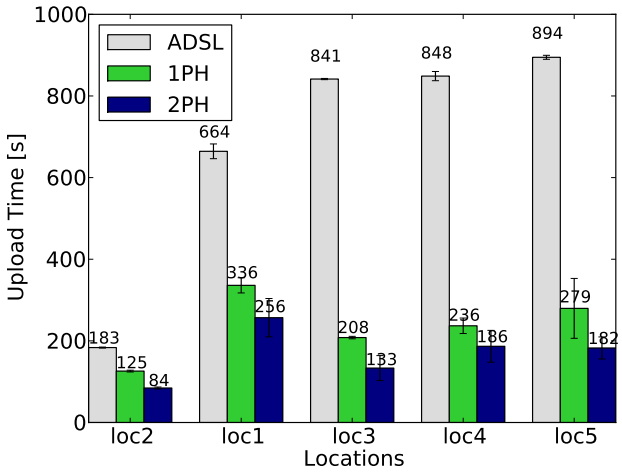


Figure 4.10. ADSL vs 3GOL upload time using 1 and 2 device starting from idle.

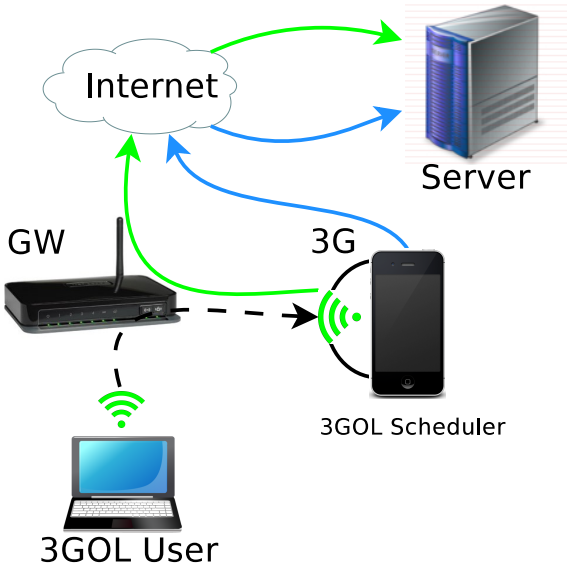


Figure 4.11. Clientless solution architecture with one device used for upload applications.

evaluations, we compare the client-based (CB) and the clientless (CL) solution, running the experiment in our testbed with an ADSL line having a 0.89 Mbps uplink capacity and one device having a signal strength of −89 dbM/12 ASU. We obtain similar performances with the CL solution performing 3% less than the CB solution, on average. This confirms the validity of the CL deployment, which could be easily adopted by users.

Using MTCP

To exploit the bandwidth on multiple paths one can use MTCP [77] as well, although then both client and server need to support it. This requires the presence of a third party

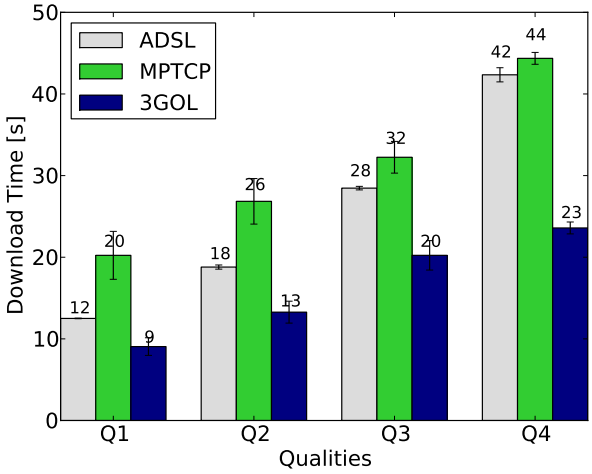


Figure 4.12. 3GOL with MTCP performance

server that handles MP-TCP flow for legacy TCP stack. We compare the performances of MTCP versus 3GOL with an ADSL line at 4 Mbps downlink, 0.512 Mbps uplink, and using one phone. In both cases the server has one 100 MB ethernet interface and it is located in the same network as the wireless client. For both client and server, we use the default configuration included in kernel 3.5.7.wheezymptcp for MTCP while we use a normal Ubuntu 12.04 distribution for both client and server for 3GOL. We evaluate the download time of a video sequence of 200 s at different qualities, repeating the experiment 10 times in each configuration. In the MTCP case, the phone was connected to the client via USB and it was set to the standard mode USB tethering. The performance achieved with MTCP is worst than using ADSL alone, and this is due to the Coupled Congestion Control (CCC) algorithm of MTCP that is not optimized for wireless use. Indeed, one TCP flow alone is able to achieve higher throughput in this case, while 3GOL is able to exploit the additional capacity available on the cellular network, as shown in Fig. 4.12, outperforming MTCP by $\times 2.3$. Furthermore, even if the CCC will be optimized in the future, a solution based on MTCP still need a specific protocol stack, which requires *both* clients and servers to support MTCP. Conversely, 3GOL is an application level solution that is readily deployable.

However, with the arrival of Android version of MP-TCP stack[11] as well as on Apple iOS7 [12], we hope to revisit and hopefully obtain better performance.

[11] `https://github.com/mptcp-galaxys2`

[12] `http://perso.uclouvain.be/olivier.bonaventure/blog/html/2013/09/18/mptcp.html`

4.5 Dealing with multiple providers

Up to this section, we have been looking into 3GOL as a network integrated service, offered by a provider with both cellular and wired footprint. The fundamental technology behind 3GOL is, however, not limited to such a scenario since it does not require explicit cooperation between the two networks beyond billing for cellular data discounting. In this section, we explore the scenario with different cellular and wired operators. In such a case, one needs to be careful using 3G data in order to avoid penalties associated with exceeding the enforced cellular data plans [94]. Second, due to caps, users may find it difficult to allocate volume to 3GOL and may find it burdensome, leading to 'price of uncertainty' [95]. Third, even if users have volume they can use towards 3GOL, we need to make sure the load 3GOL puts on the cellular network does not hurt the primary 3G service. We deal with these issues next.

Do users have spare volume under the cap?

We analyzed data collected from a medium sized Mobile Network Operator (MNO) (Table 4.1). The data set contains mobile data sessions by all mobile broadband customers. In Fig. 4.13 we show the distribution of data usage as a fraction of the contracted cap. We find that 40% of the customers use less than 10% of their cap, and 75% of the customers use less than 50%. On average, it would appear that 20 MB of volume can be used per device per day by 3GOL. Similar observations about users having disposable volume have been reported elsewhere [96].

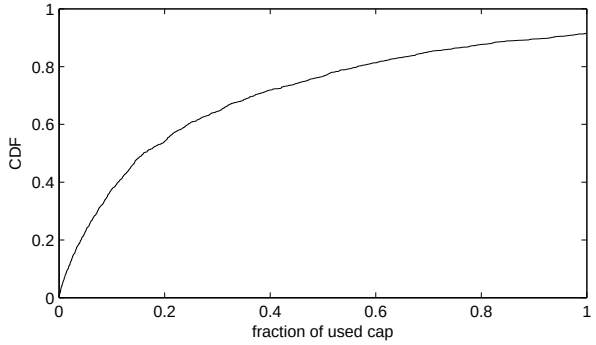
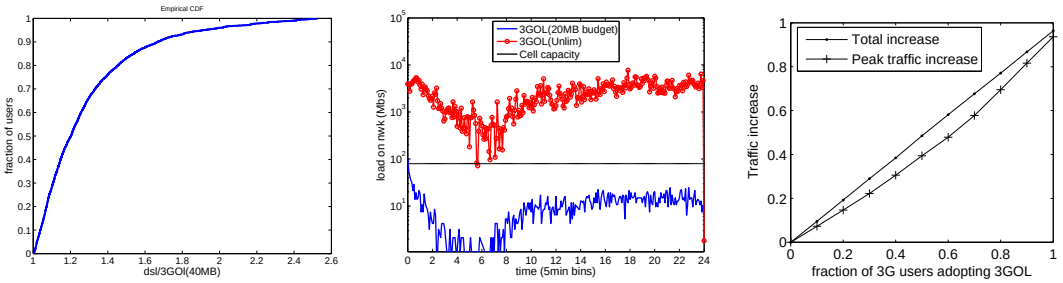


Figure 4.13. The distribution of fraction of used cap. 40% of customers use less than 10% of their cap. 75% of customers use less than 50% of the cap.

How to allocate volume towards 3GOL?

In order to alleviate the burden of the user choosing the allowance to be used for 3GOL, we propose a simple estimator with a 'guard' parameter that can minimize volume cap overruns. To compute the suggested 3GOL allowances $3GOL_a(t)$ we use a simple technique that estimates the free capacity at month t of user u. We compute the average



(a) (b) (c)

Figure 4.14. CDFs of improvement of 3GOL in terms of latency per video: (a) budgeted with 40 MB daily allowance, (b) Load on cellular network by using 3GOL (budgeted) and 3GOL (unbudgeted). Y-Axis is logscale. Solid horizontal line is total backhaul capacity (2 x 40 Mbps), (c) Relative 3G traffic (total and peak-hour) increase due to 3GOL as a function of the fraction of users adopting 3GOL.

free capacity over τ months prior to t:

$$\bar{F}_u(t) = \frac{\sum_{s=1}^{\tau} F_u(t-i)}{\tau}.$$

To minimize cap overruns, we set $3GOL_a(t)$ to the estimated free capacity $\bar{F}_u(t)$ discounted by a guard/margin:

$$3GOL_a(t) = \bar{F}_u(t) - \alpha\bar{\sigma}_u(t),$$

where $\bar{\sigma}_u(t)$ is the sample standard variation of the free capacity during τ months prior to t and α is a tunable guard parameter. By running this estimator on the MNO dataset, we find that using $\tau = 5$ and choosing $\alpha = 4$ allows around 65% of the available free capacity to be used by 3GOL with expected overrun time of under 1 day per month overall. Hence, there is room for exploiting the free/unused 3G capacity with no or very little disruption of the primary 3G service, for the user. We consider the load on the network next.

We can easily modify the architecture presented in Fig. 4.3 to handle caps. The component running on the cellular device can track 3GOL data usage $U(t)$ and estimate the 3GOL allowance $3GOL_a(t)$. If the available quota $A(t) = 3GOL_a(t) - U(t)$ is greater than zero, the device advertises itself. All devices that advertise themselves, become part of the admissible set Φ. Thus, we need no input from the network.

Load on the cellular network

When users operate under caps, we first need to check if users can still see a tangible benefit. Further we need to ensure that the cellular network does not get overloaded, even if caps are being respected.

We use trace-driven analysis to quantify the benefits even under caps. The trace we study is collected from a DSLAM in a major metropolitan city over 24 hours in April

2011. The ADSL speeds were 3 Mbps. The data contains HTTP transactions, (userid, time of request (UTC), URL) from which we separate out all sessions pertaining to video. We find that 68% of all users in the dataset have seen at least one video. The data contains the HTTP response headers, from where we can obtain the size of the video files requested. We note that users may not watch the entire video that is requested, however our choice of using the size of the entire file is *conservative* as we shall demonstrate.

In our dataset, we find that an end user [13] views 14.12 videos (mean, 6 median, 30.13 std) per day. Therefore, if we aim to accelerate *e*very video by even 20 MB (as obtained above) would lead to a daily consumption of 280 MB for the 3G data plan, thus being clearly unacceptable. Thus, we focus on using 20 MB per device per day that corresponds to 600 MB per device per month, the average amount of free/unused capacity in the MNO dataset (Table 4.1). With two devices, this gives an available daily 3GOL budget of 40 MB, that we use to boost any/all videos the end-user views. We plot the savings *per-user* over using just DSL ($\frac{DSL\ latency}{3GOL\ latency\ under\ budgets}$) in Fig. 4.14 (a). We find that 50% of the users can see at least 20% speedup, while 5% of the users can see a speedup of 2. We note here that our results are conservative. Indeed, we can accelerate via 3GOL only the bytes required to fill the playout buffer, achieving an earlier playout for the user.

There is no fixed size of the playout buffer but the general consensus is that it is an order of magnitude smaller than the file size. If the average size of a video is 50 MB [93], and the playout buffer is 5 MB, then 3GOL can speed up 8 videos per day per user—a tangible benefit. From Sec. 4.2, we saw an increase in capacity with more devices, leading to even higher gains. Likewise, with LTE deployed, we can expect much higher benefit to the end user, *even* with budgets in place.

Now we turn to quantifying the resulting impact to the 3G network in terms of load. Using the same DSLAM dataset, we compute the amount of traffic that would be "onloaded" to the 3G network if we were to accelerate the *first* video that could benefit from 3GOL (with a size greater than 750 KB, that would require more than 2 seconds on DSL). We further assume we have two 3G devices (with HSPA+) that can be used to accelerate up to 20 MB each per day. The represented geographical area would typically be covered with 2 towers (Sec. 4.1.1).

In Fig. 4.14 (b) we show the traffic onloaded to the cellular network and compare it with the backhaul capacity of the cellular deployment that could cover a similar geographical area (2 base stations with 40 Mbps backhaul). We note that if 3GOL would operate without caps, then the 3G network will be guaranteed to be overloaded. However, when 3GOL operates within caps then the additional load introduced on the 3G network could be reasonable. We find that on average, a user would onload 29.78 MB per day, if operating with caps and using two devices. Note again that our results are conservative since we accelerate the entire video transfer, whereas prior work has demonstrated that only 10% of the whole file is downloaded [93].

[13] a DSL "subscriber"

In the previous analysis, we did not consider the existing load on the mobile network. We also focused on one application – video-on-demand. Taking this further, we evaluate how much traffic will be added to the *existing* load of the cellular network in function of the fraction of users adopting 3GOL. As before, we consider a 20 MB daily allowance per user. Using the MNO dataset (Table 4.1), we plot (Fig. 4.14) the relative increase of traffic due to 3GOL, assuming a uniform distribution of 3GOL users the network customer base. We observe that the relative increase in aggregate traffic is relatively modest, when 3GOL adoption is low. In the case of 100% adoption, the increase in traffic is around 100%. Thus if all users adopt 3GOL consuming 20 MB per day via 3GOL, the 3G traffic is expected to double. This increase can be undesirable and hence in this case 3GOL should be provided as a network integrated service (we described such a service in Sec. 4.1).

If we assume that 3GOL demands are generated based on the DSLAM flow-level trace, as described above, then if we take into account the non-alignment of peaks from Fig. 4.1, the superposition of existing 3G traffic and 3GOL traffic onloaded will benefit from this non-alignment. The relative traffic increase due to 3GOL in peak-hour is smaller than the relative increase of the aggregate traffic, albeit the difference between the two is rather small; see Fig. 4.14.

We remark that by having a network integrated solution, described in Sec. 4.1, 3GOL traffic will not harm existing 3G traffic. The previous analysis shows that as long as the 3GOL traffic remains below the threshold set by existing leftover capacity, there will be a modest traffic increase, as we are using already-paid-for capacity. However, we argue that, as with latest versions of peer-to-peer [97], OTT services have the incentive to be friendly with the network and avoid tension that could lead to rate-limiting or blocking. We therefore expect that successful OTT realizations of 3GOL will either self-moderate their consumption or reach out to 3G providers for integration.

4.6 Final remarks on 3GOL

We have proposed 3G Onloading [98], a strategy that uses the cellular network to improve application performance in the home, when constrained by the wired network. Our contribution is in studying the feasibility of 3G Onloading, in terms of technical and economic factors. Using multiple data sets, and active experiments in urban and rural locations, we have demonstrated that the benefits to application performance could be significant with reasonable overhead to the cellular infrastructure. Such overhead could further be minimal if 3GOL was rolled out as a network integrated service, offered only when the cellular infrastructure is lightly utilized. We develop a 3GOL prototype [99, 100] and test the performance in the wild. We find x4 and x6 speedups to applications needing downlink and uplink.

Chapter 5

Conclusions

IN THIS THESIS we proposed two cooperation strategies for enhancing user connectivity at home.

The first one is based on the creation of a Federated Network, where neighboring GWs can federate among each other with the purpose of sharing their backhaul connections and their local wireless resources. In order to achieve this vision, we proposed a new available bandwidth estimation (ABE) technique for 802.11 BSSes that accounts for auto-rate adaptation algorithms, different type of traffic as well as the impact of interferers. Extending previous theoretical results on the saturation throughput S, we introduced a new estimation technique for interferers in the ISM band that we validated in a real test-bed achieving high accuracy. We formulated ABE in order to rely only on passive measuring techniques and without requiring modifications on 802.11 nor support from the Wireless Stations (WSs). We divided the formulation in intra-GW, an inter-GW available bandwidth estimation. The latter formulation is able to gauge the impact of incoming WSs with given traffic profiles on the BSS available bandwidth and it can be also used to implement admission control in 802.11. We implemented the passive measurements within the linux wireless driver, and both ABE formulations as normal user space programs, demonstrating the feasibility of our proposal on commodity hardware. Using the formulation of S and measuring the throughput at the GW, we proposed a new definition of load for 802.11 BSSes, accounting for the different nature of the traffic, i.e., distinguishing the flows between elastic (TCP like) and not elastic (UDP like). We used this new concept of load as a base for intra-GW and inter-GW load assessment and we implemented a wireless resource sharing protocol that optimizes the use of the wireless resources in order to achieve energy-efficiency while avoiding congestion. We evaluated our protocol with simulation, achieving around 60% of saving, and we implemented it on a small test-bed to demonstrate its feasibility with real applications and commodity hardware.

The second strategy does not require neighboring GWs to federate among each other, but it exploits in parallel both wired and wireless connectivities available within the home in order to enhance the throughput provided to the final user. Our system, that we named

3GOL, can strategically OnLoads traffic from the wired onto the wireless network in order to boost applications limited by the wired backhaul. Using active measurements we quantified the available bandwidth on the cellular network at different residential locations, finding that the cellular network can provide non negligible throughput, even when at peak and even in rural areas. Using trace analysis we evaluated the adoption of 3GOL on a broader scale for a single application, Video-On-Demand (VoD). We obtained that we can decrease the latency of video delivery by at least 20% for 50% of users. We also found that a network integrated solution is needed to avoid congestion, service degradation for the normal data services, and cap overrun. We implemented a 3GOL prototype, featuring a fully application level solution and a greedy multipath scheduler that we tested in the wild in several residential locations with ADSL, obtaining up to 4x speedups in downlink (video streaming) and up to 6x speedups for uplink (picture upload). We compared our multipath scheduler with other multipath scheduling schemes showing that our approach is faster than the other. We also tested transport layer solution like MPTCP, obtaining better performance by a factor of two. Furthermore, we found that using one 3G connection in parallel with the ADSL is enough to achieve remarkable boost, and that the acquisition delay of 3G does not have a big impact, which decreases with the chunk size and number. We remark that there is nothing that prevents 3GOL to be implemented also in an uncoordinated fashion.

In order to be commercially usable, our technologies should be implemented in a transparent fashion, while being coordinated by a management module able to act according to the network scenario and conditions. In dense environment, where overlapping coverages and wireless stations are abundant, the network federated approach could be the best choice. In this case the connectivity is enhanced without impacting the cellular network. Conversely, in rural or sparse areas, 3GOL could be used to overcome the limitations of the wired backhaul. We conclude our study by noting that both strategies could be combined together to provide users with the best possible Internet service while at home.

Chapter 6

Acronyms

3G Third-Generation Cell-Phone Technology

3GOL 3G Onloading

4G Fourth-Generation Cell-Phone Technology

AARF Automatic Auto Rate Fall-back

ACK ACKnowledgement

ADSL Asymmetric Digital Subscriber Line

AFH Adaptive Frequency Hopping

AP Access Point

ARQ automatic repeat request

BSS Basic Service Set

CBR Constrant Bit Rate

CDF Cumulative Distribution Function

CRC Cyclic Redundancy Check

CTS Clear To Send

DCF Distributed Coordination Function

DIFS Distribuited Inter Frame Space

DSL Digital Subscriber Line

DSLAM Digital Subscriber Line Access Multiplexer

EDCA Enhanced Distributed Channel Access

ESS Extended Service Set

EU European Union

FIGARO Future Internet Gateway-based Architecture for Residential netwOrks

FGK Federated Group Key

FTP File Transfer Protocol

GW Gateway

GOP group of pictures

HLS HTTP Live Streaming

HSPA High Speed Packet Access

HTTP HyperText Transport Protocol

IAPP Inter Access Point Protocol

IEEE Institute of Electrical and Electronic Engineers

ICT Information and Communication Technologies

IP Internet Protocol

ISM Industrial, Scientific and Medical

ISP Internet Service Provider

LAN Local Area Network

MAC Medium Access Control

MINLP Mixed Integer Non-Linear Problem

MIQCP Mixed Integer Quadratic Constrained Problem

MIP Mobile IP

MTCP Multipath TCP

NAT Network Address Translation

NTP Network Time Protocol

OS Operating System

P2P peer-to-peer

PC personal computer

PER Packet Error Rate

PMIP Proxy Mobile IP

PMKC Pairwise Master Key Caching

PoE Power over Ethernet

PPP Point to Point Protocol

PPPoE PPP over Ethernet

PHY PHYsical

PSNR Peak Signal-to-Noise Ratio

QoS Quality of Service

QoE Quality of Experience

SACK Selective ACK

SNR Signal to Noise Ratio

RR Round Robin

RFN Radio Federated Network

RTS Ready To Send

RTT Round Trip Time

SIFS Short Inter-Frame Spacing

SSID Service Set IDentification

TCP Transmission Control Protocol

TDMA Time Division Multiple Access

UDP User Datagram Protocol

VoD Video on Demand

VoIP Voice over Internet Protocol

WLAN Wireless LAN

WS Wireless Station

Bibliography

[1] EU FP7 FIGARO project, http://www.ict-figaro.eu/.

[2] Global e-Sustainability Initiative (GeSI), http://www.theclimategroup.org/assets/resources/publications/Smart2020Report.pdf.

[3] R. Bolla, R. Bruschi, K. Christensen, F. Cucchietti, F. Davoli, S. Singh, "The potential impact of green technologies in next generation wireline networks - is there room for energy savings optimization?," *IEEE Communications*, vol. 49, no. 8, 2011.

[4] N. Blefari Melazzi, D. Di Sorte, M. Femminella, G. Reali, "Toward an autonomic control of wireless access networks," *IEEE GLOBECOM*, 2005.

[5] A.P. Jardosh, K.N. Ramachandran, K.C. Almeroth, E. Belding, "Understanding congestion in IEEE 802.11b wireless networks," *ACM SIGCOMM*, 2005.

[6] H. Velayos, V. Aleo, G. Karlsson, "Load balancing in overlapping wireless LAN cells," *IEEE ICC*, 2004.

[7] S.Vasudevan, K. Papagiannaki, C. Diot, J. Kurose, D. Towsley, "Facilitating access point selection in IEEE 802.11," *ACM SIGCOMM*, 2005.

[8] D. Giustiniano, E. Goma, A. Lopez, J. Morillo, I. Dangerfield, P. Rodriguez, "Fair WLAN backhaul aggregation," *ACM MobiCom*, 2010.

[9] Z. Yuan, H. Venkataraman, G-M Muntean, "MBE: Model-Based Available Bandwidth Estimation for IEEE 802.11 Data Communications",' *IEEE T. Vehicular Technology*, vol. 61, no. 5, 2012.

[10] A. Jardosh, K. Papagiannaki, E. Belding, K. Almeroth, G. Iannaccone, B. Vinnakota, "Green WLANs: On-demand WLAN infrastructure," *Springer Mobile Networks and Applications*, vol. 14, no. 6, 2009.

[11] A. Jardosh, G. Iannaccone, K. Papagiannaki, B. Vinnakota, "Towards an energy-star WLAN infrastructure," *ACM HotMobile*, 2007.

[12] J. Lorincz, A. Capone, M. Bogarelli, D. Begusić, "Heuristic approach for optimized energy savings in wireless access networks," *IEEE SoftCOM*, 2010.

[13] E. Goma, *et al.* "Insomnia in the Access or How to Curb Access Network Related Energy Consumption," *ACM SIGCOMM*, 2011.

[14] D. Giustiniano, E. Goma, A. Lopez Toledo, P. Rodriguez, "WiSwitcher: An efficient client for managing multiple APs," *ACM PRESTO*, 2009.

[15] S. Kandula, K. C. Lin, T. Badirkhanli, D. Katabi, "FatVAP: Aggregating AP backhaul capacity to maximize throughput," *ACM USENIX*, 2008.

[16] Cisco Mobile Data Traffic Forecast `http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/whitepaperc11-520862.html`

[17] K. Xu, J. Crowcroft, V. O. K. Li, P. Hui, R. Mortier, "Sharing airtime with Shair avoids wasting time and money," *ACM HotMobile*, 2009.

[18] R. Sivakumar, C-L. Tsao, "On effectively exploiting multiple wireless interfaces in mobile hosts," *ACM CoNEXT*, 2009.

[19] D. Han, K. Papagiannaki, D. G. Andersen, S. Seshan, A. Badam, M. Kaminsky, "The hare and the tortoise: taming wireless losses by exploiting wired reliability," *ACM MobiHoc*, 2011.

[20] X. Yang, P. Rodriguez, N. Laoutaris, M. Sirivianos, "Inter-datacenter bulk transfers with NetStitcher," *ACM SIGCOMM*, 2011.

[21] C. Joe-Won, Y. I. M. Chiang, S. Ha, S. Sen, "Tube: Time-dependent pricing for mobile data," *ACM SIGCOMM*, 2012.

[22] Y. Yi, I. Rhee, S. Chong, K. Lee, J. Lee, "Mobile data offloading: how much can wifi deliver?," *ACM CoNEXT*, 2010.

[23] N. Feamster, R. Teixeira, S. Crawford A. Pescape', S. Sundaresan, W. de Donato. "Broadband internet performance: a view from the gateway," *ACM SIGCOMM*, 2011.

[24] N. Vallina-Rodriguez, V. Erramilli, Y. Grunenberger, L. Gyarmati, N. Laoutaris, R. Stanojevic, K. Papagiannaki, "When david helps goliath: the case for 3g onloading," *ACM HotNets*, 2012.

[25] AirMagnet Spectrum XT, `www.airmagnet.net/products`.

[26] Bandspeed AirMaestro spectrum analysis solution. `http://www.bandspeed.com/`.

[27] D. Niculescu, "Interference map for 802.11 networks," *ACM IMC*, 2007.

[28] J. Padhye et al., "Estimation of link interference in static multi- hop wireless networks", *ACM IMC*, 2005.

[29] N. Ahmed et al., "Online estimation of RF interference," *ACM CoNext*, 2008.

[30] A. Sheth et al., "MOJO: a distributed physical layer anomaly detection system for 802.11 WLANs," *ACM MobiSys,* 2006.

[31] D. Aguayo et al., "Link-level measurements from an 802.11b mesh network," *ACM SIGCOMM,* 2004.

[32] M. Vutukuru, K. Jamieson, and H. Balakrishnan, "Harnessing exposed terminals in wireless networks," *ACM NSDI,* 2008.

[33] R. Chandra, J. Padhye, A. Wolman, and B. Zill, "A location- based management system for enterprise wireless LANs," *ACM NSDI,* 2007.

[34] Y.-C. Cheng et al., "Automating cross-layer diagnosis of enterprise wireless networks," *ACM SIGCOMM,* 2007.

[35] R. Mahajan et al., "Analyzing the MAC-level behavior of wireless networks in the wild," *ACM SIGCOMM,* 2006.

[36] Y.-C. Cheng et al., "Jigsaw: solving the puzzle of enterprise 802.11 analysis," *ACM SIGCOMM,* 2006.

[37] V. Shrivastava, S. Rayanchu, S. Banerjee, and D. Papagiannaki, "PIE in the sky: online passive interference estimation for enterprise WLANs," *USENIX,* 2011.

[38] S. Rayanchu, A. Patro, and S. Banerjee, "Airshark: Detecting non-WiFi RF devices using commodity WiFi hardware," *ACM IMC,* 2011.

[39] G. Bianchi, "Performance analysis of the IEEE 802.11 Distributed Coordination Function," *JSAC*, vol. 18, no. 3, 2000.

[40] P. Chatzimisios, A. C. Boucouvalas, V. Vistas, "Performance analysis of the IEEE 802.11 DCF in presence of transmission errors," *IEEE ICC,* 2004.

[41] 802.11 standard, `http://standards.ieee.org/about/get/802/802.11.html`

[42] A. K. Mahani, M. Naderi, C. Casetti, C.-F. Chiasserini, "MAC layer channel utilization enhancements for wireless mesh networks," *IEEE Communications,* vol. 3, no. 5, pp. 794–807, May 2009.

[43] Wireless driver, `http://linuxwireless.org/`

[44] N. Blefari Melazzi, D. Di Sorte, M. Femminella, G. Reali, "Toward an autonomic control of wireless access networks," *IEEE GLOBECOM*, pp. 954–959, Nov. 2005.

[45] A. P. Jardosh, K. N. Ramachandran, K. C. Almeroth, E. M. Belding-Royer, "Understanding congestion in IEEE 802.11b wireless networks," *ACM SIGCOMM*, 2005.

[46] G. Sawma, G. Ben-El-Kezadri, R. Aib, I. Pujolle, "Autonomic management for capacity improvement in wireless networks," *IEEE CCNC*, 2009.

[47] A. Jardosh, K. Papagiannaki, E. Belding, K. Almeroth, G. Iannaccone, B. Vinnakota, "Green WLANs: On-Demand WLAN infrastructure," *Springer Link Mobile Networks and Applications*, vol. 14, no. 6, pp. 798–814, 2009.

[48] H. Velayos, V. Aleo, G. Karlsson, "Load balancing in overlapping wireless LAN cells," *IEEE ICC*, 2004.

[49] M. Lacage, H. Manshaei, T. Turletti, "IEEE 802.11 rate adaptation: A practical approach," *ACM MSWiM*, 2004.

[50] M. Heusse, F. Rousseau, G. Berger-Sabbatel, A. Duda, "Performance anomaly of 802.11b," *IEEE INFOCOM*, 2003.

[51] C. Rossi, C. Casetti, C.-F. Chiasserini, "Energy-efficient wireless resource sharing for federated residential networks," *IEEE WoWMoM*, 2012.

[52] N. Mishra, K. Chebrolu, B. Rama, A. Patha, "Wake-on-WLAN," *ACM WWW*, 2006.

[53] X. Ai, V. Srinivasan, C.-K. Tham, "Wi-Sh: A simple, robust credit based Wi-Fi community network," *IEEE Infocom*, 2009.

[54] T. Clancy, M. Nakhjiri, V. Narayanan, L. Dondeti, "Handover Key Management and Re-Authentication Problem Statement," RFC 5169, IETF.

[55] L. Suresh, J. Schulz-Zander, R. Merz, A. Feldmann, T. Vazao,"Towards Programmable Enterprise WLANs with Odin,", *ACM HotSDN*, 2012.

[56] K. Kong, W. Lee, Y. Han, M. Shin, H. You, "Mobility management for all-IP mobile networks: Mobile IPv6 vs. proxy mobile IPv6," *IEEE Wireless Communciations*, vol. 15, no. 2, 2008.

[57] H. Fathi, S. Chakraborty, "Optimization of mobile IPv6-based handovers to support VoIP services in wireless heterogeneous networks," *IEEE Trans. on Veh. Tech.*, vol. 56, no. 1, 2007.

[58] Mobile IP traversal of Network Address Translation (NAT) devices, `http://tools.ietf.org/html/rfc3519`.

[59] C. Rossi, C. Casetti, C.-F. Chiasserini, G. Rondini, "A new metric for admission control in multi-rate 802.11 WLANs," *IEEE WONS*, 2011.

[60] C. Rossi, C. Casetti, C.-F. Chiasserini, "Bandwidth Monitoring in Multi-rate 802.11 WLANs with Elastic Traffic Awareness," *IEEE GLOBECOM*, 2011.

[61] L. Di Cioccio, R. Teixeira, C. Rosenberg, "Measuring Home Networks with HomeNet Profiler," *Springer PAM* 2013

[62] 80211v `http://standards.ieee.org/findstds/standard/802.11v-2011.html`

[63] 802.11h `http://standards.ieee.org/getieee802/download/802.11h-2003.pdf`

[64] OMNeT++, `http://www.omnetpp.org/`.

[65] INETMANET Extension, `http://github.com/inetmanet/inetmanet/wiki`.

[66] T. Chrysikos, G. Georgopoulos, S. Kotsopoulos, "Site-specific validation of ITU indoor path loss model at 2.4 GHz," *IEEE WoWMoM*, 2009.

[67] `h18000.www1.hp.com/products/quickspecs/12510_na/12510_na.PDF`

[68] `http://www.libelium.com/support/waspmote`

[69] T. Nguyen, A. Black, "Preliminary study on power consumption of typical home network devices," *Tech. Rep. 071011A*, 2007.

[70] `http://www.cisco.com/en/US/prod/collateral/wireless/ps5678/ps11983/data_sheet_c78-686782.pdf`

[71] G.-F. Zhao, Q. Shan, S. Xiao, C. Xu, "Modelling web browsing on mobile Internet," *IEEE Communications Letters,* vol. 15, no. 10, 2011.

[72] Mobile IP Deployment Difficulties `http://www.ietf.org/proceedings/83/slides/slides-83-mobopts-0`

[73] IAPP `http://gredes.ifto.edu.br/wp-content/uploads/802.11F-20031.pdf`

[74] C. Rossi, C. Borgiattino, C. Casetti, C.-F. Chiasserini, "Energy-efficient Wi-Fi gateways for federated residential networks," *IEEE WoWMoM*, 2013.

[75] S. Lu, H. Luo, H. Li, C. Peng, S-B Lee, "Traffic-driven power saving in operational 3g cellular networks," *ACM MobiCom*, 2011.

[76] P. Barford J. Sommers, "Cell vs. wifi: on the performance of metro area mobile connections," *ACM IMC*, 2012.

[77] H. Han, S. Shakkottai, C. V. Hollot, R. Srikant, D. Towsley, "Multi-path tcp: a joint congestion control and routing scheme to exploit path diversity in the internet,". *IEEE/ACM Trans. Netw.*, vol. 14, no. 6, pg. 1260-1271.

[78] B. Nechaev, V. Paxson, C. Kreibich, N. Weaver, "Netalyzr: illuminating the edge network," *ACM IMC*, 2010.

[79] N. Laoutaris, G. Smaragdakis, P. Rodriguez, R. Sundaram, "Delay Tolerant Bulk Data Transfers on the Internet," *ACM SIGMETRICS*, 2009.

[80] EU Commission, Commission staff working document on the implementation of national broadband plans `http://ec.europa.eu/information_society/newsroom/cf/itemdetail.cfm?item_id=7948`

[81] US Federal Communications Commission. National Boadband Plan `http://www.fcc.gov/measuring-broadband-america`

[82] FTTH Council Europe, Annual Report `http://www.ftthcouncil.eu/documents/Reports/FTTHCE_AnnualReport_2011-2012.pdf`

[83] Ericsson, Traffic and Market report, on the pulse of networked society `http://www.ericsson.com/res/docs/2012/traffic_and_market_report_june_2012.pdf`

[84] AT Kearney, a viable future model for the Internet `http://www.atkearney.com/images/global/pdf/Viable_Future_Model_for_Internet.pdf`

[85] Q. Xu, Z. Mao, M. Zhang, Z. Wang, Z. Qian, "An untold story of middleboxes in cellular networks," *ACM SIGCOMM*, 2011.

[86] Iperf project `http://sourceforge.net/projects/iperf/`

[87] Netperf project `http://www.netperf.org/netperf/`

[88] V. Paxson, M. Allman, G. Maier, A. Feldmann, "On dominant characteristics of residential broadband internet traffic," *ACM IMC*, 2009.

[89] HTTP Live Streaming draf-ietf `http://tools.ietf.org/html/draft-pantos-http-live-streaming-07`

[90] I. Sodagar, "The mpeg-dash standard for multimedia streaming over the internet," *IEEE MultiMedia*, 2011.

[91] E.W. Biersack, P. Rodriguez, A. Kirpal, "Parallel-access for mirror sites in the internet," *IEEE INFOCOM*, 2000.

[92] I. Stavrakakis, N. Laoutaris, "Intrastream synchronization for continuous media streams: a survey of playout schedulers," *IEEE Network*, 2002.

[93] M. Munafo', T. Ruben, S. G. Rao, A. Finamore, M. Mellia, "Youtube everywhere: impact of device and infrastructure synergies on user experience," *ACM IMC*, 2011.

[94] S. Soumya et al., "Pricing data: A look at past proposals, current plans, and future trends," *CoRR*, 2012.

[95] M-F Balcan, Avrim Blum, Yishay Mansour, "The price of uncertainty," *ACM EC*, 2009.

[96] R. Stanojevic, V. Erramilli, K. Papagiannaki, "Cognitive bias in network services," *ACM HotNets*, 2012.

[97] A. Krishnamurthy, Y. Liu, A. Silberschatz, H. Xie, Y. R. Yang, "P4P: Provider portal for applications," *ACM SIGCOMM*, 2008.

[98] C. Rossi, N. Vallina-Rodriguez, V. Erramilli, Y. Grunenberger, L. Gyarmati, N. Laoutaris, R. Stanojevic, K. Papagiannaki, P. Rodriguez, "3GOL: Power-boosting ADSL Using 3G Onloading," *ACM CoNEXT*, 2013.

[99] C. Rossi, Y. Grunenberger, K. Papagiannaki,"Power-boosting Residential Wired Broadband," *ACM WiNTECH*, 2013.

[100] C. Rossi, Y. Grunenberger, E. Goma, K. Papagiannaki, "Can Wireless Support Wired Broadband? A Practical Showcase for Residential Networks," *IEEE INFOCOM*, 2013.