POLITECNICO DI TORINO
Repository ISTITUZIONALE

Cooperative Energy-efficient Management of Federated WiFi Networks

Original
Cooperative Energy-efficient Management of Federated WiFi Networks / Rossi, C., Casetti, C.E., Chiasserini, C.F., Borgiattino, C.. - In: IEEE TRANSACTIONS ON MOBILE COMPUTING. - ISSN 1536-1233. - STAMPA. - 14:11(2015), pp. 2201-2215. [10.1109/TMC.2015.2392109]

Availability:
This version is available at: 11583/2584398 since: 2015-11-26T11:48:47Z

Publisher:
IEEE - INST ELECTRICAL ELECTRONICS ENGINEERS INC

Published
DOI:10.1109/TMC.2015.2392109

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

05 September 2026

Cooperative Energy-efficient Management of Federated WiFi Networks

Claudio Rossi, Claudio Casetti, *Member, IEEE*, Carla-Fabiana Chiasserini, *Senior Member, IEEE*, Carlo Borgiattino, *Student Member, IEEE*

***Abstract*—The proliferation of overlapping, always-on IEEE 802.11 Access Points (APs) in urban areas can cause inefficient bandwidth usage and energy waste. Cooperation among APs could address these problems by allowing underused devices to hand over their wireless stations to nearby APs and temporarily switch off, while avoiding to overload a BSS and thus offloading congested APs. The federated house model provides an appealing backdrop to implement cooperation among APs. In this paper, we outline a distributed framework that assumes the presence of a multipurpose gateway with AP capabilities in every household. Our framework allows cooperation through the monitoring of local wireless resources and the triggering of offloading requests toward other federated gateways. Our simulation results show that, in realistic residential settings, the proposed framework yields an energy saving between 45% and 86% under typical usage patterns, while avoiding congestion and meeting user expectations in terms of throughput. Furthermore, we show the feasibility and the benefits of our framework with a real test-bed deployed on commodity hardware.**

I. Introduction

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 [1], both an energy waste and an unnecessary increase in electromagnetic pollution. As reported in [2], the access network, including end-users and last-mile equipment, accounts for more than 90% of the energy consumed by networking devices, where the home gateway alone is responsible for up to 79% of the total bill.

In order to reduce the energy footprint due to networking, we exploit a new paradigm for home networking which is garnering widespread attention, based on the concept of *federated homes* [3]. 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 (GW)* that handle all inward and outward network traffic. GWs are advanced home devices capable of offering storage and multimedia services, including audio and video real-time streaming. They can control a IEEE 802.11 Basic Service Set (BSS) and they are supplied with a backhaul connection to an Internet Service Provider (ISP).

In order to optimize the usage of the wireless medium and save energy, federated GWs with overlapping coverages should identify and optimally relocate the Wireless Stations (WSs) among themselves, and, possibly, turn themselves off if a subset of nearby GWs can adequately support the current load requested by the WSs. Also, an underloaded (or temporarily switched off) GW should be called upon for help by GWs that experience a congested wireless medium, and asked to accept the association of some of their WSs.

Such operations require that GWs have self-load assessment capabilities and run inter-GW 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, the data rate and the packet error 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. For example, a 50% idle time does not translate into half of the WiFi channel capacity, because the achievable throughput depends on the packet error rate, which in turn affects automatic rate adaptation algorithms. Furthermore, due to the 802.11 DCF access scheme, the overhead depends on packet size and number of contending WSs. 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 provide a bound against the wireless channel capacity. Such a brief review of the literature clearly points to the need of a novel approach for load estimation that fully takes into account the protocol dynamics of an IEEE 802.11 BSS.

As for the relocation of WSs, note that this significantly differs from seamless handover solutions for heterogeneous networks [10], as well as from offloading schemes as in [11], [12]. WS relocation can be done in either a centralized or a distributed fashion. Centralized solutions are suitable for coverages resulting from controlled placement of the GWs, as is the case of corporate networks and college campuses, but they hardly fit a residential scenario where each GW is independently placed within a household. Also, centralized so-

The authors are with Politecnico di Torino, Torino, Italy.

lutions require network-side controllers to be installed, hence a multilevel hierarchical structure whose scale might limit inter-domain deployments. Centralized solutions go in the direction of "dummy" GWs, while a distributed approach tries to push more intelligence to the edge of the network, in order to relieve the load of the network core. Examples of centralized relocation schemes that enable GWs to switch themselves off have been proposed in [13]–[15]. Also, WS relocation can be either GW- or WS-initiated, as in [16]. These approaches have different impact in terms of hardware/software modification to the devices, as well as of the degree of signaling involved. Other solutions have been designed to overcome capacity limitations of single APs. In particular, [17] has suggested the use of TDMA techniques to let WSs access multiple APs at a time, while [8], [18] present drivers that aggregate the bandwidth available at different APs and balance their load. Such approaches, besides differing in scope from our work, require modification in the WSs.

In order to address the above issues, we propose a protocol that follows a *distributed, GW-initiated* paradigm that achieves energy efficiency in federated residential networks. Our main contributions are:

- A lightweight algorithm for self-load assessment of federated GWs that only relies on passive measurements and that fully considers the dynamics of a 802.11 BSS.
- An inter-GW load assessment algorithm that enables GWs to manage WS relocations without creating congestion.
- A fully-distributed protocol that implements the energy-efficient management of WSs within the federated network.
- The specification of the network architecture needed to implement our energy-efficient scheme, including security issues and handover management.
- The evaluation of both the load assessment algorithms and the distributed protocol by network simulation.
- The formulation of the optimization problem pertaining a centralized solution, which we compare against our protocol obtaining close-to-optimal results.
- The realization of a small test-bed that proves the feasibility of our proposals on commodity hardware.

We stress that all proposed solutions do not require changes to either the 802.11 standard or to the WSs, which may be even unaware of its adoption, and they can be fully implemented on the GW.

The remainder of this paper is organized as follows. In Section II, we outline the distributed protocol with some motivating examples, while the network architecture we propose is detailed in Section III. In Section IV we describe the algorithms that let GWs assess their current load. We evaluate these algorithms in Section VI. The distributed inter-GW communication protocol is presented in Section V and evaluated in Section VII, along with the aforementioned algorithms, in a realistic residential scenario. We present and evaluate our test bed in Section VIII. Finally, Section IX outlines our conclusions.

II. Outline and Motivating Examples

With respect to the different management techniques introduced above, we choose to pursue a *distributed, GW-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 GWs (i) have self- and interGW-load assessment capabilities; (ii) run inter-GW communication protocols for WS relocation; (iii) rely on robust authentication and authorization procedures provided by the federated network and (iv) use reliable Wake on Wireless LAN procedures [19].

Firstly, self-load assessment through traffic measurement allows GWs to classify their status as either *Light*, *Regular*, or *Heavy*. The assessment depends on the available bandwidth which depends on the number of associated WSs, their position and the traffic profile they are generating within the BSS that each GW controls. In particular, we compute the former by leveraging theoretical results on the BSS saturation throughput [20], [21] thus accounting for the collision probability, the channel errors, the different data rates and payload sizes used within the BSS. Status evaluation guides GW policy in either seeking relocation of their WSs or in accepting WSs handed over by nearby GWs. In 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. Heavy status, instead, characterizes an overloaded BSS, where some WSs should associate to other BSSs to benefit from congestion offloading. A GW 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 GW status does not affect the spontaneous association of a WS within one's household, thus it does not interfere with normal operations.

Secondly, the relocation of WSs within the federated network involves communication and coordination among GWs. A GW 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 GW, and its parameters included in the relocation request, the presumed transmission rate cannot be easily predicted and needs a form of remote sampling, as described later. The GW requesting help then chooses to offload the WS to the most suitable candidate GW.

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 GW authentication and its wake-on process.

An example of the procedures just outlined is presented in Fig. 1, where a 3-house neighborhood is displayed. The first example (top half of Fig. 1) shows a case of energy saving through the switching off of some GWs. With reference to the first row of the figure, in household 1, we assume that one WS is "whispering", i.e., occasionally sending low

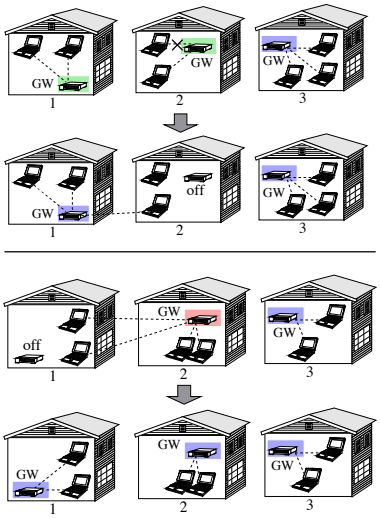


Fig. 1. Energy saving (top) and congestion avoidance (bottom) by WS offloading.

background traffic (mainly, status update for some applications and other signaling). Having completed the download of a software update, the other WS starts whispering as well. As a result, the GW in household 1 (GW 1) 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 is turned off, hence also the status of GW 2 shifts to Light. Finally, in household 3, we assume one WS is listening to music streamed over the Internet, while the other two are browsing. Their GW 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 steady state shown in the second row of Fig. 1 is reached.

Our second example (bottom half of Fig. 1) shows instead a case of congestion relief. 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 a steady state depicted in the last row of Fig. 1 (i.e., all GWs in Regular status) is reached.

III. Network Architecture

In our study, we consider a set of residential units (e.g., houses or apartments), each of them equipped with a GW, with both 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 [22]. We assume that federated households sign a cooperation agreement before joining the federated community. A number of incentives schemes have been proposed in the literature in order to regulate cooperation among peers [23], [24]. Among these, the tit-for-tat strategy could be particularly suitable for the scenario under study [25]. However, due to the extent and complexity of this problem, which would require a separate study on incentive, pricing and cooperation management, we leave it outside the scope of this paper.

A. The federated Gateways

A federated GW can communicate and coordinate with other federated GWs using multicast over an out-of-band channel, which runs through their backhaul Internet connection. We assume that all GWs are synchronized with an NTP (Network Time Protocol) server provided by their ISP, reachable through the backhaul, that is known to guarantee synchronization in the order of milliseconds.

When it is "on", a GW offers local wireless access through the 802.11 a/b/g technology over independently-managed (but possibly coordinated) frequency channels. The scenario can be extended to also include GWs adopting newest standards, i.e., 802.11n/ac, by updating the estimation of the Saturation throughput detailed in Section IV. The WSs that operate within a BSS controlled by a GW 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 GW and the WSs transmit frames at a data rate that may vary according to the experienced channel propagation conditions.

When GWs 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 [19].

B. The Radio Federated Network

We define as Radio Federated Network (RFN) within a federation, a subset of GWs 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 paper, can occur at startup with a scanning procedure. Note that modern GWs already include Automatic Channel Selection (ACS) procedures that require scanning at startup. Given the low churn rate of residential GWs, we can build the first-hop neighbors while performing the initial ACS procedure. We consider the first-hop neighbors of a GW as its RFN.

Inter-GW, out-of-band communication, GW 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), which must be refreshed periodically. The new FGK must thus be distributed to all members of the RFN including "off" GWs. To this end, also "off" GWs maintain a loose synchronization, previously acquired through an NTP server, using a standalone clock. Upon a scheduled key expiration, an "off" GW will switch on in order to update its FGK. Additionally, in order to allow WSs to seamlessly associate to a new GW, 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 [26], and will not be discussed further in this paper.

C. The Handover procedure

The handover at the MAC layer can be seamlessly implemented by using one virtual access point (VAP) for each WSs, as proposed in [27]. 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 runs as many VAPs as the number of associated WSs. Moving a VAP from a 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 [28], [29] 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, seamless handover for real-time applications 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 [30]. We assume that the federated network is managed by a single ISP which implements also PMIPV6 or a similar technology in order to guarantee a seamless handover for applications. We note that applications capable of restarting their flow after a disconnection, as is done for example by Dropbox, 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.

IV. Gateway Status Assessment and Management

Here, we first present the algorithm that lets a GW assess the load level within its BSS, hence determine whether it needs help from other GWs to relocate its WSs or not. Then, we describe how a GW assesses its suitability to provide help to others. As mentioned, the main idea at the basis of the two algorithms is to leverage the theoretical results on the saturation throughput derived in [20], [21], for the computation of the BSS capacity. We therefore start by introducing our notation and the computation of some fundamental quantities. The effectiveness of our approach will be then evaluated in Section VI-A.

A. Preliminaries

A GW reads the "protocol type" field in IP packets and collects statistics on elastic and inelastic traffic within its BSS. The GW carries out such measurements periodically over a time interval, hereinafter referred to as measurement period. The GW considers a node in the BSS 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.

Similarly to the mechanism we described in [31], [32], every measurement period, for each active node[1] k the GW 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 GW computes a running average of its own downlink (DL) throughput for both the elastic and inelastic traffic it handles for each node 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 GW records the payload size and the used data rate, and it computes the corresponding running averages. We will refer to all the above measurements the GW performs for a WS as the WS *traffic profile*. Then, the GW computes the running average of the data rate, R, and of the payload size, P, over all data frames that it sends or receives.

Another fundamental quantity for our assessment algorithms is the (aggregate) saturation throughput A, which we take as the value of wireless *BSS capacity*. The saturation throughput is defined as in [21], which extends the original Bianchi's model [20] in presence of errors due to channel propagation conditions:

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

In (1), τ is the probability that a node (either a WS or the GW) accesses the medium at a generic time slot[2], p_e is the filtered average packet error rate, 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 to channel errors, or a collision. As far as the average collision duration is concerned, its exact computation would require the GW to be aware of the number of nodes that are hidden with respect to each other. The works in [20], [21] do not account for hidden WSs and the approaches proposed in the literature are not viable in our settings. Indeed, we do not require the GW to have knowledge of the user distribution within its coverage area. Thus, we approximate the average collision duration by making a worst-case assumption: each collision involves a packet of maximum payload size among those observed by the GW during the measurement period. Clearly, the average collision time is overestimated in absence of hidden WSs, leading to underestimating the saturation throughput. This, however, is acceptable for our purposes, as also proved by the results presented in Section VI-A. Except for such a variation, the expressions of τ and $E[T]$

[1]The active node set excludes the GW but may include also wired stations connected to the GW.

[2]Considering a slotted time is the main approximation made in [20], [21].

Algorithm 1 Gateway status assessment

Compute throughput A, capacity S and initialize the load L to 0
for every active Node excluding the GW 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 $\frac{L}{S} \leq T_L \rightarrow Light$
  else if $\frac{L}{S} > T_H \rightarrow Heavy$
    else $Regular$

Algorithm 2 Assessing the Gateway suitability to provide help

Compute the estimated throughput $A^\star$, hence $S^\star$, 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

are derived following [21] and reported also in our previous work [33], while p_e 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 p_e estimation, as collisions are also included in the count[3].

We stress that, although A 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 (1) depend on the payload, data rate and access rate of each single WS.

B. Does the Gateway need help?

Every measurement period and through running averages, the GW computes the capacity of the BSS it controls using the expression of the saturation throughput in (1). Denoting by C the capacity of the GW backhaul connection, the GW calculates the available capacity S as $S = \min(C, A)$. Then, the GW computes the traffic load L within the BSS and compares it to the available capacity S, so as to gauge its own status.

In order to assess the load L, hence its status, the GW leverages the throughput measurements it has performed and computes L as the sum of all contributions due to the existing traffic (Algorithm 1). Specifically, the contribution due to inelastic flows is set to their measured throughput. For the elastic flows of each WS, instead, the GW mitigates the effect of their greedy behavior by dampening their contribution to a fraction of the available capacity, namely, αS.

If the traffic load, normalised to the available capacity, is below a threshold T_L the GW is in Light status. In this case, the GW 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 GW 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 GWs is required.

We point out that, in order to avoid frequent ping-pong handovers between GWs, a smoothing filter can be applied to the computation of the estimated load $L^\star$, thus dampening the reactivity to sudden changes. Since such changes are contextual to the individual home environment, a global optimal value for the smoothness of the filter is hard to determine and can be left as a tuneable parameter for the user.

C. Who can help the Gateway?

Upon the reception of a help request asking for WS relocation, a federated GW needs to reliably evaluate the impact on its BSS of associating additional WSs, i.e., its suitability to give help. To do so, the GW 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 denote it by ρ. We use it as a suitability index: the greater the room metric, the more suitable the GW to accommodate the WSs. For simplicity, the room metric computation is outlined below and in Alg. 2 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 GW that has to assess its suitability to provide help, and let x be the WS that another GW would like to relocate. As detailed later, through signaling exchange between GWs, GW i can acquire the uplink and downlink throughput that x expects to achieve for inelastic and elastic traffic, as well as the average payload of the frames that x transmits and receives. Using (1), GW i computes the saturation throughput $A^\star$, possibly updates S, as if x had already been 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. Next, in order to evaluate the impact that the association of x would have on the performance of existing flows, the GW estimates what the load of the BSS would be if the throughput demand of all WSs were fulfilled. To this end, its current load estimate is augmented by the uplink and downlink throughput that x expects for its elastic and inelastic traffic. Similar to the procedure for the GW status assessment, the effect of greedy elastic flows, involving either the existing WSs or x, is mitigated by letting them account for, at most, a fraction of the available capacity.

The room metric ρ is then 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 GW in Heavy status (i.e., the estimated normalized load exceeds T_H), then x is rejected; otherwise, the GW reputes that the WS can be relocated into its BSS.

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

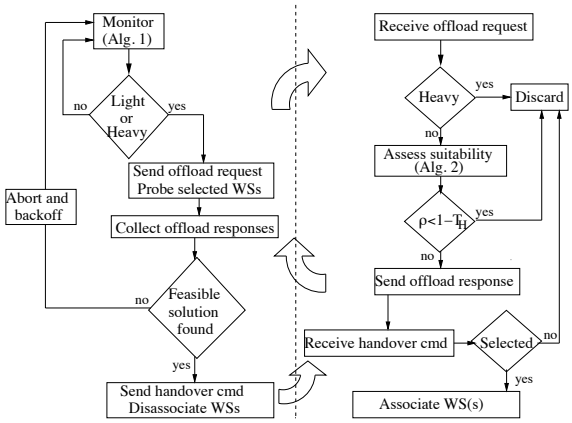


Fig. 2. Flow chart of the offload procedure: Gateway asking for help (left) and Gateway receiving the help request (right).

V. Resource Sharing Protocol

We now introduce the protocol that lets federated GWs share their radio resources. As mentioned, our objective is twofold: (i) to minimize the number of switched-on GWs, and (ii) to avoid overloading traffic conditions for "on" GWs. To achieve these goals, a GW periodically assesses its status through Alg. 1, and, if in Light or Heavy status, it starts an offload procedure, as summarized in Fig. 2. The procedure aims at relocating one or more WSs to other GWs. The federated GWs estimate which WSs they could accept in their BSS, based on the value of their room metric computed through Alg. 2, and reply accordingly. Upon finding a valid WS relocation, the GW 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 GWs, not in WSs. Also, the GW 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 GW in Light or Heavy status is detailed below.

1) Consider a GW, GW l, that finds itself in **Light status**. Then, GW l starts an offload procedure by multicasting an OFFLOAD_REQUEST message to GWs in its RFN. This message includes the following information: (i) the status of the requesting GW, 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 a request timeout value.

An OFFLOAD_REQUEST is processed only by those GWs in the RFN that are currently "on" and not in Heavy status. Since the request comes from a GW in Light status, the federated GWs 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 and thus have a greater chance to switch off. 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 the GW sends a CTS so that the WSs it is serving will be frozen for a short time while it tunes 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 replies with a CTS, the GW monitoring the frequency channel can measure the signal-to-noise ratio (SNR) and roughly estimate the data rate it could use 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. Specifically, the RTS duration field is set to the sum of the SIFS time, CTS transmission time and the hash of the 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. Such a procedure allows a GW that is not in radio proximity 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 GW then considers the WSs from which it heard a CTS and assesses which of them (if any) could be associated to its BSS. To do so, the GW evaluates the room metric for the possible combinations of candidate WSs through Alg. 2. 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 request timeout, GW l evaluates all received replies by scanning them and 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 rationale is that WSs should be handed over to the GWs that can communicate with them at the highest data rate, so as to guarantee an efficient traffic transfer. This procedure is scalable as it requires only one pass among the list of replies, which are limited by the size of the RFN[4]. Note that an allocation is valid only if *all* the WSs are assigned to another GW.

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

2) When a GW, 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. In order to minimize the number of handed-over WSs, GW h 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 WSs with the biggest impact on the used airtime are selected first, and the handover of each WS

[4]50% of the residential GWs have less than 10 neighbors [34].

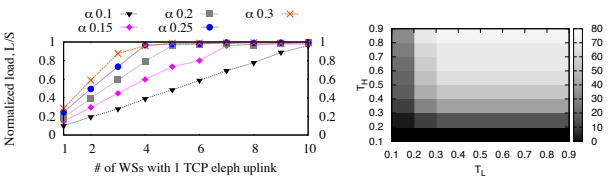


Fig. 3. Sensitivity of normalized load estimation to the algorithm parameter α, and of the percentage of "off" Gateways to thresholds T_L and T_H.

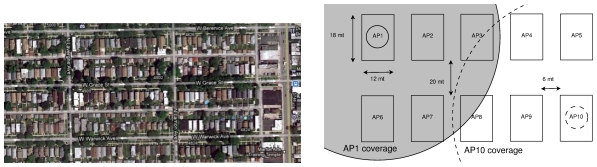


Fig. 4. Federated detached houses scenario: Google view of the area (left) and abstract representation (right). In the latter, the radio coverages of AP1 and AP10 are highlighted for the sake of example.

results into a different offload procedure.

Secondly, upon receiving an OFFLOAD_REQUEST from GW h, an "on" GW not in Heavy status will always reply. Again, the whole inter-GW 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" GW and ask for help. The GW 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 GWs. However, the exchange of landline signaling could be unfeasible due to the "off" state of the GW. 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" GW. The low-cost interface of GW w can detect the sequence and turn the rest of the GW circuitry on. However, in order to avoid attacks aiming at unnecessarily waking up "off" GWs, $\mathbb{W}$ should be followed by an encrypted Message Authentication Code (MAC), $\mathbb{M}$, which can be decoded only using the FGK. Since all GWs 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 $\mathbb{W}' = \mathbb{W}$. Conventional approaches, such as introducing an exponentially distributed delay inversely proportional to the interarrival time between successive requests, can be used to implement a protection against DDoS attacks. The woken-up GW w joins again the RFN, thus signaling every other GW that it is "on" again.

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 GW while GW w, having not associated any WS, will switch off again after a timeout. In this way, we let "off" GWs turn themselves on if needed, while limiting the number of GWs that wake up.

Finally, we remark that, upon receiving an OFFLOAD_REQUEST, a GW wishing to start an offload procedure defers its request till it receives a HANDOVER_COMMAND or an ABORT, and then perform a backoff. Since all GWs are synchronized with NTP, it is ensured that in the RFN there is only one active offload procedure at the time.

VI. Evaluation of the Load Assessment Algorithms

Since we are particularly focused on evaluating the WiFi performance and limits, we consider the case where no wired node is present and we assume that the capacity bottleneck is given by the wireless link. We implemented our algorithms in the Omnet++ v4.1 simulator starting from the IEEE 802.11g module included in the INETMANET extension. 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 [35]; also, we implemented the automatic data rate adaptation scheme Sample Rate [36], which is also implemented in the linux wireless driver [37].

Elastic traffic is simulated using TCP SACK, while inelastic traffic is represented by UDP flows, which support a non-negligible portion of user applications [38]. Since HTTP is the most used application according to [38], we consider both elephant and mice TCP flows: 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, the duration of the measurement period is set to 3 s. This value was found to represent a good compromise between the need to let the system quickly react to changes in the traffic and transmission conditions, and the need to capture a meaningful average of the system behavior. Then, we perform a sensitivity analysis in view of setting a proper value for α, T_L and T_H.

To evaluate how α affects the normalized load (L/S), we simulate a single BSS in different configurations, increasing the number of transmitting WSs at each step (from 1 to 10). We run every configuration for 2 minutes, during which each WS generates only one uplink elephant TCP flow, with different values of α. The left plot in Fig. 3 shows that, as expected, L/S grows linearly with α until it approximately reaches 1, which means that the GW is fully loaded. Clearly, setting α to 1 would result in a fully-loaded BSS in presence of just one elephant TCP flow. In accordance with the study in [34], which reports that in a household there are at most 4 active WSs 75% of the time, we set $\alpha = 0.25$, so that 4 WSs transmitting a TCP elephant flow fully load the BSS.

Next, we evaluate the impact of T_L and T_H on the percentage of switched "off" GWs. We use a realistic scenario referring to a neighborhood located in the suburbs of Chicago, IL. The RFN scenario, depicted in Fig. 4, includes 10 federated detached houses, each equipped with an IEEE 802.11g GW. The average fraction of GWs in radio visibility of a WS, when a data rate of 6 Mbps is used, is 0.8.

We initially consider 2 WSs associated to each GW, and we let each WS generate a UDP uplink flow at 2 Mbps. We

run different simulations by varying both thresholds from 0.1 to 0.9 and we observe the number of switched off gateways. As we can see from the right plot in Fig. 3, the thresholds T_L and T_H have a significant impact on the performance of our algorithm. Intuitively, T_L controls the triggering of help requests, while T_H represents the maximum normalized load above which a GW looks for another GW to hand over some of its WSs to it. Since from the plot it can be seen that the maximum percentage of "off" GWs can be obtained for $T_L > 0.3$ and $T_H > 0.8$, we set $T_L = 0.4$ and $T_H = 0.9$. Also, note that a fine tuning of T_L and T_H can be easily done for the scenario at hand at the network startup, since GW deployment as well as radio coverage can be assumed to be stationary for federated residential networks.

In the following, first we evaluate the accuracy of the proposed algorithms in assessing the GW status and then we study its suitability to accommodate additional WSs in its BSS.

A. Gateway status and suitability assessment

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

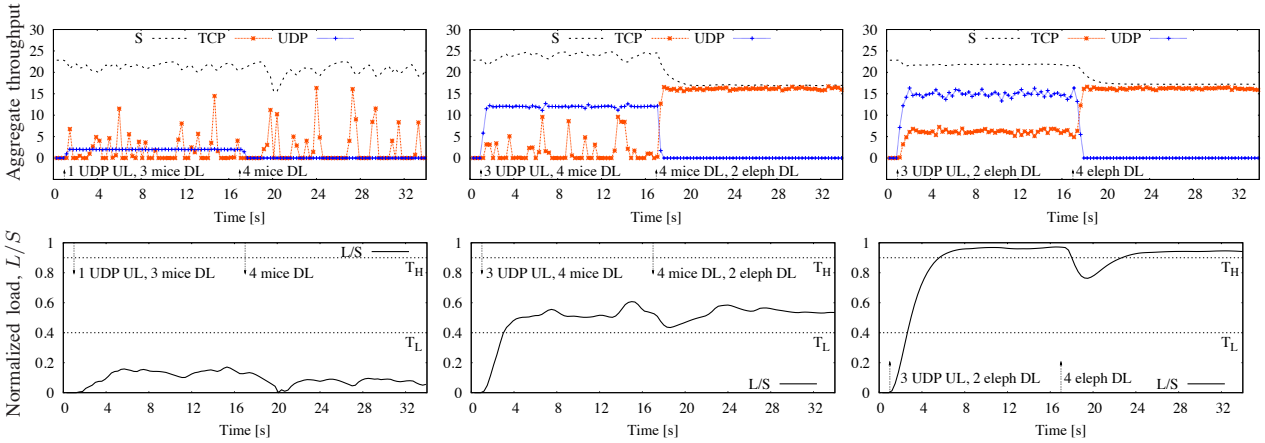
The first scenario we study corresponds to an underloaded BSS, which includes a WS originating an uplink 2-Mbps UDP flow, and three other WSs that are the destinations of one mouse TCP flow each. The aggregate throughput for elastic and inelastic traffic is depicted in the top plot of Fig. 5(a), along with the saturation throughput S. The load estimate carried out by the GW is shown in the bottom plot, from which it can be seen that the GW 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 GW 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. 5(b) presents results for a medium-loaded BSS. Specifically, now there are three WSs that originate UDP traffic at 4 Mbps and receive a mouse TCP flow; a fourth WS is the destination of one mouse TCP flow. At t=17 s, UDP streams end and two WSs become the destinations of an elephant TCP flow. As evident from the bottom plot in Fig. 5(b), the Regular status is always correctly detected.

Finally, Fig. 5(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. 6, depicts the accuracy of our algorithm in estimating the suitability of a GW to accommodate additional WSs. In this case, a tagged GW receives offload requests from its federated GWs and needs to assess how much room (if any) there is in its BSS. We refer to a scenario that includes four GWs, three of which would like to relocate a WS to the tagged GW. 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 sends an elephant TCP flow and is the destination of a 0.5-Mbps UDP stream. The tagged GW computes its value of room metric, which shows bandwidth availability (right plot), then, in our example, the WS is relocated to the tagged GW. At t=12.5 s, a second relocation request is received for a WS that transmits a 0.5-Mbps UDP stream and is the destination of an elephant TCP flow. The room metric correctly reflects the GW 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.



(a) Light (b) Regular (c) Heavy

Fig. 5. Detection of the Gateway status. Saturation and aggregate (elastic and inelastic) throughput (top plots); normalized load and status detection with respect to the thresholds T_L and T_H (bottom plots). The Light, Regular and Heavy status are always correctly detected.

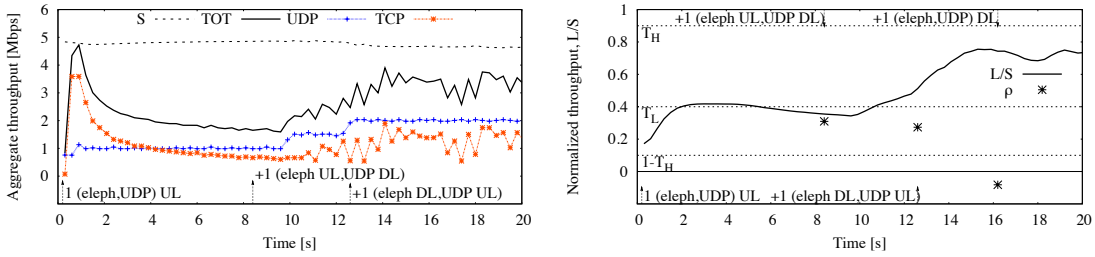


Fig. 6. Relocation requests arrive at the Gateway for WSs that are sources and destinations of elastic and inelastic traffic. The room metric ρ (denoted by the star marker in the right plot) correctly reflects the bandwidth availability within the BSS.

VII. Evaluation of the resource-sharing protocol

In this section we first compare our distributed protocol with a centralized optimal solution, and next we evaluate its performance through simulation. To this end, we use the realistic scenario introduced in Section VI. We remark that our objective is to evaluate the performance of our scheme when the gateways are actually used, i.e., in the cases when techniques like sleep-on-idle [39] do not apply. Other studies report the potential energy saving of solution like sleep-on-idle [40]. If interested, the reader can take those values as lower bound.

A. Comparison with a centralized optimal solution

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 GWs $\mathcal{G}$, and given a set of active WS ($\mathcal{N}$), we want to minimize the number of "on" GWs ($\mathcal{G}$). We indicate by T_k the total throughput of station k, by S_j the saturation throughput of the BSS GW_j, and by R_{kj} the rate achievable by station k on GW j. Since the GW energy consumption is dominated by a fixed component, due to the fact that the device is "on", minimizing the number of "on" GWs is equivalent to minimizing the energy consumed in the RFN.

Thus we formulate the following optimization problem:

$$\min \sum_{j=1}^{\mathcal{G}} x_j \tag{2}$$

$$\sum_{k=1}^{\mathcal{N}} T_k y_{kj} < S_j \qquad \forall j \in \mathcal{G} \tag{3}$$

$$\sum_{j=1}^{\mathcal{G}} y_{kj} = 1 \qquad \forall k \in \mathcal{N} \tag{4}$$

$$y_{kj} \leq R_{kj} \quad \forall k \in \mathcal{N}, j \in \mathcal{G} \tag{5}$$

$$y_{kj} \leq x_j \quad \forall k \in \mathcal{N}, j \in \mathcal{G} \tag{6}$$

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 GW j, respectively.

$$N_j = \sum_{k=1}^{\mathcal{N}} y_{kj}, \quad R_j = \frac{\sum_{k=1}^{\mathcal{N}} y_{kj} R_{kj}}{\sum_{k=1}^{\mathcal{N}} y_{kj}}, \quad P_j = \frac{\sum_{k=1}^{\mathcal{N}} y_{kj} P_k}{\sum_{k=1}^{\mathcal{N}} y_{kj}}.$$

The objective function (2) minimizes the number of "on" GWs. $x_j \in \{0 = off, 1 = on\}$ represents the GW status, $y_{kj} \in \{0, 1\}$ indicates whether WS_k is associated with Gj ($y_{kj} = 1$) or not ($y_{kj} = 0$), The limit on the maximum achievable load on a given GW is enforced by condition (3), that bounds the total throughput generated by the assigned WS to the saturation throughput S_j. Condition (4) guarantees that a given WS is assigned to one and only one GW; condition (5) allows a WS to associate only to GWs that are within coverage; condition (6) ensures that a GW 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 (1) as a function of N_j, R_j and P_j. Hence, $\hat{S}_j = \alpha_1 N_j + \alpha_2 R_j + \alpha_3 P_j + c_u$, where α_1, α_2, α_3 are the linear coefficients, while c_u is set so that $\hat{S}_j > S_j$. Since this approximation considers a saturation throughput always greater than the achievable one, the optimal solution turns out to be a lower bound with respect to the number of "on" GWs.

Thus, we can rewrite the maximum load constraint (3) as:

$$\begin{aligned} \nu^u \sum_{k=1}^{\mathcal{N}} y_{kj} &< \alpha_1 N_j + \alpha_2 R_j + \alpha_3 P_j + c_u \\ \nu^u Y_S &< \alpha_1 Y_S + \alpha_2 \frac{\sum_{k=1}^{\mathcal{N}} y_{kj} R_{kj}}{Y_S} + \alpha_3 \frac{\sum_{k=1}^{\mathcal{N}} y_{kj} P_k}{Y_S} + c_u \\ \nu^u Y_S^2 &< \alpha_1 Y_S^2 + \alpha_2 \sum_{k=1}^{\mathcal{N}} y_{kj} R_{kj} + \alpha_3 \sum_{k=1}^{\mathcal{N}} y_{kj} P_k + c_u Y_S \end{aligned} \tag{7}$$

where $Y_S = \sum_{k=1}^{\mathcal{N}} y_{kj}$. By substituting condition (3) with (7), 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 is 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. 7, our protocol achieves close to optimal results, switching at most one gateway less than the optimal solution for throughputs up to 5 Mbps. This difference

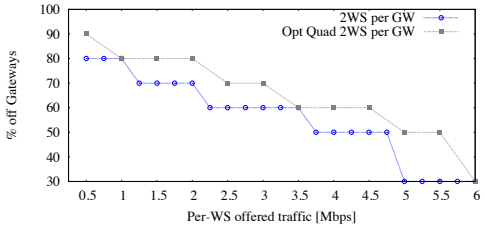


Fig. 7. Comparison between the wireless resource sharing protocol and the optimal solution.

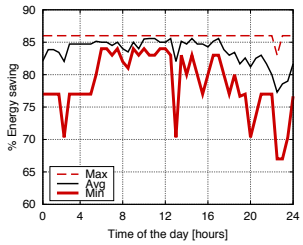


Fig. 8. Energy saving vs. hour of the day, obtained with real traffic traces.

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.

B. Energy efficiency evaluation

In order to evaluate the energy saving achieved by our scheme, we first model the GW energy consumption. 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 [41], 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 [42]. As for the rest of the GW device, previous studies [43] have observed that the power consumption of a home GW, 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 E_G of a GW 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,

$$E_G = T \cdot [\tau_{on} (P_G + P_i \tau_{on,i} + P_r \tau_{on,r} + P_t \tau_{on,t} + p_s) + \tau_{off} p_a]$$

where τ_{on} (τ_{off}) is the time fraction during which the GW is "on" ("off"), and $\tau_{on,i}$, $\tau_{on,r}$ and $\tau_{on,t}$ are the fractions of 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 [44].

In view of evaluating the energy saving achievable by our energy-efficient scheme in a real scenario, we collected packet-level traces from commercial GWs in 10 households for 1 week. First, we conducted real measurements by distributing a commercial GW to 10 households of a big European city. The GW is a Netgear WNRD-3600, which we equipped with the OpenWRT operating system and a sniffer that records all packets generated by all WSs. From the traffic traces, we extract the throughput of all WSs in every household for the different weekdays, split into 10-minute time bins. Then, we simulate the 10-household scenario depicted in Fig. 4 with the same environmental settings used in Section VI, but using variable UDP traffic sources that replicate the throughput measured at WSs. Finally, we use the energy model to compute the energy saving achieved by our solution, by comparing its energy consumption with respect to the normal condition, i.e., without any kind of cooperative federation scheme. Fig. 8 depicts the energy saving with respect to the case where GWs are always "on" throughout the day, averaged across all weekdays as well as over the 10 households. The plot also presents the minimum and maximum across a week of the energy saving averaged over all households. It can be seen that the average energy saving varies between 78% and 86% during the day, with higher values in the central hours of the day when people are usually out at work. Results (omitted for brevity) also show that the values of energy saving are consistent throughout the week, with a slight decrease on Sundays, i.e., when people spend more time at home.

C. Dynamic simulation results

All simulation parameters, including the channel model and the rate adaptation algorithm, are the same used in Section VI. We evaluate the benefits brought by our protocol in terms of energy saving, along with its performance in terms of load balancing and traffic throughput. As already said, we are interested to study the effectiveness of our protocol when gateways are active, i.e., when there is one (o more) WS, generating traffic using common applications, such as HTTP browsing/downloading and audio/video streaming.

We compute the load level of every BSS, the number of WSs associated to each GW, 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 GWs that the framework can switch off, as well as the energy consumed by the RFN with respect to the case where all GWs are always "on".

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 GWs 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 the left plot in Fig. 9); since the per-GW load is 3 Mbps, all GWs are in Light status. Then, between 60 and 68 s, every WS doubles its offered load (see the middle plot in Fig. 9), driving the "on" GWs into Heavy status.

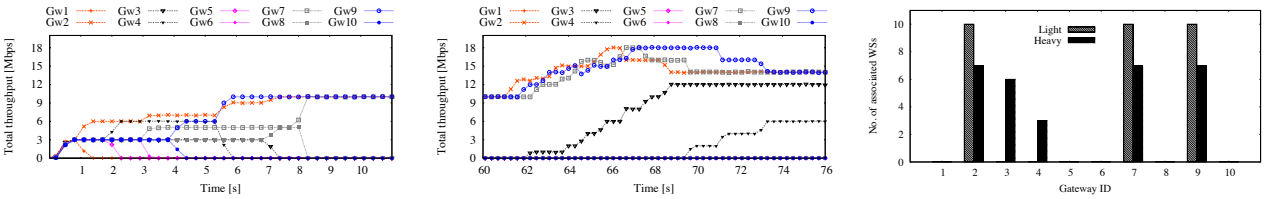


Fig. 9. Time evolution of the Gateways throughput in Light (left) and Heavy (middle) status, and WS distribution under Light and Heavy conditions (right).

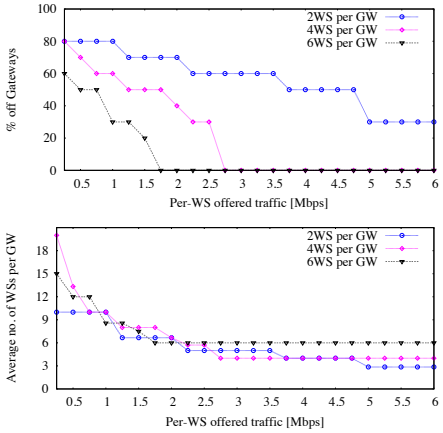


Fig. 10. Percentage of "off" Gateways (top) and average number of associated WSs per Gateway (bottom), as the WS offered load varies and for a different initial number of WSs per Gateway.

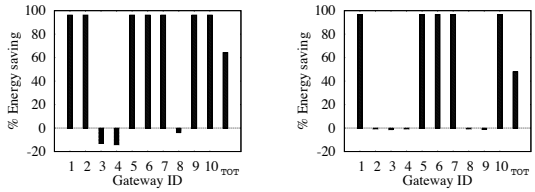


Fig. 12. Energy saving in the first (left) and second (right) scenario.

The temporal evolution of GW throughput, when all GWs are initially in Light status, is shown in the left plot of Fig. 9, where different marker/color combinations are used to represent the behavior of single GWs. The GWs that successfully carry out an offload procedure and become "off" correspond to downward curves, while GWs that associate relocated WSs see their throughput grow. A sample of a successful offload can be observed between 3 and 4 s, where a GW, upon switching itself off, relocates its three WSs to two other GWs whose throughput therefore increases. Eventually (at t=8.5 s), the federated network settles at 3 "on" GWs out of 10. Each "on" GW serves 10 WSs (see the right plot in Fig. 9) and is in Regular status.

Then, the middle plot in Fig. 9 shows the temporal evolution of the GWs throughput when a sudden traffic increase drives the three "on" GWs into Heavy status. As the WSs progressively double their offered load (between 60 and 68 s), two additional GWs 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" GWs, thus the second one is switched on only when the first can no longer associate WSs without moving into Heavy status itself. When all GWs are in Regular status (t=73 s), no further relocations occur and the network stabilizes at 5 "on" GWs. The three GWs that were "on" at the end of the period depicted in the top plot of Fig. 9 now have 7 associated WSs, while the first and the second GW that offered assistance have accepted 6 and 3 WSs, respectively, as shown in the bottom plot.

Next, we consider a different traffic scenario in Fig. 10 where initially all 10 GWs 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. The figure shows the percentage of "off" GWs, as well as the average number of WSs associated to a GW, upon reaching steady state. As expected, the number of switched off GWs 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 system dynamics.

We follow up with two traffic scenarios including TCP in order to evaluate the benefits of the offloading procedure in realistic settings. We consider that initially all GWs are "on" and have three associated WSs each.

The first scenario includes a mix of TCP and UDP (uplink/downlink) traffic flows. We set up all BSSs to initially

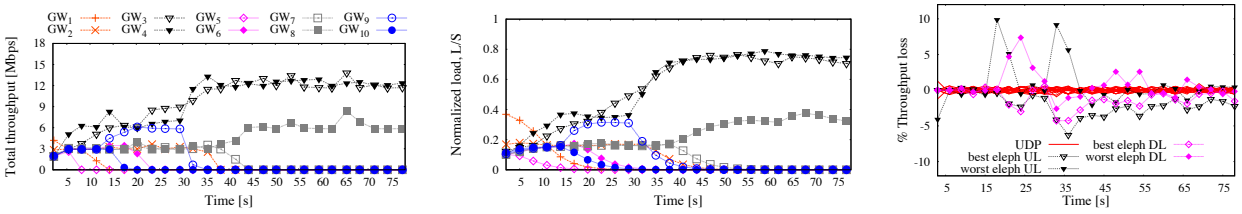


Fig. 11. First scenario: time evolution of the BSS throughput (left), of the normalized load (middle) and of the throughput loss (right).

feature the same mix of traffic. Specifically, out of the three initially associated to every GW, two WSs generate one 0.5-Mbps UDP stream each and are the destinations of, respectively, one elephant and one mouse TCP flow, while the third WS generates an elephant TCP flow. Elephant TCP flows share a 10-Mbps link in the wired section of the network.

The left plot in Fig. 11 depicts the total throughput within each BSS controlled by a GW and highlights that, being in Light status, the GWs try to relocate their WSs and turn themselves off. Around t=45 s, the RFN stabilizes with three GWs 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" GWs 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 GWs into the Heavy status (see Fig. 11, middle plot). As for the energy efficiency, the left plot in Fig. 12 depicts the saving achieved by each GW in the RFN, with respect to the case where all GWs 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 WSs experience. The right plot in Fig. 11 shows the difference between the throughput of the traffic flows in the initial configuration (i.e., all GWs "on" and three WSs per GW) and the one experienced when the resource sharing protocol is applied (i.e., only three "on" GWs). For clarity, in the case of TCP we only show 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 GWs, essentially earmarking those GWs as candidates to start a successful offloading procedure. The per-GW throughput and the normalized load are displayed in the left and middle plots of Fig. 13. As expected, the five less-loaded GWs are those that manage to offload their WSs to nearby GWs and turn themselves off, as shown by the downward curves in the bottom plot. The WS distribution over the five "on" GWs 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 GWs 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 (left plot of Fig. 13). As a result, when offload requests are dispatched, the five GWs 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 GW exhibits (left plot of Fig. 13), the changes are minimal. However, as depicted in the right plot of Fig. 13, single elephant TCP flows on those GWs 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. 12, are quite remarkable (almost 50%). Additionally, not having their total throughput affected, "on" GWs experience a negligible increase in their energy consumption.

Summary: The above results show that our approach exhibits high accuracy in estimating network load conditions and the GW capability to accommodate additional WSs. Thanks to such accuracy, our distributed resource sharing protocol can provide high energy savings, without impairing the performance experienced by 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.

VIII. Testbed Architecture

In order to demonstrate the feasibility of our proposal we implemented the load assessment algorithms and the distributed protocol on commodity hardware as proof-of-concept. We also designed a graphical interface that displays the most relevant quantities of the system. The topology in our testbed, shown in the left image of Fig. 14 and introduced in [45], is a federated network composed by three GWs, six WSs, a web

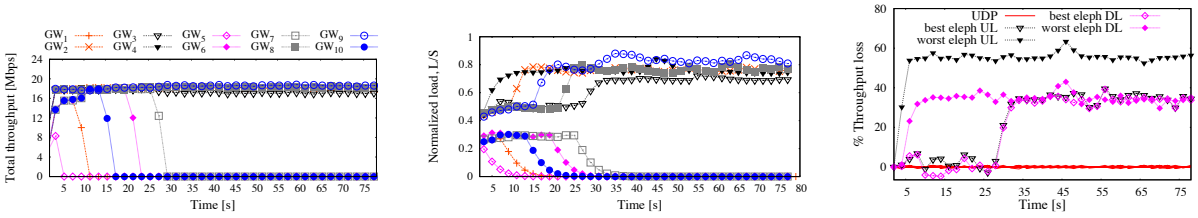


Fig. 13. Second scenario: time evolution of the BSS throughput (left), of the normalized load (middle) and of the throughput loss.

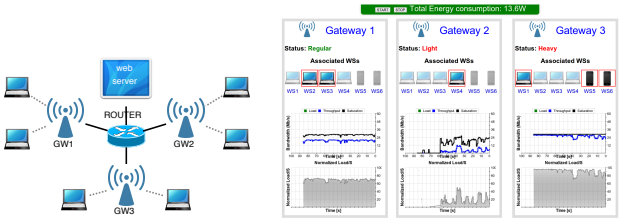


Fig. 14. Testbed architecture and monitoring interface.

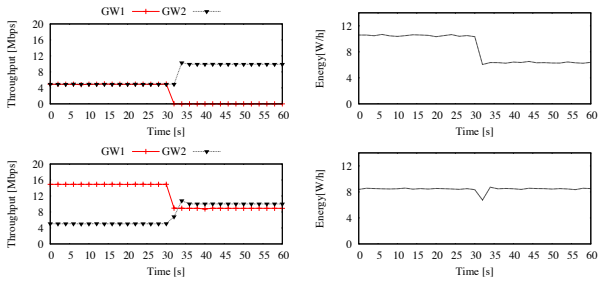


Fig. 15. Time evolution of the per-GW throughput and of the total energy consumption: light scenario (top) and heavy scenario (bottom).

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 include four laptops and two smartphones as WSs, so as to create the heterogeneity of a real-life residential environment. The router interconnects all the GWs and the web server.

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. 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. It also 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 the right image of Fig. 14.

A. 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 WSs are normal notebooks, namely the ASUS P52F, while the web server is a desktop PC, specifically the HP Compaq 8000 Elite. Both the WSs and the web server run Ubuntu 12.04. The router is a D-Link DES-1016A with 16 ports, while the smartphones are Samsung Galaxy SII with Android 2.3.

The passive traffic measurements needed by the protocol are implemented *on the GW* within the mac80211 module of the Linux wireless driver *compact-wireless* [37]. Note that, since we modified only the mac80211 module and not the GW hardware, such measurements work on any device. All measurements are made available to the application by the mac80211 module every 2 s. We implement the load assessment algorithms as well as the distributed sharing protocol as simple C programs. We modify the ODIN framework in order to use the VAP concept and to trigger the handover and specify the destination GW from our protocol. We use the dynamic channel switching procedure as proposed by 802.11h and implemented in the latest version of the linux wireless driver [37]. We did not implement any layer 3 handover management and we put all GWs as well as the WSs on the same network. Since we use a commodity gateway, the off/on switch is emulated and the total energy consumption is estimated as explained in Section VII-C.

B. Testbed results

We complement the evaluation of Section VII-C by studying the protocol behavior in presence of a sleeping GW. Thus, we start with two "on" GWs, namely GW1 and GW2 and one "off" GW, GW3. We generate synthetic UDP downlink traffic. 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 without waking up additional GWs; (2) one GW is in Heavy state and tries to relocate one WSs without waking up additional GWs. In the Light scenario, initially each "on" GW has two WSs, each receiving a downlink flow at 2.5 Mbps. After 30 s, GW1 switches "off" and hands over its WSs to GW2, which takes up the additional throughput almost instantly. Consequently, the energy consumption drops from 10.5 W/h to 6.2 W/h. Throughput and energy evolution are shown in the top plots in Fig. 15. As expected, GW3 does not wake up to help GW1. In the Heavy scenario, reported in the bottom plots in Fig. 15, GW1 has 4 WSs yielding a total of 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 relieves its congestion, while GW3 stays "off". Both GW1 and GW2 end up in Regular status and the energy consumption does not change, as expected.

IX. Conclusions

We designed a set of procedures aimed at managing underload and overload conditions in wireless GWs of federated households. After outlining some methodologies for load monitoring in presence of uplink/downlink elastic and inelastic traffic, we introduced a fully-distributed resource sharing protocol that allows (i) an underloaded GW to relocate all of its WSs and thus switch off; (ii) an overloaded GW to relocate some of its WSs and alleviate its status. We compared our proposal with a centralized optimal solution achieving close to optimal results. We extensively evaluated our protocol with simulation, showing its effectiveness in realistic federated

neighborhood scenarios. Furthermore we realized a small testbed as a proof-of concept, showing the applicability of our load metric as well as of our distributed protocol in commodity hardware.

As a final comment, 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 in order to tradeoff "greenness" and QoS according to their wishes.

REFERENCES

[1] S. Grover, M. S. Park, S. Sundaresan, S. Burnett, H. Kim, N. Feamster, "Peeking behind the NAT: An empirical study of home networks", *ACM IMC*, 2013.

[2] 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 saving optimization?," *IEEE Comm. Mag.*, vol. 49, no. 8, 2011.

[3] FIGARO - Future Internet Gateway-based Architecture of Residential Networks, http://http://www.ict-figaro.eu.

[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 Trans. on Veh. Tech.*, vol. 61, no. 5, 2012.

[10] S. Chen, Z.Yuan, G.-M. Muntean, "An energy-aware multipath-TCP-based content delivery scheme in heterogeneous wireless networks," *IEEE WCNC*, 2013.

[11] S. Andreev, A. Pyattaev, K. Johnsson, O. Galinina, Y. Koucheryavy, "Cellular traffic offloading onto network-assisted device-to-device connections," *IEEE Comm. Mag.*,, vol. 52, no .4, pp.20,31, 2014.

[12] D. Feng, *et al.*, "Device-to-device communications in cellular networks," *IEEE Comm. Mag.*,, vol. 52, no. 4, pp. 49–55, 2014.

[13] 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.

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

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

[16] E. Goma, *et al.* "Insomnia in the access or how to curb access network related energy consumption," *ACM SIGCOMM*, 2011.

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

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

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

[20] G. Bianchi, "Performance analysis of the IEEE 802.11 Distributed Coordination Function," *IEEE JSAC,* vol. 18, no. 3, pp. 535–547, 2000.

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

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

[23] L. Buttyan, J. Hubaux, "Stimulating cooperation in self-organizing mobile ad hoc networks," *ACM Mobile Networks and Applications*, vol. 8, no. 5, 2003.

[24] S. Ozdaglar, R. Srikant, "Incentives and pricing in communication networks," in *Algorithmic Game Theory*, Cambridge University Press, 2007.

[25] V. Srinivasan, P. Nuggehalli, C.F. Chiasserini, R.R. Rao, "An analytical approach to the study of cooperation in wireless ad hoc networks," *IEEE Trans. on Wireless Comm.*, vol. 4, n. 2, pp. 722–733, 2005.

[26] T. Clancy, M. Nakhjiri, V. Narayanan, L. Dondeti, "Handover key management and re-authentication problem statement," *IETF*, RFC 5169.

[27] L. Suresh, J. Schulz-Zander, R. Merz, A. Feldmann, T. Vazao,"Towards programmable enterprise WLANs with Odin," *ACM HOTSDN*, 2012.

[28] 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 Comm.*, vol. 15, no. 2, 2008.

[29] 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.

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

[31] C. Rossi, C. Casetti, C.-F. Chiasserini, "Bandwidth monitoring in multi-rate 802.11 WLANs with elastic traffic awareness," *IEEE GLOBECOM*, 2011.

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

[33] C. Rossi, C. Casetti, C.-F. Chiasserini, "A passive solution for interference estimation in WiFi Networks," *IEEE ADHOCNOW*, 2014.

[34] L. Di Cioccio, R. Teixeira, C. Rosenberg, "Measuring home networks with HomeNet profiler," *PAM* 2013.

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

[36] J. Bicket, "Bit-rate selection in wireless networks," *MIT Master's thesis*, 2005.

[37] Wireless driver, http://linuxwireless.org/

[38] A. Gupta, J. Min, I. Rhee, "WiFox: Scaling WiFi performance for large audience environments", *ACM CONEXT*, 2012.

[39] European Commission, "Code of conduct on energy consumption of broadband equipment," ver. 3, 2008.

[40] E. Bonetto, A. Finamore, M. M. Munaf, R. Fiandra, "Sleep mode at the edge: How much room is there?," *IEEE NETWORKS*, 2012

[41] 802.11nic specs. h18000.www1.hp.com/products/quickspecs/12510_ na/12510_na.PDF

[42] Waspmote http://www.libelium.com/support/waspmote

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

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

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

Claudio Rossi received his M.Sc. from the University of Illinois at Chicago and Politecnico di Torino in 2006. He worked in the industry as project manager for four years, and received his PhD from Politecnico di Torino in 2014. In 2012-2013 he was a visiting researcher at Telefonica I+D, Spain.

Claudio Casetti (M'05) received his PhD in Electronic Engineering in 1997 from Politecnico di Torino, where he is currently an Associate Professor. He has coauthored more than 140 papers in the fields of networking and holds three patents.

Carla-Fabiana Chiasserini (M'98, SM'09) received her Ph.D. in 2000 from Politecnico di Torino, where she is currently an Associate Professor. Her research interests include protocols and performance analysis of wireless networks. Dr. Chiasserini has published over 240 papers at major venues.

Carlo Borgiattino (S12) graduated in Telecommunication Engineering from Politecnico di Torino in 2011, where he is currently a Ph.D student. In 2013-2014 he has been a visiting researcher at Rutgers University, NJ.