Grant Agreement No.: 258378

FIGARO

Future Internet Gateway-based Architecture of Residential Networks

Instrument**:** **Collaborative Project**

Thematic Priority: **THEME [ICT-2009.1.1] The Network of the Future**

D3.2: Initial federated networking: interference models, handover mechanisms and multilink networking

Due date of deliverable: 30.09.2012

Actual submission date of *revised* version: 24.05.2013

Start date of project: October 1st 2010 Duration: 36 months

Project Manager: Henrik Lundgren, Technicolor R&D Paris

Abstract

This document presents the results of investigations performed during the second year of the FIGARO project relative to federation-oriented networking aspects. In particular, towards an improved access network capacity, a bandwidth aggregation and a multipath streaming framework are developed. Furthermore, federation-based as well as local wireless optimization techniques aiming at energy efficiency, interference reduction, and content-awareness are presented.

Project co-funded by the European Commission in the 7th Framework Programme (2007-2013)		
Dissemination Level		
PU	Public	✓
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Document Revision History

Version	Date	Description of change	Editor	Authors
V1.0	29.09.2012	Final version submitted to the EC	THRDF	THRDF, TID, TRDP, POLITO
V1.1	24.05.2013	Revised version with all papers in the appendix included along with copyright notice.	THRDF	THRDF, TID, TRDP, POLITO

Table of Contents

LIST OF ACRONYMS

AID	Association ID
AP	Access Point
ADSL	Asymmetric Digital Subscriber Line
BSS	Basic Service Set
CMT-SCTP	Concurrent Multipath Transfer SCTP
CTS	Clear To Send
DL	Downlink
DSL	Digital Subscriber Line
FTP	File Transfer Protocol
GW	Gateway
HTTP	Hypertext Transfer Protocol
HAS	HTTP Adaptive Streaming
LAN	Local Area Network
MAC	Medium Access Control
MIMO	Multiple Input Multiple Output
MPTCP	MultiPath TCP
PHY	Physical Layer
PSNR	Peak Signal to Noise Ratio
QoE	Quality of Experience
RAP	Random Access Point
RFN	Radio Federated Network
SCTP	Stream Control Transmission Protocol
SNR	Signal-to-Noise Ratio
TDMA	Time Division Multiple Access
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
UL	Uplink
VBR	Variable Bit Rate
VoD	Video on Demand
WLAN	Wireless LAN
Wi-Fi	Wireless Fidelity
WS	Wireless Station

EXECUTIVE SUMMARY

The work package 3 of the FIGARO project aims at developing an innovative, low energy, high performance networking framework based on the federation of residential gateways. The work package is composed of three main modules. The first two modules concentrate on the optimization of the throughput, and the energy consumption at the gateway federation level while the third module concentrates on the throughput optimization at the level of the wireless home network.

The low energy goal of the networking framework is addressed by a new paradigm based on the federation of home networks, leading to the creation of neighborhoods where network resources are shared and networked devices belonging to different users cooperate. The home network incorporates a smart gateway, equipped with a Wi-Fi AP for wireless Internet access that handles all the inward and outward network traffic. The energy savings are achieved by identifying the over-loaded, under-loaded, and normally-loaded gateways of the neighborhood and then by transferring the clients from the under-loaded gateways towards the normally-loaded gateways so that the under-loaded gateways can be shut-down. The solution requires no changes to the standard medium access control (MAC) protocol, nor to the clients, which may be even unaware of its adoption.

The high performance goal of the networking framework is addressed by the application of the resource pooling principle to the backhaul bandwidth of the federated gateways. The bandwidth aggregation is carried out at two levels: (i) At the Wi-Fi level, the gateway federation allows the APs of the gateways accept connections from their own subscribers but also from other APs in the neighborhood, therefore the gateways operate as Wi-Fi clients or APs. (ii) At the application layer, the HTTP Adaptive Streaming (HAS) technique is modified to improve the user's QoE by leveraging the multipath diversity made available due to the gateway federation. While this solution proved efficient, one major drawback is that it requires a modification of the HAS client software. Therefore, in the near future, solutions not requiring modifications to the HAS client will be explored.

Finally, the third module concentrates on performance optimization of the wireless home network. Two complimentary approaches are considered; First a content-aware PHY rate adaptation protocol for MIMO Wi-Fi networks. It improves end-users' quality of experience by optimizing wireless channel probing and PHY rate selection by exploiting the variable streaming rate information of a video. Second, we explore a new approach to improve wireless home network performance by combining MIMO with switched beam multi-sector directional antennas. We experimentally demonstrate that this combination can achieve performance improvement by reducing interference and finding better MIMO channel conditions.

At this stage, the first prototypes of the framework components are available in standalone test beds and are presented in this document. These components include an integrated federation-wide energy efficiency optimization module, a multi-path streaming module, and several wireless optimization modules

This document presents the aforementioned modules with some of the first evaluation results.

1 INTRODUCTION

Today's residential network environment poses significant challenges arising from the combination of heterogeneous technologies with content-based bandwidth hungry applications and emerging services such as home automation and tele-health care.

One concern that affects the residential side of the gateway is the proliferation of overlapping, always-on Wi-Fi APs in urban areas, which can cause inefficient bandwidth usage, energy waste and wireless interference. FIGARO investigates means to leverage the cooperation among federated APs to better cope with under-utilization and heavy load situations. Metrics and algorithms are developed to allow APs to self-assess their load status as well as their ability to help their neighbors. Signaling messages are defined to allow APs to request and initiate the handover of mobile stations when appropriate. In such manner, fully unused APs can power off completely, freeing radio resources and saving energy.

On the residential side, FIGARO addresses the limits of access broadband speeds available to end-users, which, nowadays, are still often one order of magnitude lower than those offered by Wi-Fi. One means to improve this situation is Wi-Fi backhaul aggregation, wherein users in range of neighbor APs can benefit from their neighbors' unused broadband bandwidth. While highly efficient, current approaches require the aggregation to be performed by the client device, rendering their deployment cost-prohibitive. In FIGARO, we propose an evolution of this approach, namely implementing the backhaul aggregation in the APs, using a single radio interface. While slightly less effective than the client-based alternative, the approach achieves the same purpose, without the need to upgrade the client devices.

On the Internet side, the rapid growth of content offerings and video capable devices but also the wide range of broadband speeds called for the development of novel delivery architectures and protocols capable of ensuring a continuous video experience without the need for guaranteed bandwidth. Due to its simplicity, streaming over HTTP has recently become a popular means to deliver over the top services to a wide range of devices. Nevertheless, the robustness of this solution comes at a price: the quality degrades as the network conditions deteriorate so that the service continuity can be guaranteed. In FIGARO, we extend HTTP streaming and propose to leverage the federation of gateways to exploit multipath communication to side-step the effects of network congestion and effectively increase the overall perceived quality.

At home, with the increased adoption of portable devices, notably smartphones and tablets, the communications are increasingly wireless. The development of Internet-based media offerings (VoD, catchup TV) and home media sources have set new expectations for in-home wireless delivery. However, while wireless technologies have continuously been improving in terms of datarate, the unpredictability of the shared medium still makes it challenging to deliver time-sensitive high-quality video, calling for dedicated efforts to secure wireless delivery of video streams within the home. In this deliverable, we propose to exploit content characteristics to drive probing and PHY rate selection. Our approach optimizes the PHY rate to the video streaming rate requirements and reduces rate adaptation overhead by adapting the frequency and timing of the wireless channel probing to the video streaming rate and the wireless channel variation. In addition, we experiment different multiplexing strategies to improve the support of multiple simultaneous delivery sessions.

As for wireless interference, one particular track studies interference reduction through the association of MIMO with directional antennas. We experiment with different configurations of MIMO and multisector antennas which reveal that, even without antenna directivity gain, the directionality of signals changes the MIMO channel structure and provides a way to both improve MIMO throughput performance and reduce interference.

At this stage of its lifetime, FIGARO has attained the following objectives as defined in the Description of Work document: (i) Design and develop an adaptive streaming framework which

exploits concurrent transport of media over multiple network links. (ii) Design and develop techniques for efficiently sharing neighborhood bandwidth while warranting fairness. (iii) Design and develop a novel wireless optimization framework that minimizes wireless interference and maximizes network Quality of Experience.

Notice that this document does not cover contributions to the IEEE 802.21 vertical handover standard, as it was originally planned in the DoW. Indeed, IEEE 802.21 standardization was under way at the time the FIGARO proposal was written, but in the meantime the standard was finalized and published, making any further contribution from FIGARO not required.

This document is organized as follows. Section 2 takes a federation wide optimization approach and presents an energy efficient resource sharing mechanism. Section 3 focuses on the access network and describes an AP based bandwidth aggregation mechanisms followed by our multipath video streaming method evaluation. Section 4 details two wireless network optimizations methods, the first one linking channel probing and PHY rate selection to the characteristics of video content, the second one exploring the benefits of associating MIMO with multisector antenna. Section 5 concludes and describes the next steps that will be taken by work package 3.

2 NETWORK OPTIMIZATION IN FEDERATED SETTINGS

The growing popularity of appliances and consumer devices embedding a Wi-Fi 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 federated home networking, leading to the creation of 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. Gateways are advanced home devices capable 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), hence provide wireless Internet access.

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.

In this 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.

Below, 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.

2.1 Energy Efficient Wireless Resource Sharing for Federated Residential Networks

Let us consider a set of residential units (e.g., houses or apartments), each of them equipped with a federated Gateway, with external and internal connectivity functions.

The 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. 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., an IEEE 802.15.4 card, that can be used as wake-on WLAN interface [11]. 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.

Self-load assessment through traffic measurement allows Gateways to classify their status as Light, *Regular*, or *Heavy*. In the Light status, traffic likely comes from background communications to/from the WSs, prompting the Gateway to try to 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.

2.1.1 BSS Assessment and Management

We denote by $\mathcal{N}$ the set of currently active nodes and by N its cardinality. Every measurement period and 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^u_k and ν^u_k, 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^d_k and ν^k_d, respectively. As in [13], 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, P and R. 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* [12], which we take as value of *BSS capacity*. We stress that, we compute S considering the node average behavior, thus accounting for the different air time that the WSs take.

Every measurement period and through running averages, the Gateway computes the capacity of the BSS it controls using the expression of the saturation throughput. 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. We report the procedure in Algorithm 1 below, where αS mitigates the effect of greedy elastic flows, while N_L upper bounds N for the Light status. 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 in Algorithm 2 only in the case where a single WS has to be relocated. 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.

2.1.2 Resource Sharing Protocol

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

1) A Gateway, *GW l*, which finds itself in Light status starts an offload procedure by sending 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. If their room-metric is greater than the value advertized by GW l, 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 sends a CTS so that all of its WSs will be frozen for a time τp 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 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 AID1. 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 evaluates through Algorithm 2 the room-metric for the possible combinations of candidate WSs.

Algorithm 1 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 $Regular$

Finally, it sends 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. Among the feasible solutions that allow *GW l* to relocate *all* of its WSs, the allocation maximizing the average data rate of the WSs is selected. To solve possible ties, preference is given to the allocation that minimizes the average room-metric. If a valid allocation is found, *GW l* sends 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 sends to all Gateways an ABORT message. 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.

Algorithm 2 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

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. Secondly, upon receiving an OFFLOAD REQUEST from *GW h*, an "on" Gateway not in Heavy status will always reply. If no viable relocation is found, *GW h* needs to wake up a neighboring "off" Gateway and ask for help. To accomplish this task in an energy-efficient manner, *GW h* sends through its low-power interface a wake-up sequence, 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.

2.1.3 Evaluation in residential scenario

We implemented our algorithms and protocol in the Omnet++ v4.1 simulator and evaluated their performance in a realistic scenario referring to a neighborhood located in the suburbs of Chicago, IL, shown in Figure 1.

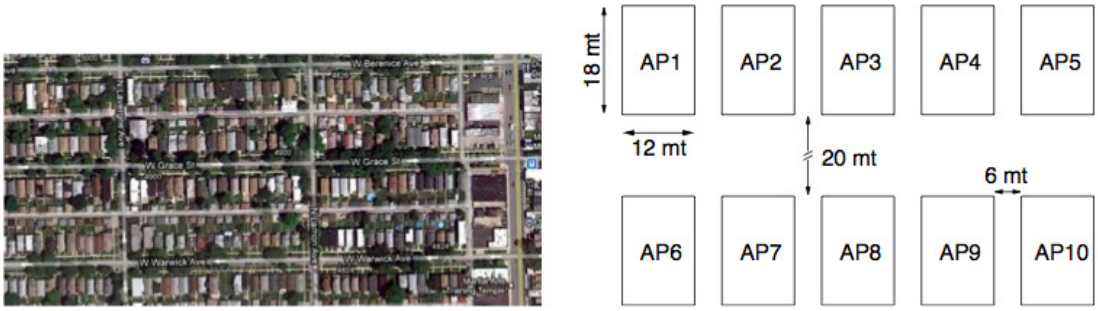


Figure 1. Federated detached house scenario.
Google Maps view (left) and schematic representation (right)

The RFN scenario includes 10 federated detached houses, each equipped with an IEEE 802.11g Gateway. 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 3s, α=0.2, N_L=10, T_L=0.4 and T_H=0.9, τ_r=0.3 s, and τ_p =0.1 s.

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. 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, 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. As for the rest of the Gateway device, previous studies [14] 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. Thus, we compute the energy consumption of a Gateway as follows:

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

where T is the observation period, 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.

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.

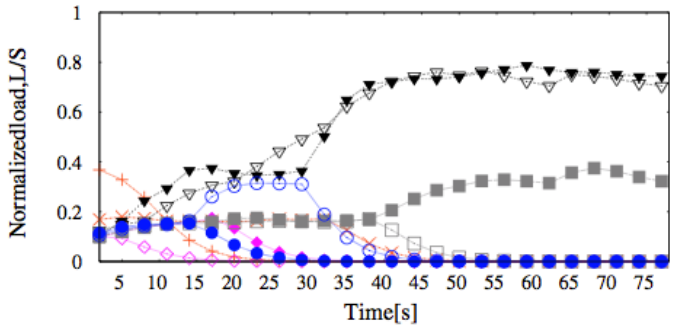


Figure 2. Time evolution of the normalized load

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 Figure 2. Note that 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.

As for the energy efficiency, the left plot in Figure 3 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 most of the time, the overall energy savings exceed 60%.

The right plot of Figure 3 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 above results show that our approach can provide high energy savings, without significantly degrading the performance experienced by the users.

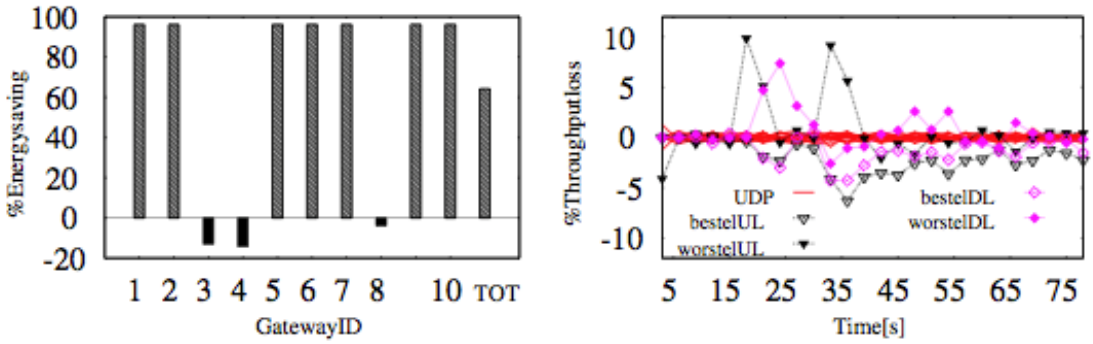


Figure 3. Energy savings (left) and throughput loss (right)

3 ACCESS NETWORK FEDERATION

Over the last decade, the Internet access speed has substantially increased in the developed regions, where the widespread xDSL technology allows access speeds of up to 50 Mbps to users that are close to the DSLAM. However, the situation is not so favorable for those users who are far from the DSLAM where the access speeds can go down to very low levels, usually under 5 Mbps. This situation is even worse in developing regions. As an example, in Colombia 97% of the DSL customers have access speeds lower than 4 Mbps. A means to increase the access speed for the customers, with minimal changes to the existing network deployment, is to aggregate the Wi-Fi backhaul bandwidth (resource pooling principle). However, the cost of the commercial deployment of current backhaul bandwidth aggregation solutions is prohibitive as they imply the modification of the operating system and the wireless driver of every possible Wi-Fi client (e.g. smart phones, tablets, laptops, etc.).

In FIGARO, we propose a cost-effective backhaul bandwidth aggregation solution (AGGRAP) based on a single-radio AP hosted by the Gateway that behaves as an AP to its clients, and as a client to neighboring APs. This solution has the advantage that the required modifications are limited to the Federated Gateways.

Furthermore, live or on-demand video streaming which ranks as one of the top services used at home are still challenging to deliver over broadband access links. The main reason is that the development of high-quality (3D, high definition video and sound) legal video streaming offers, such as Netflix, consistently renders the backhaul connectivity speeds insufficient to deliver the services to the customer.

We investigate the possibility of an application layer solution and present a modification to the HTTP Adaptive Streaming (HAS) technique in order to leverage the multipath diversity made available due to the gateway federation. The result is a better QoE for the user as the average quality is improved and the quality variations are damped. While this solution proves to be efficient, one major drawback is that it requires a modification of the HAS client software. Therefore, in the near future, solutions not requiring modifications to the HAS client will be explored.

During the remainder of the project, we will also explore the possibility to combine both techniques in a seamless way to bring superior performance.

3.1 Backhaul bandwidth aggregation

The research community has explored bandwidth aggregation on the Wi-Fi space since quite a time now [1][2]. State of the art client-based aggregation schemes propose the use of TDMA to enable a single-radio client to connect to several neighboring APs regardless of their frequency of operation.

Over duty cycles of 100 ms, the wireless client sequentially connects to all selected APs within range into a round robin fashion. The overall process is synthesized over Figure 4.

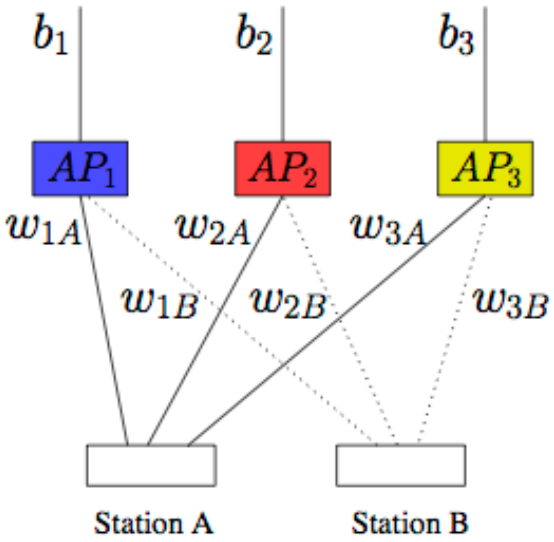


Figure 4. Typical bandwidth aggregation over Wi-Fi.

Using the integrated 802.11 Power saving feature, a 802.11 client is able to notify its period of absence to the APs it is connected to so that packets directed to it are buffered. A client performing aggregation appears to be sleeping in all APs but the one that is currently scheduled in the round robin cycle.

During the time in which the client is switching frequencies, it cannot send/transmit data. This time is referred as the switching time and for current hardware the typical expected switching time is around 1.5ms.

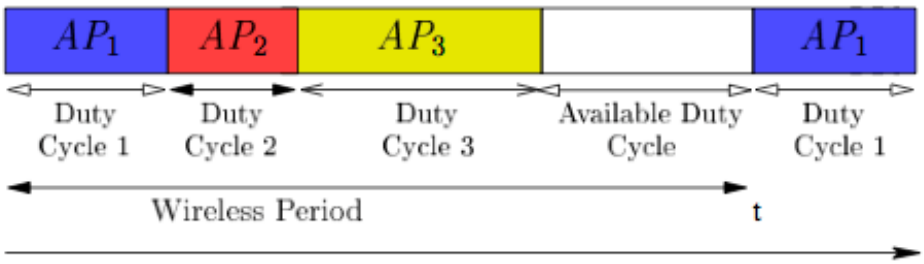


Figure 5. Time-division over the Wi-Fi airtime.

As the backhaul bandwidth capacity and the wireless link from the client to each AP might be different, client-based aggregation systems must optimize the percentage of time devoted to each AP. A basic strategy based on greedy schemes tend to lead to severe unfair throughput distribution, therefore weighted proportional fairness is enforced by altering the duty cycle time for each station contending, as indicated on Figure 5.

The THEMIS test bed [3]: has been build to evaluate the feasibility and the performance of such a scheme. To evaluate fairness, the FATVAP [2] scheme has been implemented on the same platform and used to establish a comparison term. FATVAP is built on the raw aggregation, where every

participant is greedily using the airtime to aggregate bandwidth.

As presented on Figure 6, the aggregated performance is evaluated in a realistic scenario where: station A is downloading peer-to-peer content ; station B is using normal web traffic and station C is placed far from the access point, suffering from a bad connectivity and therefore with a low physical datarate. As demonstrated on part a of the figure, traditional mechanisms such as FATVAP are exhibiting performance anomalies, related to: flow unfairness (A is taking over the available airtime due to the number of flows), topology unfairness (C is impacted by its position) and billing unfairness (B is paying as such A, but it is impacted by C and A). On the contrary, THEMIS is solving the unfairness by limited the bandwidth of A so that A and B shares the same bandwidth, and C can enjoy the bandwidth allowed by its physical capabilities.

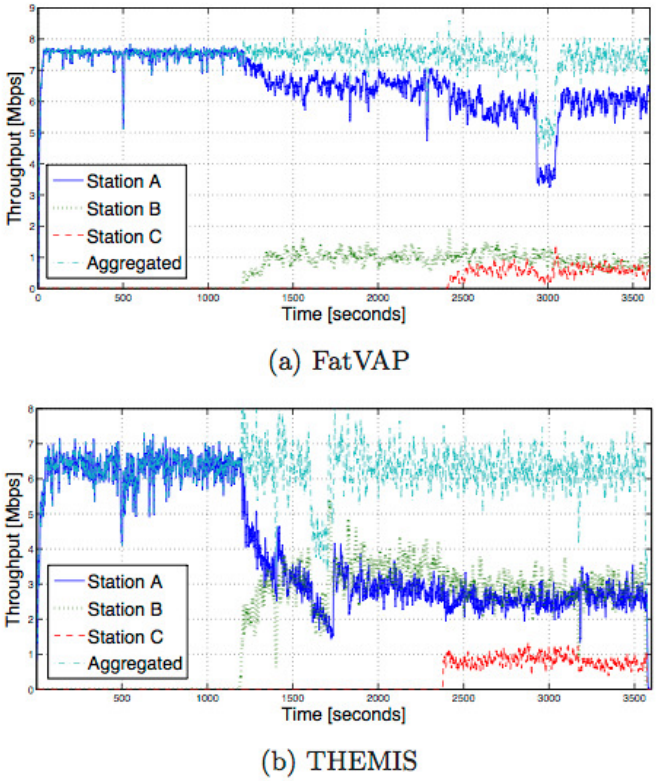


Figure 6. Aggregation performance according to FatVAP and THEMIS Schemes.

The client-based solution, while proved functional, still requires modification to the client Wi-Fi driver, and requires active collaboration of a network of gateway (for authorizing connections to them). This scheme is unachievable with the recent success with tablet and smartphones devices that have a high degree of integration and does not offer access to low-level drivers. Therefore, we introduce AGGRAP with the following characteristics:

- Single radio, i.e. the gateway does require only one radio, which is the current situation of most of the gateways deployed on the market.
- Wi-Fi clients operating using 802.11 standard.
- Limited overhead: the scheme is restricted to the immediate AP neighbors, i.e. we do not introduce more than 2 wireless hops.

- AGGRAP only aggregated unused bandwidth from neighboring APs.

The high level architecture is based on a federated community that enables its subscribers to connect to any AP participating in the sharing scheme. Beyond the traditional Wi-Fi access sharing, such as FON, in the architecture proposed it is not only users that connect to participating APs, but also other APs in order to make use of the additional backhaul capacity. Therefore, AGGRAP introduces the following mode of operations:

- In one mode, it is serving its own clients; (regular mode of operation of a gateway wireless interface).
- In the new mode, it is acting as a client to other participating APs in order to increase the uplink/downlink throughput to its clients. The latter function is triggered only in case of overuse of the primary backhaul of the gateway (80% threshold).

In the new mode, the gateway will scan for neighboring APs that can provide additional capacity, and associate with the 4-5 APs with best SNR with a minimum bandwidth of 5 Mbps. AGGRAP will use AP when these are loaded at less than 70%; moreover we allocate a 10% margin in order to avoid oscillation effect.

We started an implementation as proof of concept on Linux hardware. Our implementation involves a silencing mechanism to mute the station for a certain period of time in order to switch channel, combined with TDMA mode of operation.

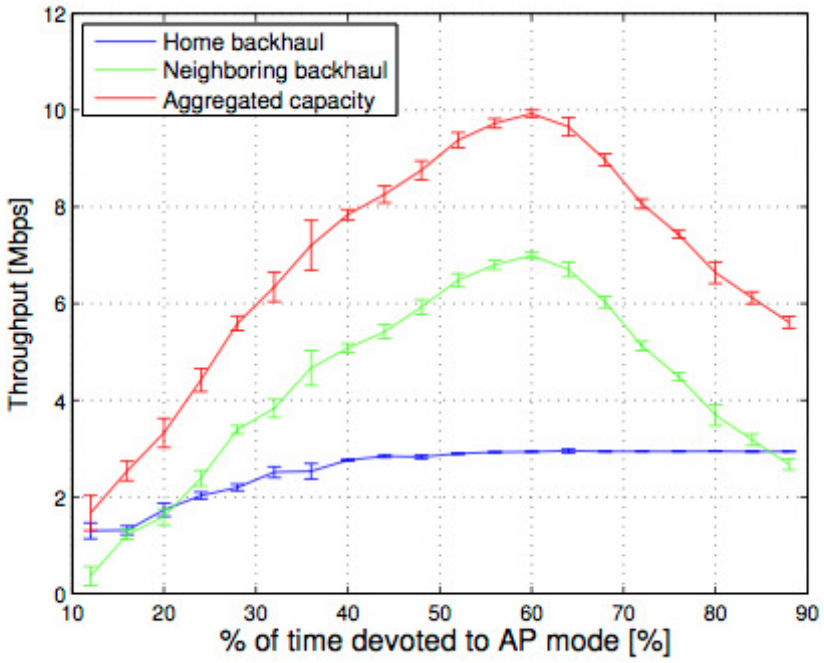


Figure 7 : AGGRAP preliminary results using a home backhaul of 3 Mbps, and a neighboring backhaul of 10Mbits.

As first preliminary results, we observed that the aggregated bandwidth remains constant despite the fraction of time allocated to AP or client mode, as presented on Figure 7. This simply indicates that the overhead of switching is fixed and only impacts overall aggregate throughput. The current proof of

concept implementation is requiring 1.5 ms to switch from one mode to another, but a lot of time is currently lost between the switch of frequency and the active notification that the AP is back from power saving. This period is currently evaluated to 6ms, and is due to the design of the interactions between the low level driver we are using and the high layers of the 802.11 stack. On a cycle of 100ms, this represents a 15% of the capacity without switching, but could be avoided with a proper design.

Therefore, the first results are encouraging and demonstrate the viability of the concept of aggregating the bandwidth in a federated way, with access points contributing to the scheme. More detailed results as well as a comparison between AP-based and client-based aggregation can be found in the full paper attached to this deliverable.

3.2 Multipath video streaming

While media streaming has been used and standardized in different forms for decades, HTTP Adaptive Streaming (HAS) has recently enjoyed a tremendous growth. This success is due to its simplicity and ability to serve a broad variety of viewing devices and most importantly its compatibility with the existing delivery infrastructure [28]. However an outstanding problem with HAS is that it is bound to adjust to current network capacity, resulting in uneven quality of experience, which is hardly acceptable for high value video (e.g. HD) playback.

In FIGARO, we propose to leverage multipath communication to level out the network performance variations and smoothen the viewing experience of HAS. The first envisaged direction for adding multipath capability is to implement HAS on top of a multipath transport protocol. Current multipath capable transport protocols are CMT-SCTP (Concurrent Multipath Transfer SCTP) and MPTCP (Multipath TCP). Unfortunately CMT-SCTP requires intermediate NATs to be upgraded, rendering its short term deployment impractical whereas MPTCP, which does not have the same limitations, is still under development. Therefore we focus on an application-layer approach to ensure that our solution is readily deployable with the current infrastructure. We will further explore MPTCP based approaches as the protocol matures throughout the rest of the project.

3.2.1 Multipath HTTP streaming

We have based our multipath adaptive streaming solution on a simple HAS implementation. Like with legacy HAS the multimedia content is pre-encoded in different bit rates (with a constant frame rate and resolution) with aligned RAPs (random access point), resulting in easily switchable, different quality streams which we will call representations. Each representation is split into segments of the same duration (e.g. 2 seconds) and made available for download by a standard HTTP server. In the case of legacy HAS, the client downloads the chunks in sequence while monitoring the available bandwidth, occasionally switching up or down to another representation according to network conditions.

The principal specificity of multipath adaptive streaming is that the client requests concurrently through each available path a slice of the next segment using the HTTP range header. Path diversity is available as soon as two interfaces are available at either of the client or the server side. Server side multipath can be implemented either by providing a server with multiple interfaces or by provisioning the content chunks to multiple servers. The two alternative network architectures envisaged for providing multipath capability are depicted on Figure 8.

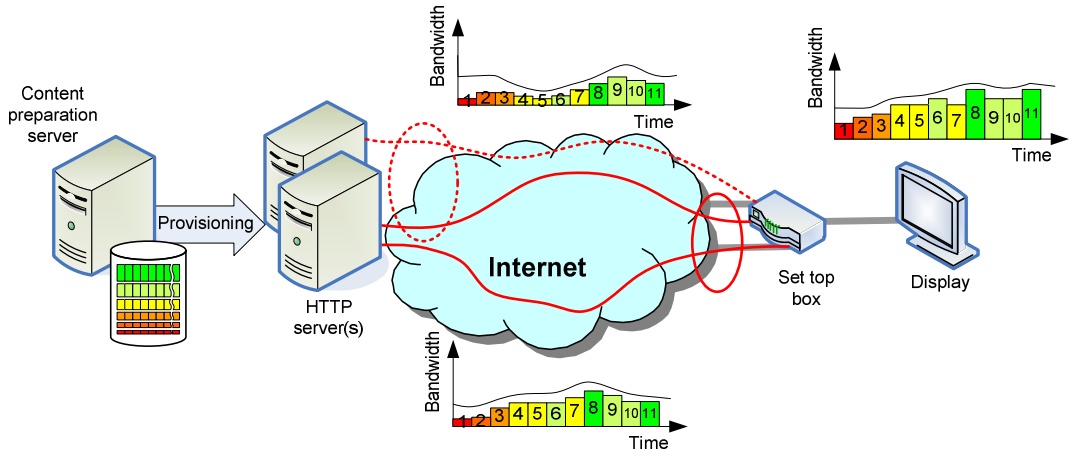


Figure 8. Different multipath delivery architectures

Client side multipath can be conveniently realized in the FIGARO context by using at least two close by residential gateways in a concurrent manner. We show in the following section how our extension of HAS leveraging the federation of gateways to exploit multipath communication helps avoiding the effects of network congestion and effectively increases the overall perceived quality.

3.2.2 Experimentation

We have performed a set of general experiments using network emulation. These were followed by field experiments using standard web servers and typical DSL and 3G access networks. The general results are summarized in Figure 9 which display the statistical distribution of the representations requested over time.

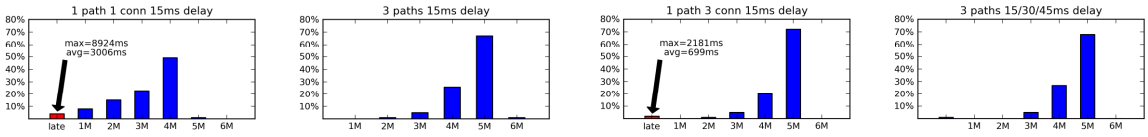


Figure 9. General results

Looking at the two leftmost charts, compared to the single path case, using 3 paths concurrently significantly raises the average playout bit rate and reduces the likelihood of large quality variations, which severely impact the viewing experience. In comparison, the third chart shows that parallel connections over the same path also increases the average playout bit rate, but fails to avoid congestion that hits all connections in a correlated manner. The fourth chart shows the relatively good performance of our solution with paths having unequal delays. Detailed results with network emulation were published in [29].

We then performed similar experiments using web hosting servers and typical 3G and ADSL connections to verify the results obtained with network emulation. Server side multipath exhibited similar results to the emulation scenarios, as shown on Figure 10. As can be seen when comparing the middle chart and rightmost charts, the addition of parallel connections towards a fast server performs worse than adding two slower but more uncorrelated paths.

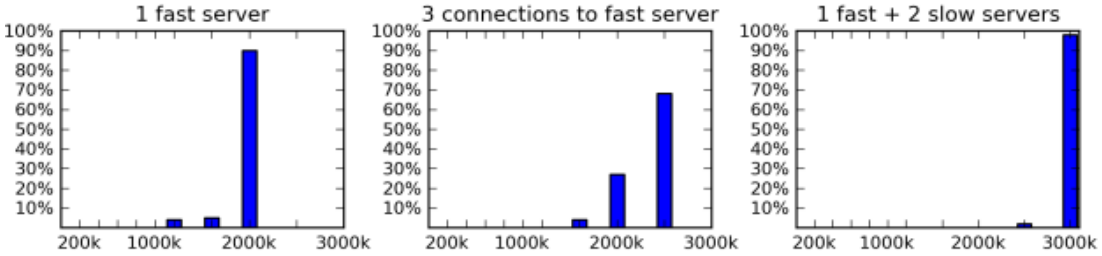


Figure 10. Single access to servers on the Internet

In yet another experiment, we have studied the effect of our multipath adaptive streaming solution with a single server and a multihomed client using two different Internet access links simultaneously. Figure 11 depicts the results when combining two ADSL access networks while accessing a single server. The two leftmost charts present the single-access performance and the right hand side chart presents the multipath performance achieved through the concurrent use of both accesses. In the single-access charts, the achieved playout bit rate is significantly lower than the plan (6Mbps downlink speed as reported by bandwidth testing tools), likely due to a bottleneck. In the dual-access case, the playout bit rate is approximately the summation of the single-access achieved bit rates, confirming that client side multipath is a practical solution for mitigating the impact of network bottlenecks.

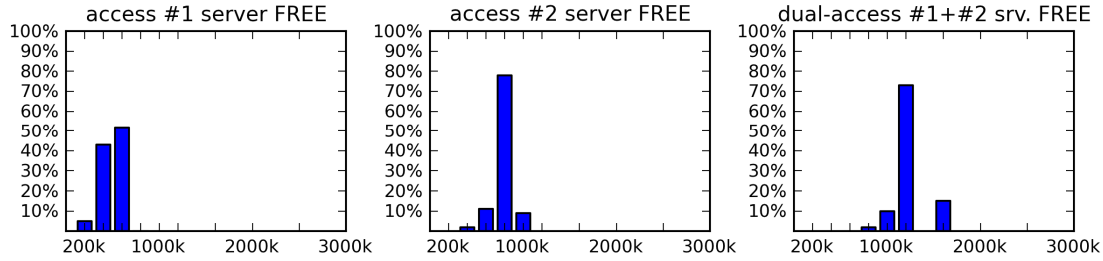


Figure 11. Single and dual access to a single server (Free)

Further experimental results are detailed in the full paper "Experimentation of Multipath HTTP Streaming over Internet" attached to this deliverable.

3.2.3 Discussion

The use of application-layer multipath has been shown to improve the performance of HAS. A simple implementation with as few as two available paths increases the average received bit rate and reduces the variance thereby dramatically improving the user experience. We have verified the applicability of multipath adaptive streaming to various scenarios through real Internet experimentations that confirm the results obtained in our emulation testbed. In addition we have shown the benefits of our solution when multipath are provided either at the server or client side.

While this solution proved efficient, one major drawback is that it requires a modification of the HAS client software. Therefore one logical axis of development will involve leveraging the residential gateway to perform multipath traffic relaying on behalf of unmodified clients. Further work will consist in investigating the feasibility of a transport layer multipath approach, as well as exploring the integration of our multipath streaming framework with the AP based bandwidth aggregation technology described in section 3.1.

4 WIRELESS HOME NETWORK OPTIMIZATION

Today, video dominates Internet traffic and by 2014 it is projected that nearly half of the video traffic will be 3D or 2D High Definition (HD) [21]. Recent video streaming technology standards such as H.264/MPEG-4 part 10 AVC [22] aim to reduce video streaming bandwidth requirements using Variable Bit Rate (VBR) video encoding. However, while average video streaming rate is reduced by efficient encoding of slow and moderate moving scenes, the peak rate remains high as it is determined by the full quality encoding of fast motion scenes. We have seen in the previous section how the FIGARO project addresses both bandwidth and reliability challenges of video delivery from a content-provider to the home gateway. However, users will consume a high fraction of this video over their wireless (Wi-Fi) home network. This imposes additional challenges for the (in-home) last hop of the video streaming service.

On one hand, the IEEE 802.11n [23] WLAN standard offers very high wireless physical-layer (PHY) data rates (up to 600 Mbps) using Multiple-Input Multiple-Output (MIMO) antenna technologies. MIMO extends the dimension of available radio resources to the time, frequency and space [4][5]. Adopted in many standard protocols [6][7] it has been already deployed to transport streamed voice and high-definition video traffic which requires high throughput. On the other hand, according to recent studies, the achieved goodput in practical 802.11n deployments can be significantly below the maximum 802.11 PHY data rates [24][25] due to not only MAC protocol overhead, but also issues such as content-obliviousness, sub-optimal PHY data rate selection, lack of favorable channel conditions for high MIMO PHY rates, and interference.

To this end, FIGARO addresses these issues with two complementary approaches to optimize the in-home wireless network. First, we introduce a content-aware PHY rate adaptation protocol for MIMO Wi-Fi networks. Its basic idea is to use the VBR video streaming rate to guide the adaptation of the wireless PHY rate. In addition, it reduces rate adaptation overhead by adapting the frequency and timing of the wireless channel probing to the video streaming rate and the wireless channel variation.

Second, we explore a novel avenue where we combine IEEE 802.11n MIMO with switched beam multi-sector directional antennas. Recently, it has been shown that directional antennas in indoor environments provide a few strong paths between nodes even without the line-of-sight path. The consequent benefits are empirically demonstrated, attracting renewed interests [15] [16] [17] [18] [19] [20]. However, the benefits and drawbacks of *directional antennas for MIMO* have not been fully studied so far, especially for indoor networks. We deploy a testbed and empirically investigate the potential performance gains that can be achieved by leveraging the possibility to obtain improved channel conditions and to mitigate interference using the directional antennas.

In the following sections, we summarize our work on these two approaches. Our published full papers on these topics are appended to this deliverable.

4.1 Content-aware Wireless Network Optimization

In this section we show that legacy rate adaptation protocols that are part of the 802.11 implementations suffer from probing-induced performance degradations that ultimately can cause video quality degradation. In response to these observed problems, we introduce VARA, a Video-Aware Rate Adaptation protocol that optimizes wireless channel probing and PHY rate selection by exploiting the VBR streaming rate information of a video. We experimentally evaluate VARA's performance using a user-space implementation in a wireless testbed equipped with off-the-shelf 802.11n cards.

4.1.1 Probing-induced capacity penalty

IEEE 802.11 protocols suffer from a fundamental problem that causes video quality degradation. This problem arises due to the probing overhead of the existing 802.11 PHY rate adaptation protocols. In order to select an appropriate PHY data rate, 802.11 rate adaptation mechanisms need to periodically probe the wireless channel. Probing can be either explicit or implicit. Explicit probing probes the channel using separate control packets; implicit probing probes the channel using packets from the ongoing data traffic. In order to reduce probing overhead, most existing ("legacy") rate adaptation protocols use implicit probing. During implicit probing, a few data packets are periodically sent at a higher PHY rate than the current PHY rate. If these transmissions succeed, the channel condition is estimated as adequate to support this higher PHY rate. Then, an even higher PHY rate is used to send the next few data packets. This process continues until a PHY rate where most probe transmissions fail. Then the previous (last successful) PHY rate is selected as the new rate, until the next probing period. The start of the next probing period is triggered by various factors, such as number of successful transmissions, measured packet error rate, time lags and so on, according to different implementations [26].

Increased probing frequency enables faster rate adjustments to channel changes (and thus potentially higher goodput), but also increases the overhead due to failed probe packets. These probing-induced losses result in channel capacity degradation due to binary exponential back-off and airtime consumption for retransmissions. We refer to this as the penalty in capacity utilization. It is important to note that probing and probe losses occur regardless of whether the current channel condition is good or bad. Unless the channel is good enough to support the highest PHY rate all the time, which is rare due to the dynamic property of wireless channels, the rate adaptation will periodically probe the channel to find a better rate to maximize the capacity. A variable channel naturally aggravates the probing overhead as it will cause the probing process to be triggered more frequently and result in increased probing-induced losses.

Although probe losses may not be perceptible in delay tolerant data applications, we show that they result in delays or losses of video frames at the application layer. This is detrimental to HD video streaming performance and can severely impact the perceived end-user quality of experience. We proceed to describe our protocol that aims to address this issue.

4.1.2 Video-Aware Rate Adaptation (VARA)

VARA is a cross-layer, video-aware PHY rate adaptation protocol. Its basic idea is to use a video streaming rate waveform from the application layer to guide the adaptation of the wireless PHY rate. For any stored video in a wireless home network video server, such a waveform can be easily generated with a play back during video recording. VARA divides time into variable-sized windows. For each window, VARA attempts to find a PHY rate that yields capacity above the peak video streaming rate in the window. VARA adapts the window sizes to take into account the wireless channel variability and probing overhead. More stable channel results in less frequent probing, and vice versa. As shown in the example of Figure 12, VARA consists of three algorithms invoked before the end of each window. First, Algorithm 1 computes the size of the next window based on past measurements of channel variability taken during the current window. Then, Algorithm 2 refines this size to satisfy rate requirements of the probing that Algorithm 3 might run during the next window. Thus, VARA eliminates the channel probing impact on the video stream by scheduling the probes during the low streaming rate periods. If the PHY rate of the current window cannot support the peak video streaming rate of the next window, Algorithm 3 probes the wireless channel for a suitable PHY rate. Note that, rather than aggressively trying to find the maximum PHY rate supported by the wireless channel (like all existing 802.11 rate adaptation protocols), VARA selects the most reliable PHY rate that supports the near-future peak streaming rate. At the beginning of the next window, Algorithm 1 sets the PHY rate found during the previous steps.

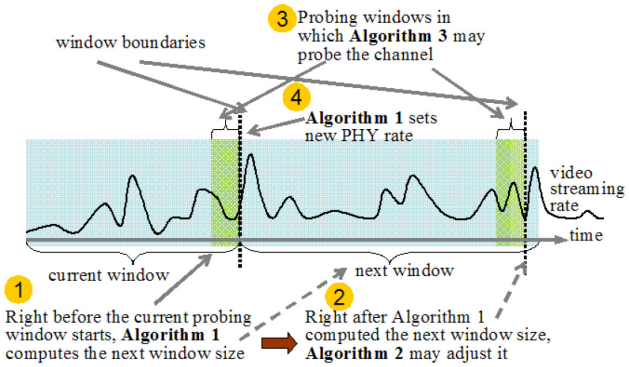


Figure 12. Illustration of VARA operation in a video streaming example.

When multiple video streams exist on a link, VARA treats their aggregated streaming rate as if from a single stream and the same algorithms for handling a single stream can be reused. However, an aggregate stream of multiple HD videos can have a very high peak rate. A higher peak rate makes it more difficult for VARA to find a matching PHY rate or it may exceed the wireless channel capacity. VARA addresses this problem using a multiplexing technique that we call Strategic Shifting. The idea is to strategically shift the starting times of the video stream waveforms such that the peak rate of the aggregate stream is minimized. This technique is further enhanced with two other shifting techniques that aim to minimize outage time or outage area with respect to a channel capacity target, which can be applied if ever there is no PHY rate that can sustain an upcoming peak streaming rate. The three algorithms and the multiplexing strategies introduced above are described in detail in the full paper version of this work, which is appended to this deliverable.

4.1.3 Evaluation

In this section we present one sample result from our evaluation of VARA. More results are available in the full version of the paper. We deploy a wireless testbed (

Figure **13**) in an indoor office environment with cubicles, meeting booths and regular offices. There is one access point (AP) at a fixed location and clients at different locations with different wireless channel conditions. Each testbed node is equipped with a Ralink RT2880 802.11n 2T3R MIMO mini-PCI card and three 5dBi omni-directional antennas. We use the RT2880 driver with RT2880iNIC Firmware version 2.0.0.1. The wireless cards are set to operate in an interference-free channel (108) in the 5GHz frequency band and use their default settings: 20MHz channel width, a Short Guard Interval (SGI) 800ns, and block acknowledgement and frame aggregation features deactivated. The total number of PHY rates including both SDM and STBC modes is 16.

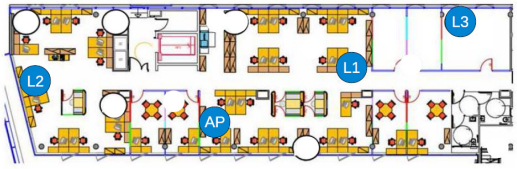


Figure 13. Testbed with Access Point and node locations.

Movie Name	Average rate	Peak rate	Variance
Panda1080p	10.26	26.12	28.71
Panda720p	5.96	15.94	10.82
MonsterAliens	5.06	14.64	5.22

Table 1. HD movie clips' properties. Rates are in Mbit/s

We first compare VARA's performance against the default auto rate of the RT2880 Ralink cards (the legacy rate adaptation algorithm) in a static environment. Therefore, we place both the AP and the clients in fixed locations as shown in

Figure **13**. Locations L1, L2, and L3 are selected to yield different wireless channel qualities. L1 has the best, L3 the worst. Table 1 shows the properties of different HD movie clips used in this experiment. Panda1080p represents the high streaming rate video, while Panda720p and MonsterAliens represent the medium and low streaming rate videos, respectively.

Each experiment run consists of two back-to-back streamings of each HD video between the AP and each client location, first using VARA and then auto rate. We repeat each run five times and show the average.

At the best channel quality location L1 auto rate supports all videos perfectly with zero loss. However, at location L2 it cannot support Panda1080p, the highest rate video. In contrast, VARA supports all videos perfectly at both locations L1 and L2. In L3, auto rate cannot support any of the videos perfectly. In contrast, VARA supports all videos perfectly except Panda1080p. This is expected because Panda1080p has a higher peak rate (26.12Mbps, Table 1) than the maximum capacity we measured at L3 (12.5Mbps). As depicted in Figure 14, with Panda1080p VARA achieves a 2% burst loss (lasting for two seconds) during the peak rate period, while auto rate results in a burst loss of 35% (lasting for six seconds) during the peak rate period. Figure 15 shows how these average burst loss rates translate to video quality as measured by average PSNR. We find that even a burst loss rate as small as 12%, as Panda720p suffers, can cause a significant degradation of video quality. In terms of subjective video quality, although a video with a PSNR of 25dB to 30dB could still be acceptable, it demonstrates obvious jitters, blocking and blurring. A video with a PSNR around 40dB is considered as a high quality video without any observable defect [27]. We observe that VARA achieves perfect PSNR for two videos, and increases PSNR about 50% for the high rate video over auto rate. Note that by definition, when there is no distortion, PSNR is infinity. For evaluation purposes we use a large number (60dB) to represent such perfect PSNR.

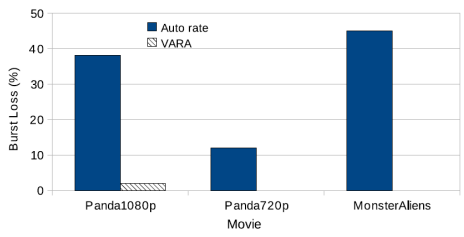


Figure 14. Packet loss at location L3.

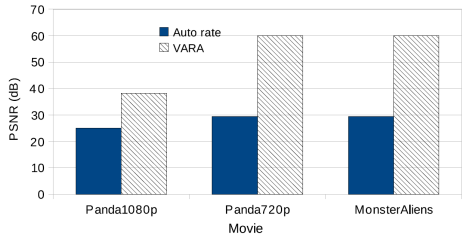


Figure 15. PSNR at location L3

The full paper version includes results for mobile scenarios, multiplexing videos using strategic shifting, as well as evaluation of different VARA protocol parameters such as probing windows sizes and fine-tuning adjustment needed to find low streaming periods.

4.2 Interference Mitigation using MIMO with directional antennas

Multiple Input Multiple Output (MIMO) has become one of the key technologies for wireless networks to accommodate the ever-increasing demands for higher throughput, stemming from

multimedia communications as well as new applications and usage patterns. MIMO combines multiple omni-directional antennas with signal processing to achieve capacity gains in rich scattering multi-path environments. Another technology that has long been receiving interests is directional antennas. These uses predetermined narrow beams to focus RF energy toward desired receivers. This achieves throughput gains and reduces interference. The trade-offs of combining these two technologies have not been well studied. It is important to understand that such a combination is not equal to MIMO beamforming. MIMO beamforming typically refers to a method where directional communication is achieved using signal processing and manipulation of phases in an antenna array. A fair amount of work has studied this technique for both indoor and outdoor networks. In contrast to such techniques, in our approach we equip standard MIMO (801.11n) network cards with switched beam multi-sector directional antennas. This approach works with off-the-shelf 802.11n cards and does not require any additional signal processing capabilities. The performance of such a system is not well understood, especially for indoor networks. On one hand, one may argue that directional antennas would decrease a degree of multi-path or signal scattering due to its narrow beams and consequently, decrease the capacity of MIMO links. On the other hand, however, the structural changes of propagation paths and signal increase by directivity could improve the capacity. In addition, a directional antenna allows to "turn off" transmission as well as reception in certain directions (by deactivating antenna sectors) and can thereby mitigate the interference problem in densely deployed home networks.

In this section, we summarize our work on experimental evaluation of the performance in MIMO wireless networks with directional antennas. Here, we focus on the interference mitigation properties. Our full paper appended to this deliverable contains additional detailed descriptions of our experimental methodology as well as results on throughput gains.

4.2.1 Experimental Setup

We use the Vivaldi multi-sector antenna developed by Technicolor Research. Each antenna has four antenna elements (sectors) printed on a PCB and covers the entire horizontal plane in the 5 GHz band. Any combination of sectors can be activated through a feeding network. This results in $(2^4-1) = 15$ different activation patterns, out of which one is four activated sectors forming an omni-directional pattern. We simply call it omni-mode. Figure 16 depicts the radiation patterns (excluding the feeding network) with one and four activated sectors, respectively. We see that the directivity gain is higher for antenna patterns with less activated sectors. However, the antenna feeding network (the switch for selecting active sectors) has been designed to introduce a feeding loss such that all antenna patterns exhibit approximately equal peak gains. This is a design choice made to reduce the so called directional hidden terminal problem [8][9][10].

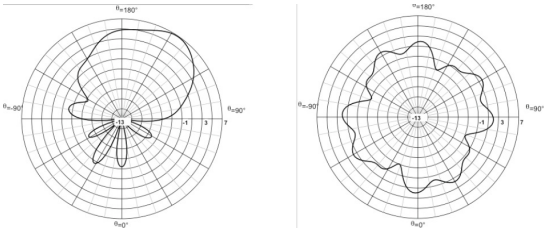


Figure 16. Antenna radiation pattern without feeding loss.

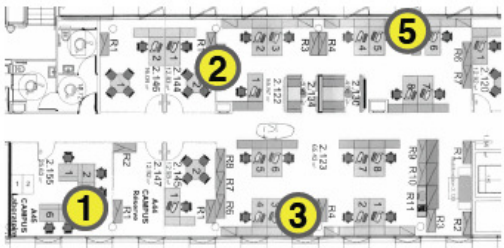


Figure 17. Testbed deployment with node locations.

Our wireless testbed shown in Figure 17 is deployed in a single floor in a typical office environment consisting of regular offices, glass meeting booths and cubicles. Due to the availability of multi-sector

antennas, we have only four nodes, but different topologies are emulated by varying the transmit power control.

Each testbed node is a PC with Intel Pentium (M) 1.73 GHz processor and 512 MB RAM, running Ubuntu Linux distribution version 10.04. The PC hosts a commercial mini-PCI 802.11n Network Interface Card (NIC) with a Ralink RT2880 chipset. A NIC includes three antenna ports with two radio chains supporting both SDM and STBC MIMO communications. We disable one of three ports to have 2-by-2 MIMO. Ralink NIC is also capable of adjusting Tx (transmit) power level to one of 18, 17, 15, 12, 9 and 6 dBm. During the experiments, the NICs are configured to operate in the 5.3 GHz band with 20MHz bandwidth with 400 ns guard-interval. During operation the NICs can select among basic 16 Modulation and Coding Sets (MCS). Each node has two Vivaldi multi-sector antennas connected to individual radio chains in a RT2880 NIC. Sector selection is controlled from the testbed node via a USB module interface connected to antenna selection switch.

4.2.2 Interference Properties Results

We present our results on interference properties of MIMO with multi-sector antennas.

Interference metric. We measure SNR and RSS for each pattern in the Tx activation sets, immediately followed by omni-mode. We use the difference $RSS_{diff} = RSS_x - RSS_{omni}$ as interference metric. A negative value means that sector activation pattern x reduces interference compared to omni-mode and increases spatial reuse. All measurements are performed at night, and the results are the average of five iterations. We perform the above experiment for both cases of absence and existence of antenna directivity gain (the directivity gain is emulated by increasing the Tx power level).

Interference without directivity gain. Figure 18 depicts the average RSS_{diff} at the neighborhood of each link. For example, for the ”link 1-2, 1 Sector” point, the average includes the RSS_{diff} values of all links 1-3 and 1-4 obtained for all 1 Sector activations of link 1-2. The RSS_{diff} values decrease as the number of active sectors decreases. With 1 Tx sector per multisector antenna, the interference over omni-mode can be reduced up to 12 dB at maximum (link 2-1) and 8 dB on average (link 2-5).

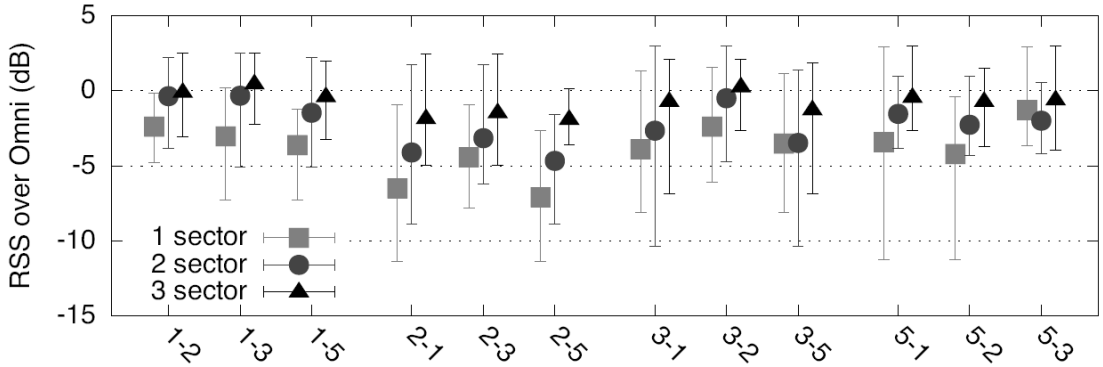


Figure 18. Average RSS_{diff} *without* antenna directivity gain: Average interference differences of Tx sector activations over omni mode at neighbors of each link.

Although sector activations reduce interference level, they may not necessarily increase throughput gain. Figure 19 depicts RSS_{diff} values in descending order of throughput gains. We observe that, for each number of activated sectors, the RSS_{diff} values are not related to the throughput gains. Especially for the highest throughput gains they remain constant. Thus, by selecting a number of activated sectors, it is possible to maximize throughput gain subject to a constant interference level, which is minimal when 1 Tx Sector activation patterns are considered.

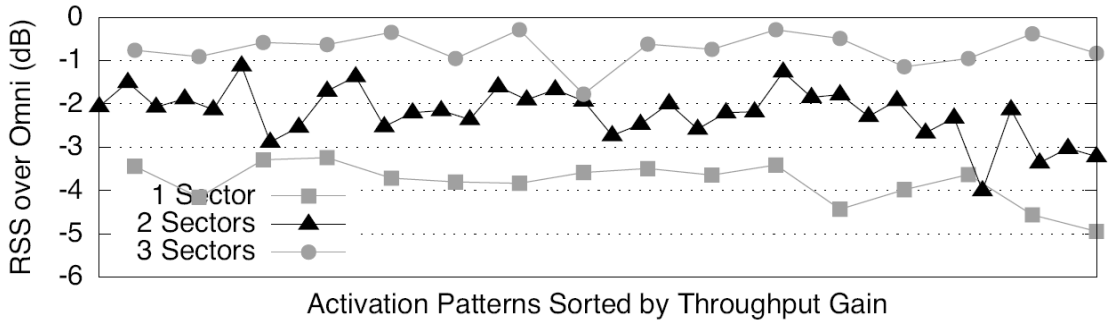


Figure 19. Average RSS_{diff} across all antenna patterns *without* antenna directivity gain. For each link, the patterns are first sorted by descending throughput gains and then the RSS_{diff} values with the same ranking are averaged.

We conclude that the interference level without antenna directivity gain is proportional to the number of activated sectors and has little correlation with the amount of throughput gain. Therefore, one can exploit spatial reuse in addition to throughput gain to enhance network-wide performance.

Interference with directivity gain. Figure 20 depicts the average RSS_{diff} at the neighborhood of each link when antenna directivity is present. In contrast to the case of no antenna directivity gain (Figure 18), across all links and Tx sector activation sets, the average RSS_{diff} is at most 3 dB, and within _7 dB range of the omni-mode. The average 3 dB interference reduction may be too low to make 802.11n medium access more aggressive1 and increase spatial reuse. Also, the interference reduction does not depend on the number of activated sectors. Although two sector activation has slightly less antenna directivity gain than single sector activation, its angular coverage is twice of a single activated sector. In addition, the use of multiple antennas in MIMO systems renders more chances for a receiver to capture more signal paths and thus receive stronger signals than a single antenna receiver. Thus, when antenna directivity is present, the interference reduction is small and does not depend on the number of active sectors.

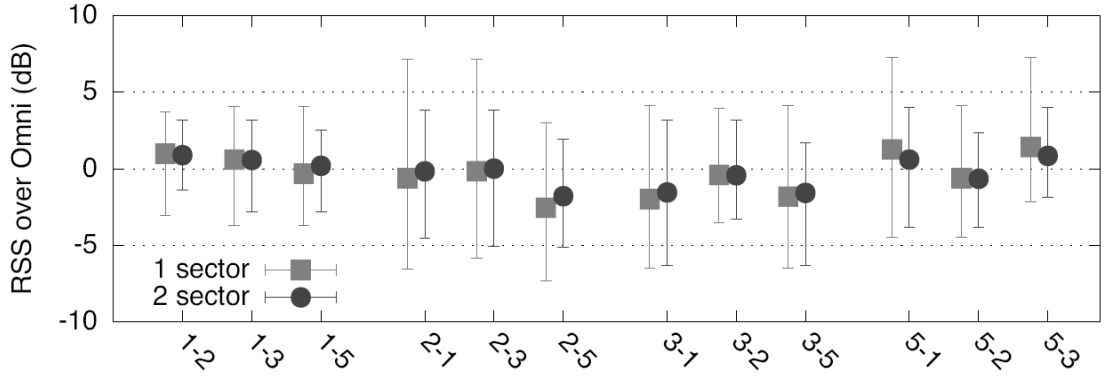


Figure 20. Average RSS_{diff} *with* antenna directivity gain: Average interference differences of Tx sector activations over omni mode at neighbors of each link.

4.2.3 Discussion

Based on our observations so far, two design choices are worth being considered: (1) multi-sector antennas without directivity gain for MIMO to utilize throughput gain and spatial reuse and (2) multi-

sector antennas with directivity gain for MIMO to increase throughput gain. Design (1) can exploit spatial reuse in addition to throughput gain to enhance network-wide performance. However, spatial reuse may introduce additional hidden terminals and require coordination mechanisms among different operating links.

In FIGARO, we can address this across home networks by leveraging the collaboration across a federated neighbourhood. Design (2) comes with modest spatial reuse, but leads to higher throughput gains and simpler protocol design. It does not require coordination among different links for sector activation. Moreover, it does not require availability of SNR information, as even RSS can be used to find a good activation pattern, providing backward compatibility with vanilla 802.11n.

5 SUMMARY AND NEXT STEPS

In this document we have presented the progress made so far regarding federated and optimized home networking. We have proposed several resource optimization and performance enhancing methods and exposed results of their evaluation. Among these, we proposed a method for efficiently allocating wireless resources between neighborhood residential gateways, which equates to identical service for the end users while reducing both energy consumption and electromagnetic pollution. We proposed and evaluated two techniques for aggregating access network capacity, resulting in a better quality of experience when consuming high bandwidth content. Finally we presented wireless optimization techniques for the in-home environment, which were shown to improve wireless video delivery and help reducing wireless interference.

As a follow up, we will further study the collaboration opportunities within the residential gateways federation, aiming towards collaborative optimization. We will also further evaluate and quantify the benefits of the technologies presented in this deliverable. We will finally also focus on demonstrating the strength of federated networking and explore the possibility of integrating backhaul bandwidth aggregation with our multipath streaming framework.

6 REFERENCES

[1] Chandra, R.; Bahl, P.; Bahl P., *MultiNet: Connecting to Multiple IEEE 802.11 Networks Using a Single Wireless Card*. IEEE Infocom, Hong Kong, http://www.cs.cornell.edu/people/ranveer/multi net_infocom.pdf, 2004.

[2] Srikanth Kandula, Kate Ching-Ju Lin, Tural Badirkhanli, and Dina Katabi. 2008. FatVAP: aggregating AP backhaul capacity to maximize throughput. In *Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation* (NSDI'08), Jon Crowcroft and Mike Dahlin (Eds.). USENIX Association, Berkeley, CA, USA, 89-104.

[3] Giustiniano, Domenico and Goma, Eduard and Lopez Toledo, Alberto and Dangerfield, Ian and Morillo, Julian and Rodriguez, Pablo, Fair WLAN backhaul aggregation, Proceedings of the sixteenth annual international conference on Mobile computing and networking, MobiCom '10, Chicago, Illinois, USA

[4] I. E. Telatar, "Capacity of multi-antenna Gaussian channels,"European Trans. Telecommun., vol. 10, no. 6, pp. 585–595,Nov./Dec. 1999.

[5] A. J. Paulraj, D. A. Gore, R. U. Nabar, and H. Bolcskei, "An overview of MIMO communications- A key to gigabit wireless," Proceedings of IEEE, vol. 92, pp. 198–218, 2004.

[6] 3GPP Workgroup R1, Requirements for further advancements for -UTRA (LTE-Advanced), 3GPP TR 36.913, Std.

[7] IEEE 802.11n, Part 11: wireless LAN medium access control (MAC) and physical layer (PHY) specifications: Enhancement for Higher Throughput, Standard Specifications, Std., Oct. 2009.

[8] Y.-B. Ko, V. Shankarkumar, and N. H. Vaidya, "Medium access control protocols using directional antennas in ad hoc networks," in Proc. IEEE INFOCOM, Mar. 2000.

[9] R. R. Choudhury, X. Yang, R. Ramanathan, and N. H. Vaidya, Using directional antennas for medium access control in ad hoc networks," in Proc. ACM MobiCom, 2002.

[10] T. Korakis, G. Jakllari, and L. Tassiulas, "A MAC protocol for full exploitation of directional antennas in ad-hoc wireless networks," in Proc. ACM MobiHoc, 2003.

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

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

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

[14] T. Nguyen, A. Black, "Preliminary study on power consump- tion of typical home network devices," *Tech. Rep. 071011A*, 2007.

[15] M. Blanco, R. Kokku, K. Ramachandran, S. Rangarajan, and K. Sundaresan, "On the effectiveness of switched beam antennas in indoor wireless environments," in Proc. Passive and Active Measurement Conference (PAM'08), Apr. 2008.

[16] A. P. Subramanian, H. Lundgren, and T. Salonidis, "Experimental chatacterization of sectorized antennas in dense 802.11 wireless mesh networks," in Proc. ACM MobiHoc, May 2009.

[17] A. P. Subramanian, H. Lundgren, T. Salonidis, and D. Towsley, "Topology control protocol using sectorized antennas in dense 802.11 wireless networks," in Proc. IEEE Internat. Conf. Network Protocol (ICNP), Sept. 2009.

[18] X. Liu, M. Kaminsky, K. Papagiannaki, and S. Seshan, "DIRC: Increasing indoor wireless capacity using directional antennas," in Proc. ACM SIGCOMM, Aug. 2009.

[19] X. Liu, A. Sheth, M. Kaminsky, K. Papagiannaki, S. Seshan, and P. Steenkiste, "Pushing the envelope of indoor wireless spatial reuse using directional access points and clients," in Proc. ACM MobiCom, Sept. 2010.

[20] A. A. Sani, L. Zhong, and A. Sabharwal, "Directional antenna diversity for mobile devices: characterizations and solutions," in Proc. ACM MobiCom, Sept. 2010.

[21] Cisco System, "Cisco visual network index: Forecast and methodology, 2009 - 2014," Cisco While Paper, 2010.

[22] ISO/IEC 1449610:2003, "Coding of Audiovisual ObjectsPart 10: Advanced Video Coding, also ITU-T Recommendation H.264 Advanced video coding for generic audiovisual services." 2003.

[23] IEEE 802.11 Working Group, "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications – Amendment 5: Enhancements for Higher Throughputs," 2009.

[24] K. Pelechrinis, T. Salonidis, H. Lundgren, and N. Vaidya, "Experimental Characterization of 802.11n Link

Quality at High Rates," in Proc. ACM WiNTECH, 2010.
[25] I. Pefkianakis, Y. Hu, S. H. Wong, H. Yang, and S. Lu, "MIMO Rate Adaptation in 802.11n Wireless Networks," in Proc. ACM MobiCom, 2010.
[26] S. H. Y. Wong, H. Yang, S. Lu, and V. Bharghavan, "Robust Rate Adaptation for 802.11 Wireless Networks," in Proc. ACM MobiCom, 2006.
[27] Y. Wang, "Survey of objective video quality measurements," in Technical report. Worcester Polytechnic Institute, 2006.
[28] Ali C. Begen, Tankut Akgul, and Mark Baugher, "Watching Video over the Web" in IEEE Internet Computing, 2011
[29] S. Gouache, G. Bichot, A. Bsila, C. Howson, "Distributed & Adaptive HTTP Streaming" in IEEE International Conference on Multimedia and Expo (ICME), 2011

APPENDIX: INCLUDED PAPERS

C. Rossi, C. Casetti, C.-F. Chiasserini, *Energy-efficient Wireless Resource Sharing for Federated Residential Networks*, IEEE WoWMoM 2012, San Francisco, CA, USA, June 2012.

D. Giustiniano, E. Goma, A. Lopez Toledo, Julian Morillo, Ian Dangerfield, Pablo Rodriguez, *Fair WLAN Backhaul Aggregation*, ACM Mobicom 2010, Chicago, September 2010.

E. Goma, A. Lozano, *AGGRAP: Making WiFi Backhaul Aggregation Practical*, to be submitted.

S. Gouache, G. Bichot, C. Howson, *Experimentation of Multipath HTTP Streaming over Internet*, NEM Summit, September 2011.

A. Chan, H. Lundgren, T. Salonidis, *Video-Aware Rate Adaptation for MIMO WLANs*, IEEE ICNP, October 2011.

T. H. Kim, T. Salonidis, H. Lundgren, *MIMO Wireless Networks with Directional Antennas in Indoor Environments*, IEEE INFOCOM (mini-conference), Orlando, FL, March 2012.

Papers are included in this appendix with proper permissions. Specifically for IEEE papers, see the copyright notice below:

Energy-efficient Wireless Resource Sharing for Federated Residential Networks

Claudio Rossi, Claudio Casetti, Carla-Fabiana Chiasserini

***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 federated APs could address these problems (i) by allowing underused devices to hand over their wireless stations to nearby APs and temporarily switch off, (ii) by balancing the load of stations among APs and thus offloading congested APs. In this paper, we outline a framework that, leveraging a multipurpose gateway with AP capabilities in every household, allows such cooperation through the monitoring of local wireless resources and wireless station offloading toward other federated gateways. Our results show that, in realistic residential settings, the proposed framework yields an energy saving between 45% and 65%, while providing load balancing and meeting the user expectations in terms of throughput.**

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, 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*. 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. Gateways are advanced home devices capable of providing wireless Internet access and of offering storage and multimedia services, including audio and video real-time streaming.

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 [1], the channel busy (or, equivalently, idle) time [2], or the aggregated BSS throughput [3], 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., [4], [5], either apply only to self-estimation of the downlink bandwidth availability, or require changes in the WSs.

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 [6]–[8]. Also, WS relocation can be either Gateway- or WS-initiated, as in [9]. 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, [10] has suggested the use of TDMA techniques to let WSs access multiple APs at a time, while [5], [11] aggregate the bandwidth available at different APs and balance their load. Such approaches, differing in scope from our work, require modification in the WSs.

II. 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 [12].

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 [13], [14] thus accounting for the collision probability as well as for the different data rates and

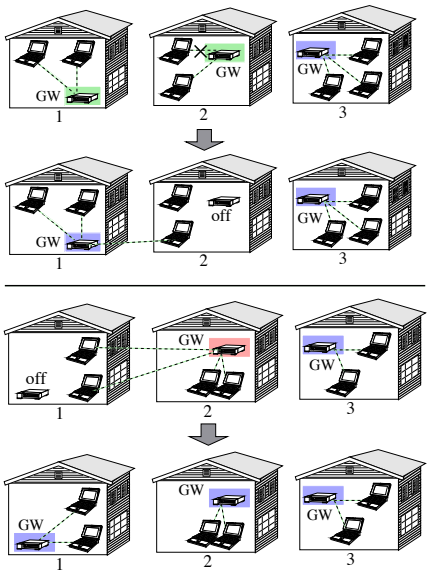


Fig. 1. Energy saving (top) and load balancing (bottom) by WS offloading.

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. 1, where a 3-house neighborhood is displayed. The first example (top part of Fig. 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. 1 is reached.

Our second example (bottom part of Fig. 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. 1 (i.e., all Gateways 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 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 [15].

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 [12]. 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 paper, 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 [16], and will not be discussed further in this paper.

In order to achieve seamless handover for ongoing applications, the network must support mobile IP or a similar technology. Recent studies [17], [18] 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 [19]. The design of a next-generation global mobility protocol capable of guaranteeing seamless handover with little or no disruption time is outside the scope of this paper.

IV. BSS 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 [13], [14], 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.

A. 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 [20], 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 [21], 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*. The saturation throughput is defined as in [14], which extends the original Bianchi's model [13] in presence of errors due to channel propagation conditions:

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

In (1), τ is the probability that a node (either a WS or the Gateway) accesses the medium at a generic time slot[1], 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 the average collision duration is concerned, its exact computation would require the Gateway to be aware of the number of nodes that are hidden with respect to each other. The works in [13], [14] do not account for hidden WSs and the approaches proposed in the literature are not viable in our set up, as we do not require the Gateway 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 packet of maximum payload size among those observed by the Gateway 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 $E[T]$ are derived following [14] and for completeness can be found in our technical report [22], while p_e is estimated by the Gateway based on the modulations used for the transmissions in the measurement period, their associated signal-to-noise ratio, and assuming independent bit errors on the channel.

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 (1) depend on the payload, data rate and access rate of each single WS.

[1]Considering a slotted time is the main approximation made in [13], [14].

Algorithm 1 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 2 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

B. 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 (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. 1, 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.

C. 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. 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 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 (1). 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.

V. 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. 1, and, if in Light or Heavy status, it carries out an offload procedure, as summarized in Fig. 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. 2, 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

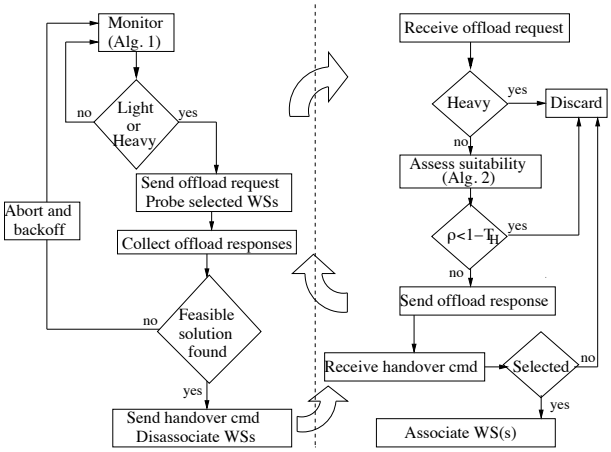


Fig. 2. Flow chart of the offload procedure: Gateway asking for help (left) and Gateway receiving the help request (right).

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[2]. Such a procedure allows a Gateway, 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 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. 2 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. Among the feasible solutions that allow GW l to relocate *all* of its WSs, the allocation maximizing the average data rate of the WSs is selected. To solve possible ties, preference is given to the allocation that minimizes the average room-metric. The rational is that, firstly, 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. Secondly, we want as many WSs as possible to associate to Gateways that have already a high traffic load and leave out those that are likely to reach a Light status, hence to switch themselves off.

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 [23].

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. Thus, upon

[2]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.

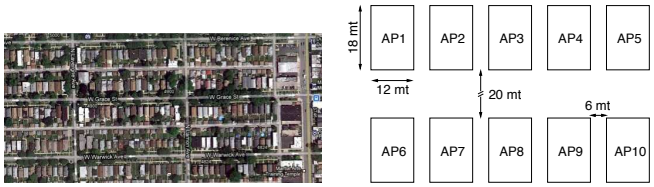


Fig. 3. Federated detached houses scenario: Google view of the area (left) and abstract representation (right).

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.

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.

VI. Evaluation in a Residential Scenario

We implemented our algorithms and protocol in the Omnet++ v4.1 simulator and 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, 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 [24]; also, we implemented the automatic data rate adaptation scheme AARF [25]. 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.

A. Gateway status and suitability assessment

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

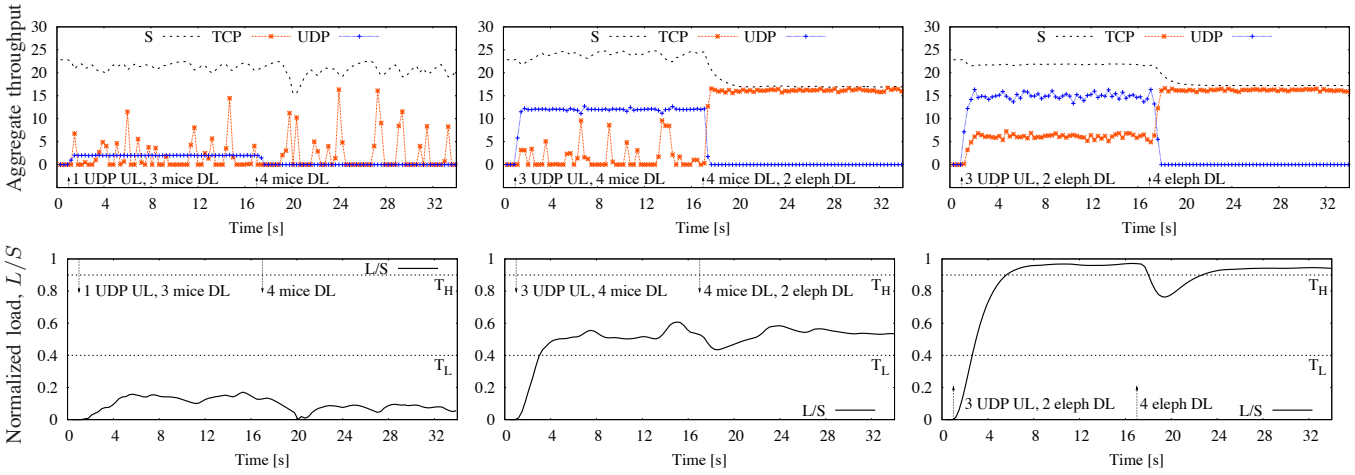
The first scenario we study corresponds to an underloaded BSS, which includes a WS originating an uplink 2-Mbps UDP flow, and three other WSs that are the destinations of one mouse TPC flow each. The aggregate throughput for elastic and inelastic traffic is depicted in the top plot of Fig. 4(a), along 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. 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. 4(b) shows that the Regular status is always detected.

Finally, Fig. 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. 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



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

Fig. 4. 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.

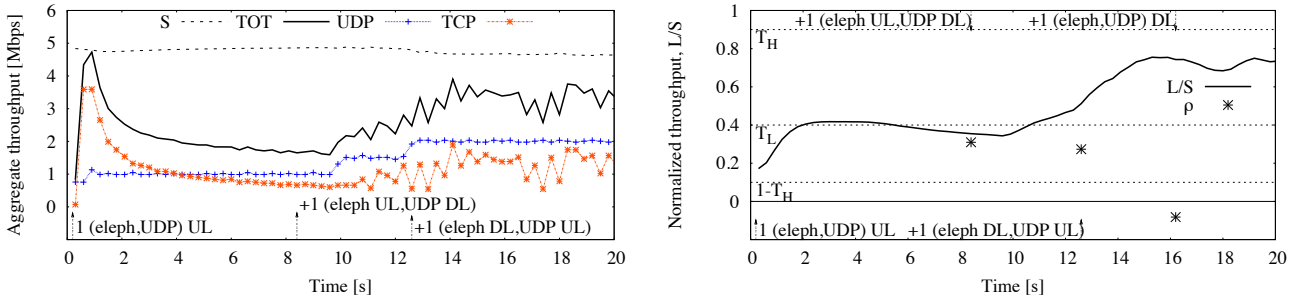


Fig. 5. Relocation requests arrive at the Gateway for WSs that are sources and destinations of elastic and inelastic traffic. The room-metric ρ (denoted by the star marker in the right plot) correctly reflects the bandwidth availability within the BSS.

status threshold), and ρ correctly drops below $(1 - T_H)$. The WS is therefore rejected.

B. 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.

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 [26], 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 [27]. As for the rest of the Gateway device, previous studies [28] 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,

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

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 [29].

Next, we consider two sample traffic scenarios that allow us to evaluate the benefits of the offloading procedure. 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.

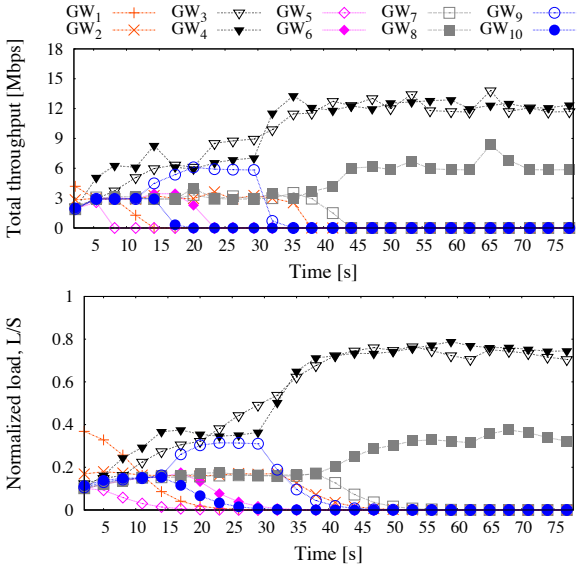


Fig. 6. First scenario: time evolution of the BSS throughput (top) and of the normalized load (bottom).

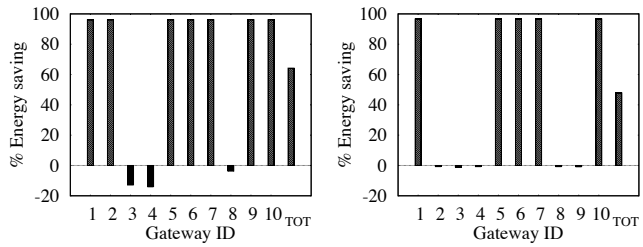


Fig. 7. Energy saving in the first (left) and second (right) scenario.

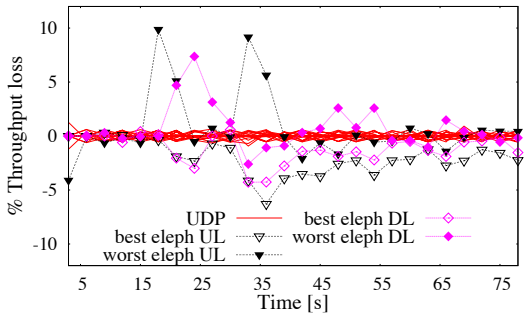


Fig. 8. First scenario: time evolution of the BSS throughput loss.

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. 6 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. 6, bottom plot). As for the energy efficiency, the left plot in Fig. 7 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. 8 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. 9. 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. 9). 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. 9), the changes are minimal. However, as depicted in Fig. 10, 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. 7, 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.

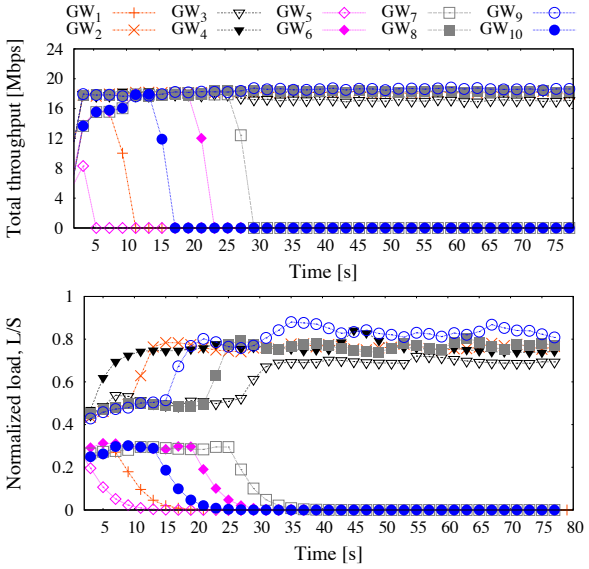


Fig. 9. Second scenario: time evolution of the BSS throughput (top) and of the normalized load (bottom).

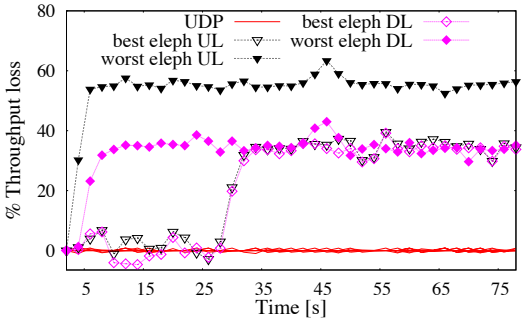


Fig. 10. Second scenario: time evolution of the BSS throughput loss.

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.

VII. Conclusion and Future Work

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. Further developments will address the implementation of our solution in real devices, along with experimental measurements and hands-on energy saving quantification, as well as the evaluation of the theoretical maximum energy saving, resulting from the optimal allocation of the active WSs to federated GWs.

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 (such as the α parameter) in order to tradeoff "greenness" and QoS according to their wishes.

References

[1] N. Blefari Melazzi, D. Di Sorte, M. Femminella, G. Reali, "Toward an autonomic control of wireless access networks," *Globecom*, 2005.
[2] A.P. Jardosh, K.N. Ramachandran, K.C. Almeroth, E. Belding, "Understanding congestion in IEEE 802.11b wireless networks," *Sigcomm*, 2005.
[3] H. Velayos, V. Aleo, G. Karlsson, "Load balancing in overlapping wireless LAN cells," *ICC*, 2004.
[4] S.Vasudevan, K. Papagiannaki, C. Diot, J. Kurose, D. Towsley, "Facilitating access point selection in IEEE 802.11," *Sigcomm*, 2005.
[5] D. Giustiniano, E. Goma, A. Lopez, J. Morillo, I. Dangerfield, P. Rodriguez, "Fair WLAN backhaul aggregation," *MobiCom*, 2010.
[6] A. Jardosh, K. Papagiannaki, E. Belding, K. Almeroth, G. Iannaccone, B. Vinnakota, "Green WLANs: On-demand WLAN infrastructure," *Mobile Networks and Applications*, vol. 14, no. 6, 2009.
[7] A. Jardosh, G. Iannaccone, K. Papagiannaki, B. Vinnakota, "Towards an energy-star WLAN infrastructure," *HotMobile*, 2007.
[8] J. Lorincz, A. Capone, M. Bogarelli, D. Begusić, "Heuristic approach for optimized energy savings in wireless access networks," *SoftCOM*, 2010.
[9] E. Goma, *et al*. "Insomnia in the Access or How to Curb Access Network Related Energy Consumption," *Sigcomm*, 2011.
[10] D. Giustiniano, E. Goma, A. Lopez Toledo, P. Rodriguez, "WiSwitcher: An efficient client for managing multiple APs," *PRESTO*, 2009.
[11] S. Kandula, K. C. Lin, T. Badirkhanli, D. Katabi, "FatVAP: Aggregating AP backhaul capacity to maximize throughput," *USENIX*, 2008.
[12] N. Mishra, K. Chebrolu, B. Rama, A. Patha, "Wake-on-WLAN," *WWW*, 2006.
[13] G. Bianchi, "Performance analysis of the IEEE 802.11 Distributed Coordination Function," *JSAC*, vol. 18, no. 3, 2000.
[14] P. Chatzimisios, A.C. Boucouvalas, V. Vistas, "Performance analysis of the IEEE 802.11 DCF in presence of transmission errors," *ICC*, 2004.
[15] X. Ai, V. Srinivasan, C.-K. Tham, "Wi-Sh: A simple, robust credit based Wi-Fi community network," *Infocom*, 2009.
[16] T. Clancy, M. Nakhjiri, V. Narayanan, L. Dondeti, "Handover Key Management and Re-Authentication Problem Statement," RFC 5169, IETF.
[17] K. Kong, W. Lee, Y. Han, M. Shin, H. You, "Mobility management for all-IP mobile networks: Mobile IPv6 vs. proxy mobile IPv6," *Wireless Communciations*, vol. 15, no. 2, 2008.
[18] 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.
[19] Mobile IP traversal of Network Address Translation (NAT) devices, http://tools.ietf.org/html/rfc3519.
[20] C. Rossi, C. Casetti, C.-F. Chiasserini, G. Rondini, "A new metric for admission control in multi-rate 802.11 WLANs," *WONS*, 2011.
[21] C. Rossi, C. Casetti, C.-F. Chiasserini, "Bandwidth Monitoring in Multi-rate 802.11 WLANs with Elastic Traffic Awareness," *GLOBECOM*, 2011.
[22] C. Rossi, "Computation of the BSS Saturation Throughput," *Tech. Rep.*, www.telematica.polito.it/casetti/SaturationThroughput.pdf, 2011.
[23] IEEE Std 802.11v, Part 11: WLAN MAC and PHY specifications. Amendment 8: IEEE 802.11 Wireless Network Management, Feb. 2011.
[24] T. Chrysikos, G. Georgopoulos, S. Kotsopoulos, "Site-specific validation of ITU indoor path loss model at 2.4 GHz," *WoWMoM*, 2009.
[25] M. Lacage, H. Manshaei, T. Turletti, "IEEE 802.11 rate adaptation: A practical approach," *MSWiM*, 2004.
[26] h18000.www1.hp.com/products/quickspecs/12510_na/12510_na.PDF
[27] http://www.libelium.com/support/waspmote
[28] T. Nguyen, A. Black, "Preliminary study on power consumption of typical home network devices," *Tech. Rep. 071011A*, 2007.
[29] http://www.cisco.com/en/US/prod/collateral/wireless/ps5678/ps11983/data_sheet_c78-686782.pdf

Fair WLAN Backhaul Aggregation

Domenico Giustiniano[†*], Eduard Goma[†], Alberto Lopez Toledo[†‖], Ian Dangerfield[‡¶], Julian Morillo[§¶], and Pablo Rodriguez[†]
[†]Telefonica Research, [‡]Hamilton Institute, [§]Universitat Politècnica de Catalunya
{domenic,goma,alopezt,pablorr}@tid.es, ian.dangerfield@nuim.ie, jmorillo@ac.upc.edu

ABSTRACT

Aggregating multiple 802.11 Access Point (AP) backhauls using a single-radio WLAN card has been considered as a way of bypassing the backhaul capacity limit. However, current AP aggregation solutions greedily maximize the individual station throughput without taking fairness into account. This can lead to grossly unfair throughput distributions, which can discourage user participation and severely limit commercial deployability.

Motivated by this problem, we present THEMIS, a single-radio station that performs multi-AP backhaul aggregation in a fair and distributed way, without requiring any change in the network. We implement THEMIS on commodity hardware, evaluate it extensively through controlled experimental tests, and validate it in a deployment spanning 3 floors of a multistory building. THEMIS is being used in a commercial trial by a major broadband provider to its customers.

Categories and Subject Descriptors

C.2.5 [**Computer Communication Networks**]: Local and Wide-Area Networks—*Access schemes*; C.2.1 [**Computer Communication Networks**]: Network Architecture and Design—*Wireless communication*

General Terms

Design, Experimentation, Performance.

1. INTRODUCTION

In urban environments, residential users can potentially see multiple 802.11 APs in range with high quality [1], usually connected to broadband links. As the speeds of 802.11 WLAN are typically an order of magnitude higher than those of standard broadband connections, one can use a single 802.11 wireless card to aggregate the bandwidth of multiple AP backhauls in range by virtualizing the card and cycling over the APs in a TDMA fashion. The result of such multi-AP aggregation scheme is that stations will connect to several APs in range and share their backhaul connections.

In that scenario, using an aggregation scheme like FatVAP [2], where stations greedily maximize their individual throughput, may lead to severe unfair situations. Fairness is important because it can impact individual users performance and reduce its applicability on a commercial setting. For example, a station that is unluckily located in an area with only one AP in range, can see its throughput significantly lowered by other stations sharing the same AP, even if those stations could get spare bandwidth from other APs. This is what we call *topology unfairness*. We argue that providing a fair distribution of throughput even in such heterogeneous situations is crucial to maintain a certain level of service across all users. Without some form of fairness, the perceived value of the system is severely reduced, and users will not participate.

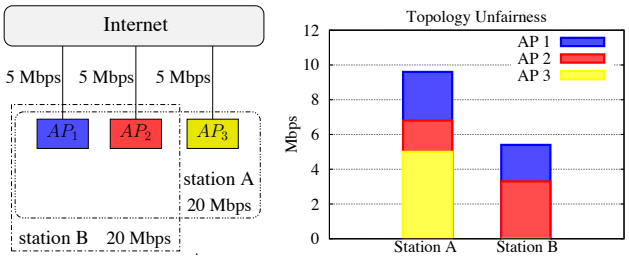
Other unfairness situations also exist. For instance, stations using applications with many TCP flows, such as P2P, can severely affect the performance of other stations running single-flow applications such as Web downloads. We call this situation *flow distribution unfairness*, and can result in some stations obtaining much less throughput than what they would obtain without sharing.

Another example of unfairness could appear in a scenario where customers with different subscription plans share their broadband links. For instance, fast broadband customers (that pay more than slow broadband customers), should obtain a greater share of the spare backhaul capacity. If this is not enforced, customers may be inclined to buy slower (and cheaper) broadband connections and free-ride on their neighbors' spare bandwidth. This is a typical "tragedy of the commons" example: people tend to over-exploit the shared resource by minimizing their contribution (their broadband contracted speed), ultimately cannibalizing the shared resource. Moreover, this eliminates the incentive of an ISP to deploy the sharing system, because it threats its business model. We call this *billing unfairness* (Section 2).

The above fairness scenarios can have a dramatic impact on the deployability of various multi-AP aggregation schemes including: a) community-based sharing schemes (e.g. FON [3], Wi-Sh [4]), b) Telco-managed sharing schemes where residential WiFi gateways are shared across all users

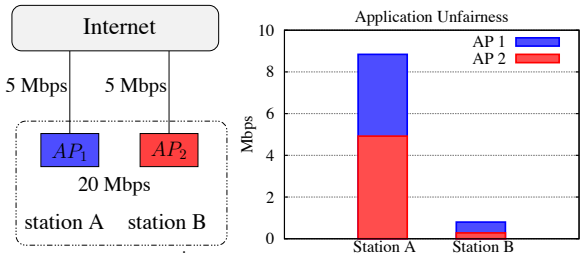
[¶]This work was carried out while the authors were at Telefonica Research Barcelona, Spain.
[*]Now at Disney Research: domenico@disneyresearch.com.
[‖]Supported by the Institució Catalana de Recerca i Estudis Avançats (ICREA).

MobiCom'10, September 20–24, 2010, Chicago, Illinois, USA.



(a) Deployment. (b) Unfair result.

Figure 1: Unfairness for users with different AP connectivity.



(a) Deployment. (b) Unfair result.

Figure 2: Unfairness for users with different number of flows.

that subscribe to the service, and c) commercial AP aggregation scenarios (e.g. airport hotspots). Moreover, existing aggregation schemes such as FatVAP [2] and VirtualWiFi [5] are not designed with fairness in mind, and hence cannot be directly applied to the above scenarios.

Motivated by these problems, we introduce THEMIS[1], a single-radio station that fairly aggregates the backhaul bandwidths of several APs. We extensively evaluate THEMIS in controlled scenarios, and show that it provides a fair distribution of the available backhaul bandwidth among users (Section 4). Finally, we validated THEMIS by emulating a typical urban neighborhood environment consisting of a setting of 10 commercial ADSLs with their correspondent 802.11g APs over 3 consecutive floors of a multistory building (Section 5).

2. FAIR WIRELESS BACKHAUL AGGREGATION

Let us consider the multi-AP backhaul aggregation system depicted in Fig. 3, where single-radio 802.11 stations simultaneously connect to one or more APs. In this scenario, the AP backhaul bandwidth of the APs is shared among the stations. Next we will give some illustrative examples which show the need for fairness and how greedy schemes, such as [2], fail.

Topology unfairness. Consider the experiment[2] depicted in Fig. 1(a), where stations A and B share 3 APs, each of them having a 5 Mbps backhaul. The wireless speed from each station to the three APs is 20 Mbps. However, because of its location, station B has only two APs in range, while station A can reliably connect to all the APs. A fair distribution of the aggregated AP backhaul would assign half of the backhaul capacity — 7.5 Mbps — to each station. Using a throughput maximization scheme as in [2], station B obtains 5 Mbps, almost half of the throughput of station A, which obtains more than 9 Mbps due to its better location (Fig. 1(b)).

Flow distribution unfairness. Consider now the experiment in Fig. 2(a), where stations A and B connect to two APs with 5 Mbps backhauls. The wireless speed between the stations and the APs is 20 Mbps. Station B starts one download per AP, each using a single TCP flow. Station A, on the other hand, starts one download per AP, but each using 10 TCP flows. The experiment is set up to guarantee that the flows are not limited by the end-to-end connection, i.e. the bottleneck is in the AP backhaul. In this scenario, a fair distribution of the AP backhaul would result in each station receiving 5 Mbps. However, if the stations aim to maximize their individual aggregate throughput without taking fairness into account as in [2], the result is a gross unfair distribution of the bandwidth, with station A receiving almost 9 Mbps, most of the aggregated bandwidth, while station B receives less than 1 Mbps (Fig. 2(b)). A similar scenario could be shown for the case of billing unfairness.

The above examples clearly illustrate the need to provide a fairness mechanism for the multi-AP backhaul aggregation scheme. However, it is important to agree on some notion of fairness, since each one could have different design implications and trade-offs. We discuss this in detail next.

2.1 What Kind of Fairness?

In order to address the unfairness situations described above, we start by describing our fairness requirements. First we would like to ensure that fairness is achieved at the level of the station's total received throughput, as opposed to individual flows or packet level fairness (**per-station fairness**). Second, we would like to ensure that users with better subscription plans (e.g. faster broadband links) obtain greater share of the aggregated AP backhaul bandwidth than users with cheaper subscription plans. Thus, in the examples above the throughput should be obtained proportionally to their priority (**weighted fairness**). Third, fairness should be enforced across all shared APs, and not just at the single AP level to ensure a fair global throughput allocation (**across-AP fairness**). Fourth, we want to provide a fairness scheme that is efficient in terms of network utilization and strikes a good balance between fairness and throughput (**efficient fairness**). And finally, we would like to provide a fairness scheme that is stable and has good convergence properties (**stable fairness**). Furthermore, in order to facilitate a wide adoption, we want to minimize the impact on the existing network infrastructure.

There are different reasons why the above requirements cannot be achieved using existing network technologies. For instance, in infrastructure mode, 802.11 does not provide per-station fairness because its downlink behavior is largely dominated by its FIFO packet-level scheduler [6]. TCP, on the other hand, only provides per-flow fairness among competing downlink flows, which is in fact the cause of the flow

[1]THEMIS is the Greek goddess of Justice, usually portrayed as an impassive blindfolded woman, holding scales outside a courthouse.

[2]All the tests and validations in this paper are performed experimentally on realistic scenarios. See Section 4 for details about the experimental setup.

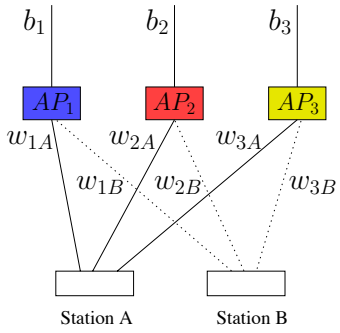


Figure 3: Multi-AP aggregation scenario.

distribution unfairness [7]. Even if one would manage to implement some fairness mechanism at the individual AP level (for example changing the FIFO behavior or introducing some clever time-based scheduler [8]), this would not result in across-AP fairness without the use of explicit signaling among the APs.

Our Choice of Fairness

In wireless systems, it is well known that fairness and throughput are often at odds [9]. For instance, imagine a scenario where two stations are sharing a wireless medium, and their wireless speeds are at a ratio of 10:1. A throughput optimal allocation would only allow the fast station to transmit, because every time slot devoted to the slow station would be wasted in low speed, losing the chance of a fast transmission. At the other extreme, a max-min fair allocation (e.g. one that maximizes the minimum of all station throughputs) would equalize the throughput transmitted by both stations. This allows the slow station to transmit most of the time, causing *performance anomaly* [9], that severely reduces the overall throughput.

Proportional fairness lies in the middle of the two extremes, providing a good compromise between fairness and efficiency (e.g. in [10]). It also achieves a good trade-off in terms of convergence and stability as shown in [11]. Finally, it allows for weighted fairness formulation. *Weighted proportional fairness* meets our efficient, stable and weighted requirements.

To comply with the other two requirements (per station and across-AP fairness), we cannot rely on existing formulations such as those in [2]. In fact [2] uses a knapsack scheduler that maximizes the individual station's throughput, and does not consider how the aggregate throughput is partitioned across stations. As a result, we need a new formulation that takes this problem into consideration. We describe it next.

2.2 Fairness Formulation

Recall the scenario depicted in Fig. 3. Let $\mathcal{S}$ be the set of stations and $\mathcal{A}$ the set of APs. Denote T_{ik} as the throughput sent from AP_i to station k. And let $y_k = \sum_{i\in\mathcal{A}} T_{ik}$ denote the total throughput received by station k. Let $U(\cdot)$ be a differentiable, strictly concave, increasing function which represents the utility at every station as a function of the received throughput. We model the fairness problem as[3]

$$\max \sum_{k\in\mathcal{S}} U(y_k) \tag{1}$$

$$\text{s. t.} \quad \sum_{k\in\mathcal{S}} T_{ik} \le b_i, \quad \forall i \in \mathcal{A}, \tag{2}$$

$$\sum_{i\in\mathcal{A}, w_{ik}>0} \frac{T_{ik}}{w_{ik}} \le 1, \forall k \in \mathcal{S}, \tag{3}$$

$$T_{ik} \ge 0, \quad \forall i \in \mathcal{A}, \forall k \in \mathcal{S}, \tag{4}$$

where w_{ik} is the wireless capacity[4] at which station k can receive from AP_i, that takes into account the interference from other clients connected to that AP, and b_i is the backhaul capacity of AP_i.

Eq. (2) is the *AP backhaul capacity constraint*, and ensures that the total traffic traversing the AP_i backhaul does not exceed the backhaul capacity b_i. Eq. (3) corresponds to the *station k wireless capacity constraint*, and guarantees that the total traffic received by station k does not exceed the total capacity of its wireless interface. Finally (4) forces the values T_{ik} to be positive.

Note that there exists an additional constraint not included in the formulation, corresponding to the *AP wireless capacity constraint*, namely $\sum_{k\in\mathcal{S}} \frac{T_{ik}}{w_{ik}} \le 1, \forall i \in \mathcal{A}$. This constraint ensures that the maximum capacity of the wireless interface at AP_i is not exceeded. However, we verified analytically that this constraint may be violated only in the extreme cases of clients severely limited by the wireless. We avoid this situation by preventing stations from connecting to *APs* if their signal-to-noise ratio (SNR) is very low. This makes sense, as a multi-AP aggregation scheme is only useful if the speed of WLAN is greater than the speed of the AP backhaul.

Finally, as described in Section 2.1, we choose a *weighted proportionally fair* utility function $U(y_k) = K_k \cdot \log y_k$, where K_k represents the relative priority of user k (for example, a value linearly dependent to the AP backhaul bandwidth owned by user k). If all the users have the same priority we use $K_k = 1$.

Decomposition and Interpretation

As described in [11], the solution of the above optimization problem can be obtained via the primal-dual formulation using a gradient descent algorithm. From there we derive the following optimal rate update rule

$$T_{ik} = \hat{T_{ik}} + \alpha \left(U'(y_k) - p_i - q_{ik} \right), \tag{5}$$

where $\hat{T_{ik}}$ is the bandwidth request in the previous step of the algorithm, $U'(y_k)$ is the derivative of the utility function evaluated at the current throughput received by the station y_k, and α is the step size of the rate update algorithm[5]. The quantities p_i and q_{ik} are the prices corresponding to

[3]For simplicity, and given that current residential traffic is heavily biased towards downloads, our formulation only considers downlink traffic. However an equivalent formulation can be designed for uplink traffic.

[4]Note that $w_{ik} = 0$ if station k does not connect to AP_i.

[5]When using proportional fairness, and in order to reduce oscillations as suggested by [12], we use $\alpha = \alpha' y_k$, with α' the new step size.

constraints (2) and (3) respectively, calculated as follows

$$p_i = \left[\hat{p_i} - \frac{\delta}{b_i}\left(\lambda b_i - \sum_{k\in\mathcal{S}} T_{ik}\right)\right]^+, \quad (6)$$

$$q_{ik} = \left[\hat{q_{ik}} - \frac{\gamma}{w_{ik}}\left(\mu - \sum_{i\in\mathcal{A}} \frac{T_{ik}}{w_{ik}}\right)\right]^+, \quad (7)$$

where $\hat{p_i}$, $\hat{q_{ik}}$ are the prices obtained in the previous step of the algorithm, and δ and γ are the step sizes of the price update algorithm. In order to improve the network utilization, and as suggested in [12], we normalize the price step size by the link capacities to favor good links. Finally $\lambda, \mu \leq 1$ are the congestion thresholds and $(x)^+ = \max(x, 0)$.

The price p_i in (6) represents the level of congestion on the backhaul of AP_i, and it is a linear function of its available bandwidth. Similarly, q_{ik} in (7) represents the level of congestion on the wireless link from station k to AP_i, and it is a function of the available card time at the station[6]. As congestion increases, the respective prices will increase and the throughput demand T_{ik} of station k through AP_i will decrease according to (5).

The values λ and μ are the *congestion thresholds*, i.e. respectively the level of utilization of the AP_i backhaul and the wireless radio-interface of station k that will trigger the algorithm congestion control. When that happens, the prices p_i and q_{ik} increase, prompting the throughput requests for their respective paths to decrease[7].

In order to distributedly solve the optimization problem in (1), each station has to periodically obtain the prices (6) and (7) for its links, and then update its rates following (5). However, implementing this algorithm locally at each station without sharing information with the APs and/or other stations has the following challenges:

- once the values T_{ik} in (5) are obtained at station k, those rates need to be enforced at AP_i (Section 3.1).
- in order to calculate the prices p_i in (6) and q_{ik} in (7), each station k needs to obtain the values of b_i and T_{ij} $j \neq k$, which are not directly available *at the station*. Moreover each station k needs to accurately know the wireless capacity w_{ik} of each AP_i (Section 3.2).
- a single-radio station has to manage the communication with multiple APs on independent radio frequencies. And it has to do it efficiently and using standard-compliant 802.11 (Section 3.3).

Addressing the above challenges in a real system requires careful design and implementation, which we describe next.

3. THEMIS

THEMIS is a single-radio wireless station based on the MadWiFi 0.9.4 driver [13] and the Click modular router 1.6.0 [14], that connects to multiple APs and aggregates their backhaul bandwidth. As shown in Fig. 4, THEMIS communicates separately to APs at different radio-frequencies using Time-Division Multiple Access (TDMA). Once connected to one AP, THEMIS transmits and receives traffic according to the 802.11 DCF protocol. The amount of time spent on AP_i is denoted *duty cycle* f_i. The constant time T that THEMIS takes to perform a standard TDMA cycle is called *wireless period*. THEMIS will use any spare duty cycle to do other operations such as AP scanning or saving energy.

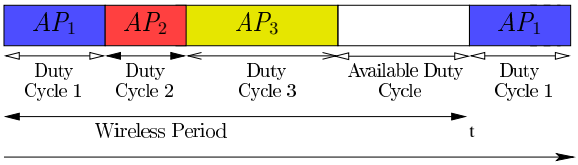


Figure 4: Time-division access to multiple APs.

3.1 Scheduler

Let us consider a THEMIS station running the optimization algorithm in (1), and calculating the request rate to AP_i to be T_{ik} in (5). In principle, in order to collect the bandwidth T_{ik} from AP_i, station k needs to connect to AP_i during a duty cycle $f_{ik} = T_{ik}/w_{ik}$, where w_{ik} is the wireless capacity from AP_i to station k. By reducing the time spent on AP_i, the duty cycle f_{ik} effectively acts as a gauge that limits the amount of bandwidth that can be received from the AP. As a consequence, TCP flows adjust their transmission rate to meet the request T_{ik}.

There are cases where station k does not receive the expected traffic T_{ik} during the duty cycle f_{ik}. There are various reasons for this discrepancy: wireless losses, congestion in the AP queue, CSMA contention delays in the wireless links, etc. We introduce a correction factor $\sigma_{ik} = T_{ik}/x_{ik}$ to account for the deviation between the expected received traffic T_{ik} and the *actual* traffic x_{ik} received by station k from AP_i during the selected duty cycle f_{ik}. As a result, THEMIS connects to AP_i for

$$f_{ik} = \sigma_{ik}\frac{T_{ik}}{w_{ik}} + c_i, \quad (8)$$

where σ_{ik} is the correction factor, and c_i is the overhead of switching from one AP to the next (see Section 3.3). Note that after applying the correction factor it may happen that the corrected duty cycles exceed the allowed time, violating the station k wireless capacity constraint, i.e., $\sum_{i\in\mathcal{A}} f_{ik} > 1$. In that case we distribute the *wireless period* proportionally among the links as described in the Appendix.

3.2 Estimators

The calculation of the duty cycle f_{ik} in (8) at station k for a given AP_i and the update of the prices p_i and q_{ik} in (6) and (7) require the following information:

- the utilization rate $\beta_i = \sum_{k\in\mathcal{S}} T_{ik}$ of the AP_i backhaul;
- the wireless capacity w_{ik}, that determines the maximum transmission rate of the wireless link; and
- the AP backhaul capacity b_i, that measures the maximum speed at which the AP_i backhaul can send traffic.

[6]The time that the card is not being used for transmitting or receiving.

[7]The values of the congestion thresholds represent a performance threshold: the closer to 1 the better if for the network utilization, but the worse is for the short-term fairness of the algorithm.

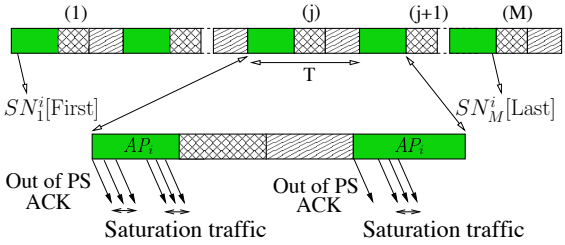


Figure 5: THEMIS estimators are based on local processing at MAC and PHY layer.

A straightforward way to obtain those values would be to introduce new signaling to exchange this information between the APs and the stations. However, that would introduce extra overhead, and would also require modifying (or replacing) the existing AP installed base. To avoid it, THEMIS estimates these values locally. Note that it is important to achieve high accuracy on the estimations, because wrong estimations would affect the performance of the scheduler. This is a hard problem because:

- *AP backhaul*: the AP backhaul is shared with other stations and any measure of the AP_i utilization rate β_i and capacity b_i must be done in the limited slice of time $f_{ik} \cdot T$ that station k dedicates to AP_i.

- *wireless link*: the wireless capacity of one AP has to be measured while the AP transmits in saturation. This is not guaranteed because the wireless link is usually not the bottleneck of the end-to-end communication.

We next describe how THEMIS estimates these values.

Utilization Rate of the AP Backhaul

The estimation of the utilization rate β_i of the AP backhaul relies on the fact that every frame sent by an 802.11 AP carries a MAC Sequence Number (SN) in the header. The SN is a module 4095 integer incremented by the AP each time a new frame is sent, and it is independent of the destination. THEMIS stations listen to the traffic sent by AP_i, and store its SNs. By counting the SNs, the THEMIS station knows the amount of packets traversing the AP_i backhaul[8]. Note that this way of counting is robust to packet loss and disconnection periods, as long as the stations do not miss more than 4095 successful frames (retransmitted frames do not increase the SN), which for an average 802.11 frame size would correspond to seconds, an order of magnitude larger than the THEMIS' TDMA period[9].

Formally, let us refer to Fig. 5. We denote SN_1^i[First] and SN_M^i[Last] the MAC sequence number of the first and last packet, respectively, sent by AP_i to any station, during a window of time $M \cdot T$, where M is an integer equal or greater than 1. Then, THEMIS derives the number of packets sent from AP_i in the time $M \cdot T$ as:

$$N^i = \big(SN_M^i[\text{Last}] - SN_1^i[\text{First}]\big) \bmod 4095.$$

Let us also denote $E[L_i]$ the average bit length per packet at IP layer over *all* the packets received by station k when it is connected to AP_i. We make the reasonable hypothesis that $E[L_i]$ does not change between the connection and disconnection time from AP_i. Finally, we calculate the AP_i backhaul utilization rate as

$$\beta_i = \frac{E[L_i] \cdot N^i}{M \cdot T}. \tag{9}$$

Wireless Capacity

THEMIS measures the wireless capacity by calculating the packet dispersion of frames directed to it when the AP is transmitting in saturation. In order to detect saturation periods, station k run-time senses the *wireless channel occupancy*, that is, the percentage of time that the channel is bus, between two consecutive received packets. These statistics are collected from specific 802.11 baseband registers, exposed by the NIC card. If the occupancy is above a certain threshold (80% in our implementation), we define the AP in saturation for that pair and store the packet length of the second packet and the dispersion between the packets. Then, referring to Fig. 5, w_{ik} is derived averaging over the window of measure $M \cdot T$ as

$$w_{ik} = \frac{\sum_{j=1}^{M} B_j}{\sum_{j=1}^{M} T_{j,SAT}^i}, \tag{10}$$

where B_j is the sum of the packet length in saturation sent from AP_i to station k and $T_{j,SAT}^i$ is the sum of the dispersions when station k receives in saturation mode during the j-th connection to AP_i. Note that w_{ik} takes into account the existing interference, and depends on the current PHY rate of APs and stations, the signal quality, and the performance anomaly [9] during the measurement period.

AP Backhaul Capacity

Several Internet services can be used to estimate the AP backhaul capacity b_i[10], some of them also provided by ISPs to their clients. Usually, a file coupled to a script is downloaded from a server. The script detects when the client has completed the download and determines b_i.

The server report may be hindered by the cross-traffic rate of the packets (eventually) being sent through the same AP_i backhaul to the other stations. THEMIS connects to a capacity server, but instead of relying on the server report, it calculates the peak reached by the utilization rate β_i during the connection time to the capacity server as

$$b_i = \max_{l=1,2,\ldots L} \overline{\beta_i[l]},$$

where L represents the number of measures during the test at the $1/(M \cdot T)$ rate, and $\overline{\beta_i[l]}$ denotes the smoothed average of $\beta_i[l]$ after the l-th calculation.

3.3 Multiple APs manager

In order to provide an *efficient* TDMA implementation in THEMIS, the wireless driver on top of the single radio interface is *virtualized*, i.e., it appears as independent Virtual

[8]Here we assume that most of the 802.11 data traffic traversed the AP backhaul, as it is often the case when using 802.11 in infrastructure mode.

[9]To increase the accuracy of the estimation, THEMIS operates in promiscuous mode, thus accounting for the information of the packets sent to other THEMIS stations. This information is never encrypted and can always be retrieved, even when the payload is encrypted.

[10]See for example `http://www.bandwidthplace.com` or `http://www.speedtest.net`.

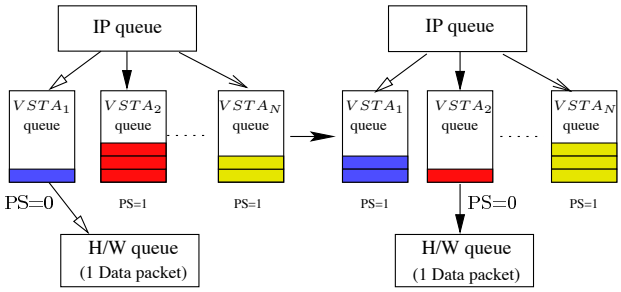


Figure 6: Queue Management.

STAtions ($VSTA_i$) associated to their respective APs. Each $VSTA_i$ is responsible for managing the data communication with AP_i and the related procedures such as association, authentication and scanning. To prevent losses during the TDMA operation, each THEMIS station k uses the 802.11 Power Save (PS) feature as follows (Fig. 6):

- During the active *duty cycle*, $VSTA_1$ exchanges traffic according to the 802.11 protocol, while the other *VSTAs* are dormant in PS mode. During the PS time, both the AP_1 and the station can only buffer packets [2, 5, 15].
- When the *duty cycle* expires, $VSTA_1$ sends a frame to inform AP_1 that it is going into PS mode. Once received the MAC ACK, $VSTA_1$ and AP_1 start to buffer the packets destined to each other.
- THEMIS assigns the control of the card to $VSTA_2$ and switches to the AP_2 radio-frequency.
- $VSTA_2$ sends a frame to AP_2 to indicate that it is ready to send/receive traffic, and awaits for the MAC ACK.
- The process continues until the station has cycled through all the *VSTAs*. The spare duty cycle can be used for other operations such as scanning or sleeping see Fig. 4). The station then restarts the TDMA cycle.

In order to minimize the switching cost c_i in (8), THEMIS achieves a fine-grained timing at MAC/PHY level, using the following techniques:

- THEMIS introduces a *MAC virtual queue* per AP. This allows to buffer packets in the MAC virtual queue, when THEMIS is selecting some other AP.
- THEMIS efficiently manages a hardware buffer (common to *all* the VSTAs) of one (1) data packet to quickly switch among MAC virtual queues. This is a challenging task, because short H/W queues cause inefficiencies that negatively affect throughput (as a comparison, the original MadWiFi driver sets the H/W queue size to 200)[11].
- In order to switch the PS state, THEMIS piggybacks the MAC PS bit on the header of the pending data on top of the MAC virtual queue. THEMIS reverts to the classical use of probes (as done in [2, 5]) in the rare event of not having data packets ready for transmission.

[11]Packets in the hardware queue must be sent before the end of the *duty cycle* assigned to the *VSTA*. This causes a delay respect to the expected end of the duty cycle imposed by the THEMIS scheduler. The efficient management of a hardware buffer of size one minimizes any extra-delay.

With the techniques described above, THEMIS incurs in a switching cost c_i of about 1.2-1.5 ms, most of which (around 800 μsec) is spent in hardware radio-channel commutation. This limited overhead, significantly less than [2,5], increases the stability of the system by reducing the jitter in the switching procedure. This enables a fine-grained selection of duty cycles assigned by the scheduler even if the station transmits in saturation mode, which is of particular importance for TCP traffic.

On top of the MAC implementation, THEMIS uses a *flow mapper* to assign new TCP flows from the upper layers to a specific *VSTA*. While we could use a more sophisticated flow mapper, we employed a proportional based mapper as in [2]: the amount of traffic r_{ik} assigned to AP_i maintains the proportions of the bandwidth obtainable from each AP and equal to $r_{ik}=\frac{f_{ik}w_{ik}}{\sum_j f_{jk}w_{jk}}$.

Finally, THEMIS implements a Reverse-NAT module that i) makes sure that the packets leave the station with the correct source IP address (i.e. the one corresponding to the outgoing *VSTA*, as assigned by the AP); and ii) presents a consistent (dummy) IP to the applications, providing IP transparency to higher layers.

4. VALIDATION

We evaluate THEMIS in an extensive set of tests. Our findings show that

- the estimators described in Section 3.2 are accurate, and stations do not need to request information from the network.
- THEMIS achieves a fair sharing of the aggregate network capacity among stations, while efficiently using the aggregated network capacity.

In our experiments, the APs are off-the-shelf Linksys, running Linux DD-WRTv24 firmware. The stations are Linux laptops, equipped with a single-radio Atheros-based wireless NIC. For every AP and station in the network, the wireless multimedia extensions (WME) and the RTS/CTS handshake are disabled. Any non-standard compliant 802.11 feature is also disabled, and H/W queues are set up with 802.11 best effort parameters.

4.1 Evaluation of THEMIS Estimators

We first verify the accuracy of the estimators used by THEMIS. We start studying the estimation of the backhaul utilization rate β_i in a test where 3 THEMIS stations download HTTP files using 3 Mbps lines. Stations are connected to the AP using a fixed connection time of 25 ms over a period of 100 ms. Stations are not synchronized, and they connect to the corresponding APs at independent times. Consequently, stations can only observe a fraction of the traffic load sent to other stations. Moreover, because of the wireless nature, they may not receive some packets sent to other stations, missing information such as the sequence number SN and packet length L_i needed by the estimator in (9).

In this configuration, we compare the estimations of the backhaul utilization rate over the time at each THEMIS station with the actual rate measured at the AP. The results in Fig. 7 show that all stations obtain a very accurate estimation.

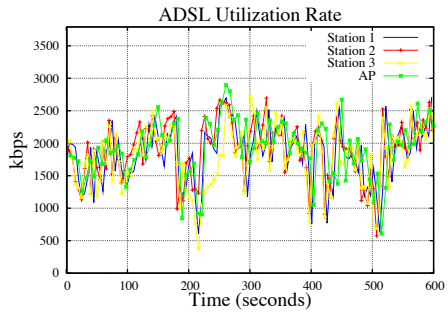


Figure 7: AP backhaul utilization estimation (β_i).

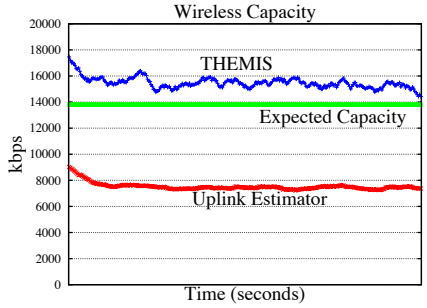


Figure 8: Wireless capacity estimation (w_{ik}).

We next evaluate the THEMIS' wireless capacity estimator described in Section 3.2. In the test, the THEMIS station connects to an AP with a duty cycle of 25 ms over a period of 100 ms, and performs several HTTP downloads from different Internet servers. Fig. 8 shows the estimation of w_{ik} in a period of 4 minutes. THEMIS estimator gives a good approximation (around 13.7 Mbps) of the speed reported with a downlink Iperf test from a server located in the same LAN of the AP.

Estimators of w_{ik} are also proposed in [2, 16]. However, these estimators are based on the time needed to transmit a packet from the 802.11 station, and so they better represent uplink speeds rather than downlink. This can result in severe errors in the estimation of the downlink wireless capacity. As an example, Fig. 8 shows the performance of the estimator in [2] for the same scenario, and we observe that it under-estimates the wireless capacity. In fact, a high downlink speed will cause a long air-time before transmitting a packet in uplink, that translates in a low (and erroneous) downlink wireless capacity estimation.

4.2 System Evaluation

We now evaluate the system implementation of THEMIS through different tests. For every scenario, we run five tests of 1800 secs and plot the average results obtained. We choose such a configuration to verify that results are stable in time and across different tests. To achieve independent tests, stations are configured so that the THEMIS estimators are reset after each test. For the transport layer, we use Linux standard TCP Reno with SACK and delayed ACK option enabled and we emulate the AP backhaul capacities using the *tc* Linux traffic shaper. Unless otherwise stated: i) we open a TCP flow per AP using *iperf*, ii) the AP backhaul capacity is known at each station k, while the ADSLs utilization rates $\{\beta_{ik}\}$, and the wireless capacities $\{w_{ik}\}$ are estimated at THEMIS station k as described in Section 3.2.

	station A	**station B**
802.11 Legacy	0.45 Mbps	6.24 Mbps
THEMIS	3.15 Mbps (f=0.19)	3.40 Mbps (f=0.15)

Table 1: Two stations connected to one AP.

THEMIS parameters

Selecting the appropriate wireless period represents a complex trade-off. On one side, switching among APs introduces overhead, so selecting long *wireless periods* reduces the overhead. However, long periods affect TCP performance because they artificially increase the end-to-end delay. On the other hand, short periods are desirable, as they reduce the disconnection time from the APs in PS mode, and prevent TCP from timing-out. In order to meet all the previous requirements, we select a *wireless period* T of 100 ms. The scheduler and estimators are updated every $20 \cdot T = 2$ seconds. We also impose that the time of connection to each AP is at least equal to the switching cost plus 2 ms (that gives a minimum *duty cycle* $f_i \geq 0.03$).

The values of α (5), δ (6) and γ (7) have been selected based on extensive simulations, with values that provide a good trade-off between convergence and stability. Similarly, we choose the congestion thresholds for the AP backhaul and the wireless capacity to be λ=0.95 and μ=0.95 respectively. A more detailed sensitivity analysis of the parameters falls outside the scope of this paper.

Two stations connected to one AP

We first consider the configuration where two stations are connected to the same AP (802.11 legacy operation). In the test, we consider that both stations receive traffic from the AP at a downlink wireless rate of about w_1=20-22 Mbps and are connected to an AP backhaul of b_1=7 Mbps[12]. We also consider that station A opens one TCP flow per AP while station B opens 10 TCP flows per AP.

The results are summarized in Table 1. With legacy 802.11, station B uses most of the backhaul capacity with an average received throughput of 6.24 Mbps while station A starves at 0.45 Mbps, at a throughput more than 13 times smaller than station A. On the other hand, each THEMIS station connects for a limited percentage of card time on each AP to collect the requested bandwidth T_{1k}. The result is that station B — that opens more flows — connects less time than A, i.e. 14% versus 19% of their time, and then for just a few ms of the entire wireless period. Indeed station B needs in average less time to achieve the bandwidth from the AP, because it is less affected by the TCP's sawtooth behavior of each flow. As a result, station A and B obtain similar throughput (3.15 Mbps vs 3.40 Mbps), with a network utilization of 6.55 Mbps instead of 6.69 Mbps, a consequence of the THEMIS congestion control.

[12]This is the AP backhaul capacity, and hence the actual speed available for TCP traffic may be lower. In fact, because of TCP's sawtooth behaviour, not all the *available bandwidth* at the bottleneck may be used at any time. The bandwidth utilization per path can increase establishing more than one TCP connection over each AP.

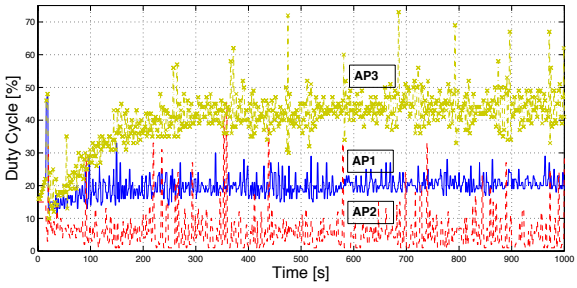
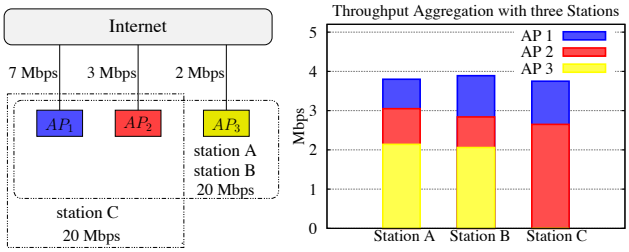


Figure 9: Duty Cycles evolution with one station.



(a) Deployment. (b) Results.

Figure 10: 3 THEMIS stations sharing 3 APs.

	AP_1	AP_2	
Capacity	b_1=5 Mbps	b_2=5 Mbps	b_2=2.73 Mbps
Duty Cycle	0.25	0.71	0.67
Throughput	4.74 Mbps	2.43 Mbps	2.21 Mbps

Table 2: Connection to two APs, one wireless bottleneck.

One station connected to Multiple APs

In these tests we evaluate the efficiency in terms of network utilization with one THEMIS station connected to two APs. Let us first consider the case where the throughput is not limited by the wireless card speed on any of the connections, i.e. the expected result is to completely utilize the available backhaul capacity of the three APs. Here, station A is associated to 3 APs, at a downlink wireless rate of about w_1=w_2=w_3=20 Mbps and is connected to AP backhauls of b_1=5 Mbps, b_2=1 Mbps and b_3=10 Mbps respectively, with a total bandwidth of 16 Mbps.

As we can see in Fig. 9, the duty cycles converge to stable range of values. THEMIS spends most of the time on the best network path (via AP_3) and less time on the worst network path (via AP_2). This results in a total aggregated throughput of 15.05 Mbps, that is with an average utilization of 94% of the network aggregated capacity, as we expect from the setting of λ=0.95.

We then consider a scenario where a THEMIS station connects to two APs, and is limited by the wireless speed on one link. In the test, a THEMIS station measures a downlink wireless capacity of w_1=20.74 Mbps on AP_1 and w_2=2.73 Mbps on AP_2 and is connected to AP backhauls of 5 Mbps each, bottlenecked by the wireless on path 2.

Results are summarized in Table 2. We consider two settings: first, the ideal case where the AP backhaul capacities are correctly estimated at 5 Mbps, and second, the most realistic scenario where the estimation of the AP backhaul capacity of the path limited by wireless (path 2) is bottlenecked by the wireless capacity b_2=w_2=2.73 Mbps.

In the first case, THEMIS spends f_1=0.25 on the path with higher wireless speed, obtaining a throughput of 4.74 Mbps. The rest of the time it is spent in the path limited by the wireless link (f_2=0.71), where it achieves a throughput of 2.43 Mbps (for an aggregated 7.17 Mbps). Note that a small time (f=1-0.25-0.71=0.04) is used by THEMIS to detect card time congestions as shown in (7).

In the second case, the throughput achieved on path 2 slightly reduces to 2.21 Mbps, with a sub-utilization of the path of 2.43-2.21=0.21 Mbps. In fact, a smaller (and wrong) AP backhaul capacity estimation causes a higher AP backhaul price p_2 on the link, that in turn causes the station to request less throughput on this connection according to (5). This translates in a smaller duty cycle f_2=0.67 rather than 0.74, that in turns reduces the bandwidth received on this path.

In both tests, THEMIS makes an efficient usage of the network: the overall throughput is higher than the one obtained being connected 100% of the time to AP_1 (at most 5 Mbps) or to AP_2 (at most w_2=2.73 Mbps).

Multiple Stations connected to Multiple APs

We evaluate the fairness and the network utilization efficiency, when different stations are connected to multiple APs. First, we analyze the case of 3 THEMIS stations, in the scenario in Fig. 10(a), with 3 APs with backhaul speeds of b_1=7 Mbps, b_2=3 Mbps and b_3=2 Mbps respectively, resulting in a total aggregated capacity of 12 Mbps. Given that none of the stations is limited by the wireless links, each station is expected to get an average aggregated speed close to 12/3=4 Mbps, even if the stations share a different number of APs.

This is demonstrated in Fig. 10(b): the 3 stations obtain a fair share of the aggregate AP backhaul speed, averaging 3.80 Mbps, 3.89 Mbps and 3.75 Mbps on station A, B and C, respectively, for a total aggregate throughput of 11.44 Mbps, again around the 95% of the overall available capacity.

Then we consider the scenario in Fig. 11(a), where station B shares two AP backhauls with station A at wired speeds of 5 and 1 Mbps, respectively. Station A can also connect to a third AP (AP_3) with a backhaul speed of 10 Mbps. Then, station B can obtain *at most* 6 Mbps and can never reach the 10 Mbps speed of AP_3 backhaul.

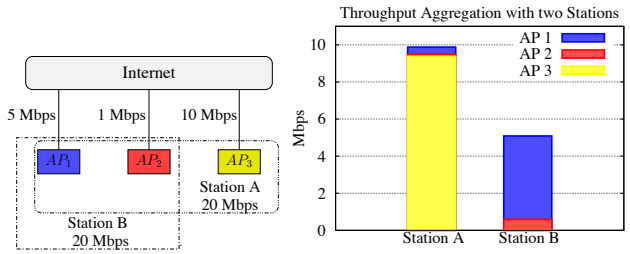
The results in Fig. 11(b) show a total aggregate TCP throughput of 9.88 Mbps on station A (with f_1=0.08, f_2=0.05 and f_3=0.47), and 5.09 Mbps on station B (f_1=0.28, f_2=0.09). Station A makes the fair decision, reducing the amount of time connected to the shared APs as much as possible.

Stations with an uneven number of TCP flows

Let us recall the flow distribution unfairness example shown in Fig. 1(a) (Section 2) where two stations are sharing two 5 Mbps backhaul APs and use an uneven number of TCP flows. Fig. 12 shows that THEMIS is able to guarantee a fair share of the aggregated backhaul capacity to each station.

Stations with different priorities

Consider the same scenario as before, where now Station A and Station B happen to be roaming and sharing two 5



(a) Deployment. (b) Results.

Figure 11: Two stations sharing partially overlapping sets of APs where station B cannot obtain the throughput obtained by station A.

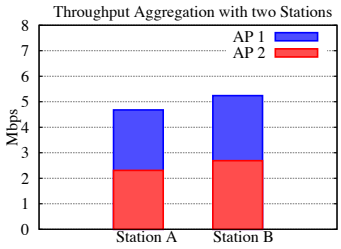


Figure 12: Fair share of the backhaul bandwidth using THEMIS.

Mbps AP backhauls. Let us consider that Station A belongs to a user that has higher priority than the user of Station B. For sake of illustration, we suppose that THEMIS applies weighted proportional fairness using $K_A = 4$ and $K_B = 1$. Therefore, it is expected that Station A obtains $K_A/(K_A + K_B) = 0.8$ of the total bandwidth while Station B obtains the remaining $K_B/(K_A+K_B) = 0.2$. The experiments show that THEMIS stations obtain a throughput of 7.64 Mbps for Station A and 2.0 Mbps for Station B.

5. THEMIS IN THE WILD

In order to test the scalability of THEMIS, we deploy a realistic testbed spanning three floors of a multistory building. The network consists of 10 commercial ADSLs with their corresponding WLAN APs and 10 THEMIS stations, i.e. the owners of each line. Nine of the ADSL lines have a nominal capacity of 3 Mbps and one has a nominal capacity of 1 Mbps. The APs are distributed every 80 square meters to emulate the average residential flat size (see Fig. 13) and are set to independent radio-frequencies in the 2.4 GHz ISM band[13].

In the bootstrap phase, the APs are selected based on a passive analysis of the SNRs of the 802.11 AP beacons with "THEMIS" essid. Stations scan for the APs in range and start authenticating and associating to the AP with highest SNR sequentially associating to APs with smaller SNR. THEMIS requires a minimum SNR of 10dB to guarantee a stable reception at 1 Mbps PHY basic rate. In each test, automatic rate selection is active in each THEMIS station, with independent instances of the Minstrel rate selection algorithm [17] over each wireless uplink.

[13]The channels optimization is out-of-the-scope of this paper.

Figure 13: Testbed deployment. APs and stations have been deployed over 3 floors, ground floor (on the left), mezzanine (in the middle), and first floor (on the right). Each circle represents an AP, while stations are placed nearby the APs, one station per AP. Only stations A, B and C, relevant for some experiments, are shown in the map. Obstacles, as walls and desks are presented between all the AP links.

5.1 Characterization

We measure the capacity of each link of the network (i.e. the ADSLs and the 10×10 wireless links). Our findings are that the 3 Mbps lines offer a constant maximum speed of 2.65 Mbps and the 1 Mbps line offers 0.89 Mbps. Regarding the wireless links, apart from the 10 "home" links where the station is located nearby the AP, the SNR measured per wireless link is consistently lower than 30 dB.

We then generate traffic from a server connected to the APs via an 802.3 LAN, activating one AP-station link at a time, with 5 minutes dedicated to each test. We calculate the average throughput and the standard deviation for each link. Then, we re-order the 10 links in descending order per-station, based on the average throughput.

Results are reported in Fig. 14. Each station can receive TCP traffic from at least 3 APs (and up to 5) at a speed higher than 10 Mbps. The results show the feasibility of aggregating the low-speed backhaul bandwidth of at least three APs.

5.2 The Effect of Location

To show the effect of location, we perform a test in which two stations (Station A and Station B as shown in Fig. 13) initially share the same set of APs and are located a few meters away from its "home" AP. For this test we use three APs connected to 3Mbps lines, hence, the total backhaul capacity that Station A and Station B share is $2.65 \times 3 = 7.95$ Mbps. As a result, a fair share of the total bandwidth would be $7.95/2 = 3.975$ Mbps per station. Both stations perform several HTTP downloads per AP during 2400 seconds. After 1200 seconds of test, Station B moves to a second location from which it can only be connected to two of the former APs. As we do not implement IP mobility in our testbed, all the connections of Station B are dropped and started again in the new location. As a consequence of the movement of Station B, the topology of the network changes and stations observe an uneven AP backhaul capacity.

We run the test using a throughput maximization algorithm as in FatVAP [2][14] (Fig. 15(a)), and using THEMIS

[14]We implemented the throughput maximization algorithm according to the description in [2]. To provide a fair compar-

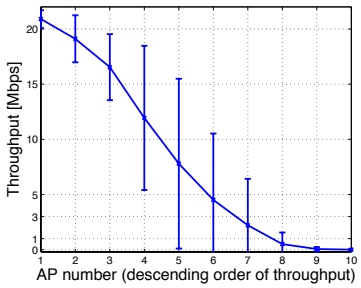


Figure 14: Wireless link quality assessment.

(Fig. 15(b)). Results show that, when the network topology is similar for both stations (they are both connected to 3 APs at similar speed), using throughput maximization results in a similar long-term performance for both stations, but with no guarantee of short-term fairness. Moreover, when the topology changes, Station B is clearly penalized by its new unlucky location obtaining 2.8 Mbps while Station A obtains 4.8 Mbps.

On the other hand, THEMIS guarantees a fair share of the backhaul capacity in both topologies, offering 3.5 Mbps to each station. Note that when Station B moves to the new position, the PHY rate is quickly reduced because of the lower signal strength, with THEMIS quickly converging to a fair assignment of the backhaul capacity. Also note that because the fairness mechanism relies on the congestion thresholds λ and μ (Section 2.2), the network utilization is slightly lower than the optimal.

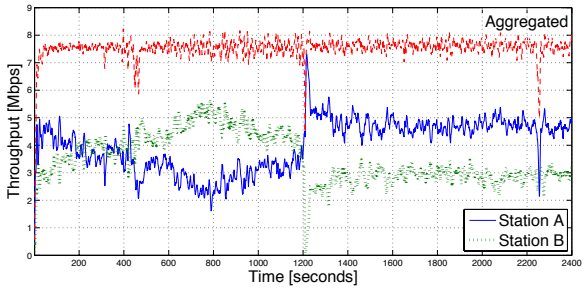
5.3 Integrated Operations

We have shown via different deployments that THEMIS is able to deal with the three types of unfairness that arise when aggregating AP backhaul bandwidth. However, in a real life scenario, these unfairness can take place at the same time. Thus, we perform a test that evaluates THEMIS in presence of a P2P station (Station A), an unluckily located station (Station B) and a low priority station (Station C). The location of the stations is shown in Fig. 13. For this test we use three APs, each with a 3 Mbps backhaul. The P2P and the low priority stations are connected to 3 APs while the unluckily located station is connected to 2 APs. Given that the low priority station owns a 1 Mbps ADSL while the others own a 3 Mbps ADSL line, the weights have been set to $K_A = K_B = 3$ and $K_C = 1$. In such experiment, a fair system should be able to allocate the bandwidth proportionally to the priority of the users.

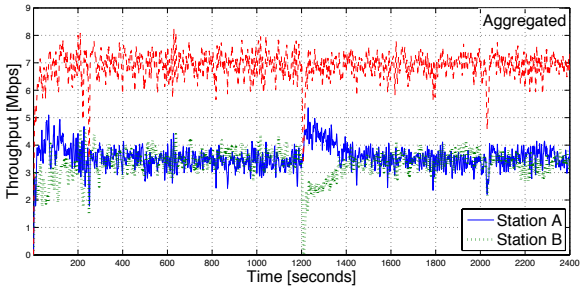
At the beginning of the test, Station A starts downloading P2P traffic from the three APs. After 1200 seconds, Station B starts several HTTP downloads from the two APs it is connected to. Finally after 1200 seconds more, Station C also starts HTTP traffic from the APs.

The result of using a throughput maximization algorithm is shown in Fig. 16(a). It is noticeable that Station A, due to the high number of TCP flows opened by P2P applications, obtains most of the backhaul capacity preventing Station B from obtaining its fair share of the bandwidth. Furthermore, when Station C starts its downloads, the absence of priority among users further reduces the throughput obtained by Station B, introducing billing unfairness. Finally, since Station B and C achieve a similar throughput despite that station B is unluckily located, the flow distribution unfairness dominates over the topology unfairness.

ison with THEMIS, we use the wireless capacity estimation of THEMIS and the APs manager described in Section 3.3.



(a) FatVAP



(b) THEMIS

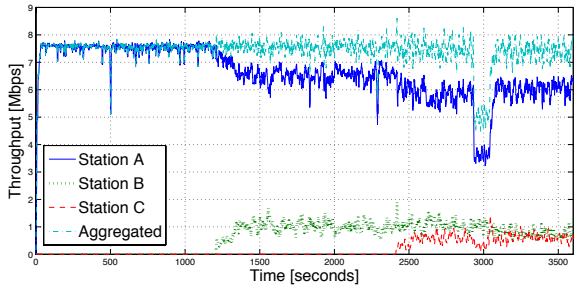
Figure 15: Assessment of the topology unfairness in the residential-like deployment.

The result of using THEMIS is shown in Fig. 16(b). When the unluckily located station B starts its downloads after 1200 seconds, the wireless capacity measured at station A over the shared APs is reduced because of the performance anomaly [9]. However, the system quickly adapts: the wireless links with lower wireless capacity receive a higher wireless price q_{ik} and hence smaller throughput demand T_{ik} and dedicated duty cycle f_{ik}. A smaller duty cycle for both station A and B means that the probability of being connected to the same AP at the same time, and consequently the occurrence of performance anomaly, is reduced. Concluding, THEMIS offers a fair share of the aggregated bandwidth to both stations, while providing a high usage of the backhaul bandwidth. Finally, when Station C starts its downloads, the priorities are preserved and Station A and B obtain a greater share of the backhaul capacity.

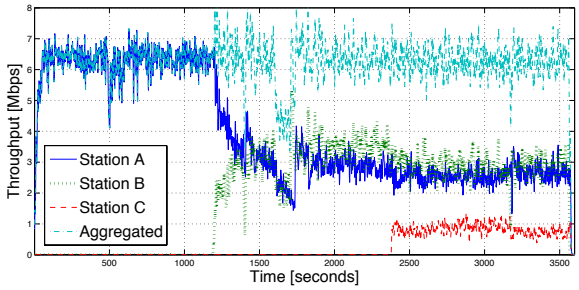
6. RELATED WORK

In recent years Wi-Fi communities have attracted the attention of both the research community and the wireless industry because of the uptake of WLAN in residential areas. In this direction [3, 4, 18] propose to allow members of the communities to share the backhaul bandwidth of their WLAN APs. Among those, Wi-Sh [4] discusses the fairness problems that can arise from sharing resources. However, it does not consider the use of multiple APs to aggregate their backhaul bandwidth.

Backhaul bandwidth aggregation has been explored in



(a) FatVAP



(b) THEMIS

Figure 16: Test with the effect of the three types of unfairness: Station A uses P2P traffic, Station B is unluckily located (starts after 1200s) and Station C is a low priority station (starts after 2400s).

[19, 20], where stations connect to their home APs via ethernet and to the remote APs using WLAN. However, they do not connect to multiple APs via WiFi, so the number of APs they can aggregate is limited by the number of physical interfaces (ethernet and WiFi) available at the stations.

The idea of connecting to multiple APs through a single radio was first shown in VirtualWiFi [5]. The authors rely on the WLAN standard power saving (PS) mode to switch among different Wi-Fi nodes in time division. Switching among Wi-Fi nodes is transparent to the applications, but at a high cost in time (30-600 ms). In fact, VirtualWiFi implements the code on top of the driver card with a MAC instance for connection.

Within the problem of single radio AP backhaul aggregation, the closest work is FatVAP [2]. The authors introduce a scheduler to select the percentage of time to spend on each AP to maximize the aggregate throughput at each station. However [2] has a limited focus because it does not resolve the unfairness across stations, and it only considers stations connected to (strictly) more than one AP. Furthermore, the local throughput maximization approach in [2] can not be extended in order to take into account priority-based per-station fairness. Compared to [2], THEMIS fairly aggregates the AP backhaul bandwidth among the different THEMIS stations, irrespectively of their location, link quality and number of APs they have in range. Moreover, THEMIS is able to adapt to different fairness objectives in order to accommodate the different scenarios discussed in Section 1, and it achieves this in a completely distributed manner. Finally, THEMIS implementation of the single-radio multi-AP TDMA access is improved compared to [2, 5], reducing the frequency switching overhead and increasing the accuracy when selecting the amount of time that the station connects to the different APs. This results in a more efficient operation and increased throughput.

Among other work, [21] introduces a support for a seamless hand-off between WLAN APs. In [15], standard solutions have been exploited to increase the aggregate throughput observed by a single station with respect to the design in [2, 5, 21]. However, these works do not consider the problem of fairness.

Link-alike [22] tackles the problem of minimizing the uplink total transfer time via multiple wireless links. However, the solution requires cooperation among the APs, with 802.11 APs transmitting and receiving at the same radio-frequency, and a custom TCP protocol over the wireless link.

Several tools have been designed to estimate the available bandwidth along a network path. However, these tools typically send active probes along a path and/or require a cooperative implementation at both the sender and receiver [23, 24].

There is little work on non-cooperative estimation of the ADSL available bandwidth. Most notably ABwProbe [25] and FAB-Probe [26] rely on the asymmetry of the ADSL downlink capacity to send TCP ACK packets of different sizes and receive small TCP RST packets from the TCP client. Since the TCP RST is at fixed length, they cannot estimate the available bandwidth from the client-side, as done by THEMIS.

The estimation of the wireless capacity has been studied with different levels of accuracy (see e.g. [2, 27]). A comparison with the implementation of THEMIS has been provided in Section 4.1. Our experimental evaluation has demonstrated that the robustness of THEMIS in realistic scenarios, under MAC contention, adaptive PHY rates and performance anomaly.

7. CONCLUSION AND FUTURE WORK

We have shown that fairness is a crucial factor for the success of multi-AP aggregation schemes. Without fairness, the perceived value of the system is severely reduced, eliminating the incentives of users to participate, and of providers to deploy it. This effectively renders such aggregation schemes unfeasible. In order to achieve fairness, existing multi-AP aggregation systems that maximize the throughput of single users cannot be extended. As a consequence a complete re-design of the system is required.

To address this problem we introduced THEMIS, a single-radio station implemented in commodity hardware that is fair in a multi-AP aggregation scenario. THEMIS operates locally at each station, using standard 802.11, without requiring any change in the network. This makes THEMIS ready to be deployed. In fact, THEMIS is being used by a major broadband provider in a commercial trial.

There are several interesting future lines for this work. On the technical side, we plan to extend THEMIS to include uplink traffic in the formulation, and investigate the impact and trade-offs that TDMA may have over the TCP performance. From an architectural point of view, we are currently exploring the use of THEMIS to design more power efficient access networks. Finally, it would be interesting to understand how THEMIS can be leveraged to perform efficient large-scale cellular data offloading, which appears to be a difficult challenge for the years to come.

8. REFERENCES

[1] D. Han, A. Agarwala, D. Andersen, M. Kaminsky, D. Papagiannaki, and S. Seshan, "Mark-and-Sweep: Getting the 'Inside' Scoop on Neighborhood Networks," in *Proc. of the ACM IMC Conf.*, (Vouliagmeni, Greece), pp. 99–104, October 2008.

[2] S. Kandula, K. Lin, T. Badirkhanli, and D. Katabi, "FatVAP: Aggregating AP Backhaul Capacity to Maximize Throughput," in *Proc. of the USENIX NSDI Conf.*, (San Francisco, CA), pp. 89–04, April 2008.

[3] "FON." `http://www.fon.com/`.

[4] X. Ai, V. Srinivasan, and C.-K. Tham, "Wi-Sh: A Simple, Robust Credit Based Wi-Fi Community Network," in *Proc. of the IEEE INFOCOM Conf.*, (Rio de Janeiro, Brazil), pp. 1638–1646, April 2009.

[5] R. Chandra and P. Bahl, "MultiNet: Connecting to Multiple IEEE 802.11 Networks Using a Single Wireless Card," in *Proc. of the IEEE INFOCOM Conf.*, (Hong Kong, China), pp. 882–893, March 2004.

[6] E. Park, D. Kim, H. Kim, and C. Choi, "A Cross-Layer Approach for Per-Station Fairness in TCP over WLANs," *IEEE Trans. Mobile Comput.*, vol. 7, no. 7, pp. 898–911, 2008.

[7] B. Briscoe, "Flow Rate Fairness: Dismantling a Religion," *ACM SIGCOMM Comp. Commun. Review*, vol. 37, no. 2, pp. 63–74, 2007.

[8] G. Tan and J. Guttag, "Time-based Fairness Improves Performance in Multi-rate WLANs," in *Proc. of the USENIX Annual Tech. Conf.*, (Boston, MA), pp. 23–24, June 2004.

[9] F. R. G. B.-S. M. Heusse and A. Duda, "Performance Anomaly of 802.11b," in *Proc. of the IEEE INFOCOM Conf.*, vol. 2, (San Francisco, CA), pp. 836–843, April 2003.

[10] H. J. Kushner and P. A. Whiting, "Convergence of Proportional-Fair Sharing Algorithms under General Conditions," *IEEE Trans. Wireless Commun.*, vol. 3, pp. 1250–1259, 2003.

[11] R. Srikant, *The Mathematics of Internet Congestion Control (Systems and Control: Foundations and Applications)*. Springer Verlag, 2004.

[12] M. P. W. Wang and S. H. Low, "Optimal Flow Control and Routing in Multi-path Networks," *Elsevier Perform. Eval.*, vol. 52, no. 2-3, pp. 119–132, 2003.

[13] "Madwifi project." `http://madwifi-project.org`.

[14] E. Kohler, R. Morris, B. Chen, J. Jannotti, and F. M. Kaashoek, "The Click Modular Router," *ACM Trans. Comput. Syst.*, vol. 18, pp. 263–297, August 2000.

[15] D. Giustiniano, E. Goma, A. Lopez Toledo, and P. Rodriguez, "WiSwitcher: An Efficient Client for Managing Multiple APs," in *Proc. of ACM PRESTO Wrkshp.*, (Barcelona, Spain), pp. 43–48, August 2009.

[16] D. Malone, I. Dangerfield, and D. J. Leith, "Verification of Common 802.11 MAC Model Assumptions," in *Proc. of the ACM PAM Conf.*, (Louvain-la-Neuve, Belgium), pp. 63–72, April 2007.

[17] "Minstrel rate control algorithm." `http://linuxwireless.org/en/developers/Documentation/mac80211/RateControl/minstrel`.

[18] "Whisher Wifi Sharing Community." `http://www.whisher.com`.

[19] E. Tan, L. Guo, S. Chen, and X. Zhang, "CUBS: Coordinated Upload Bandwidth Sharing in Residential Networks," in *Proc. of the IEEE ICNP Conf.*, (Plainsboro, NJ), pp. 193–202, October 2009.

[20] N. Thompson, G. He, and H. Luo, "Flow Scheduling for End-host Multihoming," in *Proc. of the IEEE INFOCOM Conf.*, (Barcelona, Spain), pp. 1 – 12, April 2006.

[21] A. Nicholson, S. Wolchok, and B. Noble, "Juggler: Virtual Networks for Fun and Profit," *IEEE Trans. Mobile Comput.*, vol. 9, no. 1, pp. 31–43, 2010.

[22] S. Jakubczak, D. G. Andersen, M. Kaminsky, D. Papagiannaki, and S. Seshan, "Link-alike: Using Wireless to Share Network Resources in a Neighborhood," *IEEE Trans. Mobile Comput.*, vol. 12, no. 4, pp. 1–14, 2008.

[23] D. K. J. Strauss and F. Kaashoek, "A Measurement Study of Available Bandwidth Estimation Tools," in *Proc. of the ACM IMC Conf.*, (Miami Beach, FL), pp. 39 – 44, October 2003.

[24] V. Ribeiro, R. Riedi, R. Baraniuk, J. Navratil, and L. Cot, "pathChirp: Efficient Available Bandwidth Estimation for Network Paths," in *Proc. of the ACM PAM Wrkshp.*, (San Diego, CA), April 2003.

[25] D. Croce, T. En-Najjary, G. Urvoy-Keller, and E. W. Biersack, "Non-cooperative Available Bandwidth Estimation Towards ADSL Links," in *Proc. of the IEEE Global Internet Symp.*, (Phoenix, AZ), April 2008.

[26] D. Croce, T. En-Najjary, G. Urvoy-Keller, and E. W. Biersack, "Fast Available Bandwidth Sampling for ADSL Links: Rethinking the Estimation for Larger-Scale Measurements," in *Proc. of the ACM PAM Conf.*, (Seoul, South Korea), pp. 67–76, April 2009.

[27] J. P. R. Draves and B. Zill, "Routing in Multi-radio, Multi-hop Wireless Mesh Networks," in *Proc. of the ACM MobiCom Conf.*, (New York, NY), pp. 114–128, October 2004.

APPENDIX

If after applying the correction factors in eq. (8), the resulting corrected duty cycles are such that $\sum_i f_{ik} > 1$, we apply the following algorithm to distribute the spare duty cycle:

1. we first reduce the duty cycles for those stations that overestimated it, i.e., we recalculate the adjusted duty cycles f'_{ik} as follows

$$f'_{ik} = \begin{cases} \sigma_{ik} f_{ik} & \text{if } \sigma_{ik} \leq 1 \\ f_{ik} & \text{otherwise} \end{cases}$$

2. Once adjusted, if the demanded duty cycles exceeds the capacity of the card, i.e., $\sum_i f'_{ik} > 1$, then we normalize them $f''_{ik} = f'_{ik} / \sum_i f'_{ik}$. If, on the other hand, there is spare time $f_{sp} = 1 - \sum_i f'_{ik}$, we distribute it among the links that need to increase their duty cycles ($\sigma_{ik} > 1$) proportionally to their needs as follows

$$f''_{ik} = \begin{cases} f'_{ik} + f_{sp} \frac{\sigma_{ik}}{\sum_i \sigma_{ik}} & \text{if } \sigma_{ik} > 1 \\ f'_{ik} & \text{otherwise} \end{cases}$$

3. Each station uses the resulting values f''_{ik}.

AGGRAP: Making WiFi Backhaul Aggregation Practical

Eduard Goma†, Angel Lozano⋆
†Telefonica Research ⋆Universitat Pompeu Fabra
†goma@tid.es ⋆angel.lozano@upf.edu

ABSTRACT

In light of the growing disparity between residential broadband and WiFi speeds, WiFi backhaul aggregation has been proposed as a service whereby residential customers can enjoy higher upload and download throughput when in range of participating APs. The fundamental assumption behind all current such proposals is that WiFi clients can be modified to connect to multiple APs in parallel. This same assumption, however, is the dominant reason why WiFi backhaul aggregation is seeing tremendous friction for commercial deployment: changing every client WiFi device is cost-prohibitive.

In this paper, we develop a solution that builds on previously proposed WiFi aggregation schemes while side-stepping this obstacle. We reformulate the original problem to enable the aggregation service with only AP modifications, and without the need for additional radio hardware. AGGRAP is a single-radio AP that can deliver higher throughput to its clients by aggregating the backhaul capacity of APs within range regardless of their channel of operation. We build AGGRAP with off-the-shelf hardware and evaluate its performance on a 6-node testbed using smartphones and laptops as clients. We describe the conditions under which AGGRAP leads to gains comparable to those of client-based aggregation solutions and demonstrate that the benefits are substantial despite the fact that the additional backhaul capacity must be accessed via a two-hop transmission.

1. INTRODUCTION

Over the last decade, Internet access speeds have significantly increased due to new network deployments such as xDSL[1], Fiber To The Home (FTTH) or cable. Out of these technologies, the most widespread is xDSL that can offer speeds of up to 50Mbps to users that are close to the DSLAM if the telephone cables are in good conditions.

However, the reality is that in many regions, the Internet access speeds remain low (e.g. below 5Mbps). Fig. 1 shows the access speeds of Colombia's internet subscribers as reported by the Ministry of Information and Communication Technologies[14]. The figure shows that out of the 3.37 million Internet accesses in the country, 97% offer speeds below or equal to 4Mbps, that is an order of magnitude below the speeds that WiFi can offer to these users.

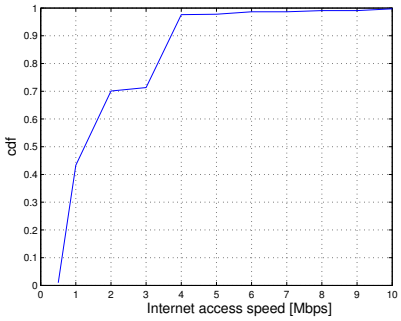


Figure 1: Distribution of Internet subscriptions in Colombia per speed in 2011.

Wireless backhaul bandwidth aggregation has been proposed as a means to increase users' connectivity speed with minimal changes to the existing network deployment [11, 7]. In addition, commercial initiatives like FON[1] aim to create nation-wide WiFi sharing communities that offer the possibility for users to connect to the Internet when they are away from home if they are close to an access point (AP) from the community. Internet Service Providers (ISPs) running wired and cellular access networks see WiFi sharing as an opportunity to offload traffic from the cellular network, and backhaul bandwidth aggregation is a mean for them to increase the quality of their service while they improve their backhaul networks.

However, taking an academic concept to a commercial deployment brings up a number of factors that need to be considered. First, the overall solution costs must be kept low. Thus, introducing extra hardware or requiring highly specialized devices is not an option. Second, the deployment of the solution must not be too complex. Then, the solution must require modifications to the smallest number of devices possible.

Current backhaul bandwidth aggregation solutions [11, 7] require driver modifications on the wireless clients

[1]xDSL comprises ADSL, ADSL+, VDSL...

to enable such scheme. Each wireless client needs to feature a virtualized wireless card. This implies that for a commercial deployment one would need to modify the operating system and wireless driver of every possible WiFi client, including smartphones, tablets, laptops, etc. As one may realize, the cost of such an approach is absolutely prohibitive.

The problem of the diversity of devices that need to be modified can be solved by deploying the aggregation scheme in a much smaller set of devices such as the APs, which are usually provided by ISPs. However, current methods to perform aggregation with single-radio devices are not meant to be used in APs. Introducing a secondary WiFi radio in the APs could provide a technical solution, but it increases the cost of a device that is subsidized by the ISP, making the solution impracticable.

The only way to proceed is to develop a solution that can enable the desired functionality simply through software AP modifications and without any client support. In this paper we present AGGRAP, a single-radio AP that makes possible a commercial deployment of wireless backhaul aggregation. AGGRAP is able to aggregate the unused capacity of neighboring APs regardless of their wireless channel and redirect it to off the shelf clients.

Our contributions can be summarized as follows: 1) we propose a feasible and cost-effective backhaul bandwidth aggregation solution, 2) we formulate the problem of wireless backhaul bandwidth aggregation through a single-radio AP that acts as an AP to its clients, and as a client to neighboring APs, and show that the problem can be mapped to the client-based solutions allowing the same optimization objectives, 3) we evaluate the performance trade-off of the AP- and client-based solutions analytically and experimentally, and 4) we implement AGGRAP and show that it is able to aggregate bandwidth from neighboring backhauls and increase the throughput of different types of off-the-shelf clients like laptops or smartphones.

2. CLIENT-BASED BACKHAUL AGGREGATION

AGGRAP builds over the concepts introduced in client-based backhaul aggregation schemes that we overview in this section.

State of the art client-based aggregation schemes [11, 7, 8] propose the use of TDMA to enable a single-radio client to connect to all the neighboring APs regardless of their frequency of operation. Over cycles of 100 ms the wireless client sequentially connects to all selected APs within range in a round robin fashion (shown in Fig. 2). Using the standard 802.11 power saving feature, a client is able to notify its absence to the APs it is connected to so that they buffer packets directed to it. This way, a client performing aggregation appears to be sleeping in all APs but the one that is currently scheduled in the round robin cycle.

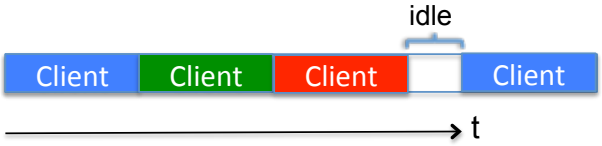


Figure 2: Wireless client TDMA aggregation cycle.

When a client finishes the time allocated to one AP, sends a packet to announce to the AP that is going to power saving mode and that it must buffer packets directed to it. Then the client tunes the WiFi card to the frequency of the next AP in the TDMA cycle and sends a packet to announce its presence and receive all the packets the AP buffered. During the time in which the client is switching frequencies, it can not send/transmit data. This time is called switching time and for state of the art systems it is 1.5ms [8].

As the backhaul capacity and the wireless link capacity from the client to each AP might be different, client-based aggregation systems must optimize the percentage of time devoted to each AP. In [11] authors propose to maximize the individual throughput of each client. In contrast, [7] shows that individual throughput maximization can result in severe unfair throughput distribution and optimizes for weighted proportional fairness to achieve a good compromise between network resource utilization and fairness. In both systems, the throughput obtained from each AP is controlled by the percentage of time devoted to it in the TDMA cycle. Note that a client might be able to collect all the backhaul capacity of neighboring APs and have spare time in which the WiFi card can go to power saving mode and reduce the energy consumption of the system.

3. AGGRAP DESIGN

Our goal is to make WiFI backhaul aggregation a practically deployable solution. To this end, our solution must fulfill the following four requirements:

- **Single-radio APs:** the solution must not increase the cost of the existing hardware.
- **Unmodified WiFi clients:** to provide aggregation to all types of devices, we require no modifications to existing WiFi clients.
- **Limited overhead:** WiFi aggregation targets only the primary AP's immediate neighbors, i.e. we do not introduce connections that transit more than 2 wireless hops.

- **Backhaul connections primarily dedicated to serve owner's devices:** AGGRAPs only aggregate unused bandwidth from neighboring APs.

The high level architecture is that of a community that enables its subscribers to connect to any AP participating in the sharing scheme. Beyond the traditional WiFi sharing, such as FON [1], in our architecture it is not only users that connect to participating APs, but also other APs in order to make use of the additional backhaul capacity. As such, AGGRAP is designed to switch between two modes: i) serving its clients, and ii) acting as a client to other participating APs in order to increase the uplink/downlink throughput to its clients. The latter function is further triggered only when the backhaul link utilization of the primary AP exceeds 80%, a case when assistance becomes desirable.

To realize the full potential of backhaul bandwidth aggregation, AGGRAPs must be able to act as clients of APs operating in a different channel. As one of our requirements is to have single-radio APs, making AGGRAP connect to APs in different frequencies is a challenge that we address in the following section.

3.1 Single-radio multi-channel operations at the AP

To perform bandwidth aggregation using a single-radio AP regardless of the channel of neighboring APs, the TDMA cycle used in client-based aggregation systems needs to be adapted to account for the AP functionalities. AGGRAP must spend the appropriate amount of time in its own frequency forwarding data to its clients collected over its backhaul link and those of its neighboring APs (as shown in Fig. 3). As such, our scheduling decision is slightly more complicated. Furthermore, we need to address one more challenge - that if the AGGRAP leaves its own frequency to utilize the backhaul links of other APs, it may lose packets sent by its clients or even lead them to disconnection. This latter problem was not there in previous client-based formulations since clients have the ability to indicate to their APs that they are going to disappear for a while, usually to enter power save mode, and that the AP should buffer packets for later delivery. However, such a functionality is not present for APs, that are assumed to be always on when they have clients associated with them. We address this issue through the use of the Network Allocation Vector (NAV).

NAV is a counter that each 802.11 device has and represents the amount of time the previous transmission will need before being completed. Prior to transmitting a packet, any 802.11 device estimates the time it will take to send it given the transmission rate and packet length and writes this value into the duration field of the packet MAC header. The other devices in range update their NAV according to the duration field of the packets they receive. This way, none of the 802.11 devices in range will try to access the medium until the end of the transmission. AGGRAP uses the NAV to reserve the channel for the amount of time it will be out of its AP channel. Then, before leaving its primary channel of operation, it sends a packet to a dummy MAC address[2] with a duration field equal to the amount of time it will be acting as a client for all the other APs. The maximum value of the duration field is ≈32ms (15 bits to indicate the duration in microseconds).

Figure 3: Wireless AP TDMA aggregation cycle.

Our design so far addresses our first three requirements. To address the forth requirement, we make sure that AGGRAPs connect to neighboring APs only when they utilize their backhaul link by more than 80%. To further enable the derivation of an appropriate TDMA schedule, AGGRAPs announce through Beacon frames their backhaul link capacity, as well as their available-for-aggregation throughput, i.e. the part of their capacity that is not utilized by their clients. Both values are reported in Mbps and are computed using weighted moving averages, updated every 1s. To further ensure that only the spare capacity of backhaul links is used for assistance, client traffic is prioritized at the AP. Such prioritization happens through the knowledge of which MAC addresses appear as APs, as well as clients. The MAC addresses that act in two roles (e.g. AGGRAPs) across time are classified as lower priority "clients" and the traffic directed to them is not taken into account in the computation of the available-for-aggregation throughput.

Discussion.

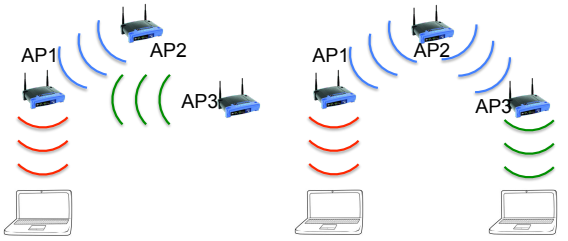
Using the NAV to reserve a channel is an effective measure to silence all the clients and prevent packet losses in the AGGRAP SSID. However, this effect is not limited to the AGGRAP WLAN and will also silence any neighboring network in range using the same channel. Thus, an AGGRAP performing aggregation might reduce the available air-time of neighboring WLANs.

The impact of AP-based aggregation on other networks depends on multiple factors like the number of overlapping networks, the timing of high bandwidth demand of different users or the percentage of time an AGGRAP is collecting bandwidth from other APs. A solution to the potential loss of resources from silencing entire contention domains is to standardize the func-

[2]If the channel reservation packet is transmitted to one of the clients of the AGGRAP, this client might start transmitting at the end of the reception of the packet.

tionality of notifying to all the clients inside an SSID that the AP will not be available for a period of time. Actually, WiFi direct has already specified this functionality to enable power saving for an AP. In WiFi direct, this is called Notice of Absence and allows 802.11 devices to communicate a planned power-down period. Notice that such functionality can also be useful in pacing uplink traffic, a problem described in [12].

3.2 An Example



(a) One client. (b) The second client arrives.

Figure 4: One client obtains the aggregated bandwidth from three APs. When a second client connects to its AP, the network topology changes.

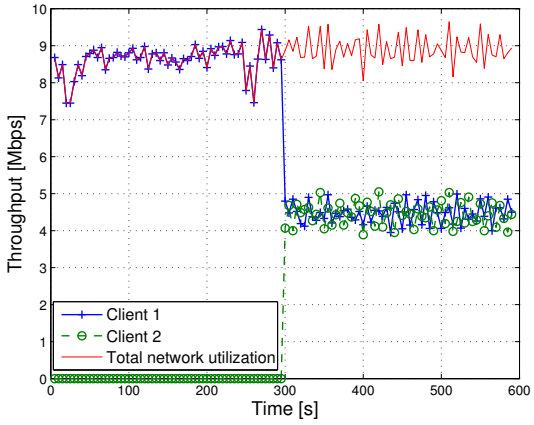


Figure 5: Aggregation in a network with three APs and two clients.

Fig. 4 demonstrates this high level design in a three household scenario. In this particular case, all three AGGRAPs are within wireless range of each other. Initially, the network features a single client that connects to its home AP (AP1). Given that the traffic requirements exceed the capacity of the backhaul link, AP1 connects to AP2 and AP3 to access more wired capacity. Fig. 5 presents the performance of our prototype of AGGRAP in the aforementioned topology. The client of AP1 is able to aggregate the capacity of all 3 backhaul links of 3 Mbps each. At time 300 seconds a second client appears in the network, and connects to its

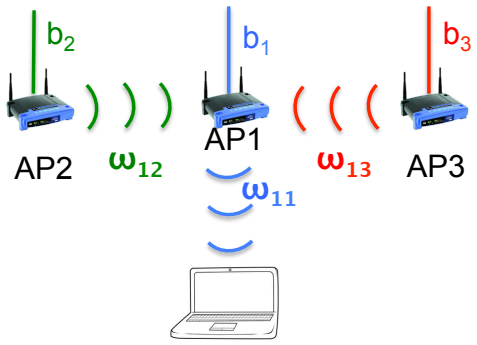


Figure 6: Aggregation with three APs and one client.

home AP (AP3). Given that both clients require more throughput than their backhaul links can provide, they use AP2 for assistance. The system is able to provide a fair share of the backhaul capacity of AP2, leading to an effective throughput of 4.5 Mbps for each client.

4. FORMULATION

AGGRAPs act as normal APs until their clients saturate their backhaul capacity. When AGGRAP's primary clients saturate its backhaul capacity i.e. their utilization is higher than 80% of the backhaul capacity, it will scan for neighboring APs that can provide additional capacity. Out of all the neighboring APs with available backhaul capacity, AGGRAPs select the four APs with best signal to noise ratio (SNR) because it has been shown in [7, 18] that in typical residential environments clients are in range of 4-5 APs with a wireless channel that offers more than 5Mbps. AGGRAPs will assist neighboring APs when they are utilized less than 70%. We leave a 10% margin to avoid AGGRAPs oscillating between the two modes. At that point in time, each AGGRAP needs to compute how much time it will spend serving its own client, and how many additional APs it will connect to and for how long. Our scheme comes up with such a schedule aiming to optimize for weighted proportional fairness, which offers a good compromise between fairness and an efficient utilization of the network resources.

Consider the example in Fig. 6 where only one of the three APs has a client that requires maximum speed connectivity. Each AP is connected to a backhaul link b_i, and the client is connected to its home AP with a wireless channel of effective throughput ω_{11}. AGGRAP$_1$ will try to get the capacity from the two neighboring APs and send the aggregate to its clients. The percentage of time that AGGRAP$_1$ needs to be acting as an AP to be able to send all the backhaul capacity to its clients is

$$f_{11} = \frac{\sum_{\forall i} b_i}{\omega_{11}} \quad (1)$$

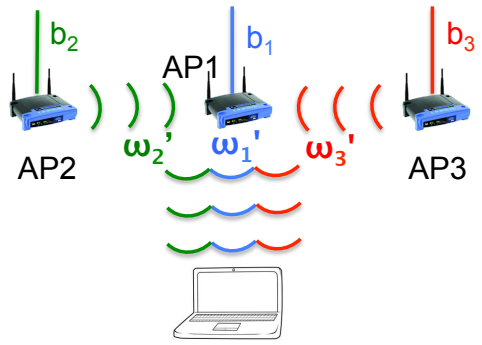


Figure 7: Overlay wireless channels corresponding to each backhaul link.

Variable name	Definition
b_i [Mbps]	Backhaul capacity of AP_i
ω_{ij} [Mbps]	Wireless capacity from AP_j to AP_i
ω_{ii} [Mbps]	Wireless capacity from AP_i to its clients
$f_{ii} \in (0,1]$	Time share that $AGGRAP_i$ devotes to act as an AP
$f_{ij} \in [0,1)$	Time share that $AGGRAP_i$ devotes to act as a client of AP_j

Table 1: Definition of variables used in the formulation of the AP-based aggregation system.

where b_i is the backhaul bandwidth of $AGGRAP_i$, ω_{ij} is the wireless speed at which AP_i can receive data from AP_j and ω_{ii} is the wireless speed at which AP_i can transmit data to its clients. The values of ω_{ij} are estimated following the same process as in [7] while each AP knows its b_i. Note that we consider wireless capacity as the TCP throughput that a client connected 100% of the time to the AP would obtain using 802.11.

The percentage of time required to obtain the bandwidth from neighboring APs is

$$f_{1j} = \frac{b_j}{\omega_{1j}} \tag{2}$$

where ω_{1j} is the wireless speed at which $AGGRAP_1$ can communicate with AP_j, and f_{1j} is the percentage of time that $AGGRAP_1$ spends acting as a client of AP_j. If $\sum_{\forall j} f_{1j} < 1$, $AGGRAP_1$ has enough time to aggregate all the backhaul bandwidth and send it to its client.

However, depending on the wireless channel quality and the capacity of the backhaul links it will not always be possible for AGGRAPs to aggregate all the available capacity. In these situations it is necessary to optimize the time allocation to obtain a weighted proportionally fair throughput distribution among APs.

Let us then formulate the problem of f_{ij} allocation. We see that f_{ii} (the time $AGGRAP_i$ spends acting as an AP for its clients) must not be smaller than $\frac{b_i}{\omega_{ii}}$. And it must also account for the time required to send the data that has been collected from other APs (i.e.- $\frac{f_{i2}\omega_{i2}}{\omega_{ii}}$ for AP_2). Then, f_{ii} satisfies:

Variable name	Definition
ω'_i [Mbps]	Wireless capacity from AP_i to AGGRAP's clients
$f'_i \in (0,1]$	Time share that $AGGRAP_i$ devotes to data coming from backhaul b_i

Table 2: Definition of variables used in the formulation of the AP-based aggregation system mapped into the client-based solution.

$$f_{ii} = \frac{b_i}{\omega_{ii}} + \sum_{\forall j \neq i} \min\left(\frac{f_{ij}\omega_{ij}}{\omega_{ii}}, \frac{b_j}{\omega_{ii}}\right) \tag{3}$$

where $\min(\frac{f_{ij}\omega_{ij}}{\omega_{ii}}, \frac{b_j}{\omega_{ii}})$ models the percentage of time required to transmit the data that can be obtained from AP_j. The limit imposed by $\frac{b_j}{\omega_{ii}}$ is only active when $AGGRAP_i$ has enough time to aggregate and transmit all the backhaul data. In all other situations, the amount of bandwidth that can be obtained from AP_j is given by the percentage of time using the link multiplied by its capacity ($f_{ij}\omega_{ij}$). Additionally,

$$\sum_{\forall j} f_{ij} \leq 1, \tag{4}$$

$$f_{ij} \leq \frac{b_j}{\omega_{ij}}, \forall j \neq i, \tag{5}$$

There is no closed form solution for the problem stated above. However, it is possible to map the problem of f_{ij} allocation at the AGGRAPs to the problem already solved in [7] of wireless backhaul aggregation from the clients.

Let us now focus on the three sources of bandwidth available and how to allocate air time to each data flow (see Fig. 7). Then, we define an overlay wireless channel per each backhaul link and the percentage of time needed to transmit data over a certain overlay channel is given by the ratio

$$f'_i = \frac{b_i}{\omega'_i} \tag{6}$$

where b_i is the backhaul bandwidth available and ω'_i is the wireless capacity of the overlay channel. Comparing Fig. 6 and Fig. 7 we see that $f'_1 = \frac{b_1}{\omega'_1}$, where $\omega'_1 = \omega_{11}$. Nevertheless, to compute $f'_2 = \frac{b_2}{\omega'_2}$ it is necessary to know ω'_2. Now ω'_2 needs to model the wireless capacity of the channel that links the backhaul link of capacity b_2 with the wireless client. Then, ω'_2 can be obtained using the time required to transmit a packet of size P through both hops:

$$tx\_time = \frac{P}{\omega_{12}} + \frac{P}{\omega_{11}} = P\frac{\omega_{11} + \omega_{12}}{\omega_{11}\omega_{12}}, \tag{7}$$

Then,

$$\omega'_2 = \frac{\omega_{11}\omega_{12}}{\omega_{11} + \omega_{12}}. \quad (8)$$

Following the same process with ω'_3 it is possible then, to reduce the problem of backhaul bandwidth aggregation to the allocation of f'_1, f'_2 and f'_3 with the same constraints as in [7] to guarantee a fair throughput allocation among all the clients.

Once the percentage of time of each bandwidth source is computed, it is necessary to map these values into the real percentages of time that AGGRAP will use (f_{11}, f_{12} and f_{13}).

From the change in notation that we have previously done it is clear to see that f'_1 accounts for the time that AGGRAP$_1$ requires to transmit b_i to its clients and does not capture the time required to send the data that it could aggregate from neighboring APs. Then, f'_2 accounts for the percentage of time required to transmit bandwidth from b_2 to AGGRAP$_1$ and from AGGRAP$_1$ to its clients. Which represents f_{12} plus some portion of f_{11}. Let us focus on the percentage of time to transmit data from AP$_2$ to AP$_1$. The mapping between f_{12} and f'_2 can be seen from the percentage of time that a packet of size P spends on each of the two links given ω_{11} and ω_{12}.

$$(\%\ of\ time\ on\ link\ 1-2) = \frac{\frac{P}{\omega_{12}}}{\frac{P}{\omega_{12}} + \frac{P}{\omega_{11}}}, \quad (9)$$

$$(\%\ of\ time\ on\ link\ 1-2) = \frac{\omega_{11}}{\omega_{11} + \omega_{12}} \quad (10)$$

Then,

$$f_{12} = f'_2 \frac{\omega_{11}}{\omega_{11} + \omega_{12}} \quad (11)$$

and

$$f_{11} = f'_1 + f'_2 \frac{\omega_{12}}{\omega_{11} + \omega_{12}} + f'_3 \frac{\omega_{13}}{\omega_{11} + \omega_{13}} \quad (12)$$

Table 3 shows the values of f'_i and ω'_i for the example shown in Fig. 6 considering that all three backhaul links are of capacity $b_i = 1Mbps$. It can be seen that after the conversion from the overlay channels to the real wireless channels, the values obtained for f_{ij} are the same as the ones that could be computed using eq. 1 and eq. 2.

To solve this example, it is not necessary to use the change of variables because AGGRAP is able to aggregate all the backhaul bandwidth. However, in case AGGRAP needs to select the best time allocation among the neighboring APs to provide a proportionally fair share of the backhaul capacity, using the overlay wireless channels provides a way to map a problem with recursive definitions of variables into a problem with known solutions [11, 7]. AGGRAP uses the solution described in [7] to select a weighted proportionally fair throughput distribution among AGGRAPs which we summarize in the appendix.

	AP$_1$	AP$_2$	AP$_3$
b_i [Mbps]	1	1	1
ω_{ij} [Mbps]	$\omega_{11} = 20$	$\omega_{12} = 10$	$\omega_{13} = 10$
ω'_i [Mbps]	$\omega'_1 = 20$	$\omega'_2 = 6.67$	$\omega'_3 = 6.67$
f'_i	$f'_1 = 0.05$	$f'_2 = 0.15$	$f'_3 = 0.15$
f_{ij}	$f_{11} = 0.15$	$f_{12} = 0.10$	$f_{13} = 0.10$

Table 3: Example of the mapping between the three "overlay channels" and the real wireless channels.

5. EXPERIMENTAL EVALUATION

In this section we briefly describe the implementation of AGGRAP on off the shelf hardware and its performance on a small scale testbed. Our evaluation focuses on: i) the performance implications of using an AP to both serve clients, as well as act as a client itself, ii) the assessment of the accuracy of the problem reformulation as described in Section 4, and iii) the performance gains of AGGRAP in a realistic environment. Our experiments were performed in an office environment using off the shelf devices, such as laptops and Android phones.

5.1 Implementation

We have implemented AGGRAP in a desktop computer with linux kernel 2.6.32 and an atheros based 802.11 PCI Express card controlled by the ath9k[13] linux driver. The ath9k code of compat-wireless-2.6.32 has an initial implementation of multi-channel virtualization. We modified that code to enable TDMA scheduling, while limiting any associated loss in performance.

- To control that the last packet sent before switching was the one indicating that AGGRAP is going to power save mode, we i) first stopped the upper layer queues and then started the switching process, and ii) reduced the number of hardware queues from "4" to "1"[3].
- We reduce the hardware queue length to increase the accuracy of the switching events.

Furthermore, we implemented the silencing mechanism by which AGGRAPs can silence their contention domain to mask the disappearance from their channel of operation to their clients. As with the original design requirements, our clients remain totally unmodified, thus allowing us to perform experiments with laptops, as well as smartphones.

5.2 Experimental Setup

Our testbed consists of two AGGRAPs, one linksys WRT54GL AP and three machines acting as servers on the wired network. To study the impact of different backhaul capacities on the performance of AGGRAP,

[3]The number of "4" queues is to enable wireless multimedia extensions (WME).

we use traffic control (tc) to rate limit the backhaul links. To study the impact of different wireless capacities, we modify the transmission rate of APs. We use Dell laptops (latitude D620) running ubuntu 8.04 (linux kernel 2.6.24) to connect to the AGGRAPs to perform the majority of the throughput experiments we show in this paper. In addition, we have tested Android phones (e.g. we have tested an HTC Nexus One and a Samsung Galaxy SII) connecting to the AGGRAPs and obtained the same performance as that obtained the laptops. We use downlink Iperf[19] TCP connections to measure the throughput achieved in each scenario. All experiments are conducted during night to avoid interference from networks that do not belong to the test bed.

5.3 Single-radio multi-channel virtualization

Before proceeding with the performance evaluation of AGGRAP as a whole, we need to evaluate whether single-radio multi-channel virtualization incurs any overhead when one of the virtual interfaces acts in AP mode. To this end we deploy the network depicted in Fig. 13(b), in which an AGGRAP acts as an AP and as a client of a neighboring AP. AGGRAP acts as an AP in channel 1 (2.412 GHz) while the neighboring AP is set to channel 5 (2.432 GHz). Using this network topology, we perform a first experiment to show the evolution of the throughput obtained in each interface depending on the percentage of time devoted to it. In this experiment we simultaneously start two TCP connections: i) one from AGGRAP to its client; and ii) one from the backhaul of the neighboring AP to AGGRAP. We apply no restrictions to the backhaul capacity of the APs, thus allowing us to assess the WiFi network alone. All APs transmit at the maximum physical layer transmission rate possible in 802.11g (i.e.- 54Mbps).

Fig. 8 shows the results of the experiment. We denote as "Home Throughput" the bandwidth obtained in the link between the AGGRAP and its client and "Borrowed Throughput" the bandwidth that AGGRAP is able to receive using the neighboring AP. The line labeled "Aggregate Throughput" in Fig. 8 refers to the sum of "Home Throughput" and "Borrowed Throughput" which shows the global utilization of the wireless card.

Fig. 8 shows that Home Throughput increases linearly with the percentage of time that the AP serves traffic to its client. This is to be expected since we have no restriction on the AP's backhaul capacity. Similarly, as the percentage of time in AP mode increases, we are effectively reducing the amount of time that the AP receives traffic from the neighboring AP. Indeed, Borrowed Throughput is decreasing as we move to the right. More importantly, if we add up the two metrics, the aggregate throughput remains almost constant across the different experiments, indicating that the overhead of

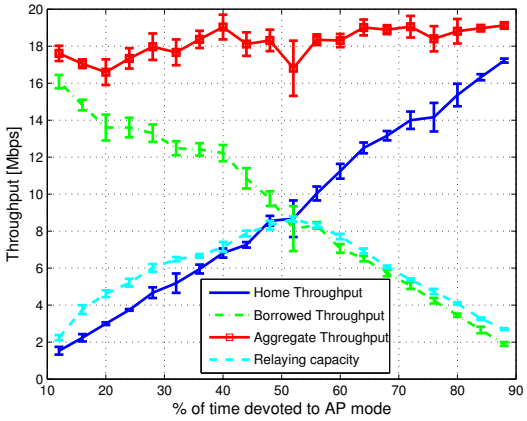


Figure 8: Comparison between the relaying capacity and the single-hop link capacity.

switching from AP to client is fixed and only impacts overall aggregate throughput by 3.2 Mbps. Our implementation requires 1.5 ms to switch from one virtual interface to another that is operating on a different frequency. But we observe that after switching frequency and sending the packet notifying the neighboring AP it is back from power saving, there is a period of time in which the card does not transmit any data. This period has a mean duration of 6 ms and is due to the design of the interactions between the driver (ath9k) queues and higher layers (mac80211). As we observe, using a TDMA cycle of 100ms, the impact of switching is of 15% of the capacity of 802.11 without switching[4].

In the presence of a 2-hop flow from the neighboring AP to the client, one would expect to observe a throughput that is the minimum of the Home and Borrowed Throughput. We instantiate such a connection across the network (in isolation this time) and vary the amount of time that the Home AP serves its client. We plot the result as Relaying Throughput. Indeed, the line nicely tracks the minimum of the Home and Borrowed Throughput.

Lastly, we observe that the total utilization of the wireless card when performing channel switching reaches at most 19Mbps. This value is actually $\omega_{11-Switching} = 0.86 \times \omega_{11-No-switching}$. This shows the impact of the channel switching overhead.

Therefore, virtualization of the WiFi card to act as both AP and client on different frequencies does lead to the expected behavior in our implementation. Having assessed the validity of our prototype we move to test the accuracy of the problem reformulation presented in Section 4.

5.4 Formulation validation

[4]We observe the same behavior of the driver using two interfaces in client mode.

In section 4 we show that using a change of variables it is possible to map the formulation of the AP-based aggregation problem to the client-based one and, then, apply the best solution from the ones already proposed in the literature.

5.4.1 Change of variables

The change of variables that helps mapping the new optimization problem into a known one consists on predicting the capacity of the two-hop link between a neighboring AP and AGGRAPs clients (ω_2') based on the wireless capacity of each hop in the communication (ω_{11} and ω_{12}). Then, $\omega_2' = \frac{\omega_{11}\omega_{12}}{\omega_{11}+\omega_{12}}$.

The change of variables comes from measuring the transmission time of a packet throughout each link and add these times to find the average capacity in the two links. This assumption is perfectly valid in paper, but we need to confirm that this is the case in our prototype and that the processing time of the packets or the buffering performed in AGGRAP before re-transmitting packets to the clients do not affect ω_2.

To this end, we measure ω_{12} for all the transmission rates of an 802.11g AP[5]. This simulates the effect of having the neighboring AP farther away with lower wireless capacity. To measure ω_{12} we performe 5 iperf tests of 50 seconds each between the neighboring AP and AGGRAP using only one virtual interface in client mode. We also measure AGGRAPs relaying capacity varying the percentage of time devoted to the AP interface. The maximum relaying capacity in each of these tests corresponds to the experimental ω_2'. In addition, we measure $\omega_{11} = 22.2Mbps$ and compute the expected ω_2' for each of the experiments.

Fig. 9 shows the result of dividing the experimental ω_2' over the expected ω_2'. We observe that the experimental values of ω_2' are 0.8 times the expected values. Which corresponds to the impact of our implementation of fast channel switching. We see then, that taking into account the cost of switching the values of ω_2' are the ones we compute using the change of variables.

We see then that all the values estimated in absence of switching will actually be reduced when performing channel switching. And now

$$\omega_2' = \frac{\gamma^2\omega_{11}\omega_{12}}{\gamma(\omega_{11}+\omega_{12})} = \gamma \times \frac{\omega_{11}\omega_{12}}{\omega_{11}+\omega_{12}} \quad (13)$$

where $\gamma = 0.8$ is a constant that takes into account the ratio between the experimental and the expected ω_2'.

We see then that the change of variables maps AGGRAP's formulation into a client-based backhaul aggregation system.

5.4.2 Fairness

[5]We do not show the results for the transmission rates of 11Mbps and 5.5Mbps as they are similar to the ones obtained for 12Mbps and 6Mbps.

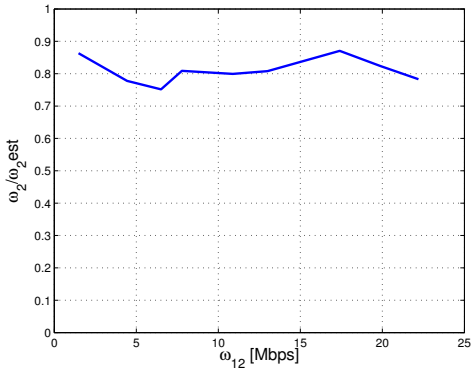


Figure 9: Accuracy ω_2.

Mapping the AP-based aggregation problem to the client-based one offers the possibility to optimize for the same goals that have been explored in the literature [11, 7]. However, performing the optimization computation in a different point of the network has some implications in the final result. Client-based solutions optimized for individual maximum throughput [11] or per-client fairness [7]. However, in the AP-based solution it is possible to optimize for the AP's maximum throughput delivered to its users or per-AP fairness.

Consider the network deployment shown in Fig. 10 where laptop1 connects to $AGGRAP_1$ and laptop2 and the Android phone connect to $AGGRAP_3$ and the capacity of each backhaul is 3 Mbps. All APs are in range of each other with maximum wireless capacity (20 Mbps) and clients communicate to their AP with the same wireless capacity. Then, we run an experiment in which laptop1 and the Android phone open 5 TCP downlink flows per backhaul they are using while laptop2 only opens one downlink TCP flow per backhaul. Fig. 11 shows the results of the experiment which clearly present the implications of running the fairness optimization in the AP. We observe that $AGGRAP_1$ and $AGGRAP_3$ offer the same bandwidth to the sum of their clients, but the Android phone obtains a greater share of the 4.5Mbps that $AGGRAP_3$ is offering. This is because the control of the throughput obtained in each wireless link is done at the AP instead of at the clients. And the sharing of the aggregate capacity of $AGGRAP_3$ is controlled by the TCP congestion control that provides per-flow fairness.

5.5 Aggregation capacity

To explore the maximum aggregation capacity of AGGRAP, we use the network topology depicted in Fig. 13(b). Now we setup the capacity of the home backhaul to 3Mbps and explore what is the maximum bandwidth that can be aggregated from a neighboring AP with a perfect channel — the neighboring AP is transmitting at a physical rate of 54Mbps —. In this scenario

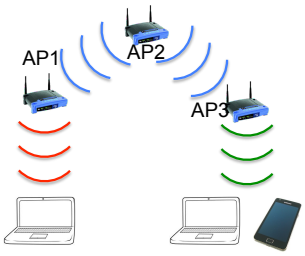


Figure 10: Topology of the fairness experiment.

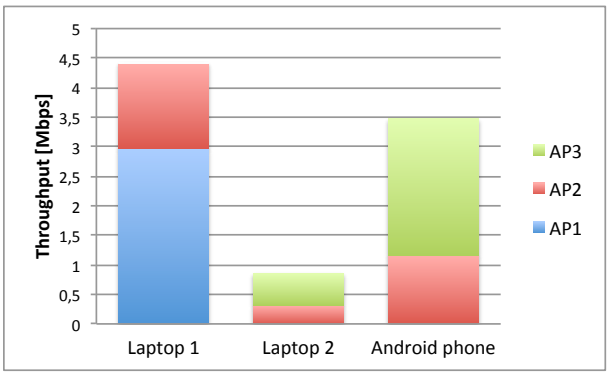


Figure 11: Fairness of AGGRAP.

AGGRAP provides a total throughput of 9.9Mbps to its client: 3Mbps coming from the home backhaul and 6.9Mbps gathered from the neighboring AP. The operation point of AGGRAP is to devote 40% of the time to collect bandwidth from the neighboring AP and the remaining 60% to serve the aggregated throughput to its client.

Fig. 12 shows the evolution of the total throughput obtained by the client of the AGGRAP network depending on the amount of time that AGGRAP devotes to acting as an AP. Note that when the percentage of time devoted to the AP interface is lower than the optimum value, the throughput obtained from the home backhaul is lower than 3Mbps. This is because the bottleneck in this situation is the link between AGGRAP and its client and all the TCP flows will compete to obtain the available capacity. Instead, when AGGRAP allocates 60% of the time or more to the AP interface, the bottleneck moves to the two backhauls i.e. the home backhaul and the link to the neighboring AP. AGGRAP must always operate in this point.

6. CLIENT- VS AP-BASED BANDWIDTH AGGREGATION

We have seen that it is technically possible to perform bandwidth aggregation using a single-radio AP and that it yields benefits for different types of WiFi clients. However, the appearance of multi-hop wireless links increases the complexity of the resulting network architecture. It is necessary then to analyze what are

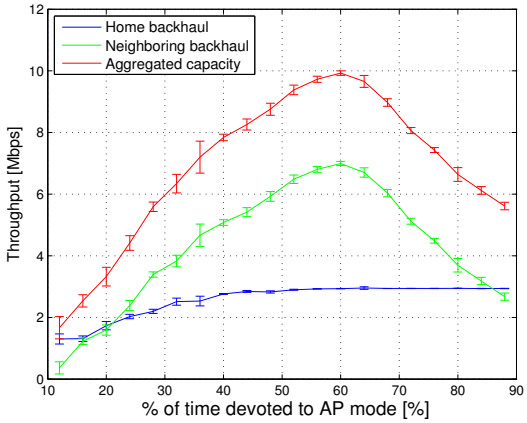
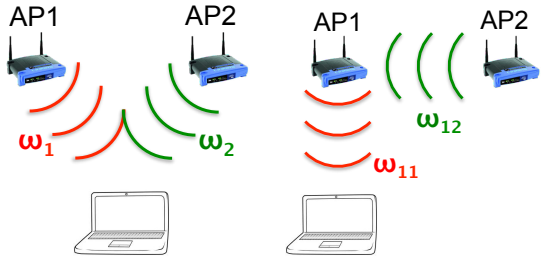


Figure 12: Aggregation capacity with a home backhaul of 3Mbps.

the drawbacks of using AGGRAP and quantify its potential benefits to assess its viability.

In order to perform an easy to follow comparison between both systems, we will make an analytical computation of the maximum bandwidth each system can aggregate in a scenario that can be solved with back of the envelope calculations. The scenario selected is: one client that can see a neighboring AP with the same quality as its home AP ($\omega_2 = \omega_{12}$). Fig. 13 shows the network configuration of the client-based and AP-based solutions for the scenario selected for the comparison.



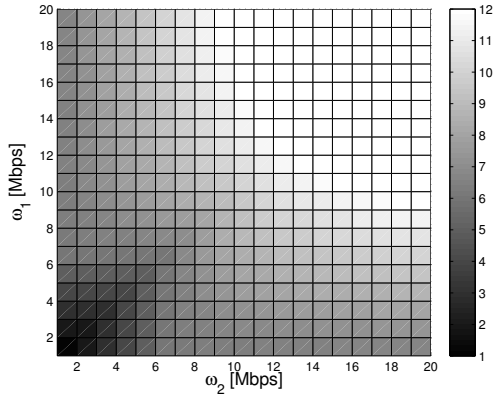
(a) Client-based solution. (b) AP-based solution

Figure 13: Topology of the studied network.

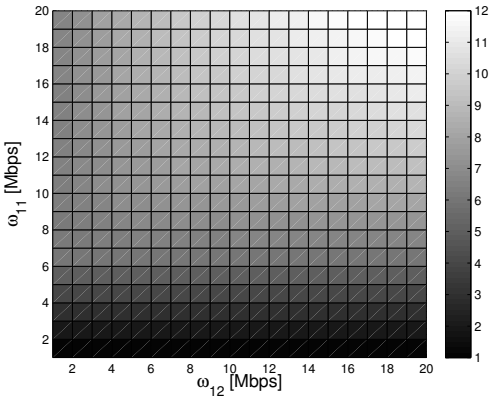
6.1 All backhauls are the same

Considering that each AP is connected to a 6Mbps wired backhaul, the throughput that a client-based solution can offer for different wireless capacities is depicted in Fig. 14(a). If we compare it to the throughput offered by the AP-based solution shown in Fig.14(b), the difference in aggregation capacity is noticeable. Comparing both figures, we observe that the client-based solution is able to fully aggregate the available 12Mbps of the backhaul links for a wider range of wireless channel capacities.

Another difference between both systems shows up in

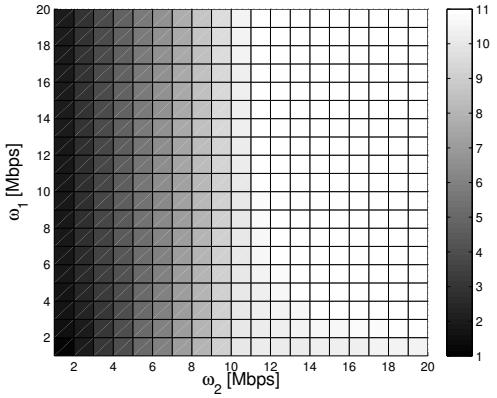


(a) Client-based aggregation capacity

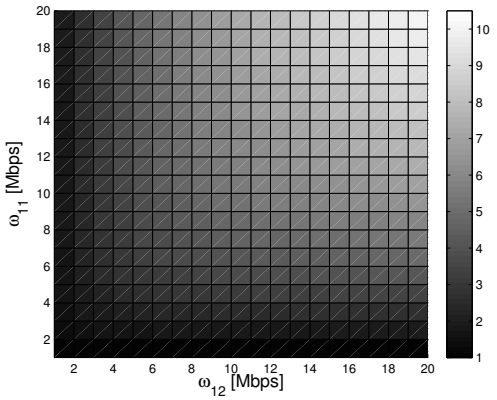


(b) AP-based aggregation capacity

Figure 14: Aggregation capacity of client- and AP-based systems with a home and a neighboring backhaul of 6Mbps.



(a) Client-based aggregation capacity 2



(b) AP-based aggregation capacity 2

Figure 15: Aggregation capacity of client- and AP-based systems with a home backhaul of 1Mbps and a neighboring backhaul of 10Mbps.

these figures as the client based shows no preference for the AP with better quality while the AP-based solution is tied to a wise decision of the AP selected. This can be seen from the fact that Fig. 14(a) is symmetric while Fig. 14(b) is not. Hence, if the home AP wireless capacity is very low, the client will obtain at most that capacity regardless of the channel quality of the link to the neighboring AP.

6.2 Different backhauls

Let us now focus in the same scenario described in the previous section but changing the capacity of the backhaul links to 1Mbps for the home backhaul and 10Mbps for the neighboring one.

Fig. 15(a) shows the results for a client-based solution while Fig. 15(b) shows the throughput that AGGRAP provides to its clients. Comparing the results from both systems we observe that:

- The client-based solution achieves 0.5Mbps higher throughput.
- AGGRAP can not aggregate when one of the two wireless links is bad.
- The client-based solution selects the neighboring AP to collect the bandwidth.
- Respect to no aggregation, AGGRAP offers at least 4 times higher throughput to its clients when $\omega_{12} > 5Mbps$.

In this scenario, the home backhaul contributes 9% of the total capacity. Then, the benefit of obtaining this capacity is minimal compared to the neighboring capacity. The result of this situation is that when the link to the neighboring AP offers higher capacity than the home backhaul, the maximum throughput is obtained by connecting to the neighboring AP. When users are at home, they use their home AP and aggregate as much as possible regardless of the imbalance in capacity. Nevertheless, this scenario shows the importance of the AP selection algorithm when users are not at home.

Further reducing the capacity of the home backhaul would lead to a situation in which AGGRAP becomes

a relay of the neighboring AP. In case the home AP is in between the client and the neighboring AP, the client will obtain higher throughput using the home AP as a relay than trying to directly connect to the neighboring AP. This leads to a situation similar to the scenario studied in [6].

6.3 Conclusions

The comparison shown above drives the following conclusions:

- Client-based solutions are more efficient.
- AP-based solutions yield significant throughput increases when there is an imbalance between home backhaul and wireless capacity.
- AP-based solutions are highly dependant of the AP-client link.

7. RELATED WORK

WiFi sharing. The use of WLANs to share backhaul resources has been discussed extensively [10, 4, 1, 2]. In [10], the authors propose to normally utilize the wire line connection to the home AP and increase the uplink capacity by connecting to neighboring APs and aggregating its backhaul capacity. In addition, FON [1] and Meraki [2] have shown, by creating WiFi sharing networks, that it is not only the research community that has interest in this topic. However, none of these approaches has addressed the possibility to aggregate the backhaul bandwidth of multiple APs.

Wireless card virtualization. The first work that proposed the use of virtualization as a means for enabling simultaneous connections to multiple WLAN networks was Multinet [16]. Juggler [15] provides a description of an implementation that reduces the switching time to 3ms to enable fast switching. However, the closest systems to AGGRAP are FatVAP [11], THEMIS [7], which perform WiFi backhaul bandwidth aggregation using single-radio clients. FatVAP optimizes the system for maximum throughput per client, but leads to unfair bandwidth distribution in certain scenarios that could refrain users from participating in the sharing scheme. THEMIS solves this problem by guaranteeing a fair sharing of the backhaul bandwidth and offers the possibility of providing price-based priorities to enable commercial deployments. These solutions require driver modifications to the clients and are difficult to deploy in large deployments because the number of different client devices results in high deployment costs. In contrast, AGGRAP provides a way to achieve backhaul bandwidth aggregation modifying only a subset of devices that can be controlled by the network operator.

Multi-hop wireless networks The use of multi-hop wireless networks to provide high-speed network connectivity was explored in [5], where the authors deploy a mesh network to grant high-speed broadband access to a poor neighborhood. The main idea of the network deployment is to spread the capacity of a high-bandwidth backhaul across a wide area using the deployment easiness of WiFi equipment instead of incurring the costs of deploying wire line infrastructure to every single household. Then, users of such networks obtain connectivity through multi-hop WiFi connections. AGGRAP makes use of multi-hop connections to obtain the backhaul bandwidth of other APs but its goal is to increase the throughput of its clients by aggregating multiple backhauls.

8. CONCLUSIONS AND FUTURE WORK

In this paper, we point out a fundamental problem with prior WLAN bandwidth aggregation solutions: their need for client modifications makes their deployment cost prohibitive. To unleash the potential of those solutions, we design and implement a system that can approach the benefits of client-based solutions requiring modifications only on the APs. We show that the formulation of such a system is a variation of the original problem definition, and propose an architecture (AGGRAP) to realize it. We compare the performance of AGGRAP with that of client-based solutions and show that, while less efficient aggregating bandwidth, the former can still yield substantial throughput increases. We further show that AGGRAP offers throughput increases to a variety of unmodified WiFi devices (e.g., laptops and different Android phones). In this exploration, we further uncover new research questions that target the following:

Flow scheduling. Given that AGGRAP performs a per-flow scheduling allocating each TCP flow to the best backhaul links, what is the optimal flow scheduling to foster the best user experience?

Client allocation. Given that AGGRAPs are static and have information about the state of its neighboring peers, it is possible to coordinately move clients from one AP to another [9]. Then, it is possible to optimize the client allocation to minimize energy consumption or maximize network utilization depending on the user's traffic demand.

9. REFERENCES

[1] FON. `http://www.fon.com/`.

[2] Meraki. `http://www.meraki.com/`.

[3] Virtualization in windows phones. `http://msdn.microsoft.com/en-us/library/windows/hardware/ff570906%28v=vs.85%29.aspx`.

[4] Ai, X., Srinivasan, V., and Tham, C.-K. Wi-Sh: A Simple, Robust Credit Based Wi-Fi Community Network. In *Proc. of the IEEE INFOCOM Conf.* (April 2009).

[5] Camp, J., Mancuso, V., Gurewitz, O., and Knightly, E. A measurement study of multiplicative overhead effects in wireless networks. In *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE* (april 2008), pp. 76 –80.

[6] Gambiroza, V., Sadeghi, B., and Knightly, E. W. End-to-end performance and fairness in multihop wireless backhaul networks. In *Proceedings of the 10th annual international conference on Mobile computing and networking* (New York, NY, USA, 2004), MobiCom '04, ACM, pp. 287–301.
[7] Giustiniano, D., Goma, E., Lopez Toledo, A., Dangerfield, I., Morillo, J., and Rodriguez, P. Fair WLAN Backhaul Aggregation. In *Proc. of MobiCom'10* (2010).
[8] Giustiniano, D., Goma, E., Toledo, A. L., and Rodriguez, P. WiSwitcher: An Efficient Client for Managing Multiple APs. In *Proc. of PRESTO'09* (2009).
[9] Grunenberger, Y., and Rousseau, F. Virtual access points for transparent mobility in wireless lans. In *In Proceedings of IEEE WCNC* (Sydney, Australia, Apr. 2010).
[10] Jakubczak, S., Andersen, D. G., Kaminsky, M., Papagiannaki, D., and Seshan, S. Link-alike: using wireless to share network resources in a neighborhood. *SIGMOBILE Mob. Comput. Commun. Rev. 12*, 4 (2008), 1–14.
[11] Kandula, S., Lin, K. C.-J., Badirkhanli, T., and Katabi, D. FatVAP: Aggregating AP Backhaul Capacity to Maximize Throughput. In *Proc. of NSDI'08* (2008).
[12] Li, Y., Papagiannaki, D., and Sheth, A. Uplink traffic control in home 802.11 wireless networks. In *Proc. of ACM HomeNets'11* (2011).
[13] Linux Wireless. ath9k linux driver. `http://linuxwireless.org/en/users/Drivers/ath9k`.
[14] Ministry of Information and Communication Technologies of Colombia. Report of Internet statistics in Colombia. `http://www.mintic.gov.co/images/documentos/cifras_del_sector/informe_2011_4T.zip`.
[15] Nicholson, A. J., Wolchok, S., and Noble, B. D. Juggler: Virtual Networks for Fun and Profit. *IEEE Transactions on Mobile Computing 9* (2010), 31–43.
[16] R. Chandra, and Bahl, P. MultiNet: connecting to multiple IEEE 802.11 networks using a single wireless card. In *Proc. of IEEE INFOCOM '04.* (2004), vol. 2.
[17] Srikant, R. *The Mathematics of Internet Congestion Control (Systems and Control: Foundations and Applications).* Springer Verlag, 2004.
[18] Thompson, N., He, G., and Luo, H. Flow Scheduling for End-host Multihoming. In *Proc. of INFOCOM'06* (2006).
[19] University of Illinois. Iperf. `http://iperf.fr/`.
[20] W. Wang, M. P., and Low, S. H. Optimal Flow Control and Routing in Multi-path Networks. *Elsevier Perform. Eval. 52*, 2-3 (2003), 119–132.

APPENDIX

In the following lines we summarize the formulation used in [7] to provide a fair sharing of the total backhaul bandwidth among users.

Let $\mathcal{S}$ be the set of stations and $\mathcal{A}$ the set of APs. Denote T_{ik} as the throughput sent from AP_i to station k. And let $y_k = \sum_{i\in\mathcal{A}} T_{ik}$ denote the total throughput received by station k. Let $U(\cdot)$ be a differentiable, strictly concave, increasing function which represents the utility at every station as a function of the received throughput. The fairness problem is modeled as

$$\max \sum_{k\in\mathcal{S}} U(y_k) \tag{14}$$

$$\text{s. t.} \quad \sum_{k\in\mathcal{S}} T_{ik} \le b_i, \quad \forall i \in \mathcal{A}, \tag{15}$$

$$\sum_{i\in\mathcal{A}, w_{ik}>0} \frac{T_{ik}}{w_{ik}} \le 1, \forall k \in \mathcal{S}, \tag{16}$$

$$T_{ik} \ge 0, \quad \forall i \in \mathcal{A}, \forall k \in \mathcal{S}, \tag{17}$$

where w_{ik} is the wireless capacity[6] at which station k can receive from AP_i, that takes into account the interference from other clients connected to that AP, and b_i is the backhaul capacity of AP_i.

Eq. (15) is the *AP_i backhaul capacity constraint*, and ensures that the total traffic traversing the AP backhaul i does not exceed the backhaul capacity b_i. Eq. (16) corresponds to the *station k wireless capacity constraint*, and guarantees that the total traffic received by station k does not exceed the total capacity of its wireless interface. Finally (17) forces the values T_{ik} to be positive.

Finally, the utility function chosen is weighted proportional fairness given by $U(y_k) = K_k \cdot \log y_k$, where K_k represents the relative priority of user k (for example, a value linearly dependent with the AP backhaul bandwidth owned by user k). If all the users have the same priority $K_k = 1$.

Decomposition and Interpretation

As described in [17], the solution of the above optimization problem can be obtained via the primal-dual formulation using a gradient descent algorithm. From there authors in [7] derive the following optimal rate update rule

$$T_{ik} = \hat{T_{ik}} + \alpha \left(U'(y_k) - p_i - q_{ik} \right), \tag{18}$$

where $\hat{T_{ik}}$ is the bandwidth request in the previous step of the algorithm, $U'(y_k)$ is the derivative of the utility function evaluated at the current throughput received by the station y_k, and α is the step size of the rate update algorithm[7]. The quantities p_i and q_{ik} are the prices corresponding to constraints (15) and (16) respectively, calculated as follows

$$p_i = \left[\hat{p_i} - \frac{\delta}{b_i} \left(\lambda b_i - \sum_{k\in\mathcal{S}} T_{ik} \right) \right]^+, \tag{19}$$

$$q_{ik} = \left[\hat{q_{ik}} - \frac{\gamma}{w_{ik}} \left(\mu - \sum_{i\in\mathcal{A}} \frac{T_{ik}}{w_{ik}} \right) \right]^+, \tag{20}$$

where $\hat{p_i}$, $\hat{q_{ik}}$ are the prices obtained in the previous step of the algorithm, and δ and γ are the step sizes of the price update algorithm. In order to improve the network utilization, and as suggested in [20], we normalize the price step size by the link capacities to favor good links. Finally $\lambda, \mu \le 1$ are the congestion thresholds and $(x)^+ = \max(x, 0)$.

The price p_i in (19) represents the level of congestion on the backhaul of AP_i, and it is a linear function of its available bandwidth. Similarly, q_{ik} in (20) represents the level of congestion on the wireless link from station k to AP_i, and it is a function of the available card time at the station. As congestion increases, the respective prices will increase and the throughput demand T_{ik} of station k through AP_i will decrease according to (18).

The values λ and μ are the *congestion thresholds*, i.e., the level of utilization of the AP_i backhaul and the wireless radio-interface of station k that will trigger the algorithm congestion control, causing the prices p_i and q_{ik} to increase respectively, prompting the throughput requests for their respective paths to decrease.

[6] Note that $w_{ik} = 0$ if station k does not connect to AP_i.
[7] When using proportional fairness, and in order to reduce oscillations as suggested by [20], $\alpha = \alpha' y_k$, with α' the new step size.

Experimentation of Multipath HTTP Streaming over Internet

Stéphane Gouache, Guillaume Bichot and Christopher Howson
Technicolor, Rennes, France
E-mail : <first name>.< family name> @technicolor.com

Abstract: **Streaming over HTTP has become popular in recent years, enabling continuous delivery of live and VOD content without the need for guaranteed bandwidth. It is however challenging to maintain a smooth quality of experience for high value content such as High Definition and 3D. Capitalizing on a distributed adaptive streaming framework based on standard HTTP range requests, we experiment a real implementation over Internet. We discuss two implementation schemes, present an overview of our tests in a simulation environment and discuss the results of our Internet experimentation .**

Keywords: Adaptive streaming; multipath; HTTP, Internet.

1. INTRODUCTION

Since the advent of HTTP based adaptive streaming technologies the popularity of over-the-top streaming services has kept increasing. Today there is a clear trend towards the delivery of HD/3D video content to the home in an over-the-top configuration. HTTP based adaptive streaming is a convenient end-to-end solution that mitigates network congestions as it leverages existing servers/infrastructure and keeps running even in a tough network environment.

However an outstanding problem with current solutions is that they are bound to adjust to current network capacity, resulting in uneven quality of experience, which is hardly acceptable for high value, high quality video (e.g. HD) playback.

A possible mitigation would be to increase the number of content delivery servers in a CDN (Content Delivery Network) infrastructure and to have one server close by each user. As underlined by Leighton [5], this approach is scalable but expensive, complex regarding management and still does not fully guarantee the consistency of the service since network problems are still possible. Yet another solution is to increase the buffering in the receiver. The receiver buffer is used to mitigate the network jitter wherein the size of the buffer varies function of the video quality and the network dynamics. In some situations however, buffering is not practical as with live content and may be of limited use if network degradations last for too long.

The solution exploited in this paper is to operate several communication paths concurrently with one or more servers thereby providing path diversity for fighting network congestion and smoothing quality variations with no bandwidth overhead.

We have recently published a solution [1], in which we combine distributed streaming, as introduced by Nguyen and Zhakor [12], with HTTP based adaptive bit rate streaming. This paper details a real Internet experiment in order to confirm the laboratory results of [1]. In addition, we show that the solution is applicable to different multipath schemes. Distributed streaming provides path diversity thanks to the concurrent usage of multiple sources. Another multipath scheme is to provide multi-homing on the client side (the latter is connected to at least two different access networks).

In Section 2 we introduce our HTTP based multipath adaptive streaming framework. In Section 3 we present our network emulation test bed set-up and outline the associated results disclosed in [1]. In Section 4 we discuss the Internet experiments. In Section 5, we review the related works and finally conclude in Section 6.

2. MULTIPATH HTTP STREAMING

In this section we present our implementation of the multipath HTTP streaming client.

2.1 Multipath HTTP Streaming Client Implementation

We have based our multipath adaptive client on a simple adaptive streaming implementation; therefore it shares many concepts with typical HTTP adaptive streaming solutions as listed in Section 5.

First of all the multimedia content is pre-encoded in different bit rates (with a constant frame rate and resolution) with aligned RAPs (random access point), resulting in easily switchable, different quality streams which we will call representations. Each representation is split into segments of the same duration (e.g. 2 seconds) as shown in Figure 1.

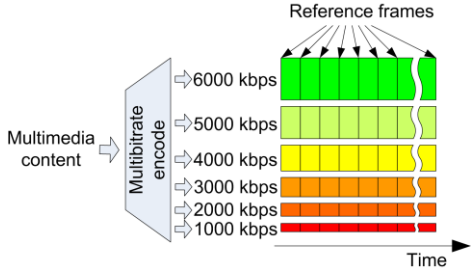


Figure 1 Content preparation

Corresponding author: S. Gouache, Technicolor, Rennes, +33(0)2 99 27 31 22, stephane.gouache@technicolor.com

The segments are provisioned to one or several conventional HTTP servers as depicted in Figure 2. The software agent in the user terminal (mobile phone, set-top box or PC) starts the session with one known server by downloading the service description that contains the number of segments, duration of a segment, number of representations and the addresses of all contributing servers.

Then the actual adaptive streaming session starts. The software agent continuously measures the bandwidth with each path using a round trip time evaluation of HTTP requests/responses. A smoothed version of this bandwidth measurement - discussed in the following section – is used by the software agent to decide from which representation to request the next segment.

The principal specificity of multipath adaptive streaming is that the software agent requests concurrently through each available path a slice of the next segment using the HTTP range header. The size of the slice requested through each path is proportional to the estimated bandwidth for the corresponding path.

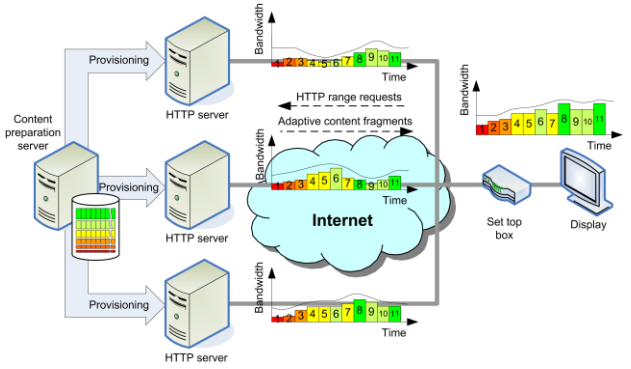


Figure 2 Concurrent multipath operation

2.2 Alternative multipath implementation scheme

In the previous section, we have considered the case of "distributed streaming" wherein the multipath capability is provided through the concurrent usage of multiple sources. Another approach is to operate a multi-homed client or server. We operate exactly the same framework except that only one server is used. In Figure 3, the client (the set-top-box) is connected to Internet through two different access networks.

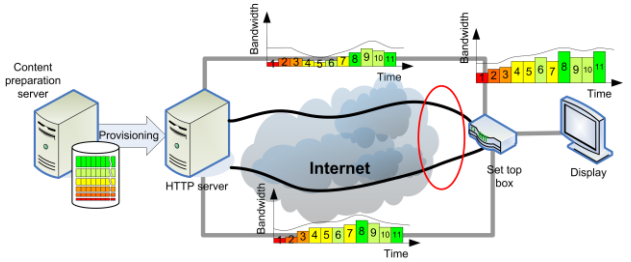


Figure 3 Multipath operating multihomed client

The advantage of the previous scheme in the context of CDN (Content Delivery Network) is its inherent load balancing capacity precluding over provisioning. The path diversity mitigates mostly the server side performance issues (ISP network bottleneck and/or server overload). With multi-homing supported by the terminal, the path diversity mitigates mostly access network bottlenecks. In both cases, the solution compensates also possible Internet bottlenecks (mostly due to ISP network interconnection capacities) as shown in Section 4.

2.3 Algorithms

2.3.1 Bandwidth estimation

We have implemented a bandwidth estimation algorithm inspired by the smooth RTT computation in TCP which is used identically for one path streaming or multiple paths streaming and whatever the implementation scheme. The algorithm used in our implementation is iterative and attempts to infer the achievable bit rate from the bit rate measured during the previous iterations. Each iteration encompasses all the operation necessary to compute the bandwidth estimation and retrieve the next segment through the available paths. An iteration lasts the time corresponding to the segment duration (2 seconds in our set-up).

The bandwidth estimation algorithm is detailed in [1]. The algorithm can be tuned according to a few parameters. We can control how fast the average bit rate estimate varies, how fast the past variations of the measured bit rate are forgotten and the weight given to these past variations.

With the chosen values for our experiment, the estimation algorithm tends to use the link close to its maximum capacity and quickly recover from network errors. This is deliberate since we intend to stream high bit rate content such as 1080p or 3D video. We do not buffer the received segments to favor reactivity for live streaming or yet channel change and trick play operation. Therefore for single path operation the selected representation for the current iteration is the representation having the closest average bit rate lower or equal to the current bit rate estimate.

For multipath adaptive streaming, since we consider the paths independent and retrieve the segment in parallel from all the available paths, the estimate used to select the next segment is simply the sum of the estimates for the individual paths.

2.3.2 Repair algorithm

Additionally, to cope with the case of path/server failure to deliver its slice on time, the software agent is capable of redistributing the retrieval of the corresponding slice to all paths having already delivered their own slice and having time to perform a further transaction and so within the same iteration. For each iteration and for each path, the software agent measures the time elapsed since the sending of the slice request and before getting the response. If this time goes beyond a certain threshold

Corresponding author: S. Gouache, Technicolor, Rennes, +33(0)2 99 27 31 22, stephane.gouache@technicolor.com

(e.g. segment duration/2) the software client performs the following actions:

- Stops processing the ongoing slice request
- Selects a subset of the fastest paths whose slice retrieval is already finished
- Splits the slice to be retrieved in sub-slices
- Submits the corresponding requests over the selected paths.

The efficiency of such repair algorithm depends to the number of paths and the segment duration. More paths offer more opportunities to recover from a missing slice. The larger the segment duration, the larger is the time to recover missing slices. However, for a given segment duration, extending the number of paths beyond an optimum proves to be counterproductive. In our implementation, with a segment duration of 2 seconds, we have established empirically the limit to 6 paths. Understanding the impact of segment duration onto the algorithms is a subject for further study.

3. EXPERIMENTAL SETUP

This section describes the setup used to measure the effects of multipath streaming under various conditions and the results collected from each experiment.

3.1 Controlling Path Characteristics

To measure the effects of multiple paths we have used a network emulator to control the path characteristics. Since we envisage one end of our network to be a residential broadband connection we limit the downlink rate to 8Mbit/s (Figure 4) with the assumption of no loss. We individually control the delay, packet loss rate and burst length on each path. Unless stated otherwise, we fixed the delay of the links to 15ms.

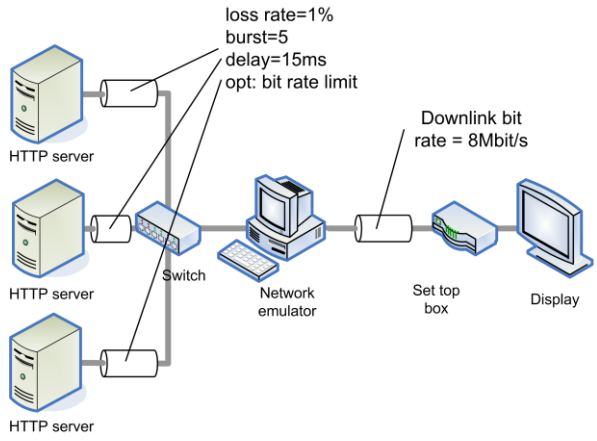


Figure 4 Experimental setup

The network emulator is a Linux box running a 2.6.31-22 kernel with a NetemCLG [4] 1.1 patch applied and iproute2-2.6.35 used to configure the traffic parameters. In all experiments the employed loss model is loss_GI which allows controlling of the packet loss rate and the average length of packet loss bursts. Set top box and servers are Linux boxes running a 2.6.31-22 kernel with its default TCP CUBIC variant.

3.2 Result Interpretation

The traces collected during each adaptive streaming session are collected and presented in a synthetic form. The raw data collected during a session include, for each iteration (500 iterations per session), the bit rate measured as in the above described algorithm, the derived estimated bit rate and the bit rate of the video segment retrieved and played by the client. An example of such a raw data collected for a single path session is shown in Figure 5.

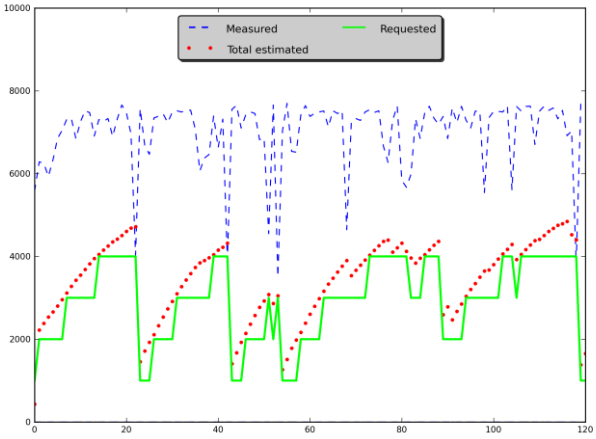


Figure 5 Single path example plot

The dashed line at the top of the figure represents the measured bit rate, exhibiting quick variations typically due to burst errors. The dotted line in the center of the figure is the estimate computed from the measured bit rate. The solid line at the bottom is the encoding bit rate of the segments requested and played by the client. The vertical axis is in kbps. The horizontal axis is the elapsed time in seconds.

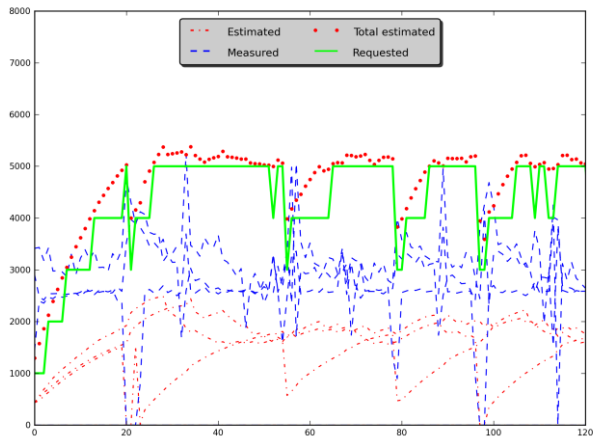


Figure 6 Multipath example plot

In the case of multipath operation with three servers as shown in Figure 6, the individual measured bit rates appear as dashed lines at the center of the plot. The corresponding estimates are visible as dotted lines at the

Corresponding author: S. Gouache, Technicolor, Rennes, +33(0)2 99 27 31 22, stephane.gouache@technicolor.com

bottom. The aggregate estimate is represented as a dotted line at the top of the plot. Finally the play out bit rate is the solid line immediately below the aggregate estimate.

We see from Figure 5 that our bandwidth estimation algorithm mimics TCP behavior with an evident aggressiveness. Indeed the estimation rises quite quickly and drops severely when a network failure occurs. The average play out (i.e. requested) bit rate seems to be around 3Mbit/s compared to 4Mbit/s in Figure 6. The play out bit rate exhibits large variations between 1 and 4 Mbit/s in Figure 5 whereas it is steadier in Figure 6 resulting in different user experiences. There have been numerous studies about time varying quality effect on user perception. Hamberg and De Ridder [6] have shown that users rely mainly on the worst quality when judging the overall streaming session experience. M. Pinson, S. Wolf [7] discussed the memory effect in user experience encouraging steady quality with as few quality variations as possible and progressive decrease (versus sharp drop off). There is some evidence looking at the playout bit rate exhibited in Figure 5 and Figure 6 that the user experience is improved in this first introductive experiment.

Apart from that the comparison between the aforementioned plots is difficult. Therefore since we are mainly interested in the play out bit rate we introduce another representation of the same results in the form of a bar chart. The height of the bars shows the number of iterations of each content play-out bit rate expressed as a percentage of the total number of iterations.

In addition, since our algorithm is iteration based, we collect the number of iterations when the requested segment was delivered late, thus implying the need for a reception buffer to avoid playback interruptions.

Figure 7 is the bar chart corresponding to the single path experiment previously shown on Figure 5. As can be seen on this chart, there is a majority of segments at 4Mbit/s. But it is worth noting that 30% of the received segments are between 1 and 2Mbit/s, resulting in poor playback quality 30% of the time. Furthermore, the amount of late segments is about 4% with an average lateness of 3 seconds and a maximum value of 9 seconds thus requiring a rather large reception buffer.

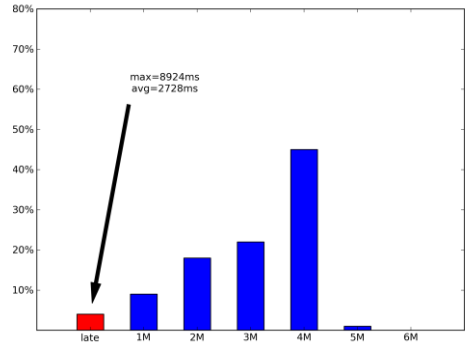


Figure 7 Single path example histogram

3.3 General results

We have performed several measurements with the above described setup by varying the number of servers and the paths' characteristics.

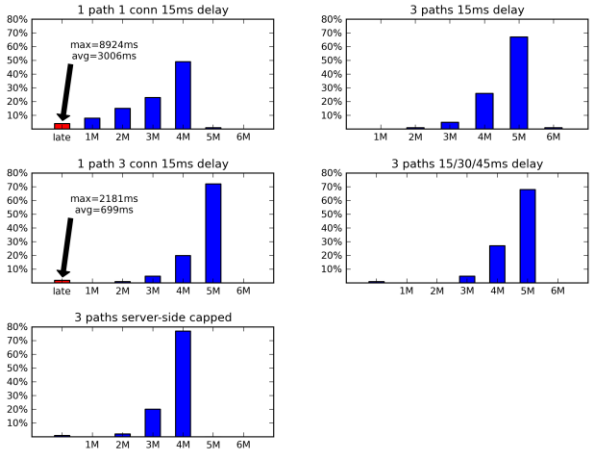


Figure 8 General results

Summarizing the findings of [1], as shown in the top row charts, using 3 servers concurrently significantly improves the average playout bit rate and reduces the likelihood of sharp drop-offs. The second row's left hand side chart shows that multiple connections over the same path provide a kind of resiliency to light congestions but suffer from the correlation of failures, rendering slice recovery nearly impossible. The second row's right hand side chart shows the relatively good performance of our solution with paths having unequal delays. Finally the bottom chart shows the impact of limiting the servers' output bit rate wherein the maximum achievable throughput is reduced but the quality variations damping effect is still clearly visible.

4. INTERNET EXPERIMENTS

For this set of experiments, we have used 3 typical HTTP hosting services located in France offered by Internet service providers. We have uploaded, into the 3 servers the 2 seconds video segments corresponding to a video sequence encoded in different qualities ranging from 200kbit/s to 3Mbit/s (200, 400, 600, 800, 1000, 1200, 1600, 2000, 2500, 3000kbits).

The first experiment shown on Figure 9 aims at validating the results found with our emulation platform by connecting through a single access network to a variable number of servers. As shown in the leftmost chart of, with a single path towards the fastest server, most segments are received at the highest bit rate close to the path's capacity (2Mbit/s) but a couple of times the bit rate drops significantly. With 3 connections towards the same server (center chart), the playout bit rate keeps oscillating between two values, probably because we are amplifying the congestions. With 3 paths towards different servers, the received bit rate almost never drops from the maximum value.

Corresponding author: S. Gouache, Technicolor, Rennes, +33(0)2 99 27 31 22, stephane.gouache@technicolor.com

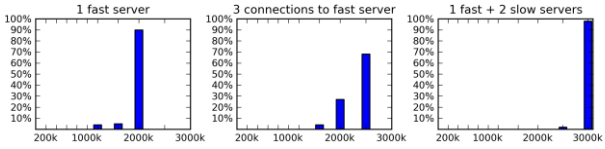


Figure 9 Single access to servers on the Internet

In a second experiment, we have shown the effect of our multipath adaptive streaming solution with a single server and a multihomed client using two different Internet access links simultaneously as presented in section 2.2. In the following, access #1 is using Orange ADSL access (8Mbps), access #2 is using SFR ADSL access (8Mbps) and access #3 is using Orange 3G access (3.6Mbps). The results when accessing an HTTP server hosted by the ISP SFR are depicted on Figure 10.

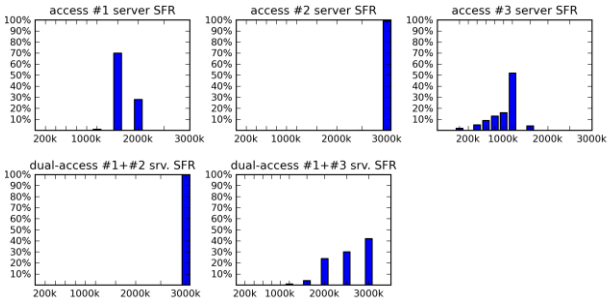


Figure 10 Single and dual-access to SFR server

The top row charts present the individual performance of each access using a single connection towards the server. As can be seen the performance is very different depending on the access. The access through #1 (Orange ADSL) results in an average playout bit rate of 1600kbps whereas access through #2 (SFR ADSL) achieves the maximum available bit rate. Access through #3 (Orange 3G) is quite good but exhibits relatively frequent variations. We can note that the dual access through #1 and #2, shown on the bottom left chart, cannot do better than #2 alone but has no negative effect. More interesting is the rightmost bottom chart wherein accesses #1 and #3 are combined which results in a summation of the throughput available from the individual accesses, significantly improving both the average and the minimum playout bit rate.

We repeated the experiment using a different server hosted by an independent ISP named Free as depicted on Figure 11.

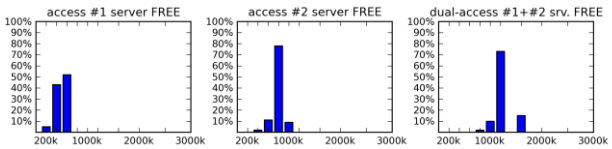


Figure 11 Single and dual-access to Free server

The two leftmost charts present the single-access performance and the right hand side chart presents the multipath performance achieved through the concurrent use of both accesses.

When using only one access network, whichever it is, the achieved bit rate is far from the plan and seems to be limited by the interconnection of the ISPs. Clearly for this specific server the access through SFR seems to be better than through Orange. Moreover, using the Orange access results in the playout bit rate changing frequently, resulting in an uneven quality.

Finally, combining both accesses allows us to reach an average bit rate close to the sum of the individual bit rates achievable independently by the separate accesses, which confirms the result of the experiment with the SFR server corresponding to the rightmost bottom chart of Figure 10.

By analyzing the output of the traceroute command we have observed that

- Going to different servers through the same access network uses different exchange (interconnection) nodes that explains the throughput difference
- A similar performance difference is exhibited when using two different access networks towards the same server
- Logically the achieved throughput is excellent when the server is hosted by the ISP providing the access

These observations confirm the usefulness of our multipath streaming solution for various multipath schemes.

5. RELATED WORK

Adaptive streaming techniques have recently emerged in the Media Industry. They are all HTTP based and rely on content aware delivery where the quality of the video varies function of the available bandwidth as tackled by Färber et al. [11] based on multi bit rate coding and by Feng work [10] based on scalable video coding. The benefit of using the HTTP protocol in these adaptive streaming solutions is its capability to cross over NAT routers and firewalls seamlessly. Recently standard bodies as 3GPP intensified their effort regarding HTTP based adaptive streaming. The first 3GPP specification on adaptive streaming provided the possibility to retrieve segments from different servers. This is however a rather static configuration as the manifest/configuration file associates an URL with a segment, precluding the concurrent segment retrieval of our implementation. In the latest release [9] the manifest format has been extended to permit multiple alternative URLs for each segment, thus becoming compatible with our solution.

The limited uplink bandwidth that is typical with unbalanced access network (e.g. ADSL) is a well known drawback for P2P. Distributed streaming (i.e. multisource concurrent streaming) complements logically the P2P technology as observed by Nguyen and Zhakhor [12] who have introduced the notion of distributed streaming as operated in this paper. Their framework is based on a receiver driven protocol using FEC (Forward Error Correction) and UDP delivery but with rate control and delivery synchronization driven by the receiver.

Corresponding author: S. Gouache, Technicolor, Rennes, +33(0)2 99 27 31 22, stephane.gouache@technicolor.com

Abdouni et al. [13] implement distributed streaming and discuss the most efficient way to distribute the load among the multiple paths based on their characterization including detection of shared segments. The assumption however was again based on the usage of UDP like traffic associated with erasure codes.

Using conditions similar to our experiments, Wang et al.[15] implemented multipath streaming over TCP. They have shown that the ratio between the aggregate throughput and the video bit rate is better with two path streaming than with only one. In their work, adaptive streaming was not used. Our results tend to confirm their findings especially regarding unbalanced delay paths and the aggregate throughput/video bit rate ratio.

A lot of work has been done managing multiple paths at the transport layer based e.g. on concurrent SCTP [17] even MPTCP [18]. Our work is based on distributed streaming wherein we operate multiple senders/servers emulating a multipath infrastructure. One possible investigation would be to examine how HTTP adaptive streaming performs over a transport layer based multi path protocol such as concurrent MPTCP.

The effect of quality variation on user perception has been addressed by numerous studies [6], [7]. As shown by Cranley et al. [8] with their Optimal Adaptation Trajectory (OAT), the user perceived quality is maximized when adapting the encoding parameters to the content type. Although this was not relevant in this experiment based on bit rate optimization, these important results will complement our further work.

6. CONCLUSION AND FURTHER WORK

The use of multipath communication has been shown to improve adaptive streaming. A simple implementation with as few as two available paths increases the average received bit rate and reduces the variance thereby dramatically improving the user experience. We have shown the applicability of multipath adaptive streaming to various scenarios through real Internet experimentations that confirm the results obtained in our emulation testbed. In addition we have shown the applicability of our solution to the case where multipath is provided through multihoming.

Further work is envisaged based on the variation of the segment duration, GOP structure and encoding parameters (e.g. resolution).

Acknowledgement

This work was partly funded by the European Union, through its 7th Framework Programme for Research (FP7), under grant agreement 258378 - FIGARO project.

References

[1] S. Gouache et al. Distributed & Adaptive HTTP Streaming. IEEE ICME 2011

[2] E. Nygren, R. K. Sitaraman, and J. Sun, "The Akamai network: a platform for high-performance internet applications," ACM SIGOPS, Operating Systems Review, vol. 44, no. 3, pp. 2–19, 2010.

[3] netem - Network Emulation http://www.linuxfoundation.org/collaborate/workgroups/networking/netem

[4] NetemCLG (Correlated Loss Generator) http://netgroup.uniroma2.it/twiki/bin/view.cgi/Main/NetemCLG

[5] T. Leighton. Improving Performance on the Internet. *Communication of the ACMs* February 2009, vol 52, No2

[6] R. Hamberg and H. De Ridder. Time-varying image quality : Modeling the relation between instantaneous and overall quality. *SMPTE Journal*. 1999, vol. 108, no11,

[7] M. Pinson, S. Wolf. Comparing subjective video quality testing methodologies. *SPIE Video Communications and Image Processing*. Conference, Lugano, Switzerland, July 2003.

[8] N. Cranley, P. Perry, L. Murphy. User perception of adapting video quality. *International Journal of Human-Computer Studies*.Volume 64 Issue 8, August 2006

[9] Group Services and System Aspects Transparent end-to-end Packet-switched Streaming Service (PSS); Progressive Download and Dynamic Adaptive Streaming over HTTP (3GP-DASH) (Release 10) - 3rd Generation Partnership Project; *Technical Specification* 3GPP TS 26.247 V1.0.0 (2010-09)

[10] W. Feng. On the efficacy of quality, frame rate, and buffer management for video streaming across best-effort networks. *Journal of High Speed Network*, volume 11, Numbers 3-4/2002

[11] N. Färber, S. Döhla and J. Issing. Adaptive progressive download based on the MPEG-4 file format. *Journal of Zhejiang University SCIENCE A* 2006 (suppl. 1): 106:111

[12] T. PQ Nguyen and A. Zakhor. Multiple Sender Distributed Video Streaming. *IEEE Transactions on Multimedia*, Vol. 6, No. 2, April 2004

[13] Abdouni et al. Multipath Streaming: Optimization and Evaluation. *Multimedia computing and networking*. MMCN 2005.

[14] Abib, J. Chuang. MMS: A Multihome-aware Media Streaming System. *Proceedings* of SPIE, 2006

[15] B.Wang, W. Weiy, Z. Guo, D. Towsley. Multipath Live Streaming via TCP: Performance and Benefits. *Proceedings of the 2007 ACM CoNEXT conference*,

[16] L. Ma and W.T. Ooi, Congestion Control in Distributed Media Streaming. *26th IEEE International Conference on Computer Communications*. IEEE INFOCOM 2007.

[17] J.R. Iyengar, P.D. Amer; R. Stewart. Concurrent Multipath Transfer Using SCTP Multihoming Over Independent End-to-End Paths. *IEEE/ACM Transactions on Networking*. October 2006 Volume 15, Issue 5.

[18] Sébastien Barré, Christoph Paasch and Olivier Bonaventure. MultiPath TCP: From Theory to Practice, IFIP Networking, 2011

Corresponding author: S. Gouache, Technicolor, Rennes, +33(0)2 99 27 31 22, stephane.gouache@technicolor.com

Video-Aware Rate Adaptation for MIMO WLANs

An Jack Chan*
University of California, Davis, USA
anch@ucdavis.edu

Henrik Lundgren
Technicolor, Paris, France
henrik.lundgren@technicolor.com

Theodoros Salonidis
Technicolor, Paris, France
theodoros.salonidis@technicolor.com

***Abstract*—The IEEE 802.11n standard supports very high physical layer data rates using Multiple Input Multiple Output (MIMO) antenna technologies. Despite such high rates, High Definition (HD) video streaming is still challenging in WLAN deployments. In this paper, we show that the wireless channel probing overhead of existing 802.11n data rate adaptation mechanisms can be detrimental to HD video performance. We propose VARA, a Video-Aware Rate Adaptation protocol that addresses this problem by adapting the frequency and timing of wireless probing to both video encoding rate variations and wireless channel variations. In addition, VARA employs novel strategies that multiplex several Variable Bit Rate (VBR) HD video streams by minimizing their aggregate peak rate requirement. Our experimental evaluations for static and mobile scenarios in a MIMO 802.11n wireless testbed demonstrate the practical benefits of VARA over state-of-the-art 802.11n rate adaptation protocols.**

I. Introduction

Today, video dominates Internet traffic and by 2014 it is projected that nearly half of the video traffic will be 3D or 2D High Definition (HD) [1]. A high fraction of HD video traffic will be consumed by users that access Wireless Local area Networks (WLANs) in homes, enterprises or public spaces. This vision is fueled by two major technology trends. First, recent video streaming technology standards such as H.264/MPEG-4 part 10 AVC [2] reduce HD video bandwidth requirements using Variable Bit Rate (VBR) video encoding. Second, the IEEE 802.11n [3] WLAN standard offers very high wireless physical-layer (PHY) data rates (up to 600 Mbps) using Multiple-Input Multiple-Output (MIMO) antenna technologies.

Despite these advances, the problem of streaming HD video in WLANs is far from being solved. VBR technologies reduce the average video streaming rate by efficient encoding of slow and moderate moving scenes. However, the peak rate remains high as it is determined by the full quality encoding of fast-motion scenes. Also, according to recent studies, the achieved goodput in practical 802.11n deployments can be significantly below the maximum 802.11 PHY data rates [4], [5], due to lack of favorable channel conditions for high MIMO PHY rates, MAC protocol overhead, sub-optimal PHY data rate selection, interference, or even backward compatibility with the previous lower-rate 802.11 standards.

In this paper we show that, in addition and irrespective of the above issues, 802.11n suffers from a fundamental problem that causes video quality degradation. This problem arises due to the probing overhead of the existing 802.11n PHY rate adaptation protocols. These protocols periodically probe different PHY rates to discover the maximum PHY rate supported by the wireless channel. In order to reduce probing overhead, most of these protocols perform *implicit probing* with packets of the ongoing data traffic instead of separate probe packets. Probe packets are lost when transmitted at PHY rates that cannot be supported by the current wireless channel state. Although such probe losses may not be perceptible in delay tolerant data applications, we show that they result in delays or losses of video frames at the application layer, which are both detrimental to HD video streaming performance.

We introduce VARA, a Video-Aware Rate Adaptation protocol, that optimizes wireless channel probing and PHY rate selection by exploiting the VBR streaming rate information of a video. VARA eliminates the channel probing impact on the video stream by scheduling the probes during the low-streaming-rate periods. Furthermore, rather than aggressively trying to find the maximum PHY rate supported by the wireless channel (like all existing 802.11 rate adaptation protocols), VARA selects the most reliable PHY rate that supports the near-future peak streaming rate. To further reduce the probing overhead, VARA monitors the Frame Error Rate (FER) and adapts probing frequency to the measured wireless channel variability.

We then introduce three multiplexing strategies that assist VARA in supporting multiple HD video streams. These strategies inspect and carefully multiplex the different VBR variations of streaming rates in multiple videos to minimize aggregated peak rate, outage time or outage area subject to a capacity target.

Finally, we experimentally evaluate the performance of VARA and the multiplexing strategies using a user-space implementation in a wireless testbed equipped with off-the-shelf 802.11n cards. Our experiments show that VARA reduces the loss rate of higher-rate HD video streams to only a few percent and that this loss reduction translates to 50% video quality increase in terms of Peak Signal-to-Noise Ratio (PSNR). For lower-rate HD video streams, VARA reduces the loss rate to zero and achieves perfect PSNR. During multiple simultaneous video streaming sessions, the multiplexing techniques reduce the aggregated peak rate by up to 25%, which in turn assists VARA to reduce the average loss rate by 20% to 50%.

In summary, our contributions are as follows:

1) We identify the detrimental effect of probing overhead of legacy PHY rate adaptation mechanisms on video

*This work was undertaken while the first author was interning at Technicolor Research Laboratory, in Paris, France.

streaming quality.

2) We introduce VARA, a protocol that adapts PHY rate based on video streaming rate and channel quality.
3) We introduce three multiplexing strategies that assist VARA to efficiently support multiple video streams.

To the best of our knowledge, VARA is the first video-aware wireless rate adaptation protocol for 802.11n MIMO WLANs. In addition to its performance benefits, it is a practical solution that does not require modification of the 802.11 standard. We believe that this approach demonstrates the feasibility and potential of practical cross-layer content-aware techniques in MIMO WLANs and can also be applied to other types of wireless networks.

The rest of the paper is organized as follows. Section II illustrates, explains and quantifies the problem using a practical example in our wireless testbed. In Section III and Section IV, we present the design of VARA and the multiplexing techniques, respectively. In Section V, we experimentally evaluate VARA and the multiplexing strategies in our testbed. Section VII discusses related work and Section VIII concludes.

II. Motivation

In this section we first show an illustrative example where a standard 802.11n rate adaptation protocol results in video streaming quality degradation. We then explain the underlying reasons of the observed problem and finally perform a micro-benchmark to experimentally quantify its impact.

A. Video Streaming Example

We set up a stable 2x3 MIMO 802.11n link in our testbed and measure its maximum capacity[1] over all PHY data rates as 28Mbps. We then stream a video with 26Mbps peak rate over this link with the default automatic rate adaptation protocol (auto rate) option turned on.

Figure 1 shows the video streaming rate and the PHY data rate at the sender and the goodput mesaured at the receiver, during the entire video streaming period. We observe that auto rate most often uses a PHY rate of 52Mbps and periodically probes at 78Mbps. During the peak streaming rate period between 12s and 15s, the goodput drops below the video streaming rate. This indicates that some video packets are lost. Indeed, Figure 2(a) shows that the video packet loss experienced during this period is up to 14% and the video snapshot in Figure 2(b) shows that this loss results in low perceptual video quality.

As an extra validation, we repeat the above experiment with the PHY rate fixed at 52 Mbps, the most frequently selected PHY rate by auto rate during the previous experiment. In this test run with a fixed rate, the video streaming is perfectly supported without any packet loss or quality degradation. This example illustrates that the 802.11n rate adaptation induces a penalty that can significantly degrade the video streaming quality. We proceed to explain the origin of this penalty.

[1]We define as "capacity" of a link at a PHY rate (or auto rate), the maximum (backlogged) UDP goodput measured on this link when this PHY rate (or auto rate) is used.

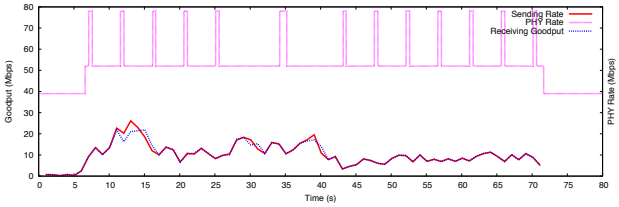
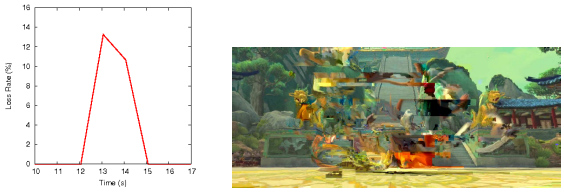


Fig. 1. Video streaming rate, goodput and PHY rate.



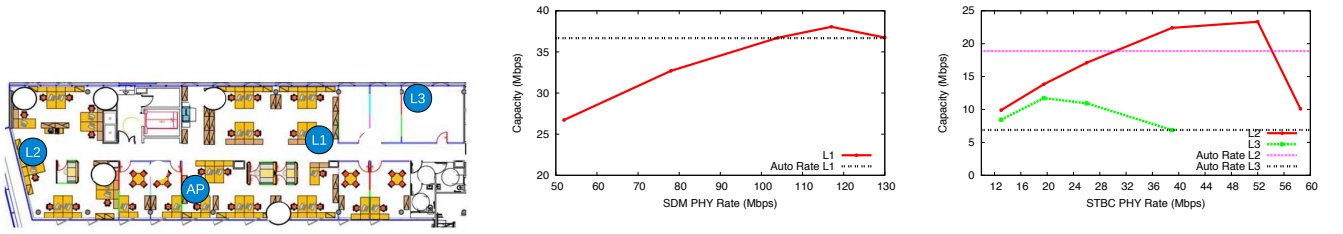
(a) Packet loss (b) Video frame snapshot

Fig. 2. Peak period packet loss and video quality degradation.

B. Probing-induced Capacity Penalty

In order to select an appropriate PHY data rate, 802.11 rate adaptation mechanisms need to probe the wireless channel. Probing can be either explicit or implicit. Explicit probing probes the channel using separate control packets; implicit probing probes the channel using packets from the ongoing data traffic. Most existing ("legacy") rate adaptation protocols use implicit probing due to its reduced overhead. During implicit probing, a few data packets are periodically sent at a higher PHY rate than the current PHY rate. If these transmissions succeed, the channel condition is estimated as adequate to support this higher PHY rate. Then, an even higher PHY rate is used to send the next few data packets. This process continues until a PHY rate where most probe transmissions fail. Then the previous (last successful) PHY rate is selected as the new rate, until the next probing period. The start of the next probing period is triggered by various factors, such as number of successful transmissions, measured packet error rate, time lags and so on, according to different implementations [6].

Increased probing frequency enables faster rate adjustments to channel changes (and thus potentially higher goodput), but also increases the overhead due to failed probe packets. These probing-induced losses result in channel capacity degradation due to binary exponential back-off and airtime consumption for retransmissions. We call it the penalty in capacity utilization. It is important to note that probing and probe losses occur regardless of whether the current channel condition is good or bad. Unless the channel is good enough to support the highest PHY rate all the time, which is rare due to the dynamic property of wireless channels, the rate adaptation will periodically probe the channel to find a better rate to maximize the capacity. A variable channel naturally aggravates the probing overhead as it will cause the probing process to be triggered more frequently and result in increased probing-induced losses.



(a) Access point and node locations (b) Measured capacities at location L1 (c) Measured capacities at locations L2 and L3

Fig. 3. Rate adaptation micro-benchmark testbed and results.

C. Rate Adaptation Micro-benchmark

In order to quantify the probing-induced capacity penalty, we perform a micro-benchmark where we measure the capacity of three different 802.11n links with different fixed PHY rates and auto rate in our testbed, shown in Figure 3(a).

Figure 3(a) shows the location of the access point (AP) and three clients at locations, L1, L2 and L3. For the coherence of our discussion of micro-benchmark, we defer the details of the experimental settings to Section V-A. We use a CBR traffic generator to measure capacities at different fixed PHY rates and auto rate at all three locations. Figures 3(b) and 3(c) show the results. Each data point is the average of 10 4-minute test runs. Instead of a single data point, each auto rate capacity is drawn as a horizontal line for comparison to the fixed PHY rate capacities at the same location.

Figure 3(b) depicts the channel capacities at location L1. At this location, the channel can support the high rate Spatial Division Multiplexing (SDM) mode of MIMO 802.11n where different data streams are sent over different antennas [3]. In this case, higher PHY rates can be used and auto rate achieves close to the maximum capacity of this link. In this case, probing has limited capacity penalty.

Figure 3(c) shows the channel capacities at locations L2 and L3. At these locations the channel conditions deteriorate and SDM mode yields very low or zero goodput. These locations only support the use of the Space Time Block Coding (STBC) mode of MIMO 802.11n, which aims at increasing the robustness by sending copies of a single data stream over different antennas [3]. Figure 3(c) shows that auto rate can only achieve 70%-80% of the maximum capacity. For example, at L2, although auto rate selects PHY rate 52Mbps most of the time, it achieves 75% of the capacity achieved by a fixed PHY rate of 52Mbps.

This micro-benchmark shows that the probing-induced capacity penalty in our testbed ranges between 5% and 25% (and 20% on the average), depending on the channel conditions. Note that the legacy auto rate adaptation mechanism of the 802.11n cards in our testbed employs implicit probing. If explicit probing is used, the probing overhead and capacity penalty would be higher due to the extra control packets required for probing.

We proceed to describe VARA, our video-aware rate adaptation protocol that aims at reducing the probing-induced capacity penalty and minimizing its impact on the video streaming quality by exploiting video streaming information.

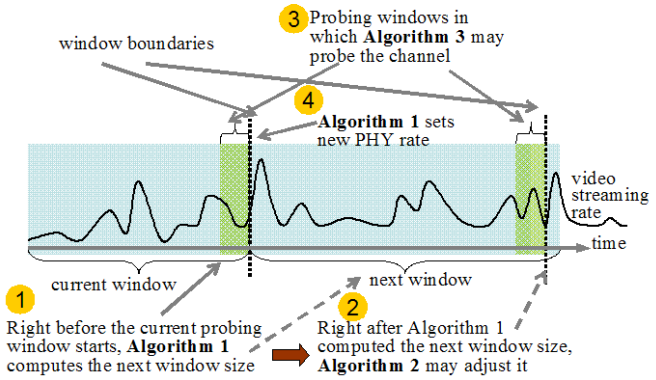


Fig. 4. VARA operation in a video streaming example.

III. VARA Design

VARA is a cross-layer, video-aware PHY rate adaptation protocol. Its basic idea is to use a video streaming rate waveform from the application layer to guide the adaptation of the wireless PHY rate. For any stored video in a wireless home network video server, such a waveform can be easily generated with a play back during video recording.

VARA divides time into variable-sized windows. For each window, VARA attempts to find a PHY rate that yields capacity above the peak video streaming rate in the window. VARA adapts the window sizes to take into account the wireless channel variability and probing overhead.

As shown in the example of Figure 4, VARA consists of three algorithms invoked before the end of each window. First, Algorithm 1 computes the size of the next window based on past measurements of channel variability taken during the current window. Then, Algorithm 2 refines this size to satisfy rate requirements of the probing that Algorithm 3 might run during the next window. If the PHY rate of the current window cannot support the peak video streaming rate of the next window, Algorithm 3 probes the wireless channel for a suitable PHY rate. At the beginning of the next window, Algorithm 1 sets the PHY rate found during the previous steps.

In the next sub-sections, we describe in more detail the operations of these three algorithms.

A. Algorithm 1: window size and PHY rate adaptation

Algorithm 1 is the "master" algorithm that invokes algorithms 2 and 3. Its operation is depicted in Figure 5. Algorithm 1 is invoked α seconds before the end of the current window. The parameter α is a system-set parameter and is high enough to include the computations and probings described below.

Window size adaptation. Algorithm 1 computes the size of the next window based on the wireless channel variability of the current window. Let n_{total} be the total number of MAC frames (including MAC re-transmissions) transmitted during the current window. The channel variability is computed as the variance $var(\mathbf{L})$ of a set of N Frame Error Rate (FER) values, where $\mathbf{L} = \{l_1, ..., l_N\}$. The i-th FER value l_i, is the fraction of lost MAC frames within the i-th block of c transmitted MAC frames during the current window (i.e., $N = \lfloor n_{total}/c \rfloor$).

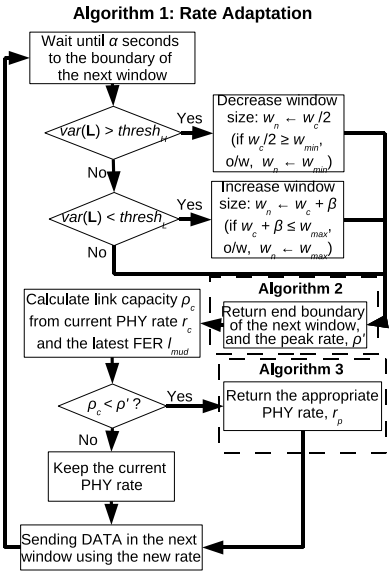


Fig. 5. Flow chart of Algorithm 1.

The size of the next window is determined from the current window size by comparing the variance $var(\mathbf{L})$ to a low threshold $thresh_L$ and a high threshold $thresh_H$. If the variance is lower than $thresh_L$, the current window is additively increased by β seconds; if it is greater than $thresh_H$, the current window is halved; if it is in-between the two thresholds, the current window remains fixed. The window size is bounded by a minimum size, w_{min}, and a maximum size, w_{max}. Multiplicative decrease helps VARA to react rapidly to high channel variability. Additive increase helps VARA slowly discover and lock the window size to a channel variability target.

Algorithm 1 calls Algorithm 2 (Section III-B) to refine and finalize the size of the next window based on the probing overhead requirements. Algorithm 2 returns the refined window boundary and the peak video streaming rate, ρ', of the next window.

Capacity estimation. Algorithm 1 estimates the channel capacity ρ_c of the next window based on the most recently measured FER value l_N and the PHY rate r_c of the current window. Recall that the channel capacity of a PHY rate is defined as the maximum UDP goodput that the PHY rate can achieve in a 802.11n link. It can be computed by the following formula [7]:

$$\rho_c = \frac{P}{t_{idle} + t_{tx}} \tag{1}$$

where P is the UDP payload size used by the video stream, t_{idle} and t_{tx} are the average idle and average transmission times, respectively. The average transmission time t_{tx} is given by:

$$t_{tx} = \frac{P + H}{(1 - l_N)^{ETX} T_{nom}} \tag{2}$$

where H is the UDP header size, and $ETX = 1/(1-l_N)$ is the expected number of link-layer retransmissions. Also, T_{nom} is the lossless capacity (FER=0), which can be measured offline or computed with analytical formulas [7].

The idle time t_{idle} is given by:

$$t_{idle} = \begin{cases} F(0, \lfloor ETX \rfloor - 1), & \text{if } ETX < m \\ F(0, m-1) + \sigma \frac{(\lfloor ETX \rfloor - m)(W_m - 1)}{2}, & \text{otherwise} \end{cases} \tag{3}$$

where σ is the 802.11 slot duration, $m+1$ is the total number of backoff stages. So, W_0 and W_m are the 802.11 backoff window sizes in stage 0 and in stage m respectively. $F(a, b)$ is the total average backoff time between backoff stages a and b. Assuming the node randomly chooses the backoff window size in a uniform distribution, $F(a, b)$ is the expected backoff window size multiplied by σ.

PHY rate adaptation. Algorithm 1 compares the calculated channel capacity, ρ_c, with the peak streaming rate, ρ', returned by Algorithm 2. If ρ_c exceeds ρ', the PHY rate r_c of the current window will be used in the next window. Otherwise, Algorithm 3 (Section III-C) is called to probe the channel and determine the appropriate PHY rate to use. Once the PHY rate of the next window is determined, Algorithm 1 sets it at the beginning of the next window.

B. Algorithm 2: Window Size Refinement

Algorithm 2 refines the size of the next window computed by Algorithm 1 to handle the probing overhead. Since probing only occurs near the end boundary of the window, the position of the end boundary should be carefully chosen to minimize the impact of probing on video streaming performance.

Let b_n be the end boundary of the next window computed by Algorithm 1. Based on the streaming rate before b_n, Algorithm 2 calculates a probing window size η that can support all probing packets. More specifically, η is computed to satisfy the following:

$$\int_{b_n - \eta}^{b_n} f(t)\, dt > n_p (P_{802.11})(|\, \mathbf{R} \,| - 1) \tag{4}$$

where $f(t)$ is the video streaming rate at time t, n_p is the number of ongoing data frames used for probing at each PHY rate, $P_{802.11}$ is the average WLAN frame size used for video

streaming, and $\mathbf{R}$ is the set of 802.11n PHY rates.[2]

The average video streaming rate ρ_a from time $b_n - \eta$ to b_n should also satisfy:

$$\rho_a < \rho_u \tag{5}$$

The maximum rate requirement ρ_u exists to ensure that the probing overhead will not cause the rate to exceed the peak video streaming rate ρ' of the next window. Such probing overhead causes capacity penalty. From Figures 3(b) and 3(c), the average capacity penalty in our testbed is 20%. Let h be the average capacity penalty caused by probing, then:

$$\rho_u = \frac{1}{1+h}\rho' \tag{6}$$

If (5) is *not* satisfied, Algorithm 2 moves the end boundary of the next window in steps of ξ seconds until it is satisfied. If the current window size was increased by Algorithm 1, Algorithm 2 moves the end boundary later; otherwise it moves it earlier. After each step of moving, b_n, ρ', ρ_u, ρ_a and η are updated accordingly. Once (5) is satisfied, Algorithm 2 returns the final end boundary b_n and the final peak rate ρ'.

As in Algorithm 1, the minimum window size, w_{min}, and the maximum window size, w_{max}, also apply to Algorithm 2. If those limits are exceeded and (5) has not yet been satisfied, Algorithm 2 returns the window size and boundary that was originally computed by Algorithm 1. However, instead of ρ' it returns a higher value $(1+h)\rho'$ for the peak rate. Then, Algorithm 1 will check if the current PHY rate satisfies this higher peak rate which includes the probing overhead. If not, it will call Algorithm 3 to search for a such PHY rate.

C. Algorithm 3: Channel Probing

Algorithm 3 probes the channel when the capacity ρ_c of the current window cannot support the peak streaming rate ρ' of the next window.

During operation, an IEEE 802.11n system must select among 16 PHY rates, that include both SDM and STBC MIMO modes. Algorithm 3 reduces probing overhead by reducing the number of probed PHY rates. This is achieved by using the property that FER is an increasing function of PHY rate within each MIMO 802.11n mode (STBC or SDM) [5]. This in turn implies that, for either STBC or SDM mode, the capacity as a function of PHY rate has a single maximum as shown in Figures 3(b) and 3(c).

Algorithm 3 probes each of SDM and STBC modes separately as follows. First, it determines the probing direction by probing a PHY rate one step lower and a PHY rate one step higher than the current rate r_c. For each rate it uses implicit probing, i.e., it sends n_p consecutive data frames of the ongoing traffic at that PHY rate and measures the FER. Then it uses Equation (1) to compute the corresponding capacity. If both capacities are lower than the capacity ρ_c of r_c, Algorithm 3 returns the rate r_c because it yields maximum capacity for this mode. Otherwise, if the lower (higher) step rate gives higher capacity than ρ_c, Algorithm 3 continues probing all rates at lower (higher) steps one by one until it finds one with a capacity higher than the peak rate requirement ρ'. If no such rate is found, Algorithm 3 yields the PHY rate of maximum capacity.

Finally, Algorithm 3 compares the capacities of the two PHY rates found for SDM and STBC modes and returns to Algorithm 1 the PHY rate whose capacity exceeds ρ' and has a lower FER (i.e., it is more robust). If none of these two capacities exceed ρ' Algorithm 3 returns to Algorithm 1 the PHY rate of higher capacity between the two.

[2]802.11n supports 77 different PHY rates. Offline configurations such as channel width and guard interval reduce them to 16 during network operation [3]. Since the current PHY rate does not require probing, at most $|\mathbf{R}| - 1 = 15$ rates will be probed.

IV. Multiple Video Streams

When multiple video streams exist on a link, VARA treats their aggregated streaming rate as if from a single stream and the same algorithms for handling a single stream can be reused. However, an aggregate stream of multiple HD videos can have a very high peak rate. A higher peak rate makes it more difficult for VARA to find a matching PHY rate or it may exceed the wireless channel capacity.

VARA addresses this problem using a novel multiplexing technique that we call *Strategic Shifting*. The idea is to strategically shift the starting times of the video stream waveforms such that the peak rate of the aggregate stream is minimized. This technique is enhanced with two other shifting techniques that aim to minimize outage time or outage area with respect to a channel capacity target.

A. Strategic Shifting

Two videos. Suppose a new video session v_2 of duration D_2 is requested to start at time t_q of an ongoing video session v_1 of duration D_1. Strategic Shifting will delay the start of v_2 for δ seconds after t_q. The parameter δ cannot exceed a delay budget of Δ seconds, the highest start-up delay the user of the incoming video can tolerate. Given the two video waveforms $f_{v_1}(t)$ and $f_{v_2}(t)$, the peak aggregate streaming rate $\phi(t_q, \delta)$ when v_2 starts δ seconds after time t_q of v_1 is given by:

$$\phi(t_q,\delta) = \begin{cases} \max\limits_{0\le t\le t_{r_1}} [f_{v_1}(t+t_q+\delta) + f_{v_2}(t)], & t_{r_1} \le D_2 \\ \max\limits_{0\le t\le D_2} [f_{v_1}(t+t_q+\delta) + f_{v_2}(t)], & \text{otherwise} \end{cases} \tag{7}$$

where t_{r_1} is the remaining time $D_1 - (t_q + \delta)$ of video v_1.

Strategic Shifting finds an optimal δ' in $[0, \Delta]$, which yields the minimum aggregated peak rate $\phi(t_q, \delta)$.

$$\delta' = \arg \min_{0\le\delta\le\Delta} \phi(t_q,\delta) \tag{8}$$

Multiple videos. If there are more than two incoming videos, VARA randomly selects one of them to multiplex with the existing video using Strategic Shifting. Then a second video among the incoming videos is randomly selected and multiplexed with the current aggregate stream, and so on. Therefore, the incoming streams are multiplexed sequentially one by one. In this way, Strategic Shifting is repeatedly applied by treating the early admitted videos as a single aggregated stream and multiplexing one new video at a time.

B. Outage Minimized Shifting

Strategic Shifting minimizes peak video streaming rate, but there is still a risk that the channel capacity is exceeded during video streaming. *Outage Minimized (O-M) Shifting* seeks to enhance Strategic Shifting by minimizing the impact of outage when a channel capacity target ρ_c is exceeded. The capacity target parameter can be the average, maximum or most recent capacity measured on the link during the streaming of the existing videos.

If the aggregated peak rate ρ'' returned by Strategic Shifting exceeds ρ_c, O-M Shifting finds a new δ' to minimize *outage time* or *outage area*. The outage time $O_t(\delta)$ is the total time in which the aggregated streaming rate exceeded the channel capacity target ρ_c. The outage area, $O_a(\delta)$ is the area between the waveform of the aggregated streaming rate and the channel capacity target ρ_c. It represents the total number of bits that will be dropped if the target is exceeded and is defined as follows:

$$O_a(\delta) = \int_0^{\max(t_{r_1}, D_2)} [f_{v_1}(t + t_q + \delta) + f_{v_2}(t) - \rho_c]\, dt \quad (9)$$

The performance of both Strategic Shifting and the two versions of O-M Shifting are evaluated in the next section.

V. Evaluation

In this section, we experimentally evaluate VARA's performance using our MIMO 802.11n wireless testbed. We first show that compared to the default auto rate adaptation protocol, VARA significantly reduces packet loss and achieves perfect, or close-to-perfect, video quality in terms of PSNR. We then show that VARA can efficiently adjust windows and schedule probing for different window sizes and video streams. Finally, we show that our multiplexing strategies improve the support for multiple simultaneous video streams.

A. Experimental Settings

We deploy a wireless testbed (Figure 3(a)) in an indoor office environment with cubicles, meeting booths and regular offices. Each testbed node is a Linux PC with Intel Pentium M 1.73GHz processor and 512MB RAM, and is equipped with a Ralink RT2880 802.11n 2T3R MIMO mini-PCI card and three 5dBi omni-directional antennas. We use the RT2880 driver with RT2880iNIC Firmware version 2.0.0.1. The wireless cards are set to operate in channel 108 in the 5GHz frequency band and use their default settings: 20MHz channel width, a Short Guard Interval (SGI) 800ns, and block acknowledgement and frame aggregation features deactivated. The total number of PHY rates including both SDM and STBC modes is 16. i.e., $\mid \mathbf{R} \mid = 16$.

One testbed node acts as Access Point (AP) and the others as clients. The AP is at a fixed location while the clients are deployed at different locations with different wireless channel conditions. Comparative experiments are scheduled back-to-back and repeated. To increase experimental repeatability we carry out experiments during evenings and weekends. Also, we use sniffers to ensure that there are no external interferences or hidden terminals during the experiments.

TABLE I
HD movie clips' properties. Rates are in Mbps

Movie Name	Average rate	Peak rate	Variance
Panda1080p	10.26	26.12	28.71
Panda720p	5.96	15.94	10.82
MonsterAliens	5.06	14.64	5.22

B. VARA in Static Environment

We first compare VARA's performance against the default auto rate of the RT2880 Ralink cards (the legacy rate adaptation algorithm) in a static environment. Therefore, we place both the AP and the clients in fixed locations as shown in Figure 3(a). Locations L1, L2, and L3 are selected to yield different wireless channel qualities. L1 has the best, L3 the worst. Table I shows the properties of different HD movie clips used in this experiment. *Panda1080p* represents the high streaming rate video, while *Panda720p* and *MonsterAliens* represent the medium and low streaming rate videos, respectively.

Each experiment run consists of two back-to-back streamings of each HD video between the AP and each client location, first using VARA and then auto rate. We repeat each run five times and show the average.

At the best channel quality location L1 auto rate supports all videos perfectly with zero loss. However, at location L2 it cannot support *Panda1080p*, the highest rate video. In contrast, VARA supports all videos perfectly at both locations L1 and L2. In L3, auto rate cannot support any of the videos perfectly. In contrast, VARA supports all videos perfectly except *Panda1080p*. This is expected because *Panda1080p* has a higher peak rate (26.12Mbps, Table I) than the maximum capacity of L3 (12.5Mbps, Figure 3). As depicted in Figure 6(a), with *Panda1080p* VARA achieves a 2% burst loss lasting for two seconds during the peak rate period, while auto rate results in a burst loss of 35% lasting for six seconds during the peak rate period.

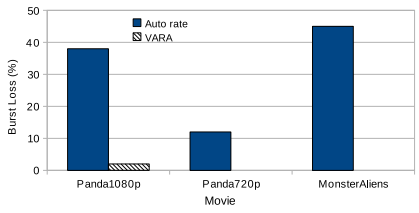
Figure 6(b) shows how these average burst loss rates translate to video quality as measured by average PSNR. We find that even a burst loss rate as small as 12%, as *Panda720p* suffers, can cause a significant degradation of video quality. In terms of subjective video quality, although a video with a PSNR of 25dB to 30dB could still be acceptable, it demonstrates obvious jitters, blocking and blurring. A video with a PSNR around 40dB is considered as a high quality video without any observable defect [8]. We observe that VARA achieves perfect PSNR[3] for two videos, and increases PSNR about 50% for the high rate video over auto rate.

C. VARA in a Mobile Environment

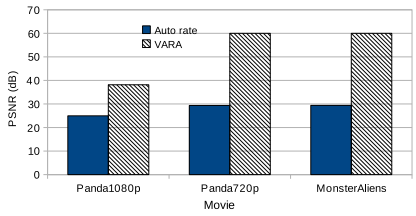
We now evaluate the performance of VARA when the channel condition changes more drastically. We use a controlled mobility scenario where a client moves along the path L4-L5-L6 at walking speed (see Figure 7).

We first measure the FERs and capacities at these locations by sending UDP packets in different PHY rates. Table II shows

[3]By definition, when there is no distortion, PSNR is infinity. For evaluation purposes we use a large number (60dB) to represent such perfect PSNR [9].



(a) Loss rate location L3.



(b) PSNR at location L3.

Fig. 6. Comparing auto-rate and VARA in terms of loss rate and PSNR.

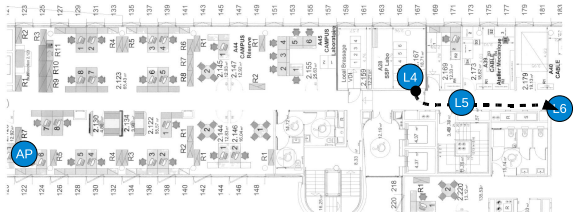


Fig. 7. Testbed mobility experiments.

the averages of these two quantities over five experiment runs[4]. As the client moves from L4 to L5 and then to L6, the FER

TABLE II
AVERAGE FER AND CHANNEL CAPACITY (MBPS) MEASUREMENTS AT LOCATIONS L4, L5 AND L6 FOR DIFFERENT PHY RATES. CAPACITIES ARE IN BOLD.

Data Rate	19.5	26	39	52
L4	-	-	0, **30.34**	98.5, **0**
L5	-	0.12, **19.24**	0.45, **10.27**	0.79, **0.02**
L6	0, **16.75**	0.49, **5.89**	0.59, **3.92**	-

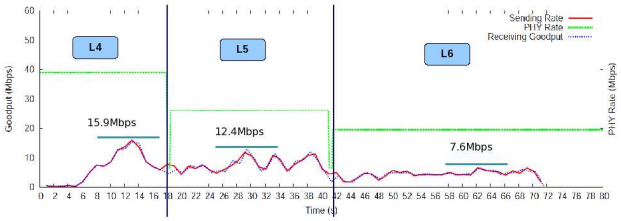
of a particular PHY rate changes. For example, when 39Mbps PHY rate is used and when the client moves to L4, L5 and to L6, the FER increases from 0, to 45% and to 59%, respectively. At the same time, the capacity decreases.

We then perform an experiment in which a client moves with the same mobility pattern using VARA and auto rate while the *Panda720p* HD movie clip is being streamed. Figure 8(a) shows the impact of VARA's rate adaptation on the goodput in this scenario. VARA computes different window sizes when the client is in different locations. Window size increases as the variation of FER is small when the current PHY rate is used. (It is worth noting that a large FER does not necessary mean a large FER variation.) When the end boundary of each window approaches, VARA evaluates if the current PHY data rate can provide the capacity large enough for the peak rate of the next window.

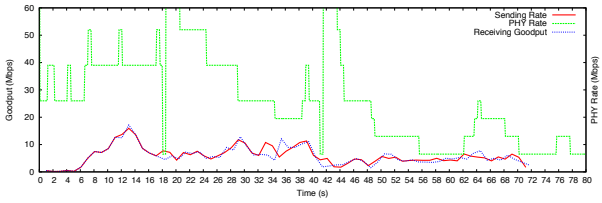
[4]The dashes exist because we only measured at PHY rates that VARA selected in subsequent video streaming experiment to save experiment time.

By comparing the peak rate requirements denoted in Figure 8(a) at the windows and the capacities in Table II, VARA selects the appropriate PHY rates. At L4, when the first window starts, VARA selects PHY rate of 39Mbps, whose capacity of 30.34Mbps can satisfy the peak rate of 15.9Mbps. When the end boundary of the first window approaches, VARA discovers that the capacity of PHY rate 39Mbps cannot satisfy the peak rate of 12.4Mbps of the next window as the client has already moved to L5. By comparing the capacities yielded by 52Mbps and 26Mps, it finds that 26Mbps can support the peak rate. It then selects 26Mbps and uses it in the next window. Similarly, when the client moves to L6, VARA selects a PHY rate of 19.5Mbps for the next window.

Figure 8(a) also shows that VARA ensures that the goodput curve always matches the video streaming rate curve. An exception are the time instants where rate changes and the PHY rate shows deep spikes. These spikes are due to our user-space implementation. When the user-space issues a PHY rate change to the driver, there is a short-term interruption which causes the capacity reduction. Such effects can be removed with a kernel space VARA implementation.



(a) Video streamings when VARA is used.

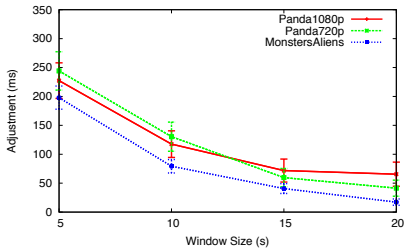


(b) Video streaming when the auto rate is used.

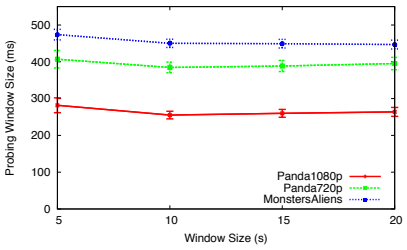
Fig. 8. Video streaming in a mobile environment.

Figure 8(b) shows the streaming rate, goodput and PHY rate changes when auto rate is used in the same scenario. Auto rate also tries to select the right PHY rate when the client moves from L4, to L5 and to L6. However, frequent unnecessary probings consume the capacity especially when the channel condition degrades (e.g., in L5 and L6). We can deduce the lack of channel capacity from the non-overlapping of the sending rate and the goodput curves. In addition, there is 30% burst loss rate when auto rate is used while there is no loss when VARA is used.

Furthermore, from our traces we find that the lack of capacity caused by the frequent unnecessary probings also increases the end-to-end delay of the streaming to around 500ms. In contrast, when VARA is used, the end-to-end delay is within 10ms.



(a) Adjustment of the end boundary.



(b) Probing window size.

Fig. 9. Effect of window size.

D. Analysis of window adjustments and probing windows

The results of both static and mobile experiments validate that VARA selects the right PHY rates and can significantly improve the video streaming quality. In this section, we use the three videos in Table I to evaluate the ability of Algorithm 2 to compute the end boundary and the probing window size η of the next window so that video streaming rate requirements are satisfied and probing overhead is minimized.

Window adjustments. To find the right PHY rates, VARA still needs probings. Therefore, it is crucial that Algorithm 2 in VARA can successfully find the appropriate boundaries of windows such that the probing will not lead to video streaming quality degradation. Even though an appropriate boundary is found, if the adjusted boundary computed by Algorithm 2 deviates from the original boundary computed by Algorithm 1 too much, it will drastically change the window size computed by Algorithm 1. Since Algorithm 1 in VARA computes the window size to adapt to the channel condition change, a large boundary adjustment by Algorithm 2 will lead to ineffectiveness in responding to the channel condition change.

Figure 9(a) shows that, as the window size increases, the adjustment by Algorithm 2 decreases. In VARA, probing occurs near the end boundary of the window. Therefore, the end boundary should not be set in the period in which the streaming rate is high. When the window size increases, due to the rate variability in VBR videos, there is a higher chance that the streaming rate near the original boundary is not close to the peak rate in the window. Therefore, as the window size increases, the adjustment needed by Algorithm 2 decreases.

Probing window size. In VARA, the probing process must be completed before the current window ends. If the probing window is too large, it will also lead to the decrease of the probability of finding an appropriate end boundary.

Figure 9(b) shows the probing window size computed by Algorithm 2 as a function of window size, for $n_p = 10$ probes, and also $|R| = 16$ PHY rates. The probing window size does not strongly depend on the window size. We also observe that the probing window size for panda1080p is the smallest while that of MonsterAliens is the largest among the three. This is because the probing window size is determined by the streaming rate. With implicit probing, videos with high streaming rate can finish the probing process faster than those with low streaming rate, since during the probing process a high streaming rate can send more probes per time unit.

Success rate. In all cases, the success rate for finding the appropriate boundary is over 95%. It is worth noting that when Algorithm 2 cannot find an appropriate end boundary, it does not mean VARA fails. In that case, Algorithm 2 will return a rate equal to $(1 + h)\rho'$ as the peak rate. Then Algorithm 1 will check if the current PHY rate can provide a capacity large enough for that returned peak rate, or Algorithm 3 will be called to find a such PHY rate.

The results of this section further prove the effectiveness of VARA. In all cases, the adjustments of the end boundary are small (less than 0.5s). Such small adjustments by Algorithm 2 would not change the window size computed by Algorithm 1 too much. That means the original window size of Algorithm 1 which adapts to the channel condition change is largely preserved. Moreover, the small probing window size (all are less than 0.5s) implies that even if VARA sets small windows to adapt to the dynamic channel condition change, it is still easy to find the period within the window for channel probing.

E. Reducing the Peak Rate by Strategic Shifting

We now investigate the ability of Strategic Shifting to minimize the aggregated peak rate of multiple videos in practice. To find the optimal shift δ' that minimizes the aggregated peak rate, our implementation uses exhaustive search to solve Equation (8). The computation delay and the quality of the solution of this implementation depend on the granularity of δ, the delay budget Δ, and the length of the videos.

We test the algorithm on four video clips whose lengths range from 72s to 2.5 minutes and their peak rates range from 14.54Mbps to 26.12Mbps. Three of them are from the videos in Table I. We also add another lower-rate HD video, *Wolfman*, which has the similar rate properties as *MonsterAliens* in this evaluation. We first use a granularity of 0.2s for δ and test the algorithm with two delay budgets $\Delta = 5s$ and $\Delta = 10s$, respectively.

At each experiment, we start a video and within the first five seconds we randomly schedule a time where one, two, or three other videos request to start streaming simultaneously.

In all experiments, our implementation of Strategic Shifting multiplexed any two streams within one second. Figure 10 shows the resulting aggregated peak streaming rates. We observe that Strategic Shifting provides an aggregated peak rate reduction between 15% and 25% compared to the case where no shift is applied. Thus, a δ granularity of 0.2s can already

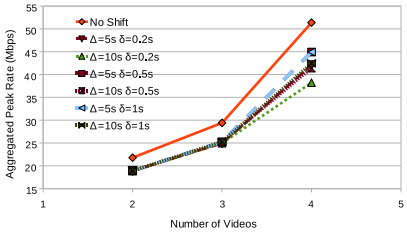
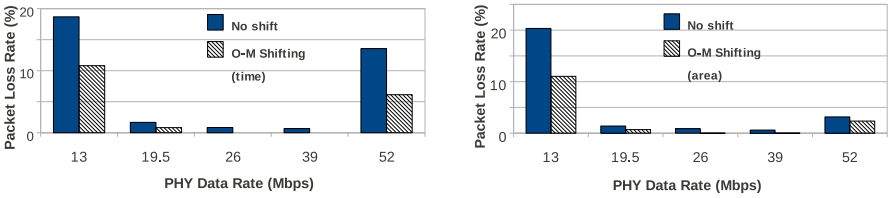


Fig. 10. The aggregated peak streaming rate when different number of videos are multiplexed.



(a) The outage time is minimized (b) The outage area is minimized

Fig. 11. The video streaming loss rate when using different O-M shifting.

demonstrate the benefits of Strategic Shifting. A δ granularity less than 0.2s might reduce the aggregated peak rate further, but it would also incur higher computation delay. In Figure 10, we also present the results when larger granularities, 0.5s and 1s, are used. They also demonstrate significant peak rate reduction compared to the case where no shift is applied. Furthermore, a larger Δ allows Strategic Shifting to further reduce peak rate, especially as the number of videos increases, as shown in 4 video case in Figure 10.

F. Reducing Loss by Outage Minimized (O-M) Shifting

We now evaluate the performance of Outage Minimized (O-M) Shifting in practice. Each experiment consists of two back-to-back runs, where two 2.5-minute HD videos (*Wolfman* and *MonsterAlien*) are streamed together on a link. The experiments are carried out when the client is in the location L3 in Figure 3(a).

In the first run we multiplex the two videos using O-M Shifting with $\Delta = 5s$ and δ search granularity of 0.2s. In the second run, no shifting is applied. Each experiment is repeated ten times and performed at different PHY rates.

Figure 11 shows the average packet loss rate of the second video stream.[5] Both O-M Shifting strategies of minimizing outage time and minimizing outage area, result in 20% to 50% reduction in packet loss rates compared to the no-shift case. At some PHY rates, which achieve the capacity of the wireless link, O-M Shifting removes all losses and supports the video streaming perfectly.

Similar to Strategic Shifting, if we use a larger delay budget, Δ, and a smaller granularity for the shift, δ, O-M Shifting can reduce the loss rate further, at the cost of higher computation time.

VI. Discussion

Our VARA evaluations focused on basic scenarios where one or more video flows are streamed on a single link. VARA can also operate in more complex scenarios such as hidden terminals, multiple competing links or when video traffic co-exists with best-effort traffic such as file downloads.

In presence of hidden terminals (and high collision rates), VARA will measure a higher FER and make a conservative decision of a low PHY rate. This is a common issue in all rate adaptation mechanisms and can be addressed by methods of separating collisions from channel errors (e.g. [7]). When multiple links compete, each link will select a PHY rate based on its video rate requirement and its own estimate of remaining channel capacity. When best-effort traffic is present, VARA can operate transparently with existing traffic differentiation mechanisms which would give higher priority to the video traffic (e.g. 802.11e traffic classes).

In addition, VARA selects the minimum rate that satisfies the video rate requirement. On the one hand, this results in more robust transmissions than higher eligible PHY rates. On the other hand, it results in packets of longer duration which may be more susceptible to interference or slow down other competing links transmitting at higher PHY rates.

In our future work we plan to perform a comprehensive evaluation of VARA in the above scenarios.

[5]The loss rate of the first video is similar to that of the second video.

VII. Related Work

802.11 rate adaptation mechanisms. References [6], [10]–[12] do not probe at multiple rates but make rate adaptation decisions based on SNR [10], [11], Channel State Information (CSI) [12] or FER [6] measurements at the current rate and then increase or decrease the rate based on pre-computed thresholds and rate lookup tables. However, lookup tables and thresholds change dynamically with the wireless channel and require in-situ training [13]. Also, most 802.11 chipsets cannot accurately measure SNR due to interference [14] and, except the implementation of [12] on Intel's iwl5300 chipsets, 802.11n chipsets typically do not export CSI from PHY layer to firmware or driver level. SampleRate [15] probes the channel by selecting a different PHY rate at random at every tenth data packet and reduces probing overhead by avoiding PHY rates that yield low link quality. In [5], Pefkianakis et al. introduce rate adaptation techniques for MIMO 802.11n systems which reduce implicit probing overhead by exploiting the monotonicity property between FER and PHY rate within SDM or STBC MIMO mode. SoftRate [16] adapts the PHY rate by differentiating whether the frame loss is due to the collision or channel error. However, it requires PHY layer information not provided by standard 802.11 hardware and shows significant throughput improvement only in mobile scenarios after training.

VARA differs from the above approaches in several aspects. First, it is the first approach that takes into account video rate requirements. Second, it is fully compatible with the

802.11 standard. Third, it measures FER and uses probing thus avoiding the challenge of determining rate tables and thresholds in a dynamic environment. Fourth, it not only reduces the probing frequency and probing overhead as [5] but also takes a cross-layer approach and uses video streaming information to efficiently schedule the probing during low-streaming-rate periods. This enables VARA to eliminate the probing's negative impact on video streaming. Finally, the techniques of separating collisions from channel errors developed in [16], [17] and in [7] are orthogonal to VARA and can assist on making more accurate rate adaptation decisions.

Video streaming systems. Recently, different systems are used for video streaming. Adaptive streaming techniques (e.g. HTTP adaptive streaming) rely on a server hosting multiple copies of a video in different streaming rates and qualities. Scalable Video Coding (SVC) encodes a video into different layers [18]. Chunks from the different copies or layers can be periodically requested to adapt the streaming rate to changing network capacities. The availability of multiple copies is typically limited to large-scale online video services with large storage capacities. For SVC, the clients should also be equipped with special SVC decoders. VARA does not rely on such systems. In addition, VARA takes an orthogonal approach and optimizes the wireless rates to best accommodate the video stream, rather than the other way around.

VBR smoothing. There is a body of work that considers smoothing VBR video streaming by decoupling the VBR video streaming rate from the actual transmission rate (see e.g., [19] and references therein). These schemes typically use *a priori* knowledge about the video stream to schedule pre-fetching of packets to the buffer to lower streaming peak rate and variance. Our strategic shifting aims at reducing peak streaming rate in the case of multiple streams. Our work differs in that it does not change the transmission scheduling from the server. We also do not need a large buffer in the receiver side for storing the pre-fetched packets. Our work multiplexes several video streams, whereas smoothing schemes typically consider only a single video. Moreover, recent work [20] has shown that VBR traffic characteristics after smoothing exhibits significant variability with H.264 compared to older encoding standards used in [19]. Our Strategic Shifting is therefore complementary to VBR smoothing.

VIII. Conclusions

Supporting HD VBR video in WLANs is a timely and challenging problem. Although WLAN technologies such as 802.11n MIMO support very high wireless PHY rates, we showed that in practice the probing overhead of existing state of the art 802.11n PHY rate adaptation protocols can be detrimental to video performance.

Our wireless rate adaptation protocol (VARA) addresses this problem by adapting the frequency and timing of wireless probing to the video streaming rate and the wireless channel variations. We also proposed three novel Shifting techniques to efficiently multiplex HD videos by minimizing peak aggregate streaming rate, outage time and outage area.

The feasibility of VARA was demonstrated in an indoor 802.11n testbed. Our experiments showed that VARA and the Shifting techniques can efficiently support one or more HD video streams. Compared to the default auto rate adaptation protocol it can eliminate or reduce packet loss rates to a few percents. For the video quality in PSNR, VARA can increase the lower-rate HD video to original perfect quality, while it can increase the PSNR of the higher-rate HD video by 50%.

To the best of our knowledge, VARA is the first video-aware wireless rate adaptation protocol to support HD videos in 802.11n WLANs. In our future work we will seek to further optimize HD video performance by combining VARA's wireless rate adaptation approach with adaptive streaming approaches such as HTTP streaming and SVC.

References

[1] Cisco System, "Cisco visual network index: Forecast and methodology, 2009 - 2014," *Cisco While Paper*, 2010.

[2] ISO/IEC 1449610:2003, "Coding of Audiovisual ObjectsPart 10: Advanced Video Coding, also ITU-T Recommendation H.264 Advanced video coding for generic audiovisual services." 2003.

[3] IEEE 802.11 Working Group, "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications – Amendment 5: Enhancements for Higher Throughputs," 2009.

[4] K. Pelechrinis, T. Salonidis, H. Lundgren, and N. Vaidya, "Experimental Characterization of 802.11n Link Quality at High Rates," in *Proc. ACM WiNTECH*, 2010.

[5] I. Pefkianakis, Y. Hu, S. H. Wong, H. Yang, and S. Lu, "MIMO Rate Adaptation in 802.11n Wireless Networks," in *Proc. ACM MobiCom*, 2010.

[6] S. H. Y. Wong, H. Yang, S. Lu, and V. Bharghavan, "Robust Rate Adaptation for 802.11 Wireless Networks," in *Proc. ACM MobiCom*, 2006.

[7] T. Salonidis, G. Sotiropoulos, R. Guerin, and R. Govindan, "Online Optimization of 802.11 Mesh Networks," in *Proc. ACM CoNEXT*, 2009.

[8] Y. Wang, "Survey of objective video quality measurements," in *Technical report*. Worcester Polytechnic Institute, 2006.

[9] A. Chan, K. Zeng, P. Mohapatra, S.-J. Lee, and S. Banerjee, "Metrics for Evaluating Video Streaming Quality in Lossy IEEE 802.11 Wireless Networks," in *Proc. IEEE INFOCOM*, 2010.

[10] G. Holland, N. Vaidya, and P. Bahl, "A rate-adaptive mac protocol for multi-hop wireless networks," in *Proc. ACM MobiCom*, 2001.

[11] G. Judd, X. Wang, and P. Steenkiste, "Efficient channel-aware rate adaptation in dynamic environments," in *Proc. ACM MobiSys*, 2008.

[12] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "Predictable 802.11 packet delivery from wireless channel measurement," in *Proc. ACM SIGCOMM*, 2010.

[13] J. Camp and E. Knightly, "Modulation rate adaptation in urban and vehicular environments: Cross-layer implementation and experimental evaluation," in *Proc. ACM MobiCom*, 2008.

[14] J. Zhang, K. Tan, J. Zhao, H. Wu, and Y. Zhang, "A practical snr-guided rate adaptation." in *Proc. IEEE INFOCOM*, 2008.

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

[16] M. Vutukuru, H. Balakrishnan, and K. Jamieson, "Cross-layer wireless bit rate adaptation," in *Proc. ACM SIGCOMM*, 2009.

[17] S. Rayanchu, A. Mishra, D. Agrawal, S. Saha, and S. Banerjee, "Diagnosing wireless packet losses in 802.11: Separating collision from weak signal," in *Proc. IEEE INFOCOM*, 2008.

[18] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the h.264/avc standard," *Circuits and Systems for Video Technology, IEEE Transactions on*, 2007.

[19] J. Salehi, Z.-L. Zhang, J. Kurose, and D. Towsley, "Supporting stored video: reducing rate variability and end-to-end resource requirements through optimal smoothing," *IEEE Transactions on Networking*, 1998.

[20] G. Van der Auwera, P. David, M. Reisslein, and L. Karam, "Traffic and quality characterization of the h.264/avc scalable video coding extension," *Journal Advances in Multimedia*, 2008.

MIMO Wireless Networks with Directional Antennas in Indoor Environments

Tae Hyun Kim[†], Theodoros Salonidis[‡], and Henrik Lundgren[‡]
The University of Illinois at Urbana-Champaign[†] Technicolor Paris Research Lab[‡]

***Abstract*—We perform an experimental characterization of an indoor MIMO system with directional antennas (our prototype multi-sector antennas). The study reveals that, even without antenna directivity gain, the directionality of signals changes the MIMO channel structure and provides a way to improve MIMO throughput performance. It also shows that it is sufficient for the improvement to consider only a small subset out of all possible antenna direction combinations. Finally, our study shows that it is possible to achieve both throughput gains and interference reduction, thus increasing network spatial reuse.**

I. Introduction

Due to the increasing demands for large throughput wireless communications, Multiple Input Multiple Output (MIMO) has become one of the key technologies. MIMO combines multiple omni-directional antennas with signal processing techniques to extend the dimension of available wireless resources to time, frequency, and space [1]. Adopted in many standard protocols such as [2]) MIMO has already been widely deployed to transport streamed voice and high-definition video traffic applications.

A second wireless technology that has long been receiving interests is directional antennas. Directional antennas use narrow beams to focus RF energy toward desired receivers. This achieves throughput gains and reduces interference. Moreover, the directivity of antennas facilitates determining proper orientation in outdoor environments [3]. Recently, it has been shown that directional antennas in indoor environments provide a few strong paths between nodes even without the line-of-sight path. The consequent benefits are empirically demonstrated, attracting renewed interests [4], [5].

The benefits and losses of combining MIMO and directional antennas have not been fully studied to date. MIMO achieves capacity gains in rich scattering multi-path environments. Outdoor environments typically have a single strong line-of-sight path and a directional antenna would decrease the capacity of a MIMO link. In contrast, it is not clear how directional antennas for MIMO would perform in indoor environments. On one hand the narrow beams of directional antennas might decrease the degree of multi-path or signal scattering and decrease the capacity of MIMO links. On the other hand, directivity might change the structure of propagation paths and increase signal strength thus increasing the MIMO link capacity.

This paper takes an experimental approach to develop a principled understanding of the performance of MIMO wireless networks with directional antennas. For experiments, our 802.11n based MIMO testbed is equipped with multi-sector antennas without directivity gain. We use the term directivity gain to indicate the additional antenna gain of a directional antenna toward one direction, compared to an omni one. Also, the term sector activation pattern or simply sector pattern indicates a combination of activated sectors for all antennas involved in a transmission. We investigate the impact of different factors such as RSS, the number of activated sectors, Tx (transmit) or Rx (receive) sector activation, geographic location, and interference.

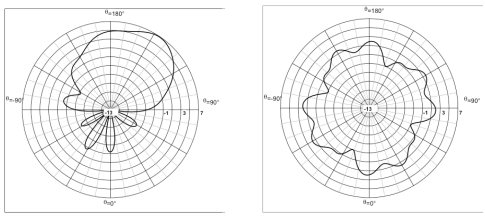
We make three interesting observations through the empirical study. First, even without directivity gain, multi-sector antennas can yield large throughput gains by changing the structure of MIMO channels. However, they may also produce large throughput loss if sector activation patterns are not properly selected. Interestingly, this gain and loss are all observed in a very small subset of activation patterns. Second, the sector patterns with the highest throughput largely depend on the environment and are not associated with the number of activated sectors, Tx or Rx sector activation or antenna orientation. Third, the interference level when multi-sector antennas are used is proportional to the number activated sectors. This allows us to have a coordination mechanism for more concurrent transmissions, improving the spatial reuse of a network.

Related Work: Aside from [3], [4], [5], the performance of MIMO with directional antennas was studied in both indoor and outdoor environments in [6]. However, the study was under fixed orientation of the antennas while this paper is about the sector activation, which is equivalent to the change of antenna orientation. Unlike their conclusion that the gain of using directional antennas is marginal, we will show that the control of antenna orientation can bring a huge throughput gain. In [7], only outdoor to indoor scenarios are considered with directional multiple antennas. In this case, less scattering is expected, compared to the indoor-to-indoor case that this paper concerns about.

II. Experimental Setup

A. Multi-sector Antennas

We use the Vivaldi multi-sector antenna developed by Technicolor Research. Each antenna has four antenna elements (sectors) printed on a PCB and covers the entire horizontal plane in the 5 GHz band. Any combination of sectors can be activated through a feeding network, which provides $(2^4 - 1) = 15$ activation patterns. One of them is four activated sectors, corresponding to an omni directional mode



(a) Single sector pattern. (b) Omni-mode pattern.

Fig. 1. Radiation pattern of multi-sector antenna without feeding loss.

TABLE I
ANTENNA GAINS (GAINS IN DBI AND LOSS IN DB)

	Omni	3 Sec	2 Adj	2 Opp	1 Sec
Directivity gain	2.4	3.5	4.6	5.7	6.9
Feeding loss	0	-1.25	-3	-3	-6
Overall gain	2.4	2.25	1.6	2.7	0.9

pattern. We simply call it omni-mode. Fig. 1(b) and 1(a) depict the radiation patterns with one and four activated sectors, respectively. Table I shows that the antenna directivity gain for each pattern depends on both number of activated sectors and, for the case of two activated sectors, on whether they are opposite (*2 Opp*) or adjacent (*2 Adj*). We see that the directivity gain is higher for sector patterns with less activated sectors. However, the antenna feeding network has been designed to introduce a feeding loss such that all sector patterns exhibit approximately equal peak gains.

B. Testbed

Our wireless testbed is deployed in a single floor in the Technicolor Lab, as shown in Fig. 2. This is a typical office environment consisting of cubicles, booths and offices separated by glass walls. Due to the availability of multi-sector antennas, we have only four nodes, but different topologies are emulated by Tx power control.

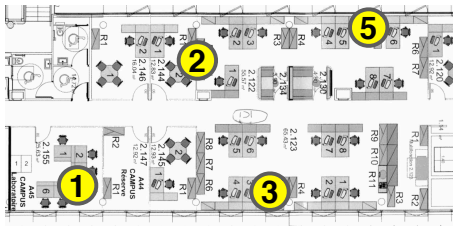


Fig. 2. The testbed deployment in Technicolor Paris Research Lab.

Each testbed node is a PC with Intel Pentium (M) 1.73 GHz processor and 512 MB RAM, running Ubuntu Linux distribution version 10.04. The PC hosts a commercial mini-PCI 802.11n Network Interface Card (NIC) with a Ralink RT2880 chipset. A NIC includes three antenna ports with two radio chains supporting both Spatial Division Multiplexing (SDM) and Space-Time Block Coding (STBC) MIMO communications. We disable one of three ports to have 2×2 MIMO. Ralink NIC is also capable of adjusting Tx (transmit) power level to one of 18, 17, 15, 12, 9 and 6 dBm.

During the experiments, the NICs are configured to operate in the 5.3 GHz band with 20MHz bandwidth with 400 ns guard-interval. During operation the NICs can select among basic 16 Modulation and Coding Sets (MCS), which correspond to 8 modulation rates under SDM MIMO mode and 8 under STBC MIMO mode. Each node has two Vivaldi multi-sector antennas connected to individual radio chains in a RT2880 NIC. The spacing between antennas is set to 7 cm according to our previous work with omni-antennas [8]. The feeding network of each Vivaldi antenna is controlled via a USBIO24 R Digital I/O Module interface, which allows us to control sector activations directly from the PC host using user-level Linux shell scripts.

III. THROUGHPUT ESTIMATION

In this section, we describe how we address multi-sector antenna MIMO measurement challenges and show that SNR can be used as a throughput predictor.

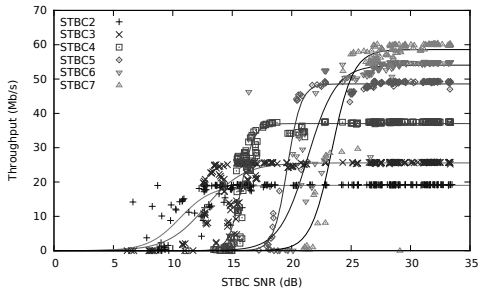
A. Measurement Challenges

We use UDP throughput as performance metric. However, direct measurement of the maximum throughput of a MIMO link with multi-sector antennas entails the following challenges.

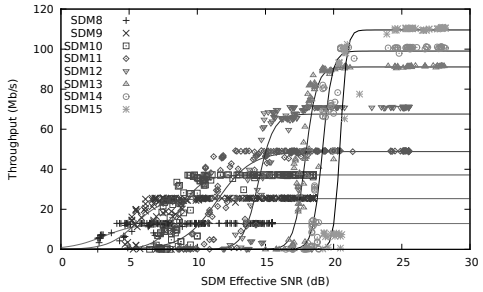
Large number of activation patterns: In a $M \times M$ MIMO system with K MCS data rates, where both Tx and Rx use multi-sector antennas of s sectors each, testing all combinations requires $K \times (2^s - 1)^{2M}$ throughput measurements. This corresponds to 810,000 throughput measurements in our system, where $M = 2$, $s = 4$ and $K = 16$. To address this, we restrict the number of activation patterns considered for each link. Specifically, sector activation is performed at either Tx or Rx, with the other end of the link in omni-mode. Moreover, the number of active sectors is kept the same for each antenna. We call the restricted set of activation patterns as a *pattern set*. This reduces the number of considered sector patterns from $(2^s - 1)^{2M}$ to $\sum_{x=1}^{s-1} \binom{s}{x}^M$. In our case, $\sum_{x=1}^{3} \binom{4}{x}^2 = 68$ which comprise 6 pattern sets.

Multiple MCS rates: We cope with multiple MCS rates, using UDP throughput vs. SNR mappings. In addition to Received Signal Strength (RSS) provided by most 802.11 wireless cards, the hardware of our Ralink R2880 chipset internally stores Signal-to-Noise Ratio (SNR) information for each received packet, but does not export it. Our workaround is to directly access the internal memory to obtain this information. If a packet is encoded in the SDM mode, a pair of SNR values (SNR per spatial stream) is available, and, if encoded in the STBC mode, a single SNR value is available. We call these types of SNRs as SDM SNR and STBC SNR, respectively.

Measurement under channel variation: Another challenge is that the measurement results for different activation patterns may be affected by time variation of wireless channels. To quantify this impact, we measure the fraction of SDM effective SNR samples which fall into $\pm\delta$ dB range of the long term



(a) STBC SNR to throughput.



(b) SDM effective SNR to throughput.

Fig. 3. Ralink RT2880 chipset-specific SNR to throughput relationships.

average. The effective SNR is a single representative SNR for a pair of SDM SNRs proposed in [9]. At night, only 10% of the samples deviate more than 2 dB from the long term average, even for time intervals up to 200 seconds. In daytime, around 15% of the samples deviate more than 2 dB for time intervals of 20 seconds, which is close to the 1 dB granularity of our 802.11n cards. To further minimize the outliers' effect, we perform all measurements at night (unless specified otherwise), measure throughputs of each activation pattern and omni-mode back-to-back, and take their difference, measuring relative change at each instance.

B. SNR-based Throughput Estimation

We set up a series of experiments to map our hardware-specific SNR information to UDP throughput. For 5 seconds, we measure the UDP throughput of each MCS rate with each of the restricted pattern sets, using the `iperf` and `tcpdump` tools. We found that 5 seconds measurement duration for each configuration is sufficient and that this duration results in the best mapping between SNR and throughput. The traffic load is set to be higher than each MCS rate. Since our workaround to extract SNR does not work when the NIC is busy to process incoming packets, as soon as the throughput measurement is over, a light traffic load (1 Mbps) is applied to collect STBC and SDM SNR information for 5 seconds for each. Then, the SNR values are averaged and mapped to the throughput. Note that for SDM, each SNR value pair is combined to a single effective SNR [9] for one-to-one mapping to the throughput. We emulate different topologies by power control to diversify the range of samples.

Fig. 3 shows each of the averaged throughput samples and corresponding SNR or (effective SNR) values. To generate the mappings, we fit a generalized Sigmoid function to each set of throughput samples with the same MCS rate.

In the rest of the paper, we estimate the throughput of a link using a certain sector pattern by measuring the STBC SNR and SDM effective SNR values, converting these two SNR values to throughputs using the mappings, and taking the maximum.

IV. Throughput Gains

Experimental setup. We use the same setup for SNR measurement as Section III. The SNR of omni-mode is also measured in a back-to-back fashion to obtain the throughput gain via the constructed mapping in the previous section. We repeat experiments five times for 10 links with all 6 pattern sets.

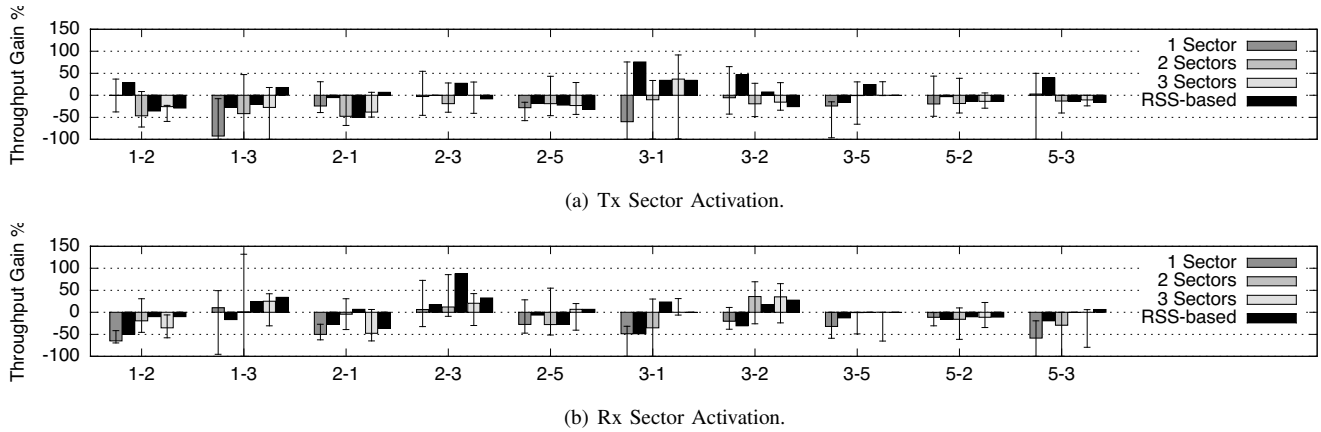
Throughput gains. Fig. 4 shows the throughput gains for all pattern sets that we consider. Each gray bar in Fig. 4 shows the median gain for one, two and three activated sectors per antenna; its error bars indicate maximum and minimum throughput gains. The black bars will be explained shortly.

Most links achieve a positive maximum throughput gain over omni mode (at most 130% on link 1-3 with 2 Rx sectors per antenna and 21% averaged across all links). This appears counter-intuitive because, as mentioned earlier, our multi-sector antennas do not provide antenna directivity gain over omni mode. Since multi-sector antennas transmit or receive at equal or less signal power, and some of diverse paths between Tx and Rx are suppressed, one might argue that it would not be possible to observe positive throughput gains.

The positive throughput gains are due to the clustered propagation of signals in the angular domain. Propagation measurements in indoor environments [10] have shown that the angles of departure (AoD) and arrival (AoA) form correlated signal clusters. Moreover, only 2 to 4 clusters mostly contribute to the received signals. The sector patterns that achieve positive throughput gains, are aligned in phase with these dominant signal clusters, thus avoiding negative gains. At the same time, they are also misaligned with other clusters that induce signal correlations at the antenna input. This misalignment reduces the received signal correlation, making multiple information streams in MIMO channel appear more de-correlated. *Therefore, sector activation without directivity gain can create throughput gains by structurally changing the MIMO channel.*

Despite the potential for a large positive maximum throughput gain, Fig. 4 also shows that most links achieve negative median throughput gain (-9.3% in average), and the minimum gain can reach as low as -100% (e.g., link 1-3 for 2 Rx sectors per antenna). Thus, less than half of the sector patterns provide positive gains and if a sector activation pattern is not carefully chosen, it may yield a large penalty on throughput performance.

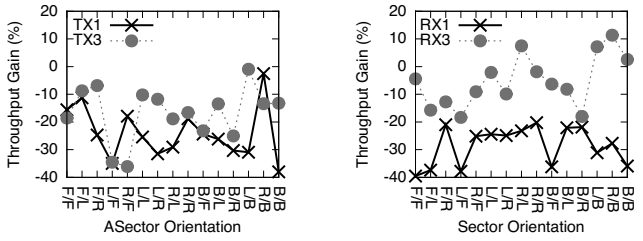
Pattern selection criteria. Can we determine patterns with maximum or positive throughput gain in a systematic way? The answer may depend on different criteria investigated below.



(a) Tx Sector Activation.

(b) Rx Sector Activation.

Fig. 4. Median throughput gain per link and the gain when a pattern with the highest RSS is chosen. Error bars on each median gain are the maximum and minimum gains, respectively.



(a) Tx activation

(b) Rx activation

Fig. 5. Throughput gain vs orientation of active sectors. TX1 and TX3 indicate 1 Tx and 3 Tx sector activation patterns, respectively (RX1 and RX3 correspond to RX activation). F, L, R, and B abbreviate Face, Left, Right, and Back, respectively. For example, F/F in TX1 (TX3) orientation means both antennas have their active Tx sectors which face the receiver.

1) Number of activated sectors per antenna: Fig. 4 shows that the maximum throughput gains do not depend on the number of activated sectors or whether Tx or Rx activations are used.

2) Geographical orientation of active sectors: Fig. 5 depicts the average throughput gains over all links, as a function of sector pattern orientations sorted by geographical direction of activated sectors toward the other end of the link. Unlike the case with a single multi-sector antenna per node [4], the geographical relationship is not correlated with throughput gains, again regardless of number of active sectors and Tx or Rx activation.

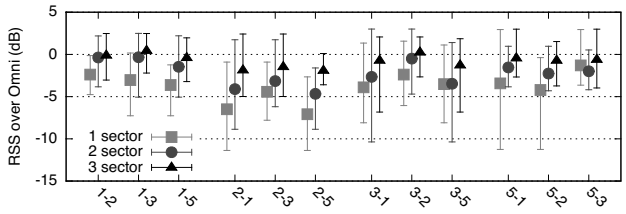
3) Reciprocity: For most node pairs (x,y) in Fig. 4, the performance of link x-y with Tx activation can be radically different to the performance of reciprocal link y-x with Rx activation. Thus, in general link reciprocity cannot be leveraged to reduce measurement overhead.

4) RSS: Each black bar next to a gray bar in Fig. 4 is the throughput gain when the pattern with the maximum RSS is chosen among the corresponding activation patterns. For most links, the throughput gains from RSS-based selection are not close to the maximum throughput gains. Instead, they are close to the median throughput gains, which are primarily negative. This holds for both Tx and Rx activation in Figs. 4(a) and 4(b), respectively. Thus, without antenna directivity gain, the RSS cannot predict throughput gain well.

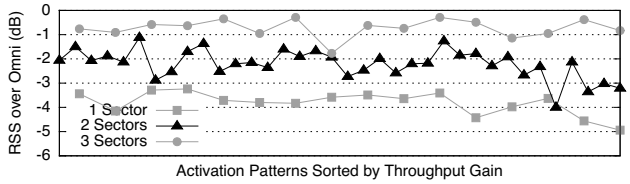
5) Location: One may guess that the trend of throughput gains over sector patterns in a fixed pattern set is similar to those of other links, irrespective of their locations. If so, a probability distribution for the gains could be found, which might lead us to a systematic way to check if it is possible to achieve a positive gain and to further find good candidate patterns to achieve it. Not shown in this paper due to the space limit, however, it is observed that individual links show radically different trends. Therefore, it is not straightforward to develop a systematic way to find a good pattern, which works across all links.

Reason for arbitrary throughput gains. Based on the above observations, we conclude that the throughput gains appear "arbitrary." Our interpretation is that sector activation changes the structure of MIMO channels in a way that depends highly on the surrounding environment, which is "arbitrary." In theory, the relationship between Tx and Rx antenna signals in a $M \times N$ MIMO channel can be simplified as $\mathbf{y} = \gamma \cdot \mathbf{H}\mathbf{x}$ where $\mathbf{x}$ is an $1 \times M$ input (Tx symbol) vector, γ is a scalar path loss-based channel gain, $\mathbf{H}$ is the normalized $M \times N$ MIMO channel matrix, and $\mathbf{y}$ is an $1 \times N$ output (Rx symbol) vector. Our previous observation that RSS is not a good indicator of the throughput gain means that the major contributor to throughput gain is the structure of MIMO channel matrix $\mathbf{H}$ instead of the scalar gain γ. Existing studies state that the structure of $\mathbf{H}$ highly depends on the surrounding environments of a link and the resultant paths [10]. In indoor environments, the surroundings for each link are noticeably different and lead to different throughput gain characteristics across activation patterns.

We conclude that it is possible to achieve positive throughput gains even with a limited subset of all patterns. By collecting SNRs of activation patterns from either Tx or Rx pattern set with a fixed number of activated sectors, multi-sector antennas provide 21% of throughput gains in average



(a) Average RSS_{diff} without antenna directivity gain: Average interference differences of Tx sector activations over omni mode at neighbors of each link.



(b) Average RSS_{diff} across all sector patterns: Average interference difference at neighbors. For each link, the patterns are first sorted by descending throughput gains and then the RSS_{diff} values with the same ranking are averaged.

Fig. 6. Interference properties without antenna directional gain.

for MIMO communications. Our measurements also show that it is challenging to nd a criterion to select a pattern with positive throughput gain from the limited pattern set because of the arbitrary change of MIMO channel structure in the indoor environment. Thus, selecting a good pattern might require to periodically collect SNRs of all patterns in a pattern set. Although one may argue that the measurement overhead cannot be ignored, we claim that it can be reduced as SNRs are collected at a lower frequency.

V. Interference properties

So far the focus of performance characterization has been on a link. This section provides some insights into network performance through the analysis on interference properties of MIMO multi-sector antennas.

Experimental setup. Using a similar experimental methodology, we measure SNR and RSS for each pattern in the Tx activation sets, immediately followed by omni-mode. We use the difference $RSS_{\text{diff}} = RSS_x - RSS_{\text{omni}}$ as interference metric. A negative value means that sector activation pattern x reduces interference compared to omni-mode and increases spatial reuse. All measurements are performed at night, and the results are the average of five iterations.

Interference without directivity gain. Fig. 6(a) depicts the average RSS_{diff} at the neighborhood of each link when sector activation is in use. For example, the 1 sector point of link 1-2 is the average of RSS_{diff} values from all neighbor links, which are link 1-3 and link 1-4. From Fig. 6(a), the interference reduction increases as the number of active sectors decreases. With 1 Tx sector per multi-sector antenna, the interference over omni-mode can be reduced up to 12 dB at maximum (link 2-1) and 8 dB on average (link 2-5).

Although sector activations reduce interference level, they may not necessarily increase throughput gain. Fig. 6(b) depicts the interference amount in descending order of throughput gains. We observe that, for each number of activated sectors, the amount of interference is not related to the throughput gains. Thus, by selecting a number of activated sectors, it is possible to maximize throughput gain subject to a constant interference level, which is minimum when 1 Tx Sector activation patterns are considered.

We conclude that the interference level without antenna directivity gain is proportional to the number of activated sectors and has little correlation with the amount of throughput gain. Therefore, one can exploit spatial reuse in addition to throughput gain to enhance network-wide performance. However, spatial reuse comes at the expense of coordination mechanisms of multiple concurrent transmissions.

VI. Conclusions

In this paper we presented performance characteristics of multi-sector antenna-equipped IEEE 802.11n MIMO wireless networks. Even with the absence of directivity gain, the use of sector activation can improve throughput over omni-mode, and leaves a room for further improvement through spatial reuse. Our empirical study confirms the benefits of multi-sector antennas for MIMO communications, calling for future work on the details of how it can be harvested in real network scenarios.

References

[1] A. J. Paulraj, D. A. Gore, R. U. Nabar, and H. Bolcskei, "An overview of MIMO communications- A key to gigabit wireless," *Proceedings of IEEE*, vol. 92, pp. 198–218, 2004.

[2] 3GPP Workgroup R1, *Requirements for further advancements for E-UTRA (LTE-Advanced)*, 3GPP TR 36.913, Std.

[3] T. Korakis, G. Jakllari, and L. Tassiulas, "A MAC protocol for full exploitation of directional antennas in ad-hoc wireless networks," in *Proc. ACM MobiHoc*, 2003.

[4] A. P. Subramanian, H. Lundgren, T. Salonidis, and D. Towsley, "Topology control protocol using sectorized antennas in dense 802.11 wireless networks," in *Proc. IEEE Internat. Conf. Network Protocol (ICNP)*, Sept. 2009.

[5] X. Liu, A. Sheth, M. Kaminsky, K. Papagiannaki, S. Seshan, and P. Steenkiste, "Pushing the envelope of indoor wireless spatial reuse using directional access points and clients," in *Proc. ACM MobiCom*, Sept. 2010.

[6] N. Razai-Ghods, M. Abdalla, and S. Salous, "Characterisation of MIMO propagation channels using directional antenna arrays," in *in Proc. IEEE Sarnoff Symp.*, Princeton, NJ, Mar. 2009.

[7] C. Hermosilla, R. Feick, R. A. Valenzuela, and L. Ahumada, "Improving MIMO capacity with directive antennas for outdoor-indoor scenario," *IEEE Trans. Wireless Comm.*, vol. 8, no. 8, pp. 2177 – 2181, May 2009.

[8] K. Pelechrinis, T. Salonidis, H. Lundgren, and N. Vaidya, "Analyzing 802.11n performance gains," in *Proc. ACM MobiCom (Poster Session)*, Sept. 2009.

[9] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "Predictable 802.11 packet delivery from wireless channel measurements," in *Proc. ACM SIGCOMM*, Aug. 2010.

[10] Q. H. Spencer, B. D. Jeffs, M. A. Jensen, and A. L. Swindlehurst, "Modeling the statistical time and angle of arrival characteristics of an indoor multipath channel," *IEEE Journal on Sel. Areas in Commun.*, vol. 18, no. 3, pp. 347–360, 2000.