On Demand Connectivity Sharing: Queuing Management and Load Balancing for User-Provided Networks

Ioannis Psaras[a], Lefteris Mamatas[b]

[a] *Dept. of Electronic and Electrical Engineering*
University College London
WC1E 7JE, Torrington Place, London, UK
[b] *Dept. of Technology Management*
University of Macedonia
Thessaloniki, Greece

Abstract

We introduce the concept of *"On Demand Connectivity Sharing"*, which we build on top of *User-Provided Networks* (UPNs). UPNs were recently proposed as a new connectivity paradigm, according to which home-users share their broadband Internet connection with roaming guests. We enhance this paradigm with incentives, rules and policies, based on which: (i) home-users provide on-demand connectivity only (i.e., they do not explicitly allocate a portion of their bandwidth) and (ii) guest-users utilize resources that remain unexploited from the respective home-users.

We realize the *"On Demand Connectivity Sharing"* concept through a queuing algorithm that classifies traffic according to its source (i.e., home- or guest-traffic) and prioritizes home- against guest-traffic accordingly and a probabilistic load-balancing algorithm that guarantees smooth cooperation between home- and guest-users. We show both analytically and through extensive performance evaluation that it is indeed possible for a home-user to share his connection with guest-, roaming-users without any practical impact on his own network performance.

Email addresses: i.psaras@ee.ucl.ac.uk (Ioannis Psaras), e.mamatas@uom.gr (Lefteris Mamatas)

Preprint submitted to Elsevier *September 10, 2010*

The concept of *"On Demand Connectivity Sharing"* through *User-Provided Networks* is expected to receive a lot of attention in the years to come, since it enables a new notion of autonomous and self-organized mobile computing. For example, we gather information regarding the location and range of real WiFi access points in the city center of London and we show that a walking user can receive acceptable services, when acting as a guest-user and gets resources from near-by home-networks.

1. Introduction

One of the main features of the Future Internet is that it is going to be *everywhere*, at *anytime* and for *everyone*. Two main approaches have been identified till now to achieve that goal: (i) mobile Internet through 3G links (e.g., [1], [2]) and (ii) mobile Internet as an extension of the network itself (e.g., [3], [4]). The first approach assumes that users use telecommunication channels to reach the Internet (i.e., 3G links), while the second assumes extension of the network infrastructure and cooperation among users (possibly in an ad hoc manner) in order to bring connectivity further away from the strict boundaries of the traditional Internet (e.g., VANETs [3], or mesh networks [4]). Mobile operators, however, initially designed and setup their networks to carry voice traffic only. Then SMS text messages came into play and formed a major source of revenue for telecommunication vendors (i.e., very few bytes for a relatively expensive price - almost infinite Return Of Investment) and lately Internet connectivity is provided as part of the user's contract with the mobile operator. It is questionable though whether the telecommunication network will be able to handle large volumes of data instead of simple voice transfers and short text messages, once billions of users make use of such services. That said, extension of the network itself in order to offload 3G traffic to WiFi networks and achieve ubiquitous connectivity seems to be a more realistic, elegant and scalable approach[1].

[1]E.g., http://www.muniwireless.com/2010/02/19/will-wifi-rescue-3g/

The first step on that direction has been made with *mobile ad hoc and mesh networks* (e.g., [4], [5], [6], [7]), where nodes act as relays for messages they are not interested in. These nodes, obviously, consume resources (e.g., energy) to receive and transmit messages further in order to reach the destination, without having explicit gain themselves. In our opinion, this formed the fundamental first step towards the *"On Demand Resource Sharing"* era. With the evolution of "peer-to-peer" networks [8], users provide access to their local resources (i.e., hard-drive) to other users in order to speed up bulk data transfers. Data transfers take place in a distributed manner keeping congestion away from the main backbone links. Lately, a number of new concepts have attracted the attention of the research community, again in the direction of *On Demand Resource Sharing*. For example, in *Delay-Tolerant Networks* (DTNs) [9], [10], nodes *store, carry and forward* messages in order to deal with large delays, or disrupted connections. Hence, they act as relays and consume resources to forward data they are not particularly interested in, or *Online Social Networks* [11], [12], where users receive, store, carry and forward messages to and from other users regarding social interests [13] and habits [14], [15].

In this study, we propose that *"On Demand Resource Sharing"* has to be complemented with *"On Demand Connectivity Sharing"* in order to: (i) extend the traditional Internet's connectivity boundaries, (ii) give rise to the Future Mobile Internet and (iii) enable new types of technologies and applications, such as DTNs, CCNs and Online Social Networks. To reach the point where users can share connectivity resources, however, there is a number of issues that need to be investigated, such as for example, *incentives* for the home-user to offer resources and *performance implications* due to sharing [16]. We attempt to address the above issues building on the concept of *"User-Provided Networks"* [17].

In *User-Provided Networks* (UPNs), a home-user, owner of a broadband Internet connection, provides connectivity to unknown, mobile users that roam within the home-user's Access Point (AP) connectivity range. In that sense, the home-user is a consumer of Internet resources provided by the Internet Service

Provider (ISP) and at the same time provider of Internet services to roaming users. The home-user is, thus, a *micro-provider* [17]. Throughout this paper, we use the terms *"home-user"* and *"micro-provider"* interchangeably to denote the owner of the broadband Internet connection.

According to the authors of [17], the UPN connectivity paradigm needs to: (i) be easily deployable in terms of software and hardware modifications, (ii) provide the appropriate incentives to users to join the community (i.e., the more the micro-providers, the more dense connectivity is), (iii) guarantee that only liable users can access the *micro-provider's* resources and moreover, that they do not misbehave and (iv) manage roaming users connectivity-wise in a self-adaptive and autonomous way.

In this study, we show that *it is possible for the home-user to share his broadband connection with mobile, roaming users without any noticeable impact on his own performance*. In turn, the home-user gains unlimited connectivity (by other micro-providers) when he is out of home or office. We achieve the above by introducing the following two algorithms for UPNs:

1. An **active queue management** algorithm, that classifies and schedules home- and guest-packets according to different priorities. These priorities are designed so that (i) the home-packets are always favored against guest-packets and (ii) guest-packets go through only if there is sufficient space for them at the UPN-AP's queue. We show that this packet classification and scheduling algorithm guarantees that *sharing has practically no impact on the home-user's performance*.
2. A simple **load-balancing** mechanism that routes guest-traffic through the least-congested UPN-AP. That is, if a roaming user is within range of more than one UPN-APs, then his traffic is routed through the least-utilized AP. This algorithm is also complemented by a probabilistic switching scheme to guarantee that not all mobile-users switch to the same UPN-AP.

Therefore, the incentive for the home-user to share his broadband Internet connection is simple: he shares a portion of his bandwidth *seamlessly*, when at home and gains unlimited connectivity from other home-users, when mobile. We call this scheme *Offer Nothing - Gain Something*. Of course, this *"Something"* depends on the density of the available access points as well as on the bandwidth utilization of the respective home-users. That is, connectivity may be poor and intermittent, making DTN technology an essential complementary part for UPNs. The argument here, however, is that even poor, opportunistic connectivity is enough to download a web page, an e-mail or a map, which are popular mobile applications; we verify this claim by realistic simulation experiments. We argue that *connectivity is for mobile computing what bandwidth is for the wired core of the Internet*. Therefore, instead of high-speed core Internet links, *the backbone for a mobile environment is the area of dense connectivity*. We contend that *On Demand Connectivity Sharing*, through *User-Provided Networks*, comes as the natural evolution of *On Demand Resource Sharing* schemes; these new connectivity and data management paradigms that enable and support the notion of *autonomous* and *self-organizing networks* are essential for the realization of the Future Internet that inherently supports mobility.

We clarify the following:

- A mobile user that connects to different access points and stays there connected for some time (e.g., from the hotel-AP to the restaurant-AP and later on to the conference venue-AP) requires manual network discovery and re-connection; this notion of mobility has already been introduced by FON [18], OpenSpark [19] and Whisher [20]. However, users may be mobile 100% of the time, e.g., a walking-user roaming on the street or inside a vehicle. This user needs autonomous (i.e., self-adaptive) and on-the-fly connectivity to several different APs. Here, we focus on the second case of mobility, which we consider as an enabler for the future mobile Internet.

- Service differentiation through active queue management has received a lot of attention in the past few years (e.g., [21], [22], [23]), mainly in order to prioritize high-paying customers' traffic against low-paying ones', or to boost the performance of non-elastic, or non-congestive applications against bursty TCP transfers. Load-balancing, on the other hand, has been studied mainly in the context of server farms (e.g. [24]), peer-to-peer networks (e.g., [25]) and multipath routing mechanisms (e.g., [26], [27], [28]). The novelty introduced herein is the actual marriage and application of these techniques in the context of UPNs in order to realize new communication technologies and connectivity paradigms that explicitly support mobile computing. To the best of our knowledge, this is the first study that puts these two research fields under a different context, that of connectivity sharing and elaborates on their potential to set the foundations for the realization of the *"On Demand Connectivity Sharing"* through *User-Provided Networks*.

The rest of the paper is organized as follows: we begin with a description of the incentives that would convince the end-user to switch to the UPN scheme and share his broadband Internet connection with mobile users (Section 2). In Section 3, we present our design proposals; namely the *UPN Queuing* (UPNQ) algorithm for *service differentiation* between the home- and the guest-user and the *UPN Load Balancing* (UPNLB) algorithm for load-balancing among roaming users. Next, in Section 4, we describe our experimental setup and present our simulation results. Finally, in Section 5, we discuss open issues and future challenges and conclude the paper in Section 6.

2. The Offer Nothing - Gain Something Sharing Scheme

We consider that the success of the *"On Demand Connectivity Sharing"* scheme depends mainly on the specific incentives given to home-users in order to motivate them to shift into this new connectivity paradigm. This process consists of two main steps: (i) offer attractive "deals" to prospective users and (ii) design the corresponding algorithms to guarantee smooth and scalable operation of the proposed offers.

We attempt to pose and answer questions that would naturally come to a user's mind when discussing whether or not and for which reasons to shift into the *User-Provided Networking* paradigm. The justifications included herein come in the context of our proposal: the *Offer Nothing - Gain Something* scheme.

What is the main challenge here?

The major advantage of *User-Provided Networks* is that once deployed they will provide ubiquitous connectivity, at least to densely populated areas. One such example may be vehicle communications, or even open-air environments, such as streets, parks and cafes within cities. In such environments, the degree of success of the UPN scheme comes with the density of access points, or in other words with the number of *micro-providers*. That said, the larger the number of end-users that share their broadband Internet connection, the better the quality of service that users are provided with when they are mobile. So, *the main challenge here is to give the appropriate incentives to the end-users to shift into the User-Provided Networking scheme.* In essence, we need to guarantee that the home-user sees no difference at his home-Internet speed, while as a guest-user he receives acceptable service, quality-wise.

Why would a home-user share resources with unknown, mobile users?

The main motive here is that users joining the UPN scheme will be liable to have unlimited connectivity when mobile. Registered users will be provided with a username and password in order to avoid giving access to users that do not belong to the UPN community. Many users nowadays use a laptop or 3G (and beyond) mobile phone devices which provide them a wide range of new mobile-phone applications (e.g., music streaming and sharing [15], latest news broadcast, GPS navigation, social networking platforms [11], [14]). The owner of a broadband connection can use these services for free when he is out of home or office.

What's the impact on the home-user's network performance?

As roughly explained before and further elaborated later on in this paper, sharing the Internet connection with mobile-users, using the proposed algorithm, has statistically no impact on the home-user's performance. The guest-user exploits *only* a portion of the bandwidth that is *not* utilized by the *micro-provider*. Therefore, in essence, the *micro-provider offers nothing and gains something*. This *"something"* is given to the home-user in terms of Internet connectivity when acting as a mobile, roaming user (i.e., when he is out of home or office).

What is the guest-user's quality of experience?

Obviously, the available bandwidth that the mobile-user can exploit depends largely on the home-user's network usage. As a bottom line, the mobile-user's network performance is sometimes low and/or intermittent, but still enough to download web-pages, e-mails or maps to his mobile device. As UPNs evolve and more users participate in the UPN communities, this performance is expected to improve accordingly.

Which are the alternative proposals and how do they operate?

- **FON Community** [18]: FON is the biggest connectivity-sharing community around the world. According to the FON scheme, home-users allocate explicitly and at all times a portion of their bandwidth to mobile users. The success and growth of this connectivity scheme shows that end-users are indeed interested in sharing their broadband connection in order to be granted access when they are out of home or office. Here, we focus on the FON package provided by BT [29]. Depending on the type of contract, the home-user gets up to 20Mbps for his connection at home and 250 "FON or roaming minutes" when mobile. The maximum speed that the guest-user can have is 512kbps, regardless of the home-user's network usage. In Section 4, we compare our proposed algorithms with the BT FON scheme and discuss its advantages and disadvantages.

- **OpenSpark Community** [19]: According to OpenSpark, there are no limits on the resources that the guest-user can exploit and there is no extra charge, or allowed connectivity time (e.g., "FON minutes") for the mobile-user. That is, the mobile-user can enjoy the same services, resource-wise, both when at home and when mobile. However, according to this scheme, the mobile-, guest- or visitor-users are added manually. In particular, if a mobile-user is a member of the community, then when he/she moves to a cafe, hotel or house of another OpenSpark community member, he/she is allowed to access the Internet. Clearly, this scheme is not so flexible in giving access to mobile, roaming users. This approach is suitable for users that have a trusting relationship between each other.

- **Wifi.com** [30] (ex-Whisher Community) [20]: This scheme follows a social-networking approach: each member of the community grants access to friends and colleagues at will and free of charge, realizing a so-called *Shared Hotspot*. Furthermore, there are *Commercial Hotspots*, such as hotels, restaurants, airports, where the guest-user has to pay a small, per-minute fee in order to gain access. In all cases, there are no bandwidth limitations for the guest-user.

The success and growth of the above initiatives (already millions of users sharing connectivity, e.g., [29], [18], [19], [30]) clearly shows that users are keen on giving away a portion of their bandwidth in order to have access when out of home or office. We argue that these initiatives have already put the first stone towards *"On Demand Connectivity Sharing"*. There are, however, some issues worth of further skepticism. For example, *"Why would an end-user share his connection if there are no guarantees for his own quality of service when mobile users are connected?"*, or *"Do the above approaches enable connectivity for mobile users (e.g., roaming on the street, or in a vehicle)?"*. We attempt to answer the above questions below, in the context of our proposal, the *Offer Nothing - Gain Something* scheme. We refer the reader to our earlier work [16] for a detailed categorization of related studies on sharing and security issues.

Why is the proposed approach different and which are its advantages?

In our opinion, the *On Demand Connectivity Sharing* paradigm has to balance between two fragile service-points: (i) the guarantee to the home-user for seamless sharing and (ii) the mobile-user's quality of experience. Requirement (i) is met only by the FON scheme, while requirement (ii) is only partially met by the OpenSpark and the Wifi.com schemes. That is, we argue that the mobile-user should be able to attach to different APs *on-the-fly* in order to benefit from mobile Internet services. In that sense, our proposal comprises an extension/enhancement to the above-mentioned schemes. We base our design on the fact that underutilization of resources in home networks is the rule and not the exception. That given, we highlight that *the proposed scheme does not upper-bound the portion of bandwidth that the guest-user can exploit, provided that sharing is seamless to the home-user in all cases.*

3. Our Design Proposals

3.1. Service Differentiation for UPNs

When joining the UPN community, each user is provided with a `username` and a `password`. This pair of identification is used to access the Internet: if the user connects to his own access point, then he is identified as the home-user or the *micro-provider*, while if he is mobile and connects to an unknown access point, he is identified as the guest-user. In turn, whatever packets are sent or received by that user are classified accordingly. This way, we implement a packet-classification algorithm to (i) discriminate between *home-* and *guest-packets* and (ii) apply the corresponding service differentiation. We call this algorithm *User-Provided Network Queuing* (UPNQ).

3.1.1. User-Provided Network Queuing (UPNQ)

Service differentiation is applied on a per packet basis and depends largely on the queue utilization at the time a packet arrives at the *micro-provider's* UPN-AP. In particular, we implemented a packet scheduling mechanism based on the non-preemptive priority queuing scheme. In our case, the home-packets receive higher priority at all times, which gives a constant performance advantage to the home Internet connection owner. We have shown in [31] that a small rate of packets can be serviced with statistically zero impact on the performance of the flows sharing the same channel. To guarantee that home-users see no performance difference even in the case of traffic-demanding guests, we complement the scheduling mechanism with one extra capability. Whenever the percentage of the home-packets in the queue exceeds a certain threshold, which we call *upnthresh*, the newly-arriving guest-packets are forcefully dropped.

3.1.2. Numerical Analysis of UPNQ

We consider two classes of traffic, the home-traffic, arriving at the bottleneck queue with rate λ_1 and the guest-traffic, arriving with rate λ_2; we make the following assumptions:

1. Traffic from both classes is formed by a large number of flows[2]
2. All packets arriving at the bottleneck queue follow a Poisson distribution[3]. This applies to both home- and guest-user traffic.
3. Class 1 (home-traffic) has full priority over class 2 (guest-traffic).
4. Both classes use 1000-byte packets.

Our main goal is to model the impact of the guest-user's presence on the home-user for different proportions of traffic demand. Based on that, we can confine the guest-user's packet-arrival-rate (λ_2) to a value that has insignificant impact on the home-user. We use a non-preemptive head-of-line priority system per class and an M/D/1 queue. The relation between the home-user's traffic rate, λ_1, the guest-user's traffic rate, λ_2, and the *upnthresh* is as follows:

[2]Although one may contend that guest-traffic is probably not going to consist of large number of flows, due to inherent difficulties of doing so (e.g., a human cannot follow more than one or two applications simultaneously), we note the following: (i) as mobile devices (e.g., smart phones, laptops, iPods) become "smarter" and connectivity-dependent, a large number of processes will require Internet access in regular time-intervals (e.g., software updates), without knowledge or demand of the user himself, and (ii) we envision that future applications, such as social networking platforms for instance, will exchange millions of messages in order to keep the user up-to-date with regard to his social interests and habits (e.g., [14], [13]). That said, although the user himself may trigger a couple of applications only, background processes and applications will demand for many more active flows.

[3]It is widely adopted (e.g., [32], [33], [34], [35]) that the packet arrival process for highly multiplexed environments tends to follow a Poisson Distribution.

- If $\lambda_1 \geq upnthresh$, then all guest packets are forcefully dropped.
- If $\lambda_1 = 0$, then the guest-user can exploit the home-network's bandwidth.
- If $\lambda_1 < upnthresh$, then the analysis provided below applies.

We note that both the home- and the guest-user traffic may consist of more than one home- or guest-users. This is also supported by our first assumption above, that both traffic classes are formed by a large number of flows. We summarize our notation in Table 1.

Symbol	**Description**
λ_1	Arrival rate of class 1
λ_2	Arrival rate of class 2
T_{S1}	Average service-time of class 1
T_{S2}	Average service-time of class 2
$\lambda = \lambda_1 + \lambda_2$	Total arrival rate
$u_1 = \lambda_1 T_{S1}$	Utilization of class 1
$u_2 = \lambda_1 T_{S1} + \lambda_2 T_{S2}$	Cumulative utilization
T_{Q1}	Average queuing delay for class 1
T_{Q2}	Average queuing delay for class 2
T_Q	Average queuing delay
T_{W1}	Average total waiting time (i.e., service plus queuing time) for Class 1
T_{W2}	Average total waiting time (i.e., service plus queuing time) for Class 2
T_W	Total Average waiting time for both Classes
$upnthresh$	Threshold after which guest-traffic is dropped

Table 1: Notation Table

We define the following:

Waiting-Time: Waiting-Time represents the amount of time a packet waits for service in the queue.

Service-Time: Service-Time represents the amount of actual service time required by a packet and is proportional to its size[4].

Time-in-System: Time-in-system equals to the Waiting Time plus Service Time (in our case is the same as Queuing Delay).

[4]Given that both classes use 1000-byte packets, they also have the same average service-time.

The packet-departure rate equals to the service time distribution, because we assume a single server.

We calculate the average waiting time for each of the two classes as follows; recall that in all cases Class 1 (i.e., home-traffic) has full priority over Class 2 (i.e., guest-traffic):

$$T_{W1} = \frac{\lambda_1 T_{s1}^2 + \lambda_2 T_{s2}^2}{2(1 - u_1)} \tag{1}$$

$$T_{W2} = \frac{\lambda_1 T_{s1}^2 + \lambda_2 T_{s2}^2}{2(1 - u_1)(1 - u_2)} \tag{2}$$

Consequently, the total average waiting time equals to the average of T_{W1}, T_{W2} weighted by the arrival rate for each class:

$$T_W = \frac{\lambda_1}{\lambda} T_{W1} + \frac{\lambda_2}{\lambda} T_{W2} \tag{3}$$

We calculate the queuing delay for each class and estimate the total average time-in-system:

$$T_{Q1} = T_{W1} + T_{S1} \tag{4}$$

$$T_{Q2} = T_{W2} + T_{S2} \tag{5}$$

$$T_Q = \frac{\lambda_1}{\lambda} T_{Q1} + \frac{\lambda_2}{\lambda} T_{Q2} \tag{6}$$

In case of zero guest-traffic, $\lambda_2 = 0$, hence, Equations (1) and (4) become:

$$T'_{W1} = \frac{\lambda_1 T_{s1}^2}{2(1 - u_1)} \tag{7}$$

$$T'_{Q1} = T'_{W1} + T_{S1} \tag{8}$$

The purpose here is to calculate the impact of the guest-traffic presence on the home-traffic in terms of average queuing delay. Based on Equations (1), (4), (7), (8), we have:

$$\frac{T_{Q1}}{T'_{Q1}} = 1 + \frac{\lambda_2 T_{s2}^2}{2T_{s1} - \lambda_1 T_{s1}^2} \tag{9}$$

From Equation (9), we observe that the queuing delay of the home-user increases by $k = \frac{\lambda_2 T_{s2}^2}{2T_{s1} - \lambda_1 T_{s1}^2}$, when the guest-traffic rate equals λ_2. Based on that we can control the home-user's queuing delay increment k (due to the guest's traffic) by limiting the guest-user's arrival rate λ_2:

$$\lambda_2 \leq \frac{k(2T_{s1} - \lambda_1 T_{s1}^2)}{T_{s2}^2} \tag{10}$$

Assuming that home- and the guest-packets are the same, size-wise, we can further assume that their average service times, T_{sx}, are also equal (i.e., $T_{s1} \approx T_{s2}$). Therefore, Equation (10) can become:

$$\lambda_2 \leq \frac{k(2 - \lambda_1 T_{s1})}{T_{s2}} \tag{11}$$

In [31], we have shown that when the impact of Class 2's queuing delay to Class 1 (i.e., k) is below 5%, this is, in general, unnoticeable from the Class 1 applications. That said, setting $k \approx 0.01 - 0.05$ (i.e., 1% - 5%) and regulating the rate of Class 2 traffic (here the guest-user, λ_2) according to Equation (10) or (11) will have statistically no impact on the performance of Class 1 traffic (here the home-user, λ_1).

We observe the following:

1. **If Equation (11) holds true, connectivity is shared seamlessly.** Indeed, the home-user will see no difference when the guest-user's rate is kept below the threshold $\frac{k(2-\lambda_1 T_{s1})}{T_{s2}}$.
2. **The guest-user's rate, λ_2, decreases as the home-user's rate increases.** As expected, the guest-user will be restricted from using resources that the home-user intends to exploit. Remember that in all cases,

we need to guarantee that the home-user receives the best possible service. Given that Equation (11) is of the form $y = a - bx$, where $a = \frac{2k}{T_{s2}}$ and $b = k$, we can guarantee that the guest-user is never misbehaving.

3. **The guest-user can benefit from high-speed connectivity if the home-user is absent, or is not making heavy use of his connection.** Again, since Equation (11) is of the form $y = a - bx$, the guest-user is free to exploit all available resources (i.e., λ_2 has the maximum value). As we show later on, alternative schemes, such as BT FON [29], do not provide such an opportunity to their guest-users (i.e., the maximum guest-rate is bounded, even when the home-resources are not utilized, e.g., the home user is absent).

3.1.3. Simulation Validation for UPNQ

We use *ns-2* [36] to illustrate the basic performance of the proposed UPN Queuing (UPNQ) algorithm. The topology is shown in Figure 1. The home-users run five FTP applications each for the whole duration of the simulation, which is 300 seconds, while the guest-users download one long file each (i.e., 50MBs), again through FTP. As shown in Figure 1, guest users are connected through 11Mbps wireless IEEE 802.11g links to the home APs, while home-users connect to the Internet though standard ADSL 8Mbps lines. We simulate increasing number of guest-users, in order to prove the scalability of the proposed approach. The purpose of this congestive scenario is to show that even when the guest-users demand for heavy traffic (i.e., long FTPs) at a time when home-users make full use of their bandwidth, the UPNQ algorithm can still achieve the following: (i) guarantee zero impact on the home-user's Goodput performance (see Figure 2(a)) and (ii) provide acceptable performance for the mobile-user (see Figure 2(c)). We compare UPNQ with simple DropTail to show the benefit of our proposed approach. Using DropTail implies that the UPN-AP is simply left unlocked for guests to use.

At this point we do not compare the proposed algorithm with alternative connectivity sharing schemes (e.g., BTFON), since our target here is to evaluate the performance of different flavors of UPNQ.

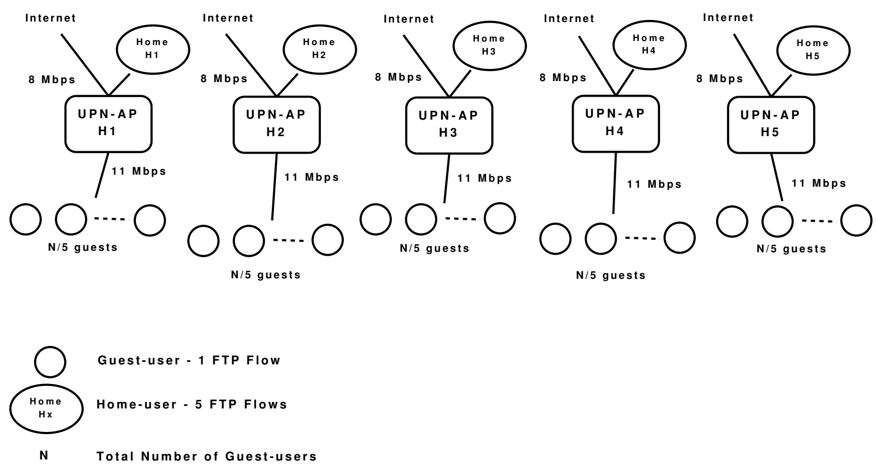
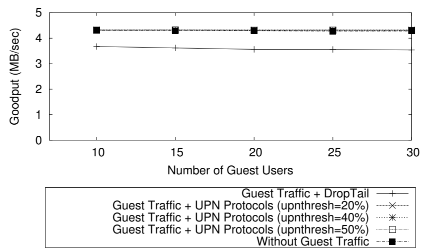


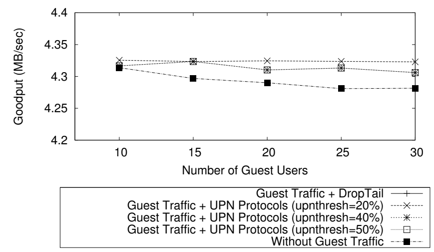
Figure 1: UPN Topology

In Figure 2(a), we see that simply leaving the AP open to guests, using DropTail queuing, can result in more than 10% decrease of the home-users' Goodput (Figure 2(a)).

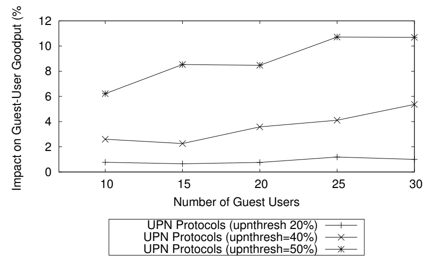
In Figure 2(c), we present the percentage $\frac{UPNQ\ GuestUser\ Goodput}{DropTail\ GuestUser\ Goodput}$. Differently put, 10% performance in Figure 2(c) means that the UPNQ guest-user receives the $(1/10)^{th}$ of the bandwidth he would get if the AP was simply left open (i.e., with DropTail). From Figure 2(c), we observe that a *upnthresh* equal to 20% performs pretty aggressively leaving only a few Bytes per second for the guest-user to download. Instead, *upnthresh* equal to 50% allows for considerable amount of data to be downloaded to the guest's device. In our case this value ranges from 30KBs to almost 70KBs per second per guest-user, depending on the number of the guest-users. Although this bandwidth would result in poor performance for bulk data transfers, this is rarely the case with mobile-users: these users are usually interested in applications such as e-mail, web (e.g., tourist info), or map download/directions, for which, this rate is already enough. We elaborate further on such applications in Section 4. In fact, we show that even a value equal to 20% for the *upnthresh* is enough for these applications, especially when the home-user is not constantly running FTP, which represents a far more realistic scenario.



(a) Home-user Goodput



(b) Home-User Goodput Zoom In



(c) UPNQ against DropTail

Figure 2: Congestive Scenario

The impact of the *upnthresh* value on the performance of the home-user, in terms of Goodput, is shown in Figures 2(a) and 2(b). In Figure 2(a), we show that the impact of the guests-users on the home-users is negligible. A closer look, in Figure 2(b), reveals that there is some slight impact indeed, which increases with the value of *upnthresh*. This impact may be up to 5% in case of 30 guest-users and *upnthresh* equal to 50% (Figure 2(b)). We still consider, however, that this impact is insignificant for the home-user's application, according to the findings in [31]. We also note that, although there is no strict correlation between the value of the *upnthresh* and the number of guest users in the system, it is apparent that the higher the number of guests, the less the share that each of them can exploit. We consider as a future work to explore the optimal value of the *upnthresh*. This setting has to be studied on the basis of the micro-providers' density and the guest-users' demand. We discuss such issues in Section 5.

Our proposal inherently assumes that the bottleneck is at the microprovider's

AP, which is a reasonable argument for today's networks. That is, the DSL connection to the ISP is usually slower than the wired or wireless LAN that spreads further away from the home-router/AP towards the edge of the network (i.e., the last-mile wireless home network; see Figure 1). Therefore, in all cases the guest-user will not cause network overload further away from what the micro-provider's contract allows. In case the bottleneck is at the ISP link, then an adjustment of the incoming/outgoing AP's bandwidth can move the congestion bottleneck back to the UPN-AP. This, together with the fact that UPNQ is designed to provide seamless sharing to the home-user, verifies the *Offer Nothing - Gain Something* functionality included in the *On Demand Connectivity Sharing* framework proposed herein.

3.2. User-Provided Network Load-Balancing

We complement the above service differentiation scheme with a load balancing mechanism; the purpose of the latter is to split guest-traffic among all available UPN-APs (provided the guest-user is within range of more than one UPN-APs, e.g., Figure 3). We call this algorithm *UPN Load Balancing* (UPNLB).

We consider the topology illustrated in Figure 3: three home-users (H_1, H_2 and H_3) provide connectivity to three roaming, guest-users (G_1, G_2 and G_3); each UPN-AP broadcasts the current queue-usage from its corresponding home-user to all guest-users within range in regular time intervals[5]. We assume that the least-utilized UPN-AP is that of home-user H_2. We investigate whether and in which cases it is efficient for both guests G_1 and G_3 to change their point of attachment to H_2. We identify three main approaches:

1. **Guests move to the least-utilized UPN-AP within range with probability** $p = 1$**.** According to this approach, once guest-users receive a better offer from a less-utilized UPN-AP than their current point of at-

[5]The broadcast notification messages consist of a few bytes and cross one hop only, hence, the associated communication overhead is insignificant.

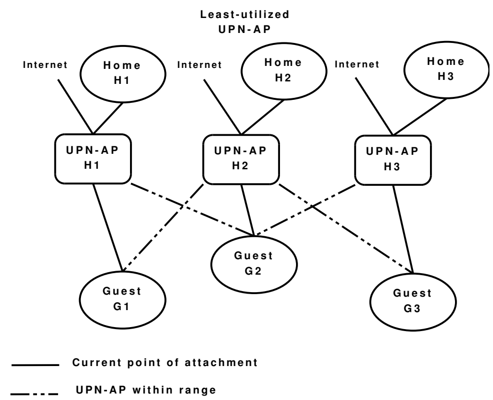


Figure 3: Load-Balancing Topology

tachment, they switch to the new one. In our case (i.e., Figure 3) guests G_1 and G_3 will switch to H_2 with probability $p = 1$, while G_2 will not consider switching, since he did not receive any better offer. Therefore, at the next step, all three guest-users will be hanging from H_2. This is the simplest approach to UPN Load Balancing, which has good chances of providing acceptable quality to mobile-users when the number of such users is relatively low, but would not scale once a large number of roaming-, mobile-users seek connectivity through UPN-APs (e.g., in densely populated areas, such as crowded streets). For example, if all guest-users within range of the least-utilized UPN-AP switch to this specific AP, then this switching would probably overload the otherwise underutilized AP, possibly leading to performance degradation for the guests. To overcome this undesired situation, we attempt to optimize the load balancing algorithm using a probabilistic analysis approach.

2. **Guests move to the least-utilized UPN-AP within range with probability** $p < 1$**.** We model this approach as follows: each guest-user decides whether to switch to a less-utilized UPN-AP with probability $p < 1$; here, we set this probability to 0.5. Users already attached to the least-utilized UPN-AP do not consider switching (e.g., user G_2 in our example). Since each user makes a decision independently of the rest

and without any common knowledge on the existence of other users (i.e., users G_1 and G_3 in our case), we assume that the act of *switching* or *not switching* is a *Binomial Random Variable*, X, with parameters (n, p), where n is the number of users that consider switching (i.e., G_1 and G_3, therefore two in our case) and p is the probability of switching (i.e., here $p = 0.5$). Therefore, the probability that both G_1 and G_3 decide to switch to H_2 is given by $p(i) = \binom{n}{i}p^i(1-p)^{n-i} = \frac{n!}{(n-i)!i!}p^i(1-p)^{n-i}$, where $i = 2$. Based on the above, we get that $P\{X = 2\} = 0.5^2 \cdot 0.5^0 = 0.25$. We consider this to be a more sophisticated approach to load-balancing between mobile-users, than the more "blind" one presented before. Still, however, guest-users do not exploit any knowledge with regard to the perceived load on their current point of attachment. Hence, we go one step further to make decisions on whether to switch or not taking also into account the load of the available UPN-APs. We elaborate on this approach next.

3. **Guests move to the least-utilized UPN-AP with probability equal to the queue-usage at their current point of attachment.** To guarantee that a guest-user does not switch to a different UPN-AP unless there is real need to do so, we implement an algorithm that utilizes information regarding the queue-usage of UPN-APs within range. That is, each UPN-AP advertises its queue-usage to guest-users and each user moves to an alternative point of attachment with probability equal to the queue-utilization at its current UPN-AP (Figure 3). For instance, if the queue-utilization at H_1 is 75%, then G_1 will move to a less-utilized UPN-AP (H_2 in this case) with $p = 0.75$, while if the queue is less utilized (e.g., 25%), then the probability, p, of switching to another UPN-AP becomes 0.25. The cumulative distribution function F of the binomial random variable X is given below. For ease of illustration and implementation, we have chosen four main *queue-utilization* (q_u) thresholds:

q_u at H_3 \ q_u at H_1	25%	50%	75%
25%	0.0625	0.125	0.185
50%	0.125	0.25	0.375
75%	0.185	0.375	0.5625

Table 2: Probability that both G_1 and G_3 move to H_2

$$F(q_u) = \begin{cases} 0.25, & if\ q_u < 25\% \\ 0.5, & if\ 25\% \leq q_u < 50\% \\ 0.75, & if\ 50\% \leq q_u < 75\% \\ 1, & if\ 75\% \leq q_u \leq 100\% \end{cases}$$

The rationale here is that since guest-users have no knowledge regarding the existence of other guest-users that are interested in moving to the least-utilized UPN-AP, they should do so only if the queue at their current point of attachment is overwhelmed. Similarly to the simple computations previously, we present in Table 2 the probability that both G_1 and G_3 move to H_2 UPN-AP; note that G_2 will still be attached to H_2 since this is the least-utilized UPN-AP within range.

As expected, we observe that the probability that all three guests switch to the same UPN-AP (i.e., H_2) increases with the utilization of the alternative UPN-APs (i.e., H_1 and H_3). The efficiency of this approach, however, has to be evaluated taking also into account the utilization of the UPN-AP at H_2. That is, if H_2 is less than 25% utilized, then even if both G_1 and G_3 switch to H_2, this UPN-AP will normally still be able to serve both the guests and the home-user without any performance degradation. In the opposite case, where H_2 is also heavily loaded (75%, say, when q_u at H_1 and H_3 is above 75%) and hence, there is little space for extra guests, we note that the probability that G_2 switches to an alternative UPN-AP is equally big (i.e., 0.75). Therefore, although chances are that

G_1, G_3, or both, will move to H_2, G_2 has equally big chances of moving to either H_1 or H_3. Note that in all cases q_u measures the micro-providers queue-utilization only.

We show through simulation results later on that this simple load balancing algorithm can provide efficient routing services among mobile-, roaming-users. In all cases, we guarantee, through the service differentiation UPNQ algorithm, that the home-user's performance remains intact.

4. Simulation Results

For the first two simulation scenaria included herein (i.e., Sections 4.1 and 4.2), we begin with the simplistic topology depicted in Figure 1. The purpose is to investigate and understand the performance of the proposed framework in a simple topology. Next, for the third and the fourth scenario, we use more realistic topologies; the specifics of these last two topologies are given in Sections 4.3 and 4.4, respectively. We refer to UPNQ and UPNLB algorithms as *UPN Protocols* and we compare their performance with that of: (i) a BTFON [29] user, and (ii) a simple DropTail queuing (i.e., completely open APs). Recall that the BT-FON home-user allocates 512kbps to the guest-user, which in turn means that the guest-user's bandwidth is bounded by that value. Furthermore, according to the OpenSpark [19] and the Wifi.com [30] connectivity sharing approaches, the guest-user is not bounded as for the bandwidth he can use. Therefore, in the following, the performance of DropTail can be assumed to represent these connectivity-sharing schemes. The *upnthresh* value for UPNQ is set to 20% for the purpose of our simulations herein, unless explicitely stated otherwise; as shown earlier in Section 3.1.3, this is the worse-case scenario for UPNQ.

We note that the BTFON, OpenSpark and Wifi.com connectivity schemes were not originally designed for mobile environments. Here, we evaluate their performance comparatively to our design proposals in order to illustrate their potential in mobile environments.

4.1. Scenario 1: Home-User Traffic with "On/Off" Periods

In this scenario, we simulate home-users that occasionally stop downloading; the "On/Off" pattern is shown in Figure 4. We use the topology shown in Figure 1; in order to illustrate also the benefits of UPNLB, we consider that all guest-users are within range of all UPN-APs. The goal of this scenario is to unveil the benefits of our Load Balancing algorithm (UPNLB). That is, once a home-user suspends his downloading activities, some of the guest-users should attach to that specific UPN-AP, since resources there remain unexploited.

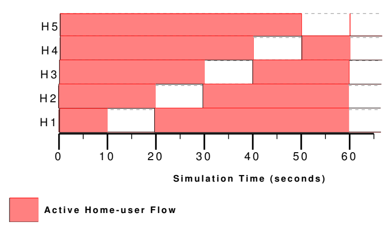
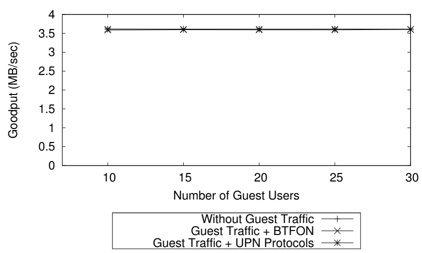
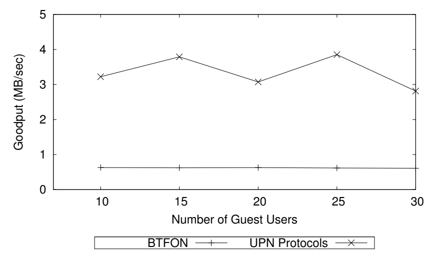


Figure 4: On/Off Home-User Periods (y-axis: house-number from Figure 1)

Indeed, we observe in Figure 5 that in all cases the presence of guest-users is seamless to the performance of the home-users (Figure 5(a)); although the BTFON scheme allocates explicitly 512kbps to the guest-user and therefore, the home-user's resources are reduced by that portion, this is still not noticeable from the home-user (i.e., the allocated bandwidth is less than 2%). The benefit of the guest-user when the proposed UPN Protocols are deployed is clear, compared to the BTFON scheme, as we can see in Figure 5(b). In particular, the UPNQ algorithm guarantees that the guest-user does not over-exploit bandwidth from the home-user, while the UPNLB algorithm is responsible for choosing the least-utilized UPN-AP, in order to take advantage of available resources. This way, the guest-user is always better off, as we have also proved in Section 3.2. The fluctuation of UPN Protocols' performance owes to the following fact: according to our scheme, UPN-APs advertise their queue-utilization every two seconds. In turn, guest-users switch according to the switching properties of our UPN Protocols, presented in Section 3. Now, in case one of the home-users suspends his transfer at the time when some of the guest-users have just been hooked to a nearby AP, these guests' performance will obviously not benefit as much as they would if they "knew" about this free AP. This results to reduced performance for the UPN Protocols as observed in Figure 5(b). Even in this case though, our proposed solutions outperform the BTFON approach to connectivity sharing. We also note that this is a worse-case scenario for UPNQ, since the *upnthresh* is set to 20%. Furthermore, we simulated a large number of guest-users (i.e., up to 30) which is a realistic scenario (e.g., in the center of a busy/touristic city). That given, we claim that the proposed solutions scale well with regard to the number of roaming users.



(a) Home-User Goodput



(b) Guest-User Goodput

Figure 5: Home-User Traffic with "On/Off" Periods

4.2. Scenario 2: Gradually Decreasing Home-User Traffic

To illustrate the scalability properties of UPN Protocols, we simulate a scenario, where the home-users gradually suspend their transmissions. The corresponding suspension pattern is shown in Figure 6. Our target here is to illustrate the scalability properties of UPN Protocols, when large amounts of network resources become available.

We begin with a small number of guest-users in order to highlight the functionality of UPNLB. In particular, we see in Figure 7(b) that UPN Protocols outperform DropTail, when the number of guest-users is less than four. This may sound contradicting, since by default simple DropTail queuing (i.e., a wide-open AP) provides no guarantees to the home-user and hence, allocates more resources to the guest. The reason is as follows: DropTail flows attach and stay

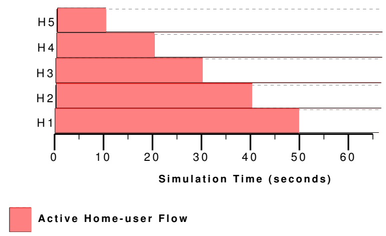
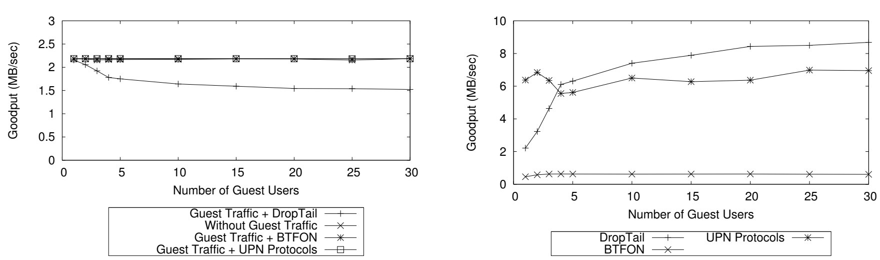


Figure 6: Decreasing Home-user Traffic Pattern

attached to a specific home-user regardless of the network resources available elsewhere; in our case guest-user G1 attaches to home-user H1, guest-user G2 attaches to home H2 etc. Thus, these flows take advantage of free resources only after the 50^{th} and 40^{th} second of the simulation, respectively. In contrast, UPNLB takes advantage of the available resources (i.e., home H5, H4 etc., according to Figure 6) and changes the guests' point of attachment accordingly. This guarantees the best possible service for the roaming users, as we see in Figure 7(b).

Once the number of guest flows increases to more than four, we see that DropTail becomes greedy and provides more resources to guests (Figure 7(b)), but with severe impact on the performance of home-users (up to approximately 30%, according to Figure 7(a)). In all cases, BTFON and UPN Protocols guarantee seamless connectivity sharing for the home-users (Figure 7(a)), while BTFON can provide only poor service to the roaming user (Figure 7(b)).



(a) Home-User Goodput

(b) Guest-User Goodput

Figure 7: Gradually Decreasing Home-Traffic

4.3. Scenario 3: Mobility Topology 1 - UCL to British Museum

In this scenario, we simulate a realistic mobility setup. We have gathered from [29] detailed information regarding the exact position and the number of BTFON users in the city center of London, which we parsed into the *ns-2* simulator. Based on this information, we simulate guest-users walking a 553-meter distance, from the Electrical Engineering Department of UCL to the British Museum. The topology is shown in Figure 8. This setting could represent a guest-user attempting to attach a photo to an e-mail, or a video to a social-networking site, or download a map or touristic info file to his device. We investigate three different instances of this scenario: Figure 9, where we increase the number of mobile-users (file size is 10MB and moving speed is 1m/s); Figure 10, where we vary the size of the file under transmission (moving speed is again 1m/s and we simulate five mobile users); and Figure 11, where we increase the speed of the mobile users (file size is set to 10MB and we simulate five mobile users).

In Figure 8, we show all the available BTFON APs. Our walking users do not connect to all APs shown in Figure 8, but only to those that are within range according to the data gathered from [29]. Similarly to the previous experiments, we compare the performance of the proposed UPN Protocols with DropTail (i.e., completely open APs) and with the BTFON scheme. To represent a realistic situation, we simulate home-users that transfer a 100MB file at random times and then suspend. We measure the time that guest-users need in order to complete their task (i.e., transfer their file through the available APs); we call this time the *Task Completion Time* (TCT).

We assess the performance of the home- and the guest-users for increasing number of guest-users. The goal is to evaluate the scalability properties of the candidate connectivity sharing approaches under realistic mobility conditions. In Figure 9(a), we observe (as expected) that the least aggressive algorithm with regard to the guest-user's traffic (i.e., DropTail) outperforms BTFON and the UPN Protocols. BTFON on the other hand, as the most aggressive approach against guest traffic increases TCT for the guest-user. In Figure 9(b), we see

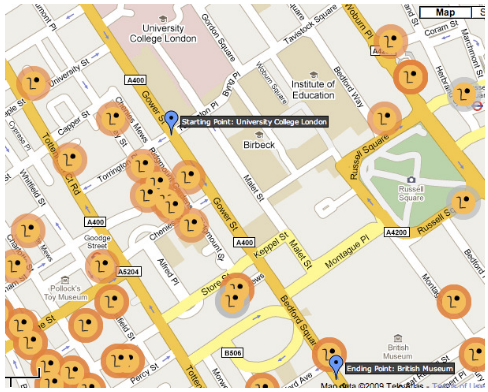
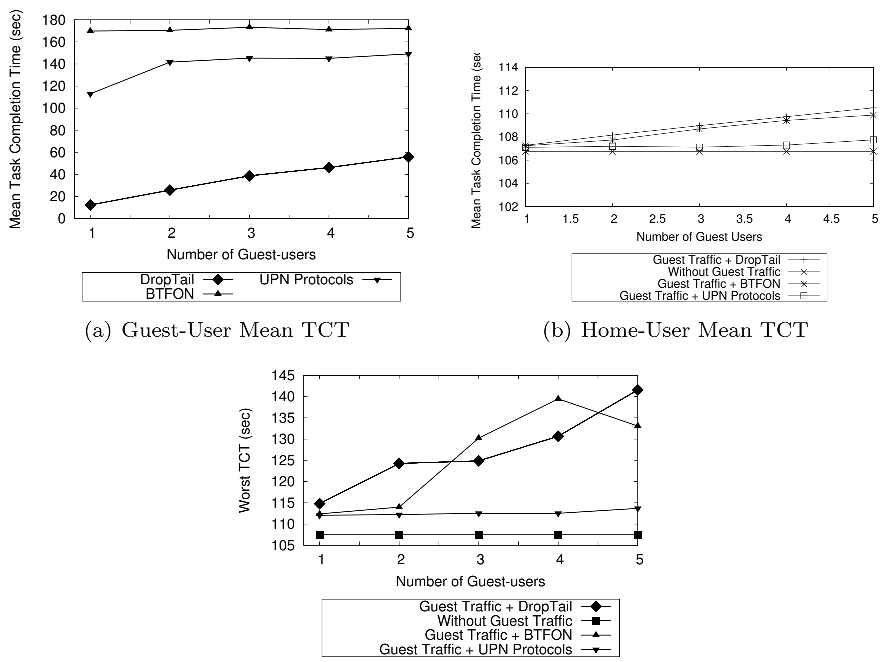


Figure 8: Mobility Topology 1: UCL to British Museum

that both DropTail and BTFON impact the performance of the home-user, while UPN Protocols perform nearly optimal (i.e., very close to the case where there is no guest-user). An interesting observation regarding the scalability properties of the protocols comes from Figure 9(c), where we present the home-users' worse-case TCT. In particular, we observe that both BTFON and DropTail, due to their inflexible transmission patterns, severely impact the performance of some home-users. Instead, the Load Balancing mechanism of UPN Protocols will choose and transmit the guests' traffic through the least-utilized UPN-AP within range. This feature together with the UPN Queuing algorithm, which occasionally becomes aggressive against the guest-users if conditions indicate so, guarantees smooth performance for the home-user, regardless of the number of guests (Figure 9(c)).

We go one step further to investigate the scalability properties of the algorithms with regard to the guest-user's file size. In particular, we simulate one walking guest-user and we vary the amount of data to be transferred. The results are shown in Figure 10.

The performance problems of the inflexible and non-adjustable flat-rate of the BTFON scheme (i.e., 512Kbps) are made clear in this scenario. We see in Figure 10(a) that the BTFON guest-user may need up to 550 seconds to transfer a 30MByte file, while the proposed UPN Protocols complete the task



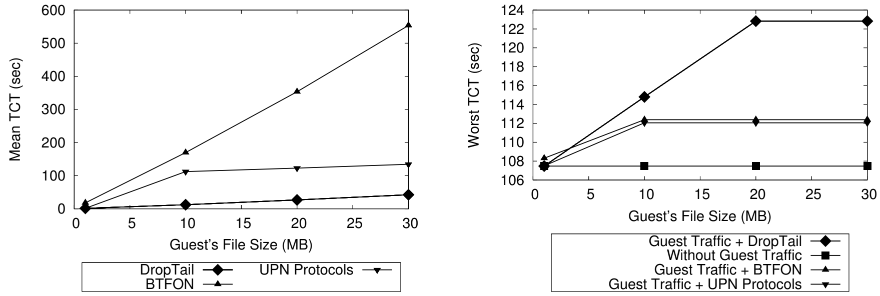
(a) Guest-User Mean TCT (b) Home-User Mean TCT

(c) Home-User Worst TCT

Figure 9: Mobility Scenario 1 - Increasing Number of Guests

in less than 150 seconds. As expected, the DropTail approach is even faster (i.e., DropTail needs 50 seconds at most to complete the task), but this comes at the cost of the home-user's performance. In Figure 10(b), for example, we see that this performance degradation for the home-user may reach up to 14%.

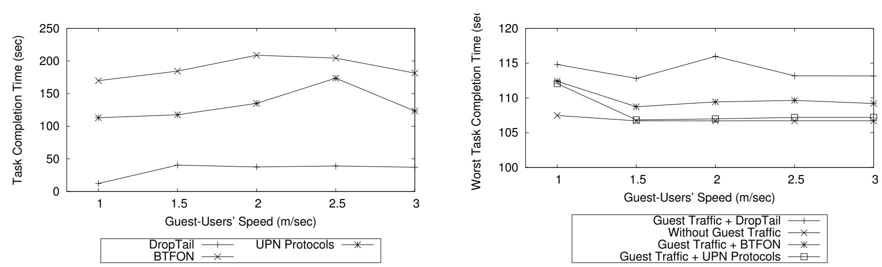
Finally, we vary the speed of the walking user from 1m/sec (3.6km/hour) to 3m/s (10.8km/hour), the average walking speed for humans being 5km/hour. We simulate one guest-user, whose task is to transfer a file of size 10MB. The difference between the BTFON and the UPN Protocol guest-user is now in the order of one minute, which we consider to be significant for mobile settings such as the ones presented here (Figure 11(a)). Again, DropTail is the fastest to complete the guest's task, but again with an impact of around 8% at least on the performance of the home-user (Figure 11(b)). In contrast, the UPN Protocols present near optimal performance with regard to the impact on the



(a) Guest-User Mean TCT (b) Home-User Mean TCT

Figure 10: Mobility Scenario 1 - Increasing File Size

home-user's TCT (Figure 11(b)). This owes to the UPN Protocols' ability to identify and transmit data through the most appropriate, utilization-wise, UPN-AP. In this scenario, for example, our traces show that the UPN Protocol guest-user transmits most of the file near the end of the route, since the protocols found more free space for transmission in the UPN-APs located there.



(a) Guest-User Mean TCT (b) Home-User Worst TCT

Figure 11: Mobility Scenario 1 - Increasing Speed of Guests

4.4. Scenario 4: Topology 2 - Pentonville Road

In order to further evaluate the validity of our claims, we randomly pick a road in central London and repeat the previous simulations. The distance from the starting to the ending point is now 1,217 meters to allow for more diverse connectivity conditions; the topology is shown in Figure 12.

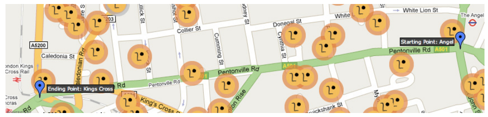
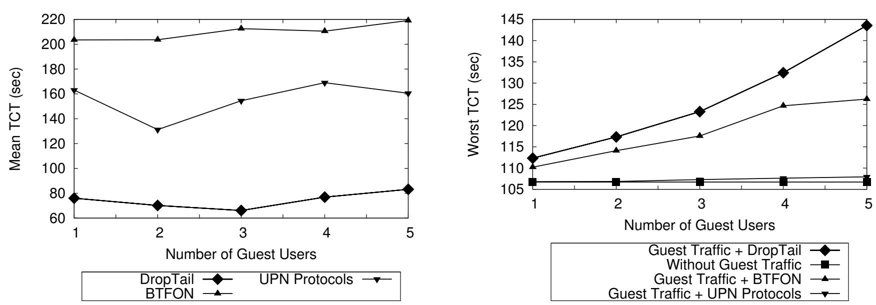


Figure 12: Mobility Topology 2: Pentonville Road

We observe that the performance difference between UPN Protocols and the alternatives now widens even more (Figures 13, 14 and 15). For example, in case of a group of mobile-users the average Task Completion Time is approximately 25% lower for UPN Protocols than for the BTFON user, as we see in Figure 13(a). At the same time, the impact to the home-user is negligible for UPN Protocols, even when considering the worst-case performances (i.e., Figure 13(b)). In contrast, in Figure 13(b), we observe that the home-users that share their connection with strangers under the BTFON or the simple DropTail schemes may occasionally experience severe performance degradation, which may be up to 30-40%, respectively (Figure 13(b)).

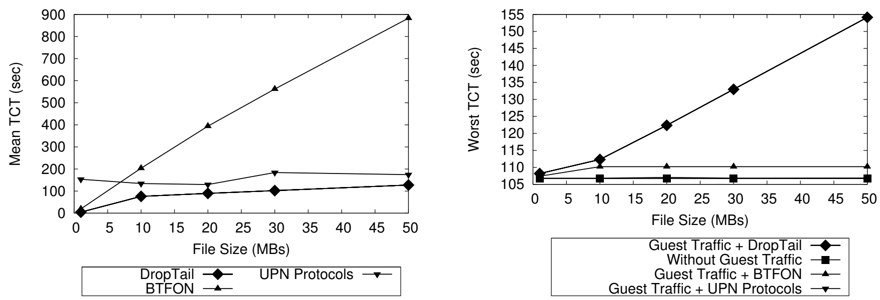


(a) Guest-User Mean TCT

(b) Home-User Worst TCT

Figure 13: Mobility Scenario 2 - Increasing Guests

Increasing the size of the file under transmission, we observe in Figure 14(b) (similarly to the previous experiment, i.e., Figure 10) that although the BTFON scheme does not impact the performance of the home-user, it fails to provide acceptable services to the mobile, roaming guest. In Figure 14(a), for example, we see that the BTFON user may need up to 10 minutes to download/upload a 30MByte file; UPN Protocols, on the other hand, exploit available resources in the most efficient manner and complete the transmission in less than 3 minutes. Our packet-level traces, which we do not present here due to space limitations, show that the UPNLB algorithm always switches the guest to empty UPN-APs, if any, or to the least-utilized ones, otherwise. In all cases, UPN Protocols preserve the quality of experience for the home-user (Figure 14(b)).



(a) Guest-User Mean TCT (b) Home-User Worst TCT

Figure 14: Mobility Scenario 2 - Increasing File Size

The performance gain of our proposed Load-Balancing mechanism is made clear in our final simulation scenario, where we vary the speed of the roaming users (Figure 15). The speed-range simulated here varies from 1m/s (i.e., 3.6km/h) for a slow-walking pedestrian to 20m/s (i.e., 72km/h) for a moving vehicle. We observe that in some cases UPN Protocols complete their task even faster than DropTail, which explicitly owes to the UPNLB algorithm. That is, DropTail guests get hooked to the first AP within range and stay attached to that specific one till they complete their task or lose connectivity. In contrast, UPN Protocols constantly probe for less-utilized UPN-APs and switch once a better offer appears (Figure 15).

Although the results of this particular scenario cannot be generalized, since clearly connectivity times/points depend on the specific setting and the availability of APs, we still claim that the UPN Protocols' increased performance is due to the efficient handling of the UPNLB load-balancing mechanism introduced earlier on. Here, for example, we observe that with an average speed of 10-20m/s, the mobile users reach a "connectivity island" (i.e., a spot with many available APs) faster than a walking user and complete their tasks before connectivity is lost again. Of course, moving faster means that mobile-users will exit the APs' range faster. Here, however, we simulate transmission of a 10MByte file, which seems to be small enough in order to be transmitted within the first connectivity island.

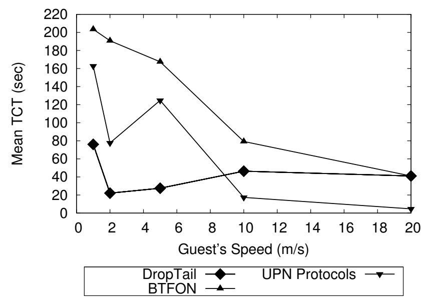


Figure 15: Mobility Scenario 2 - Increasing Speed

5. Open Issues

In this section, we discuss open issues related to our proposed UPN Protocols and their potential extensions. The target is to identify important future challenges for the *"On Demand Connectivity Sharing"* framework proposed previously.

5.1. UPN Queuing Algorithm

We have seen that different values for the *upnthresh* value affect the performance of both the guest- and the home-user (e.g., Figure 2). A detailed investigation on that direction can conclude on a sophisticated algorithm to adjust the value of *upnthresh* based on specific criteria. These criteria can be designed based on: (i) *mac-layer contention*: the higher the contention, the lower the value of *upnthresh*, in order to guarantee seamless sharing for the home-user, (ii) *the application running on top*: some non-congestive applications may get priority even though they come from the guest-user, and (iii) *the* *signal strength*: the lower the signal strength, the higher the *upnthresh* in order to provide acceptable performance to the guest-user. An interesting first step on that direction has been made in [37].

Finally, the traffic arrival process at the bottleneck queue is considered to be Poisson herein. Although this is a widely accepted assumption (e.g., [32], [33], [34], [35]), alternative arrival distributions have to be considered, e.g., hyper-exponential, in order to further verify the results presented in this paper.

5.2. UPN Load Balancing Algorithm

1. Provided that guest-users are mobile-, roaming-users for the most part, they will need to switch/handover to the different UPN-APs due to signal fading [38]. Load balancing based on the signal level as well, instead of the queue utilization only, is not investigated in the present study. For example, at some point the moving user should switch to the next available UPN-AP, regardless of the queue-utilization advertisement. This may raise scalability issues. However, we argue that such issues need to be investigated together with different mobility patterns (e.g., pedestrians vs vehicles), which are beyond the scope of this paper.
2. Related to the above point, the moving pattern of mobile users can be exploited as well. Research on Delay-/Disruption-Tolerant Networks has already considered such issues. For example, studies such as [39], [40], [41] have already exploited the random, or non-random node mobility to connect to neighbor nodes. Similar issues can be investigated for UPNs to choose the most appropriate UPN-AP, based on the mobile-user's direction/mobility pattern.
3. The distribution of home- versus mobile-users for given time-windows throughout the working day would be another interesting topic for investigation. In other words, the social behaviors of users with regard to their in-door versus out-door activities and the corresponding utilization of network resources is considered essential. This study will also unveil

the properties of the business and cost plan that will have to be integrated into the *On Demand Connectivity Sharing* scheme.

4. Given that guest-users receive lower priority compared to home-users and in order to increase the guests' quality of experience, switching between UPN-APs may have to be done on the basis of the application running on top. That is, delay-sensitive, non-congestive guest-applications may have to be routed through different paths than non-elastic, congestive applications. Furthermore, the mobile device may be constrained with regard to energy, hence, faster response with less overhead may be of greater importance than throughput. We plan to investigate and evaluate *proactive service differentiation* for challenged, intermittently connected environments and UPNs along the lines of [42].

6. Conclusions

We have introduced the concept of *"On Demand Connectivity Sharing"*, according to which home-users can share their connection seamlessly with guest-, roaming users. We have built our framework on top of the recently proposed *User-Provided Networking* [17] connectivity paradigm. We proved and showed that both in theory and in practice careful design can guarantee seamless resource sharing for the home-user, while the guest can still enjoy acceptable performance. Our proposed UPN Queuing (UPNQ) and UPN Load Balancing (UPNLB) algorithms comprise the first milestone in a series of topics that warranty further investigation.

References

[1] T. Halonen, R. Romero, J. Melero, GSM, GPRS and EDGE performance: evolution towards 3G/UMTS, Wiley, 2003.

[2] H. Holma, A. Toskala, Wcdma for Umts, Wiley New York, 2000.

[3] J. Bernsen, D. Manivannan, Routing Protocols for Vehicular Ad Hoc Networks That Ensure Quality of Service, in: Proceedings of International

Conference on Wireless and Mobile Communications, Vol. 0, IEEE Computer Society, Los Alamitos, CA, USA, 2008, pp. 1–6.

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

[5] S. Corson, J. Macker, Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations, RFC 2501 (January 1999).

[6] I. F. Akyildiz, W.-Y. Lee, K. R. Chowdhury, Crahns: Cognitive radio ad hoc networks, Ad Hoc Networks 7 (5) (2009) 810 – 836. doi:DOI: 10.1016/j.adhoc.2009.01.001.

[7] I. Akyildiz, X. Wang, A survey on wireless mesh networks, Communications Magazine, IEEE 43 (9) (2005) S23–S30.

[8] S. Androutsellis-Theotokis, D. Spinellis, A Survey of Peer-to-Peer Content Distribution Technologies, ACM Comput. Surv. 36 (4) (2004) 335–371.

[9] K. Fall, A Delay-Tolerant Network Architecture for Challenged Internets, in: Proceedings of SIGCOMM '03, ACM, New York, NY, USA, 2003, pp. 27–34.

[10] I. Psaras, L. Wood, R. Tafazolli, Delay-/Disruption-Tolerant Networks: State of the Art and Research Challenges, available at: http://www.ee.ucl.ac.uk/ uceeips/dtn-survey-ipsaras.pdf (2009).

[11] J. Scott, P. Hui, J. Crowcroft, C. Diot, Haggle: A Networking Architecture Designed Around Mobile Users, in: Proceedings of WONS 2006, Invited paper, Les Menuires, France, 2006.

[12] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, C. Diot, Pocket-Switched networks and human mobility in conference environments, in: Proceedings of the 2005 ACM SIGCOMM Workshop on Delay-tolerant networking (WDTN '05), ACM, New York, NY, USA, 2005, pp. 244–251.

[13] P. Hui, J. Crowcroft, E. Yoneki, Bubble Rap: Social-based Forwarding in Delay-Tolerant Networks, in: MobiHoc '08: Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing, ACM, New York, NY, USA, 2008, pp. 241–250.

[14] A. J. Mashhadi, S. B. Mokhtar, L. Capra, Habit: Leveraging Human Mobility and Social Network for Efficient Content Dissemination in DTNs, in: Proceedings of WoWMoM09, Kos, Greece, 2009.

[15] L. McNamara, C. Mascolo, L. Capra, Media sharing based on colocation prediction in urban transport, in: Proceedings of MobiCom '08, ACM, New York, NY, USA, 2008, pp. 58–69.

[16] L. Mamatas, I. Psaras, Incentives and Algorithms for User-Provided Networks, in: Proceedings of the ACM SIGCOMM Workshop on Home Networks (HomeNets) 2010, 2010.

[17] R. Sofia, P. Mendes, User-provided networks: consumer as provider, IEEE Communication Magazine 46 (2008) 86–91.

[18] FON Community, http://www.fon.com.

[19] OpenSpark Community, http://open.sparknet.fi.

[20] Whisher Community, http://www.whisher.com.

[21] I. Stoica, S. Shenker, H. Zhang, Core-Stateless Fair Queueing: Achieving Approximately Fair Bandwidth Allocations in High Speed Networks, in: SIGCOMM 98, 1998, pp. 118–130.

[22] R. Pan, B. Prabhakar, K. Psounis, CHOKe - A Stateless Queue Management Scheme for Approximating Fair Bandwidth Allocation, in: IEEE INFOCOM, 2000.

[23] M. Claypool, R. Kinicki, A. Kumar, Traffic sensitive active queue management, in: Proceedings of the 8th IEEE Global Internet Symposium, IEEE, Miami, Florida, 2005.

[24] J. Brendel, C. Kring, Z. Liu, C. Marino, World-wide-web server with delayed resource-binding for resource-based load balancing on a distributed resource multi-node network, uS Patent 5,774,660 (June 1998).

[25] B. Godfrey, K. Lakshminarayanan, S. Surana, R. Karp, I. Stoica, Load balancing in dynamic structured P2P systems, in: IEEE INFOCOM, Vol. 4, 2004, pp. 2253–2262.

[26] P. Pham, S. Perreau, Performance analysis of reactive shortest path and multi-path routing mechanism with load balance, in: Proceedings of IEEE Infocom, Vol. 1, Citeseer, 2003, pp. 251–259.

[27] Y. Ganjali, A. Keshavarzian, Load balancing in ad hoc networks: single-path routing vs. multi-path routing, in: IEEE Annual Conference on Computer Communications (INFOCOM), Citeseer, 2004.

[28] M. Marina, S. Das, On-demand multipath distance vector routing in ad hoc networks, in: Proceedings of IEEE International Conference on Network Protocols (ICNP), Vol. 1, Citeseer, 2001.

[29] BT FON Community, http://www.btfon.com/.

[30] Wifi.com, http://www.wifi.com/.

[31] L. Mamatas, V. Tsaoussidis, Differentiating Services with Noncongestive Queuing (NCQ), IEEE Transactions on Computers 58 (5) (2009) 591–604.

[32] S. Fred, T. Bonald, A. Proutiere, G. Régnié, J. Roberts, Statistical Bandwidth Sharing: A Study of Congestion at Flow Level, Proceedings of the 2001 SIGCOMM conference 31 (4) (2001) 111–122.

[33] L. Massoulié, J. Roberts, Bandwidth Sharing and Admission Control for Elastic Traffic, Telecommunication Systems 15 (1) (2000) 185–201.

[34] A. Kherani, A. Kumar, Performance Analysis of TCP with Non-persistent Sessions, in: Workshop on Modeling of Flow and Congestion Control, Ecole Normale Superieure, Paris, France, 2000.

[35] G. de Veciana, T. Konstantopoulos, T.-J. Lee, Stability and Performance Analysis of Networks Supporting Elastic Services, IEEE/ACM Transactions on Networking 9 (1) (2001) 2–14.

[36] S. McCanne, S. Floyd, NS-2 Network Simulator, http://www.isi.edu/nsnam/ns (1997).
URL `http://www.isi.edu/nsnam/ns`

[37] A. Subramanian, P. Deshpande, J. Gaojgao, S. Das, Drive-By Localization of Roadside WiFi Networks, in: Proceedings of Infocom '09, 2008, pp. 718–725. doi:10.1109/INFOCOM.2008.122.

[38] R. Prakash, V. Veeravalli, Adaptive Hard Handoff Algorithms, IEEE Journal on Selected Areas in Communications 18 (2000) 2456–2464. doi:10.1109/49.895049.

[39] W. Zhao, M. Ammar, E. Zegura, A message ferrying approach for data delivery in sparse mobile ad hoc networks, in: MobiHoc '04: Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing, ACM, New York, NY, USA, 2004, pp. 187–198. doi:http://doi.acm.org/10.1145/989459.989483.

[40] Q. Li, D. Rus, Sending messages to mobile users in disconnected ad-hoc wireless networks, in: MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking, ACM, New York, NY, USA, 2000, pp. 44–55. doi:http://doi.acm.org/10.1145/345910.345918.

[41] W. Zhao, M. Ammar, E. Zegura, Controlling the mobility of multiple data transport ferries in a delay-tolerant network, in: in IEEE INFOCOM, 2005.

[42] I. Psaras, N. Wang, R. Tafazolli, Six years since first dtn papers. is there a clear target?, in: 1st Extreme Conference on Communications (ExtremeCom), 2009.