Experimenting with P2P traffic optimization for Wireless Mesh Networks in a federated OMF-PlanetLab environment

Giovanni Di Stasi*, Roberto Bifulco*, Francesco Paolo D'Elia *, Stefano Avallone*, Roberto Canonico*,
Apostolos Apostolaras†, Nikolaos Giallelis†, Thanasis Korakis†, Leandros Tassiulas†
*Università di Napoli Federico II, Dipartimento di Informatica e Sistemistica,
Via Claudio 21, 80125, Napoli - Italy
†Department of Computer and Communication Engineering, University of Thessaly,
Centre for Research and Technology Hellas (CERTH), Volos, Greece

***Abstract*—The ultimate success of the Wireless Mesh Network paradigm (WMN) in large scale deployments depends on the ability to test it in real world scenarios. A typical application scenario which is worth to be investigated in such a context is peer-to-peer traffic management. The creation of large scale testbeds for evaluating wireless mesh technologies and protocols, and for testing their ability to support real world applications in realistic environments, is then a crucial step. OMF (*cOntrol and Management Framework*) is a well-established control, measurement, and management framework for wireless testbeds. In this paper we present how we integrated an OMF-based wireless testbed in the planetary-scale PlanetLab testbed, making it possible for PlanetLab users to run experiments spanning on both PlanetLab and an OMF-based wireless testbed. In order to demonstrate the usefulness of such an integrated scenario, we tested on it an innovative peer-to-peer traffic optimization technique for the BitTorrent file sharing application. The possibility of running this kind of experiments highlighted several real-world issues which could be investigated thanks to our hybrid experimental scenario.**

I. Introduction

Testing Wireless Mesh Networks in large scale deployments and with realistic traffic loads is of crucial importance [1]. Due to the inherent difficulty of capturing all the relevant aspects of the real behaviour of these systems in analytical or simulation models, research on WMNs has always heavily relied on experimental testbeds [2]. Setting up large-scale wireless mesh testbeds, though, is a difficult and costly process. Moreover, since wireless mesh networks are usually employed as access networks to the Internet, taking into account the complexity of the real Internet is part of the efforts to be made for a realistic assessment of these networks. To this purpose, we reckon that it is important to carefully evaluate the impact of peer-to-peer applications on wireless mesh access networks and their resource management schemes, since nowadays more than 50 percent of the overall Internet traffic is produced by applications of this kind [3].

In this paper we present an architectural integration of geographically distributed OMF-based wireless testbeds in the global scale PlanetLab environment [4]. We show a pratical example of such an architecture by describing the integration of the WiLEE testbed, an OMF-based wireless testbed located in Naples, Italy, with the PlanetLab Europe testbed. OMF (*cOntrol and Management Framework*) is a well-established tool to manage wireless testbeds. Originally developed for the ORBIT wireless testbed at Winlab, Rutgers University, OMF is now maintained and improved by NICTA, the Australia's ICT Research Centre of Excellence, and deployed in several testbeds in Australia, Europe, and in the U.S. [5]. Our system allows the seamless integration of the OMF-resources into the global scale PlanetLab infrastructure, creating a synergic interaction between the two environments. In particular, thanks to our contribution PlanetLab users may run experiments involving resources provided and controlled by the OMF-based wireless testbeds. The contribution we present into this paper is in line with current ongoing efforts towards the so called "federation" of experimental infrastructures, an approach that appears as the most reasonable way to build large-scale heterogeneous testbeds. The problem of heterogeneous testbeds federation is currently under investigations of both the GENI initiative in the U.S. [6] and the FIRE initiative in Europe [7].

Other efforts have been made in order to integrate OMF-based testbeds in PlanetLab, like [8] and [9]. Those approaches differ significantly from ours in the level of integration achieved, as they see the OMF testbed as a single resource, that can be attached to a PlanetLab slice as a whole, i.e. only one experiment at a time is allowed on the OMF testbed. In our approach, as we employ a Scheduler which is able to reserve single nodes and wireless channels of the OMF-based testbed for slices, multiple concurrent experiments are possible, which improves significantly the utilization of the OMF-based wireless testbed.

We also illustrate how we used the integrated testbed setup to conduct an experiment aimed at evaluating a peer-to-peer traffic optimization technique for the BitTorrent file sharing application. This is a typical distributed experiment in the PlanetLab wired environment, but in our case it involves the usage of a wireless mesh as an access network, which would not be possible in the plain PlanetLab environment.

The rest of the paper is organized as follows. In section II we

978-1-61284-254-7/11/$26.00 ©2011 IEEE

briefly discuss the problem of peer-to-peer traffic optimization and present a traffic optimization scheme for the bittorrent file sharing application. In section III we describe the OMF testbed management framework and the NITOS Scheduler component developed at CERTH. In section IV we describe the integration steps that we developed to allow for distributed experiments involving both PlanetLab and geographically distributed OMF-based wireless mesh testbeds. In section V we describe the experiments we performed to evaluate our BitTorrent optimization scheme in a peer-to-peer network whose peers include nodes from the WiLEE testbed and several other nodes located elsewhere in PlanetLab Europe. Finally, in section VI we draw our conclusion on the relevance of our contribution and its potential for future developments.

II. Peer-to-peer traffic optimization

One of the inherent characteristics of peer-to-peer systems is that they build *network overlays* among their peers, and route traffic among them along the virtual links of such an overlay. Peer-to-peer routing decisions are made at the application layer, independently of Internet routing and ISP topologies. Hence, overlay routing decisions collide with those made by underlay routing, i.e. ISP routing decisions [10]. As a consequence of such a dichotomy, several inefficiencies may result. For instance, it is not uncommon that adjacent nodes of an overlay network are in different ASes. Such a topology arrangement leads to traffic crossing network boundaries multiple times, thus overloading links which are frequently subject to congestion, while an equivalent overlay topology with nodes located inside the same AS could have had same performance. Such a behavior is undesirable for ISPs, also because their mutual economic agreements take into account the volume of traffic crossing the ISP boundaries.

From what we described above, it emerges that overlay routing, and peer-to-peer applications, may benefit from some form of underlay information recovery, or in general from cross-layer information exchange. Aggarval et al. in [11] suggest that such a cooperation would be beneficial for both ISPs and users. When creating an overlay network, the choice of the nodes to be connected, i.e. the network topology, can be done by taking advantage of information from the underlay network. Different strategies have been proposed recently in the literature that attempt to introduce some cooperation between the two routing layers [12]. Given the role of access networks played by wireless mesh networks, it is interesting to experiment with such techniques when peers are attached to different WMNs connected to the Internet. Our contribution in this paper makes such experiments possible.

A. BitTorrent traffic optimization

Here we describe a traffic optimization solution for a BitTorrent file-sharing peer-to-peer system. BitTorrent is used to efficiently distribute files of large size from one or more initial *seeds* to a population of large numbers of downloaders, forming what is referred to as a *swarm*. Files are exchanged in smaller *chunks* that can be individually retrieved. One of the peculiarities of BitTorrent is that downloaders, a.k.a. *leechers* in BitTorrent terminology, also contribute to spread the content to other peers. As soon as a peer obtain all the chuncks of the desired file, it becomes a seed on its own. We have designed and implemented a solution that aims at incentivating traffic exchange in a BitTorrent system between peers that are located within the same Autonomous System. Our solution does not require any modification to the BitTorrent protocols, nor to the application used by end users. The only modified component of a typical BitTorrent system is the *tracker*, i.e. the system that is contacted by peers to obtain a list of other peers to contact, in order to retrieve chunks of the file to download. In our system, the tracker returns to peers a sorted list of peers to be contacted, where the sorting criterion is by-increasing-AS-distance. In other terms, as soon as a peer contacts the tracker, the tracker determines the AS-number associated to the IP address of that peer, and returns a list of peers whose first items are the closest peers in the swarm (in terms of AS distance), while the last items are the furthest peers.

III. The OMF framework

OMF (cOntrol and Management Framework) is a platform which supports the management and the automatic execution of experiments on a networking testbed. An OMF testbed is made of a number of nodes, equipped with a number of wireless interfaces and available for running the users' experiments, and a few software components, providing the infrastructural service needed to govern the system and its configuration. These software components support all the phases of an experiment lifecycle, from the provisioning of resources to the collection of experimental data. The most important component is the *Experiment Controller* (EC), which is also the interface to the user. The EC is fed with an user-provided experiment description and takes care of orchestrating the testbed resources in order to accomplish the required experiment steps. Such an experiment description is a script written in the OEFDL language, a domain-specific language derived from Ruby. The EC interacts with the *AggregateManager*, the entity responsible of managing and controlling the status of the testbed resources.

The EC also interacts with *Resource Controllers* (RCs), components running in each of the testbed nodes. RCs are responsible of performing local configuration steps, e.g. configuring the channels on the WiFi interfaces, and of controlling the applications running in the single nodes.

A. The NITOS scheduler

In a OMF testbed, resources are basically divided in two categories: nodes and spectrum. In its basic form, OMF assigns resources to users following a FCFS strategy: the user supplies an experiment description and the system tries to assign the resources requested by the experiment if they are available. OMF can be customized, though, to support some kind of reservation of resources. In the NITOS and WiLEE testbeds, we adopted an extendend version of OMF which allows the execution of multiple experiments in parallel on the same testbed, by guaranteeing that the requested resources are actually exclusively assigned to each experiment for its entire duration. This is achieved by assigning a different subset of nodes and wireless channels to each user for a specific time interval, through a testbed resource scheduler, the NITOS Scheduler, described in [13]. These subsets are reserved in advance through the Scheduler and the access to them is enforced during experiment time so that users can have access only to the resources, i.e. nodes and wireless channels, they had previously booked.

While allowing the concurrent execution of multiple experiments on the same testbed, the NITOS Scheduler does not support the integration of the testbed resources with other experimental facilities, like PlanetLab. For this reason we decide to extend it, in order to enable the seamless integration of OMF and PlanetLab Europe resources, as explained in the following section.

IV. PLANETLAB AND OMF INTEGRATION

Our main goal has been to integrate a local OMF-based wireless testbed with the global scale PlanetLab Europe infrastructure. The architecture of the integration we envisioned, however, is general, and it allows to integrate any OMF-based wireless testbed in PlanetLab.

PlanetLab is a geographically distributed testbed for deploying and evaluating planetary-scale network applications in a highly realistic context. Nowadays the testbed is composed of more than 1000 computers, hosted by about 500 academic institutions and industrial research laboratories. In the rest of this paper we will refer to PlanetLab Europe, a European-wide testbed which is federated with PlanetLab since 2007. Thanks to this integration, an experiment created in PlanetLab Europe may comprise nodes belonging to PlanetLab, and viceversa.

To run a distributed experiment over PlanetLab, users need to be associated to a *slice*, which is a collection of virtual machines (VMs) instantiated on a user-defined subset of all the testbed nodes. Slices run concurrently on PlanetLab, acting as network-wide containers that isolate services from each other. An instantiation of a slice in a particular node is called a *sliver*.

Thanks to the integration we achieved, PlanetLab users can add to their slices, in addition to PlanetLab nodes, also nodes belonging to the integrated OMF-based wireless testbed, i.e. the WiLEE testbed. In one of our previous works [8], we realized the integration of an OMF-based wireless testbed in PlanetLab Europe, but only one slice at a time was allowed to include nodes of the wireless testbed. In this work, by leveraging and extending the features of the NITOS Scheduler, we allow multiple PlanetLab slices to include resources from the same OMF-based wireless testbed. Each of these slices is given a subset of nodes and wireless channels, resources which have to be reserved in advance by means of the extended NITOS Scheduler. More details about the extensions we made on the NITOS Scheduler are given in the IV-B section.

In the integrated scenario, PlanetLab users can either run isolated experiments on the OMF-based wireless testbed, i.e. experiments which involve only nodes of the OMF-based wireless testbed, or can use the wireless testbed as an access network for the set of *co-located PlanetLab nodes*, i.e. a set of PlanetLab nodes which are in range of wireless transmission with the OMF-based wireless testbed. In the latter case, PlanetLab users have also to reserve a wireless interface from one of the co-located PlanetLab node. The wireless interfaces are the means by which the interconnection on the experimental plane among the co-located PlanetLab nodes and the OMF-based wireless testbed happens.

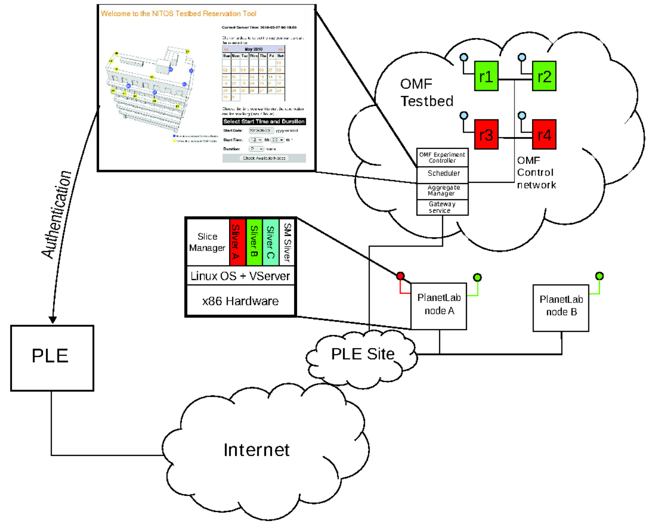

Fig. 1. OMF-PlanetLab integrated architecture.

The architecture we propose is depicted in Fig. 1. It consists of the following elements:

- A PlanetLab site S whose nodes are equipped with one or more WiFi interfaces that allow them to be connected to a local wireless OMF testbed. In the following these nodes are called *PlanetLab Edge Nodes* (PL-Edge Nodes).
- The PlanetLab Europe Central server (PLE), which hosts the information on the PlanetLab Europe testbed, e.g. user accounts, slices.
- The OMF testbed and its components: the Aggregate Manager, the Experiment Controller and the Gateway Service.
- The extended NITOS Scheduler, used to manage the reservation of resources shared through booking.

The Gateway Service is implemented in a Linux box and acts as a *Network Address Translator* (NAT). It is needed for enabling Internet access to the nodes of the OMF-based wireless testbed, whose NICs are assigned private IP addresses.

A. Usage model

In the following we list the sequence of steps needed to execute an experiment using an OMF testbed at site S as access network for PlanetLab.

The experiment is going to be executed over a specific time interval T= $[T\_START, T\_END]$.

1) PlanetLab user U adds one or more PlanetLab Edge Nodes (PL-edge nodes) to his/her slice;
2) U logs into the Scheduler at site S and books the resources (nodes, channels, WiFi interfaces of PL-edge nodes nodes) he needs for his/her experiment over time interval T, providing the slice identifier. According to PlanetLab's resource management scheme, booked resources are actually associated with such slice rather than with the user that performed the reservation;
3) While time is in T, each slice's user is allowed to access the OMF EC (Experiment Controller) to perform his/her experiment involving the OMF resources assigned to him/her.

The procedure for running isolated experiments on the OMF-based wireless testbed is similar and just implies that no wireless interfaces belonging to the co-located PlanetLab nodes are reserved.

B. Implementation details

In order to support the usage model described above, we had to extend the NITOS Scheduler and make some additions to the software which manages PlanetLab nodes.

As seen in the III-A section, the purpose of the NITOS Scheduler is to allow the reservation of resources, i.e. wireless nodes and channels, on isolated OMF-based wireless testbeds. We extended it in order to allow the reservation also of non-virtualized PlanetLab resources, i.e. the WiFI interfaces of the co-located PlanetLab nodes, and in order to authenticate users with their PlanetLab credentials.

In order to support the reservation of WiFi interfaces of the co-located PlanetLab nodes, we first had to modify the NITOS Scheduler, Scheduler in the following, web interface and the database where the reservations are stored. We also had to design and implement a mechanism for allowing the Scheduler to interact with the co-located PlanetLab nodes, in order to actually enforce the association between slices and WiFi interfaces. The process of enforcing the reservations, in details, is the following. Users reserve the WiFi interfaces by means of the Scheduler web interface. Reservations are stored in the Scheduler local database. When a reservation starts or ends, the Scheduler establishes a secure ssh connection with the co-located PlanetLab nodes the reservation refers. This connection is made using the credentials of the *SM Sliver* (Scheduler Management Sliver), i.e. a management sliver we instantiated on the co-located PlanetLab nodes to enable the Scheduler interaction. This sliver has no particular privilege, but the possibility of making the association, or dissociation, of the WiFi interfaces with the users' slices. The association or dissociation is made by exploiting the features of *vsys* [14], a subsystem of PlanetLab which allows to grant to slivers the privileges required for calling a script in the root context of the node, i.e. the privileged context of the node. It is necessary to have access to the root context of the node, because only in that context is possible the execution of privileged operations, like the association or disassociation between interfaces and slices. Hence it is the script in the root context which makes the

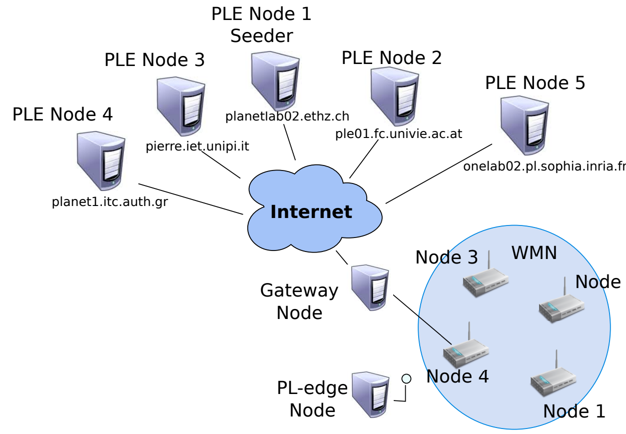

Fig. 2. Experiments setup.

actual association or disassociation, and does that by adding or deleting some *iptables* [15] rules. Such rules state, in case of association, that only packets belonging to the slice which reserved the WiFi interface are allowed to go out through that WiFi interface, and in turn, reach the OMF-based wireless testbed. In case of disassociation, the rules previously added are deleted.

We believe that the architecture chosen for the interaction between the Scheduler and the co-located PlanetLab nodes can be useful in all the circumstances where it is necessary to add a functionality to PlanetLab by means of external entities, e.g. the extended NITOS Scheduler in our case. These entities can interact with PlanetLab nodes by means of secure ssh connections to unprivileged slivers, and then execute privileged operations in a safe way by means of *vsys*. This approach does not require modifications to the *NodeManager*, i.e. the software entity which runs in the root context of PlanetLab nodes and is responsible of the node resources, thus making it easier and safer to add new features to the PlanetLab infrastructure.

When a reservation starts or ends, the Scheduler has also to interact with the *OMF Console*, the Linux machine which hosts the OMF Experiment Controller (OMF EC). The Scheduler creates on this machine an UNIX account for each slice which has nodes reserved on the OMF-based wireless testbed, and associates to this account the public keys of the slice users. It also adds some *iptables* rules in order to allow each slice to access the reserved nodes, and only those. As for the case of interaction with the co-located PlanetLab nodes, secure ssh connections are used.

As previously stated, we also had to modify the authentication procedure of the Scheduler. The modified authentication procedure authenticates users against the *PLE-MA* (PlanetLab Europe Management Authority), i.e. the PlanetLab entity which manages user credentials. To do that, we realized a new authentication plugin to be added to the Joomla-based [16] authentication system of the Scheduler. This plugin checks the user credentials by calling an authentication function on the *PLE-MA API* [17], by means of a secure HTTPS connection.

V. Experiments

In this section we describe an experiment aimed at investigating a problem that is frequently studied on top of PlanetLab, i.e. peer-to-peer traffic optimization. The peculiarity, in our case, is that we create a distributed setup for our experiment involving the use of our wireless mesh testbed as access network to the Internet. In fact, we intend to investigate this problem, and compare its solutions, in the specific context of WMNs, where specific cross-layer approaches can be part of the solution. In this paper we present how we conducted the experiments and the reasons that make our integrated infrastructure useful for evaluating wireless meshes in realistic conditions.

The tests were performed on the WILEE (WIreLEss Experimental) WiFi Mesh Testbed, located in the Computing Department of University of Napoli Federico II. It consists of:

- 5 Netgear WG302Uv1 access points based on the network processor Intel XScale IXP422B at 266Mhz, with 32 Mbyte of DRAM memory and 16 Mbyte of flash disk. Each node is equipped with two 802.11a/g wireless cards. As operating system we employed OpenWrt [18], the well known Linux distribution for embedded devices;
- 1 Linux machine acting as gateway towards the Internet and hosting the *OMF Experiment Controller (OMF EC)*, the extended *NITOS Scheduler* and the *OMF Aggregate Manager*;
- 1 node belonging to a private PlanetLab deployment, i.e. the *co-located PlanetLab node* (or *PlanetLab edge node*) equipped with a IEEE 802.11a wireless card.

	N1	N2	N3	N4	PLEd	PLE-N2	PLE-N3	PLE-N5
Node1(N1)	-	4.4	35.3	3.0	5.1	-	-	-
Node2(N2)	3.6	-	36.4	3.3	4.5	-	-	-
Node3(N3)	-	-	-	-	-	9.5	20.0	18.3
Node4(N4)	5.0	5.9	30.4	-	6.5	-	-	-
PL-Edge Node (PLEd)	6.4	6.0	32.0	3.4	-	-	-	-

TABLE I
TRAFFIC MATRIX FOR AN EXPERIMENT WHERE THE AS-AWARE TRACKER WAS USED. ON THE ROWS THE RECEIVING NODES; ON THE COLUMNS THE SENDING NODES. VALUES ARE IN MBYTES.

	N2	N3	PLEd	PLE-N1	PLE-N2	PLE-N3	PLE-N5
Node1(N1)	-	-	47.7	-	-	-	-
Node2(N2)	-	28.7	-	-	9.2	9.9	-
Node3(N3)	0.1	-	-	47.7	-	-	-
Node4(N4)	-	27.5	-	20.3	-	-	-
PL-Edge Node(PLEd)	-	1.7	-	-	-	-	46.0

TABLE II
TRAFFIC MATRIX WHERE THE STANDARD QUASH TRACKER WAS USED.

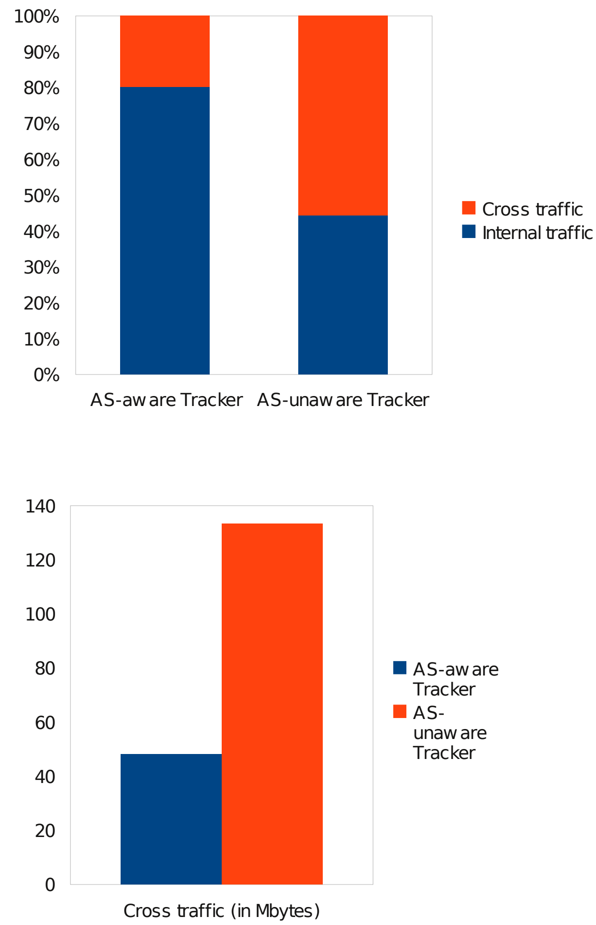

Fig. 3. Bittorrent optimization experiment results.

A. Experimental results

Our experiment is aimed at evaluating our tracker-based solution when a significant fraction of peers are connected to the Internet through the same wireless mesh network. Our objective is to show that in this case, by adopting our strategy, a substantial amount of traffic is reduced through the wireless mesh gateway, i.e. the node connecting the wireless mesh to the wired Internet. To this purpose we created a slice involving five PlanetLab Europe nodes and the PlanetLab node situated at the edge of the WILEE testbed (PL-edge node in the following). To this slice, some bookable resources, i.e. five wireless nodes from the WILEE testbed and the WiFi interface of the PL-edge node, were added by using the extended NITOS Scheduler.

The wireless nodes were configured by using the facility offered by OMF to form a single-channel WMN and provide Internet access to the PL-edge node. A Bittorrent client, `transmissioncli`, was installed on the PlanetLab Europe nodes, on the PL-edge node and on the wireless nodes. One of the PlanetLab Europe nodes was chosen as the seeder of the Bittorrent swarm, which consisted of a file of approximately 50 megabytes. The scenario of the experiments is illustrated in Fig. 2.

We performed a set of experiments by employing alternatively a standard Bittorrent tracker, *quash*, and the same tracker modified by us in order to take into account the ASes (Autonomous Systems) of peers. In particular, the modified Tracker replies to requests with lists in which the first peers are in the same AS of the asking peer. The information about the AS of a peer is got by the tracker at runtime by a request to the RIPE database [19].

Each experiment performed was about the complete download of the files of the swarm by all the peers. At the end of each experiment, when all the peers had downloaded the files, we measured the traffic which belonged to connections which had been either originated or destined to nodes located behind the OMF Gateway, i.e. the wireless nodes and the PL-Edge node. Our objective was to demonstrate that the traffic crossing the WMN boundaries was minimized by using our modified tracker. In Fig. 3 and in tables I and II we report the results of two experiments, one where we employed the standard Quash tracker and the other one where the modified tracker was used. The same two experiments were repeated several times and we got each time very similar results. The figures and the tables show that the amount of traffic flowing through the OMF Gateway, i.e. the amount of traffic between nodes located behind the gateway and nodes located elsewhere in PlanetLab, was significantly lower in case the modified tracker was used. If we compare the overall amount of bytes exchanged by peers, the results show that, in case the modified tracker was used, the file was downloaded from the outside slightly more than once, and then disseminated in the WMN among nearby nodes. In case the unmodified tracker was employed, instead, it is as though the file was retrieved almost three times (about 133 Mbytes downloaded from the outside), thus indicating a non-optimal peer selection strategy.

While conducting the experiment, some real world issues arised and made evident the usefulness of having such an heterogeneous network scenario.

The first problem was about the private addressing of the WMN and the need to NAT the traffic generated from the wireless nodes and destined to the Internet. This was, however, not sufficient, as the Bittorrent protocol requires that the clients be reachable from the outside on public IP-port pairs. For this reason, we had to setup a NAT-PMP service on the gateway node. Through this protocol, clients are able to request a port to be forwarded from the gateway node, so that they can accept incoming connections from other peers on the gateway IP and the assigned port.

Clients, therefore, announce themselves to the Tracker with their public IP-port pair. This requires, in turn, that the connections between two wireless nodes go through the gateway machine and be NATted, even if they do not involve a node on the Internet. Solutions to this problem require modification to the Bittorrent client, in order to implement a local peer discovery process.

VI. Conclusions

The availability of large scale testbeds integrating several local wireless mesh testbed in a realistic global-scale environment is necessary to test WMNs with realistic traffic loads. In this paper we present an integration architecture for experimenting with local OMF-based wireless testbeds in the context of PlanetLab. In particular, we present an experimental performance evaluation of a BitTorrent traffic optimization system in the context of WMNs used as access networks to the Internet. Our experiments combine both an OMF-based wireless testbed and PlanetLab nodes located across Europe. The possibility of running this kind of experiments in such a hybrid experimental scenario highlighted several real-world issues, such as the impact on p2p systems of NAT traversal systems, that are worth to be further investigated and that could only be reproduced thanks to our integrated environment.

Acknowledgment

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 224263 (OneLab2).

References

[1] I. Akyildiz, X. Wang, and W. Wang, "Wireless mesh networks: a survey," *Computer Networks,*, vol. 47, no. 4, pp. 445–487, 2005.

[2] B. Blywis, M. Günes, F. Juraschek, and J. Schiller, "Trends, advances, and challenges in testbed-based wireless mesh network research," *ACM/Springer Mobile Networks and Applications (MONET),*, vol. 15, no. 3, pp. 315–329, June 2010.

[3] T. Karagiannis, A. Broido, N. Brownlee, K. Claffy, and M. Faloutsos, "Is p2p dying or just hiding?" in *Proceedings of IEEE GLOBECOM '04*, vol. 3, 2004, pp. 1532–1538.

[4] B. Chun, D. Culler, T. Roscoe, A. Bavier, L. Peterson, M. Wawrzoniak, and M. Bowman, "PlanetLab: An Overlay Testbed for Broad-Coverage Services," *ACM SIGCOMM Computer Communication Review,*, vol. 33, no. 3, pp. 3–12, July 2003.

[5] T. Rakotoarivelo, M. Ott, G. Jourjon, and I. Seskar, "OMF: A control and management framework for networking testbeds," *ACM SIGOPS Operating Systems Review,*, vol. 43, no. 4, pp. 54–59, 2009.

[6] C. Elliott, "GENI: Opening up new classes of experiments in global neworking," *IEEE Internet Computing,*, vol. 14, no. 1, pp. 39–42, 2010.

[7] A. Gavras, A. Karila, S. Fdida, M. May, and M. Potts, "Future Internet research and experimentation: the FIRE initiative," *SIGCOMM Computer Communication Review,*, vol. 37, no. 3, pp. 89–92, 2007.

[8] G. D. Stasi, S. Avallone, and R. Canonico, "Integration of OMF-based testbeds in a global scale networking facility," in *Proceedings of ICST QShine 2009*, ser. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 22. Las Palmas de Gran Canaria (Spain): Springer, November 2009.

[9] R. Mahindra, G. Bhanage, G. Hadjichristofi, S. Ganu, P. Kamat, I. Seskar, and D. Raychaudhuri, "Integration of heterogeneous networking testbeds," in *Proceedings of TridentCom 2008*, Innsbruck (Austria), March 2008.

[10] Y. Liu, H. Zhang, W. Gong, and D. Towsley, "On the Interaction Between Overlay Routing and Underlay Routing," in *Proceedings of IEEE INFOCOM 2005*, Miami (Florida, USA), March 2005, pp. 2543–2553.

[11] V. Aggarwal, A. Feldmann, and C. Scheideler, "Can ISPs and P2P systems co-operate for improved performance?" *ACM SIGCOMM Computer Communications Review (CCR)*, vol. 37, no. 3, pp. 29–40, July 2007.

[12] H. Xie, R. Yang, A. Krishnamurthy, Y. Liu, and A. Silberschatz, "P4P: Provider Portal for Applications," *ACM SIGCOMM Computer Communications Review (CCR)*, vol. 38, no. 4, pp. 351–362, October 2008.

[13] A. Anadiotis, A. Apostolaras, D. Syrivelis, T. Korakis, L. Tassiulas, L. Rodriguez, I. Seskar, and M. Ott, "Towards maximizing wireless testbed utilization using spectrum slicing," in *Proceedings of TridentCom 2010*, Berlin (Germany), May 2010.

[14] S. Bhatia, "VSys: A Privilege Allocation Tool," Princeton university, Tech. Rep., September 2008. [Online]. Available: http://www.cs.princeton.edu/\~sapanb/vsys/vsys.pdf

[15] "Netfilter/IPtables," http://www.netfilter.org.

[16] "Joomla," http://www.joomla.it/.

[17] "PlanetLab API," http://www.planet-lab.org/doc/plc_api.

[18] "OpenWRT," http://openwrt.org/.

[19] "RIPE Network Coordination Centre," http://www.ripe.net/.