Ad Hoc Networks 9 (2011) 1389–1403

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

Interconnection of geographically distributed wireless mesh testbeds: Resource sharing on a large scale

Giovanni Di Stasi [a], Roberto Bifulco [a], Stefano Avallone [a], Roberto Canonico [a,*], Apostolos Apostolaras [b], Nikolaos Giallelis [b], Thanasis Korakis [b], Leandros Tassiulas [b]

[a] Dipartimento di Informatica e Sistemistica, Universita di Napoli Federico II, Via Claudio 21, 80125 Naples, Italy
[b] Department of Computer and Communication Engineering, University of Thessaly, Centre for Research and Technology Hellas (CERTH), Volos, Greece

ARTICLE INFO

Article history:
Available online 14 March 2011

Keywords:
Wireless mesh network testbeds
Large scale research infrastructures
PlanetLab
Federation

ABSTRACT

Creating large scale testbeds for evaluating wireless mesh technologies and protocols, and for testing their ability to support real world applications in realistic environments, is a crucial step towards the ultimate success of the WMN paradigm. In this paper we suggest the hierarchical federation of a planetary scale infrastructure, such as PlanetLab, with a number of local OMF-based wireless testbeds as a viable approach towards this goal. Along such direction, we present an architectural model for integrating at the technical level these two kinds of infrastructures and our initial implementation of such a model. We also present some test case experiments we run on our initial implementation of the integrated architecture, to illustrate how an experiment on peer-to-peer traffic optimization can be executed by combining both wireless nodes of a OMF-based 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 that are worth to be further investigated.

1. Introduction

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 [1]. Due to the inherent difficulty of capturing all the relevant aspects of the real behavior of these systems in analytical or simulation models, research on WMNs has always heavily relied on experimental testbeds. In fact, the creation of such experimental testbeds has been an active area of research in wireless mesh networking over the last ten years [2]. However, it is difficult (and costly) to setup a large-scale wireless mesh testbed to experiment with new applications, services and protocols. Also, wireless mesh networks are usually employed as access networks to the Internet, hence testing new solutions thoroughly requires to take the complexity of the real Internet into account.

To allow for a realistic evaluation of new applications, services and protocols specifically designed for wireless mesh networks, we analyzed the existing projects that enable to share and manage testbeds and resources over a large geographic area. On the one hand, PlanetLab is universally known to be an open platform to conduct realistic experiments on a planetary scale [3]. On the other hand, OMF (*cOntrol and Management Framework*) is a well-established software platform that supports the management and automatic execution of experiments on a networking testbed. Originally developed for the ORBIT wireless testbed at Winlab, Rutgers University [4,5], OMF is now deployed in several testbeds in Australia, Europe, and in the US [6].

In this paper we present a contribution towards the interconnection of geographically distributed OMF-based wireless testbeds through PlanetLab. Our approach allows

* Corresponding author. Tel.: +39 0817683831; fax: +39 0817683816.
E-mail address: roberto.canonico@unina.it (R. Canonico).

1570-8705/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.adhoc.2011.03.002

the making of experiments involving the use of resources provided by a local wireless testbed in combination with other resources provided by other remote sites connected to the PlanetLab planetary-scale testbed. This allows running experiments on wide-area infrastructures, involving several kinds of technologies, both in the core of the network, where they cannot be controlled by experimenters, and at the edges, where they can be selected to compare several kinds of access networking technologies, such as Wi-Fi, WiMAX, UMTS, wireless mesh networks. The contribution we present into this paper is in line with current ongoing efforts towards the so called "federation" of experimental infrastructures. A testbed federation has been recently defined as *the interconnection of two or more independent testbeds for the creation of a richer environment for experimentation and testing, and for the increased multilateral benefit of the users of the individual independent testbeds* [7] and it currently appears as the most reasonable way to build large-scale heterogeneous testbeds. Roadmaps envisioned by the most significative research initiatives focusing on future research infrastructures, such as GENI [8,9] and FIRE [10], assign a key role to federation of existing testbeds. Actually, we envision a hierarchical federation model, as depicted in Fig. 1, in which global scale Tier-1 testbeds, federated among them in a peer-to-peer way, act as "aggregators" of local Tier-2 testbeds. In this view, we assume PlanetLab and PlanetLab Europe as existing Tier-1 testbeds, whose federation is already in place and operational since 2008.

Federation of heterogeneous testbeds involves a number of both technical and organizational issues. With regards to the technical challenges, they comprise the problem of sharing user credentials, as well as armonising usage models and resource management policies among testbeds. Our contribution accounts for such problems and we will describe hereinafter how we dealt with them. Thus, our contribution can be viewed as a preliminary effort in the direction of the *federation* of two different kinds of testbeds that we feel are of extreme importance for researchers working on wireless mesh networks.

In particular, in this paper we present how we integrated some basic mechanisms for accessing the resources provided by a OMF-based wireless testbed from the PlanetLab environment. Our contribution allows the seamless integration of the OMF resources into the global-scale PlanetLab infrastructure, creating a synergic interaction between the two experimental facilities.

The rest of the paper is organized as follows. In Section 2 we briefly describe the architecture of PlanetLab, its usage model and resource management techniques. Likewise, in Section 3 we briefly describe the architecture of OMF, its usage model and resource management techniques.

In Section 4 we describe the integration steps that we developed to allow for distributed experiments involving two OMF-based wireless mesh testbeds, in combination with a number of PlanetLab hosts spread all over the world. In particular, we describe a software system that is able to manage resource scheduling for both resources included in the OMF-based testbeds and in the PlanetLab nodes.

In Section 5 we describe the two OMF-based testbeds involved in our validation experiments: the NITOS wireless testbed located at the University of Thessaly and the WILEE testbed located at University of Napoli Federico II in Italy.

In Section 6 we illustrate how we used the integrated testbed setup to conduct an experiment aimed at evaluating a peer-to-peer traffic optimization technique. 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.

In Section 7 we compare our contribution against similar integration efforts that have been proposed in the past years. Finally, in Section 8 we draw our conclusion on the relevance of our contribution and its potential for future developments.

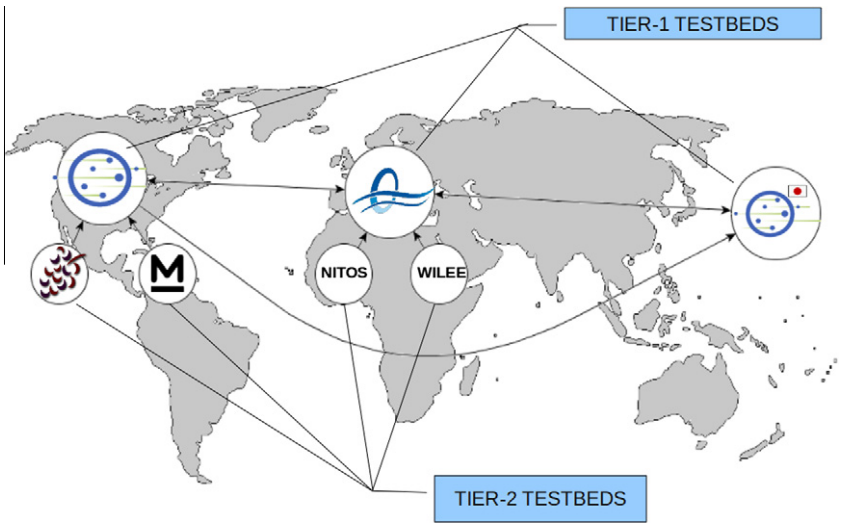

Fig. 1. Hierarchical federation of heterogeneous testbeds.

2. PlanetLab: architecture, usage model and resource management

The most relevant large scale distributed testbed for networking research as of today is PlanetLab [3]. 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. One of the main limitations of PlanetLab, however, is its lack of heterogeneity. Nearly all PlanetLab nodes are server-class computers connected to the Internet through high-speed wired research or corporate networks. As a consequence, it has also been noted that the behavior of some applications on PlanetLab can be considerably different from that on the Internet [11,12]. Several efforts have been done in the last few years to add different kinds of networking technologies to PlanetLab (e.g. UMTS integration in PlanetLab is described in [13]) or to integrate new kind of terminals (e.g. the integration of non-dedicated devices made available by residential users is described in [14]). However, it is now clear that PlanetLab can be usefully complemented by a variety of other testbeds, in particular when experimentation with wireless technologies is required.

2.1. Architecture

Fig. 2 shows a conceptual view of the current architecture of the PlanetLab testbed, whose node set is the union of disjoint subsets, each of which is managed by a separate authority. As of today, two such authorities exist: one is located at Princeton University (PLC) and the other is located at Université Pierre et Marie Curie UPMC in Paris, France (PLE). An experiment in PlanetLab is associated to a so-called *slice*, i.e. a collection of virtual machines (VMs) instantiated on a defined subset of all the testbed nodes. Each testbed authority hosts an entity called *Slice Authority* (SA), which maintains state for the set of system-wide slices for which it is responsible. The slice authority includes a database that records the persistent state of each registered slice, including information about every user that has access to the slice [15].

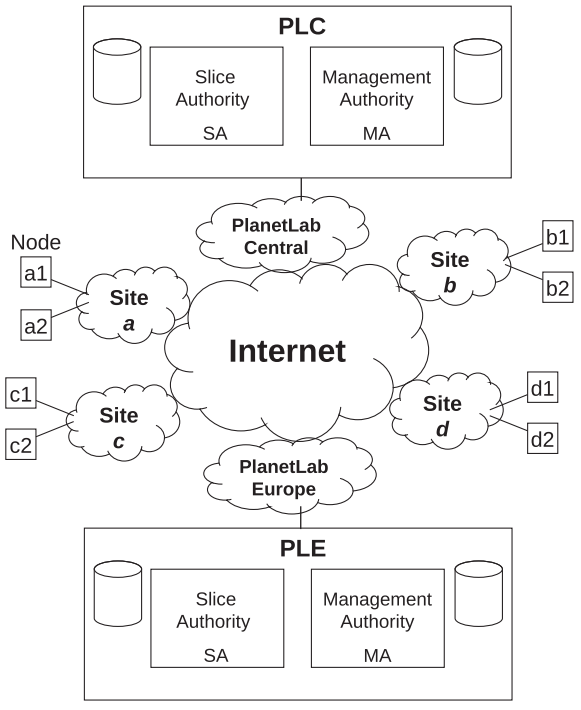

Fig. 2. Conceptual PlanetLab architecture.

Testbed authorities also include a so called *Management Authority* (MA), which is responsible of installing and managing the updates of software running on the nodes it manages. It also monitors these nodes for correct behavior, and takes appropriate action when anomalies and failures are detected. The MA maintains a database of registered nodes at each site. Each node is affiliated with an organization (owner) and is located at a site belonging to the organization.

2.2. Usage model

To run a distributed experiment over PlanetLab, users need to be associated with a slice. 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*. Slivers are Virtual Machines created in a Linux-based environment by means of the VServer virtualization technology. By means of so-called *contexts*, VServer hides all processes outside of a given scope, and prohibits any unwanted interaction between a process inside a context and all the processes belonging to other contexts. VServer is able to isolate services with respect to the filesystem, memory, CPU and bandwidth. However, it does not provide complete virtualization of the networking stack since all slivers in a node share the same IP address and port space. The adoption of VServer in PlanetLab is mainly motivated by the need of scalability, since up to hundreds of slivers may need to be instantiated on the same physical server [16]. Fig. 3 shows the internal view of a PlanetLab node.

2.3. Resource management

In PlanetLab, slice creation and resource allocation are decoupled. When a slice is first created, a best effort service

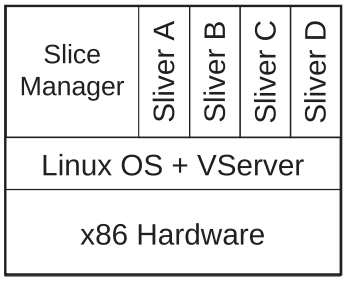

Fig. 3. Internal view of a PlanetLab node.

is associated with it and resources are acquired and released by the slice during its entire lifetime. Therefore, by default, slices are not bound to sets of guaranteed resources. Such an approach has been deliberately chosen in the original PlanetLab design. PlanetLab, in fact, has not been designed for controlled experiments, but to test services in real world conditions [17,18]. After its initial development, PlanetLab has been extended with a calendar service, called SIRIUS, whose purpose is to allow users to obtain a "better service" from all the nodes participating to a given slice. In practical terms, this means that, during a reserved time slot, a slice may be granted 25% of each processor's CPU capacity, and 2 Mbps of link bandwidth. The actual usage of SIRIUS by PlanetLab users is quite modest, since it does not allow precise control over the reservable resources.

3. OMF: architecture, usage model and resource management

OMF (*cOntrol and Management Framework*) is a Testbed Control, Measurement and Management Framework. In the following of this section we will briefly describe OMF architecture, usage model and resource management. We also describe how experiments may coexist in the same OMF testbed, thanks to the NITOS scheduler.

3.1. Architecture

The components of OMF (Fig. 4) work together to automatically perform all the phases needed to execute the experiment, 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. It accepts as input an experiment description and takes care of orchestrating the testbed resources in order to accomplish the required experiment steps. It interacts with the *Aggregate Manager*, the entity responsible of the resources of the testbed as a whole, and provides some basic services to the EC, such as checking the status of a node, rebooting a node, etc.

The EC also interacts with the *Resource Controllers* (RCs) installed on the testbed nodes. These latter entities are responsible of performing local configuration steps, e.g. configuring the channels on the Wi-Fi interfaces, and of controlling the applications, e.g. the traffic generator. The communication between the EC and the RCs is based on a publish/subscribe paradigm, where the EC publishes the messages on a XMPP server [19] and the RCs pick the messages addressed to them.

An important companion library of OMF is OML (*OMF Measurement Library*), which is used to automatically filter and collect experiment data on one or more measurement servers. OMF is able to instrument the OML library, in order to configure and guide the collection of experiment data.

3.2. Usage model

In order to perform an experiment, users have to log into the *testbed console*, i.e. the host running the Experiment Controller (EC). The execution of an experiment can be requested to the EC by submitting an experiment description in the domain-specific OEDL language, which is derived from Ruby. The experiment description usually consists of two parts: (i) a first declarative part, comprising a list of required resources and applications, with their configuration; (ii) a second part, describing the set of actions to be performed in order to realize the experiment. The execution of specific actions may depend on events which are defined by the platform, e.g. all the nodes are up and running.

3.3. Resource management

OMF, in its basic form, 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 ORBIT a Scheduler interface is provided to support the reservation of the entire testbed.

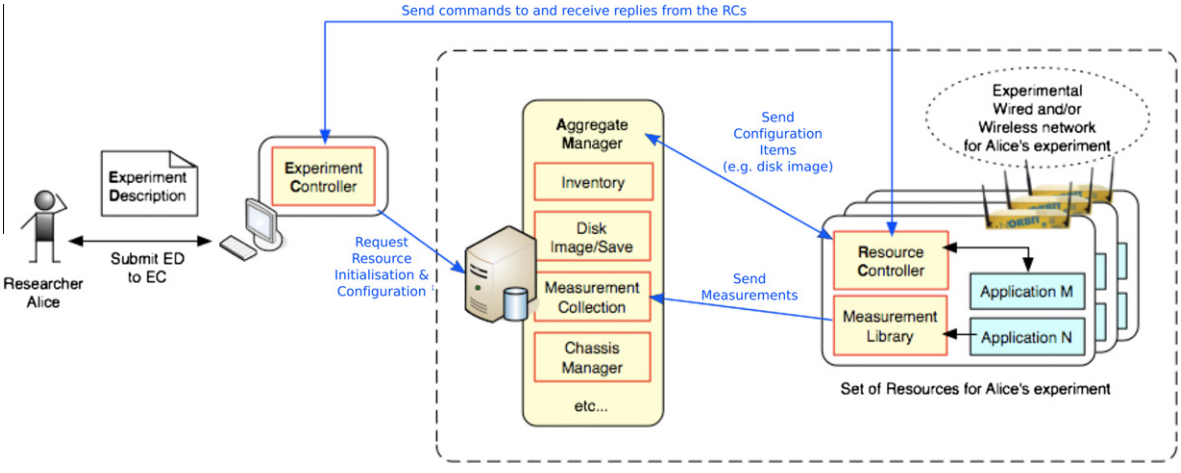

Fig. 4. OMF architecture overview.

The user books the testbed in advance and during the reserved time slot he/she is the only one allowed to log into the testbed console and run his/her own experiments.

In the NITOS and WILEE testbeds a different Scheduler, i.e. the NITOS Scheduler, is employed. Differently from ORBIT, different users can perform experiments in parallel on the same testbed. This is achieved by assigning a different subset of nodes and wireless channels to each user. 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. To achieve that, modifications to OMF were required, as explained in the following section.

3.4. The NITOS scheduler

Currently OMF does not include any scheduling algorithm to synchronize the execution of experiments. Also, permissions to access the testbed resources are not checked. However, in a public, multiuser environment, we need a system that is able to assign resources only to the users that have the right to use them, while providing the experimenters with a way to specify the resources that they need for their experiments. In our work, resources are divided in two categories: nodes and spectrum. Thus, we provide a tool which is used by the experimenters to reserve nodes and spectrum for a specified time interval (whose duration must not exceed some limit). By slicing, we mean the partitioning of the testbed based on some criteria. With spectrum slicing, we aim to partition the testbed into smaller, virtual, testbeds which are using different spectrum and, hence, they do not interfere with each other in the entire testbed infrastructure. Using spectrum slicing, our tool makes the testbed available to users who would like to use different resources (spectrum, nodes) at the same time [20]. For example, many users can use the testbed simultaneously since we can allocate a particular group of channels to a group of nodes that can be assigned to one user.

3.4.1. The NITOS Connectivity Tool

Before describing the NITOS Scheduler and how users select nodes and frequencies, we briefly present a tool that provides updated information on the channel link quality in order to help users decide which nodes are the most appropriate for their experiments. Most wireless testbeds are not RF isolated, hence the link quality between any pair of nodes may unexpectedly vary at any point in time due to external interference. For this reason, the static distribution approach, that is used in RF isolated wireless testbeds [21], is not efficient for these deployments. Therefore, there is the need for updated information in terms of measurements of link quality, that will bring a more accurate channel quality estimation. To this purpose, a management tool called *NITOS connectivity tool* has been developed for assessing channel quality information and measuring channel connectivity among Wi-Fi interfaces. We have implemented the NITOS connectivity tool based on TLQAP (see [22]), which is a protocol used to assess the connectivity and the quality of a link by estimating the packet delivery ratio (PDR) for all requested channel, rate and transmission power combinations. Specifically, TLQAP builds a measurement history log, creates a channel utilization profile and stores that information in a database that is used for link quality information retrieval by the NITOS connectivity tool.

The NITOS Connectivity Tool is comprised of three entities: a web interface, a database and a set of .dot scripts. Through the web interface, the user selects a node he/she wants to use in the experiment, an operating frequency (among those specified by the IEEE 802.11a/b/g standards) and a transmission rate. The database storing the information on the channel link quality (that is periodically updated by running TLQAP) is queried to retrieve the requested information. The result (a set .dot files) is presented to the user through a set of graphs, each of which is related to a Wi-Fi interface of the selected node. Fig. 5 shows the graphs corresponding to the two wireless interfaces of node 4. Each graph shows the links between a wireless interface on the selected node and the interfaces of the neighbor nodes. Upon each link, the MAC address of the neighbor's interface and the PDR of the link are reported.

3.4.2. Scheduler scheme

Slices are dynamically created on the testbed upon the user reservation. A user first reserves nodes and channels for a specified time range and then logs into the testbed and executes his experiments. Once the reservation procedure is concluded, the system is aware of the resources that the user needs and the time range that he will keep them. During this time range, no other user can use any of the reserved nodes or the reserved channels.

Existing public Wi-Fi testbeds only allow exclusive reservations in a given time period. Our scheduler instead allows multiple users to share the testbed at the same time. Indeed, the scheduler guarantees that they use distinct nodes and distinct frequencies, so that their experiments do not interfere with each other.

We now describe the reservation procedure. First of all, the user has to set the date and time that he would like to reserve a slice. The time is slotted with each slot duration set to 30 min. Then, he checks for the available resources in terms of nodes and channels. Fig. 6a shows a user checking for available nodes on May 30, 2010 for 2 h starting at 12:00 pm. Also, a map of the building is shown, in order to give the user a better perspective of his reservation.

The scheduler keeps all reservations in a database. A reservation is a set of nodes, channels and a time range. When a user checks for available nodes, the scheduler searches its database for any possible record in the time range that the user specified. Then, it only returns the available set of nodes and channels, i.e., the nodes and channels that have not been selected by any other user in the specified time range (Fig. 6b). In this way, the system ensures that both the time and the frequency division requirements will be met. After the user has made and confirmed its selection, the scheduler database is updated. From this point on, the scheduler is responsible for ensuring that the user will only use the reserved slice for the specified time period.

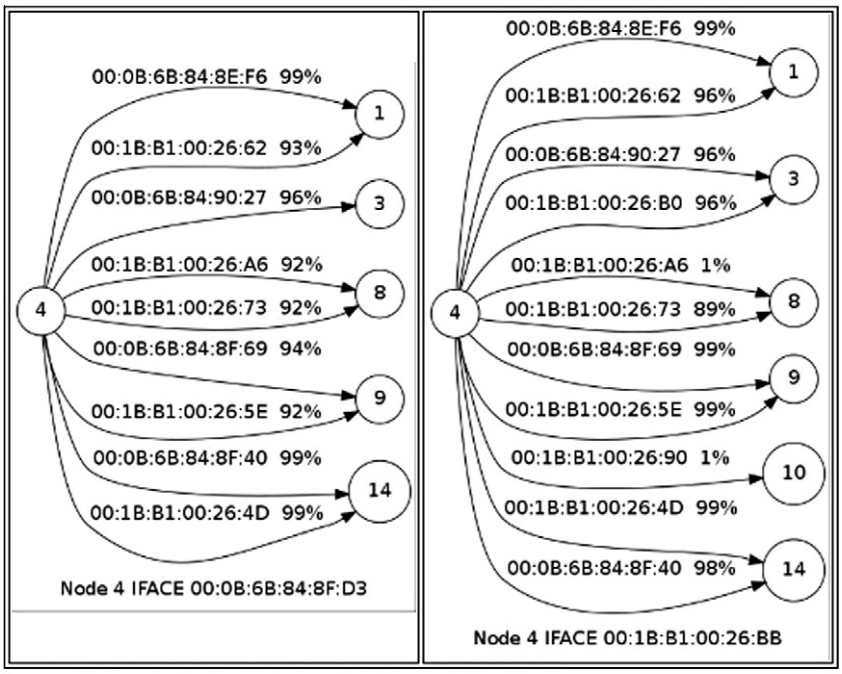

Fig. 5. Link quality for node 4.

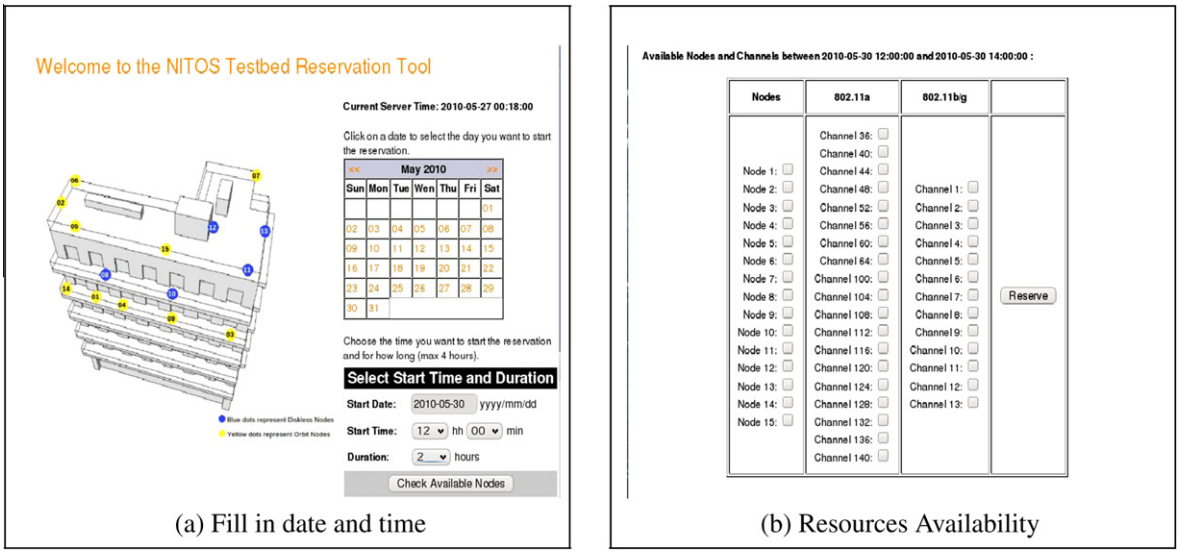

Fig. 6. Resources reservations.

3.4.3. OMF extension to support NITOS scheduler slicing features

The scheduler mainly consists of two parts: a user interface, which is responsible for guiding the user through the reservation process making sure that he does not make a reservation conflicting with reservations made by other users, and a system component, which controls the slices by ensuring that this user's experiments will only use the reserved resources. The user interface role has been illustrated in the previous subsection, while the system encapsulation of the scheduler will be illustrated in this subsection.

So far we have described the part of the scheduler which is focused on the experimenter and his choices at reservation. However, we also need to ensure that the experimenters will stick on their choices and, even if they try, the system will not allow them to use any resources that they have not reserved. For this purpose, we have

chosen to extend OMF. Here, we give a detailed description of the additions and the extensions we had to make inside this framework to integrate spectrum slicing support.

Firstly, we need a way for OMF and the scheduler's database to communicate. For this purpose, we have added one more service group to the Aggregate Manager named scheduler and one more service to the inventory service group. Next, we show what these services are responsible for. First of all, the inventory service group is developed inside OMF and provides a set of webservices that provide general information about the testbed (such as node names, IP addresses, etc). This information is stored in a database residing on the testbed server and the inventory service group reads this database to return the proper response. Our addition here is a service which gets a node location (i.e., its coordinates) based on its IP address. Note here that the information on the node location is the same on both the scheduler's and the testbed's database and, thus, we can use it to do the matching (coordinates do not refer to real data, but on an internal mapping that helps partitioning the testbed into groups while also allowing the identification of each node by OMF). We have added this service because, when an experiment is executed, OMF does not know a node's location, but only its IP address.

Now that scheduler knows the exact location of the node, it can use the scheduler service group to get any information needed from the scheduler's database. Namely, the services provided by this group provide functionality to get a node reservations based on its coordinates, the spectrum that this reservation contains and the user that owns it. Furthermore, it provides services that can do the matching between a channel or a frequency number and the respective spectrum identification number as it is stored in the database. All this information will be used by the Resource Controller, which decides whether to allow the user to use the channel or not.

Thus, RC is responsible for deciding whether the resources declared in the experiment should be allocated to the experimenter. In order to decide, the RC has to ask the scheduler's database if the specified resources have been reserved by the experimenter. So, when the experiment sets the wireless card channel, this information is passed to the RC, which now knows the channel along with its own IP address. All he needs is the user identification to check with the scheduler's database if this channel (and, of course, node) should be allocated to that user.

However, this is not straightforward, since the user usually logs into the node as root (keep in mind that the experiment loads his own image to the nodes, so he has full privileges on them). So, we need to track where did he use the username that he also used for registering. The scheduler is designed in such a manner that, when a user registers to the system, then an account with the same username and password is automatically created to the testbeds server. The user uses this account to both access the user interface and the testbed server (using secure shell connection). This can solve our problem, since we can say for sure that the user that is running the experiment is logged into the console with the same username that he has made his reservation.

This information, though, relies on the testbed server, while the RC runs on the client side, i.e., on the nodes. We need to pass that information from the server to the clients. This is done by the Experiment Controller, the OMF service that is running on the server side and is responsible for controlling the experiment execution. Using its built-in message passing mechanism, EC tells the RC the username of the experimenter and now the last one has almost everything he needs to do the matching, except the date. The system should not rely on the experimenter to keep the clock of his clients synchronized with the testbed. This is why, EC sends, along with the username, the current date and the RC adjusts its clock to match the server's clock.

At this point, RC has all the information needed to check with the scheduler if the requested resources should be allocated to the experimenter. Using the web services we described above, the RC checks if there is a reservation at that time for that user and if the spectrum reserved at this reservation matches the channel that the experimenter has requested to assign to the network card through his experiment.

If all data match, then the RC lets the experiment execution move on. Otherwise, it notifies the EC that a resource violation has taken place and stops its execution (without assigning the channel to the node network card). When the EC receives that message, the execution is terminated immediately and an ERROR message is thrown back to the experimenter describing the resource violation. Then the user is prompted to reconfigure its experiment with the permitted frequencies that he is allowed to use and he has already reserved during the scheduling process (see Section 3.4.2).

3.4.4. NITOS scheduler advantages

NITOS scheduler provides all the appropriate tools to allow slicing to its resources. Because of the external deployment of NITOS testbed, interference from external WMN links cannot be avoided. For that reason, NITOS Connectivity tool aids in identifying resources that best fit to the users experiment requirements. Moreover, NITOS Scheduler and its tools can be modified with minor changes and adapted to any wireless testbed that needs usage efficiency no matter if it is located in an isolated environment or it is located among external WMNs. In this way, NITOS scheduler aims to achieve better utilization of testbed resources, while also enables users to deploy their experiments in a more efficient way.

4. PlanetLab and OMF integration

Our main goal is to integrate a global-scale PlanetLab infrastructure with a local OMF-based wireless testbed. In particular, we aim at using the OMF-based testbed as an access wireless mesh network for a set of PlanetLab nodes co-located (i.e. in range of wireless transmission) with it.

As described in the introduction, we recognize a value in this integration, as a first necessary step for the federation of these two kinds of infrastructures, and because it

adds new capabilities to the PlanetLab environment. Our system allows the seamless integration of the OMF resources into the global-scale PlanetLab infrastructure, creating a synergic interaction between the two environments.

4.1. Integrated architecture

The architecture we propose is depicted in Fig. 7. It consists of the following elements:

- A PlanetLab site S whose nodes are equipped with one ore more Wi-Fi 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 OMF testbed's nodes, whose NICs are assigned private IP addresses.

The PL-Edge nodes are multi-homed PlanetLab nodes which can act as clients for the OMF wireless testbed. The lack of proper support for multihoming in PlanetLab led us to the developement of *sliceip*, a tool for allowing the definition of slice-specific routing tables that will be presented later.

In the OMF-PlanetLab integrated scenario, two kinds of resources are made available to experimenters:

- *bookable resources*, i.e. resources that can be exclusively assigned to an experiment over a given time interval;
- *non-bookable resources*, i.e. resources that cannot be exclusively assigned to an experiment over a given time interval, as they are shared among concurrently running experiments;

The purpose of the extended NITOS scheduler is to allow the reservation of bookable resources in the integrated scenario. These resources comprises both OMF wireless nodes and channels, and PlanetLab non-virtualized resources, i.e. the Wi-FI interfaces. To do that, the extended NITOS scheduler interacts with the *OMF Console*, in order to enable or disable access to slices to the Experiment Controller, and with the PlanetLab nodes, in order to enable or disable the access to specific slices to the wireless interfaces. The communication with the PlanetLab nodes is performed by means of a management sliver, called *SM Sliver* (Scheduler Management Sliver), which accepts requests by the Scheduler through a secure ssh connection and performs the association between the slices and the wireless interfaces. We remember that we allow only one slice at a time to have access to a wireless interface, in order to limit interferences among experiments.

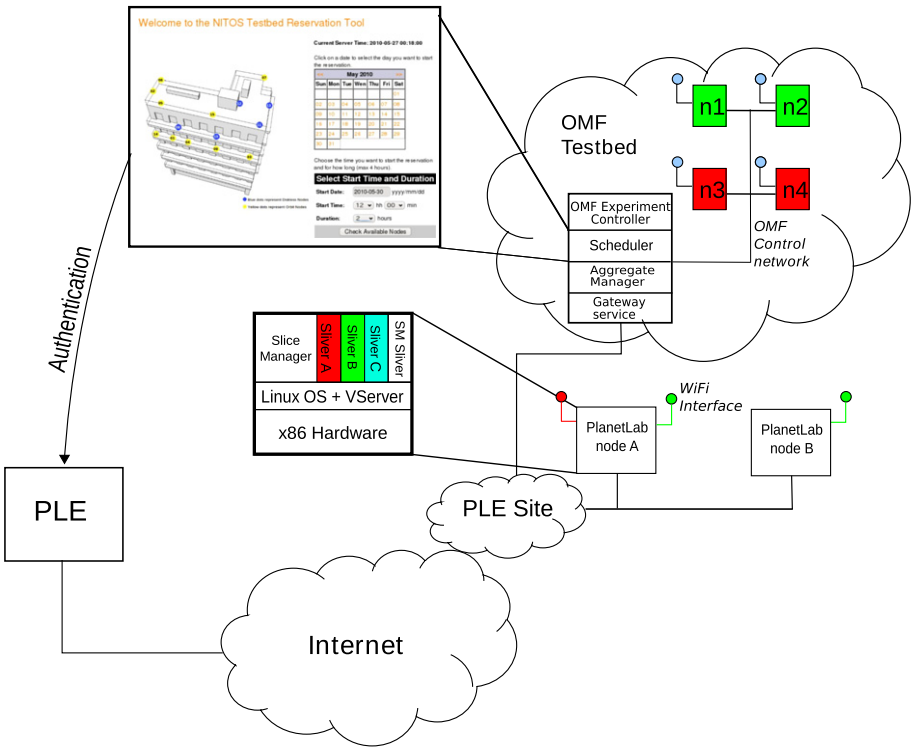

Fig. 7. OMF-PlanetLab integrated architecture.

The Scheduler performs authentication of the user on the PLE, thus allowing access to the Tier-2 OMF wireless testbed to PlanetLab Europe users. Local users, i.e. users of the wireless testbed, are supported and their credential are stored on the Scheduler. These class of users however, i.e. users of the Tier-2 testbed, have not access to the global infrastructure, i.e. the Tier 1 testbed.

In the OMF wireless testbed private IP addressing is used. Therefore, in order to allow experiments involving nodes located elsewhere on the public Internet, a node acting as a NAT router is needed. This function is performed by the Gateway Service. In the case of experiments involving OMF nodes located at different PL-OMF sites, site-to-site IP tunnels might be established between PL-OMF Edge Nodes. This process would be easy to be managed if these nodes were VINI nodes.

After user authentication the OMF Scheduler, by means of cron scripts, enables/disables access to OMF testbed nodes from the user's slice.

4.2. 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 PL-OMF Edge Nodes (OP) to his/her slice.
2. U logs into the Scheduler at site S and books the resources (nodes, channels, Wi-Fi interfaces of OP 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.

4.3. Multihoming support in PlanetLab

While trying to support the proposed usage model, we run across a serious limitation of the PlanetLab management software. Such a limitation is about the correct managing of multi-homed nodes, i.e. nodes connected to more than one access network. This has not been a problem for a long time, as PlanetLab mainly consisted of just a set of hosts connected to Internet through a single, high speed corporate connection. In such a scenario, there is no need for users to be able to modify the routing table, as the route for the Internet is only one. In recent times, though, some attempts to enhance the heterogeneity of PlanetLab have been made. In the context of the OneLab European research project, different kinds of wireless access technologies (such as UMTS, WiMAX and Wi-Fi) have been made available to a subset of nodes connected to PlanetLab Europe, in addition to the default wired connection to the Internet. In [13], the software tools that have been developed to manage a UMTS connection in that context are described. In this paper we describe a generalization of that software, allowing it to work with any kind of network interface.

4.3.1. The sliceip tool

In order to fully exploit the possibility of multi-homed PlanetLab nodes we developed a tool called *sliceip*. The purpose of this tool is to enable slice-specific routing tables in PlanetLab. Using this tool, the user is able to define routing rules which apply only to traffic belonging to his/her slice. This is required for users to be able to choose which interface to use for their experiments. For instance, a user can specify that he or she wants to reach a certain destination on the Internet, e.g. another PlanetLab node, through the Wi-Fi interface. For achieving this result, he or she would add a routing rule in his/her own routing table by means of our tool, in the same way he or she would do with conventional tools like *ip* of *route*. This is not possible in PlanetLab, because PlanetLab users do not have the superuser privileges required to modify the routing table of the node. Even if they had such privileges, any modification they performed on the routing table would interfere with all the experiments running on that node, thus breaking the isolation among experiments. With sliceip, instead, we give to the user the ability to define his/her own routing table, with no effects on experiments performed by other users.

sliceip enables slice-specific routing tables by leveraging a feature of the Linux kernel and a feature of the VNET+ subsystem of PlanetLab [23]. The Linux kernel has the ability to define up to 255 routing tables. To have some traffic routed with a particular routing table, it is necessary to associate that traffic to it by means of rules applied with *iproute2*. The rules can specify packets in terms of the destination address, the netfilter mark, etc. In our case, we set the netfilter mark of packets belonging to the user's slice (i.e. the packets that are generated or are going to be received by an application running on that slice) by exploiting a feature of the VNET+ subsystem of PlanetLab. By means of an *iptables* rule, we instruct VNET+ to set the netfilter mark equal to the slice id to which they belong. We then add an *iproute2* rule to associate packets belonging to the slice to the slice-specific routing table. We also set an *iptables* SNAT rule (*Source Network Address Translation*) in order to set the source IP addresses of packets that are going out through a non-primary interface (the primary interface is the one the default routing rule points to). This rule is required because the source ip addresses of packets are set after the *first routing process* happens. In fact, in case more than a routing table is used, the routing process follows these steps: (1) the interface for sending the packets is decided following the rules of the main routing table and the source ip addresses are set accordingly (this is the first routing process); (2) if the user changes the mark of the packets in the *mangle chain* of *iptables* and a rule is defined for routing those packets with a different routing table, a *rerouting process* is triggered. This rerouting process follows the rules of the selected (i.e. the slice-specific) routing table and the interface to be used is set accordingly; (3) the packet is sent out using the selected interface. During the

step 2, the source ip addresses of packets are left unchanged, so we need to change them explicitely before the packets are sent during the step 3.

The user interacts with *sliceip* by means of a front-end that resides in the slice. This front-end extends the syntax of the *ip* command of the *iproute2* suite with the following two commands:

- *enable* <*interface*>: initialise the routing table for the user's slice, set the rule to mark packets belonging to the user's slice, add a rule to associate those packets with the routing table of the slice and add the SNAT rule for <interface>;
- *disable* <*interface*>: remove the SNAT rule for <interface>, remove the rule to associate the packets to the routing table of the slice and remove the rule that marks the packets of the user's slice.

4.4. Extension of the NITOS scheduler to manage PlanetLab resources

In order to support the reservation of bookable Planetlab resources, i.e. the Wi-Fi interfaces of the PL-edge nodes, we had to extend the NITOS Scheduler and make some additions to the management software of the PL-edge nodes.

The Scheduler has been extended to show among the available resources also the Wi-Fi interfaces of the PL-Edge Nodes and to allow the user to reserve them. Reservation records are kept in the Scheduler database and it is Scheduler responsibility to make sure that reservations made by two users do not overlap.

In order to enforce the assignment of the interface to the slice, when the reservation time starts, the Scheduler interacts with the *Scheduler Management Sliver* allocated on the PL-edge node. Such interaction is performed through a secure ssh connection. By means of *vsys* [24], the *Scheduler Management Sliver* is able to execute a script in the root context. This script makes the actual assignment of the Wi-Fi interface to the slice by setting some *iptables* rules which block all packets that are about to go out through the Wi-Fi interface and do not belong to the slice for which the Wi-Fi interface has been reserved.

The Scheduler checks the user's credentials by means of the PLC API and enables/disables access to OMF testbed nodes from the user's slice for the specific time and duration. In particular, the Scheduler interface is extended to support authentication of users by means of PLC managed usernames and passwords, while access to the OMF EC is performed by means of users' public keys linked to the slice, retrieved using the PLC API.

5. Experimental setup

5.1. The NITOS testbed

It is important to give an overview of the hardware facilities that comprise the heterogeneous profile of NITOS testbed. NITOS is a wireless testbed located in the University of Thessaly campus. NITOS as the main wireless testbed in the Onelab2 project, aims to provide all the software and hardware facilities that can gather multiple wireless communication technologies under a common structure. The main technology that is available in NITOS for implementation and testing is Wi-Fi. Large scale testbeds are likely to feature hardware of different architecture and performance. NITOS testbed features three different types of computer main boards, two types of wireless media as well as two other types of peripherals. More specifically the NITOS testbed features 10 Alix embedded PoE nodes with 500 Mhz i386 CPUs, which are primarily used for development of networking systems, 10 Orbit AC powered nodes (1 Ghz i386 CPUs and 1 Gb ram) and 20 Commel AC powered nodes that feature 2.4 GHz core duo CPUs (x86_64). Wireless media includes 50 Atheros 5212 interfaces and 10 Atheros 5001 interfaces. Orbit nodes are equipped with high quality USB cameras that can be used for video enabled experiments and six commel nodes are attached with GNU Radio peripherals that support PHY layer experimentation.

5.2. The WILEE testbed

The WILEE (WIreLEss Experimental) Wi-Fi Mesh Testbed is located in the Computing Department of University of Napoli Federico II. It consists of three Soekris net4826-48 Single Board Computers and eight Netgear WG302Uv1 access points. It also features a node belonging to a private PlanetLab deployment which acts as the PlanetLab Edge node and a Linux machine acting as gateway towards the Internet.

The Soekris net4826-50 SBC is based on the AMD Geode SC1100 CPU (at 266 Mhz), has 128 Mbyte DRAM memory, a 128 Mbyte Flash disk, a FastEthernet interface and two 802.11a/g Atheros wireless cards. The Netgear WG302Uv1 access point features on an Intel XScale IXP422B network processor (at 266 Mhz), has 32 Mbyte DRAM memory, a 16 Mbyte flash disk, a FastEthernet interface and two 802.11a/g Atheros wireless cards.

6. 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 testbeds as access networks 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, due to space limits, we only present how we conducted the experiments and the reasons that make our integrated infrastructure useful for evaluating wireless meshes in realistic conditions.

6.1. Testing overlay routing strategies in WMN-based access networks

An increasing number of popular Internet applications, such as Bittorrent, Skype, GoogleTalk, and P2P-TV relies

on the peer-to-peer paradigm. These applications produce more than 50% of the overall Internet traffic. 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 [25]. 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. [26] 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 [26,27]. 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. In the next subsection, we report the results of experiments carried out to show that our approach makes it very simple to perform realistic experiments to test overlay routing strategies.

6.2. Our experiments

In this section we describe an experiment aimed at evaluating 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

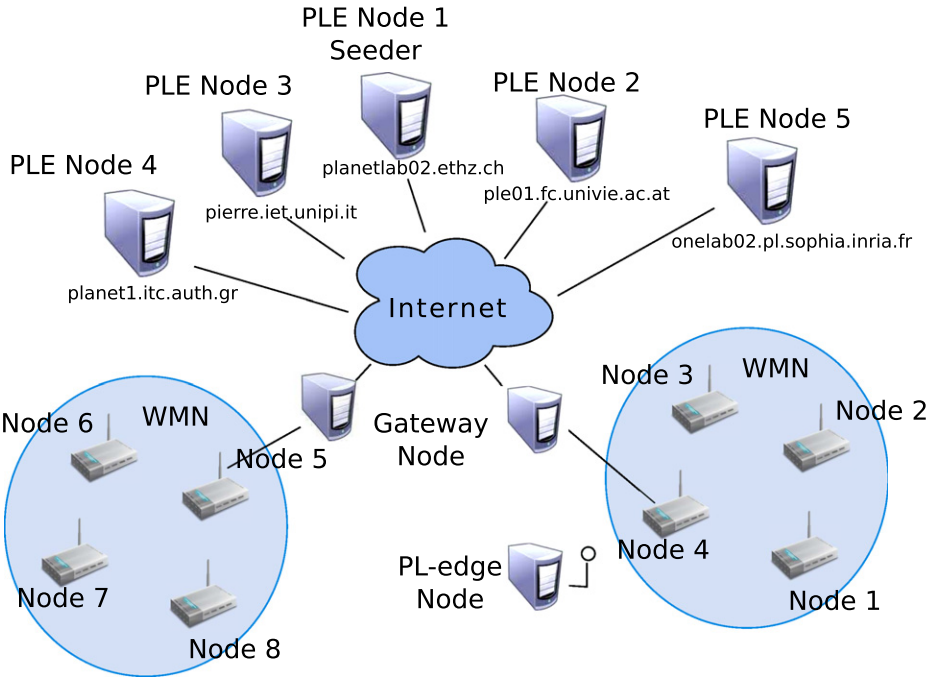

Fig. 8. Experiments setup.

AS-number associated with 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. 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 10 PlanetLab Europe nodes and the PlanetLab edge node situated at the edge of the WILEE testbed. To this slice, some bookable resources, i.e. four wireless nodes from the WILEE testbed and the Wi-Fi interface of the PL-edge node, were added to the slice by using the extended NITOS Scheduler at the WILEE site. In the same way, other four nodes belonging to the NITOS testbed were added by using the exented NITOS Scheduler at the NITOS site.

The wireless nodes were configured by using the facility offered by OMF to form two single-channel WMNs and, in case of WILEE nodes, also to provide Internet access to the PL-edge node. A Bittorrent client (*TransmissionBT*) 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 Mbytes. The scenario of the experiments is illustrated in Fig. 8.

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 distance between peers in terms of ASes.

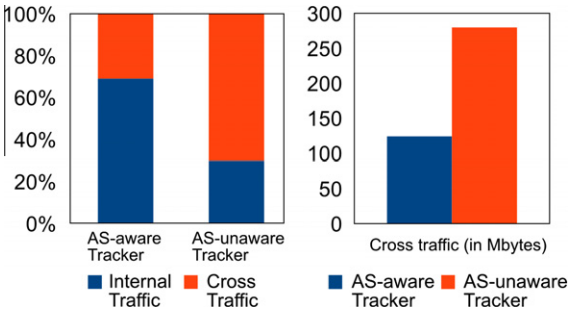

Fig. 9. Experiments: internal vs. cross traffic (percentage of total traffic) on the left; cross traffic volume on the right.

At the end of each experiment we measured the traffic belonging to connections which were either originated or destined to nodes located behind the OMF gateways, i.e. the NITOS and WILEE wireless nodes and the PL-Edge node. Our objective was to demonstrate that the traffic crossing the WMNs boundaries was minimized by using our modified tracker. In Fig. 9 we report the results averaged on 10 repetitions. The figure shows that the amount of traffic flowing through the OMF Gateways 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 in average from the outside slightly more than once for each WMN, and then disseminated in the WMNs among nearby nodes. In case the unmodified tracker was employed, instead, it is as though the file was retrieved almost three times by each WMN (about 280 Mbytes downloaded from the outside by the two WMNs), thus indicating a non-optimum peer selection strategy. Tables 1 and 2 report the traffic matrices for two experiments. On the rows are the receiving nodes, while on the columns are the sending nodes. N1, N2, etc. stand for Node1, Node2, etc., while PlanetLab is a meta node which comprises all the PlanetLab nodes. All the values are in Mbytes. It can be seen that, in case the modified tracker is used (Table 1), traffic is exchanged mainly between nodes located inside the same WMN, while in case the standard tracker is used (Table 2), wireless nodes often download from nodes which are outside their WMN.

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 [28]. 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 source NATted, at the

Table 1
Traffic matrix for an experiment with the modified Tracker.

	N1	N2	N3	N4	PL-Edge	N5	N6	N7	N8	PlanetLab
N1	0	2.34	1	0	0.44	0	0	0.81	0	41.03
N2	0	39.77	0.06	0.06	1.39	0	0	0	0	5.69
N3	13.99	3.9	0	1.89	27.19	0	0	0	0	0
N4	13.36	3.61	5.27	0	26.45	0	0	0	0	0
PL-Edge	40.7	4.23	0.64	0.09	0	0	0	0	0	0
N5	0	0	0	0	0	0	0.13	0.09	0	45.03
N6	0	0	0	13.2	0	29.79	0	0	3.55	0
N7	0	0	0	0	0	20.12	23.91	0	2.29	0
N8	0	0	0	0	0	8.17	1.95	0.5	0	37.05

Table 2
Traffic matrix for an experiment with the standard Quash Tracker.

	N1	N2	N3	N4	PL-Edge	N5	N6	N7	N8	PlanetLab
N1	0	0	0	2.88	0	0	0	0	0	43.37
N2	0	0	0	5.5	62.3	0	0	0	0	4.38
N3	0	0	0	0	4.73	0	0	0	0	48.84
N4	44.43	7.52	0	0	0	0	0	0	0	0
PL-Edge	0	0	7.88	0	0	0	0	0	0	46.88
N5	0	0	0	0	0	0	0	0	22.97	24.29
N6	0	0	0	0	0	0	5.31	0	0	40.82
N7	0	0	0	0	13.65	0	0	0	0	37.53
N8	0	0	0	0	10.82	19.88	0	0	0	16.14

gateway node, even if they do not involve a node on the Internet. Solutions to this problem require modification to the Bittorrent client, e.g. in order to implement a local peer discovery process.

7. Related work

In this paper we have presented an architectural solution to integrate a number of local OMF-based wireless testbeds with the global-scale PlanetLab environment. Our solution is a first technical solution towards the federation of these two kinds of testbeds. The problem of heterogeneous testbeds federation is under investigations of both the GENI initiative in the US and the FIRE initiative in Europe. For instance, federation between PlanetLab and EMULAB is currently being investigated in the context of the GENI initiative, as reported in [29]. An attempt to add heterogenity in PlanetLab by integration of ORBIT testbeds is in [30]. In this paper, the authors propose two models of integration. The first model (PDIE, *PlanetLab Driven Integrated Experimentation*) is intended to serve PlanetLab users who want to extend their experiments to include wireless networks at the edge without changing the PlanetLab interface, while the second model (ODIE, *ORBIT Driven Integrated Experimentation*) is intended to serve ORBIT wireless network experimenters who want to augment their experiments by adding wired network features without major changes to their code.

Our proposed model of integration is more similar to the PDIE model, with a difference with regard to the connectivity model between the two environments. In order to integrate an OMF testbed in PlanetLab, the authors propose the use of a gateway PlanetLab node, whose function is to open tunnels between itself and the selected nodes in the OMF testbed. Differently from our approach, the gateway node is not a client of the OMF testbed, but merely creates the tunnels. Our approach does not employ tunnels. A similar approach was taken in [31]. The authors aimed at integrating the VINI virtual network infrastructure [32] with OMF-based testbeds. The intention was to enable Layer 3 experimentations by allowing users create virtual topologies spanning both wired and wireless links. Also this approach relies on the use of tunnels.

Our approach intends to recreate in the testbed the same operational situation that exists in real networks, in which a private addresses mesh is connected to the Internet through NATing gateways. Our integrated experimental facility allows experimentation of low level mechanisms within the wireless mesh environment provided by the OMF testbed, and end-to-end mechanisms and applications in the global hybrid integrated environment. These features create a synergy between the two kinds of facilities. As we mentioned in the introduction of the paper, for achieving a full-fledged federation of the two environments, other issues need to be fully solved, such as the creation of a single sign-on mechanisms for the two environments.

8. Conclusions

The availability of large scale testbeds integrating several local wireless mesh testbed in a realistic global-scale environment is necessary to test WMNs in the wild. In this paper we present an integration architecture that allows to combine local OMF-based wireless testbeds with the planetary-scale PlanetLab infrastructure. In particular, we described how we solved the problem of harmonizing the resource management schemes of the two testbeds, that comprise both bookable and non-bookable resources. We also present some test case experiments we run on our initial implementation of the integrated architecture. In particular, we describe an experiment aimed at evaluating a BitTorrent traffic optimization system. Our experiment includes two OMF-based wireless testbeds (namely, NITOS and WILEE) as well as a number of 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 performance of NAT traversal systems, that are worth to be further investigated and that could only be reproduced thanks to our integrated environment.

Acknowledgments

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

References

[1] I. Akyildiz, X. Wang, W. Wang, Wireless mesh networks: a survey, Computer Networks 47 (4) (2005) 445–487.
[2] B. Blywis, M. Günes, F. Juraschek, J. Schiller, Trends, advances, and challenges in testbed-based wireless mesh network research, ACM/

Springer Mobile Networks and Applications (MONET) 15 (3) (2010) 315–329.
[3] B. Chun, D. Culler, T. Roscoe, A. Bavier, L. Peterson, M. Wawrzoniak, M. Bowman, PlanetLab: an overlay testbed for broad-coverage services, ACM SIGCOMM Computer Communication Review 33 (3) (2003) 3–12.
[4] D. Raychaudhuri, M. Ott, I. Secker, ORBIT radio grid tested for evaluation of next-generation wireless network protocols, in: Proceedings of TridentCom 2005, Trento, Italy, 2005, pp. 308–309.
[5] M. Ott, I. Seskar, R. Siraccusa, M. Singh, ORBIT testbed software architecture: supporting experiments as a service, in: Proceedings of TridentCom 2005, Trento, Italy, 2005, pp. 136–145.
[6] T. Rakotoarivelo, M. Ott, G. Jourjon, I. Seskar, OMF: a control and management framework for networking testbeds, ACM SIGOPS Operating Systems Review 43 (4) (2009) 54–59.
[7] T. Magedanz, S. Wahle, Control framework design for future internet testbeds, e&i Elektrotechnik und Informationstechnik 126 (2009) 274–279.
[8] GENI Planning Group, GENI design principles, IEEE Computer, 39 (9) (2006) 102–105.
[9] C. Elliott, GENI: opening up new classes of experiments in global networking, IEEE Internet Computing 14 (1) (2010) 39–42.
[10] A. Gavras, A. Karila, S. Fdida, M. May, M. Potts, Future Internet research and experimentation: the FIRE initiative, SIGCOMM Computer Communication Review 37 (3) (2007) 89–92.
[11] J. Ledlie, P. Gardner, M. Seltzer, Network coordinates in the wild, in: Proceedings of NSDI 2007, Cambridge, MA, USA, 2007.
[12] H. Pucha, Y.C. Hu, Z.M. Mao, On the impact of research network based testbeds on wide-area experiments, in: Proceedings of ACM IMC '06, Rio de Janeiro, Brazil, 2006.
[13] A. Botta, R. Canonico, G.D. Stasi, A. Pescapè, G. Ventre, S. Fdida, Integration of 3G connectivity in PlanetLab Europe, ACM/Springer Mobile Networks and Applications (MONET) 15 (3) (2010) 344–355.
[14] M. Dischinger, A. Haeberlen, I. Beschastnikh, K.P. Gummadi, S. Saroiu, Satellitelab: adding heterogeneity to planetary-scale network testbeds, SIGCOMM Computer Communication Review 38 (4) (2008) 315–326.
[15] L. Peterson, S. Muir, T. Roscoe, A. Klingaman, PlanetLab Architecture: An Overview, Tech. Rep. PDN–06–031, PlanetLab Consortium, 2006.
[16] S. Soltesz, H. Pötzl, M.E. Fiuczynski, A. Bavier, L. Peterson, Container-based operating system virtualization: a scalable, high-performance alternative to hypervisors, ACM SIGOPS Operating Systems Review 41 (3) (2007) 275–287.
[17] L. Paterson, T. Roscoe, The Design Principles of PlanetLab, ACM SIGOPS Operating Systems Review 40 (1) (2006) 11–16.
[18] L. Peterson, V. Pai, N. Spring, A. Bavier, Using PlanetLab for Network Research: Myths, Realities, and Best Practices, Tech. Rep. PDN–05–028, PlanetLab Consortium, 2005.
[19] P. Saint-Andre, RFC 3921, Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence, 2004. <http://www.ietf.org/rfc/rfc3921.txt>.
[20] A. Anadiotis, A. Apostolaras, D. Syrivelis, T. Korakis, L. Tassiulas, L. Rodriguez, I. Seskar, M. Ott, Towards maximizing wireless testbed utilization using spectrum slicing, in: Proceedings of TridentCom 2010, Berlin, Germany, 2010.
[21] Orbit lab. <http://www.orbit-lab.org/>.
[22] A. Anadiotis, A. Apostolaras, D. Syrivelis, T. Korakis, L. Tassiulas, L. Rodriguez, M. Ott, A new slicing scheme for efficient use of wireless testbeds, in: Proceedings of the 4th ACM International Workshop on Experimental Evaluation and Characterization, WINTECH '09, 2009, pp. 83–84.
[23] VNET+ subsystem of PlanetLab, <http://www.cs.princeton.edu/sapanb/vnet/>.
[24] S. Bhatia, VSys: A Privilege Allocation Tool, Tech. Rep., Princeton university, 2008. <http://www.cs.princeton.edu/sapanb/vsys/vsys.pdf>.
[25] Y. Liu, H. Zhang, W. Gong, D. Towsley, On the interaction between overlay routing and underlay routing, in: Proceedings of IEEE INFOCOM 2005, Miami, Florida, USA, 2005, pp. 2543–2553.
[26] V. Aggarwal, A. Feldmann, C. Scheideler, Can ISPs and P2P systems co-operate for improved performance?, ACM SIGCOMM Computer Communications Review (CCR) 37 (3) (2007) 29–40
[27] H. Xie, R. Yang, A. Krishnamurthy, Y. Liu, A. Silberschatz, P4P: provider portal for applications, ACM SIGCOMM Computer Communications Review(CCR) 38 (4) (2008) 351–362.
[28] M.K. Stuart Cheshire, K. Sekar, Nat Port Mapping Protocol (NAT-PMP), Tech. Rep., 2008. <http://www.files.dns-sd.org/draft-cheshire-nat-pmp.txt>.
[29] U. of Utah, P.U. Proposal, Statement of Work: Exploring Federation of Testbeds with Diverse Models, Tech. Rep., 2008. <http://www.geni.net/docs/dev_emu-plab-fed.pdf.
[30] R. Mahindra, G. Bhanage, G. Hadjichristofi, S. Ganu, P. Kamat, I. Seskar, D. Raychaudhuri, Integration of heterogeneous networking testbeds, in: Proceedings of TridentCom 2008, Innsbruck, Austria,, 2008.
[31] G.C. Hadjichristofi, A. Brender, M. Gruteser, R. Mahindra, I. Seskar, A wired-wireless testbed architecture for network layer experimentation based on ORBIT and VINI, in: Proceedings of ACM WINTECH 2007, Montréal, Québec (Canada), 2007, pp. 83–90.
[32] A. Bavier, N. Feamster, M. Huang, L. Peterson, J. Rexford, In VINI veritas: Realistic and controlled network experimentation, in: Proceedings of ACM SIGCOMM 2006, Pisa, Italy, 2006.

Giovanni Di Stasi is a Ph.D. Student at University of Napoli Federico II. He received the Laurea degree in Computer Engineering from University of Napoli Federico II in 2007. In 2008 he worked at CINI (Consorzio Interuniversitario Nazionale per l'Informatica) for the European Research Project ONELAB. In 2009 he was a visiting member of the ONELAB engineering group at INRIA (Sophia Antipolis, France) under the supervision of Dr. Thierry Parmentelat, technical director of the ONELAB project. In 2010 he was a visiting member of the Computer Networking Group at the Karlstad University, Sweden. His current research interests include experimental research infrastructures and testbeds, routing and channel assignment algorithms for wireless mesh networks, peer-to-peer traffic optimization, network emulation and simulation.

Roberto Bifulco is a Ph.D. Student at University of Napoli Federico II. He received the Laurea degree (cum laude) in Computer Engineering in 2008 from University of Napoli Federico II. In 2009 he worked at CINI (Consorzio Interuniversitario Nazionale per l'Informatica) for the European Research Project ONELAB2. Roberto Bifulco's current research interests include virtualization, network emulation, experimental reserch infrastructures and testbeds, cloud computing and network virtualization.

Stefano Avallone is Assistant Professor at the Department of Computer Engineering and Systems of the University of Napoli "Federico II". He received the M.S. degree in Telecommunications Engineering (2001) and the PhD degree in Computer Networks (2005) from the University of Napoli "Federico II". His research interests include computer networks, traffic engineering, QoS routing, wireless mesh networks. He was a visiting researcher at the Delft University of Technology (2003–2004) and at the Georgia Institute of Technology (2005). In 2004 he was awarded a research funding from the European Doctoral School of Advanced Topics in Networking (SATIN), the instrument employed by E-NEXT (an EU FP6 Network of Excellence) to invest in education of researchers for the European Research Area.

Roberto Canonico is Associate Professor at University of Napoli Federico II. He received the Laurea degree (cum laude) in Electronic Engineering from University of Napoli Federico II in 1995, and a Ph.D. in Computer Engineering from the same University in 2000. In 2000, he was a visiting member of the Distributed Multimedia Research Group at Lancaster University, UK. Roberto has been involved in several European Research Projects, such as CADENUS, INTERMON and ONELAB, and Networks of Excellence, such as E-NEXT and CONTENT. His current research interests include experimental reserch infrastructures and testbeds, network virtualization, cloud computing, peer-to-peer traffic optimization, overlay networks, network emulation and simulation.

Apostolos Apostolaras received the BS and MS degrees in Computer Engineering & Telecommunication Networks from Department of Computer Engineering and Telecommunications at the University of Thessaly Greece. He is a candidate PhD student at the same Department under the supervision of professor Leandros Tassiulas and is also a scholar of CERTH (The Centre For Research & Technology Hellas) . His research interests include wireless communication with applications in control & management of wireless testbeds. Moreover, he focuses on resource allocation techniques, power control and random access connectivity in wireless networks.

Nikolaos Giallelis received the diploma in Computer Engineering & Telecommunication Networks from the Polytechnic School of Department of Computer Engineering and Telecommunications at the University of Thessaly Greece in 2009. He is also a candidate PhD student in the same department. His research interests lie in the areas of wireless networking protocols, management of wireless experimental network facilities (testbeds). Moreover, he focuses on resource allocation techniques, and random access connectivity in wireless networks. Finally, he has experience in Web Applications and Database Management Systems.

Thanasis Korakis received the BS and MS degrees in informatics and telecommunications from the University of Athens, Greece, in 1994 and 1997, respectively, and the PhD degree in computer and communication engineering from the University of Thessaly, Greece, in 2005. Since 2005, he has been a research assistant professor in the Electrical and Computer Engineering Department, Polytechnic Institute of NYU. He has also been affiliated with the New York State Center for Advanced Technologies in Telecommunications (CATT) and the Wireless Internet Center for Advanced Technology (WICAT), Polytechnic Institute of NYU. Currently he is also a research scientist in CERTH, Greece. In the summer of 2004, he was a visiting researcher in the Computer Science and Engineering Department, University of California, Riverside. His research focuses on access layer protocols, cooperative networks, directional antennas, quality of service provisioning, and network management. He has served as a chair for ACM WiNTECH 2010, and TPC chair for TRIDENTCOM 2011. He is a voting member of the IEEE 802.16 standardization body and a member of the IEEE.

Leandros Tassiulas (S'89, M'91, SM/05 F/07) is Professor of Telecommunication Networks in the Department of Computer Engineering and Telecommunications at the University of Thessaly Greece since 2002 and research associate with the Center for Research and Technology Hellas (CERTH). He holds a Diploma in Electrical Engineering from the Aristotelian University of Thessaloniki, Greece, in 1987, and a Ph.D. degree in Electrical Engineering from the University of Maryland, College Park in 1991. He has held positions as Assistant Professor at Polytechnic University New York (1991–1995), Assistant and Associate Professor University of Maryland College Park (1995–2001) and Professor University of Ioannina Greece (1999–2001). His research interests are in the field of computer and communication networks with emphasis on fundamental mathematical models, architectures and protocols of wireless systems, sensor networks, high-speed internet and satellite communications. Dr. Tassiulas has been the principal investigator in several research projects funded by government and industry in USA as well as by European Commission. Currently he coordinates on behalf of CERTH the EU FP7 STREPs NCRAVE and OPNEX. He was the Greek National Expert on Telecommunications for IST in the 6th Framework Program. Dr. Tassiulas is a Fellow of IEEE while his research has been recognized by several awards including the inaugural INFOCOM 2007 Achievement Award for fundamental contributions to resource allocation in communication networks, the INFOCOM 1994 best paper award, a National Science Foundation (NSF) Research Initiation Award in 1992, an NSF CAREER Award in 1995, an Office of Naval Research Young Investigator Award in 1997 and a Bodosaki Foundation award in 1999.