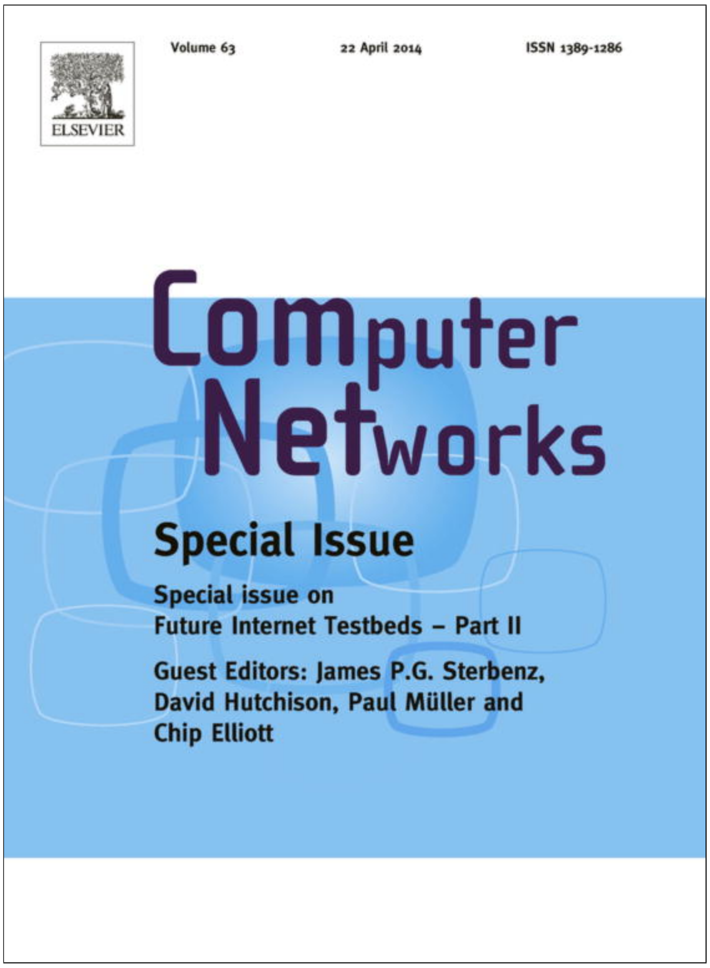

Volume 63
22 April 2014
ISSN 1389-1286
ELSEVIER
Computer Networks
Special Issue
Special issue on
Future Internet Testbeds – Part II
Guest Editors: James P.G. Sterbenz,
David Hutchison, Paul Müller and
Chip Elliott

Computer Networks 63 (2014) 48–67

Contents lists available at ScienceDirect

Computer Networks

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

Experimentation on end-to-end performance aware algorithms in the federated environment of the heterogeneous PlanetLab and NITOS testbeds ☆

Stratos Keranidis [a,b], Dimitris Giatsios [a,b,*], Thanasis Korakis [a,b], Iordanis Koutsopoulos [a,b], Leandros Tassiulas [a,b], Thierry Rakotoarivelo [c], Max Ott [c], Thierry Parmentelat [d]

[a] University of Thessaly, Argonafton & Filellinon, 38221, Volos, Greece
[b] Centre for Research and Technology Hellas, 6th Klm. Charilaou - Thermi Road, P.O. Box 60361 GR - 570 01 Thermi, Thessaloniki, Greece
[c] National ICT Australia (NICTA), Locked Bag 9013, Alexandria NSW 1435, Australia
[d] Inria Sophia Antipolis - Méditerranée 2004, Route des Lucioles - BP 93 06902 Sophia Antipolis, Cedex, France

A R T I C L E I N F O

Article history:
Received 2 July 2012
Received in revised form 20 December 2013
Accepted 26 December 2013
Available online 8 January 2014

Keywords:
Networks
Testbed
Federation
Association algorithm

A B S T R A C T

The constantly increasing diversity of the infrastructure used to deliver Internet services to the end user has created a demand for experimental network facilities featuring heterogeneous resources. Therefore, federation of existing network testbeds has been identified as a key goal in the testbed community, leading to a recent activity burst in this research field. In this paper, we present a federation scheme that was built during the Onelab 2 EU project. This scheme federates the NITOS wireless testbed with the wired PlanetLab Europe testbed, allowing researchers to access and use heterogeneous experimental facilities under an integrated environment. The usefulness of the resulting federated facility is demonstrated through the testing of an implemented end-to-end delay aware association scheme proposed for wireless mesh networks. We present extensive experiments under both wired congestion and wireless channel contention conditions that demonstrate the effectiveness of the proposed approach in realistic settings. The experiments are also reproduced in a well-established network simulator and a comparative study between the results obtained in the realistic and simulated environments is presented. Both the architectural building blocks that enable the federation of the testbeds and the execution of the experiment on combined resources, as well as the important insights obtained from the experimental results are described and analyzed, pointing out the importance of integrated experimental facilities for the design and development of the Future Internet.

1. Introduction

Wireless Mesh Networks (WMNs) are currently considered as the default solution for delivering high-speed Internet access to users within the last few network miles in non-urban areas. As a result, the interest of the research community in proposing WMN-related approaches has dramatically increased during the last few years. The inherent inability of simulation models to accurately estimate performance of wireless networks [2], in accordance with the unique characteristics introduced by the complex nature of WMNs [3] have directed research efforts towards implementation approaches and evaluation through experimentation in real world network scale and settings.

However, development of large scale WMN testbeds is a rather challenging task that requires careful design and

☆ Part of the material in this paper appeared in [1], in Tridentcom 2012.

* Corresponding author at: University of Thessaly, Greece. Tel.: +30 6973304560.

E-mail address: dimitris.giatsios@gmail.com (D. Giatsios).

1389-1286/$ - see front matter © 2014 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.bjp.2013.12.026

induces high deployment and maintenance costs. Moreover, as WMNs are usually considered as a promising technology for Internet access provision, experimentation across global scale networks that feature real Internet characteristics is required, in order to acquire experiment results under realistic congestion conditions. Such requirements have led the research community to the decision to develop a global large scale infrastructure that results from the federation of heterogeneous types of networks, such as wired (local, wide-area or optical) and wireless (local, mesh or sensor) networks.

Federation between inherently heterogeneous testbeds introduces several issues that arise due to the difference in the nature of experimental resources, but more importantly due to use of different software frameworks for resource management and control. In this work, we describe the establishment of the federation between two well-known heterogeneous network testbeds, namely the NITOS wireless testbed and the wired PlanetLab Europe (PLE) testbed. The utilization of a common experiment control framework, OMF [4] (cOntrol and Management Framework), and the adoption of the slice abstraction, as the building block for the federation, have been the main keys which made this federation possible.

In order to demonstrate the usefulness of the resulting integrated architecture, we develop and implement an association scheme for WMNs that is aware of end-to-end delay, part of which is generated in the wired section (PLE) and part in the wireless section (NITOS). The implemented mechanism is based on novel association metrics [5] that consider wireless channel contention, which are further enhanced, by taking into account delays incurred in the wired section of the end-to-end path. The evaluation of the proposed mechanism is performed through extensive experiments conducted on the combined network architecture, which results from the federation of the two heterogeneous experimental facilities. In an effort to explore the key differences and potential performance dissimilarities between realistic testbed and simulated environments in performing Mesh Network experimentation, we reproduced several of the testbed experiments in a well-established network simulator.

This paper is organized as follows. In Section 2 we discuss research work related with both association in WMNs and federation of heterogeneous experimental facilities. In Section 3 we describe the architecture of the two heterogeneous testbeds and moreover provide details about the approach followed and the tools used for the establishment of the testbed integration. In Section 4 we analyze and discuss the proposed association approach. The measurement methodology followed in our experiments, as well as a brief experiment description that is OMF compatible are presented in Section 5. In Section 6 we present and comment on the results obtained from the experimental evaluation of the implemented mechanism. Results obtained through the simulation of the real experiments, along with achieved user experience are discussed in Section 7. Finally, in Section 8 we summarize our work, by pointing out conclusions and directions for future work.

2. Related work

2.1. Association in wireless mesh networks

WMNs are composed of Mesh Routers (MRs), which form the wireless backhaul access network and Mesh Clients (MCs). MRs forward packets acting as intermediate relay nodes and may also provide wireless access services to MCs, in which case they are referred to as Mesh Access Points (MAPs). WMNs also consist of Internet Gateway nodes (IGWs) that provide Internet access to the network, through direct connection to wired infrastructure. MCs associate with a certain MAP in order to access the network and do not participate in packet forwarding.

The affordable cost and ease of deployment of IEEE 802.11 compliant equipment has led the majority of WMNs to be based on conventional IEEE 802.11 devices, although this does not limit the potential application of other standards. According to the IEEE 802.11 standard, which was originally proposed for infrastructure Wireless Local Area Networks (WLANs), MCs perform scanning to detect nearby MAPs and simply select to associate with the MAP that provides the highest Received Signal Strength Indication (RSSI) value. The performance of the standard association policy has been extensively studied [6] in the context of IEEE 802.11 WLANs and it is well known that it leads to inefficient use of the network resources. In WMNs, the entire path between the MC and the IGW is composed of two discrete wireless parts: the single-hop access link between the MC and the MAP it is associated with and the multi-hop backhaul part that connects the MAP with the IGW. As the standard policy considers only factors affecting performance on the wireless access link, its direct application on WMNs becomes inappropriate. As a result, more sophisticated association schemes are required to capture performance achieved in both the access and the backhaul network parts.

Trying to address the issues generated by the unique two-tier architecture introduced by WMNs, several approaches on MAP selection have been proposed in the recent literature. An innovative cross-layer association mechanism that considers not only the access link but also routing in the multi-hop backhaul part is proposed in [7]. The authors in [8] consider also the interaction of physical (PHY) layer transmission rate with the packet size and hop count and propose a signaling mechanism through which information about congestion on both parts is passed from the MAPs to the MCs. In [9], a new metric is proposed that takes into account the impact of 802.11 MAC layer contention on bandwidth sharing and results in accurate link throughput estimations. Another approach, proposed in [10], considers also estimation of real-time traffic load conditions trying to cope with the variability of network conditions, which is an inherent characteristic of WMNs. The common characteristic of the works referenced above is that they rely only on simulation-based evaluation of the proposed mechanisms.

Recent research studies in the field of WMNs jointly consider problems that traditionally were considered in isolation, such as association and routing. However, as simulation models [2] are not not able to capture the interaction among different layers [3], research related to WMNs is mainly performed in experimental facilities. A recent work in the field [11] proposes a cross-layer association mechanism, which is implemented and evaluated through experimentation in a wireless testbed. However, the evaluation of the implemented scheme is restricted in experiments conducted in a small scale testbed composed of conventional laptop computers and not in a customized large scale Mesh testbed.

At this point, we argue that approaches proposed for WMNs should be fully implemented and properly evaluated through extensive experimentation under real interference and congestion conditions. In an effort to support realistic and large-scale experimentation with heterogeneous network platforms, both the GENI initiative in the U.S. [12], as well as the FIRE initiative in Europe [13] are currently investigating federation of heterogeneous testbeds.

2.2. Integration of heterogeneous experimental facilities

An initial effort on federation of testbeds was proposed in [14], where the wireless EmuLab testbed and the wired planetary-scale PlanetLab testbed [15] were integrated through the *EmuLab-PlanetLab portal*. The integrated interface provided useful extensions to the PlanetLab's management system. Moreover, several integration challenges were identified for the first time and appropriate solutions were provided. Another work, proposed in [16], aimed at integrating PlanetLab with the ORBIT wireless testbed. The authors considered also the ability of performing experiments on the integrated framework concurrently. Although the PlanetLab testbed provided support for virtualization of resources in the wired part, virtualization of the wireless part had to be further investigated in order to overcome the issues that the broadcast nature of the wireless medium generates. Two discrete integration models were proposed in this work, where the first one aimed to support PlanetLab users in extending their experimental topologies with wireless nodes, while the second one was introduced to provide users of the ORBIT testbed with the extra ability of adding wired network extensions to their experiments.

An important issue that the aforementioned federation approaches had to cope with was the scarcity of a common management system, as well as a common experiment description language. However, this issue was overcome with the introduction of OMF, which provides tools for the management and execution of experiments on testbed infrastructures. Nowadays, OMF has been deployed and used on multiple testbeds supporting many different types of technologies. The work proposed in [17] presented the integration of an OMF-controllable WiFi testbed and PLE, through the addition of an extra wireless interface in PLE nodes that were located within the range of the wireless testbed. This integration was achieved through the development of special tools that supported the definition of slice-specific routing table rules and the exclusive use of the wireless testbed by a single experimenter. Although this integration attempt provided an integrated environment, where all resources could be instrumented through OMF, it also faced the drawback of realizing the wireless testbed as a single resource and thus limited the access of the federated environment to a single user for each reservation slot.

In an effort to maximize the utilization of OMF-based experimental facilities, NITOS introduced a testbed Scheduler [18] that enables the assignment of different subsets of nodes and channels to different users during specific reservation slots. The work in [19] proposes an integration architecture, which combines an OMF-based wireless testbed supported by the NITOS scheduling mechanism with several PLE OMF-enabled nodes. The resulting federated environment formed a realistic global-scale WMN that supported the execution of multiple concurrent experiments, through the NITOS Scheduler. Moreover, the authors demonstrated an experimental scenario that provided interesting insights regarding real-world experimentation with peer-to-peer systems.

3. OneLab federation of NITOS and PlanetLab

OneLab [20] is an initiative to provide an open, general-purpose experimental facility, aimed at promoting innovation among network and ICT researchers in Europe, both in academia and industry. It is primarily based on the results of two EC FP7 projects, namely the Onelab and Onelab2 projects. One of the most important goals of the initiative is to establish a federated environment between different, possibly heterogeneous testbeds. As several testbeds have been deployed independently by research institutions across Europe during recent years, and each of them has developed or adopted a different control and management framework, a complex and inconvenient mosaic arises. In this mosaic, experimentation in different testbeds implies familiarization with the respective control frameworks, while combined experiments between different facilities are extremely difficult to setup. The federation between NITOS and PLE, two testbeds of entirely different architecture, which took place during the Onelab2 project, demonstrated that through agreements and collaborations among the involved administrative entities, it is possible to establish architectural paradigms that allow for combined experiments across heterogeneous platforms. In this Section, after describing the two facilities, we analyze the components of the federated environment, which allowed for a combined experiment.

3.1. PlanetLab Europe

PlanetLab Europe is the European portion of the publicly available PlanetLab testbed, a global facility for the deployment of new network services. It is tightly federated with PlanetLab Central, offering a total of 1000+ nodes worldwide. Each node runs a set of isolated virtual machines that run either internal PlanetLab services (like e.g. for accountability purposes), or widely and publicly available services (like e.g. CoDeen for content distribution), or more narrow-term experiments that focus on a specific networking environment. PlanetLab has been a pioneering system with respect to federation. PlanetLab Central and PlanetLab Europe have been running the facility in a decentralized manner for more than 6 years now, even if this relies on a PlanetLab-specific paradigm, that is currently being replaced by the much more general Slice-based Facility Architecture (SFA) paradigm [21].

3.1.1. PlanetLab Europe capabilities

In addition to regular infrastructure servers, PlanetLab Europe offers a few unique features, including:

- Reservable nodes: PLE exposes a few of its nodes as reservable, which means that these nodes only run one user slice at any given time. As a result, the experimenter needs to reserve the nodes at specific timeslots, except from adding the corresponding slice to the reservable node, so that the corresponding virtual machine shows up accordingly.
- Topology management: PLE also provides basic tools for implementing overlay topologies based on tunnels, either inside a PlanetLab slice or towards the outside world, wherever a tunnel can be terminated.
- Wireless nodes featuring WiFi capability; this can come in handy when setting up an experiment that involves end users connected through wireless links; for more focused studies of WiFi, dedicated testbeds are of course more relevant.
- An embedded emulation tool, named *Dummynet* [22], which is able to perform basic actions like packet dropping or delaying and also embeds various predefined models so as to treat a link like e.g. a Wifi link.

3.1.2. Experimental plane tools

Historically, PlanetLab was known to not provide many features in terms of experiment control; given that every virtual machine is accessible through *ssh*, it is of course possible to resort to wide-purpose ssh-based tools like *pssh*. Over the recent years, more tools have been added, for much more powerful control of experiments. Among these, let us quote the OMF Experimental Controller tool, which enables experiments to be entirely controlled through the OMF experiment control framework. Another example of an experiment management tool is *NEPI* [23], which has the ambition to support a wide range of hybrid simulated/emulated/real experiments, based on the use of a custom developed PlanetLab driver. As an example, creating topologies within PlanetLab is definitely much simpler with *NEPI* than it would be by using the basic PlanetLab tools.

3.1.3. Slices/slivers

The notion of a slice is a rather central notion in PlanetLab; it typically allows the modeling of resource allocation, by relating a set of users and a set of resources (nodes). Once created, the slice "owns" one private server (sliver) on each of the selected nodes, and to the designated users, being part of the slice means UNIX shell access to all these slivers. The PlanetLab software is tailored for smoothly orchestrating a complex workflow that involves a large number of people, with different roles (from the legal paperwork, down to locally vouching for users and remote IT management); it also needs to deal with accountability of the resulting network traffic, especially given its scale and diversity of usages, that by design often leads to untypical shapes of traffic; but it admittedly offers little help in managing a slice, and encourages users to leverage third-party tools for the actual experimentation phase.

3.1.4. MyPLC

MyPLC [24] is the software that was packaged by the PlanetLab operators to let others run their own private PlanetLab system. It was created by Princeton University and is currently being codeveloped by Princeton and OneLab partner INRIA. It provides a ready-to-install set of packages, for both infrastucture-side (XMLRPC API, with related database, software server for securely booting and upgrading nodes), and node-side (slivers management, accountability, remote operations and monitoring). MyPLC is rather flexible, and several tens of instances of MyPLC have been deployed around the world, either for entirely local testbeds, or at the scale of a research consortium.

3.2. NITOS testbed

NITOS is a wireless testbed featuring 50 WiFi-enabled nodes that are deployed outdoors in the premises of a University of Thessaly campus building. It is remotely and publicly accessible to any researcher wishing to use its resources, after a plain registration procedure and its approval by the testbed administrators. Below, we describe the two basic software entities of the NITOS testbed, OMF/OML and NITOS Scheduler.

3.2.1. OMF/OML

NITOS has adopted OMF as its testbed control and management framework. The architecture of OMF is based on three main software components: the Aggregate Manager (AM), the Experiment Controller (EC) and the Resource Controller (RC). The AM provides a set of services to the testbed (inventory, image loading, etc.). The EC, which is the user's interface, receives and parses an experiment script describing configuration of resources and the actual experimental scenario. This script is written in a domain-specific language called OEDL (OMF Experiment Description Language). The instructions in the script are transferred to the RCs of the respective resources, which are responsible to perform the local configurations and application invocations. The different components communicate asynchronously through an XMPP publish-subscribe system, where each message is transferred to an XMPP server, which relays it to its intended destination.

OML (OMF Measurement Library [25]), a companion framework for OMF, is responsible for handling measurements. It consists of two architectural components, the OML server and the OML client libraries. The client libraries are responsible for capturing measurements generated at the resources and, possibly after some manipulations, injecting them in streams headed towards the OML server. The OML server receives the data and stores them in organized databases, one per experiment. The measurements are automatically timestamped; thus it is straightforward to plot them against time. An extremely user-friendly web-based graph application is also associated with OML. The user can point his browser to a specific address and port, and be able to see the evolution of the variables being measured at real time. The exact set of variables to be monitored is specified in the OMF experiment description file.

The typical requirement for the user when using OMF/OML is to develop OMF/OML wrappers for the applications that will run on the resources during the experiment. The role of these wrappers is twofold. First, they act as proxies translating shell commands to a Ruby-based format that can be parsed by OMF. Second, specific output variables of the source application are linked with the OML client library, so that their values can be sent to the OML server during the experiment execution. Wrappers for several popular network testing applications have already been developed and are typically available at testbeds featuring OMF support. In cases where a user is not satisfied by this set of applications and wants to develop additional wrappers, the procedure is very simple, and it is documented in detail on the OMF website [26].

3.2.2. NITOS scheduler

NITlab has developed a reservation and access control software tool for NITOS, called NITOS Scheduler. This tool provides a web-based reservation front end for users of the testbed, accessed through the NITOS website, and a set of back end scripts, daemons and services responsible for controlling access to the testbed's resources, according to the corresponding reservations. In the front end, the user reserves a set of nodes and frequencies for a time duration not exceeding four hours, with a half-hour granularity. In the Scheduler's back end, two main functionalities are worth mentioning: the Scheduler's interaction with the XMPP framework used in OMF and the Spectrum Slicing framework. Both of them are utilized to enable slicing of the testbed, that is, to enable simultaneous experimentation by multiple users through allocation of disjoint sets of resources. Each user at NITOS is associated with a slice. Unlike the typical PlanetLab setup, where slices imply the existence of virtual machines, at NITOS a slice is an abstract entity. In Table 1 we can see a snapshot of part of the NITOS XMPP PubSub nodes tree-like structure, as observed at a given time instant. For each slice in NITOS, the PubSub nodes /OMF/<slicename> and /OMF/<slicename>/resources are always present. When a reservation for a NITOS resource starts, an additional PubSub node /OMF/<slicename>/ resources/ <resource_name> is created. When this reservation ends, this entry is deleted. As a result, access to NITOS nodes is restricted through this dynamic association and disassociation of resources to the corresponding slices, based on queries to the NITOS scheduler's database. Since all OMF communication takes place via the XMPP protocol [27], this mechanism is equivalent to OMF-based dynamic access control of the NITOS resources. Indeed, when the PubSub node /OMF/<slicename>/resources/ ResourceID is not present, users of slice <slicename> cannot access the RC of resource ResourceID and, therefore, they cannot send OMF commands to it. The addition and removal of PubSub entries is controlled by a backend daemon, running every half an hour, so that the reservation data as stored in the Scheduler's database exactly matches the granted access rights.

Table 1
Instance of NITOS XMPP PubSub nodes structure.

NITOS XMPP PubSub records
"/OMF/efkerani"
"/OMF/efkerani/resources"
"/OMF/efkerani/resources/omf.nitos.node023"
"/OMF/efkerani/resources/omf.nitos.node022"
"/OMF/efkerani/resources/omf.nitos.node019"
"/OMF/efkerani/resources/planetlab.test.upmc.ple2"

Regarding spectrum slicing, when users submit their reservation, they are only allowed to reserve frequencies that are orthogonal to the ones selected by other users, who have made a reservation for the same or an overlapping time interval (802.11a frequencies are also available, so that sufficient flexibility is achieved). The Resource Controller software provided by NITOS has been modified to allow only frequency configuration requests that are compliant with the existing frequency reservation data in the scheduler database. In this way, cross-interference among simultaneous OMF experiments is avoided.

3.3. Federation framework

In this subsection, we describe the basic components of the federated environment between NITOS and PLE. The development of these components took place during the project Onelab2 and enabled, from an architectural point of view, the conduction of the experiment presented in this paper.

3.3.1. Single sign up

One important characteristic of a federated environment is that a user of such a facility should not be obliged to register at its different components separately, but instead be able to use common credentials. To achieve this operation between NITOS and PLE, a single-sign up mechanism has been developed, so that any user of PLE can log into the NITOS portal without going through any extra registration process. This single-sign up process is based on PlanetLab's standard XMLRPC user authentication API. In particular, when a user attempts to log into the NITOS portal, providing a username and a password, the portal's underlying code not only tries to match the credentials with an entry from the native user database, but also contacts the authorization server of PLE through the standard API. If a match is found among PLE's users, an affirmation is sent back to NITOS, which then automatically generates (in the case that id does not already exist) a slice in NITOS having the name of the PLE slice and moreover logs him in the NITOS portal with the provided credentials. The process is transparent to the user and incurs no significant delay. An improved procedure of single sign up will be offered through the SFA federation framework, making use of the SFA Registry service [28]. This is a coherent and scalable solution. It is gradually being incorporated in several testbeds in Europe, and it is expected to replace earlier schemes, such as the aforementioned solution. This service will be accessed with an SFA client application. Currently, the most advanced SFA client available is MySlice [29]. This graphical client application will also provide the capability to reserve resources from multiple federated testbeds, along with other attractive features. MySlice can already handle reservations of PLE nodes, while an enhanced version of NITOS Scheduler will be incorporated in this tool soon.

3.3.2. Deployment of OMF/OML at PLE resources

A major difficulty when trying to run combined experiments using heterogeneous facilities is that different languages are used to describe resource configurations and actions. There is the need for agreement to use a common language for experiment description (ED), which must be able to handle the broadest range of resource types possible and easily add support for new resource types in a modular fashion. OEDL is a perfect candidate, as it meets these requirements.

Therefore, PLE decided to incorporate OMF support on demand, in the form of so-called 'OMF-friendly' slices. For slices with this tag activated, at slice-creation time, a pre-installed and pre-configured OMF Resource Controller is initiated in the related slivers. In this way, a PLE resource can be viewed as any other resource of an OMF-based testbed and it can be configured through instructions issued by the experimenter in an experiment script written in OEDL.

3.3.3. XMPP communication using slices

OMF employs a Pub-Sub asynchronous communication scheme between the EC and the RCs running on the resources, based on the XMPP protocol. In order for all the resources to be able to communicate with the EC, they must be registered in the same XMPP server or in a set of XMPP servers peered with each other.

In the case that the first approach is followed, nodes belonging to the remote testbed should feature public IPs addresses, so that efficient communication with the single XMPP server is feasible. This works in the case of PLE nodes, but not for NITOS nodes that use private IP addresses. By following the first approach, experiments can only be initiated by NITOS nodes, while users have to configure RCs of the local testbed resources to enable communication with the remote XMPP server. In order to provide for a smoother node communication and measurement collection procedure, we adopted the second option, where each resource talks to its local XMPP server, and messages are transferred to their final destination via the XMPP server-to-server (S2S) protocol. The added value is that experimenters do not have to proceed with configurations of the local testbed RCs,and thus the overall procedure becomes totally transparent to the end-user. Therefore, the NITOS and PLE testbeds enabled incoming server-to-server connections in their local XMPP servers, by performing some relatively simple configuration tasks.

In a federated OMF experiment, the PubSub nodes related to a single slice all reside on one of the XMPP servers, which acts as the "rendezvous point" for all participants in the slice. Messages published for that node by users connected to other XMPP servers are relayed to that host via the server-to-server protocol. This architecture is depicted in Fig. 1. The user can freely decide about where the corresponding slice will be hosted. In the case of the federated experiment described in this paper, we only had two XMPP servers, and we decided to host the slice PubSub nodes in the NITOS server.

When this communication procedure is followed, the Resource Controllers of the nodes to which the messages have to be relayed using the S2S protocol, must be configured to use the same XMPP domain, which corresponds to the server hosting the slice's PubSub nodes. In particular, in the RC configuration files of the PLE resources used in our experiment, we had to set the parameter *xmpp.domain* to the value *nitlab.inf.uth.gr*, which is the NITOS XMPP server that hosts the experiment's slice.

It must be noted here that OEDL scripts are completely independent and ignorant of slices. In an experiment description, each resource is referred to by its unique HRN, e.g. *omf.nitos.node019*. In this way, experiment description scripts can easily be reused by different slices. In Section A of the Appendix, we present a code snippet example that shows how resource names are used in the OEDL script of our experiment.

3.3.4. Advantages of experimenting in the federated environment

The use of the aforementioned tools provides the user with a number of benefits, compared to the case where he attempts to conduct an experiment combining resources from the two testbeds without the existence of federation tools. Below we summarize these benefits, organized according to the expected workflow associated with the user experience.

No federation:

- *User authorization*: The user needs to create and maintain one account per testbed
- *Resource reservation*: The user needs to reserve resources from each testbed separately, potentially through different tools
- *Experiment setup and orchestration*: Considerable expertise in network programming and extensive effort is needed to coordinate distributed actions. Moreover, potentially existing and incompatible frameworks are likely to create obstacles in coordination.
- *Measurement collection*: Incompatibilities in format of measurements lead to extensive post-processing. Extensive scaffolding development work is required to collect and synchronize measurements.

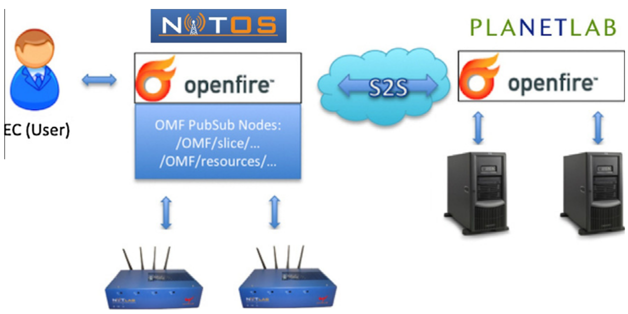


Fig. 1. XMPP server to server peering in OMF.

NITOS-PLE federation:

- *User authorization*: User only needs a single account
- *Resource reservation*: No federation in this aspect was present at the time of the experiment presented in this paper. However, the MySlice SFA client tool, which already works with PLE and is being incorporated into NITOS, will soon provide the capability of multi-testbed resource reservation from a single point.
- *Experiment setup and orchestration*: OMF does all the background orchestration work for the user. Moreover, a single experiment description file using a single common language is used for description of the distributed experiment, leaving no room for inconsistencies.
- *Measurement collection*: Common format for measurements, directly parsable even at real time, removes the need for post processing of measurements. Trivial development effort is required for inserting custom measurement points into applications.

4. Proposed association mechanism

In order to demonstrate the usefulness of the federated environment that combines the wired PLE with the wireless NITOS testbeds, we developed a novel association mechanism proposed for WMNs that is end-to-end performance aware. In this Section, we describe the developed association mechanism and moreover provide details about its driver level implementation.

4.1. System model and metrics definition

As end-to-end performance in WMNs depends also on the performance experienced on the wireless backhaul part of the network, as well as on the wired infrastructure on which the IGWs are connected, both factors are taken into account by the proposed mechanism to provide for efficient associations. In this work, we consider a special case of WMNs that do not feature a wireless backhaul part, but are composed of MAPs that are directly connected to the wired infrastructure and thus operate as IGWs. A representation of the described topology is illustrated in Fig. 2.

Each MC chooses to associate with a single MAP among the MAPs that operate in its vicinity. Each network node n has a set of neighbors, that reside in its sensing area and operate on the same channel with n. This set of "1-hop" neighbors, that can be either MAPs or MCs, is denoted by A_n. In our previous work [5], we decided on two discrete throughput based metrics for uplink and downlink communications that conform with the special case of infrastructure 802.11 networks. In this work, we consider only the case of uplink communications, which provides for a simple analysis of the proposed mechanism. In uplink communications, frames are transmitted by each MC and destined to the specific AP it is associated with. In our previous work, we arrived at an expression that considers the medium sharing of each MC_i with its "1-hop" neighbors (A_i) and estimates throughput on uplink as follows:

$$T_{ij}^{up} = \frac{1}{\frac{f_i}{R_{ij}} + \sum_{k=1}^{|A_i|} \frac{f_k}{R_k}} \tag{1}$$

where R_{ij} and R_k denote the PHY rates used by MC_i and each node $k \in A_i$ accordingly, while f_i and f_k are defined as activity indicator factors reflecting the activity intensity of MC_i and node $k \in A_i$ in comparison with each other.

4.2. End-to-end performance aware association mechanism

Based on the analysis in the previous Section, we are able to estimate throughput performance for the single-hop access link between the MC and each potential neighboring MAP. More specifically, the denominator of expression 1 estimates the average time duration required for a single bit of information to be transmitted over the access link. In this work, we develop an association framework that is based on Round-Trip Time (RTT) measurements. Readers interested in related work including comparative study of more sophisticated association metrics are directed to our previous work [5]. In this paper, we estimate the RTT required for the initial transmission and subsequent retransmissions of a frame with specific length, by multiplying the calculated delay with the number of bits that are transmitted over the access link and moreover double the resulting value to estimate the total delay required for both transmissions. Concluding, we estimate the RTT for a specific frame of M bits that is transmitted over the access link from MC_i to MAP_j and back again, as follows:

$$RTT_{ij}^{A} = 2 * M * \left(\frac{f_i}{R_{ij}} + \sum_{k=1}^{|A_i|} \frac{f_k}{R_k} \right) \tag{2}$$

In our approach, we develop a simple mechanism to estimate RTT experienced on the wired backhaul part of the network as well. More specifically, each MAP_j periodically transmits probe packets and measures RTT_j^B for the wired

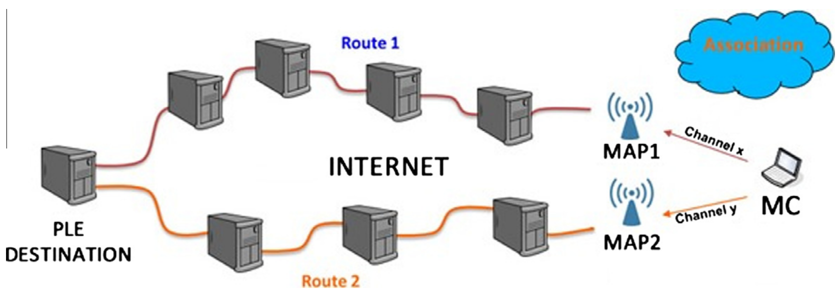


Fig. 2. Topology representation.

network backhaul. These values are broadcasted to all MCs in range and as a result each MC_i is able to estimate end-to-end RTT for each potential MAP_j, as follows:

$$RTT_{ij}^{total} = RTT_{ij}^{A} + RTT_{j}^{B} \quad (3)$$

4.3. Implementation details

For the implementation of our mechanism, we used the Mad-WiFi open source driver. Details about the mechanism aiding in performance estimation on the wireless part can be found in our previous work [5]. In this Section we will provide details about the developed mechanism that enables application layer information regarding wired RTT information to reach neighboring MCs. First of all, we use a simple application level program that runs at the APs and sends probe packets to the destination host to calculate RTT_j^B values. In order to broadcast this information to all neighboring MCs, we first had to make the RTT_j^B value available to the kernel level, as all MAC layer mechanisms are implemented as loadable kernel modules by the MAD-WiFi driver. An efficient way to transfer information to the kernel is through the proc virtual filesystem, which resides in the kernel memory. The proc files used by the MAD-WiFi driver are stored in the folder location /proc/sys/net/wlan/athX, where X denotes the specific interface. Another script running locally at the APs periodically writes values to the specified proc file and as soon as a new record is written the driver is informed. As for the next step we had to broadcast the RTT_j^B value to all neighboring MCs. In order to do this, we extended the *Beacon* and *Probe-Response* frames to carry this information. This frame extension does not affect the normal operation of the 802.11 protocol, as these frames feature a dynamic part that supports extension, according to the standard [30]. The MCs constantly estimate the RTT_{ij}^A values for each potential MAP. The third step is also performed at the MC side, where the driver combines the RTT_j^B value with the RTT_{ij}^A and calculates the RTT_{ij}^{total}. Finally, each MC_i associates with the MAP_j that features the lowest RTT_{ij}^{total} value.

5. Experimental setup

In this Section, we provide details about the measurement methodology and the various tools that we used in order to establish a proper experimental setup. Moreover, we provide a brief description of the OEDL ED script that we used for the orchestration of the conducted experiments.

5.1. Measurement methodology

Fig. 3 represents the actual topology used in our experiments. We consider a typical scenario, where one traffic flow is generated from the MC node and relayed through the two available MAPs to the final PLE destination node. The MAPs act as IGWs and get access to the wired network part through NITOS Server. As NITOS nodes are assigned private IP addresses, we had to enable a Network Address Translation (NAT) service at NITOS Server through proper IPtables [31] configurations, in order to provide Internet access to the two nodes operating as MAPs. We also followed a similar procedure to provide for proper relaying of traffic generated by the MC through the two MAPs.

In a part of our experiments, we required the injection of artificial delay in the wired backhaul link. For this purpose, we used the *Dummynet* [22] tool, which is able to emulate queue and bandwidth limitations, delays, packet losses, and multi-path effects, by intercepting packets in their way through the protocol stack. As an outcome of the OneLab project, PLE natively supports *Dummynet* as a kernel module in all nodes, configurable from the sliver through a command-line tool. Since it is straightforward to wrap shell commands inside the OMF experiment description file, we used this approach to avoid having to log into the remote PLE node and issue the commands manually. Furthermore, incorporating the *Dummynet* commands inside the OMF script allows perfect synchronization with the rest of the applications and with the measurement collection process.

As all packets received at the destination, share the same IP address of the NITOS Server, we base packet discrimination on the port numbers. To this aim, we used a simple *Nmap* [32] script to dynamically detect the specific ports used for incoming connections at the PLE node. For the wireless part, we enable a pair of nodes that operate on each of the channels used by the MAPs and generate channel contention conditions of varying traffic rate. Fig. 4 shows a screenshot of the OMF visualization tool representing delay emulation for two discrete flows generated

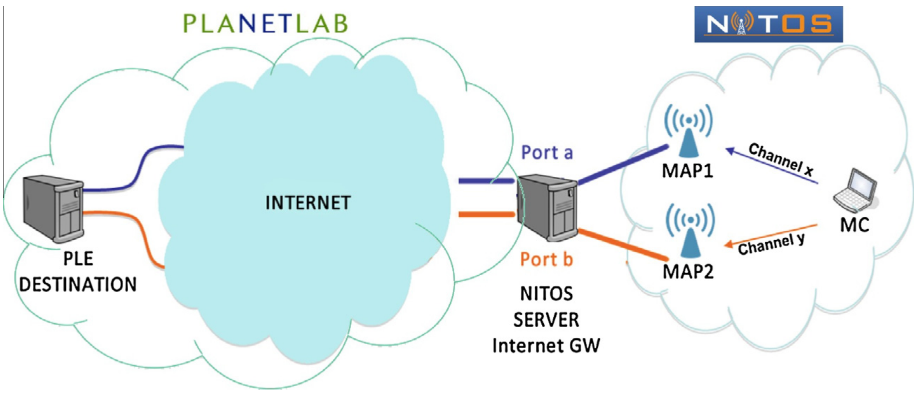


Fig. 3. Experimental topology representation.

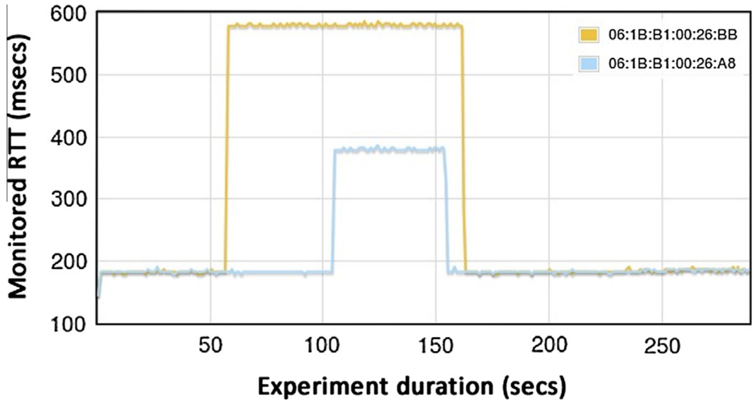


Fig. 4. RTT monitored at the PLE for two flows generated by each MAP.

by each one of the MAPs, as monitored at the PLE node. The throughput performance of the experiments is measured by using Iperf [33]. In our experiments, we run an Iperf Client at the MC to generate TCP flows and UDP flows of varying rate and also an Iperf Server residing at the PLE node to receive traffic and collect the corresponding measurements.

Considering the video experiments presented last in Section 6, we used the *Video LAN Player (VLC)* [34] to stream a video of H.264 format from the MC to multiple PLE destination nodes. In order to provide for a simple experimental setup, we run both the VLC Server and Client applications on the MC node and also utilized the *Socat* [35] platform to be able to relay, through UDP sockets, all packets that are received on a specific port at the PLE node back to the MC.

5.2. Experiment description

We described our experiment with the OMF Experiment Description Language (OEDL) script. An OEDL script is basically comprised of two parts. The resource configuration part and the experimental scenario part. In the first part, configuration parameters for both the network interfaces of the resources and the applications to be executed on them are determined, while the measurements to be collected are also defined. A brief code snippet from our ED, which describes the configuration of the first MAP, is presented in Section B of the Appendix.

We place a single resource in the group called 'MAP1', determine two applications to run on this resource (OMF wrappers for these applications must be present in the resource's image), and we also define values for the properties of the wireless interface to be brought up and used. For the purposes of the experiment, we used some existing OMF wrappers for common applications, but also developed some new ones for the specific needs of our experiment. For instance, we developed a wrapper for the *iwconfig* application, which is used to dynamically identify the current association of the MC, for visualization purposes.

In the second part of the ED script, the exact course of actions to take place is determined. The timing of the different tasks is described, i.e. when to start the applications on a resource, how long they will run, or when they will stop. Moreover, we are able to define the exact time that specific events will take place, e.g. the injection of artificial delay that is executed through the call to the *Dummynet* application. Considering the *Dummynet* application, we did not develop a custom wrapper, as there is no need for gathering any measurements from this application. The *Dummynet* application is called using the exec command, which in this case is applied to the group of the PLE node, using standard delays defined at the beginning of the experiment. Based on the custom developed *Nmap* script, we are able to dynamically detect the specific ports used for incoming connections at the PLE node and we feed the output of this script as an input parameter to the *Dummynet* application, which then injects delay solely to flows that use the specified ports. Representative code snippets that are used to configure both the delay attribute that are used by the *Dummynet* applications, as well as to instruct the *Dummynet* application, are presented in Section C of the Appendix accordingly.

The RTT measurement applications are Ruby applications, that use the client–server model. The client side is installed at the MAPs and the server side is installed at the PLE node. Since the applications we used are written in Ruby, we used the OML library, in order to inject the measurements gathered during runtime i.e. RTT values, to the sqlite experiment specific database that is created by the OMF. More details regarding the experimental description, as well as all the important applications and scripts are publicly available and can be downloaded directly from the NITOS web portal [36].

6. Experimental evaluation

In this Section, we evaluate several experimental scenarios that aim at presenting the abilities offered by the federated testbed environment, rather than presenting innovative research results. More specifically, these experiments have been designed to demonstrate the effectiveness of association mechanisms that jointly consider factors affecting both wireless and wired performance. The execution of such combined experiments requires integrated testing, which would not be feasible without the existence of the federated environment.

The conducted experiments are organized in two parts, where in the first part we measure the performance for a static topology, where the MC communicates with a single MAP that is forwarding traffic to a single PLE destination. Under this topology, we focus on measuring the resulting performance under various settings of artificially injected delay in the wired part and channel contention in the wireless access link as well. In this first part of experiments, we decided to vary the wired delay artificially and not by using different network paths, in order to isolate the impact of other factors, such as traffic congestion that would significantly vary among different network topologies.

In the second part of our experiments, we do not alter any settings in the wireless network part, but we focus on testing the performance of the proposed scheme under realistic RTT variations in the wired part, which conditions are generated through the use of different PLE nodes as the final path destinations. In order to accomplish this, we use multiple PLE nodes that span across different continents and moreover offer different load conditions, thus featuring diverse end-to-end characteristics for the wired part.

6.1. Experimentation based on a single PLE node

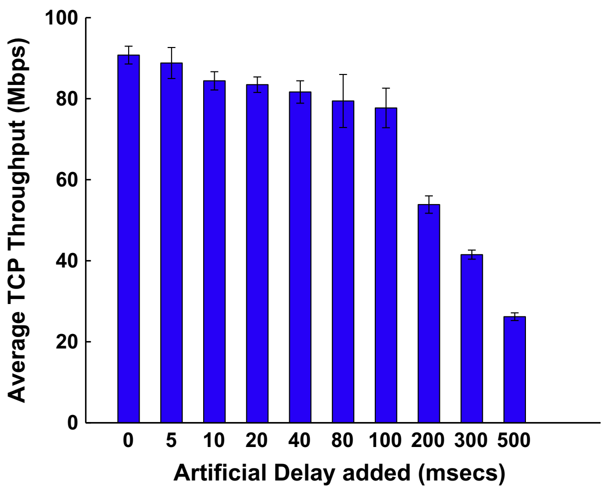
The experiments described in this first part are organized in two sets, where in the first set we generate conditions of varying delay in the wired backhaul part, while in the second set we vary the delay in the single hop wireless access link. Moreover, each experiment is performed in two discrete phases, where in the first one we compare the effect of injected delay on performance affecting either the wired (1st set) or wireless (2nd set) part solely, while in the second phase we consider the impact on the combined network architecture.

The conducted experiments aim at presenting the performance improvement that can be offered through the application of the proposed association mechanism and thus measure the performance for a static scenario, where the MC communicates with a specific MAP. Under this scenario we alter the delay induced in each part and monitor the resulting performance in terms of TCP/UDP throughput, packet loss and jitter values. The initial RTT in the wired part between NITOS Server and the PLE node that is used in these experiments and resides in France is around 80 ms, while in the wireless part the reported RTT between the MC and each MAP without any external contention is below the value of 1 ms. In all the conducted experiments, the default Rate adaptation algorithm of the driver has been used.

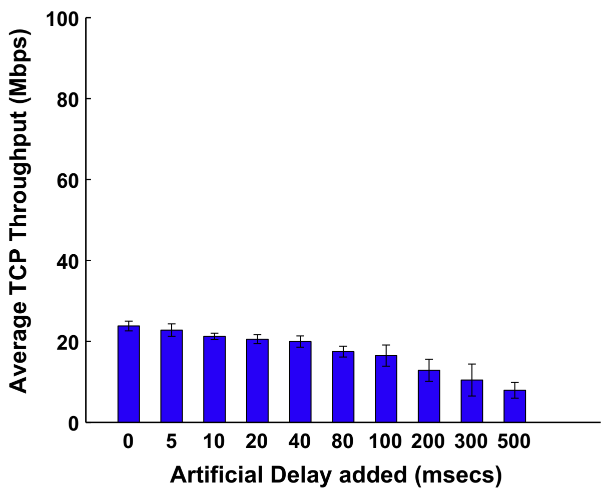
6.1.1. Wired – Combined set of experiments

In this first set of experiments we use the *Dummynet* tool to generate artificial delay. Fig. 5(a) and Fig. 5(b) illustrate the TCP throughput achieved under various artificial delay values in the wired and the combined architectures accordingly. We notice that even small variation of delay in the wired part significantly affects TCP throughput and therefore should be taken into account. Moreover, we notice that the wireless access link acts as the performance bottleneck that significantly limits yielded performance. A particular observation is that the same experiments provide higher deviation values when conducted in the combined topology, for the cases of 200, 300 and 500 ms of injected delay, in comparison with the execution solely in the wired part. However, average throughput values show similar performance in the above cases. Based on the observed results, we remark that relatively high values of injected delay make TCP performance in the combined network highly unstable.

In Fig. 6(a) and Fig. 6(b), we present the duration required for the successful transmission of a file with size of 100 MBs in the wired and combined network accordingly. We easily notice that even a low increase in RTT values of 20 ms increases file transmission duration up to 5.5 s (15%). Moreover, we notice that the effect regarding the increased deviation values is also clearly illustrated between Fig. 6(a) and Fig. 6(b). We also conducted experiments based on UDP transmissions. However, UDP performance in terms of throughput, packet loss and jitter is not affected by the artificially injected RTT delay. This comes



(a) Wired



(b) Combined

Fig. 5. TCP throughput vs. artificial delay.

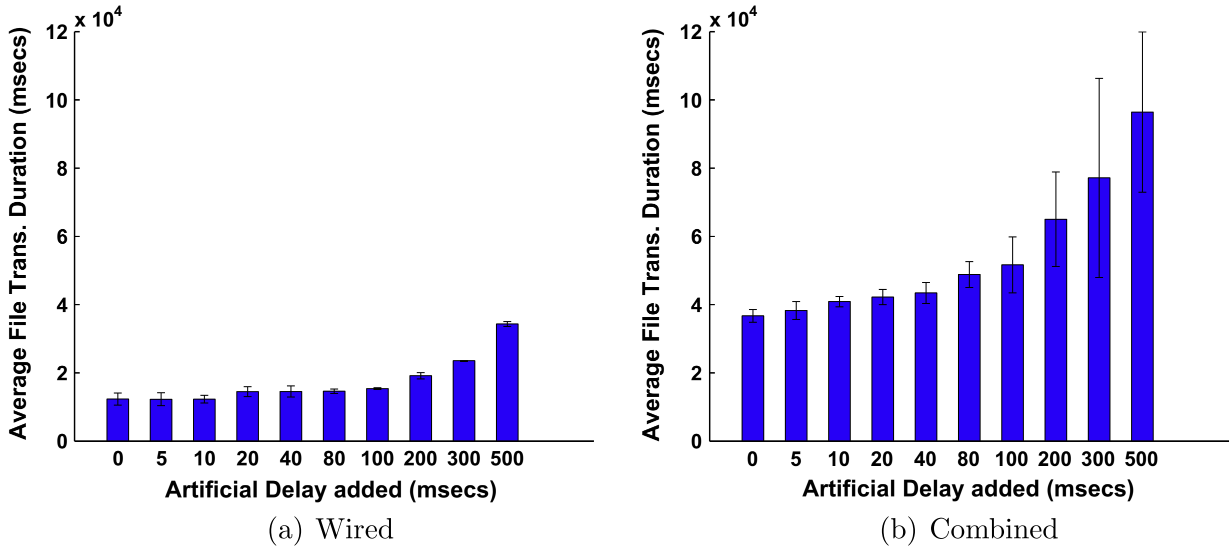


Fig. 6. TCP file transmission duration vs. artificial delay.

from the fact that even high values of artificial delay cannot result in packet loss, as the high capacity of operational system buffers supports storage of packets that arrive during the artificial delay interval even at the maximum traffic rate of 90 Mbps that is used in our experiments.

6.1.2. Wireless – Combined set of experiments

The second set of experiments has been designed to demonstrate the impact on end-to-end performance of channel contention in the wireless access link. Fig. 7(a) and Fig. 7(b) illustrate TCP and UDP throughput achieved in the wireless access link and the combined network accordingly, under various values of traffic rate for the contending flow.

For the UDP case, we notice that even contending flows of low traffic rate highly impact performance in both cases. In addition, we observe that results obtained in the wireless and combined networks are very similar and both feature relatively low deviation values. Packet loss measurements illustrated in Fig. 8(a) and Fig. 8(b) show that UDP performance is directly related to loss of packets. As the MC injects packets with high traffic rate, the wireless network capacity is exceeded due to the simultaneous transmissions of the contending flow. The resulting channel contention yields packet loss, which cannot be detected by the UDP protocol and thus the rate of data entering the network is not restricted within the network capacity region.

However, in the TCP case, we observe lower throughput performance yielded in the combined network (Fig. 7(b)). This is due to the fact that the TCP protocol involves RTTs estimation in its adaptive retransmission procedure, as part of the congestion control mechanism that aims at estimating the available network capacity to prevent link congestion. As a result, the TCP protocol reacts upon the detection of increased RTT values that result from the

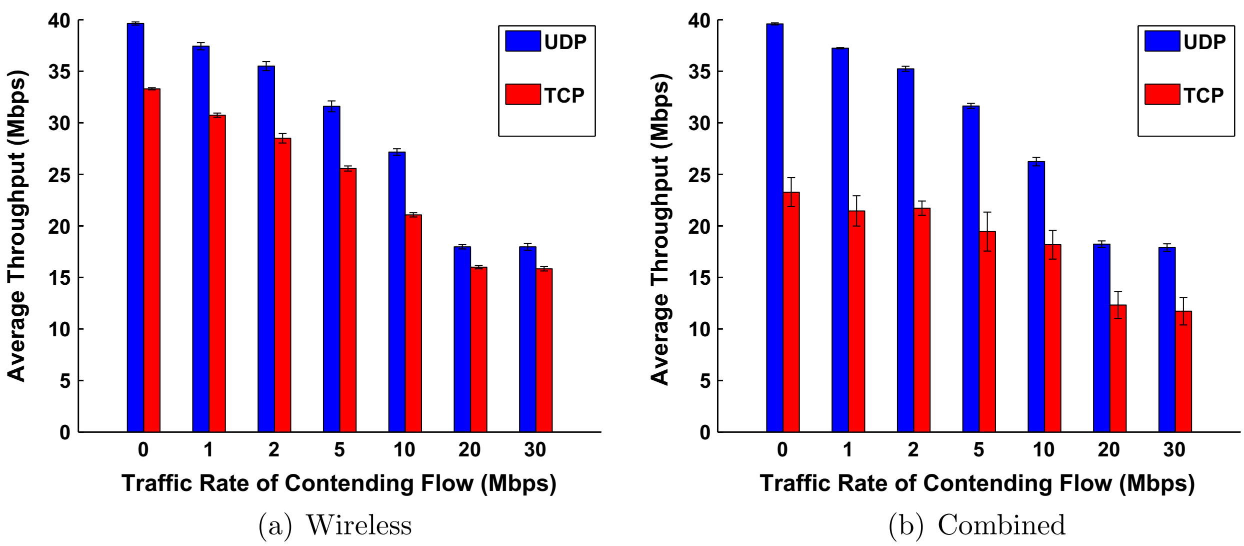
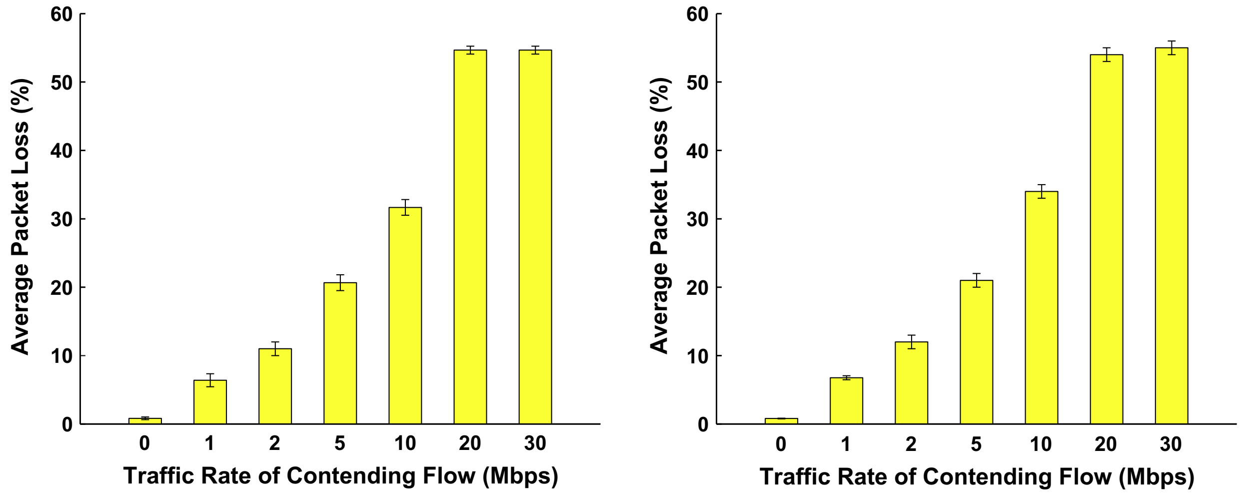


Fig. 7. TCP – UDP throughput vs. contention.



(a) Wireless (b) Combined

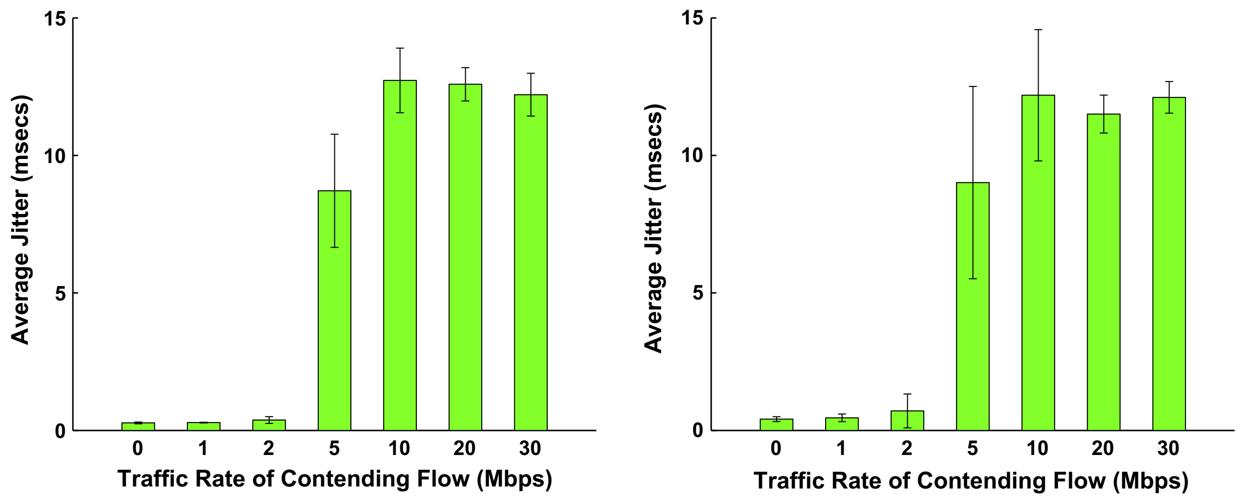
Fig. 8. UDP packet loss vs. contention.

augmented network range and consequently limits the rate of traffic injected by the MC, in order to control congestion in the network. Another particular observation regarding the TCP case is related to the high deviation values observed among the multiple executions of the experiment in the combined architecture, compared with the low deviation values observed in the local wireless network. This is due to the fact that the generated traffic flows go over the Internet through PLE during experimentation in the combined architecture and thus high deviation values are recorded as a regular characteristic of experimentation on realistic planetary scale networks.

Concerning UDP packet loss values presented in Fig. 8(a) and Fig. 8(b), we observe similar performance between experimentation on the wireless link and the combined network. Similar results are also obtained in terms of UDP Jitter between experimentation on the two different network architectures, as illustrated in Fig. 9(a) and Fig. 9(b). Moreover, we used the OMF visualization tool, in Fig. 10, to plot RTT values reported from the two MAPs with red and blue colors and also yellow color for values reported from the MC. Particularly, we observe that the MC is always associated with the MAP that features the lowest RTT delay and moreover we notice a handoff that lasts between the 120 and 140 s of the experiment.

6.2. Experimentation based on multiple PLE nodes

In this second part of experiments we focus on testing the performance of the proposed scheme under more dynamic topologies, where different PLE nodes are used as the final path destinations. Through the use of multiple PLE nodes that exist in different locations, we are able to evaluate performance under realistically varying RTT values. The experiments described in this part are organized in two discrete sets. In the first set, we investigate the impact of a specific contending flow on the throughput performance of different target flows and under varying



(a) Wireless (b) Combined

Fig. 9. UDP jitter vs. contention.

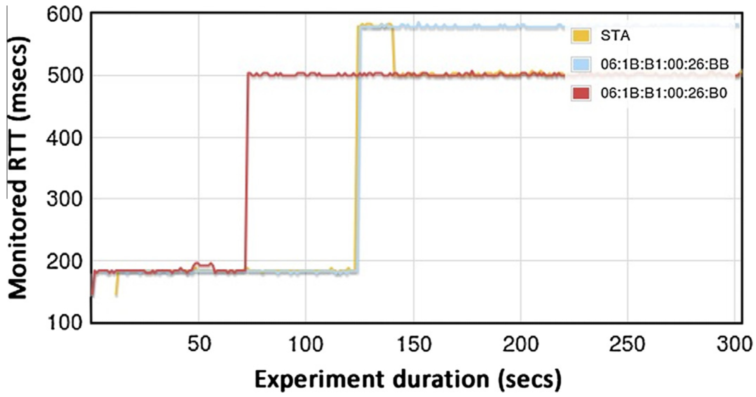


Fig. 10. Handoff demonstration.

levels of traffic contention, while in the second experimental set, we evaluate the performance of a specific flow during the streaming of a video file that is destined towards different PLE destination nodes.

6.2.1. Throughput experiments

The target flows that are considered in this set of experiments, are originated from different PLE nodes and are destined towards a single PLE node. In order to investigate the impact of traffic congestion on the throughput performance of the target flows, we use a contending flow under various traffic conditions. The contending flow is originated from a node in NITOS and shares the same destination with the target flows. The sources of the different target flows are three PLE nodes that are located in Russia, Japan and Chile accordingly. This experimental scenario has been designed in order to investigate the relation between the measured RTT for the different paths used by the target flows versus the obtained throughput performance in each case.

Fig. 11(a) and Fig. 11(b) depict the UDP throughput performance and the monitored RTT values, during experimentation with the different target flows, in the case that the PLE destination node is located in France. Similar results are illustrated in Fig. 12(a) and Fig. 12(b) for the case that the destination PLE node is located in Australia. In these experiments, we start by measuring performance of the target flows, while the contending flow is inactive and continue by altering the level of traffic that is injected by the contending flow among the values of 20, 50, 100 Mbps. The blue bars indicate performance in the case that the contending flow is inactive, while the red, green and yellow bars represent the results obtained in the cases that the contending flow is transmitting at the rate of 20, 50 and 100 Mbps accordingly.

As clearly demonstrated through Fig. 11(a) and Fig. 12(a), the UDP throughput performance of all target flows is significantly affected by the two different contending flows, in the case that the traffic rate of the contending flow reaches the value of 100 Mbps. However, in Fig. 11(a), we notice that the target flow that is originated from Russia starts to witness throughput degradation even in the case that the contending flow is transmitting at the rate of 50 Mbps, which is not observed for the other target flows. Moreover, we notice in Fig. 12(a) that all the different target flows start to witness throughput reduction, as soon as the contending flow reaches the rate of 50 Mbps, except for the experiment with the PLE node in Russia,

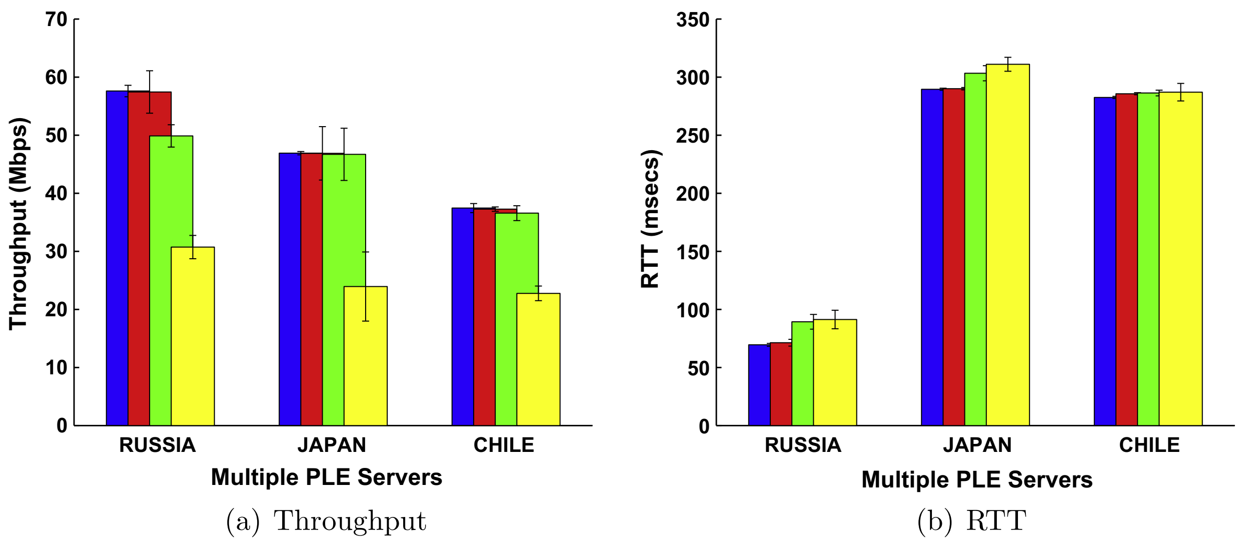
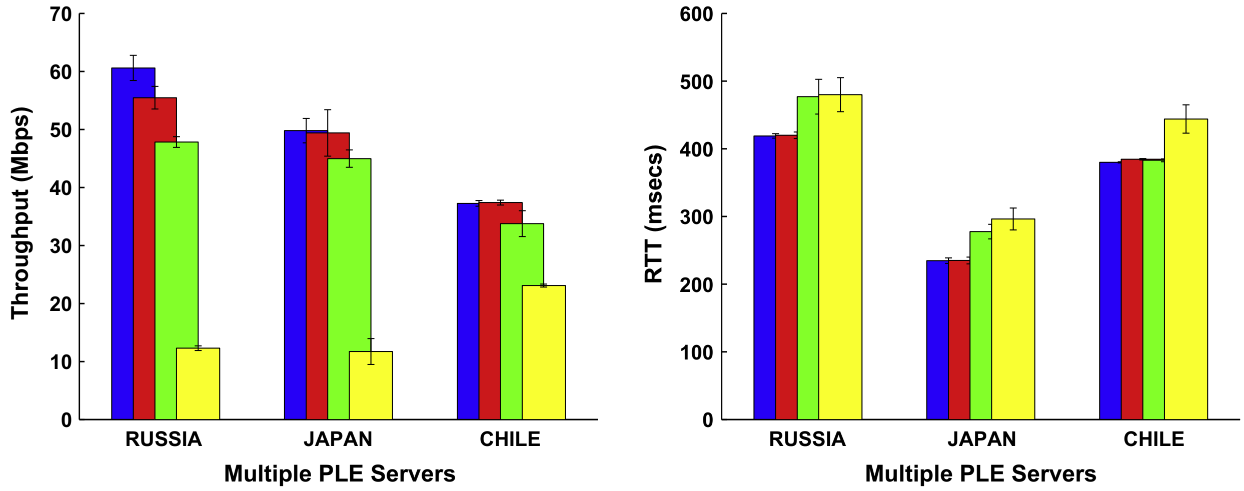


Fig. 11. Measurements for the target flow that is originated from a node in NITOS and destined to a PLE node in France.



(a) Throughput (b) RTT

Fig. 12. Measurements for the target flow that is originated from a node in NITOS and destined to a PLE node in Australia.

which yields degraded performance, even at the rate of 20 Mbps. The observed results can be interpreted, based on the fact that each target flow shares a different part of common network paths with the contending flow and as a result traffic congestion affects their performance in a different degree. On the other hand, the prevailing traffic conditions on each different network path also impact the resulting performance. However, we cannot monitor these conditions that are generated from external to our experiment sources, as the PLE network is a shared environment that uses the common Internet architecture.

Through the RTT values representation, which are illustrated in Fig. 11(b) and Fig. 12(b), we observe that the RTT for all the different network paths increases and also presents higher deviation values, in the cases that the contending flow severely impacts the UDP performance of the target flow. Based on this observation, we remark that variation of RTT values may act as an identification factor of network congestion. We also conducted experiments based on TCP transmissions. However, we omit presenting these results, as the obtained TCP throughput performance, across different network paths, presents the same univocal relation with RTT variations that was observed through the TCP experiments under artificially varying RTT values.

6.2.2. Video experiments

The goal of this final set of experiments is to show how the federated environment between NITOS and PLE testbeds can be utilized in testing the performance of video streaming applications that utilize discrete network paths. We consider a typical scenario for streaming video, which is based on the topology that is represented in Fig. 3, where a video file is streamed from the MC, forwarded to the MAP it is associated with and finally reaches different PLE destination nodes. The video streamed through the PLE network goes over the Internet giving experimenters the characteristics of a realistic network. Based on the experimental setup that is generated through the use of the *Socat* platform, the MC receives the corresponding traffic streams that have resulted through transmissions across different network paths and is finally able to store different versions of the initially transmitted video sequence. Consequently, we are able to evaluate the perceived video quality in each case, based on comparison of the delivered video files.

The number of received frames across multiple PLE destination nodes that feature different RTT values, is illustrated in Fig. 13. Results show comparable performance for the PLE nodes that are allocated in Greece, Australia, Portugal and Canada. The video file that is streamed between nodes in the internal network of NITOS is delivered

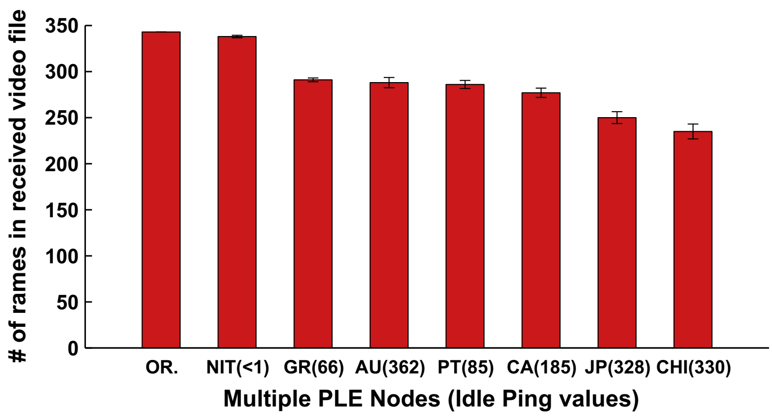


Fig. 13. Number of received frames during the transmission of the same video file through different PLE source nodes.

(a) NITOS (b) Portugal (c) Chile

Fig. 14. Representative screenshots of the same frame as transmitted through different PLE nodes.

nearly unvaried. Moreover, we notice that the lowest performance is yielded in the network paths that include PLE nodes located in Japan and Chile. Considering the obtained results, there exists no direct relation between the RTT values and the number of delivered video frames. In addition, we present some indicative screenshots in Fig. 14, which present the same video frame as delivered through different network paths. As our aim is not on comparing the video performance in terms of advanced video quality metrics, we simply rely on the screenshots representation to depict the video quality degradation that is yielded during the streaming of a video sequence through different PLE nodes over the Internet.

7. Simulation based evaluation

In order to explore the key differences between realistic testbed and simulated environments in performing Mesh Network experimentation, we decided to reproduce the WMN topology under consideration in a network simulator tool. For this purpose, we decided to use the well established NS3 [37] discrete-event network simulator, which provides for experimentation in both wireless and wired topologies. By properly configuring NS3 to simulate WMN topologies, we can evaluate the comparative complexity of configuring simulation and testbed environments. Moreover, we are able to conduct extensive sets of experiments and, through direct comparison between the results obtained in the realistic and simulated environments, we determine whether simulation models are able to reliably estimate the performance of WMNs. To this aim, we reproduced the full set of the experiments presented in Section 6.1, in the NS3 environment. In the rest of this Section, we refrain from showing the full set of obtained results and only present a subset of selected experiments that presented dissimilar performance, with regard to the results obtained in the realistic federated testbed environment. Based on the simulated experiments, we also comment on the obtained results and the perceived user experience.

7.1. Wireless topology – Channel contention

The first testbed experiment that we tried to reproduce is related to the evaluation of the impact that wireless channel contention induces on end-to-end performance of the wireless access link. In order to replicate the actual topology that was used in the real experiments, we simulated a grid topology consisting of 4 wireless nodes that are spaced 2 meters apart. In this scenario, we evaluate the throughput performance of the first communicating pair of nodes, considering varying traffic rate for the second contending flow. The nodes were configured to support the IEEE802.11a standard, through the applied *"WiFi_Phy_Standard_80211a"* tag. We also configured the nodes to use the fixed PHY-bitrate of 54 Mbps and 6 Mbps, through the application of the *"OfdmRate54Mbps"* and *"OfdmRate6Mbps"* tags, for transmissions of data and control frames accordingly. In order to model the wireless propagation environment, we used the basic Friis [38] model, which assumes that the received power drops as

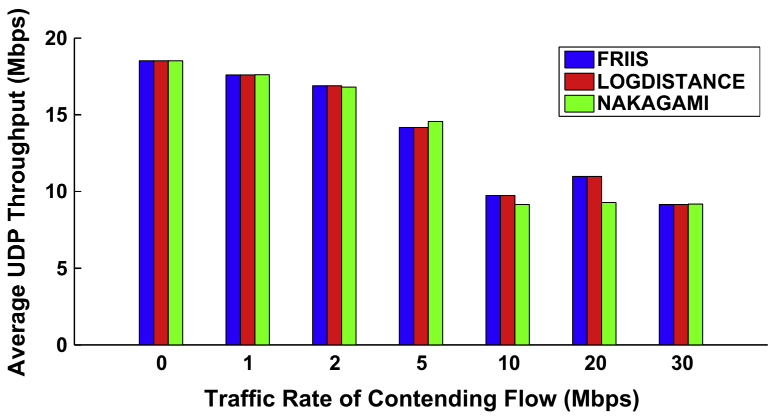


Fig. 15. UDP throughput vs. contention, in a wireless grid topology of 2 m node distance, considering different NS3 propagation models.

the square of the distance between the transmitter and receiver nodes, as it occurs in free space.

In Fig. 15, we use the blue color to depict the resulting UDP throughput performance. We easily notice that the maximum achievable throughput of 18.52 Mbps is significantly lower, than the maximum value of 39.6 Mbps that was obtained in the real experiment. Moreover, we observe a remarkable performance anomaly in the case that the contending flow uses the traffic rate of 20 Mbps, where throughput increases despite the fact that a higher level of channel contention is configured. In an effort to overcome the observed performance dissimilarities, we decided to experiment with different wireless propagation models, while keeping the node, topology and network configurations stable and exactly corresponding the realistic scenario.

To this aim, we first configured the LogDistance [39] propagation model, which assumes exponential path loss over the distance between transmitter and receiver nodes. More specifically, we set the path loss exponent value equal to 3, which is suitable for modeling urban environments, in an effort to approximate the real environment in which NITOS is located. However, based on the obtained results, which are presented through the red colored bars in Fig. 15, we notice that performance of the considered topology was not affected by the application of a different propagation model. Our next choice was to increase the node distance, in order to evaluate whether the increased distance would impact the resulting performance under the different propagation models. Nevertheless, although the distance between the two node pairs was increased up to 20 m, significantly differing from the original topology, monitored performance in the case of the Friis and LogDistance models was totally unaffected and thus we refrain from presenting the resulting plot.

Our next choice was to apply a signal fading model, and more specifically the Nakagami model [40], which encompasses different fading formulas for short-distance and long-distance communications. The resulting performance is illustrated in the green colored bars of Fig. 15. Application of the default NS3 Nakagami propagation model implementation resolved the observed performance anomaly in the case that the contending flow was using the traffic rate of 20 Mbps, when considering a wireless grid topology with 2 m of node distance. We also increased the node distance up to 20 m and noticed that the Nakagami model presented a slight performance decrease in all the considered cases, in contrast to the Friis and LogDistance models, in which cases performance was totally unaffected by the distance increase.

The overall conclusion drawn through this set of experiments is that several rounds of simulator reconfigurations were required in order to approach the performance obtained in the realistic wireless environment. However, although we configured all the proper settings to exactly resemble the realistic scenario, we did not manage to achieve the maximum throughput performance of 39.6 Mbps that was monitored in the real experiment. Moreover, the obtained results confirm observations of related work [41], where the authors argue that direct application of NS3 propagation loss models with the default parameters, do not result in realistic network behavior. Based on our own observations, we also remark that adjustment of several NS3 parameters is required to accurately simulate performance of wireless experiments, even when simple scenarios are considered. Another particular observation is related to the fact that there were only minor performance variations observed between measurements collected through multiple executions of the same experiment. As a result, we conclude that NS3 was not able to accurately reproduce the highly dynamic and stochastic nature of the wireless medium in the simulated environment.

7.2. Wired topology – Injected delay

The second testbed experiment that we tried to reproduce aims at assessing the impact of artificially injected delay on end-to-end performance of the wired backhaul part. To this aim, we created a simple topology consisting of two wired nodes that are connected through a point-to-point (P2P) link that offers configuration of the link capacity and delay, through the *"DataRate"* and *"Delay"* attributes. In order to replicate the wired experiment presented in Section 6.1.1, we configured the maximum P2P link capacity equal to 100 Mbps and also configured the link delay to vary between zero and 500 ms. Under this scenario, we monitor the resulting performance in terms of TCP/UDP throughput performance.

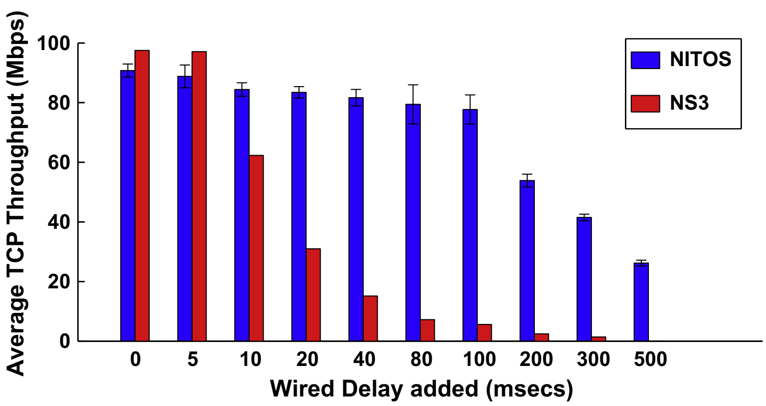


Fig. 16. TCP throughput vs. wired delay, in wired topology.

In Fig. 16, we illustrate the TCP throughput achieved under various artificial delay values, in both the realistic testbed and NS3 simulated environments. Based on the obtained results, we notice that the NS3 simulated TCP protocol considers RTT estimations as part of the TCP congestion control mechanism, and thus adapts the injected traffic rate, upon the detection of increased RTT values. However, the NS3 simulated TCP protocol reacts far more aggressively to the wired delay increase than the TCP implementation that is loaded on the Linux-based operating systems of the PLE and NITOS testbed nodes. More specifically, significant deterioration of the TCP throughput performance is observed, as soon as the delay reaches the value of 10 ms. Moreover, TCP throughput tends to decrease proportionally to the wired delay until it reaches extremely low values, when delay increases above the 200 ms. However, in the respective testbed experiments, throughput performance is only slightly decreasing for wired delay value up to 100 ms, while significant drop, that is proportional to the injected delay, is only monitored for wired delay values above 200 ms. Similar experiments were also conducted based on UDP transmissions, in which case throughput performance was not affected by the configured P2P delay, thus presenting network behavior fully identical to the testbed experiments.

Concluding, we remark that configuration of simulated wired experiments required less user intervention and moreover that the performance results approximated to some extent the experimentally obtained ones. In spite of the performance dissimilarities that were observed due to the aggressive behavior of the TCP protocol implementation in NS3, results on maximum TCP and UDP performance were identical to the experimentally obtained ones. We also observed only minor performance deviation for measurements collected through multiple executions, which shows that the P2P-based delay injection mechanism introduces far less environment variability in comparison to the one implemented through the *Dummynet* tool.

7.3. Combined topology – Injected delay

In the next step, we tried to reproduce the experiment that considers the combined network topology illustrated in Fig. 2. In order to to simulate the wired backhaul between the final server destination and the MAP that provides access to the associated MC, we used a P2P link of configurable delay, as in the aforementioned exclusively wired scenario. Following an approach identical to the one used in the simulated wired experiments, we aimed at detecting whether the combined topology further impacts end-to-end performance, as it was observed in the respective testbed experiments. For this purpose, we reproduced the combined topology, following settings identical to the ones configured in the two previous simulated experiments, considering each discrete network path accordingly. Under this scenario, we monitor the resulting performance in terms of TCP/UDP throughput performance.

Fig. 17 illustrates both TCP and UDP throughput achieved in the simulated combined network topology, under various wired link delay values. Regarding the TCP throughput performance, we observe that it deviates from the realistic behavior presented in Fig. 6(b). First, we notice that throughput decrease follows a trend similar to the one observed in the simulated wired experiment. Second, we note that the maximum achievable throughput performance of 12.62 Mbps is significantly lower than the one obtained in the realistic experiments (24.1 Mbps). Considering the UDP throughput experiments, we notice that performance is totally unaffected by the injected delay, for delay values lower than 80 ms. However, in cases when the injected delay increases above 100 ms, throughputs tend to decrease, reaching a maximum reduction of approximately 2 Mbps, in the case that 500 ms of delay are configured for the P2P link. UDP performance results obtained in the simulated combined scenario come in contrast with the ones obtained in the testbed environment, where UDP throughput was not affected by the wired delay, even for the highest configured delay values of 500 ms.

At this point, we would like to present some generic conclusions regarding the experience gained through the execution of identical experiments in both testbed and simulation based environments. Considering the complexity of experimental configuration, we remark that although testbed experimentation requires significant effort in the early configuration stages, the ease of use provided by experiment instrumentation frameworks, such as OMF and NEPI, efficiently simplifies the overall process. On the other hand, execution of introductory scenarios in the

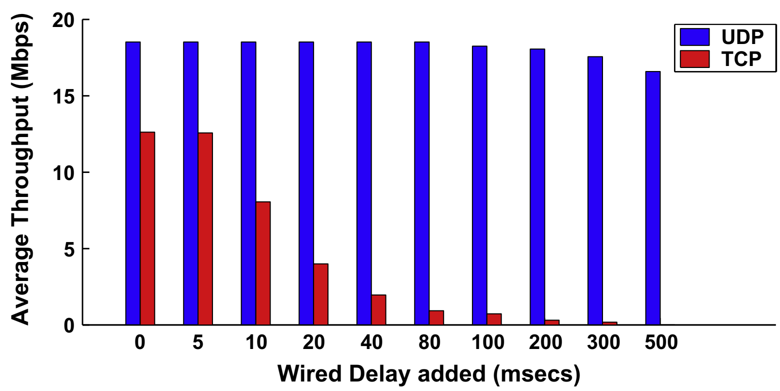


Fig. 17. Throughput vs. wired delay, in combined topology.

NS3 simulator tool was straightforward. However, several rounds of simulator reconfigurations were required in order to approach the performance obtained in the testbed environment, especially when configuring wireless experiments. Regarding the ability of simulator tools to accurately simulate performance of testbed experiments, we reached the conclusion that NS3 is able to provide some basic feedback about the expected performance of real testbed experiments. However, based on the various flaws that were observed, we conclude that NS3 is not able to accurately reproduce testbed experiments, especially topologies that include wireless or combined network parts. We have to mention that our experience with network simulators is limited to NS3 and that other mode advanced simulator environments, such as OPNET [42], might potentially approximate real experiments more accurately. However, sophisticated network simulator environments require experimenters to obtain high cost licences, in order to have access to the full list of available components. As a result, we decided to use the publicly available NS3 platform that is primarily targeted for research use and directly compare its performance with experimental results obtained through the federated environment consisting of testbeds that provide free remote access to experimenters worldwide.

8. Conclusions and future work

The unique two-tier architecture introduced by realistic WMNs has directed research efforts towards experimentation on global scale realistic environments that result from federation of heterogeneous networks. In this work, we present the federation of the wired PLE and the wireless NITOS testbeds. The resulting architecture has enabled the execution of realistic association experiments in the context of WMNs, which presented several characteristics of experimentation under real world scale and settings. Several of the testbed experiments were also reproduced in a network simulator tool, demonstrating the inability of simulated environments to accurately approximate the complex nature of the wireless medium and the high variability that experimentation over Internet generates. Through the extensive experiments and the corresponding collected results, we validated the importance of integrating experimental facilities for the design and development of the Future Internet. As part of our future work, we plan on investigating performance of more complex WMN topologies that also feature a wireless multi-hop backhaul in the aforementioned federated environment. Furthermore, we aim at enhancing the existing tools for experimentation in federated testbeds and developing new ones, towards upgrading the perceived user experience.

Appendix A. OEDL use of resource names

```
#---------RESOURCE NAMES--------#
defProperty ('MAP1_node',
  "omf.nitos.node023", "1st MAP")
defProperty ('MAP2_node',
  "omf.nitos.node022", "2nd MAP")
defProperty ('PLE_node',
  "planetlab.test.upmc.ple2", "PLE")
defProperty ('MC_node', "omf.nitos.node019",
  "MC")
#---------RESOURCE NAMES--------#
```

Appendix B. OEDL configuration of the first MAP

```
#-----------MAP1-----------#
# Definition of the RTT measurement
  application.
defGroup ('MAP1', property.AP1_node) do
  |node|
   node.addApplication ("app_measure")
  {|app|
       app.setProperty
  ('address',"132.227.62.120") #PLE node IP
       app.setProperty ('sampling',1)
  #Sampling interval
       app.setProperty ('debug',true)
  #Enable debugging flags
       app.setProperty ('port_number',4042)
  #port number of the
       #server-side application running at
  the PLE node
       app.setProperty ('proc-file',"/proc/
  net/madwifi/ath0/onelab_proc_file")
       #proc file used to parse RTT results to
  the custom
       #Mad-WiFi driver running at the MAPs
       app.measure
  ('roundtrip')#Measurement table suffix
   }
#Definition of the Iwconfig wrapper
  application.
   node.addApplication ("app_iwconfig") do
  |app|
       app.setProperty ('interface',
  'ath0') #Interface parameter
   app.setProperty ('sampling', 3)
  #Sampling interval
   app.measure ('iwconf') #Measurement
  table suffix
   end
   node.net.w0.mode = property.mode2
   #Setup of operational mode (MC/MAP)
   node.net.w0.type = property.wifi
   #Setup of wireless mode (802.11 a/b/g)
   node.net.w0.channel = property.channel1
   #Setup of wireless channel
   node.net.w0.essid = "MAP1"
```

(continued on next page)

```
    #Setup of ESSID name
    node.net.w0.ip = "192.168.0.1"
    #Setup of IP address to be used
end
#-----------MAP1-----------#
```

Appendix C. Dummynet application configuration and execution in OEDL

```
#--------DEFINITION OF DELAYS--------#
defProperty ('first_delay', 100, "First
  delay value")
defProperty ('second_delay', 200, "Second
  delay value")
defProperty ('third_delay', 350, "Third
  delay value")
#--------DEFINITION OF DELAYS--------#
```

```
#-------DELAY INJECTION TO PLE
  NODE-------#
group (PLE_Destination).exec (/bin/sh/home/
  certhple_nitosl/
net_add.sh #{property.first_delay})
#-------DELAY INJECTION TO PLE
  NODE-------#
```

References

[1] S. Keranidis, D. Giatsios, T. Korakis, I. Koutsopoulos, L. Tassiulas, T. Rakotoarivelo, T. Parmentelat, Experimentation in heterogeneous European testbeds through the Onelab facility: the case of PlanetLab federation with the wireless NITOS testbed, in: Proceedings of TridentCom 2012, 8th International ICST Conference on Testbeds and Research Infrastructures for the Development of Networks & Communities, 2012.

[2] K. Tan, D. Wu, A. Chan, P. Mohapatra, Comparing simulation tools and experimental testbeds for wireless mesh networks, in: Proceedings of IEEE International Symposium on World of Wireless Mobile and Multimedia Networks (WoWMoM), 2010.

[3] P. Pathak, R. Dutta, A survey of network design problems and joint design approaches in wireless mesh networks, IEEE Commun. Surveys Tutorials 13 (3) (2011) 396–428.

[4] T. Rakotoarivelo, M. Ott, G. Jourjon, I. Seskar, OMF: a control and management framework for networking testbeds, SIGOPS Oper. Syst. Rev. 43 (4) (2010) 54–59.

[5] S. Keranidis, T. Korakis, I. Koutsopoulos, L. Tassiulas, Contention and traffic load-aware association in IEEE 802.11 WLANs: algorithms and implementation, in: Proceedings of WiOpt, 2011.

[6] Y. Bejerano, S. Han, E.L. Li, Fairness and load balancing in wireless LANs using association control, in: Proceedings of MOBICOM, 2004.

[7] G. Athanasiou, T. Korakis, O. Ercetin, L. Tassiulas, Dynamic cross-layer association in 802.11-based mesh networks, in: Proceedings of INFOCOM, 2007.

[8] S. Makhlouf, Y. Chen, S. Emeott, M. Baker, A network-assisted association scheme for 802.11-based mesh networks, in: Proceedings of WCNC, 2008.

[9] L. Luo, D. Raychaudhuri, H. Liu, M. Wu, D. Li, End-to-end performance aware association in wireless municipal mesh networks, in: Proceedings of Globecom, 2009.

[10] H. Wang, W. Wong, W. Soh, M. Motani, Dynamic association in IEEE 802.11 based wireless mesh networks, in: Proceedings of ISWCS, 2009.

[11] Y. He, D. Perkins, S. Velaga, Design and implementation of CLASS: a cross-layer association scheme for wireless mesh networks, in: Proceedings of INFOCOM, 2010.

[12] C. Elliott, GENI: opening up new classes of experiments in global networking, IEEE Internet Comput. 14 (1) (2010) 39–42.

[13] A. Gavras, A. Karila, S. Fdida, M. May, M. Potts, Future Internet research and experimentation: the FIRE initiative, SIGCOMM Comput. Commun. Rev. 37 (3) (2007) 89–92.

[14] K. Webb, M. Hibler, R. Ricci, A. Clements, J. Lepreau, Implementing the Emulab-PlanetLab portal: experience and lessons learned, in: Workshop on Real, Large Distributed Systems (WORLDS), 2004.

[15] B. Chun, D. Culler, T. Roscoe, A. Bavier, L. Peterson, M. Wawrzoniak, M. Bowman, PlanetLab: an overlay testbed for broad-coverage services, ACM SIGCOMM Comput. Commun. Rev. 33 (3) (2003) 3–12.

[16] R. Mahindra, G. Bhanage, G. Hadjichristofi, S. Ganu, P. Kamat, I. Seskar, D. Raychaudhuri, Integration of heterogeneous networking testbeds, in: Proceedings of TridentCom, 2008.

[17] G.D. Stasi, S. Avallone, R. Canonico, Integration of OMF-based testbeds in a global scale networking facility, in: Proceedings of ICST QShine, 2009.

[18] 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.

[19] G. Di Stasi, R. Bifulco, F. D'Elia, S. Avallone, R. Canonico, A. Apostolaras, N. Giallelis, T. Korakis, L. Tassiulas, Experimenting with P2P traffic optimization for wireless mesh networks in a federated OMF-PlanetLab environment, in: Proceedings of WCNC, 2011.

[20] S. Fdida, T. Friedman, T. Parmentelat, OneLab: An Open Federated Facility for Experimentally Driven Future Internet Research, Springer, 2010.

[21] P. Antoniadis, S. Fdida, T. Friedman, V. Misra, Federation of virtualized infrastructures: sharing the value of diversity, in: Proceedings of the 6th International Conference, Co-NEXT '10, 2010.

[22] Dummynet, <http://info.iet.unipi.it/luigi/dummynet/>.

[23] M. Lacage, M. Ferrari, M. Hansen, T. Turletti, W. Dabbous, NEPI: using independent simulators, emulators, and testbeds for easy experimentation, SIGOPS Oper. Syst. Rev. 43 (4) (2010) 60–65.

[24] PlanetLab Central (PLC) portable installation, <https://svn.planet-lab.org/wiki/MyPLCUserGuide>.

[25] J. White, G. Jourjon, T. Rakotoarivelo, M. Ott, Measurement architectures for network experiments with disconnected mobile nodes, in: Proceedings of TridentCom 2010, 6th International ICST Conference on Testbeds and Research Infrastructures for the Development of Networks & Communities, 2010.

[26] OMF Website, <http://mytestbed.net/>.

[27] XMPP Protocol RFC, <http://xmpp.org/rfcs/rfc6120.html>.

[28] SFA Registry, <http://svn.planet-lab.org/wiki/SFATutorialRegistryInterface>.

[29] MySlice, <http://myslice.info/>.

[30] IEEE 802.11-2007 Wireless LAN Medium Access Control and Physical Layers Specifications, 2007.

[31] IPtables, <http://netfilter.org>.

[32] Nmap, <http://nmap.org/>.

[33] Iperf, <http://dast.nlanr.net/Projects/Iperf/>.

[34] VLC Media Player, <http://www.videolan.org/vlc/>.

[35] Socat Relay Platform, <http://freecode.com/projects/socat>.

[36] NITOS-PLE Federation Experiment, <http://nitlab.inf.uth.gr/NITlab/images/stories/tutorials/federation.zip>.

[37] NS3 – Network Simulator for Internet Systems, <http://www.nsnam.org/>.

[38] H.T. Friis, A note on a simple transmission formula, Proc. IRE (5) (2006) 254–256.

[39] V. Erceg, L.J. Greenstein, S.Y. Tjandra, S.R. Parkoff, A. Gupta, B. Kulic, A.A. Julius, R. Bianchi, An empirically based path loss model for wireless channels in suburban environments, IEEE JSAC 17 (7) (2006) 1205–1211.

[40] M. Nakagami, The m-distribution – a general formula of intensity distribution of rapid fading, in: Statistical Methods in Radio Wave Propagation, 1960.

[41] M. Stoffers, G. Riley, Comparing the NS-3 propagation models, in: Proceedings of IEEE 20th International Symposium on Modeling, Analysis Simulation of Computer and Telecommunication Systems (MASCOTS), 2012.

[42] OPNET Network Simulator, <http://www.opnet.com/>.

Stratos Keranidis received the Diploma of Computer Engineering and Communications in 2008 and the M.S. degree in Communication Engineering in 2011, both from the University of Thessaly (UTH), Greece. He is currently a Ph.D. candidate under the supervision of Prof. Iordanis Koutsopoulos and also a research associate in the Centre for Research and Technology Hellas (CERTH), Greece. He is also a member of the Network Implementation Testbed Laboratory (NITlab), working on the wireless NITOS testbed. His research interests lie in the areas of design and implementation of resource allocation algorithms in wireless networks and their study in experimental network facilities (testbeds).

Dimitris Giatsios received the Diploma of Electrical and Computer Engineering from Aristotle University of Thessaloniki in 2007, and a M.Sc. degree in Telecommunication and Network Engineering from University of Thessaly in 2011. Since 2012, he is a Ph.D. candidate at University of Thessaly under the supervision of Prof. Leandros Tassiulas. He is also a research associate in the Centre for Research and Technology Hellas. His research interests include design of novel architectures for wireless networks, dynamic spectrum access networks, and experimentation in network testbeds.

Thanasis Korakis is a lecturer at the Department of Computer and Communications Engineering at University of Thessaly, Greece. From 2005 to 2010 he was a research assistant professor in the ECE Department, Polytechnic Institute of NYU. His research interests lie in the area of wireless networks, with expertise in protocol prototyping and wireless testbeds. He has published more than 60 papers and holds 5 patents. He has been the PI for several NSF projects in USA and has participated in several testbed related EU projects. Since 2007 he has been a voting member of the IEEE 802.16 standardization group.

Iordanis Koutsopoulos (S'99, M'03, SM' 13) is Assistant Professor at the Department of Informatics of Athens University of Economics and Business (AUEB) since 2013. He was Assistant Professor (2010–2013) and Lecturer (2005–2010) with the Department of Computer Engineering and Communications, University of Thessaly. He is also affiliated with Center for Research and Technology Hellas, Institute for Telematics and Informatics (CERTH-ITI). His research interests are in the general area of network control and optimization.

Leandros Tassiulas is a Professor at the Department of Computer and Communications Engineering at University of Thessaly, Greece, and Research Professor in the ECE Department at University of Maryland College Park since 2001. His research interests are in the areas of communication networks design and analysis, with emphasis on fundamental mathematical models of communication networks, architectures and protocols of wireless systems, sensor networks, high-speed Internet and satellite communications. He has published in excess of 180 papers and holds one patent. He has received several national and international acknowledgements and awards for his research (NSF Research Initiation Award 1992, NSF Career Award 1995, INFOCOM 1994 best paper award). He is the director of NITlab, a research lab in UTH focusing on implementation of wireless protocols and testbed experimentation.

Thierry Rakotoarivelo is a senior researcher with the Network Research Group at NICTA, where he works on Frameworks for Large Scale Worldwide Experimental Studies. He also works on integrating the use of these frameworks together with experimental platforms in the teaching of networking courses. His recent research interests are in the study of online communities, such as one-click file sharing communities, and the application of large-scale sensor networks to structural monitoring. Before joining NICTA, He received his Ph.D. from the University of New South Wales and the Institut National Polytechnique de Toulouse. During his Ph.D., he worked on the distributed discovery and management of alternate Internet paths with enhanced Quality-of-Service. Prior to that, He was a senior research engineer at the Motorola Australian Research Centre (MARC, now closed), where he worked with the Wireless Technology Lab (WTL) on Quality-of-Service oriented MAC protocols for Wireless LAN (WLAN).

Max Ott is a Sr. Principal Researcher and a Research Leader in the Networks Research Group at NICTA. He is heavily involved in many of the worldwide experimental research facility activities, such as GENI (US) and FIRE with many testbeds adopting the award-winning OMF control and management framework. He is also holding an Adjunct appointment at WINLAB, Rutgers University where he is contributing to the software architecture of the ORBIT testbed. Before coming to NICTA he founded Semandex, which is a pioneer provider of Content-Based Networks, a new generation of Enterprise Information Integration and Knowledge Management systems.

Thierry Parmentelat is a senior research engineer at INRIA Sophia-Antipolis, and is currently Technical Director of the OneLab experimental facility. His areas of interest include experimental testbeds, networking, and programming languages. Prior to joining INRIA, he has held various positions within the software industry; he has been CTO with Centile, a VOIP company; and has designed and implemented critical systems for various industrial sectors, such as nuclear energy and aerospace. He graduated from the École Polytechnique in 1984, and received his Ph.D. in 1991.