Experimentation in Heterogeneous European Testbeds through the Onelab Facility: The Case of PlanetLab Federation with the Wireless NITOS Testbed

Stratos Keranidis[1,2], Dimitris Giatsios[1,2], Thanasis Korakis[1,2],
Iordanis Koutsopoulos[1,2], Leandros Tassiulas[1,2],
Thierry Rakotoarivelo[3], and Thierry Parmentelat[4]

[1] Department of Computer Engineering and Telecommunications,
University of Thessaly, Greece
[2] Centre for Research and Technology, Hellas
[3] National ICT Australia (NICTA), Alexandria, Australia
[4] INRIA, Sophia Antipolis, France
{efkerani,gidimitr,korakis,jordan,leandros}@uth.gr,
thierry.rakotoarivelo@nicta.com.au, thierry.parmentelat@sophia.inria.fr

Abstract. The constantly increasing diversity of the infrastructure that is 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 experimental testbeds 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 a realistic environment. 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.

Keywords: Testbed Federation, Wireless Mesh Networks, Experiments.

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

T. Korakis, M. Zink, and M. Ott (Eds.): TridentCom 2012, LNICST 44, pp. 338–354, 2012.

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, in accordance with the unique characteristics introduced by the complex nature of WMNs [1] 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 induces high deployment and maintenance cost. 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 conclude on realistic results under real congestion conditions. Such requirements have led the research community to create 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 the use of different software frameworks for resources management and controlling. In this work we realize the federation between two well-established heterogeneous network testbeds, namely the NITOS wireless testbed and the planetary scale wired PlanetLab Europe (PLE) testbed. The utilization of a common experiment control framework, OMF [2] (cOntrol and Management Framework), and the adoption of the slice abstraction as the building block for the federation have made the testbeds' integration 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 part (PLE) and part in the wireless part (NITOS). The implemented mechanism is based on novel association metrics [3] that consider wireless channel contention, which are further enhanced to take into account wired delay as well. 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.

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. In section 5 we present and comment on the results obtained from the experimental evaluation of the implemented mechanism. Finally, in section 6 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). The WMN consists also 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 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 [4] 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 [5]. The authors in [6] 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 [7], 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 [8], 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 are not not able to capture the interaction among different layers [1], research related to WMNs is mainly performed in experimental facilities. A recent work in the field [9] 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. [10], as well as the FIRE initiative in Europe [11] are currently investigating federation of heterogeneous testbeds.

2.2 Integration of Heterogeneous Experimental Facilities

An initial effort on federation of testbeds was proposed in [12], where the wireless EmuLab testbed and the wired planetary-scale PlanetLab testbed [13] 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 [14], aimed at integrating PlanetLab with the ORBIT wireless testbed. The authors considered also the ability of performing experiments on the integrated framework concurrently. Although 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 maintained on multiple testbeds supporting many different types of technologies. The work proposed in [15] 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 [16] that enables the assignment of different subsets of nodes and channels to different users during specific reservation slots. The work in [17] 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 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 [18] 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 is a dedicated server that runs components of PlanetLab services.

Slices/Slivers. The notion of a slice is a rather central notion in PlanetLab; it typically allows to model 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.

MyPLC. MyPLC 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 outdoor nodes 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 registration and its approval by the testbed administrators. Below we describe the two basic software entities of the NITOS testbed, OMF/OML and NITOS Scheduler.

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), 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.

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. In the Scheduler's backend, there are two main functionalities worth mentioning: the Scheduler's interaction with the XMPP framework used in OMF and the spectrum slicing framework. Both of them are utilized to create 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. 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 dissociation of resources to the corresponding slices, based on queries to the NITOS scheduler's database. Since all OMF communication takes place via XMPP, this mechanism is equivalent to OMF-based dynamic access control to the NITOS resources.

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.

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 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 NITOS portal with the provided credentials. The process is transparent to the user and incurs no significant delay.

Deployment of OMF/OML at PLE Resources. A major difficulty when trying to run combined experiments using heterogeneous facilities is that different

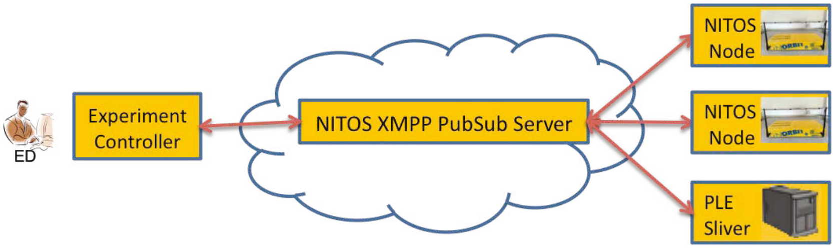


Fig. 1. XMPP Server Connectivity

languages are used to describe resource configurations and actions. There is the need for agreement to use a common language for experiment description, 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, an OMF Resource Controller is installed and 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.

XMPP Communication Using Slices. In order for all the resources to be able to communicate with the EC, they must be registered in the same XMPP server or to a set of XMPP servers peered with each other. Currently the XMPP servers of PLE and NITOS are not peered. As a result, we adopted the first choice of using a single XMPP server and more specifically the NITOS XMPP server, where the PLE node could be registered as if features a public IP address, which is not the case for NITOS nodes that use private IP addresses. We are currently working towards enabling the peering between these two XMPP servers. The architecture, used in our current work, is presented in Fig. 1. As for a next step, we logged into NITOS using the PLE slice credentials and statically associated the PLE resource to the automatically generated slice, by adding a corresponding entry to the NITOS XMPP server. In accordance with the functionality of NITOS Scheduler, the PLE resource could be accessed via NITOS only by the user related with the automatically generated slice and thus no unauthorized access issues were raised.

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.

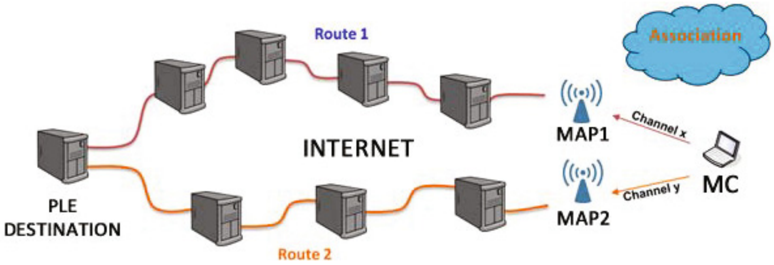


Fig. 2. Topology representation

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 [3], we concluded to 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. In order to estimate the RTT required for the initial transmission and subsequent retransmissions of a frame with specific length, we have to multiply 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:

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

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 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:

$$RTT_{ij}^{total} = RTT_{ij}^{A} + RTT_j^{B} \tag{3}$$

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 [3]. 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 /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. 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 Evaluation

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

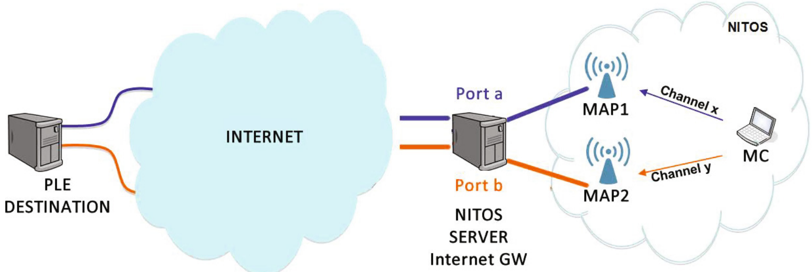


Fig. 3. Experimental topology representation

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 APs 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 [19] 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.

As our association mechanism is end-to-end delay aware, we had to generate conditions of varying delay in both the wired and the wireless parts. In order to add artificial delay in the wired backhaul link, we used the *Dummynet* [20] tool, which is able to simulate 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. 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* [21] 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

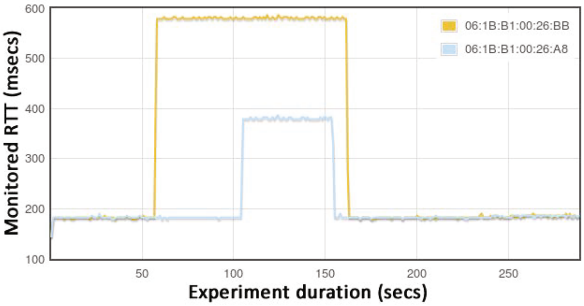


Fig. 4. Delay monitored at the PLE node for two flows generated by each MAP

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 by each one of the MAPs, as monitored at the PLE node.

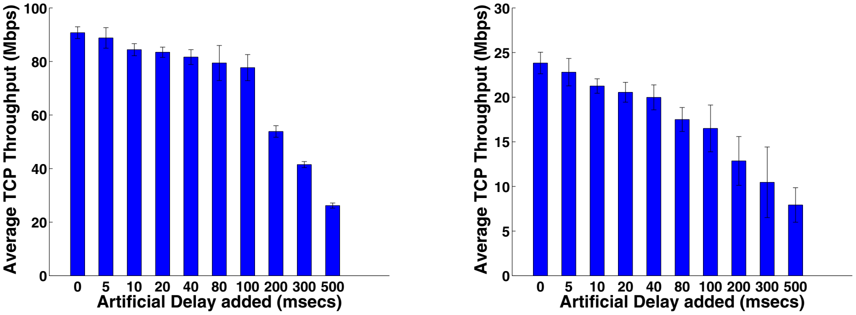
The throughput performance of the experiments is measured by using Iperf [22]. In our experiments, we run an Iperf Client at the MC to generate TCP flows and UDP flows of varying rate and also an Iperf Server residing at the PLE node to receive traffic and collect the corresponding measurements. We run each experiment 10 times and each run lasts for 2 minutes.

5.2 Experiments

The conducted experiments 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 residing 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.

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

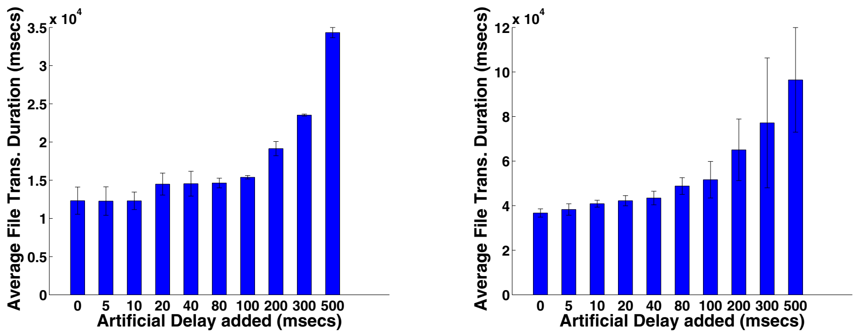


(a) Wired (b) Combined

Fig. 5. TCP Throughput vs Artificial Delay

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 msecs 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



(a) Wired (b) Combined

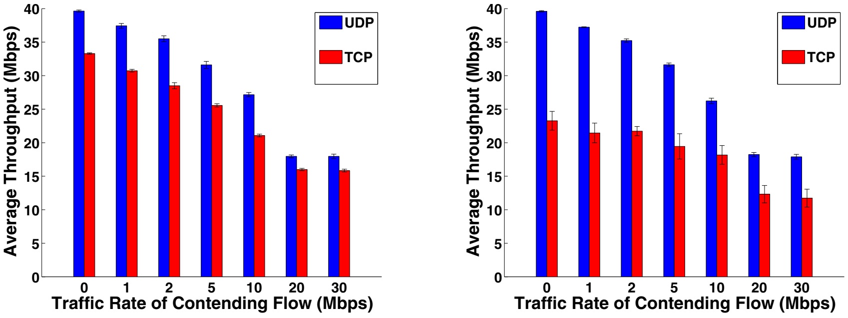
Fig. 6. TCP File Transmission Duration vs Artificial Delay

increases file transmission duration up to 5,5s (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 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.

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.

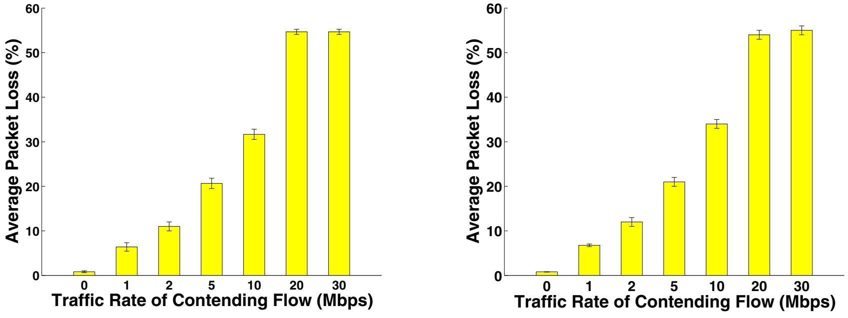
For the UDP case, we notice that even contending flows of low traffic rate highly impact performance in both cases. In addition, we observe that results obtained in the wireless and combined networks are very similar and both feature relatively low deviation values. Packet loss measurements illustrated 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 and thus upon the detection of increased RTT values that result from the augmented network range, limits the rate of traffic injected by the MC. Another



(a) Wireless (b) Combined

Fig. 7. TCP - UDP Throughput vs Contention

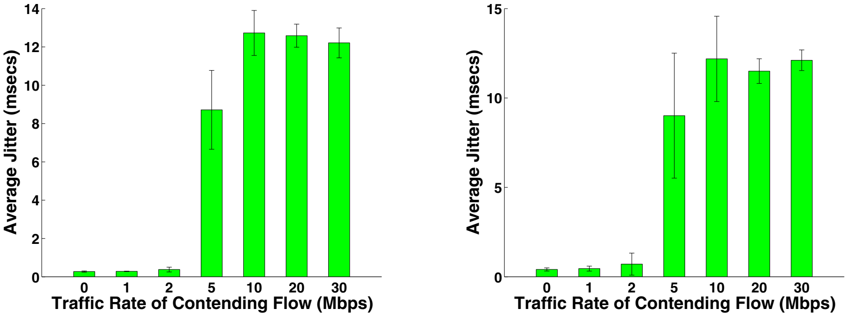


(a) Wireless (b) Combined

Fig. 8. UDP Packet Loss vs Contention

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



(a) Wireless (b) Combined

Fig. 9. UDP Jitter vs Contention

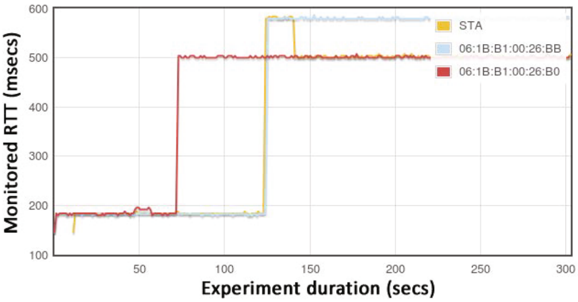


Fig. 10. Handoff demonstration

RTT delay and moreover notice a handoff that lasts between the 120 and 140 seconds of the experiment.

6 Conclusions and Future Work

The unique two-tier architecture introduced by 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. As part of our future work, we plan on investigating performance of typical WMNs that also feature a wireless multi-hop backhaul in the aforementioned federated environment.

References

1. Pathak, P.H., Dutta, R.: A Survey of Network Design Problems and Joint Design Approaches in Wireless Mesh Networks. IEEE Communications Surveys Tutorials 13(3), 396–428 (2011)
2. Rakotoarivelo, T., Ott, M., Jourjon, G., Seskar, I.: OMF: A Control and Management Framework for Networking Testbeds. SIGOPS Oper. Syst. Rev. 43(4), 54–59 (2010)
3. Keranidis, S., Korakis, T., Koutsopoulos, I., Tassiulas, L.: Contention and Traffic load-aware Association in IEEE 802.11 WLANs: Algorithms and Implementation. In: Proceedings of WiOpt (2011)
4. Bejerano, Y., Han, S., Li, E.L.: Fairness and load balancing in wireless LANs using association control. In: Proceedings of MOBICOM (2004)
5. Athanasiou, G., Korakis, T., Ercetin, O., Tassiulas, L.: Dynamic cross-layer association in 802.11-based mesh networks. In: Proceedings of INFOCOM (2007)
6. Makhlouf, S., Chen, Y., Emeott, S., Baker, M.: A Network-Assisted Association Scheme for 802.11-Based Mesh Networks. In: Proceedings of WCNC (2008)

7. Luo, L., Raychaudhuri, D., Liu, H., Wu, M., Li, D.: End-to-End Performance Aware Association in Wireless Municipal Mesh Networks. In: Proceedings of Globecom (2009)
8. Wang, H., Wong, W., Soh, W., Motani, M.: Dynamic association in IEEE 802.11 based wireless mesh networks, 2009. In: Proceedings of ISWCS (2009)
9. He, Y., Perkins, D., Velaga, S.: Design and Implementation of CLASS: A Cross-Layer ASSociation Scheme for Wireless Mesh Networks. In: Proceedings of INFOCOM (2010)
10. Elliott, C.: GENI: Opening Up New Classes of Experiments in Global Neworking. IEEE Internet Computing 14(1), 39–42 (2010)
11. Gavras, A., Karila, A., Fdida, S., May, M., Potts, M.: Future Internet research and experimentation: the FIRE initiative. SIGCOMM Computer Communication Review 37(3), 89–92 (2007)
12. Webb, K., Hibler, M., Ricci, R., Clements, A., Lepreau, J.: Implementing the Emulab-PlanetLab portal: Experience and lessons learned. In: Workshop on Real, Large Distributed Systems, WORLDS (2004)
13. Chun, B., Culler, D., Roscoe, T., Bavier, A., Peterson, L., Wawrzoniak, M., Bowman, M.: PlanetLab: An Overlay Testbed for Broad-Coverage Services. ACM SIGCOMM Computer Communication Review 33(3), 3–12 (2003)
14. Mahindra, R., Bhanage, G., Hadjichristofi, G., Ganu, S., Kamat, P., Seskar, I., Raychaudhuri, D.: Integration of heterogeneous networking testbeds. In: Proceedings of TridentCom (2008)
15. Di Stasi, G., Avallone, S., Canonico, R.: Integration of OMF-based testbeds in a global scale networking facility. In: Proceedings of ICST QShine (2009)
16. Anadiotis, A.-C., Apostolaras, A., Syrivelis, D., Korakis, T., Tassiulas, L., Rodriguez, L., Seskar, I., Ott, M.: Towards Maximizing Wireless Testbed Utilization Using Spectrum Slicing. In: Magedanz, T., Gavras, A., Thanh, N.H., Chase, J.S. (eds.) TridentCom 2010. LNICST, vol. 46, pp. 299–314. Springer, Heidelberg (2011)
17. Di Stasi, G., Bifulco, R., D'Elia, F.P., Avallone, S., Canonico, R., Apostolaras, A., Giallelis, N., Korakis, T., Tassiulas, L.: Experimenting with P2P traffic optimization for wireless mesh networks in a federated OMF-PlanetLab environment. In: Proceedings of WCNC (2011)
18. Fdida, S., Friedman, T., Parmentelat, T.: OneLab: An Open Federated Facility for Experimentally Driven Future Internet Research. Springer (2010)
19. IPtables, `http://netfilter.org`
20. Dummynet, `http://info.iet.unipi.it/~luigi/dummynet/`
21. Nmap, `http://nmap.org/`
22. Iperf, `http://dast.nlanr.net/Projects/Iperf/`