

Federating Wired and Wireless Test Facilities through Emulab and OMF: The iLab.t Use Case

Stefan Bouckaert, Pieter Becue, Brecht Vermeulen, Bart Jooris, Ingrid Moerman, and Piet Demeester

IBBT - Ghent University, Department of Information Technology (INTEC),
Gaston Crommenlaan 8, Bus 201, 9050 Ghent, Belgium
{firstname.lastname}@intec.ugent.be
<http://ilabt.ibbt.be>

Abstract. The IBBT iLab.t technology centre provides computing hardware, software tools and measurement equipment to support researchers and developers in building their ICT solutions, and in measuring the performance of these solutions. Among other things, the iLab.t hosts several generic Emulab-based wired test environments called the Virtual Walls, and two wireless test environments which are grouped under the name w-iLab.t. Until very recently, these wired and wireless test facilities each had their own history: they were deployed and maintained by a different group of people, were operated using different tools, and each had their own community of experimenters. This paper provides insight on the origin and evolution of the Virtual Wall and w-iLab.t facilities. It explains how these facilities were federated, by using the best parts of both the OMF and Emulab frameworks. It discusses the benefits of our local federation as well as our future federation plans.

Keywords: testbed, wireless, wired, emulab, omf, federation.

1 A Short History Behind the iLab.t

The history behind the iLab.t test facilities goes back to 1997, and starts with a first experimental set-up consisting out of 1 ATM switch and 2 Windows PCs deployed in the premises of the IBCN research group of the INTEC department of the Ghent University, Belgium. One year later, the Atlantis testlab was born, and over the years additional nodes, a diverse range of network equipment and measurement equipment was added. After the IBBT (Interdisciplinary Institute for Broadband Technology) was founded in 2004, the activities of the Atlantis lab —at that point already counting well over 100 network nodes— were continued and significantly expanded under the flag of the IBBT iLab.t technology centre. By 2006, the iLab.t counted over 300 rack mount PCs, a wide selection of network devices and technologies, professional test and measurement equipment including wired and wireless sniffers, packet generators and QoS analysers. Currently, about 120 40U racks are used. In 2007, it was decided to make the testbeds more generic by installing the Emulab software [1] on 100 servers, that

we named ‘the Virtual Wall’. In this way, people could use the same resources repeatedly and create also larger testbeds for experiments.

At that time, the popularity of wireless WLAN devices and sensor nodes was also significantly increasing. Being a research institute involved in multidisciplinary demand-driven research projects that often demand a proof-of-concept demonstrator, several small-scale desktop wireless testbeds with diverse types of hardware and wireless interfaces started to appear throughout the office building. However, it quickly became clear that there are significant drawbacks to maintaining several small-scale (wireless) testbeds; to name just a few drawbacks, with multiple individual test set-ups, it was often time-consuming to re-run ‘old’ experiments after the equipment had been stored in a cupboard for an extended period of time, especially if a certain experiment was not well-documented. Furthermore, re-using (parts of) previously programmed code and scripts is difficult, for example if one set-up is based on a COTS wireless Wi-Fi router and another set-up makes use of desktop PCs with Wi-Fi cards. As a final example, for scalability and efficiency reasons, in many cases it makes much more sense to build expertise on devices of a single type and buy a large amount of these nodes, than to have fragmented and more limited knowledge on operating a more diverse selection of hardware and have several testbeds of a smaller scale. These and other experiences led to the design and deployment of a 200-node wireless testbed, called the w-iLab.t in 2007 [2]. Since 2007, the w-iLab.t has been further developed and at this moment, there are two instances of the lab. Being deployed in the IBBT offices in Gent, Belgium, the original deployment is called the w-iLab.t office. The new location is known as w-iLab.t Zwijnaarde, after the town in which it is located, approximately 5 km away from the central IBBT offices.

In 2010 and 2012 respectively, a second and third Virtual Wall were installed. As a result, there are now 300 servers available in 3 Emulab configurations.

2 The Virtual Wall and w-iLab.t Facilities

To get a better understanding of the Virtual Wall and w-iLab.t facilities, a high-level description of both facilities is provided below. Next, Section 2.3 explains how both facilities are currently being used.

2.1 High-Level Description of the Virtual Wall

The iLab.t currently has 3 Virtual Wall testbeds. A Virtual Wall exists out of 100 servers, all connected with 4 or 6 Gigabit interfaces to a central switch (Force 10 E1200) which functions as a patch panel. Besides this, there is a control interface per node through which people can login. The testbed runs the Emulab software of the University of Utah [1]. Experimenters can build (large) experiments by drawing a topology in a graphical user interface or by creating NS2 files describing the topology of the experiment. One of the advantages of the Virtual Wall is that the Emulab software controls the switch as a virtual patch panel, so



Fig. 1. Emulab based Virtual Wall testbed

experimenters can remotely change topologies and configurations. There is no virtualisation involved, so the experiments run on the bare machine hardware, which is very important for performance experiments. As can be seen in Figure 1, 20 of the 100 nodes are connected to a display so that also visual feedback can be gathered, for example from large scale gaming experiments or video streaming experiments. As indicated above, in 2010 and 2012 two additional Virtual Walls were installed of 100 nodes. The advantage of having the testbed locally, is that we can also connect measurement hardware (e.g. Spirent Testcenter) or specific test devices besides getting visual feedback.

2.2 High-Level Description of the w-iLab.t

As indicated above, the original w-iLab.t deployment is located in an office building, and spans 3 floors of 90m x 18m. At 200 spots throughout offices, meeting rooms and hallways, wireless nodes are mounted to the ceiling. More precisely, at each of the spots, an embedded PC (PC Engines Alix3c3 [3]) is installed. All embedded PCs are connected over Ethernet to a central control server. Each embedded PC is equipped with two Wi-Fi a/b/g mini-PC interfaces. Moreover, a TelosB sensor node is connected via a custom-built so called ‘environment emulator’ to a USB port of the embedded PCs. This environment emulator allows experimenters to take more control over their sensor node experiments (e.g. emulation of sensor node inputs, reading/setting analog and digital I/O pins) and enables advanced logging functionality [4]. As a result, each node in the w-iLab.t office can be activated as a sensor node, a Wi-Fi node, or a combination of sensor and Wi-Fi, for example to act as a gateway. How the sensor nodes and/or Wi-Fi nodes behave is completely decided by (and reprogrammable by) the

experimenter. For example, a Wi-Fi node uses the Madwifi [5] driver by default. As such, each Wi-Fi node can be programmed to behave as an access point, or as a station, or can be configured in monitor mode. It is also possible to install a different wireless driver, and/or to extend or modify any of the nodes in the same way as an experimenter would be able to do with a node that resides on his desktop, thus guaranteeing full flexibility in the experiments. The control software of the w-iLab.t office is based on the Motelab software [6] but has been extended over the years [2] to improve the functionality and usability.

The main driver for deploying a second w-iLab.t testbed, was the fact that the 2.4GHz ISM spectrum band in the office environment is at times heavily interfered by operational wireless networks, cordless phones and microwave ovens. Furthermore, since people are working in the office, the wireless link quality during daytime experiments may be very unpredictable. While for some experiments, this interference and unpredictability is a welcome challenge, it is obviously also very helpful to be able to execute experiments in an environment where no people are working and where no devices external to the experiment are causing interference. Such environment was found in a utility room above a cleanroom in Zwijnaarde. In this 66m x 20.5m room, 60 fixed nodes (Wi-Fi a/b/g/n, custom sensor node, Bluetooth) are installed. In addition, 20 mobile nodes based on a vacuum cleaning robot available with similar characteristics as the fixed nodes are soon to be added. The new nodes are more powerful than those in the office deployment [7]. While the low-power nodes in the office deployment are powered over Ethernet, the more powerful nodes in the Zwijnaarde testbed are powered via Power Distribution Units (PDUs). A high-level overview of w-iLab.t Zwijnaarde architecture is provided in Figure 2. While not listed on the figure, the w-iLab.t Zwijnaarde is also home to software defined radios and advanced spectrum sensing components.

When planning the new Zwijnaarde deployment during 2011, the requirements were determined based on the experience gained when installing and maintaining the w-iLab.t office environment. The wish list of new functionalities included expectations such as a more easy versioning system (to make it easier to go back to a previous version of the code under test and to specific parameter settings), easier sharing of experiments and code between users, a more uniform interface for including and manipulating sensor nodes and Wi-Fi based nodes in experiments, a more flexible way to share and comment on results, support for mobile nodes, and support for multi-site experiments. As such, it was clear the control software that was used for the office environment would have to be significantly extended. While such redesign would have been possible, or new and clean code could have been written from scratch, after studying the state of the art at that time we decided to start from and contribute to the OMF [8,9] control and management framework, since part of the functionality that was on our wish list became available with no or limited adjustments, by adopting OMF. Furthermore, while OMF does not fully support all functionalities available in the w-iLab.t testbed (e.g. environment emulator or our custom visualisation and processing extensions), we believe this lost functionality can easily be transferred

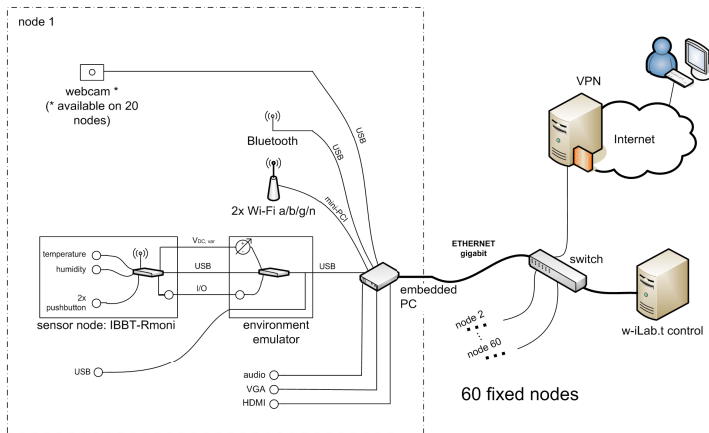


Fig. 2. High-level overview of the w-iLab.t Zwijnaarde architecture

to the OMF. To summarize: using the functionality provided by, and contributing to a well-known and widely used tool was found to be more useful than continuing to work on a custom code base. As will be described in Section 3, it is the OMF tool that is also used to enable the federation between the w-iLab.t Zwijnaarde and the Virtual Wall.

2.3 Role of the Facilities

Until recently, the (wired) Virtual Wall and (wireless) w-iLab.t facilities, although operated on the same physical location, were installed and largely maintained by different people. For each of the facilities, different user tools were/are provided, and in general, the knowledge on operating and using the platforms and the specific extensions was/is divided among two distinct user groups. In hindsight, the divide between the wired and wireless experimentation facilities grew in a natural way and is largely historically linked to the organization of our research in research clusters, such as ‘wireless and mobile networks’ and ‘cloud computing’. As such, when initially implementing both facilities, the choice was made to start the Virtual Wall based on the the Emulab software, and the (original) w-iLab.t (office) on Motelab, based on multiple site visits of similar testbeds in the US and Europe. In 2007-2008, those choices fitted best our needs. Obviously, as the experimental facilities were further developed, the fact that different people were involved in setting up and maintaining the testbeds also means that the Virtual Wall and the w-iLab.t have grown to what they are today while fully focussing on their respective experimenter communities, without having to make compromises towards the other community. Table 1 summarizes the most important differences.

Table 1. Comparison of facility characteristics

	Virtual Wall (wired)	w-iLab.t (wireless)
Choice of nodes	Specific nodes are chosen based on technical requirements (e.g. number of interfaces).	As the location of the nodes influences the topology, the experimenter hand-picks the nodes involved in his/her experiment.
Reservation	After reserving nodes, user controls experiment via SSH (or custom scripts). Scheduling an experiment does not necessarily mean executing an experiment.	Experiment is usually fully defined before experiment can be scheduled; scheduling experiment equals executing an experiment.
Node capacities	Powerful multi-core PCs	Embedded PCs for control and Wi-Fi, low-power low-capacity sensor nodes. Ethernet usually only management.
Interfaces	Multiple identical Ethernet interfaces per node	Heterogeneous wireless interfaces (Wi-Fi, Bluetooth, IEEE 802.15.4)
Storage	NFS mounts, or custom solution by the experimenter	NFS mount, custom database solution for sensor nodes
Experiment life-cycle support	Support for swapping in images	Office: support for flashing/installing nodes, collecting, processing, visualising results, trigger events during experiments; Zwijnaarde: OMF/OML based deployment

Although, as will be described in Section 3, a federation between these wired and wireless experimentation facilities is now a fact (at the moment of writing, the final steps of the integration happened little over 2 weeks ago), so far the differences in experimentation communities are still reflected in the user accounts for the Virtual Wall and w-iLab.t. When comparing the user account lists from both platforms, only 9 experimenters have an account for both testbeds, with the total number of accounts for the Virtual Wall and w-iLab.t being 120 and 60 respectively (excluding student group accounts and other generic accounts which are used by multiple users, for instance in the scope of a research project). An example experiment performed by a ‘typical’ Virtual Wall user on the w-iLab.t involved reasoning on top of sensor values that were dynamically obtained in real-time from the wireless sensor nodes of the office testbed. In this case, a central machine with a lot of processing power, which is not available in the ‘normal’ w-iLab.t set-up was required. The ad-hoc solution in this case was simply to add an external computing node to the control network of the w-iLab.t for the duration of the experiment. In addition to the accounts for individual experimenters, both iLab.t facilities are used in many national and international research projects,

and are used by PhD and master students for research and educational purposes. An selection of international research projects and the role of the iLab.t is listed in Table 2.

Table 2. Selected projects in which the iLab.t facilities are used

Use and development of experimentation infrastructure	
BonFIRE www.bonfire-project.eu	BonFIRE designs, builds and operates a multi-site Cloud prototype Future Internet facility. The Virtual Wall is one of the facilities in the BonFIRE cloud, providing support for experiments with a need for a fully controllable network environment.
CREW www.crew-project.eu	The w-iLab.t is one of the core testbeds in the CREW federation. Among other things, the w-iLab.t test environment is used to create reproducible interference environments modelled according to the typical use of wireless networks at different locations, such as a home or an office.
OFELIA www.fp7-ofelia.eu/	The OFELIA project creates an experimental facility that allows researchers to not only experiment ‘on’ a test network but also to control and extend the network itself in a precise and dynamic way, using OpenFlow networking technology. Both the Virtual Wall and the w-iLab.t are part of the OFELIA facilities.
OpenLAB www.ict-openlab.eu/	OpenLab brings together the essential ingredients for an open, general purpose and sustainable large scale shared experimental facility. The w-iLab.t testbed is made available to selected experimenters as part of the OpenLab federation.
Research projects	
SPITFIRE www.spitfire-project.eu/	The w-iLab.t is used to evaluate the implementation and usage of embedded web service technology based on the IETF CoAP protocol. Novel solutions for facilitating sensor deployment, discovery and access are designed and evaluated.
SPARC www.fp7-sparc.eu/	The SPARC project studies carrier grade extensions to split architectures including OAM, restoration and reliability, network virtualization, and resource isolation in order to open up carrier networks to the benefits of split architectures. The iLab.t Virtual Wall is used to prototype carrier class applications of OpenFlow and to evaluate their scalability and performance.
EULER www.euler-fire-project.eu/	The main objective of the EULER research project is to investigate new routing paradigms so as to design, develop, and validate experimentally a distributed and dynamic routing scheme suitable for the future Internet and its evolution. The Virtual Wall is used for the prototyping, functional validation and performance measurements of the routing protocols.
CONSERN www.ict-consern.eu	CONCERN aims at developing and validating a novel paradigm of dedicated, purpose-driven small scale wireless networks that are characterized by energy awareness and service-centric evolution. Within this context, w-iLab.t is used to implement and experimentally measure the power savings that can be achieved by different cooperative mechanisms in heterogeneous network environments.

From this table, it can be seen that while the Virtual Wall and the w-iLab.t facilities are used in many projects, currently only the OFELIA project makes use of both facilities. Considering the above observations, it is a very valid question whether it makes sense to federate wireless and wired experimentation facilities. In the next section, it will be explained why we decided to federate our wireless and wired facilities, regardless of these observations.

3 Federating the Virtual Wall and w-iLab.t Facilities

3.1 Complimentary Experimentation Tools

As can be seen in the overview Table 3, both Emulab and OMF have specific functionality in the experiment life cycle. Some of the functionality is overlapping, while other functionality is unique for one of the frameworks. Especially the unique features of one of the frameworks are very interesting to share among all experimenters/experiments. Besides these, the standard OMF framework is lacking some functionality regarding secure use of nodes, which can be perfectly filled in by Emulab functionality. Because of these reasons, it was obvious for us to try to combine the functionality of both frameworks in a federated environment. For free, we got a similar environment for wired and wireless experiments, which is very interesting for both testbed administrators and experimenters. Customised code is limited to the minimum and fed back to the official frameworks. This means also that we can open up the testbeds more easily to the outside world, as e.g. APIs created for projects as Bonfire or Ofelia can now talk to wired and wireless testbeds.

Table 3. Comparison of facility characteristics

	Virtual Wall (Emulab)	w-iLab.t Zwijnaarde (OMF)
Resource discovery	proprietary + SFA	Inventory database on AM
Resource reservation	Only current availability on a first come, first served base	-
Resource initialization	SFA/XMLRPC	Manual
Experiment control	ssh + startup scripts	ssh + OEDL scripts
Monitoring	-	OML
User storage	NFS (secure user and project home mounted)	NFS (user home mounted)
Authorization	webbased + XMLRPC + SFA Credential API	-
Experiment topology setup	Web interface or NS file	Ruby topology script
NIC config	done by Emulab	OEDL scripts

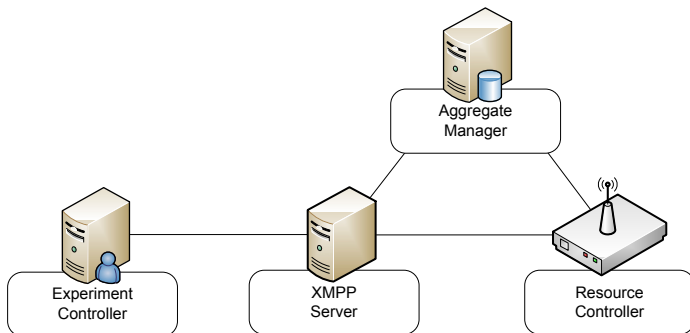


Fig. 3. OMF system architecture

3.2 iLab.t Federation: Building Blocks

The initial blocks for the federation are two separate facilities: the Virtual Wall, based on Emulab, and the w-iLab.t Zwijnaarde. The choice for federating with the w-iLab.t Zwijnaarde in this initial phase is made because this newer testbed is currently already equipped with OMF, while the w-iLab.t office is yet still managed by our older, Motelab-based management software. In a later phase, the w-iLab.t office is also to be integrated in the iLab.t federation. The remainder of this section is organized as follows: first, the Emulab and OMF frameworks and the relevant methodology are introduced in Sections 3.3 and 3.4. Next, the federation approach is explained in Section 3.5.

3.3 OMF Basics and Experimentation Lifecycle

The OMF framework consists of 4 major blocks, illustrated in Figure 3, that are interconnected over a wired control framework. First, there is the Aggregate Manager (AM), which can be considered as the main server of the testbed. The AM is responsible for initializing and configuring testbed resources. The AM also runs the OML (OMF Measurement Library) server [4] to collect results from the nodes during the experiment. Second, the Experiment Controller (EC) is the machine on which experimenters log in to run their experiments on the testbed. The EC can start and stop experiments and interact with the nodes by sending commands to the resource controllers during the experiment. Third, the Resource Controller (RC) is a software daemon which runs on each of the resources (which are in the case of w-iLab.t the wireless nodes) and waits for instructions from the experiment controller. Finally, there is an XMPP (Extensible Messaging and Presence Protocol) server, used to propagate the communication between the experiment controller and the resource controllers.

To understand the relevance of the proposed federation solution, it is important to understand the installation process and experiment lifecycle of an OMF testbed; The first step in **installing** an OMF testbed, is to install an aggregate manager (AM). This AM runs a DHCP, DNS and TFTP server. Whenever a

node is added to the testbed, its specifications should be added to the inventory database on the AM. Every node should also be registered to a certain part of the testbed (also called a ‘slice’) via the XMPP server. The AM also runs a Frisbee server [6] to load images on the nodes. The second step is to install the experiment controller, and complete the EC configuration with the IP addresses of the AM and the XMPP server. Finally, the resource controllers are installed and configured on the nodes. In the corresponding configuration file, the control interface (e.g. eth0), the XMPP server address and the experiment slice the node operates in by default (e.g. the `default_slice`, spanning the entire testbed) have to be specified.

After the installation, the **OMF experiment lifecycle** is the following.

- When a user wants to run an experiment on an OMF-enabled testbed, he first has to decide which image (OS, drivers, ...) he wants to configure the nodes with. By executing an *omf load* command on the EC, the image is multicast from the AM to the nodes that require that image.
- The second step for the user is to write an OEDL script (OMF Experiment Description Language). This script selects which nodes will be part of the experiment and what applications will be executed at what time during the experiment. We use the Human Readable Name (HRN) to define a node. This HRN is also listed in the inventory database and is registered to a slice in the XMPP server.
- Now the user can start the experiment by executing an *omf exec* command on the EC.
- The EC now checks the status (power on/off) of the nodes through the AM. The AM can then power on the nodes if necessary via a software component which is part of the so called Chassis Manager, responsible for the management of the power state of the nodes.
- Once the nodes are powered on, the EC sends commands to the resource controllers (RC) on the nodes through the XMPP server. Commands can be to install software, start scripts, configure wireless interface, ...
- During the experiment, results can be collected with OML [10]. The OML server runs on the AM. Resource nodes need to have the OML client library installed.

3.4 Emulab Basics and Experimentation Lifecycle

Figure 4 illustrates a basic view on an Emulab based testbed. There are 2 central servers (which run FreeBSD): BOSS is in charge of all configuration and organisation work, while OPS is the NFS server. Experimenters can login on OPS to get to the storage after their experiment was finished. Then there is a large switch to which all nodes are connected with as much as possible ethernet ports. This switch only functions as a patch panel, so it substitutes the manual patch cable connections. In this way, experimenters can use nodes as software based routers with multiple ports, use nodes as delay or packet loss nodes, and so on.

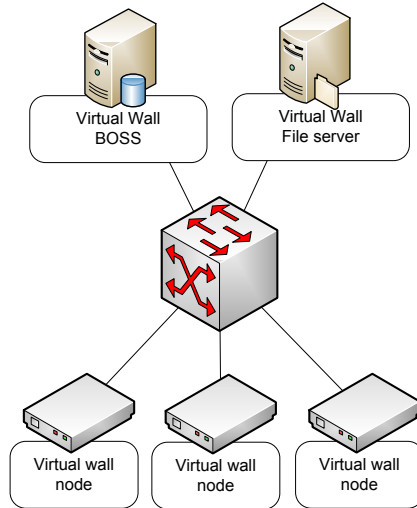


Fig. 4. Emulab architecture

Some nodes have 4 harddisks to make it possible to create RAID0 configurations for fast disk access. Emulab also has a user/group/project model which works by delegation. A user which is a 'grouproot' can admit other users to his experiments, so this makes the testbeds scalable to many users with a low sysadmin overhead. All experimenters in a group have access to the nodes of experiments of this group.

When starting an experiment, Emulab swaps in the right images on the right nodes, configures the IP addresses, the VLANs on the switch and the user passwords. There is also a very useful dynamic DNS system which makes that you can address your nodes in the experiment always through the same hostname, no matter which node is chosen for the experiment. At the end of swapping in the experiment, a script can be executed, or the experimenter can take manual control through SSH or RDP (remote desktop). Emulab has also a pubsub system with barriers, but experimenters tend to implement a similar thing in their experiment software, we have learned.

3.5 The iLab.t OMF-Emulab Federation

The basic approach of our federation solution is to extend the Virtual Wall (Emulab) testbed to support OMF experiments. Because no structural changes were made to the original setup, users can continue to use the testbed as before, as such guaranteeing backward compatibility and continuity of both experimentation environments. Figure 5 provides a high-level overview of the federation set-up.

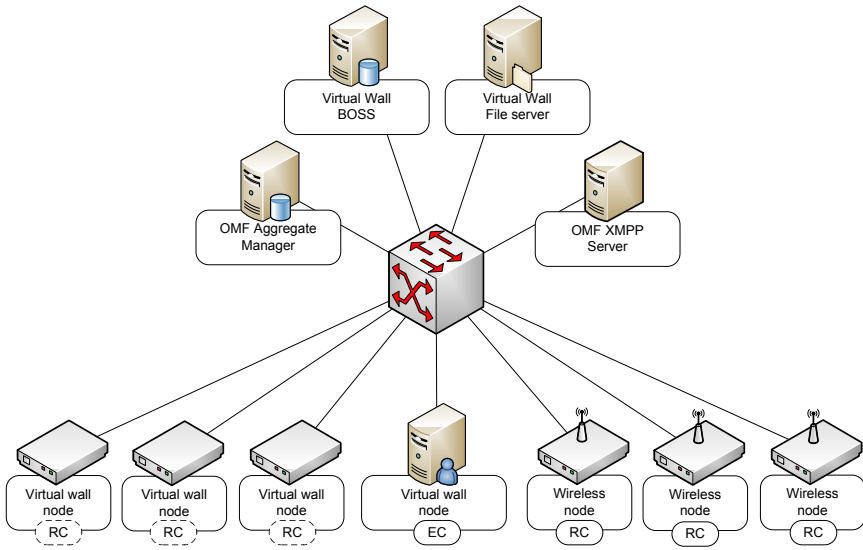


Fig. 5. The Virtual Wall – w-iLab.t federation

As shown in this Figure, two servers were added to the Virtual Wall set-up. One server acts as OMF Aggregate Manager, the other one as XMPP server. Although the OMF framework allows experimenters to install their own Experiment Controller on their local desktop, experimenters are assisted in setting up a joint experiment as follows. A specific script was created, that can be included when configuring a Virtual Wall experiment. By (automatically) executing this script, a regular Virtual Wall node is configured as an OMF Experiment Controller. As such, whenever users want to run an OMF experiment on the Wall, one of the Virtual Wall nodes can be swapped in, and this conversion script can be started automatically. The user home directory on the EC is persistent, as it is mounted on the home directory of the user on the Virtual Wall file server, which can be accessed at all times (also after the experiments are completed).

A similar approach is used for the nodes that are part of a joint experiment. Both Virtual Wall nodes and wireless nodes can be part of an OMF experiment. To accommodate this, a first requirement is that the nodes should run an OMF Resource Controller. We therefore provided a script that turns a regular (Virtual Wall or wireless) node into an OMF-enabled node. The RC configuration file only requires the address of the XMPP server, which is a static server added to the Emulab setup. As discussed in Section 3.3, the second requirement of the OMF framework is that all nodes should be listed in the inventory database and should be registered to a certain slice in the XMPP server. We therefore created a second script (started after the conversion script), which registers the node to the AM inventory and the XMPP server. For this we use the Emulab DNS system that generates a unique dynamic DNS name and IP address for every node that is swapped in for an experiment. Based on the MAC address, the register script

inserts the IP address and HRN (a unique DNS name assigned by Emulab) in the inventory and uses the HRN again to add the node to an experiment slice in the XMPP server. After the experiment is finished, the nodes should be removed from the inventory and the XMPP server to avoid problems when re-scheduling an experiment, since Virtual Wall resource names can be different for every run of an experiment. Note that for wireless nodes, the Emulab software is configured to select specific nodes. This is needed because the location of a node is very important in wireless experiments. In this setup, the Virtual Wall can be considered as the core testbed, having the possibility to swap in the OMF framework when requested by the user. This loose coupling approach has the advantage of requiring very little effort to achieve and ensures the correct working of the core testbed software. Furthermore, the Virtual Wall users can now also make use of the OML functionality to log their experiment results. In the chosen implementation, the Virtual Wall replaces some of the AM functions: the AM normally has to check whether nodes are on or off when an experiment is started and switch them on if necessary. In this implementation, this functionality is not needed anymore, since the Emulab software ensures that the nodes are switched on once they are swapped in. In order to overrule the AM functionality, we configured the AM to use a dummy Chassis Manager, which informs the AM core that the resources are always on. Furthermore, the AM functionality to load images on nodes is now also performed by the Virtual Wall, which also uses the Frisbee disk loader. This means that previously configured images of the w-iLab.t can be reused in the federated testbed.

Figure 6 demonstrates the setup of an experiment, using the new federation possibilities. In this experiment two arbitrary Virtual Wall (pcX and pcY) nodes are swapped in. One is configured as OMF EC, the other node is part of the experiment and can run additional software (e.g. video streaming server for home security system). Two fixed wireless nodes are chosen to act as wireless access points. Note the difference between the Virtual Wall nodes (arbitrary chosen by the system) and the wireless nodes (specifically requested by the user). All the nodes in this example are running Ubuntu 11.04, but other operating systems can be used as well. The only requirement for the nodes is that they should be able to run the experiment or resource controller, programmed using ruby scripts

4 Discussion and Future Plans

The iLab.t federation approach explained above demonstrates that federation – although exactly defining the term ‘federation’ in the context of Future Internet infrastructures is a discussion on its own – should not always be a time consuming and complex process. The implemented loose coupling between the OMF and the Emulab framework results in benefits for both the experimenters and ourselves as testbed owners/maintainers. For experimenters, the federation means easy and unified access to both a wired and a wireless experimentation facility. Executing experiments which concurrently use (parts of) the Virtual Wall and (parts of)

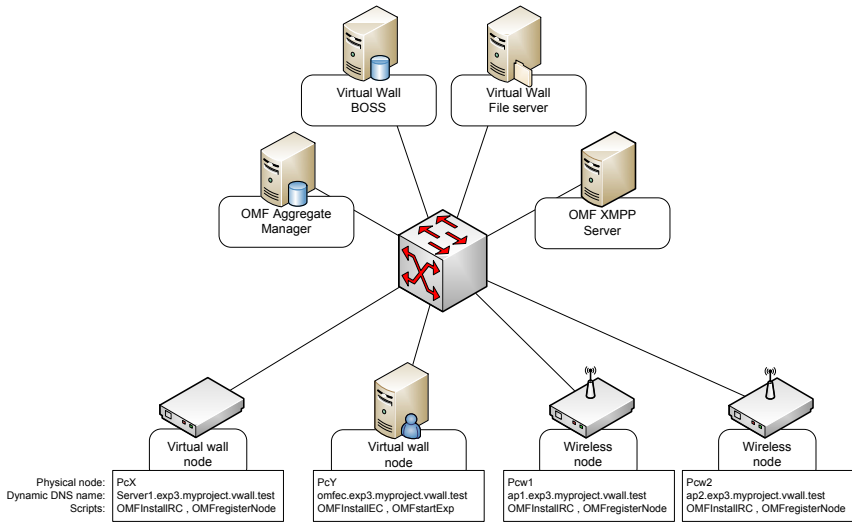


Fig. 6. Example federated experiment

the w-iLab.t is now possible. Even when combining nodes of both testbeds is not of importance to the experimenter, there is still the benefit of added functionality: Virtual Wall experimenters now have better control over their experiment lifecycle thanks to OMF, and the OML measurement library can be used to collect and process measurements. Vice versa, for experimenters using the w-iLab.t Zwijnaarde deployment, the federation means that nodes can now be configured via a user-friendly Emulab GUI. Furthermore, while in the ‘normal’ OMF deployment of the w-iLab.t Zwijnaarde, experimenters could (un)willingly destroy another experimenter’s experimenter by executing an OMF-load command on a node that was already in use, this is now impossible, as an Emulab experiment cannot be swapped out by another user. Also, the whole user/project delegation system of Emulab is now available to the wireless testbed, meaning that creation of accounts happens in a unified way, and, that secure use of storage is possible now.

Without any doubt, this functionality that was added by federating the Emulab and OMF approaches could also have been added by modifying or extending the code of either Emulab or OMF. The advantage of gaining the functionality through federation, is that there was very little implementation effort required to gain functionality while being assured of backwards compatibility. We also profit in this way of the best things of two standard frameworks with almost no customisation. It must be noted that enabling this local federation from a technical point of view is just one little aspect of the iLab.t federation story: having the technical possibility to let OMF and Emulab - wireless and wireless - experimentation facilities cooperate does not mean that experimenters will instantly be aware of the possible benefits and start scheduling federated experiments. However, for the administrators maintaining the Virtual Wall and w-iLab.t

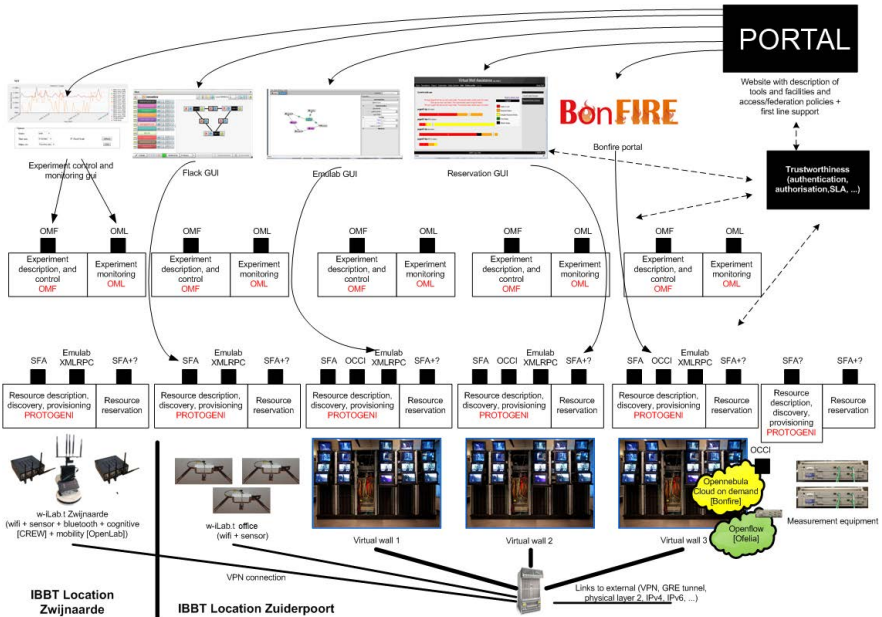


Fig. 7. Future federation vision: at the bottom we see the two wireless testbeds, the 3 Virtual Walls and measurement equipment. On top we see the resource provisioning through Emulab’s Protogeni and an extension for future reservation. Also shown, is the OMF layer for experiment and monitoring control. On top, the different graphical user interfaces (community, platform or project specific as e.g. Bonfire) are shown together with a Portal pointing the experimenter towards the right tools and a distributed trustworthiness layer.

facilities, federating the facilities has been a trigger to promote discussion and to better cooperate on future extensions to the facilities. As an example, the adoption of SFA was one of the drivers: an SFA interface for OMF is still under development, while the Protogeni SFA interface on top of Emulab is already very stable. In this way, both our wired and wireless testbeds will be reachable through SFA (see Figure 7). The gain of OMF experiment control for the wired experimenters will be very important. In the near future, we plan to organize workshops targeted to the w-iLab.t and Virtual Wall communities, in which the federation approach is explained and example federated experiments are provided. It is expected that future extensions to the iLab.t facilities will be useful to both the Virtual Wall and w-iLab.t communities, and that the federation will further stimulate cooperation. Also from a technical point of view, the local Virtual Wall – w-iLab.t federation is not an endpoint. In the future, the federation exercise will be repeated continuously at a larger scale. A first step will be to also integrate the w-iLab.t office in the federation. Next, federation at a larger scale, with experimental facilities owned by other legal entities will be pursued.

5 Conclusion

Driven by some very practical problems which arose when the IBBT iLab.t grew from individual experimenter testbeds, over an Emulab based Virtual Wall installation, to multiple large wired and wireless testbeds, we have studied and implemented the best way forward towards the future for controlling our testbeds. We ended up by using the Emulab software in a multi-testbed configuration with shared accounting (which makes it easier for upgrading and maintenance than one really big testbed) for the node provisioning and account/project handling topped of with the OMF/OML framework for controlling the experiments and measurements themselves. This federation was rather straightforward compared to extending one or another framework, and seems to be very promising for the testbed administrators and experimenter communities as know-how can be shared easily now to make the best use of all available frameworks. Besides this, we believe that also external federation through the Emulab Protogeni SFA interface will be a very interesting way forward towards a more global federation.

Acknowledgment. The research leading to these results has received funding from various national funds, and from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreements nr 258301 (CREW), nr 258365 (OFELIA) and nr 287581 (OpenLab).

References

1. Emulab. Home page, <http://www.emulab.net>
2. Bouckaert, S., Vandenberghe, W., Jooris, B., Moerman, I., Demeester, P.: The w-iLab.t Testbed. In: Magedanz, T., Gavras, A., Thanh, N.H., Chase, J.S. (eds.) TridentCom 2010. LNICST, vol. 46, pp. 145–154. Springer, Heidelberg (2011)
3. PC Engines. Alix system board, <http://www.pcengines.ch/alix.html>
4. Tytgat, L., Jooris, B., Mil, P.D., Latré, B., Moerman, I., Demeester, P.: Wilab, a real-life wireless sensor testbed with environment emulation. In: European Conference on WSNs, Cork, Ireland (February 2009)
5. Madwifi. Multiband atheros driver for wifi (2010), <http://madwifi.org/>
6. Werner-Allen, G., Swieskowski, P., Welsh, M.: Motelab: a wireless sensor network testbed. In: Fourth International Symposium on Information Processing in Sensor Networks, IPSN 2005, pp. 483–488 (April 2005)
7. The CREW project. w-ilab.t zwijnaarde, <http://www.crew-project.eu/portal/wilab/open-environment-testbed-zwijnaarde>
8. Rakotoarivelo, T., Ott, M., Jourjon, G., Seskar, I.: Omf: a control and management framework for networking testbeds. SIGOPS Oper. Syst. Rev. 43, 54–59 (2010)
9. OMF. The testbed control and management framework - home page, <http://omf.mytestbed.net>
10. White, J., Jourjon, G., Rakotoarivelo, T., Ott, M.: Measurement Architectures for Network Experiments with Disconnected Mobile Nodes. In: Magedanz, T., Gavras, A., Thanh, N.H., Chase, J.S. (eds.) TridentCom 2010. LNICST, vol. 46, pp. 315–330. Springer, Heidelberg (2011)