Making the Best of Two Worlds: A Framework for Hybrid Experiments

Razvan Beuran
Hokuriku StarBED Technology Center,
National Institute of Information and Communications Technology,
Ishikawa, Japan
razvan@nict.go.jp

Shinsuke Miwa
Hokuriku StarBED Technology Center,
National Institute of Information and Communications Technology,
Ishikawa, Japan
danna@nict.go.jp

Yoichi Shinoda
Research Center For Advanced Computing Infrastructure,
Japan Advanced Institute of Science and Technology,
Ishikawa, Japan
shinoda@jaist.ac.jp

ABSTRACT

In this paper we present the design and implementation of a framework for hybrid experiments that integrates a real-world wireless testbed with a wireless network emulation testbed. The real-world component of the framework, that we call *physical realm*, can be used for those experiment aspects that are difficult to perform through emulation, such as real-life communication conditions under the effect of perturbations and weather conditions. Correspondingly, the emulated part of the framework, that we call *emulated realm*, can be used for those characteristics that are difficult to address in the real world, such as technologies that are not yet available, large-scale mobility, or for reasons of financial cost. The paper includes a series of proof-of-concept experiments that demonstrate the feasibility of the proposed technique.

Categories and Subject Descriptors

C.4 [**Performance of Systems**]: Measurement techniques

Keywords

Hybrid experiments, real-world trials, wireless network emulation

1. INTRODUCTION

In recent years real-world trials have gained in importance, perhaps as a reaction to the dominance of simulation in network research. Despite the obvious advantages of validation through real-world trials, such as confidence in the realism of the experimental results, this approach still presents a number of difficulties, mainly related to experiment scale and repeatability, that discourage its extended use.

Network emulation has been proposed as an alternative experiment technique that bridges the gap between network simulation and real-world trials. Emulation combines the flexibility and control available in simulation with the realism of real-world trials, albeit through a trade-off.

WiNTECH'12, August 22, 2012, Istanbul, Turkey.

In this paper we propose a novel experiment technique that tries to make better use of the advantages of emulation and real-world trials by combining them into a single experiment platform that provides a unified communication environment. Real network applications and protocols are then run on top of this platform that seamlessly integrates an emulation testbed with a real-world testbed. Even though in this paper we focus on the application of the hybrid experiment concept to wireless networks – where we believe it is most beneficial – it is also possible to use this approach for doing hybrid wired network experiments.

The advantages of the hybrid experiment technique that we propose are multiple. Thus, the real-world components can be used for those aspects which are difficult to reproduce through other methods, such as: (i) real-life communication conditions under the effects of communication environment perturbations and weather conditions; (ii) technologies that are only available as physical implementations, such as prototypes or military equipment.

Correspondingly, emulation can be used for those experiment aspects that are impossible or difficult to perform in the real world. Examples in this category include: (i) technologies that are not yet available as physical prototypes, or whose use is prohibited in certain conditions (such as outdoor use of 802.11a in Japan); (ii) mobility of a large number of nodes and in large areas, which is difficult to recreate in a repeatable way in real-world trials; (iii) prohibitive financial costs related to large-scale deployments; (iv) deployment speed, for instance in network planning activities related to post-disaster network recovery.

In summary, our approach makes it possible to run experiments by partially using real wireless hardware, while compensating for its disadvantages in terms of scalability and mobility through the use of an integrated emulation platform. The main contributions of this paper are:

- We define the requirements for practically creating a framework for hybrid experiments;
- We detail the design and a prototype implementation that satisfies these requirements;
- We present a series of proof-of-concept experiments that demonstrate the feasibility of the proposed technique.

The remainder of this paper is organized as follows. In Section 2 we present the requirements of the hybrid experiment technique and our design that meets these requirements. In Section 3 we discuss the prototype implementation of the hybrid experiment framework based on our design. Then, in Section 4 we detail several experiments that validate the operation of our prototype implementation. In Section 5 we present the main directions of future work. A discussion of related work follows in Section 6. We end the paper with conclusions, acknowledgments and references.

2. FRAMEWORK DESIGN

The goals of our hybrid experiment framework are to create a platform that has all of the following features: (i) scalability; (ii) flexibility/control; (iii) repeatability; (iv) realism. In this section we present the requirements that must be met in order to achieve these goals, and a general design that addresses these requirements.

2.1 Requirements

As discussed in the introduction, there are multiple advantages of having a framework that makes possible combining emulation and real-world trials into a hybrid experiment platform. However, in order to benefit from all these advantages, the resulting framework must meet a series of requirements. In this context we shall use the following terms:

- *Physical realm*: Denotes the set of real-world components, including both the physical devices (wireless nodes) and the communication environment;

- *Emulated realm*: Represents the set of emulated components, including both the emulated devices and the emulated communication environment (propagation attenuation, obstacles, etc.).

Through the use of the physical realm, we unequivocally provide a way for users to make experiments with real equipment and in real-life conditions, hence with maximum realism. In its turn, the emulated realm provides the necessary features to achieve the other three goals mentioned above, namely scalability, flexibility/control and repeatability.

However, the use of the two realms together is not sufficient, and there are other requirements that must be met in order to create the intended hybrid experiment platform. We distinguish first a mandatory requirement, which is essential for the hybrid experiment technique. Thus, the framework must be able to create a *unified environment* that combines the physical and emulated realms into a single virtual entity. For this purpose a *communication mechanism* between the two realms is also necessary. This unification that creates a common communication environment differentiates our approach from other solutions that only juxtapose tools, such as a simulator running in parallel with a testbed.

There is also an optional requirement, whose fulfillment facilitates the use of the hybrid experiment framework. Thus, we recommend having a *unified control mechanism* for the real and emulated components of the framework.

2.2 Design overview

In Figure 1 we show the overview of our hybrid experiment framework design that addresses the above requirements. The following components are included:

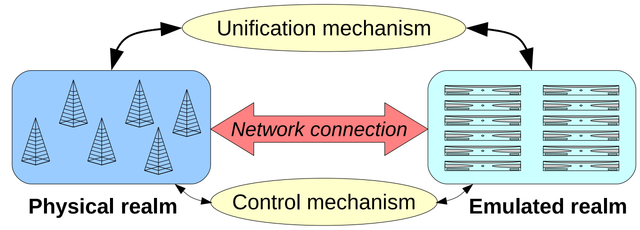

Figure 1: The generic design of the hybrid experiment framework.

1. A wireless network testbed for creating the physical realm, or an equivalent wireless-network deployment;

2. An emulation testbed for creating the emulated realm, typically a wired-network testbed with wireless network emulation capabilities;

3. A software component that integrates the physical and emulated realm into a unified virtual entity;

4. A network connection between the physical and emulated realms that makes possible exchanging information between the two;

5. A control mechanism that facilitates running experiments on the hybrid testbed constructed using the previous four components.

The components in Figure 1 interact as follows. The *physical realm* on the left-hand side executes the part of the experiment that is run on real wireless devices. The *emulated realm* on the right-hand side of the figure performs the part of the experiment that is run on emulated wireless devices. These two realms exchange information that flows between the physical and the emulated devices through a dedicated *network connection*.

The essential component that brings value added to the hybrid experiment framework is the *unification mechanism* that is depicted in the upper part of Figure 1. It is this component that helps meet the mandatory requirement outlined previously, namely to enforce the effects the realms have on each other from the point of view of communication (including aspects such as interference and channel contention), as it will be discussed in Section 3.3. Moreover, the *control mechanism* shown in the lower part of the figure helps meet the optional requirement we introduced in Section 2.1.

3. FRAMEWORK IMPLEMENTATION

In this section we discuss in detail the prototype implementation of the hybrid experiment framework.

3.1 Physical realm

The physical realm in our prototype hybrid experiment system is represented by a set of four wireless routers deployed within our center. The devices we selected for this purpose are Buffalo WZR-HP-AG300H, which are high-power multiple standard portable wireless routers supporting IEEE 802.11a/b/g/n communication. These routers are compatible with the open-source firmware called OpenWrt [7] that we used in our experiments (revision 30402 with a 2.6 series Linux kernel). The routers have two wireless interfaces and

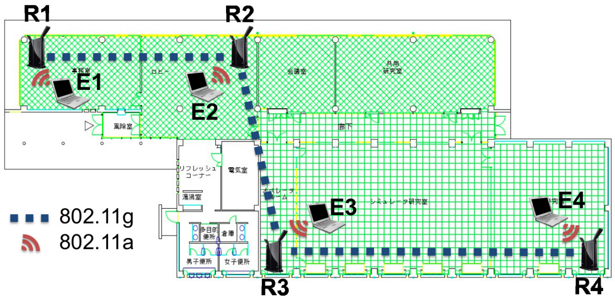

Figure 2: Wireless router deployment: the physical realm of our prototype hybrid experiment platform.

two wired interfaces, with one of them providing externally four ports (by using an internal switching chip).

The wireless routers, denoted by R1 to R4, were deployed in several points of our center so that they form a mesh network that covers entirely the area (see Figure 2). As mesh network technology we used the IEEE 802.11s implementation available in OpenWrt, which creates a Layer 2 mesh network. We used the IEEE 802.11g standard for the backhaul of the mesh network in order to provide better communication conditions through indoor obstacles and at large distances. The backhaul links are shown with dotted lines. Because the routers R1 and R3, and also R2 and R4 can occasionally communicate with each other, sometimes they became connected as well. In order to ensure route stability, we prevented such parasitic links through the use of the command "iw station plink_action block" .

Each wireless router in our deployment behaves both as a node in the mesh network (mesh point), and also as an access point. Figure 2 shows four wireless end nodes, denoted by E1 to E4, and connected to the wireless routers R1 to R4, respectively. Taking advantage of the two wireless interfaces of the Buffalo routers, we used the IEEE 802.11a standard for the connection between the end nodes and the access points. The end nodes were used during the real-world trials we performed, and were also emulated for our proof-of-concept hybrid experiments (see Section 4). Note that it is also possible to build a mesh network by using the same standard both for the backhaul and access, but its performance will be inferior.

3.2 Emulated realm

In our hybrid experiment framework implementation the emulated realm was created by using QOMB, a wireless network emulation testbed with a rich set of features, including IEEE 802.11 network emulation and node mobility [1]. The main components of QOMB are StarBED and QOMET.

StarBED is a large-scale wired-network testbed at the Hokuriku StarBED Technology Center of the National Institute of Information and Communications Technology, located in Ishikawa, Japan [5]. With over 1100 interconnected PCs, users can perform a wide range of network experiments on StarBED, which is the physical infrastructure of QOMB. A key characteristic of StarBED is that experiment traffic and control traffic are handled by different networks, so that there is no impact of the control traffic on the experiments. A software suite called SpringOS is used to manage experiments on StarBED.

QOMET (Quality Observation and Mobility Experiment Tools) is a set of tools for wireless network emulation that can be executed on StarBED [1]. QOMET allows the definition of various complex scenarios, including experiments with various wireless technologies, with node mobility and in urban settings. The most important components of QOMET are the libraries called *deltaQ* and *wireconf*. The deltaQ library is in charge of computing the communication conditions between wireless nodes given a user-defined XML-based scenario. These communication conditions are recreated during the real-time experiment by the wireconf library, which applies the corresponding network degradation, i.e., packet loss, delay and bandwidth limitation, to the experiment traffic; this is effectively done by means of the dummynet/ipfw3 system [4].

Because IEEE 802.11s emulation is not yet supported in QOMET, creating a hybrid experiment platform by combining the physical realm described in Section 3.1 with the emulated realm realized using QOMB is a good example of the power of this approach. Thus, the hybrid technique makes it possible to perform experiments that include a Layer 2 mesh network – that would not have been possible at this moment through emulation only – but that are not limited in terms of the total number of nodes by the size of the available physical deployment.

3.3 Realm unification

The integration of the physical and emulated realms so that they form a virtual unified environment is the most important and the most challenging task related to the hybrid experiment technique. In order to achieve this integration, the following conditions must hold:

- The influences of the nodes in the physical realm, such as interference and contention, must affect communication in the emulated realm;
- Similarly, the influences of the nodes in the emulated realm must affect communication in the physical realm.

3.3.1 Architecture components

The overall architecture of our implementation that addresses these conditions is shown in Figure 3. The key elements in the figure are: (i) the wireless routers in the physical realm discussed in Section 3.1, generically denoted by "Rt"; (ii) the StarBED experiment hosts that form the emulated realm detailed in Section 3.2, denoted by "Exp"; (iii) the modules denoted by "QMT" that run on both the wireless routers and the experiment hosts in order to assist the realm unification mechanism, as it will be explained below; (iv) the experiment and control networks that ensure inter-realm connectivity (see Section 3.4); (v) the experiment-support software SpringOS and the StarBED hosts labeled "Ctrl" that mediate the control of the wireless routers, which form the control mechanism of our experiment framework (see Section 3.5).

The wireless routers "Rt" have two wireless interfaces, one named "mesh0", which ensures connectivity to the mesh network backhaul, and another one named "wlan1", which allows clients to connect to the routers acting as access points. The wired interface "eth0" connects the routers to the experiment network in StarBED. Due to infrastructure difficulties, the routers in our deployment are not connected directly to the control network in StarBED, and a mediation mecha-

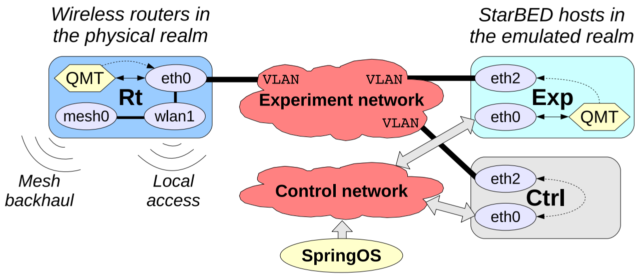

Figure 3: Architecture of the realm unification mechanism implementation.

nism using the control hosts "Ctrl" was required, as it will be detailed in Section 3.5.

The experiment hosts "Exp" are used for running the emulated realm. They have several network interfaces, including a wired interface "eth0" used for control traffic, and a wired interface named "eth2" used for experiment traffic; for clarity purposes the remaining interfaces are not shown here.

The key components of the unification mechanism are the "QMT" modules. The experiment hosts in StarBED could already use these modules, that include the deltaQ and wireconf libraries of QOMET, to control the communication conditions of the emulated nodes according to a user-defined scenario. For this purpose the wireconf library gathers statistics about channel utilization from each emulated node. These statistics are sent via multicast in the control network to all the other experiment hosts. Then the channel utilization information is used by the deltaQ library to adjust the communication conditions in real time by inferring the effects of the corresponding channel utilization in terms of interference and contention. This mechanism was previously detailed in [6].

3.3.2 *Unification mechanisms*

In order to achieve the realm unification functionality required for hybrid experiments, the physical realm must be integrated in the mechanism discussed above. This meant that we had to run the two QOMET libraries on the wireless routers "Rt" as well. Since the ipfw3 system that wireconf uses has support for OpenWrt, our first task consisted in compiling ipfw3 for the OpenWrt version that we used. The second task was to port the QOMET deltaQ and wireconf libraries to OpenWrt. Both tasks only required minor changes to the respective source code.

Leveraging the above mechanism, realm unification works as follows. As "QMT" modules run both on physical and emulated nodes when using the hybrid experiment platform, all the nodes in the experiment, both emulated and real, become aware of traffic being generated by the other nodes via the statistics collected by the wireconf library. In addition, for hybrid experiment purposes an extension is required as follows: the difference between real nodes and emulated ones is that real nodes *must ignore* the statistics regarding the other real nodes, because their influence is accounted by the natural interference and contention in the physical wireless domain (this aspect is currently under development).

One more issue that needs to be addressed is the communication in the unified environment. The emulated nodes can communicate with each other and with the real nodes via the experiment network of StarBED. In this context it is necessary to prevent the direct communication of the emulated nodes when this is not supposed to take place in the emulated scenario. A typical example is the case when an emulated node is "associated" to a real node acting as an access point. In this case, the communication must not take place directly but via the access point. We create this constraint by using VLANs, which is the standard mechanism for defining network topologies on StarBED. An emulated node and the real access point to which it is associated will be placed in the same VLAN, thus preventing it from communicating directly with an emulated node that is connected to a different access point; instead communication will take place via the mesh network backhaul. The only issue with this mechanism is that two emulated nodes connected to the *same* real access point will be able to communicate with each other directly, because they will be in the same VLAN. Our proposed solution is to add appropriate rules in ipfw3 so as to prevent direct communication (see Section 5).

In the context of the unification mechanism mobility is an important aspect. When mobility only takes place in the emulated realm, then the emulation mechanisms that are already in place will take it into account. However, if one wishes to have mobility in the physical realm, it has to take place according to a predefined scenario, or a specific mechanism must be created that communicates the position of the mobile physical node to the other experiment nodes.

3.4 Inter-realm connection

The network connection between the physical and emulated realms is realized by connecting the four wireless routers in the physical realm presented in Section 3.1 to the StarBED experiment network infrastructure of the emulated realm introduced in Section 3.2.

The main role of the inter-realm connection is to make sure that information can flow between the two realms. This holds both for the experiment traffic, for instance sent by the physical nodes to the emulated ones, and for the control traffic, which is used in order to create a unified communication environment, as it was detailed in Section 3.3. However, due to practical difficulties the physical realm could not be connected directly to the control network, and a special solution was used, as discussed next.

3.5 Control mechanism

As control mechanism for our hybrid experiment framework we used SpringOS, which is the main experiment support software tool for StarBED [5]. SpringOS provides various mechanisms that make it possible for users to easily perform complex experiments with a large number of hosts. SpringOS uses a client-server architecture to drive experiment execution. Since SpringOS was developed for StarBED, it is fully-supported for making experiments on QOMB, hence for using the emulated realm. Using SpringOS in the physical realm means first of all making the real nodes accessible via the control network, and secondly porting the SpringOS client to OpenWrt (under development).

Because a direct connection to the control network was not possible, we used a set of hosts in StarBED (one per wireless router) to create ssh tunnels between the control network and the wireless routers. These hosts (denoted by "Ctrl" in Figure 3) have the interface "eth0" connected to the control network, and the interface "eth2" connected to the experiment network and in the same VLAN with the wireless router it mediates access to. Although this solution does not allow to use the full power of SpringOS, it makes possible to execute remotely commands on the wireless routers, hence simplifies considerably experiment execution.

4. EXPERIMENTAL RESULTS

In this section we present several experiments that demonstrate the feasibility of our hybrid experiment platform. The experiments were performed both as real-world trials, in which all the wireless network devices used were real, and as hybrid experiments, in which some of the wireless network devices were real and some were emulated.

The Buffalo wireless routers R1 to R4 mentioned in Section 3.1 (see Figure 2) represent the physical realm of our hybrid experiment platform, and they were used as such both in the real-world trials and the hybrid experiments.

The end nodes shown in Figure 2, denoted by E1 to E4, were either physical devices or emulated nodes, depending on the experiment. In the real-world trials we used Mac notebooks that connect via 802.11a technology to their respective access points. In the hybrid experiments we used experiment hosts on StarBED running CentOS 6.2 with a 2.6 series Linux kernel. In both cases the traffic produced by the end nodes was forwarded over the mesh backhaul created by the real wireless routers R1 to R4. In summary, for both experiment types the 802.11g mesh backbone was real, whereas the 802.11a nodes were either real or emulated.

The real-world trials took place indoor in our center, shown in Figure 2. Its length is of about 50 m (horizontal axis in the figure), and its width is of about 20 m (vertical axis in the figure). The distance between the wireless routers is of about 20 m from R1 to R2 and from R2 to R3, and of about 30 m from R3 to R4. Their placement lead to good direct communication only over one hop at the transmit power level of 18 dBm. For the emulation involved in hybrid experiments we modeled the building environment and considered the attenuation coefficient for electromagnetic wave propagation to be equal to 4.02, a typical parameter for indoor environments, which resulted in conditions that were similar with the real world.

4.1 Single sender

In the first series of experiments we have sent traffic from one of the end nodes E1, E2, and E3 (connected to the wireless routers R1, R2, and R3, respectively) to the end node E4 (connected to the router R4). The traffic was generated using iperf-2.0.5 [10], with the iperf server being deployed on E4 and the iperf client on one of E1, E2, or E3. We run tests using either TCP or UDP traffic, with a test duration of 1 minute and three runs per test. In the case of UDP the offered load was 20 Mbps. All the other parameters had default values.

The results of both the real world and hybrid experiments are shown in Figure 4. For each pair of end nodes we show the average throughput in Mbps calculated for each experiment type.

The tests from E1 to E4 show that there is an very good match between the hybrid experiments and the real-world trials. As expected UDP throughput is higher than that for TCP, since the inelastic nature of UDP enables it to push more data through the network. The results for the experiments from E2 to E4 show a similar trend, but the absolute values are higher, because communication is done over 2 hops of the mesh backhaul instead of the 3 hops in the previous case. As for the results of the experiments from E3 to E4, although the match between hybrid experiments and real-world trials is good again, we do notice a difference of about 1 Mbps in the case of TCP throughput. We are still investigating the cause of this particular discrepancy, which we currently assume to be related to the sensitive nature of TCP to the differences in execution platforms (e.g., physical and implementation characteristics). Such differences become more important in good communication conditions, such as the 1 hop communication between E3 and E4, but are less significant in the more difficult communication conditions of the previous experiments.

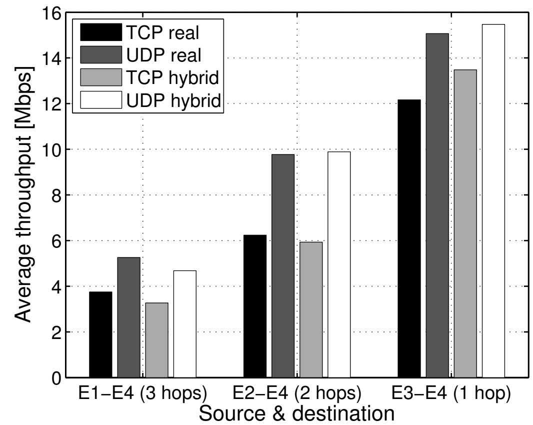

Figure 4: Average throughput results for experiments with single senders.

4.2 Multiple senders

Another series of experiments focused on the case when there are multiple traffic senders. Thus, we set up an experiment in which node E2 starts sending TCP traffic to router R1, followed after 20 s by node E3 which starts sending TCP traffic to router R4. Traffic was again generated by using iperf-2.0.5, with the iperf client running on the end nodes, and the iperf server running on the routers. Each end node sent traffic for 60 s, and the total experiment duration was 80 s. This experiment is intended to show how interference is handled on the hybrid experiment platform. As before, the end nodes in the real-world trials were Mac notebooks, and in the hybrid experiments they were StarBED hosts.

The results are displayed in Figure 5. Both in real-world trials and in hybrid experiments the throughput from E2 to R1 is around 14 Mbps during the first 20 s. Then E2 throughput drops to around 10 Mbps starting at second 20, as end node E3 starts generating traffic, and continues at this level until node E2 ceases traffic generation at time 60 s. On the other hand, node E3 has a throughput of about 10 Mbps between second 20 to 60, due to the contention with E2. Starting at second 60 throughput raises to 14 Mbps, as E3 becomes a single sender for that time interval. The results of the real-world trials and the hybrid experiments match very closely, indicating that interference and contention take place on the hybrid experiment platform in a similar manner to the real world.

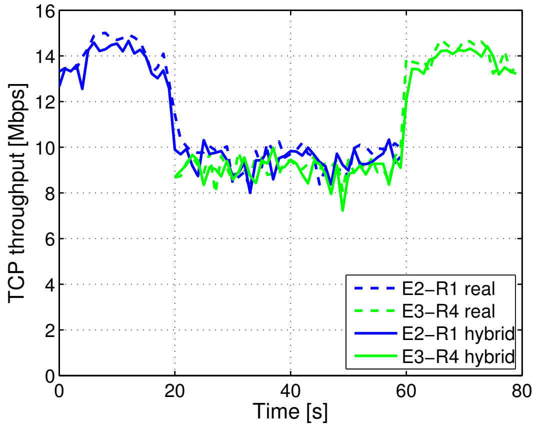

Figure 5: TCP throughput results versus time for experiments with multiple senders.

5. DISCUSSION & FUTURE WORK

In this paper we have presented the design of a hybrid experiment framework, but the current implementation is only a prototype that demonstrates its feasibility, and our development work continues. We detail here those aspects that are currently under development.

The deployment in our center discussed in Section 3.1 is only for validation purposes. We intend to build a larger-scale wireless network testbed with about 100 nodes in the outdoor area around our center. This future testbed, that will be called "AirBED", is intended to be an open-access testbed that can be used by various users for various wireless network experiments in real-life conditions. The hybrid integration of this testbed with StarBED and QOMB will provide an added value compared to typical wireless testbeds that are limited by their scale and hardware characteristics.

The realm unification mechanism in Section 3.3 requires additional ipfw3 filters that are needed in order to ensure that communication between the multiple emulated nodes takes place in accordance with the experiment scenario. An advantage of this method is that topology changes can be applied faster then in the case of VLANs (which require reconfiguring the network infrastructure), hence it can be used for emulating other network features, such as handover.

The only drawback of the current inter-realm connection mechanism is that the physical realm is not directly connected to the StarBED control network. Such a connectivity is planned for the AirBED deployment mentioned above, as it would significantly simplify the management tasks, and it would create the separation from the experiment network, thus avoiding perturbations.

In order to fully integrate the physical realm with the emulated one from a control point of view, the SpringOS experiment-support software will be ported to OpenWrt, so that the wireless nodes in AirBED can be used transparently in the same manner with the StarBED nodes.

Once the development of the hybrid experiment platform is finalized we plan to conduct more detailed experiments with it focusing on issues such as: (i) mobility, both in the emulated realm and in the physical one; (ii) seamless integration of the physical and emulated realms in scenarios that use the same wireless spectrum in both realms.

6. RELATED WORK

To the best of our knowledge there is no system which has an identical functionality with our hybrid emulation framework. However, there are several projects which have similar proposed goals or approaches. We include below several representative examples.

6.1 Wireless testbeds

ORBIT is the largest open-access wireless network testbed, with more than 400 nodes available for experiments in a 20 x 20 m area [8]. What distinguishes it from other testbeds is that ORBIT makes available several mechanisms to create communication conditions that differ from those naturally available. For instance, through the use of noise generators it is possible to raise the noise floor level in order to create communication conditions that are similar to those between nodes located far away from each other, even though the nodes do not move physically. One advantage of ORBIT is that the use of real wireless interfaces make it possible to do PHY/MAC layer experiments, something which is not possible in the emulation realm of our platform, but could be done in the physical realm.

Since all the ORBIT nodes are placed in a single limited location (a network topology that differs significantly from real ones), scale is obtained by trading off realism. Moreover, due to various factors, repeatability of the experiments run on ORBIT is only partial, as shown in [3]. One other issue with ORBIT is that it doesn't really support node mobility, since only a "virtual" motion of nodes can be created by activating different nodes on the grid at different moments of time. On the contrary, our framework makes possible experiments at scale (including mobility) in the emulated realm, while having real-life communication conditions in the physical realm.

Emulab is another network testbed that focuses on wired network experiments, but also includes wireless nodes [12]. Even though the wired and wireless Emulab nodes can be used in the same experiment, in Emulab there is no unification mechanism similar to that in our approach. Therefore the environment built from the wired (even when in emulation mode) and wireless nodes does not represent a single virtual world, hence it provides only a quantitative extension of the wireless testbed but not a qualitative one.

6.2 Hardware-in-the-loop simulation

Several network simulators, such as QualNet [9], offer the possibility to do so-called "hardware-in-the-loop" simulations. This approach connects an experiment environment that is simulated in real time with real network devices via interfaces that have the capability to inject packets from the simulation into the physical network nodes and viceversa. However, since the real devices are not integrated with the simulated environment, only their traffic is taken into account, and not their overall influence. Our hybrid approach doesn't have this drawback since it seamlessly integrates the emulated realm with the physical one.

6.3 Simulation-augmented emulation

The authors of RFnest have used a hybrid approach to scale up the capabilities of an FPGA-based network channel emulator with virtual simulated nodes that are run on the same platform [11]. The hardware component of RFnest uses 8 RF interface nodes that are inter-connected via a filter bank which makes it possible to control the communication conditions between the nodes by adding delay and attenuation. The simulated virtual nodes can interact with the real nodes, and help increasing the number of nodes in an experiment.

The integration of the simulated nodes with the emulated ones is mediated by means of "surrogate virtual transmitters and receivers" that act as intermediary for any packets sent from a virtual node to a real one and viceversa. Our hybrid framework takes this approach one step further, and integrates real nodes with emulated ones, to provide more realism of the communication conditions and better scalability characteristics.

6.4 Remote emulation

The authors of [2] present an emulation framework called Airplug for vehicular network experiments. In this framework the user creates the necessary application modules, such as for packet generation or mobility management, that are then executed on the experiment nodes. The core program of Airplug manages all the inter-application communication of these nodes through a kind of overlay network.

Airplug is able to use a "remote mode" in which an instance deployed on a real node can communicate with other instances, real or emulated, via sockets. However, what is being deployed remotely by Airplug are the applications themselves, not the emulation capabilities, which are not addressed. Moreover, since Airplug requires writing specific applications, it cannot be used with normal network applications and protocols as is the case of our hybrid experiment framework.

7. CONCLUSION

In this paper we have defined the requirements that must be met in order to practically create a framework for hybrid experiments that combines real nodes from a so-called *physical realm*, and emulated nodes from a so-called *emulated realm*, into a single unified experiment platform. This is done by means of a unification mechanism that enforces the effects the realms have on each other from the communication point of view. We have also presented a generic design that addresses these requirements.

We have then detailed a prototype implementation that follows these design guidelines. This implementation integrated a four wireless router physical deployment with the large-scale wireless network emulation testbed called QOMB. The prototype implementation was used for a series of proof-of-concept experiments that validated the feasibility of the hybrid experiment approach, and emphasized some of it advantages.

We are currently extending the prototype implementation to a full-scale implementation of a hybrid experiment system. The main enhancements focus on a more generic unification of the physical realm with the emulated one, as well as on the better integration of the physical realm with the control mechanisms of QOMB. Further experiments will be conducted once the implementation is finalized so as to evaluate the seamless integration in the hybrid experiment framework, including in the context of mobility.

8. ACKNOWLEDGMENTS

The authors would like to thank Marta Carbone for her support in compiling ipfw3 on OpenWrt. We also thank Muhammad Imran Tariq for his assistance in performing the real-world trials presented in this paper.

9. REFERENCES

[1] R. Beuran, L. T. Nguyen, T. Miyachi, J. Nakata, K. Chinen, Y. Tan, Y. Shinoda, *QOMB: A Wireless Network Emulation Testbed*, IEEE Global Communications Conference (GLOBECOM 2009), Honolulu, Hawaii, USA, November 30-December 4, 2009.

[2] A. Buisset, B. Ducourthial, F. El Ali, S. Khalfallah, *Vehicular Networks Emulation*, International Conference on Computer Communications and Networks (ICCCN) 2010, August 2-5, 2010, pp. 1-7.

[3] R. Beuran, Y. Tan, Y. Shinoda, *Challenges of Using Wireless Network Testbeds: A Case Study on ORBIT*, 6th ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization (WiNTECH 2011), in conjunction with MobiCom 2011, Las Vegas, Nevada, USA, September 19-23, 2011, pp. 11-18.

[4] M. Carbone, L. Rizzo, *Dummynet revisited*, ACM SIGCOMM Computer Communications Revue, Vol. 40, No. 2, April 2010.

[5] T. Miyachi, K. Chinen, Y. Shinoda, *StarBED and SpringOS: Large-scale General Purpose Network Testbed and Supporting Software*, International Conference on Performance Evaluation Methodologies and Tools (Valuetools 2006), ACM Press, Pisa, Italy, October 2006.

[6] L. T. Nguyen, R. Beuran, Y. Shinoda, *AEROMAN: A Novel Architecture to Evaluate Routing Protocols for Multi-hop Ad-hoc Networks*, 19th International Conference on Computer Communications and Networks (ICCCN 2010), Zurich, Switzerland, August 2-5, 2010.

[7] OpenWrt, *Web page*, https://openwrt.org

[8] D. Raychaudhuri, I. Seskar, M. Ott, S. Ganu, K. Ramachandran, H. Kremo, R. Siracusa, H. Liu, M. Singh, *Overview of the ORBIT Radio Grid Testbed for Evaluation of Next-Generation Wireless Network Protocols*, IEEE WCNC 2005, New Orleans, Louisiana, USA, March 2005.

[9] Scalable Network Technologies, *QualNet Developer*, Inc., http://www.scalablenetworks.com.

[10] SourceForge, *Iperf Homepage*, http://sourceforge.net/projects/iperf/

[11] J. Yackoski, B. Azimi-Sadjadi, A. Namazi, J. H. Li, Y. Sagduyu, R. Levy, *RFnest^{TM}: Radio frequency network emulator simulator tool*, Military Communications Conference (MILCOM) 2011, Baltimore, MD, USA, November 7-10, 2011, pp. 1882-1887.

[12] B. White, J. Lepreau, L. Stoller, R. Ricci, S. Guruprasad, M. Newbold, M. Hibler, C. Barb, A. Joglekar, *An Integrated Experimental Environment for Distributed Systems and Networks*, Symposium on Operating Systems Design and Implementation (OSDI02), Boston, Massachusetts, USA, December 2002, pp. 255–270.