Challenges of Using Wireless Network Testbeds: A Case Study on ORBIT

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

Yasuo Tan
School of Information Science, Japan Advanced Institute of Science and Technology
Nomi, Ishikawa, Japan
ytan@jaist.ac.jp

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

ABSTRACT

Wireless network testbeds have emerged as a valuable alternative to network simulation, but actually using them for experiments presents a number of challenges. We use the ORBIT testbed at Rutgers University, New Jersey, USA as a case study to illustrate the most important issues in this context. We performed several types of evaluation tests on ORBIT, as well as a series of equivalent experiments with up to 50 nodes, which were done also on the QOMB wireless network emulation testbed, and by using the QualNet network simulator. We conclude that a testbed such as ORBIT provides a viable approach to network testing, although this is conditioned by an awareness of the potential pitfalls that may affect experimental results; a series of lessons we learned from our experience is included in the paper. We also show that real-world trials on ORBIT can be broadly reproduced through emulation and even through simulation.

Categories and Subject Descriptors

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

General Terms

Experimentation, Measurement, Verification

Keywords

Wireless network testbed, wireless network emulation, wireless network simulation, OLSR

1. INTRODUCTION

In addition to network simulation experiments, researchers traditionally use custom setups to perform real-world network experiments. Such custom settings make it often impossible for peers to reproduce the results, since it implies building an identical experimental setup. This is particularly true for wireless network experiments, where creating an identical setting refers not only to using the same hardware, but also to using the same placement for the hardware, and to recreating the same communication conditions between the wireless nodes.

Open-access wireless network testbeds have been proposed as a solution that ensures result reproducibility by allowing researchers to use a common experiment platform. While this may be true at first sight, a deeper analysis reveals several issues. Using a common testbed guarantees having the same network hardware, but not the same network conditions. Moreover, result reproducibility is not the only aspect related to network experiments. How representative the results are, and how they can be used to derive general observations and conclusions, are also important elements.

In this paper we use the ORBIT wireless network testbed at Rutgers University, New Jersey, USA [8] as a case study to investigate the aforementioned issues. We selected ORBIT for this purpose since, to the best of our knowledge, it is the largest and most-used open-access wireless network testbed. To investigate the challenges of using ORBIT we performed several types of evaluation tests in simple setups that demonstrate the influence the different testbed components and conditions have on experimental results. For each type of test we summarize our conclusions, and present recommendations based on the lessons we learned.

To further examine the challenges related to wireless network testbeds, we also carried out a series of equivalent experiments with up to 50 nodes on ORBIT, on QOMB – the wireless network emulation testbed at the Hokuriku StarBED Technology Center of the National Institute of Information and Communications Technology, in Ishikawa, Japan [2] – and by using the QualNet network simulator [10]. The analysis of the results provides an insight into several aspects related to wireless network experiments, especially regarding the significance of experimental results.

The contributions of this paper are as follows:

- An evaluation of several experiment tools and testbed mechanisms available on ORBIT, and of their interactions;
- A comparative analysis of experimental results obtained using the OLSR protocol on ORBIT with those from emulation and simulation;

WiNTECH'11, September 19, 2011, Las Vegas, Nevada, USA.

- A series of recommendations for testbed users aimed to ensure that experiments are carried out effectively, and that their results are meaningful.

Our general conclusion is that a testbed such as ORBIT provides a viable approach to network testing, especially as a platform for testing algorithm implementations on real hardware and at large scale. However, the usefulness of the experimental results is conditioned by an awareness of the potential pitfalls that may affect them. We also show that real-world trials on ORBIT can be broadly reproduced through emulation and simulation, even though the match between emulation and simulation themselves is much better.

The remainder of this paper is organized as follows. Section 2 summarizes the most important properties of ORBIT, and of the tools and mechanisms that were used in our investigation. We proceed in Section 3 to evaluate the interactions between the different testbed components and experiment tools. In Section 4 we discuss several experiments aimed at assessing the properties of the ORBIT testbed as a whole. Then we present a comparative analysis of equivalent experiments that we have performed using ORBIT, QOMB, and QualNet (Section 5). We end the paper with sections of conclusions, acknowledgments, and references.

2. ORBIT OVERVIEW

ORBIT is a wireless network testbed developed and operated by WINLAB at Rutgers University, New Jersey, USA [8]. The main radio grid of ORBIT has 400 nodes positioned at 1 m intervals in a 20 x 20 m indoor area (see Figure 3, where each node is represented by a square). The nodes are equipped with several wireless network interfaces, typically IEEE 802.11a/b/g cards by Atheros or Intel. Although there are other types of wireless interfaces, and also other test facilities operated by ORBIT, such as sandboxes and an outdoor testbed, in this paper we shall focus on the testbed represented by the main radio grid, and use the name ORBIT to refer to it, unless otherwise indicated in the context.

2.1 Noise Generation

A specific feature of ORBIT is that one can use noise generation in order to artificially inject electromagnetic noise. The main purpose of using artificial noise is to raise the noise floor of the communication environment, and thus create an effect similar to that of having larger distances between the wireless nodes. This feature is particularly important given the reduced size of the testbed, which creates unrealistically crowded wireless setups when using a large number of nodes.

The triangles labeled 'N' in the corners of the ORBIT grid in Figure 3 show the position of the four antennas connected to the noise generator. Individual attenuators can be configured for each of the antennas to create asymmetrical noise effects.

We wish to emphasize here that the effect of increasing distances through the use of noise generators is not a uniform one, i.e., it is not equivalent to increasing equally the distances between every node. Instead, the nodes in the vicinity of the noise generator antennas, where the noise level is higher, will be virtually "farther away" from each other compared to the nodes which are far from the noise generator antennas (those in the middle of the ORBIT grid). The latter nodes will be less affected, hence not so much distanced virtually, because the noise power level decreases proportionally with the square of the distance. A consequence of this effect will be shown in Section 5.3.

2.2 OMF

Similarly to other large testbeds, ORBIT uses a specific framework for control, measurement and management named OMF (cOntrol and Management Framework). The main two roles of OMF are:

For testbed users Provide a set of tools to describe, deploy and configure an experiment, then to execute the experiment, and to collect its results.

For testbed operators Provide a set of services for efficiently managing and operating the testbed resources.

Although originally developed exclusively for ORBIT, starting from 2007 OMF is being actively extended to operate on testbeds with different type of network and resource technologies.

More practical details about the operation of ORBIT via OMF can be found on the testbed's website [12]. However, most up-to-date information regarding OMF is available on an OMF-specific website [6]. We note that, at the time of writing, the latest OMF version was 5.3; however, ORBIT supported only the legacy version 4.4, as well as the newer version 5.2 (recommended). For all the experiments presented in this paper we have used OMF version 5.2.

3. KNOW YOUR TOOLS

A testbed's hardware is not sufficient for making experiments. Several software tools are necessary, such as a management framework and network measurement tools. In this section we analyze the influence of such tools on the experimental results for the practical case of ORBIT.

3.1 OMF Measurements

As mentioned in Section 2.2, OMF is the control, measurement and management framework of ORBIT, and as far as we know the only supported way of conducting experiments on ORBIT. In this section we focus on the traffic measurement features of OMF. To generate and receive traffic we employed the tools called OTG (Orbit Traffic Generator) and OTR (Orbit Traffic Receiver), respectively, which are included with the OMF framework.

We configured the OMF measurement tools to log basic data (time, packet size, source and destination IP addresses, etc.) for each generated and received packet, similar to what one is able to do by using a wireless network analyzer. Our experiment used two nodes on the main ORBIT grid located at about 1.5 m from each other to generate and receive 1024 byte UDP packets for a period of 1 minute. Other relevant node settings were 8 dBm transmit power and 54 Mbps operating rate (IEEE 802.11g standard).

Table 1 shows the results obtained in two extreme cases, as follows. In experiment A, the generator was configured to send 8 kbps traffic; the results are close to expectations, although the 7% loss may seem somewhat high. In experiment B, the generator was configured to send at 54 Mbps rate; while the average sent rate of about 36 Mbps may seem reasonable (although higher than one would expect given the overhead of the wireless technology), the received rate is very low, and as a result packet loss is close to 100%.

Table 1: Average OMF measurement results

Exp.	Sent [kbps]	Received [kbps]	Loss [%]
A	8.2	7.6	7.0
B	36494.7	37.5	99.9

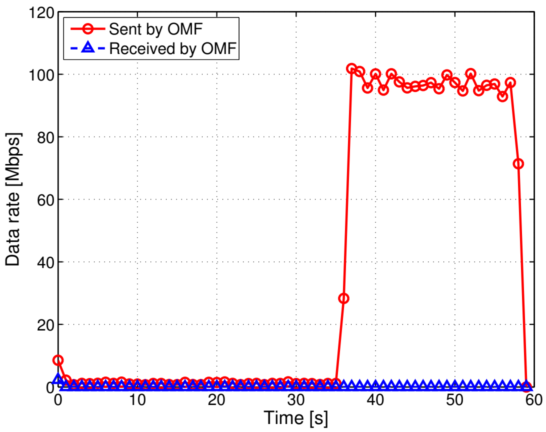

Figure 1: Data rate versus time for 54 Mbps offered load in OMF.

A closer look at the dynamic behavior of the sender and receiver versus time provides an insight into why the above results were obtained in experiment B. Figure 1 makes it obvious that the OMF measurement system is unable to cope with the high transmission rate configured for the traffic generator. As such, for half of the experiment duration, the reported sending rate is very low (around 1 Mbps), and jumps to around 100 Mbps starting at 35 s, which is an unrealistic value given the 54 Mbps operating rate used. Moreover, during most of the experiment period, the reported received rate is 0 Mbps.

Given all these, we conclude that the OMF measurement system cannot handle rates above approximately 1 Mbps if basic data is to be logged for *all* the generated packets. One possibility would be to configure the OMF measurement system to only provide average values over certain intervals (such as every 1 s), but this reduces its usefulness, since short-term dynamic effects will go unnoticed. Moreover, this approach diminishes the advantage of having a measurement tool integrated with the testbed, and brings OMF closer to a generic network measurement tool.

3.2 Iperf Measurements

To the above considerations we add the fact that using the OMF measurement system makes it difficult to directly compare the ORBIT results with those obtained in arbitrary field trials, when OMF cannot be used. Therefore, we decided to also make measurements with the widely-used network testing tool called "iperf" [11], for which some support is already present on ORBIT.

In particular, the OMF distribution includes the legacy iperf version 1.7.0. For comparison purposes, we also compiled and installed on ORBIT the latest iperf version, which at the time of writing was iperf 2.0.5.

We show in Figure 2 a comparison of the results obtained with the two versions of iperf. Note that iperf uses a client-server architecture, with the client generating traffic, and the server receiving it and making throughput calculations. We identified several scenarios of interest in relation with the use of OMF to perform experiments:

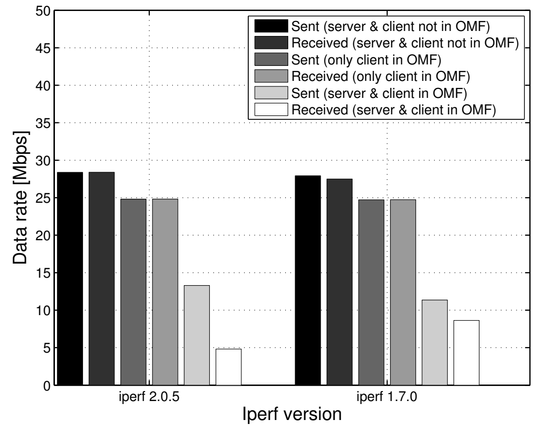

Figure 2: Data rate for 54 Mbps offered load in iperf versions 2.0.5 and 1.7.0.

1. Both the iperf server and client are run manually, without using OMF mechanisms;
2. Only the iperf client is run using OMF mechanisms, whereas the server is run manually;
3. Both the iperf server and client are run using OMF mechanisms.

Figure 2 shows the traffic measurement results as reported by the two versions of iperf in the three cases above. In all cases the respective iperf client was configured to generate 54 Mbps UDP traffic with 1024 byte packets for a 1 minute period. Other relevant node settings were 0 dBm transmit power and 54 Mbps operating rate (IEEE 802.11g standard). Note that, in order to prevent most interferences, for these experiments we used two nodes in one of the sandboxes of ORBIT, sb1, for which antennas are connected to each other via cables.

One conclusion of these tests is that the two versions of iperf have a similar behavior. The other conclusion is that, when both the iperf server and client are run using OMF mechanisms, the throughput is the lowest. We speculate this is caused by the overhead OMF itself adds to the execution of the iperf client and server, leading to an offered load lower by almost 60% compared to the other cases, and to an even lower received amount of traffic. The best results in our tests are obtained when both the iperf server and client are run manually. The throughput in these conditions is higher by about 12% compared to the case when the iperf server is executed manually, but the iperf client is run from within OMF, hence still subjected to its overhead.

3.3 Discussion

Given the poor performance of the OMF measurement system in high offered-load conditions, we decided against using it in our experiments, and opted for using iperf instead (namely version 2.0.5). This was mainly because our subsequent experiments did focus on high-load scenarios. Iperf also has the advantage of being a generic measurement system, usable in other circumstances as well.

In addition, considering the fact that running the iperf client manually is tedious when one wants to perform long series of experiments (because of the need to synchronize with the other operations of the sender node), we decided to use in our subsequent experiments OMF-based execution for the iperf client, and manual execution for the iperf server. This is a reasonable usability trade-off, since the server can be started once for an entire series of experiments. We consider that the lower throughput by about 12% that we measured for this usage is acceptable, and within typical variations of the throughput amongst various experimental platforms [2].

The lessons we have learned from the tests discussed in this section can be summarized as follows:

1. Users should evaluate all the experiment tools before actually using them, in conditions similar to those they plan to employ in their future experiments. Only this guarantees that the potential limitations and overhead of the tools will have no unexpected effects on measurements;

2. Sometimes users have to sacrifice performance for usability in order to make possible running long series of experiments with ease. The trade-off must however be made with care that its consequences are well understood.

4. KNOW YOUR TESTBED

For a measurement instrument, one has to know how it can be employed, and what kind of measurements it can be used for. Similarly, for a network testbed one has to understand what type of experiments can be performed on it, and in what conditions. In this section we discuss the challenges related to wireless network testbed usage, focusing again on ORBIT as a case study.

4.1 Basic Measurements

The basic components of a testbed are the wireless nodes. Therefore, the first step in understanding a testbed is quantifying the properties of the wireless nodes, in particular those of the wireless adapters with which they are equipped.

At a first level, the most important properties regarding the communication performance of a wireless adapter are the transmit power settings and receive sensitivity thresholds for each operating rate. This information is made available by most manufacturers for many of their adapters. Other characteristics of the wireless adapters (buffer sizes, implementation details), and of the PCs in which they are installed (operating system, CPU), also influence performance to a certain extent. However, as they are difficult to generalize, and often not explicitly related to the wireless communication process itself, we shall not discuss them here.

The following types of experiments can be used to understand the properties of wireless adapters:

1. Fix the transmit power, and make measurements for several operating rates;

2. Fix the operating rate, and make measurements for several transmit power settings.

In our experiments on ORBIT we used nodes with Atheros AR5212 wireless adapters configured to use the IEEE 802.11g standard. The relevant properties of these adapters are:

- *Available transmit power levels*: 0, 8, 10, 12, 14, 16, 18, and 20 dBm;

- *Receive sensitivity thresholds*: −72 dBm (54 Mbps), −88 dBm (11 Mbps), −90 dBm (5.5 Mbps), −92 dBm (2 Mbps), and −95 dBm (1 Mbps).

Due to lack of space we are not able to show in detail our experimental results that explore the relationship between transmit power, operating rate and performance. A summary of our experiments in this context is given next, followed by our conclusions:

1. We fixed transmit power to 0 dBm, and made measurements for the next operating rates: 54, 11, 5.5, 2, and 1 Mbps;

2. We fixed the operating rate to 54 Mbps, and made measurements for the next transmit power settings: 0, 8, 14, 20 dBm. Then we fixed the operating rate to 11 Mbps, and repeated the previous series of measurements.

These measurements were carried out using iperf both on two main grid nodes located at about 1.5 m from each other, and on two sandbox nodes. The measured throughput had the expected value of around 25 Mbps at 54 Mbps operating rate even for the lowest transmit power, 0 dBm. Results of measurements for other operating rates were also conforming to expectations.

One issue we encountered in these basic experiments is that the first series of tests showed that at low transmit power settings (i.e., 0 and 8 dBm) the measured throughput on the main grid nodes was only around 2 Mbps. This was obviously wrong given the proximity of the nodes, and we searched for a cause. To the best of our knowledge, it is not possible to check the state of the noise generator on ORBIT, nor is it automatically turned off before an experiment. However, by turning the noise generator *off* manually we were able to obtain the expected results in throughput measurements even at low transmit power settings. We assume therefore that the initial incorrect results were caused by the noise generator having been turned *on* and left *on* by an ORBIT user that had employed the testbed before us.

One other issue we want to warn our readers about is that the ORBIT testbed is located in a building where wireless production networks are also in use. Therefore, in order to prevent interference with the production network, users should choose a channel that is sufficiently isolated for that of the production network. In all our experiments we used channel 1, given that the production network is said to operate on channel 6. Ideally, one may wish to have access to a wireless traffic analyzer in order to identify more reliably the available channels, but as far as we know such a tool is not available to ORBIT users.

4.2 Range Measurements

The basic measurements that we presented so far were aimed at evaluating the performance of two wireless nodes located in the vicinity of each other. However, one must also quantify how a node can communicate with other testbed

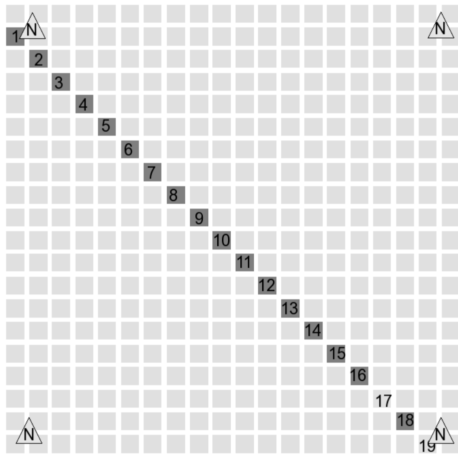

Figure 3: Active nodes for range measurements.

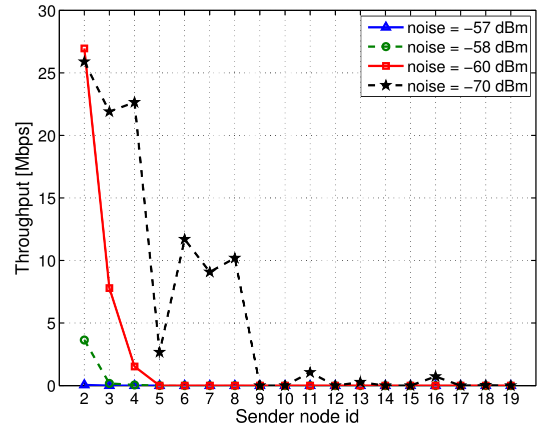

Figure 4: Throughput versus sender node for range measurement experiments.

nodes at remote locations, in other words ascertain its communication range.

Determining the communication range of a wireless node is important for two reasons:

- To decide the placement of the nodes so that certain experiments criteria are met. For example, one may wish to place nodes at the limits of their communication range in order to ensure multi-hop communication;
- To understand and explain the results of a certain experiment, for instance in conjunction with the use of analytical techniques.

For a testbed with a square shape, such as ORBIT, the simplest way to quantify the communication range of the nodes is to use active nodes placed on one of the diagonals of the square, as shown in Figure 3. The dark colored squares represent the nodes that were activated during the experiments, and the light colored squares represent the nodes that remained inactive; the triangles labeled 'N' represent the noise generator antennas. Out of the 19 diagonal nodes using Atheros AR5212 cards, two were not available at the time of the experiments, hence not used, namely nodes 17 and 19. Node 1 played the role of the traffic sink in our experiments, and the other active nodes played the roles of traffic generators. Note that only one active node was generating traffic at any one time; in particular, the respective iperf client was configured to generate 54 Mbps UDP traffic with 1024 byte packets for a 1 minute period.

Preliminary experiments have shown that two ORBIT nodes located at the most remote locations on the main grid (i.e., nodes 1 and 18 in our setup) can still communicate with each other in good conditions (about 19 Mbps throughput) even when fixing the operating rate to the highest available setting for 802.11g, 54 Mbps, and the transmit power to the lowest available setting for Atheros AR5212 adapters, 0 dBm.

An experiment in which several tenths and even several hundreds of nodes are all in the same communication area may not be sufficient to investigate many complex aspects related to wireless networks, such as multi-hop communication. As a consequence, we decided to use the noise injection system on ORBIT in order to artificially create multi-hop conditions, and measure communication range in these conditions as well.

The most significant experimental results in this context are displayed in Figure 4, which shows the throughput per sender node for several levels of the injected noise level (5 runs of 1 minute per level). Note that no attenuation was configured for the antennas, hence the resulting communication environment was symmetrical.

By analyzing Figure 4 we conclude the following. An injected noise level of −57 dBm or higher essentially prevents communication even between the closely located nodes 1 and 2, which are at a distance of approximately 1.5 m. An injected noise value of −58 dBm ensures that only one hop communication is possible, but in poor conditions (around 4 Mbps); nodes 1 and 3 are essentially not able to exchange packets. An injected noise level of −60 dBm ensures good communication between nodes 1 and 2 (around 27 Mbps), tolerable communication between nodes 1 and 3 (around 8 Mbps), and poor communication between nodes 1 and 4 (around 1 Mbps); traffic cannot be received from the nodes located any farther. For comparison purposes we also present the results obtained for noise level −70 dBm, in which case throughput becomes low or essentially zero for sender nodes 9 and higher.

We noticed that, for the −70 dBm noise level, throughput values exhibit a certain variation as the sender gets farther away from the receiver. In particular, throughput from node 5 drops to around 2.5 Mbps, significantly lower than that from the neighboring nodes. While we have yet to fully understand the cause of this behavior, this consistent drop in repeated experiments leads us to believe that it is determined by a characteristic of the testbed itself, not by a transient effect. A misalignment of the antennas, or an influence of the non-uniformity of the injected noise, are possible candidates for an explanation.

4.3 Discussion

The lessons we learned from the series of experiments presented in this section are the following:

1. Basic measurements in simple and well-understood se-

tups are essential for determining the fundamental properties of a testbed, and help identify any potential oddities that may be caused by testbed conditions that are not fully mastered;

2. Assessing testbed conditions before and while doing experiments is strongly recommended in order to uncover undesired interferences from various testbed systems (such as the noise generators on ORBIT) or external wireless networks.

In this context, previous measurement experience and other techniques such as modeling and simulation are critical for being able to determine the expected results of an experiment, so that they can be compared with the actual results obtained on the testbed.

Making testbed condition assessment possible in an easy manner would significantly increase the usability of ORBIT, and avoid surprises for unsuspecting users. In its absence, users are advised to actively turn off all the unused testbed systems, and even monitor testbed conditions by themselves, for instance by instrumenting one of the testbed nodes as a traffic sniffer.

5. COMPARATIVE ANALYSIS

After investigating the basic properties of ORBIT, we go even further with our examination of the challenges of using such a wireless network testbed. In this section we present a comparative analysis for an experiment that uses the OLSR protocol to build the network topology for a 50-node ad hoc network. In addition to ORBIT, we used the QOMB wireless network emulation testbed and the QualNet network simulator for this analysis. Note that a comparison of throughput results for several single-hop and multi-hop experiments done for these three systems has been previously presented in [3].

5.1 QOMB

The QOMB testbed was created by integrating the wireless network emulator QOMET with the large-scale network experiment environment StarBED as discussed in [2].

QOMET is a wireless network emulator that employs a two-stage approach to convert a user-defined scenario representation into network parameters (bandwidth, packet loss, delay and jitter) [1]. These parameters are then used to configure a link-level emulator such as Dummynet [9]. Thus, QOMET reproduces in a wired network the communication conditions of the emulated wireless network. The use of StarBED [5] as an infrastructure, with more than 1000 PCs available for experiments, makes possible large-scale experiments with QOMET.

The integration of QOMET and StarBED resulted in the QOMB testbed. This testbed makes use of the experiment-support software provided on StarBED, called SpringOS, to simplify the task of making experiments, and adds several customized programs to enable the wireless network emulation. The following are the most important features of QOMB as used in this paper:

- Support for IEEE 802.11a/b/g network emulation;
- Support for defining the position of the nodes and their properties (e.g., transmit power, receive sensitivity);
- Support for defining the wireless communication environment in terms of electromagnetic wave attenuation and a uniform noise floor level.

5.2 QualNet

QualNet is a widely-used commercial network simulator from Scalable Network Technologies, Inc. [10]. In this paper we used the IEEE 802.11g model of QualNet to simulate the wireless nodes on ORBIT. We also used one of the OLSR protocol implementations available in QualNet, called OLSRv2-Niigata, which we consider to be close to current OLSR implementations being effectively used in real networks. Although QualNet does not support setting a noise floor level, we reproduced its effect by an equivalent decrease of the transmit power compared to the case of QOMB.

5.3 OLSR Experiments

The setup for the OLSR experiments that we present here used a total of 50 nodes selected in a uniform manner from the 400 nodes of the ORBIT main grid. The selection was made with the constraint that the nodes use Atheros AR5212 adapters, so that there are no hardware differences between them. The position of the active nodes in our experiments is represented by circles in Figure 5.

Given the measurements presented so far, we used the IEEE 802.11g standard and fixed the operating rate to 54 Mbps; node transmit power was configured to 0 dBm. Artificial noise was injected for most experiments, with equal levels for each of the four antennas; however, noise injection was turned off in some of the tests so as to have a baseline for comparison.

Each of the selected ORBIT nodes executed the OLSR protocol implementation olsrd-0.5.5 [7]. The routing metric used was ETX [4]. One of the nodes used the plugin named "txtinfo" to make it possible to extract in real time the topology of the nodes while the experiment was running. Each experiment lasted for 3 minutes; the topology data presented in this paper was obtained at 2 minutes after the start of the experiment, which we determined as being sufficient for the OLSR topology to settle.

A similar experiment was performed on QOMB by creating a scenario with nodes placed in a virtual space at the same locations at which they were on ORBIT. The operating rate and transmit power were configured to the same values as above. We assumed free-space propagation in the emulation setup. The noise floor was set *uniformly* for the virtual environment; this is different from the non-uniform noise levels that are created on ORBIT, which cannot be currently reproduced on QOMB. The same OLSR implementation was executed on the participating StarBED hosts, and topology information was extracted using the same plugin.

In QualNet too, node position was reproduced, and similar settings with QOMB were used. The unavailable noise floor setting was reproduced through lower transmit power levels. Throughput-based calibration with respect to QOMB was needed at this point because of minor differences in how adapter settings are implemented in QOMB and QualNet. As stated before, the OLSR version used for QualNet experiments was OLSRv2-Niigata.

For illustration purposes we show in Figure 5 a comparison between the OLSR topologies created on ORBIT and QOMB in one of our experiments. The results are obtained for −60 dBm non-uniform noise injection on ORBIT, and

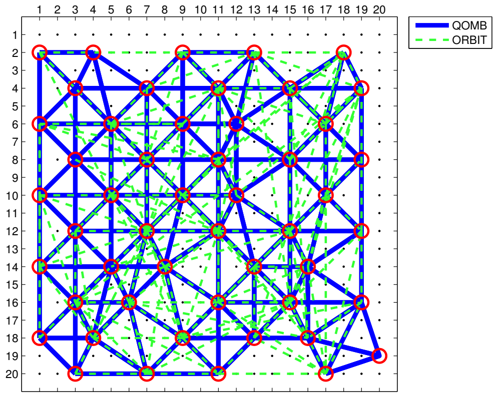

Figure 5: OLSR topology for 50 node experiment: ORBIT versus QOMB.

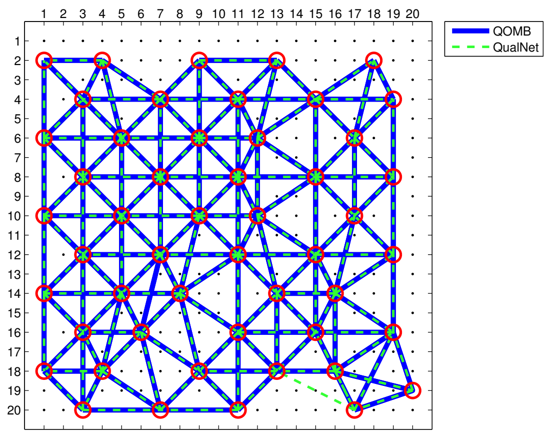

Figure 6: OLSR topology for 50 node experiment: QualNet versus QOMB.

−80 dBm uniform noise level setting for QOMB. It is obvious that the topologies obtained on ORBIT and QOMB differ somewhat, mainly in the central regions of the grid, where noise level differences between the two environments are more significant (see the discussion in Section 2.1).

Figure 6 compares the results obtained on QOMB in the conditions presented above (−80 dBm uniform noise level setting), and those of the equivalent experiment done through simulation on QualNet. One can determine even visually that the match is remarkable, as only one different link appears in lower right corner of the figure.

A quantitative evaluation of several of our experimental results is shown in Table 2. For each experiment we show the number of links in each environment, and in parentheses the percentage of common links to all environments. The ORBIT noise level settings were −60 dBm, −70 dBm, and "no noise", for experiments A, B, and C, respectively. The noise level settings for QOMB and QualNet were −80 dBm for experiment A, and −83 dBm for experiments B and C.

Table 2: OLSR topology statistics: link count for each environment, and percent of links that are common to all environments

Exp.	ORBIT	QOMB	QualNet
A	169 (52%)	152 (58%)	153 (58%)
B	234 (60%)	247 (56%)	239 (55%)
C	318 (53%)	247 (68%)	239 (67%)

The results show that the match between the three environments (i.e., the percent of common links) lies between 50% and 70%, hence the results can be said to be broadly similar.

The comparison of our experimental results (including results not presented here) indicate that it is possible to reproduce to a certain extent the ORBIT settings and results on other experiment platforms, such as emulators and simulators. However, the related modeling has to be done sufficiently accurately if one wishes to obtain similar results. In our case, the most difficult aspect was related to the noise injection system. On the other hand, the results we obtained through emulation and simulation match well, mainly because these two platforms use similar assumptions regarding noise, despite the other differences between them.

5.4 Discussion

This section illustrated how ORBIT can be practically used for a relatively large-scale experiment. The challenges that we have encountered in this context are:

1. Difficulty in activating wireless nodes at desired locations;

2. Issues with understanding experiment conditions and the relationship with an equivalent real setup when using the noise injection system.

The problem we had when trying to activate 50 ORBIT nodes at preselected locations was that about 20% of the initially selected nodes were unavailable for experiments (i.e., could not be initialized correctly). This meant that the unavailable nodes had to be reallocated in an iterative fashion, since each time we encountered a number of unavailable nodes. This made the node selection process rather tedious, and made us wish that ORBIT nodes had a higher availability.

As for the noise injection system, we believe that users need to spend some time modeling it so as to be able to understand what would be the real-world setup that corresponds to a certain level of the injected noise. We believe this to be mandatory should one wish to generalize the results obtained on ORBIT, and to draw conclusions that have a wider scope than ORBIT itself.

6. CONCLUSIONS

In this paper we have discussed several challenges of using wireless network testbeds, and presented a number of lessons learned through the use of the ORBIT testbed, that we summarize below as advices for all wireless network testbed users:

1. Evaluate the measurement tools you intend to use, and be aware of the potential overhead of the measurement framework (we discussed possible pitfalls in Section 3);

2. Assess the properties of the testbed you plan to use, as well as the conditions before and during experiments (we showed consequences of not doing so in Section 4);

3. Try to understand the experimental results, and their correspondence to equivalent field trials, possibly through complementary techniques, such as analytical modeling, simulation, and emulation (see our OLSR study in Section 5).

We conclude that ORBIT is a versatile wireless network testbed, especially in the context of testing algorithm and protocol implementations on real hardware and at large scale. Improvements are however desired in terms of the performance of the measurement framework, the assessment of experimental conditions, and the availability of the wireless nodes.

Our experience with ORBIT also leads us to formulate a more general caveat regarding wireless testbed usage, namely that the results obtained on such a testbed may be too particular for being of general significance, since influenced by a series of local conditions. Hence, we believe that testbeds are not necessarily a *preferred* alternative to simulation, but rather a *complementary* platform for network experiments. Thus, we suggest that testbed experiments should ideally be supplemented by simulation and/or emulation experiments, which – if conducted correctly and by using the appropriate tools – may very well provide the missing degree of abstraction required to make general predictions about a system's performance. This idea is supported by the general match that we have observed between the results obtained on the three experiment platforms that we have used.

As future work, we envisage creating a more accurate representation of ORBIT on the wireless network emulation testbed QOMB, which opens the path to using it as an alternative "virtual" ORBIT that would provide a better control of the experimental conditions, and an easier to define correspondence to target field trials.

7. ACKNOWLEDGMENTS

We would like to thank the members of the ORBIT and OMF development teams, in particular to Ivan Seskar and Thierry Rakotoarivelo, for their advices on how to use the ORBIT testbed and its mechanisms.

8. REFERENCES

[1] R. Beuran, J. Nakata, T. Okada, L. T. Nguyen, Y. Tan, and Y. Shinoda. A multi-purpose wireless network emulator: QOMET. In *Proc. of IEEE International Conference on Advanced Information Networking and Applications (AINA 2008) Workshops, FINA 2008 symposium*, pages 223–228, Okinawa, Japan, March 2008.

[2] R. Beuran, L. T. Nguyen, T. Miyachi, J. Nakata, K. Chinen, Y. Tan, and Y. Shinoda. QOMB: A wireless network emulation testbed. In *Proc. of IEEE Global Communications Conference (GLOBECOM 2009)*, Honolulu, Hawaii, USA, November-December 2009.

[3] R. Beuran, L. T. Nguyen, and Y. Shinoda. QOMB wireless network emulation testbed: Evaluation and case study. In *Proc. of ACM International Workshop on Wireless Network Testbeds, Experimental evaluation and Characterization (WiNTECH 2010)*, pages 1–8, Chicago, Illinois, USA, September 2010.

[4] D. S. J. De Couto, D. Aguayo, J. Bicket, and R. Morris. A high-throughput path metric for multi-hop wireless routing. In *Proc. of ACM International Conference on Mobile Computing and Networking (MobiCom 2003)*, San Diego, California, September 2003.

[5] T. Miyachi, K. Chinen, and Y. Shinoda. StarBED and SpringOS: Large-scale general purpose network testbed and supporting software. In *Proc. of International Conference on Performance Evaluation Methodologies and Tools (VALUETOOLS 2006)*, Pisa, Italy, October 2006.

[6] NICTA. OMF website. http://mytestbed.net/.

[7] olsr.org. Olsrd: An adhoc wireless mesh routing daemon. http://www.olsr.org/.

[8] D. Raychaudhuri, I. Seskar, M. Ott, S. Ganu, K. Ramachandran, H. Kremo, R. Siracusa, H. Liu, and M. Singh. Overview of the ORBIT radio grid testbed for evaluation of next-generation wireless network protocols. In *Proc. of IEEE WCNC 2005*, New Orleans, Louisiana, USA, March 2005.

[9] L. Rizzo. Dummynet: A simple approach to the evaluation of network protocols. *ACM SIGCOMM Computer Communication Review*, 27(1):31–41, 1997.

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

[11] SourceForge. Iperf homepage. http://sourceforge.net/projects/iperf/.

[12] WINLAB. ORBIT website. http://www.orbit-lab.org/.