QOMB: A Wireless Network Emulation Testbed

Razvan Beuran[1,2], Lan Tien Nguyen[2], Toshiyuki Miyachi[1,2], Junya Nakata[1,2],
Ken-ichi Chinen[2,1], Yasuo Tan[2,1], Yoichi Shinoda[2,1]

[1] National Institute of Information and Communications Technology,
Hokuriku Research Center, Ishikawa, Japan
[2] Japan Advanced Institute of Science and Technology, Ishikawa, Japan

***Abstract*—In this paper we present QOMB, a testbed we designed and implemented for the evaluation of wireless network systems, protocols and applications. The testbed uses the wireless network emulation set of tools QOMET so as to reproduce in a wired network, in real time, the wireless network conditions corresponding to a given scenario. In this context QOMET also provides support for features such as realistic virtual 3D environments, and node mobility generation. The infrastructure of QOMB is StarBED, the large-scale network experiment testbed at the National Institute of Information and Communications Technology, Hokuriku Research Center, in Ishikawa, Japan. The multi-hop wireless network emulation experimental results related to OLSR performance analysis in mesh networks and MANETs illustrate the main features and the usability of QOMB.**

Keywords-emulation; real-time; testbed; wireless networks

I. INTRODUCTION

The development of wireless network protocols and applications implies testing the developed systems in order to verify their functionality. Furthermore, before deployment in mission-critical environments, the performance characteristics of existing network protocols and applications must also be assessed in various specific circumstances.

Currently, researchers rely mainly on simulators, such as the widely-used ns-2 [1], for evaluating their new algorithms and techniques, and for comparing them with those developed by others. However, since simulators use logical models of applications and protocols, it is not possible to accurately assess how the system under test would behave in a real environment, in which it will interact in real time with real network applications and protocol implementations. Often, the simulated wireless network conditions are far from reality as well, by setting fixed communication ranges between nodes, or by using random mobility models.

Real-world tests are the method of choice before the deployment of a network system, and play an important role in assessing system performance. An issue is that in a real-world wireless network factors such as interference may influence the experiment. Moreover, due to time and financial constraints, real-world tests cannot possibly explore exhaustively the entirety of experimental conditions that could be met during the life time of a system. This is particularly true for systems that are intended to be used in special environments, such as disaster situations, and other emergency conditions.

Network emulation is an experimental technique intended to bridge the gap between simulation experiments and real-world tests. In Table I we compare the three experiment techniques from the point of view of the execution (real time or not), the level of control over the experiment, the range of experimental conditions that can be reproduced, the realism of the results, the experimentation cost, and the ease of use (the comparison is only qualitative; we intend to make a quantitative comparison of the three experiment techniques in a future paper). As it can be seen, emulation is done usually in real time, allows controlling the experiment and can be automated, makes it possible to experiment in a wide range of conditions, and results have at least a medium degree of realism. Emulation experiments are more difficult to set up than simulation, because of the real components used, but changing conditions is considerably easier than for the real tests, therefore the total cost and ease of use are medium.

TABLE I. COMPARISON OF EXPERIMENT TECHNIQUES

	Simulation	**Real world**	**Emulation**
Real-time execution	NO	YES	USUALLY
Control level	HIGH	LOW	HIGH
Condition range	LARGE	SMALL	LARGE
Result realism	LOW	HIGH	MEDIUM
Experimentation cost	LOW	HIGH	MEDIUM
Ease of use	HIGH	LOW	MEDIUM

In this paper we present QOMB, a wireless network emulation testbed based on the versatile wireless network emulation set of tools QOMET [2], and using as infrastructure StarBED, the large-scale network experiment environment at the National Institute of Information and Communications Technology, Hokuriku Research Center, in Ishikawa, Japan [3]. The main contributions of this paper are:

1. Present the redesigned QOMET that allows QOMB to support multi-hop routing protocols such as OLSR, and dynamic communication condition computation, depending on traffic conditions (Section II.A);
2. Illustrate how emulation experiments on QOMB can be used to assess real implementations of routing protocols (Section III).

The paper is organized as follows. Section II introduces QOMB and its components, with emphasis on QOMET redesign. Section III presents several experimental results that demonstrate the use of QOMB. Section IV discusses related work, and is followed by conclusions and references.

978-1-4244-4148-8/09/$25.00 ©2009
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2009 proceedings.

II. QOMB TESTBED

Since 2006 we have started developing a set of tools related to wireless network emulation for which we use the name QOMET (Quality Observation and Mobility Experiment Tools). Although the focus was initially on WLAN emulation, QOMET has grown into an extensive collection of tools for wireless network emulation, including additional wireless network technologies such as active RFID tag or ZigBee, 3D scenario representation, realistic node mobility, etc.

Most of the experiments performed so far with QOMET used as infrastructure StarBED, the large-scale network experiment environment in Ishikawa, Japan. Although QOMET can be used in principle on any operating system that allows network degradation control, so far we have mainly used the FreeBSD operating system and *dummynet* [4] to control network conditions.

Flexibility, usability and scalability were gradually increased as we used the continuously evolving QOMET on StarBED for several purposes, such as: (i) emulation of networked robots for motion planning algorithm research [5] – up to 400 virtual robots running on 100 PCs, and using a QOMET library to compute in real time the communication conditions with each other; (ii) active RFID tag emulation experiments for pedestrian localization applications [6] – up to 133 active RFID tags, for which both the active tag processor and the wireless communication were emulated. Such large-scale experiments were easier to configure and run since we employed the two experiment-support software tools of StarBED, SpringOS and RUNE (see Section II.B).

While RUNE and SpringOS made it possible to run large-scale experiments on StarBED, and QOMET could recreate wireless link conditions in a wired network environment, they were not enough for a true wireless network emulation testbed. The main issues were that no standard wireless network routing protocol was usable on StarBED (thus QOMET could only emulate point-to-point communication), and communication conditions were computed independently of the traffic sent by experiment nodes. The two additional features required to create a genuine multi-hop wireless communication emulation testbed were:

- Support for the execution of a multi-hop routing protocol such as OLSR;
- Support for the dynamic computation of the communication conditions between the emulated nodes, depending on traffic conditions.

Adding these features required a redesign of QOMET architecture as it will be mentioned in Section II.A. The result of this redesign is an easy to use testbed for multi-hop wireless network emulation experiments that we call QOMB (QOMet on starBed). QOMB also benefits of other recent advances in QOMET, such as realistic environments based on GIS (Geographic Information System) map data. All these features allowed us to make for the first time multi-hop wireless network emulation experiments in the context of OLSR performance analysis in WLAN mesh networks and MANET (see Section III).

A. QOMET Wireless Network Emulation

QOMET uses a scenario representation to compute the point-to-point network quality degradation (ΔQ) corresponding to the real-world events. This computation is implemented by QOMET as a library named *deltaQ*. For point-to-point communication emulation the ΔQ description is applied in real time by the library called *wireconf* to configure the FreeBSD kernel module *dummynet*. This recreates the wireless conditions of the user-defined scenario on the wired network testbed during the effective emulation process.

In order to add support for multi-hop routing, and to increase experiment realism by taking into account traffic conditions, we had to improve this simple approach, as depicted in Fig. 1. This improvement was achieved in QOMB by introducing several processing steps that are done in real time, and that are implemented as two new modules of the *wireconf* library (*routing support* and *statistics processing*), and by enhancing the *dummynet configuration module* so as to adjust on-the-fly the ΔQ description before effectively applying the configuration to *dummynet*.

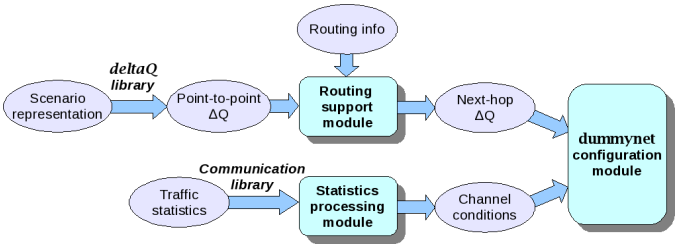

Figure 1. Architecture of QOMET wireless network emulation.

We review next the QOMET architecture as used in QOMB (see Fig. 1). A QOMET scenario consists of an XML-based description of the wireless network nodes, topology elements, motion patterns, and communication environment properties. For wireless network nodes one can specify the wireless adapter technology and properties, such as transmitted power or antenna gain. The virtual environment topology and objects are also defined by the user; this information and node positions at each moment in time are used to compute the communication conditions. Node mobility is supported as follows: linear motion, circular motion, random-walk motion, and behavioral motion (a motion model that realistically determines the motion trajectory in a certain topology given the starting position and destination of the node).

The *deltaQ* library is used to compute the point-to-point ΔQ parameters that correspond to the user scenario. The current *deltaQ* library implementation (as of QOMET v1.6) enables emulation of IEEE 802.11a/b/g WLANs using a model based on adapter receive-sensitivity thresholds that are available from most manufacturers, and that includes effects of background noise. For propagation we use the log-distance path loss shadowing model. Support for 802.11g stations operating in compatibility mode (i.e., when in the presence of 802.11b stations) is also present. Our model can statically take into account interference between neighboring nodes if continuous transmission is assumed, but QOMB uses dynamic adjustment based on real (measured) traffic conditions instead. In addition to WLAN, QOMET also supports emulation of active RFID

978-1-4244-4148-8/09/$25.00 ©2009
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2009 proceedings.

tag communication, such as the AYID32305 tags from Ymatic Corporation, and also has basic support for IEEE 802.15.4 communication emulation.

The ΔQ parameters and routing information retrieved from the FreeBSD kernel are used by the *routing support module* of the *wireconf* library to determine the ΔQ parameters that should be used for the communication from a given node to the next hop, as follows. Since in multi-hop routing the next hop is not necessary the destination, we first identify the next-hop information for a certain destination. The corresponding next-hop ΔQ description will be used in the next stage to configure the outgoing *dummynet* queue of the current node. OLSR and other applications also use broadcast traffic. For broadcast traffic, quality degradation must be applied when packets arrive at destination. In this case the incoming *dummynet* queue will be configured by using the ΔQ description data corresponding to the broadcast source and the current node.

Note that the ΔQ parameters mentioned above are computed by QOMET before the emulation experiment, and correspond to contention-free conditions. In order to make it possible to take into account contention, each node gathers real traffic statistics from *dummynet* queues. These statistics are multicasted through the management network of StarBED. The *statistics processing module* of the *wireconf* library that is running on each node receives these statistics, and uses them to estimate channel utilization conditions in the virtual wireless environment.

Channel utilization together with next-hop ΔQ information are used by the *dummynet* configuration model of the *wireconf* library to: (i) adjust next-hop ΔQ so as to take into account contention by using the *deltaQ* library; (ii) configure the wired-network emulator *dummynet* with the adjusted next-hop ΔQ for outgoing queues, and with broadcast-related ΔQ for incoming queues. The *wireconf* library does this reconfiguration at regular time intervals so as to reproduce on QOMB the varying conditions of a wireless network. We also provide a timer library to ensure that the *dummynet* configuration is changed at accurate time intervals during the emulation process.

B. StarBED

StarBED is the experiment execution infrastructure of QOMB. The core of StarBED consists of a cluster of around 1000 standard PCs, the *experiment nodes*, which have redundant full connectivity by means of a switch cluster (see Fig. 2). Specific switch configurations are used to produce a target network experiment topology on StarBED by making use of VLANs. Virtualization techniques such as VMware can be employed to increase the number of logical experiment nodes. In addition to the core *experiment network*, there is a dedicated *management network* that controls and monitors node and switch activity. Nodes can be loaded with the appropriate software, controlled, and monitored by using the management network, thus not affecting the experiments. The experiment switch cluster can also be used to monitor network activity, or to connect the core network to external networks.

By using QOMET on the wired-network testbed StarBED, the resulting testbed, named QOMB, becomes an appropriate platform for wireless network emulation experiments. In addition to the emulation features inherited from QOMET, QOMB's advantages deriving from the use of StarBED are: (i) use of real nodes in a large-scale setup that makes it possible to realistically emulate large network environments; (ii) flexibility of the test environment that allows using different network configurations; (iii) a powerful management system that enables easy control, quick reconfiguration, and concurrent use of the facility for independent experiments.

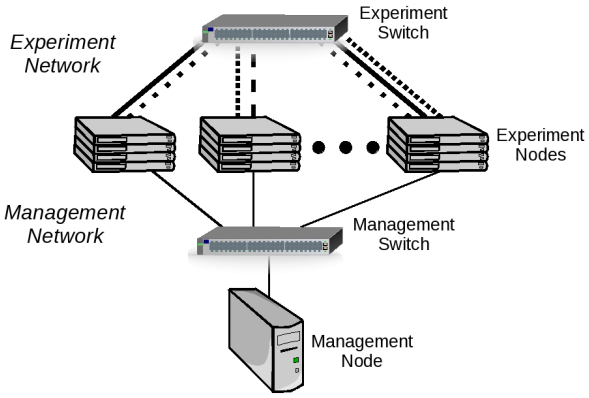

Figure 2. StarBED topology.

StarBED provides an experiment-support system named SpringOS that assists users in making experiments [3]. When using SpringOS one must describe the experiment by a configuration file, and SpringOS will run the experiment following that configuration file. The functions of SpringOS are: (i) assign experiment nodes; (ii) upload the appropriate operating system and software to the nodes; (iii) configure switches to build the target topology; (iv) drive the execution of the experiment scenario. StarBED can be manipulated by several users simultaneously, and permits concurrent resource usage, such as node assignment or switch configuration. Access restriction and mediation mechanisms for sharing and managing those resources are built into SpringOS by design. SpringOS functionality is split into various software modules, such as experiment coordinator (*kuroyuri*), experiment resource manager (*erm*), node initializer (*ni*), Wake on LAN Agent (*wol-agent*), etc.

For ubiquitous network experiments, QOMB required a finer-grain control than the one provided by SpringOS; for this purpose we implemented a specific experiment-support software tool. RUNE [7] (Real-time Ubiquitous Network Emulation environment) was designed from start to support emulation of large ubiquitous networks, having features such as: (i) emulation of the surrounding environments; (ii) support for real-time concurrent execution of numerous nodes; (iii) provision of multi-level emulation layers, etc. The basic element of the logical structure of a RUNE-driven emulation is the *space*. A space is an entity that behaves as one of the emulated devices, depending on its function. Spaces can emulate: (i) nodes, i.e., physical devices, such as WLAN devices; (ii) environments, such as the thermal field in a room; (iii) networks, i.e., a physical communication channel, such as a WLAN connection. Spaces are connected with each other by elements called *conduits*. The role of conduits is to create an abstract error-free communication path between two spaces in a manner that is transparent to the user.

978-1-4244-4148-8/09/$25.00 ©2009
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2009 proceedings.

III. EXPERIMENTAL RESULTS

In what follows we shall present several experimental results that illustrate the new features of QOMB based on QOMET v1.6, such as support for the OLSR protocol and realistic environments. The version of OLSR that we use is olsrd-0.5.5 [8]. Although we are currently employing QOMB to evaluate routing metrics that we develop to improve OLSR performance, the results presented here use the ETX (eXpected Transmission Count) metric, which is currently the *de facto* standard for OLSR. For generating traffic we use iperf v2.0.1 [9], and for estimating round-trip time (RTT) we use ping. These experiments in this section are intended to show: (i) how emulation can be used to assess real implementations of multi-hop routing protocols (Section III.A), and (ii) the influence of motion patterns on application performance in multi-hop scenarios (Section III.B).

A. *Mesh network scenario*

One of the typical usages of OLSR is in mesh networks, such as the topology presented in Fig. 3. We considered a topology with a regular placement of the nodes, as one may expect in a large public area, such as an airport or railway station. The mesh routers, numbered from 1 to 12, would receive traffic from the WLAN users, and forward it to the Internet gateway (node 0). The distance in the virtual environment between nodes, both in horizontal and vertical directions, is 40 m, and the transmission power was set to -20 dBm, so as to avoid excessive interference. The resulting range, assuming free space propagation, is of about 90 m. This ensures that a node can communicate with at most its second-order neighbors, and allows studying multi-hop routing effects.

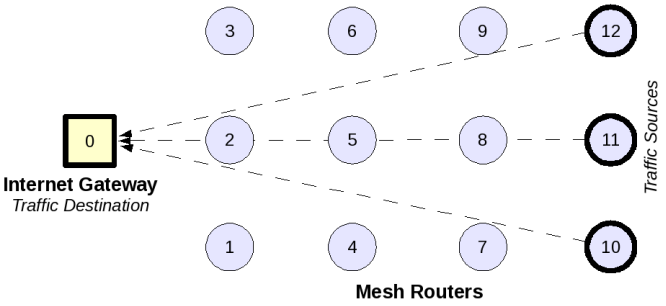

Figure 3. Mesh network topology.

To illustrate one possible use of our emulator, we show here the results of a test in which 3 routers (#10, #11, and #12) generate traffic towards the destination #0, whereas the other routers only forward it. While simultaneously varying the offered load (sending rate) of traffic streams from source nodes, we measured performance parameters, such as throughput and RTT. The experiment consists of 30 s of no traffic, allowing OLSR to build the routes, followed by 120 s of CBR traffic from each source. The average performance metrics presented are computed for the interval 10 to 110 s of the active period.

In Figs. 4 and 5 we show the average throughput and RTT for the three traffic sources. It can be seen that for loads of 256 kbps or more the throughput is limited, and RTT increases significantly. This is because the traffic produced by nodes #10, #11 and #12 interferes with itself as it is being forwarded on a hop-by-hop basis to destination, and saturation is eventually achieved as offered load increases.

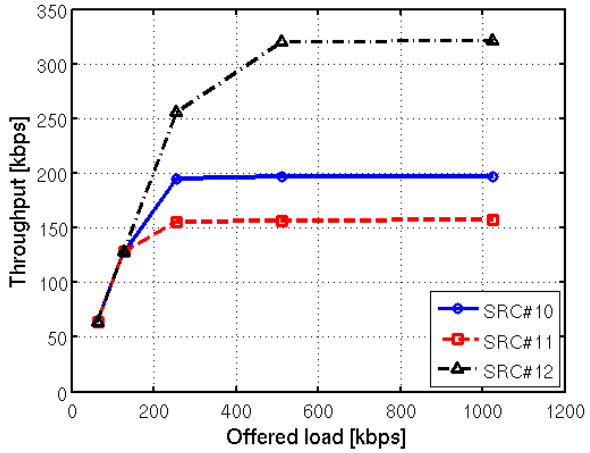

Figure 4. Average throughput per traffic source versus offered load for the mesh network scenario.

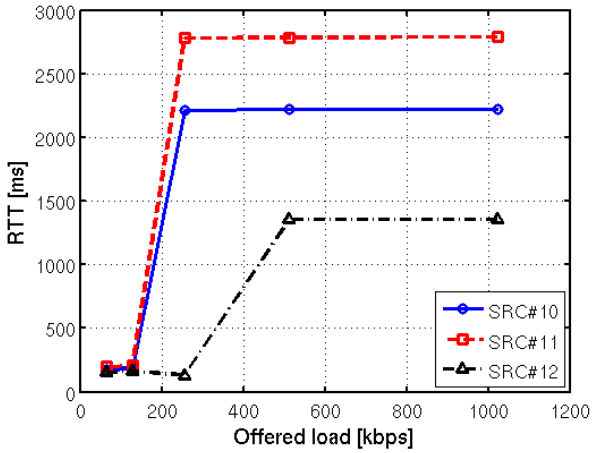

Figure 5. Average RTT per traffic source versus offered load for the mesh network scenario.

One may have expected that, given the symmetry of the setup, the three data streams will have similar performance characteristics. Nevertheless, Figs. 4 and 5 emphasize the fact that performance doesn't only depend on physical position, but also on how the routing algorithm decides the path to destination. In our experiment OLSR preferred central nodes for forwarding packets, which leads to the low performance of node #11, situated on the symmetry axis of the setup, when compared to the other two nodes. However, nodes #10 and #12 too have rather different performance characteristics. This is because the routes created by OLSR provided better conditions for node #12 compared to node #10, as the path from node #12 to destination was not as congested as those of the other nodes. This demonstrates that the ETX metric is unable to correctly estimate congestion in some circumstances. We are currently working on eliminating such unfairness from OLSR, and on further improving performance by taking into account operating rate and wireless channel congestion indicators when making routing decisions.

978-1-4244-4148-8/09/$25.00 ©2009
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2009 proceedings.

B. *MANET scenario*

One of our research interests is related to the use of ad-hoc networks in emergency conditions. The performance of applications such as VoIP or video streaming in such circumstances needs to be assessed in advance. To exemplify such research we have created the scenario depicted in Fig. 6. In a virtual environment representing a realistic street topology based on real map data for Kawasaki, Japan, we consider a base location, denoted by GW0, from which 12 emergency workers, denoted by EW#01 to EW#12, depart for a search & rescue operation. They are heading to the destinations D01 to D12, respectively; we use the behavioral model in QOMET to generate their trajectory. While moving, the emergency workers send a stream of data (such as voice reporting) to the gateway. This stream is reproduced in our experiment by a 64 kbps CBR traffic stream. Transmission power was set to 10 dBm, and the environment was considered to have an attenuation of 3.32, typical for outdoor environments [10].

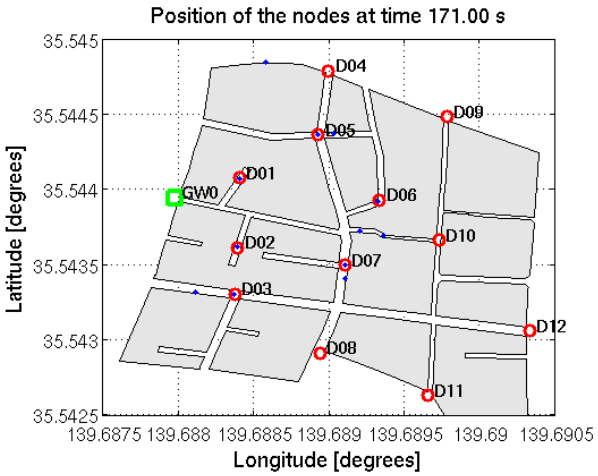

Figure 6. MANET topology: a search & rescue operation scenario.

In Figs. 7 and 8 we present the average throughput and RTT results for several of the wireless nodes. To show the range of performance that one can expect in such a scenario, we selected nodes that will reach destinations located at small, medium and long distances compared to the gateway, as follows: EW#03 heading for D03, EW#04 heading for D04, and EW#12 heading for D12.

This experiment starts with a 30 s period without traffic that gives the OLSR daemon time to build routes between nodes. Then data is sent for 300 s (the active period), which is also a sufficient time for all the nodes to reach their destinations. We show in Figs. 7 and 8 the results in the interval from 10 to 290 s within the active period.

Throughput values in Fig. 7 are comparable between nodes, although one can notice gaps in communication both for EW#04 and EW#12; on the other hand overall packet loss is under 2% for EW#03. Delay, as shown by RTT (Fig. 8), provides more insight into what happens with the communication conditions. For example, the delay variation for EW#03 around 110 s is caused by operating rate changes. In addition, since EW#04 exits the range of GW0 at around 175 s, an interval when no traffic can be sent follows until EW#04 is able to begin multi-hop communication with the gateway. One can also notice from Fig. 8 that it generally takes around 15 s for OLSR to re-establish a route connecting the nodes to GW0, as shown by the gaps between 175 and 190 s for EW#04, and between 140 and 155 s for EW#12.

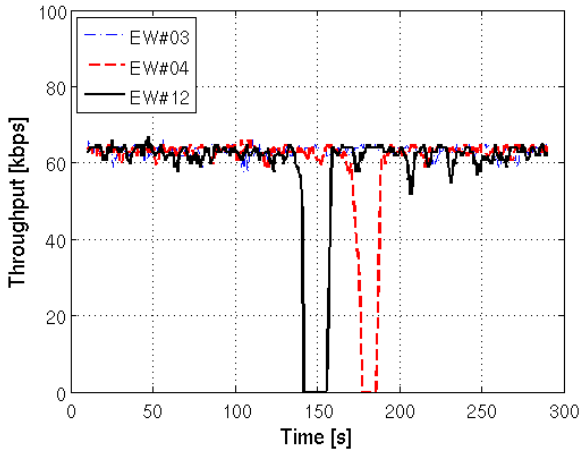

Figure 7. Average throughput per traffic source versus time for selected nodes in the MANET scenario.

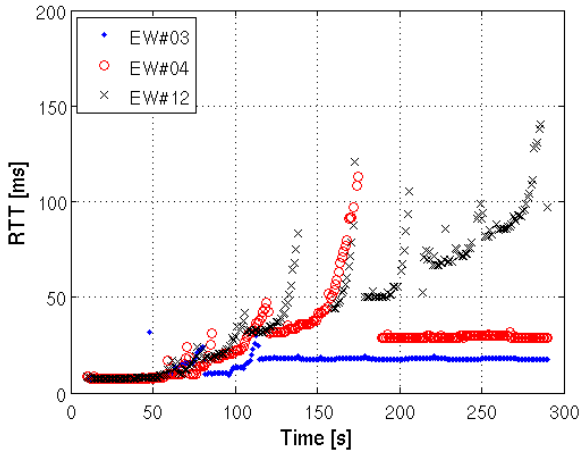

Figure 8. Average RTT per traffic source versus time for selected nodes in the MANET scenario.

IV. Related Work

Emulation has gained ground in recent years, which is proven by the fact that most currently used simulators such as the previously mentioned ns-2, or QualNet [11] provide emulation features. Nevertheless, given the discrete-event approach of simulation tools, the size and complexity of the networks, and the traffic volume that can be used in such emulation experiments is limited.

One approach to emulation is to use real wireless equipment in a controlled environment (although, in our view, this only represents a "facilitated" real-world experiment platform). This approach is that of, for instance, the public wireless testbed Emulab [12]. However, since such emulators use real wireless equipment, condition control is difficult, and

978-1-4244-4148-8/09/$25.00 ©2009
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2009 proceedings.

undesired interferences can perturb experimental results. If mobility is also to be studied, controlled movement of wireless nodes has to be orchestrated; this is a cumbersome task with a high management overhead (and impossible if using a fixed testbed such as Emulab). Some ways to tackle this problem are the dense-grid approach of ORBIT [13], or the more realistic robot-based Mobile Emulab [14].

In our approach, the wireless network is emulated by reproducing the communication conditions in a wired network. Examples of emulators that employ a similar methodology are Seawind [15], and EMPOWER [16]. Such tools make it possible to introduce network layer effects, such as bandwidth limitation, delay, and packet loss. However, these effects are directly provided by the user who configures the emulator; thus, the relationship between these effects and reality becomes a user task, and it is possibly not accurately defined. Such oversimplified designs running on one or even a small number of PCs fail to recreate realistic communication conditions, and may not scale to larger networks and larger amounts of traffic. Another wireless network testbed, SWOON [17], tackles scalability by using two experimental nodes to emulate one single wireless node, but the communication condition reproduction realism is still low.

There are also attempts to develop emulators that recreate network conditions that correspond to real events, for instance W-NINE [18], the wireless network emulation extension of SDNE [19], or CORE [20]. These implementations start from a description of node positions and movements. However, the accuracy of the conditions they recreate is relatively low because of the simplicity of the models used. W-NINE, for example, uses tables to associate IP throughput to received signal levels, loss probability is considered to be either 0 or 1, etc. CORE simply uses communication range to determine network conditions, in a manner similar to ns-2. A more realistic design is that of TWINE [21], which uses WLAN computer models for real-time experiments. TWINE must nevertheless combine emulation with simulation in order to achieve scalability.

V. CONCLUSION

In this paper we presented QOMB, a wireless network emulation testbed based on the wireless network emulation collection of tools QOMET, and using as infrastructure StarBED, a large-scale experiment environment in Ishikawa, Japan. QOMET was redesigned for the purpose of creating QOMB so as to support realistic multi-hop communication and routing. The rich set of additional features of QOMET, such as several wireless network technologies, realistic 3D map-based topologies, and mobility, make it possible to carry out a wide range of emulation experiments.

To illustrate the flexibility and usability of QOMB, we discussed several experiments that we carried out in the context of OLSR performance analysis. Our results demonstrate that by using QOMB it is possible to make global performance estimates, as well as gain a deep insight into various aspects of the routing protocol implementation through network scenarios that are difficult to recreate in real-world tests.

Our future work will focus on using QOMB for analyzing the performance of the new routing metrics that we are currently developing for OLSR, so as to improve its robustness and to ameliorate application performance in congested wireless networks.

REFERENCES

[1] University of Sothern California, Information Sciences Institute, "Network Simulator - ns-2", http://isi.edu/nsnam/ns/.

[2] R. Beuran, J. Nakata, T. Okada, L. T. Nguyen, Y. Tan, Y. Shinoda, "A Multi-purpose Wireless Network Emulator: QOMET", *Proc. of IEEE FINA2008*, Okinawa, Japan, March 25-28, 2008, pp. 223-228.

[3] T. Miyachi, K. Chinen, Y. Shinoda, "StarBED and SpringOS: Large-scale General Purpose Network Testbed and Supporting Software", *Proc. of Valuetools 2006*, Pisa, Italy, Oct. 2006.

[4] L. Rizzo, "Dummynet FreeBSD network emulator", http://info.iet.unipi.it/~luigi/ip_dummynet/.

[5] R. Beuran, J. Nakata, T. Okada, Y. Tan, Y. Shinoda, "Real-time emulation of networked robot systems", *Proc. of SIMUTools 2008*, Marseille, France, March 3-7, 2008.

[6] R. Beuran, J. Nakata, T. Okada, T. Kawakami, K. Chinen, Y. Tan, Y. Shinoda, "Emulation System for Active Tag Applications", *Proc. of ISSNIP 2008*, Sydney, Australia, December 15-18, 2008, pp. 1-6.

[7] J. Nakata, T. Miyachi, R. Beuran, K. Chinen, S. Uda, K. Masui, Y. Tan, Y. Shinoda, "StarBED2: Large-scale, Realistic and Real-time Testbed for Ubiquitous Networks", *Proc. of TridentCom 2007*, Orlando, Florida, U.S.A., May 21-23, 2007.

[8] olsr.org, "The olsr.org OLSR daemon", http://www.olsr.org/.

[9] iperf, http://sourceforge.net/projects/iperf/.

[10] D. B. Faria, "Modelling Signal Attenuation in IEEE 802.11 Wireless LANs–Vol. 1", *Technical Report TR-KP06-0118*, Kiwi Project, Stanford University, July 2005.

[11] Scalable Network Technologies, Inc., "QualNet", http://www.scalable-networks.com/.

[12] University of Utah, School of Computing, "Emulab - Network Emulation Testbed", http://www.emulab.net/.

[13] Rutgers University, Wireless Information Network Laboratory , "ORBIT - Wireless Network Testbed", http://www.orbit-lab.org/.

[14] D. Johnson, T. Stack, R. Fish, D. Montrallo Flickinger, L. Stoller, R. Ricci, J. Lepreau, "Mobile Emulab: A Robotic Wireless and Sensor Network Testbed", *IEEE INFOCOM 2006*, Barcelona, Spain, April 23-29, 2006, pp.1-12.

[15] M. Kojo, A. Gurtov, J. Manner, P. Sarolahti, T. Alanko, K. Raatikainen, "Seawind: a Wireless Network Emulator", *Proc. of MMB 2001*, Aachen, Germany, September 2001.

[16] P. Zheng, L.M. Ni, "EMPOWER: A Network Emulator for Wireline and Wireless Networks", *Proc. of IEEE INFOCOM 2003*, San Francisco, U.S.A, March 30-April 3, 2003.

[17] Y. L. Huang, *et al.*, "SWOON: A Testbed for Secure Wireless Overlay Networks", *Proc. of the Conf. on Cyber Security Experimentation and Test (CSET'08)*, San Jose, CA, USA, July 28–August 1, 2008.

[18] T. Perennou, E. Conchon, L. Dairaine, M. Diazet, "Two-Stage Wireless Network Emulation", *Proc. of WCC2004*, Toulouse, France, August 2004.

[19] M. Bateman, C. Allison, A. Ruddle, "A Scenario Driven Emulator for Wireless, Fixed and Ad Hoc networks", *Proc. of PGNet2003*, Liverpool, U.K., June 2003, pp. 273-278.

[20] Naval Research Laboratory, Networks and Communication Systems Branch, "Common Open Research Emulator (CORE)", http://cs.itd.nrl.navy.mil/work/core/.

[21] J. Zhou, Z. Ji, R. Bagrodia, "TWINE: A Hybrid Emulation Testbed for Wireless Networks and Applications", *Proc. of IEEE INFOCOM 2006*, Barcelona, Spain, April 23-29, 2006.

978-1-4244-4148-8/09/$25.00 ©2009

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2009 proceedings.