Kun-chan Lan, Eranga Perera, Henrik Petander, Christoph Dwertmann, Lavy Libman, Mahbub Hassan, MOBNET: The design and implementation of a network mobility testbed for NEMO protocol, in Proceedings of the 14th IEEE Workshop on Local and Metropolitan Area Networks (LANMAN 2005), Chania, Greece, 18-21 September 2005, pp. 1-6.

MOBNET: The Design and Implementation of a Network Mobility Testbed for NEMO Protocol

Kun-chan Lan[*] Eranga Perera[*,§] Henrik Petander[*,†]
Christoph Dwertmann[*,‡] Lavy Libman[*] Mahbub Hassan[*,§]

[*] National ICT Australia Ltd
[†] Helsinki University of Technology
[‡] University of Bremen
[§] University of New South Wales

Abstract— **The inherent difficulty in faithfully modeling wireless channel characteristics in simulators has prompted researchers to build wireless network testbeds for realistic testing of protocols. While previous testbeds are mostly designed to provide a research environment of static wireless networks, our work is aimed to assess protocols used for mobile wireless networks (such as an on-board network on public transport vehicles). In this work, we describe our on-going efforts in designing and implementing a network mobility testbed for Network Mobility (NEMO) protocol. This paper attempts to provide an initial reference to identify the feature set necessary for a network mobility testbed. We first describe the architecture of our testbed. Next, we present some preliminary results to demonstrate the use of our testbed in evaluating the performance of NEMO protocol under different scenarios.**

I. INTRODUCTION

There is a growing interest in deploying high-speed wireless LANs on public transport vehicles to allow travelers to connect their devices to the Internet [1]–[3]. Such an on-board architecture typically consists of a high-speed mobile LAN and a Mobile Router (MR) that provides connectivity to the Internet through a wireless link (e.g. cellular or satellite). Thus, the mobile users simply connect their devices to the on-board network to enjoy the Internet service.

Many researchers have been working towards developing mechanisms for such an on-board architecture so that all nodes in the mobile network can still maintain their ongoing sessions as the mobile network changes its point of attachment to the Internet. The use of mobile router allows one to aggregate traffic from nodes in a mobile network, so that each node in the mobile network do not need to maintain an individual connection with bases stations. Such a traffic aggregation is particularly important when spectrum resources are scarce. The NEMO protocol [4] has been proposed as a standard, based on an extension of MIPv6 [5], to provide well-defined mechanisms for such a context.

While simulations are widely used for testing and evaluating protocols, it is well-known that physical layer models in most of the simulators are over-simplified and, as a results, introduce the majority of inaccuracies in wireless simulations. Motivated by the need of realistic environment for wireless protocol testing and evaluation, a number of wireless network testbeds [6]–[14] have been developed over the years. However, most of these existing wireless network testbeds were not designed to have the support of testing of network mobility protocols (such as NEMO) in mind.

In this paper, we present our on-going efforts in designing and implementing a network mobility testbed. To the best of our knowledge, our work is one of the first attempts to identify the feature set necessary for a network mobility testbed. We first present an overview of our testbed, including the network mobility protocol we implemented, the architecture of our testbed, and the software and hardware we utilized (Section III). We then present some preliminary results to demonstrate the use of our testbed in evaluating the performance of NEMO protocol (Section IV).

II. RELATED WORK

The difficulty and complexity of modeling wireless channel characteristics in the simulation has prompted the development of a number of wireless network testbeds over the years. In this section, we look at these wireless testbeds in terms of their design goal and their mobility control mechanism in comparison with our testbed.

The multi-hop ad-hoc testbed built at CMU [13] is used to test the implementation of Dynamic Source Routing (DSR) protocol in a real world environment. The APE testbed [14] is built to evaluate several different routing protocols. Rice TAP project [15] MIT Roofnet [16] aim to provide wireless infrastructure for mesh networking research. The common feature of the above testbeds is that they are all designed for specific project requirements. On the other hand, projects like ORBIT [6] and WHYNET [8] are aimed at producing large scale shared testbeds that are available to the entire research community and can be used for a broad range of wireless network research. Our MOBNET testbed is aimed at evaluating network mobility protocols and currently focusing on the NEMO Basic Support Protocol. Thus our testbed can be classified into the first category since we mainly target at the mobile network research community. On the other hand, our testbed can be also seen as a miniaturized testbed similar

---

[*] National ICT Australia is funded through the Australian Government's *Backing Australia's Ability* initiative, in part through the Australian Research Council.

to Mint [9] and EWANT [11] since we attempt to develop our testbed in a small laboratory environment.

Mobility control is essential in a wireless testbed to allow users to specify mobility patterns for their experiments. In APE testbed, node placement is done manually and the mobility is introduced by explicitly choreographing movement of volunteers carrying laptops. In CMU testbed, the nodes are implemented with cars carrying laptops acting as mobile adhoc nodes. The mobility of nodes in EWANT is emulated by connecting one wireless card to four external antennas through a RF multiplexer and switching antennas to emulate the node movement. In MiNT, each node is placed on a mobile robot that can be remotely controlled. Finally, in ORBIT testbed the mobility of a node is simulated through a separate mobility server which transfers the location state of a mobile node from one node in the grid to another.

In our testbed, we utilize a mobility emulator to emulate the movement of mobile router. By changing transmission power levels of the access points, we are able to emulate the variations of radio propagation for the mobile router in our testbed without physically moving anything. In addition, the mobility emulator software allows us to control the power levels of multiple access points using pre-made mobility patterns which can be based on signal strength measurements collected from real world. Note that, by providing a more realistic radio environment, the mobility emulator can also be utilized to facilitate the emulation of auto-rate selection mechanism implemented in most of the wireless cards today.

III. NETWORK MOBILITY TESTBED

The goal of this paper is to provide a useful starting point for researchers interested in designing a network mobility testbed. In this section, we first give a brief overview of NEMO protocol [4], the network mobility protocol supported by our testbed. Next, we present the architecture of our testbed. Our network mobility testbed consists of five main components: mobile router, mobility emulator, network emulator, remote management server and node generator. Finally, we describe the software and hardware we employ.

A. Overview of Network Mobility Protocol

A Mobile Network (MN) is a network that can move and attach to arbitrary points in the Internet. A Mobile Network Node (MNN) is a node that resides in the MN. A MN can only be accessed via specific gateways called mobile routers (MRs). The MR manages the movement of the entire MN and provides continuous and uninterrupted Internet access to the MNNs. Based on where it obtains its address and its mobility management capability, a MNN can be further classified into Local Fixed Node (LFN), Local Mobile Node (LMN) and Visiting Mobile Node (VMN) [1].

Our testbed implements the NEMO basic protocol [4]. We have implemented NEMO support as an extension to the Helsinki University of Technology's Mobile IPv6 implementation for Linux [17]. A mobile router combines Mobile IPv6 mobile node functionality with basic router functionality and manages the delivery of packets to and from the mobile network. Home agent (HA) is a mobility anchor point which assists MR by keeping track of the current point of network attachment, also known as care-of address (COA), of MR and delivering packets destined to the mobile network to the current care-of address of MR.

When a MR starts up, it sets up a bi-directional tunnel with the HA whereby it uses this tunnel to advertise the prefix address of networks that are connected to it. A MR has a home address which is used by the HA to communicate to it. When a MR moves away from its home network, it acquires a COA just like a mobile node would in MIPv6. To set up a bi-directional tunnel with the HA, the MR first sends a Binding Update to the HA and the HA then replies with a Binding Acknowledgement. The bi-directional tunnel between the HA and MR have endpoints with the address of the HA on one end and the COA of the MR on the other end. When messages are sent through this tunnel from the HA (originated from a Corresponding Node (CN)), it is encapsulated in another IP packet with destination address as the COA of the MR. The MR then decapsulates this packet and forwards it to the correct Mobile Network interface. Similarly packets are encapsulated at the MR end and decapsulated at the HA when a message is originated from a MNN.

B. Testbed Architecture

In our opinion, a mobile network testbed should provide the following *minimum* functionalities: the emulation of a mobile network, experimental control (such as topology configuration and mobility configuration), and management of the testbed [2]. We have implemented a NEMO-based mobile router (and are in the process of implementing a Node Generator) to emulate the mobile network. We design a mobility emulator to facilitate mobility control and utilize NISTNET network emulator to configure the topology. Finally, as an on-going work, we are implementing a Remote Management Server to monitor and maintain the testbed.

1) Mobile Router: Our NEMO implementation supports only NEMO implicit mode, in which HA is pre-configured with information about the mobile network behind MR, so registration messages need to carry only the current location of MR and its identity. The movement of MR between two different networks can incur connection outages to nodes in the mobile network. Our implementation can use the information from the link layer to trigger the handoffs when MR moves to a new wireless network. Furthermore, we implement Optimistic DAD [19] in the MR to remove the delay caused by IPv6 network attachment (about 1-2.5 seconds). The combination

---

[1] LFN is a node that belongs to the same subnet as the MR and has no mobility management capability (such as MIPv6). LMN is a node that belongs to the same subnet as the MR and has mobility management capability. VMN is from a different subnet and has its own mobility management capability.

[2] De et al. [18] has described a more comprehensive set of requirements for a general-purpose multi-hop wireless network testbed. We plan to extend our testbed to meet these requirements in our future work

of the link layer triggered handoffs and Optimistic DAD effectively reduce the handoff delays. Only the handoff delays related to the registration of the new location with MR remains. Such a delay equals to the round trip time between MR and its HA. In addition to reducing the handoff latency using the above techniques, we also implement a route optimization mechanism [20] in the MR to reduce the routing latency. More details are described in Section IV.

2) Mobility Emulator: To test the operation and evaluate the performance of the mobile network, we develop a mobility emulator to emulate the movement of the MR. By changing transmission power levels of the access points, we are able to emulate the variations of radio propagation for the MR in our test-bed without physically moving anything. The mobility emulator allows us to control the power levels of multiple access points using pre-made mobility patterns. These patterns can be created by hand, randomly based on a Markov Chains model, or from signal strength measurements collected from real world, and can be reproduced over and over again. A user-friendly graphical interface is also provided to allow users to visualize the selected patterns before, during and after the emulation, as shown in Figure 1.

a) Methodology:

a.1) Access point power level control: Some wireless access points on the market allow the alteration of the transmission power during operation via remote control. This can be done using a web interface, a shell-like environment (Telnet or SSH) or SNMP. The Cisco Aironet 1200 AP's used in our testbed support the first two methods. There are only a limited set of transmit power levels that can be set for our Cisco AP (e.g. the configurable power levels are 1, 5, 20, 30, 50, 100mw).

a.2) Modeling the variations of signal strength: To emulate the randomness of the mobility of the mobile device, one can model the signal strength variation as a random process. Since the number of available power levels of an AP point is finite, it is feasible to model such randomness using a finite-state Markov chain process. Two types of Markov chain models are currently implemented: continuous mode and discrete mode. In continuous mode, the current state stays for a random period of time. In discrete mode, the process stays at any state for a fixed time interval.

b) Architecture: Our mobility emulator (MobE) consists of three main components: an input parser, a graphical user interface and an access point controller, as shown in Figure 2.

There are two types of input files for the parser: configuration and emulation files. The configuration files contain information about the APs, such as IP addresses, credentials, names and the number of power levels supported by the AP. Furthermore, the emulation mode (deterministic or Markov) and the Markov states are set here. Emulation files contain the actual emulation data for the deterministic mode, and can be generated from real-world measurement data.

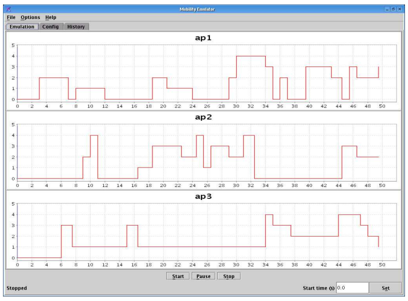

Fig. 1. Snapshot of the Emulator

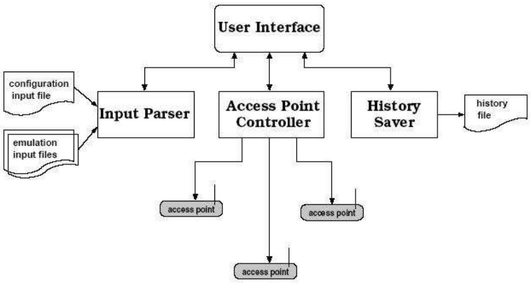

Fig. 2. Architecture of Mobility Emulator

The user interface is implemented in Java. It has three main views: Configuration, Emulation and History, as shown in Figure 1. After loading a configuration file, the AP configuration can be viewed in the "Configuration" tab. Changes can be made here to most of the parameters. When running the emulation, the emulation procedure is visualized as a moving graph in the "Emulation" tab. After the emulation has been completed, the "History" tab shows the emulation history graph, which then can be saved to a text file or exported as a JPEG or PNG image file.

The access point controller controls the power level of the access point by opening a telnet connection to the Cisco IOS shell. For example, one can set the AP's transmission to 10 mW by the command "power local cck 10".

3) Remote Management Server: The function of remote management is essential in order for a remote user to access and manage the testbed. The tasks of testbed management typically involve the initial setup of the testbed, monitoring the status of each node and maintaining them, and execution of experiments. In the next version of MOBNET, we plan to incorporate a remote management server to support remote execution of experiments and administration of the testbed.

4) Node generator: At present, each MNN and CN is implemented using one single machine. This poses a problem as the experiment involves large number of mobile nodes (e.g. many passengers in a bus) or large number of corresponding nodes (e.g. a traveler browses many web sites during his trip). To improve the scalability of the testbed, we are currently in the process in implementing each MNN/CN as a single process so that multiple MNNs/CNs can be run on the same machine. We will adopt an approach similar to that used in MultiNet project [21], where a single wireless card is abstracted into multiple virtual cards, to emulate the multiple wireless connections between MNNs and the MR. In our case, each MNN process is connected to the MR via a virtual card.

The link between each MNN/CN pair can be configured using NISTNET emulator. We plan to implement a node generator to facilitate the generation and configuration of nodes (such as node type, number of nodes, node lifetime, etc.).

C. Hardware and Software

Our hardware equipment currently consists of six desktop computers and five laptops with processor speed varying from 350MHz to 3GHz and memory size from 128MB to 512MB. The current physical setup of our testbed is shown in Figure 3.

Based on a standard Linux kernel, our NEMO software implements the basic functionalities of MR and HA. The software consists of three components:

- Kernel modules for MR and HA based on Linux 2.4.26 kernel, which handle the mobility management of the mobile network.
- Configuration scripts and a statistics tool, mipdiag, to monitor NEMO traffic.
- IPv6-capable access routers are used. The routers send router advertisements via a router advertisement daemon, radvd. The daemon advertises the mobile network prefix to the nodes in the mobile network so that MR can learn its location and configure itself with a COA.

Our testbed can be configured using a set of startup scripts on each node. These scripts configure the fixed settings of the testbed which do not vary between different scenarios, unless other mobility protocols (such as HMIPv6, etc) are used in addition to NEMO. For NEMO, in most cases it is sufficient to change only the settings in the Mobile Router, although some mechanisms require also re-configuration of the HA. Different test scenarios can be setup by changing the settings of the network emulator and mobility emulator. Specifically, the network topology can be configured with the NISTNET network emulator and the mobility pattern can be configured using the mobility emulator.

IV. PRELIMINARY RESULTS

To understand the effect of NEMO protocol on UDP and TCP traffic, we perform experiments on our testbed with the mobile network first moving from home network (HN) to foreign network 1 (FN1), then moving from foreign network 1 to foreign network 2 (FN2), and finally returning back to HN from FN2. The logical topology used in our experiments is shown in Figure 4.

During movements of the mobile network, traffic is continuously sent from CN to MNN. We use ethereal [22] to collect packet traces of the traffic in our experiments. The traces are then analyzed to understand the effects of mobility on packet loss (UDP) and throughput (TCP). The sampling period for each experiment is 10 seconds. We use iperf [23] to generate UDP and TCP traffic. The bit rate of UDP stream is 100Kbits/s with the packet payload size set to 200bytes. We choose a small packet size to highlight the effect of handoff latencies. For TCP, we use the same payload size as UDP with a 16K window size. We use NISTNET to emulate the Internet. The bandwidth is limited to 2Mbits/s on all links. The propagation delays between different nodes are shown in Table I.

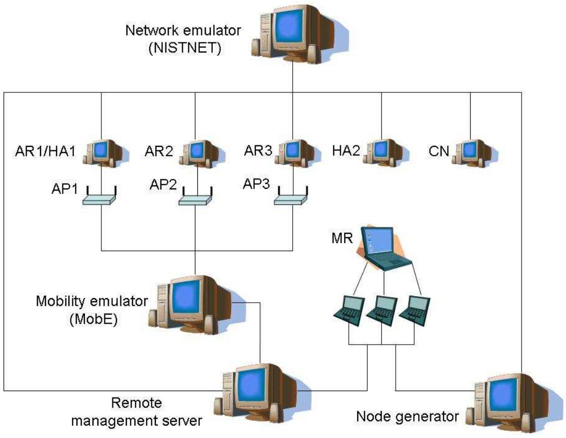

Fig. 3. Current Physical Setup of the Testbed

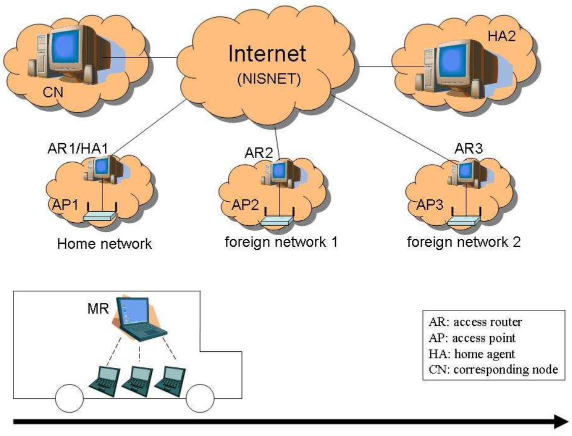

Fig. 4. Logical topology of experiments

A. Improvement of handoff latency

As shown in Table II, there is a large discrepancy in the observed performance measurements when MR moves between different networks. In particular, the performance is severely degraded when the mobile network moves from HN to FN1 and from FN1 to FN2. After a closer look, we find that multiple factors contribute to the NEMO handoff latency, including Layer 2 handoff, Duplicate Address Detection (DAD),

node pair	propagation delay
AR to $HA_{\textbf{MR}}$	10 ms
AR to $HA_{\textbf{VMN}}$	50 ms
AR to CN	40 ms
$HA_{\textbf{MR}}$ to CN	40 ms
$HA_{\textbf{VMN}}$ to CN	40 ms
$HA_{\textbf{MR}}$ to $HA_{\textbf{VMN}}$	40 ms

TABLE I

PROPAGATION DELAY BETWEEN VARIOUS NODE PAIRS SET BY NISTNET

	TCP) throughput (Mbits/sec)	UDP loss (%)
Home	1.85	0
Home to FN1	0.88	27
FN1 to FN2	1.25	17
FN2 to Home	1.46	1.85

TABLE II

EFFECT OF NEMO HANDOFF ON THE PERFORMANCE OF TCP AND UDP

distance between MNN and HA, and proxy DAD.

As shown in Table III, the dominant factor that contributes to the observed long latency is DAD which is a part of IPv6 network attachment and address auto-configuration mechanism. Due to DAD, NEMO basic handoff mechanism performs well only when MR returns to its home network from FN2 since that no DAD is performed as a part of this handoff.

Optimistic DAD [19] has previously been proposed as a mechanism to removes the handoff delay caused by IPv6 network attachment. By using Optimistic DAD, which performs the check for duplicate addresses in parallel with the use of the new address, one can reduce the NEMO handoff latency to one order of magnitude (e.g. the handoff latency is reduced from 1746ms to 178ms when moving from FN1 to FN2, as shown in Table III).

As shown in Table IV, the use of Optimistic DAD dramatically improves both the TCP and UDP performance [3]. Note that the performance of mobile network is still poor when the MR moves from its home network to FN1 due to the use of proxy DAD. The HA of a MR typically performs proxy DAD to reserve the home address of MR to ensure that no other node will use this address while MR is away from its home network. Note that, as shown in Table II, handoff latency has a larger effect on bulk TCP traffic than UDP traffic (e.g. the TCP throughput degrades more than 50% when moving from Home to FN1) due to that TCP have to slowly recover from slow-start after the handoff.

movement	Layer 2 handoff	DAD	RTT between MNN and HA	Proxy DAD
Home to FN1	110 ms	1616 ms	20 ms	1000 ms
FN1 to FN2	153 ms	1656 ms	25 ms	0 ms
FN2 to Home	161 ms	0 ms	4 ms	0 ms

TABLE III

FACTORS THAT CONTRIBUTE TO NEMO HANDOFF LATENCY

	TCP) throughput (Mbits/sec)	UDP loss (%)
Home	1.85	0
Home to FN1	1.32	11.5
FN1 to FN2	1.58	1.8

TABLE IV

EFFECT OF NEMO HANDOFF ON TCP AND UDP WITHE USE OF ODAD

B. Improvement of routing for visiting mobile nodes

Visiting Mobile Nodes (VMNs) are mobile IPv6-capable nodes which are temporarily in the mobile network. Thus, the VMNs treat the Mobile Network as a visited network, and register their location there with their Home Agent, $HA_{\textbf{VMN}}$. $HA_{\textbf{VMN}}$ delivers any packets sent to the VMNs to their current location which, in this case, happens to be the mobile network attached to MR. This arrangement results in two layers of mobility – the host mobility of VMNs and the mobility of the network containing the VMNs. The network mobility is hidden from VMNs via MR and NEMO protocol.

The packets between VMN and CN are routed through the HA of VMN first and then delivered to the current location of VMN, which appears to $HA_{\textbf{VMN}}$ to be located in the home network of MR. HA of MR , $HA_{\textbf{MR}}$, will deliver the packets from there to the current location of MR. As a result, the route between VMN and CN is not optimal and has two extra routing legs: MR-$HA_{\textbf{MR}}$ and $HA_{\textbf{MR}}$- $HA_{\textbf{VMN}}$.

Use of Mobile IPv6 route optimization [5] allows VMN to communicate almost directly with CN by informing CN of its current location. However, Mobile IPv6 route optimization does not remove the extra routing leg between MR and $HA_{\textbf{MR}}$. An extension to the MIPv6 and NEMO protocols was proposed previously [20] to allow VMNs to operate more efficiently in a mobile network. This extension allows MR to deliver packets directly to CN, instead of routing all traffic through $HA_{\textbf{MR}}$. In the extended protocol [4], MR gets a new block of addresses from the visited network access router using DHCPv6 and advertises these addresses to the VMNs in the mobile network. VMNs can then use Mobile IPv6 route optimization with the topologically correct addresses to communicate with CNs. This will result in an optimal route and also reduce protocol header overhead.

To ensure that VMN can still communicate with CNs when the MR moves, the MR sends a registration message to the old access router which now acts as a HA for the block of addresses MR acquired from it previously. The old access router will then forward all packets to this block of addresses to the new location of VMN. MR repeats this procedure every time it moves, so that the VMNs will always have a topologically optimal address. The addresses from the old access router will be expired after MR manages to obtain a new block of addresses at its current location. This overlap guarantees that VMNs can still receive packets to their old location while the MR moves to a new location.

Table V shows the effect of different route optimization techniques on the end-to-end latency between VMN and CN. By removing the routing legs between $HA_{\textbf{MR}}$ and $HA_{\textbf{VMN}}$ and between MR and $HA_{\textbf{MR}}$, we are able to reduce the end-to-end latency by half in our testbed setting. With NEMO route optimization, the TCP throughput performance for a VMN

---

[3] We do not show the statistics for the case when the MR returns to home since ODAD does not have any effect in this scenario.

[4] We refer this extension as "NEMO route optimization" for the rest of this section.

(1.75Mb/sec) is close to that of a LFN (1.85Mb/sec, as shown in Table IV).

Note that, during the setup of our testbed, we have experimented with wireless cards from different vendors, namely Lucent, IBM and Netgear. There is a large variation of L2 handoff latency between different wireless cards, as shown in Table VI. From Table VI and Table III, one can see that the choice of wireless cards can have a significant impact on the NEMO handoff latency for our experiments. Such a large variation in the layer 2 handoff latency with different hardwares was also previously observed by other researchers [24].

V. CONCLUSION AND FUTURE WORK

Motivated by the difficulty in accurately modeling wireless channel characteristics in simulations, a number of wireless network testbeds have been developed over the years to provide realistic testing of protocols. Unlike most of the existing wireless testbeds, our work is aimed to provide an experimental environment to study protocols used for moving wireless networks. In this paper, we describe our on-going efforts in designing and implementing a network mobility testbed. We also present some preliminary results to demonstrate the use of our testbed by applying some optimization techniques to improve the handoff and routing latencies of NEMO. This paper is intended to provide an initial reference to identify the necessary components for a network mobility testbed.

In our testbed, we emulate the movement of the MR by changing transmission power levels of access points. However, during our experiments we observed that sometimes there is a large variation in the noise level (which might be due to that our testbed is located on the same floor with nine other APs and a large number of computers) which, as a result, reduces the usefulness of our mobility emulator (i.e. the variation in the noise level is larger than the signal boost gained from increasing transmission power). We are investigating the causes of such noise variations to setup a more stable environment. In addition, we are currently in the process of implementing the Node generator and the Remote Management Server to facilitate larger-scale experiments and remote monitoring and management of the testbed. Due to the time constraints, in this paper we did not evaluate the level of realism that our testbed can provide (e.g. by comparing against results obtained from a pure simulation environment), We will look into such issues in our future work.

Optimization	**CN-VMN latency**	**packet header overhead**	**TCP throughput**
No optimization	188 ms	5%	1.12 Mb/sec
MIPv6 route optimization	110 ms	4%	1.46 Mb/sec
NEMO route optimization	86 ms	2 %	1.75 Mb/sec

TABLE V

PERFORMANCE AND PER-PACKET HEADER OVERHEAD WITH ROUTE OPTIMIZATION

type of cards	**Lucent**	**IBM**	**Netgear**
L2 handoff	110 ms	733 ms	1400 ms

TABLE VI

LAYER 2 HANDOFF LATENCY OF DIFFERENT WIRELESS CARDS

REFERENCES

[1] http://www.nautilus6.org. Nautilus 6 working group.

[2] http://www.connexionbyboeing.com. Connexion by Boeing.

[3] R. Tönjes, K. Mößner, T. Lohmar, and M. Wolf. OverDRiVE — spectrum efficient multicast services to vehicles. In *IST Mobile Summit*, Thessaloniki, Greece, June 2002.

[4] V. Devarapalli, R. Wakikawa, A. Petrescu, and P. Thubert. Network mobility (NEMO) basic support protocol, January 2005. Internet Request For Comments : RFC3963.

[5] D. Johnson, C. Perkins, and J. Arkko. Mobility support in IPv6, June 2004. Internet Request For Comments : RFC3775.

[6] 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 *Wireless Communications and Networking Conference (WCNC'05)*, March 2005.

[7] B. White, J. Lepreau, L. Stoller, R. Ricci, S. Guruprasad, M. Newbold, M. Hibler, C. Barb, and A. Joglekar. An integrated experimental environment for distributed systems and networks. In *Proceedings of the 5th symposium on Operating systems design and implementation*, Boston, MA, USA, December 2002.

[8] http://pcl.cs.ucla.edu/projects/whynet. WHYNET.

[9] P. De, A. Raniwala, S. Sharma, and T. Chiueh. MiNT: A miniatuirized network testbed for mobile wireless research. In *Proceedings of the IEEE Infocom*, Miami, FL, USA, March 2005.

[10] G. Judd and P. Steenkiste. Repeatable and realistic wireless experimentation through physical emulation. In *Proceedings of the 2st Workshop on Hot Topics in Networks (HotNets-II)*, Cambridge, MA, USA, November 2003.

[11] S. Sanghani, T. X Brown, S. Bhandare, and S. Doshi. EWANT: The emulated wireless ad hoc network testbed. In *IEEE Wireless Communications and Networking Conference (WCNC)*, pages 16 – 20, New Orleans, Louisiana, USA, 2003.

[12] James T. Kaba and Douglas R. Raichle. Testbed on a desktop: strategies and techniques to support multi-hop MANET routing protocol development. In *Proceedings of the 2nd ACM international symposium on mobile ad hoc networking and computing*, pages 164 – 172, Long Beach, CA, USA, 2001.

[13] D. A. Maltz, J. Broch, and D. B. Johnson. Experiences designing and building a multi-hop wireless ad hoc network testbed. *CMU TR99-116*, 1999.

[14] H. Lundgren, D. Lundberg, J. Nielsen, E. Nordstrm, and C. Tschudin. A large-scale testbed for reproducible ad hoc protocol evaluations. In *the 3rd annual IEEE Wireless Communications and Networking Conference (WCNC)*, Orlando, FL, March 2002.

[15] R. Karrer, A. Sabharwal, and E. Knightly. Enabling large-scale wireless broadband: The case for TAPs. In *Proceedings of HotNets 2003*, Cambridge, MA, USA, November 2003.

[16] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris. Link-level measurements from an 802.11b mesh network. In *Proceedings of the ACM SIGCOMM*, Portland, Oregon, USA, August 2004.

[17] http://www.mobile ipv6.org. MIPL Mobile IPv6 for Linux.

[18] P. De, A. Raniwala, S. Sharma, and T. Chiueh. Design considerations for a multi-hop wireless network testbed (unpublished).

[19] N. Moore. Optimistic Duplicate Address Detection for IPv6, February 2005. draft-ietf-ipv6-optimistic-dad-05.txt.

[20] E. Perera, A. Seneviratne, and V. Sivaraman. Optinets: An architecture to enable optimal routing for network mobility. In *International Workshop on Wireless Ad-hoc Networks (IWWAN 04)*, Lawrence, Kansas, USA, May 2004.

[21] R. Chandra and P. Bahl. MultiNet: Connecting to multiple IEEE 802.11 networks using a single wireless card. In *Proceedings of the IEEE Infocom*, Hong Kong, March 2004.

[22] http://www.ethereal.com/. Ethereal.

[23] http://dast.nlanr.net/Projects/Iperf/. Iperf.

[24] A. Mishra, M. Shin, and W. A. Arbaugh. An empirical analysis of the IEEE 802.11 MAC layer handoff process. *ACM Computer Communication Review*, 33(2), 2002.