A Plan for Performing Large-Scale DTN Experiments on the MeshTest Wireless Testbed

Brenton Walker
Laboratory for
Telecommunications Sciences
brenton@ltsnet.net

Padma Mundur
University of Maryland
College Park, MD, USA
pmundur@umiacs.umd.edu

Ginnah Lee
University of Maryland
College Park, MD, USA
jn1114@gmail.com

Kun Lin
University of Maryland
College Park, MD, USA
klin@umd.edu

***Abstract*—MeshTest is a laboratory-based wireless testbed that can subject real wireless devices to reproducible mobile scenarios. It uses shielded enclosures and an RF matrix switch to dynamically control the attenuation experienced between pairs of nodes. The testbed can be controlled remotely, and is an ideal platform for development of and experimentation with real DTN implementations.**

Unfortunately the RF switch can only accommodate 16 wireless nodes, which is insufficient for many types of experiments. In many interesting DTN scenarios, however, at any given time most nodes in the network will be completely isolated, or will be part of a small isolated group. As long as the number of nodes that are involved in wireless exchanges at any given time is less than the number of switch inputs, it should be possible to use the existing testbed hardware to run DTN experiments involving large numbers of nodes. In this paper we explain how we plan to use node virtualization and multiple MeshTest systems to enable much larger DTN experiments. Implementing this plan will be a summer project for several researchers from UMD and LTS.

I. Introduction

MeshTest consists of a rack of computers in shielded enclosures, an RF matrix switch, and a server that provides experiment control, as depicted in Figure 2. The RF from each computer's WiFi card is cabled through the enclosures and into the matrix switch of programmable attenuators. The matrix switch allows us to control the attenuation experienced between pairs of devices. By varying the switch settings in real time we are able simulate the effects of arbitrary physical scenarios and mobility. MeshTest uses ORBIT's testbed management software to control the nodes, but we are migrating to an Emulab-based setup with a more diverse collection of wireless devices.

More background on the theory and performance of MeshTest can be found in [1].

The MeshTest testbed has been used for mobile wireless Delay-Tolerant Network (DTN) experiments in the past [2]. Unfortunately the physical limitations of the system severely limit the size of experiments that can be run. The largest RF switch we have can support up to 16 nodes, and smaller switches can accommodate 8. We have investigated the feasibility of connecting many RF switches together to make a larger testbed [3], and while this approach seems feasible, it will be expensive and will only increase the size of the testbed to 32-64 nodes.

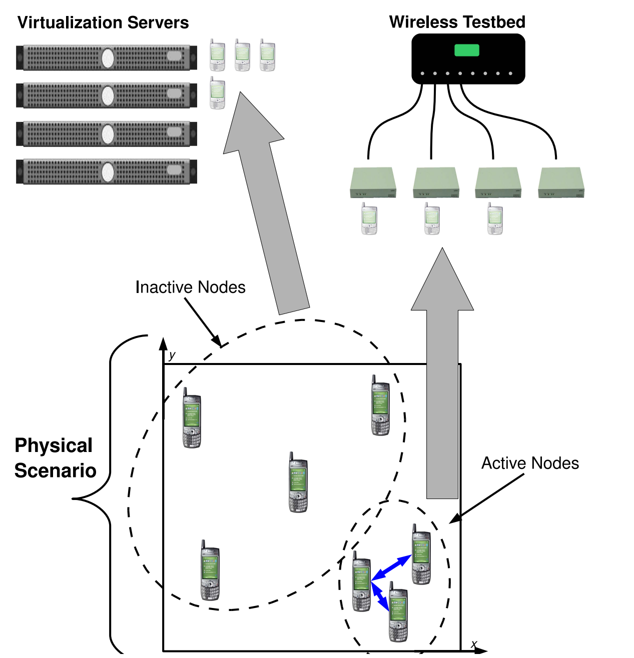

Fig. 1. Visual representation of the virtualization plan. Nodes that are within wireless range of other nodes in the underlying physical scenario will have their virtual images moved to the actual wireless pnodes. Nodes that are isolated will be run on the virtualization servers.

II. Node Virtualization Plan

Fortunately the types of experiments we are most interested in involve mobile wireless DTNs. In such a network most of the nodes will find themselves isolated, or in part of a small isolated group most of the time. This characteristic allows us to reduce a large experiment to a series of smaller-scale mobile interactions. In MeshTest the limiting resource is the number of interfaces on the RF matrix switch. If a node in our experiment is geographically isolated, there is no reason for it to be connected to the switch at all. We propose to use node virtualization to move running node images on to and off of the wireless testbed nodes during an experiment.

Let us introduce some terminology. During an experiment there is an underlying **physical scenario** that we are trying to simulate. The physical scenario will involve some arrange-

Fig. 2. The MeshTest testbed. The two racks on the left are the original twelve node testbed. The cart on the right is a smaller testbed that can support eight nodes.

ment of nodes. Each node involved in the physical scenario corresponds to a **virtual node** or **vnode**. The actual computers that the vnode images run on are **physical nodes** or **pnodes**. The actual wireless devices in the testbed are pnodes, as are several powerful rack-mount servers which have no wireless hardware. Each vnode has a running image on some pnode, and the vnode images can be moved from one pnode to another as necessary to achieve the desired physical scenario.

Vnode images that are currently involved in wireless interactions would be moved to the wireless pnodes in the testbed. Vnodes that are geographically isolated in the physical scenario would be moved to the rack-mount servers. Furthermore, vnodes involved in separate isolated wireless interactions could be moved to pnodes connected to different RF switches. This means that the testbed can be expanded to support more vnodes by increasing the number of RF switches and pnodes, as long as no individual wireless encounter involves too many vnodes. The evoloution of the underlying physical scenario and the migration of vnode images would be controlled by the Emulab testbed control system. The system will have to be similar to the Emulab-based control system used by the CMU wireless emulator [4].

III. Related Work

Virtualization can be used to increase the number of nodes that can be used in an experiment [5]. In the wireless context the wireless medium itself is a shared resource and virtualization can be used to better manage access to it. For example WINLAB researchers have investigated moving vnodes among pnodes to achieve a sort of simulated mobility with stationary nodes, and also using multiple vnode images and different channels on pnodes with multiple wireless interfaces [6].

Virtualization can also be used to run live experiments with a completely simulated PHY and MAC layer. In [7] Virtual Network User Mode Linux is used to experiment with several mobile wireless DTN scenarios.

IV. Challenges

Choosing and understanding the effects of a particular virtualization method will be our main challenge. A concise overview of several virtualization techniques is provided in [8]. The computational overhead of pure emulation is too high, especially since our wireless nodes will be mainly low power Soekris machines [9]. The paravirtualization tool, Xen [10], is extremely attractive, since it has low overhead, and also provides fast vnode image migration. One problem with Xen is that it will require modified wireless drivers which may not function as expected. The other main candidate is the OS-level virtualization tool OpenVZ [11]. OpenVZ supports quick migration of running vnode images, and the wireless interfaces are virtualized, so will not require modification.

There are still concerns about the realism of both approaches, however. One of the lessons we learned in our previous experiments is that MAC-layer effects and idiosyncrasies in the drivers can have large effects on the results of an experiment [2], [12]. If two pnodes form a MAC-layer association and then have their vnode images swapped out, they will still try to maintain the old MAC-layer association for a period of time. The driver may also maintain state from one wireless encounter to another One potential solution to this is to reinitialize the wireless interface whenever we move a vnode image onto a pnode. This still is not completely realistic, since a node will begin each wireless interaction with a newly initialized driver.

References

[1] T. C. Clancy and B. D. Walker, "Meshtest: Laboratory-based wireless testbed for large topologies," in *IEEE TridentCom 2007*, pp. 1–6, 2007.

[2] M. Seligman, B. D. Walker, and T. C. Clancy, "Delay-tolerant network experiments on the meshtest wireless testbed," in *Challenged Networks* (M. H. Ammar and K. Psounis, eds.), pp. 49–56, ACM, 2008.

[3] B. Walker and J. Seastrom, "Addressing scalability in a laboratory-based multihop wireless testbed," in *Tridentcom 2009*, 2009.

[4] "Carnegie Mellon University Wireless Emulator." http://www.cs.cmu.edu/~emulator/.

[5] S. Maier, D. Herrscher, and K. Rothermel, "Experiences with node virtualization for scalable network emulation," *Comput. Commun.*, vol. 30, no. 5, pp. 943–956, 2007.

[6] G. Bhanage, I. Seskar, Y. Zhang, and D. Raychaudhuri, "Evaluation Of OpenVZ Based Wireless Testbed Virtualization," Tech. Rep. WINLAB-TR-331, Rutgers University, 2008.

[7] L. Romero Amondaray and J. Seoane Pascual, "Delay tolerant network simulation with vnuml," in *CHANTS '08: Proceedings of the third ACM workshop on Challenged networks*, (New York, NY, USA), pp. 109–112, ACM, 2008.

[8] K. Kolyshkin, "Virtualization in Linux." http://download.openvz.org/doc/openvz-intro.pdf.

[9] "Soekris Engeneering." http://www.soekris.com/.

[10] "The Official Xen Project Site." http://www.xen.org/.

[11] "OpenVZ." http://wiki.openvz.org/.

[12] J. Seastrom, "Characterizing transport layer behavior in the meshtest wireless testbed," Master's thesis, University of Maryland-College Park, 2008.