

Design Considerations for a Mobile Testbed*

Keon Jang
KAIST
keonjang@an.kaist.ac.kr

Shinae Woo
KAIST
shinae@an.kaist.ac.kr

Sue Moon
KAIST
sbmoon@kaist.edu

Introduction

Today's cellular technologies allow users to access the Internet in a car or on a high-speed train, albeit the access bandwidth is far lower than that through a typical wired access technology such as Ethernet. Numerous new applications and services are under development for users of wireless technologies. Location-based services (e.g, GPS navigation) and streaming media services are just two examples from the vast set of candidate applications and services. Not only researchers, but application and service developers could benefit from a testbed where they could experiment with mobility in the network.

A typical testbed for a mobile wireless technology has no mobile component in it. Notable exceptions are: mobile Emulab [6] and DOME [1]. Mobile Emulab consists of mobile robots mounted with an 802.11 interface in a confined indoor setting. It offers 1 cm accuracy in robot maneuvering and location reporting. Mobile Emulab has the same access policy as Emulab: users request and are granted fixed resources in advance [9]. DOME (Diverse Outdoor Mobile Environment) has multiple DTN (Disruption Tolerant Networking) testbeds. One of them, DieselNet, consists of computers that are mounted on buses and have multiple radio interfaces. They connect with computers on other buses or scan for DHCP (Dynamic Host Configuration Protocol) offers. Another DTN in DOME is TurtleNet, which actually has Wood turtles as the carrying medium of sensors. These sensors monitor the turtles' behavior and habitats, and upload the data whenever the turtles pass by the collection post. While Mobile Emulab is open for public access, all networks in DOME are for special-purpose.

The limited number of mobile testbeds is a clear demonstration of difficulties in building and maintaining a mobile testbed. A mobile testbed poses chal-

lenges that are not present in a fixed or wired networking environment. We envision a testbed that is mobile as DieselNet and is open to public as Emulab or PlanetLab [4]. In this paper, we outline the challenges, review the design choices and discuss our approach towards an open and public mobile testbed.

Challenges

We plan to mount the nodes of our testbed on moving vehicles that run on a regular schedule along a fixed route. Human mobility is only a few kilometers per hour and is slower than most vehicular mobility. It is hard to replicate in any predictable and cost-effective way. Predictability of the schedule and the route is important, as the users of the testbed could experiment repeatedly under similar mobility conditions. We target buses and subways for installation and expect reliable remote accessibility as the key challenge. Below, we review challenges in light of reliable remote access.

(1) Power Management

A node installed on a moving vehicle should take power from the vehicle. As the vehicle is turned on and off according to the transportation schedule, the node will come up and shut down. We can attach enough battery for the node to last while the vehicle is parked and turned off. Either we keep the node up all the time or not, we should attach battery to the node for it to shut down safely after the vehicle cuts off the power supply. We have not decided yet whether the battery should last until the vehicle is turned on again. This decision depends on the space availability at the installation premise.

(2) Mobile Wireless Networking Technology

In our choice of wireless technologies, three major issues are: the pricing model, coverage, and mobility support. If a flat-rate service plan is available for a certain networking technology, then we can keep the mobile node on continuously for reliable access. If not, then we need to put a cap on the usage or design an

*Keon Jang was supported by Brain Korea 21 Project, the School of Information Technology, KAIST, in 2008. Shinae Woo and Sue Moon were supported by the IT R&D program of MKE/IITA [2007-F-038-02, Fundamental Technologies for the Future Internet].

accounting policy to share the cost. For ease of experimentation and large user base, we favor wireless networking technologies of good coverage. We also need to match mobility of our interest to the mobility of a specific technology. For example, if we want to develop wireless service for users on a very high-speed train (i.e. above 300km/h), we cannot use WiBro as it supports speed only up to 120km/h.

(3) Resource Management

Mobile wireless nodes have limited power and bandwidth, and resource management is a crucial component of the testbed. PlanetLab uses a virtualization technology to share computing resource and memory, and can limit the bandwidth on each slice. At the moment, PlanetLab nodes use only one network interface, and does not have provisions for multiple network interfaces. OneLab a federated PlanetLab in Europe, aims to incorporate heterogeneous wireless technologies for multiple users at the same time, and the resource sharing is at the core of their expansion [5].

(4) Remote Accessibility

No matter what wide-area networking technology we use, we rely on the mobile service provider for access to the deployed mobile node. Many mobile service providers have NAT (Network Address Translator) boxes at the gateway between the cellular network and the Internet, and assign only dynamic IP addresses to mobile nodes. For any other node to connect to a mobile node behind a NAT, we should devise a NAT traversal scheme tailored to each service provider's NAT configuration.

Design Choices

In this section we review design choices for the challenges listed above.

(1) Choice of wireless networking technologies

We consider the following three wide-area networking technologies: CDMA (Code-Division Multiple Access) 1xEV-DO (Evolution-Data Only), HSDPA (High-Speed Downlink Packet Access), and WiBro (Wireless Broadband). We do not include the 802.11 wireless LAN technology, for it has limited coverage of tens of meters and little support for mobility. CDMA 1xEV-DO, HSDPA, and WiBro are all deployed in Korea and offer data rates of a few hundred kilobits per second or higher. CDMA 1xEV-DO, HSDPA, and W-CDMA (Wideband CDMA), have nation-wide coverage in many countries around the world, and render our testbed easily expandable to wider geographic cov-

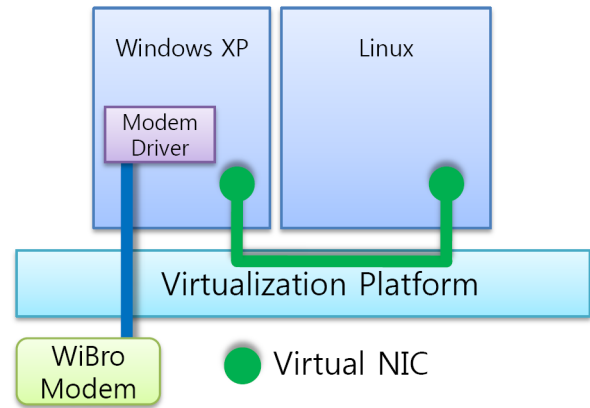


Figure 1. Example of running Windows and PlanetLab Node OS in virtualization environment.

erage.

(2) Resource management

Our open testbed requires a resource sharing policy and a resource management system to enforce the policy. The simplest approach is to add support for the wireless networking technology of our choice to PlanetLab. This approach clearly has the benefit in that PlanetLab already provides a management framework for user accounts and service deployment through user-level virtualization. Existing users of the PlanetLab could easily extend their use to include the nodes in our testbed.

(3) Support for wireless networking technology

Our testbed will be based on Linux like many other testbeds, for the kernel code is available publicly. Most 3G, 3.5G and WiBro modems in South Korea do not have publicly available Linux drivers. Developing a Linux driver is a time-consuming task. Worse yet, developing one for every modem in the market is not feasible. Fortunately, some CDMA and HSDPA modems share a small number of USB-to-serial chip sets with publicly available drivers and can run through the serial port. PPP (Point-to-point Protocol) over the serial line is easy to set up, regardless of the mobile service provider. This type of modems will be our first consideration.

WiBro modems do not work through the serial port as described above, for it is recognized as a NIC (network interface card). It requires proprietary connection software and only runs on Windows at this point.

If we decide to use WiBro, then we can only run

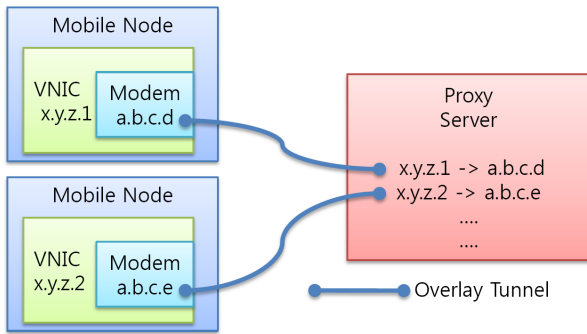


Figure 2. Proxy approach to support dynamic IP address and NAT

the WiBro modem on Windows or develop a Linux WiBro modem driver. Without a Linux WiBro driver, we should consider virtualization framework (e.g., Xen [3] and VMWare[2]) to run both Windows and Linux in a single node. Figure 1 visualizes our virtualization framework. There the WiBro modem connects to a Windows virtual machine and the Linux node connects through a virtual NIC to access the Internet.

We have not decided yet whether to use serial modems only or use virtualization to support WiBro and other modems that do not work on Linux.

(4) Addressing

One approach to support Dynamic IP address is to simply use dynamic DNS [8] to keep the mapping between a domain name and an IP address of the node up-to-date. Dynamic DNS provides a mechanism to update DNS resource record without human intervention. Most applications perform DNS resolution before making a connection and thus dynamic DNS will allow access to testbed nodes by the domain name. Dynamic DNS will not work for PlanetLab nodes, for the static IP address is a part of the authentication key in the boot-up process.

Another approach is to install proxy servers and relay traffic between testbed nodes and the rest of the Internet. The proxy server maintains a mapping between a publicly advertised static IP address for a testbed node and a dynamic IP in use. The testbed node has a virtual network interface card (VNIC) to make an illusion that it owns the static IP address that is actually assigned to the proxy server. This approach is conceptually very similar to Mobile IP[7]. We can think of the publicly advertised static IP address at a proxy server as a *home address*, the dynamic IP address of the testbed node as a *care-of address*, and the proxy server as a *home agent*. We note that mobile service providers do not use Mobile IP, as the node is

expected to stay within one network and do not roam outside.

A proxy server has several advantages over dynamic DNS.

- If we use PlanetLab, we need not modify My-PLC¹.
- Virtual NIC hides short outages in the wireless link from the IP layer and up, and TCP sessions remain intact.
- The proxy server can work as a relay and solve the NAT traversal problem.

There are disadvantages to using a proxy server. The traffic detours through the proxy and the forwarding path is different from the direct path. Packets between the proxy server and the testbed node carry extra bytes for the encapsulation and may require fragmentation and reassembly, which in turn affect the performance. As a result, performance study over the path to a testbed node may not work and TCP may perform poorly. Another drawback of this approach is waste of IP address space as each node requires one dedicated static IP address on top of the address assigned to the wireless modem.

Ongoing Work

We have set up a MyPLC server node and are in the process of installing vanilla PlanetLab nodes. These nodes will eventually turn into mobile nodes. In parallel, we are experimenting with a Sprint Nextel CDMA 1xEV-DO modem, a KTF HSDPA modem, and a SK-Telecom HSDPA modem. The first modem has a publicly available Linux device driver, while we are still searching for public device drivers for the remaining two modems. Also we are setting up Xen and VMware platforms to run multiple OSes on a single testbed node.

Acknowledgements

We thank Geoff Voelker at the University of California, San Diego and Per Johansson at California Institute for Telecommunications and Information Technology (Calit2) for their insightful comments and help with the testbed setup.

¹MyPLC is a PlanetLab Central (PLC) portable installation system. PLC manages PlanetLab nodes and handles user registration, slice creation, and site management through a web interface.

References

- [1] <http://prisms.cs.umass.edu/dome/>.
- [2] <http://www.vmware.com>.
- [3] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield. Xen and the art of virtualization. In *SOSP '03: Proceedings of the nineteenth ACM symposium on Operating systems principles*, pages 164–177, New York, NY, USA, 2003. ACM Press.
- [4] A. Bavier, M. Bowman, B. Chun, D. Culler, S. Karlin, S. Muir, L. Peterson, T. Roscoe, T. Spalink, and M. Wawrzoniak. Operating system support for planetary-scale network services. In *NSDI'04: Proceedings of the 1st conference on Symposium on Networked Systems Design and Implementation*, Berkeley, CA, USA, 2004. USENIX Association.
- [5] R. Canonico, G. Ventre, and S. D'Antonio. In *Workshop on Real Overlays And Distributed Systems (ROADS)*, 2007.
- [6] D. Johnson, T. Stack, R. Fish, D. M. Flickinger, L. Stoller, R. Ricci, and J. Lepreau. Mobile emulab: A robotic wireless and sensor network testbed. In *INFOCOM*, 2006.
- [7] C. Perkins. IP Mobility Support for IPv4. RFC 3344 (Proposed Standard), Aug. 2002. Updated by RFC 4721.
- [8] P. Vixie, S. Thomson, Y. Rekhter, and J. Bound. Dynamic Updates in the Domain Name System (DNS UPDATE). RFC 2136 (Proposed Standard), Apr. 1997. Updated by RFCs 3007, 4035, 4033, 4034.
- [9] 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 *Proc. of the Fifth Symposium on Operating Systems Design and Implementation*, pages 255–270, Boston, MA, Dec. 2002. USENIX Association.