

# Integrated Mobile Network Systems for Ubiquitous Network Connectivity

Michael Golightly and Lichun Bao

*Department of Information and Computer Science*

*University of California, Irvine*

*California Alliance for Minority Participation (CAMP) Summer Scholars*

*Emails: {mgolight, lbao}@uci.edu*

## Abstract

With the advent of various mobile devices with diverse communication capabilities, the interoperability between these networks becomes imperative in order to provide ubiquitous network connectivity with the highest throughput in mobile environments. We focus on the interoperability experiments between WiFi computer networks and Cingular GSM/GPRS networks by utilizing these two networks as the transit networks between mobile hosts and the Internet under particular quality of service (QoS) requirements. The experiments were to test the performance attainable in such hybrid networks by configuring a client host with both 802.11 and GPRS wireless network interfaces to exchange files with another Internet host through TCP/IP, then measuring the network throughput and the latency taken to re-initiate a TCP/IP session when switching between the two interfaces. Our results show that it took little time to switch from the low bandwidth GPRS connection to the high bandwidth 802.11 connection with dramatic network throughput gain, but took a long time to switch the other way around.

## 1 Introduction

Multi-hop ad hoc and cellular integrated networks, such as UCAN (Unified Cellular and Ad Hoc Network) [15], have been developed to harness the power of the emergence of third generation (3G) cellular technologies and easily deployed, cheap 802.11 ad hoc networks. The goal behind this work is to provide constant network connectivities to the ongoing data communication sessions on the mobile client hosts by leveraging the TCP/IP protocols support in both types of networks. The idea was to allow clients to use the ad hoc networks based on 802.11 standard to route traffic to and from the best available network access points, which can be either cellular or 802.11 gateways. Therefore, such networks feature the

“always on” networking capability through cellular network connectivity, while opportunistically taking advantage of the high throughput offered by the short range 802.11 networks. The development of these integrated networks has pivotal impact in supporting mobile applications on devices such as PDAs, cell phones, and laptops.

In solving the interoperability problems in the hybrid networks, user mobility poses the most significant challenge, which causes routing control overheads in the transit ad hoc networks, and communication disruptions to the end-to-end applications. Thus, it is essential for us to develop efficient routing protocols to provide the best available network connectivities and to quickly adapt to the network topology changes. Routing in ad hoc networks has been a long standing problem, and there are many solutions in this context. We provide a brief survey about these protocols, and choose one of them in our experiments.

Once deciding an effective routing protocol, our testing of the heterogeneous network interoperability rests on the toolkits for the experiments. In networking research, simulations have been a very effective tool for protocol correctness testings as well as performance benchmarking that take into account many variables such as signal propagation, user mobility, distance between nodes, and many others. Simulators such as NS2 [5] and its extensions [11] have been developed.

However, it was realized that simulations fail to capture many unexpected variables that could occur in real world tests [13]. In the 2000 - 2005 proceedings of the ACM International Symposium on Mobile ad hoc Networking and Computing (MobiHoc), they found that 75% of the papers published used simulations to test their research proposals, with 30% not identifying the simulators used, and that up to 90% of the published simulation results may include bias.

Therefore, in recent years, real world testbeds based on WiFi networks have been widely deployed, such as

the Ad Hoc City project in Seattle, Washington [9], which built a city-wide ad hoc network testbed using fleet vehicles and buses. Unfortunately, many of such testbeds did not incorporate cellular capabilities, which we tried to address in this paper.

The rest of the paper is organized as follows. In Section 2, we review the background and related work in multi-hop ad hoc networks, then provide our testing architecture in Section 3. We discuss the challenges encountered along with their solutions in Section 4, and evaluate the system performance in Section 5. Section 6 concludes the paper. This work provides insights for our future research, such as the potential use of integrated mobile networks for emerging VoIP (Voice over IP) applications.

## 2 Background and Related Work

### 2.1 The Unified Cellular and Ad Hoc Network (UCAN) Architecture



Figure 1: UCAN Architecture

Fig. 1 exemplifies the typical hybrid network deployment as used by the UCAN Architecture [15], which consists of cellular base stations, 802.11 access points, and mobile clients in the form of cellphones, PDAs, and laptops, all with cellular and 802.11 capabilities. In the ad hoc part of the hybrid networks, UCAN allowed a client to find out the best network connectivity in the network while performing the route discovery. After obtaining such route information, a client host can choose which next-hop peers to use in order to route its traffic through to the Internet.

UCAN simulations tested the performance of their algorithm, provides a "quantitative study of raw network capacity, end-to-end throughput, end-to-end delay, power consumption, per-node fairness (for throughput, delay, and power), and impact of mobility on the network performance" for cellular and wireless multi-hop

networks [8].

However, because UCAN requires changing the routing tables on the cellular base stations, which is a risky endeavor for cellular providers, the UCAN Architecture has only been tested using the NS2 simulator. Because of this limitation, one of the goals of our work is to derive some realistic performance numbers in a hybrid network system using real-world testbeds.

### 2.2 Ad Hoc Routing Protocols: AODV and DSR

Routing in a wireless ad hoc environment poses unique challenges compared to typical wired environments, and therefore requires an industrious protocol to locate mobile clients and keep routes fresh. The two benchmark routing protocols in this area are the Dynamic Source Routing Protocol (DSR) [10] and the Ad Hoc On-Demand Distance Vector Routing Protocol (AODV) [19]. Both routing protocols are to establish routes only when needed, thereby reducing the network load for routing control purposes. Both routing protocols involve two phases of routing operations: route discovery and route maintenance. Their main difference is that DSR is based on the source-route whereas AODV is distance-vector based. Further details of their differences and other ad hoc routing protocols are found in [3] and [18].

## 3 Experiments Architecture

As show in Fig. 2, we focus on the impact of switching between heterogeneous transit networks on the throughput and latency of ongoing communication sessions, particularly the TCP/IP file transfer applications. For this purpose, our client is a Dell XPS Gen 2 laptop with 802.11 connectivity provided by an integrated Intel Pro Wireless 2200BG network card, and a Bluetooth connectivity to a Cingular Slvr L7 GPRS capable cell phone provided with full service.

For interoperability testing, we adopted a software solution, called Hamachi [7] to create a Virtual Private Network (VPN) as so to provide a consistent IP address for each client host. Hamachi overcomes the disruption of network services due to network attachment changes by building a Virtual LAN among the network participants. Each participating host is assigned with a single static IP that can be shared across different network adapters, and connects with each other through the Hamachi secure server, which solves the problems associated with Network Address Translation (NAT). Reportedly, Hamachi can successful establish VPNs in 95% of cases. Unfortunately, it was unable to establish a connection through the Cingular GPRS interface in our experiments. Although

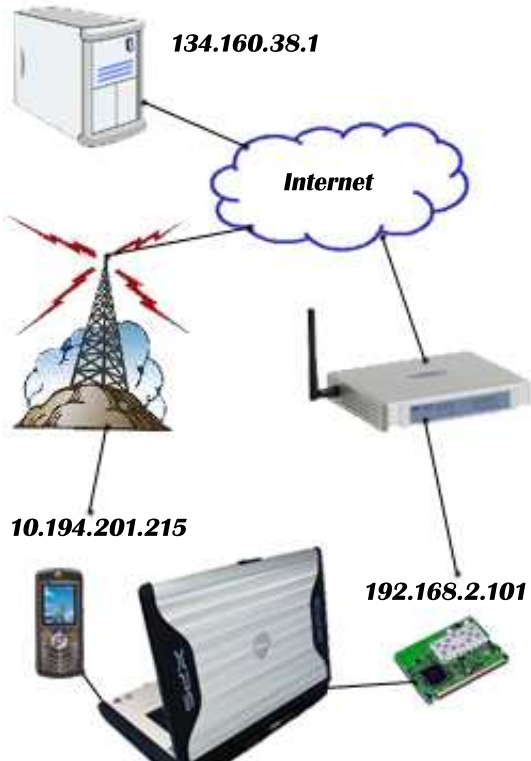


Figure 2: Our Test Architecture

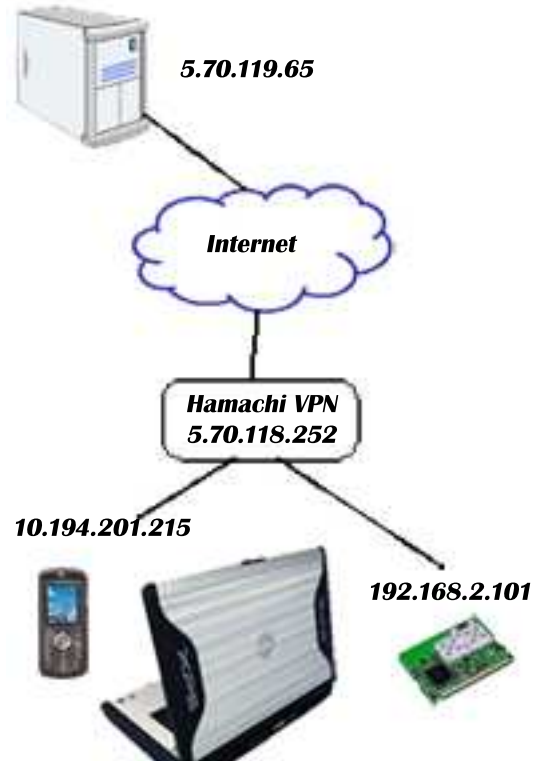


Figure 3: Hamachi VPN

Hamachi solution is still under our investigation, we post the Hamachi solution here for future extensions.

For network throughput testing purposes, we used speed test services offered by Speakeasy.net [21] to measure throughput. Although speakeasy.net might not be able to determine the maximum throughput of each device, it is still effective to provide the ratio of the throughputs between different network connectivities.

For network traffic monitoring purposes, we used Ethereal [4] to capture the packets going through each active network interface. When measuring the time taken to re-establish a TCP/IP session between adapters, this tool allowed us to see at what time the file server was contacted after a device was initialized.

#### 4 Testing Challenges and Trade-offs

Our first goal was to create a full fledged ad hoc network with multiple clients and access points using the AODV routing protocol. However, due to the difficulties we encountered, we have concluded with a simplified testbed that includes only a single client in the end. We argue that such experimental evaluations of the hybrid network achieve our interoperability testing goals, although in a simpler setup.

#### 4.1 Broadcom BCM94306MP

One of our test laptops, a HP Pavilion ze4800, came standard with a Broadcom BCM94306MP 802.11g wireless card. There are no official Linux drivers for this card, but the latest release of Fedora Core 5 includes built-in beta drivers for the wireless card. An alternative approach is to use a tool called Ndiswrapper [16] which "wraps" a Windows driver to Linux through a loadable kernel module. We tried both solutions and they only worked with certain combinations of routers and clients.

#### 4.2 USR2450 and LinuxAP

The US Robotics 2450 wireless access point was first released to market several years ago. It is based off the Eumitcom board with a PRISM based 802.11b PCMCIA 802.11 card, a serial interface, and 1 wired Ethernet port. Such a configuration allows installing custom firmware so as to run a Linux 2.4 kernel on the access point. The first major firmware distribution based on Linux 2.4 was OpenAP developed by [17], which was taken over by Keith Smith with the LinuxAP project [20] later.

LinuxAP enables a plethora of options to a network administrator such as custom routing tables, network bridging, and loadable Linux network modules, but be-

cause of the many "hacks" necessary for correct operation and the risk of "bricking" the wireless access point with unsuccessful flashes, the software modifications based on LinuxAP have consumed overwhelming amount of work in the project.

Initially, we tried to use LinuxAP-eh, LinuxAP enhanced, from [14]. This distribution was last updated February 21, 2004, and included the additional 2.1 NIST Implementation of AODV [12]. The newest version of NIST AODV is 2.2, so the LinuxAP-eh was one generation out of date. Even with such version discrepancy, running LinuxAP-eh with AODV consistently generated invalid wireless network events that crashed the router.

The last option to create a successful LinuxAP kernel on the USR2450 access point by creating custom build scripts to generate a compatible flash image to the device. The difficulty lies in the fact that the flash image requires all necessary software packages to comply with Linux kernel 2.4 and the hardware configurations. Any mistake or mismatch in software packages could render the re-flashed USR2450 platform "bricked" or non-functional.

Because of these difficulties, we have decided to go along with simpler hybrid network setup for the current phase of the project.

### 4.3 AODV Implementations

AODV is a very adaptive routing protocol that can be adjusted according to the protocol capability and the deployment scenarios. Most AODV versions have been built on Linux kernel 2.4, except for AODV-UU [2], which is reported to work on kernel version 2.6.11.

For Windows OS, UoBWinAODV [22] is written in C, links with the NDIS driver, and provides a graphical user interface for the user to monitor network connectivities and traffic. An alternative is to use a Java implementation, available from the same site, which can be run on any machine with a Java Virtual Machine.

In our experiments, we had various implementations to choose from, and we are more concerned about our access point platform which runs the NIST 2.1 AODV Kernel module.

## 5 Performance Evaluation

Using the simplistic experimental setup with one laptop, we measured throughput of the two adapters and the time taken to switch between adapters during a TCP/IP download session.

### 5.1 Throughput

The throughput testing results are given in Table 1. Using the services provide through

	Throughput (kpbs)	
	Downstream	Upstream
Cellular	43	18
802.11g	1379	495

Table 1: 802.11g and GPRS Throughput

[www.speakeasy.net/speedtest](http://www.speakeasy.net/speedtest), we chose the test server at the same remote location in Los Angeles, CA, to simulate a real world session between the client and a server. Due to the expenses of cellular bandwidth, which cost one cent per kilobyte through Cingular GPRS, there was not much room to experiment over a long period of time, or with sufficient scenario sets for trials and errors. Therefore, the accuracy of the test is crude at this stage. The speakeasy speedtest was the best alternative to measure throughput without wastefully expending cellular bandwidth.

GPRS is a 2.5 generation cellular wireless technology and has a theoretical limit of 171.2 kbps when it can use all possible data channels, but a more realistic rate is 30-80 kbps [6]. 802.11g on the other hand has a theoretical limit of 54Mbps, but a more realistic rate of 25Mbps [1]. Aside from these two facts, one has to take into account the limit of all links between client and server because the slowest will be the ultimate bottleneck.

In the end, we can see that the addition of a 802.11 network adapter to a cellular only device can greatly enhance its network performance by up to a factor of 30.

### 5.2 Adapter Switching

The measurement in our experiments was to quantify the latency for switching between heterogeneous networking adapters during a TCP/IP session.

To conduct this test, we started a FTP transfer session to a remote client with one adapter, initialized the other adapter, and measured the time taken to re-establish the TCP/IP session by monitoring traffic through Ethereal. It is interesting to note that when we initialized our GPRS connection, our 802.11 connection to the FTP server was immediately severed. Our 802.11 connection repeatedly sent ACKs for the packets it received, but never received a response from the FTP server.

From the packets moving through the GPRS interface monitored by Ethereal in Table 2, we can see that it takes GPRS about 5.59s to establish a connection with the local cellular service. There is a long dead period of about 88s during which no activity is detected before GPRS initiates a re-connect with the FTP server. This is most likely caused by the way Windows XP reacts to the addition and removal of network adapters. At the same time, we see that it takes about 2.3s for GPRS to initiate con-

Time(s)	Source	Destination	Protocol
0	Send_08	Send_08	PPP
0.1875	Receive_08	Receive_08	PPP
0.1875	Send_08	Send_08	PPP
0.1875	Send_08	Send_08	PPP
0.359375	Receive_08	Receive_08	PPP
0.359375	Send_08	Send_08	PPP
0.359375	Send_08	Send_08	PPP
0.359375	Send_08	Send_08	PPP
0.53125	Receive_08	Receive_08	PPP
0.53125	Send_08	Send_08	PPP
0.53125	Send_08	Send_08	PPP
0.71875	Receive_08	Receive_08	PPP
0.71875	Send_08	Send_08	PPP
2.015625	Send_08	Send_08	PPP
2.1875	Receive_08	Receive_08	PPP
2.1875	Send_08	Send_08	PPP
2.359375	Receive_08	Receive_08	PPP
2.359375	Send_08	Send_08	PPP
2.65625	10.194.201.215	255.255.255.255	DHCP
5.59375	10.194.201.215	255.255.255.255	DHCP
93.84375	10.194.201.215	134.160.38.1	TCP
94.890625	134.160.38.1	10.194.201.215	TCP
94.890625	10.194.201.215	134.160.38.1	TCP
94.890625	10.194.201.215	134.160.38.1	HTTP
96.140625	134.160.38.1	10.194.201.215	TCP

Table 2: 802.11 Connection Switching to Cingular GSM/GPRS Connection

*\*Note that during the idle period from 6s to 93s , the 802.11 adapter sent ACKS for the packets it received, but received no response from the FTP server.*

tact with the FTP server and establish file transfer compared to the .2s of our 802.11 test.

From the packets moving through the 802.11 interface monitored by Ethereal in Table 3, we can see it takes about 15 seconds to establish a connection with the local router. The data is gathered with the 802.11 adapter configured with a static IP for simplicity. Tests were also conducted with DHCP in which the latency averaged with 1 second of difference from that of the static IP setup. It is interesting to note that the GPRS device actually established a connection with its gateway faster than that of 802.11. This is most likely due to features of the NetBios Name Server Protocol, which is used to configure the Windows XP workgroup and domain.

### 5.3 Discussions

Mobile devices are increasingly incorporating the speed of 802.11 and the versatility of cellular wireless technologies to deliver high performance mobile applications. With the above data, one can gauge the performance they might see in VoIP, video conferencing, or other mobile technologies. In regards to VoIP and

video conferencing, we always want the highest throughput connection with the best quality of service, and the lowest switching time between networks for continuous communication, thus allowing for the best use of network resources.

It is of our great interests to develop a VoIP application in tandem with the normal cellular voice communication, and to switch between these two types of voice communication services under the constraints of costs, throughput and latency. Such a research direction is not only theoretically compelling, but also economically attractive to the customers.

## 6 Conclusions

We have discussed several interesting challenges in future generation networks, the difficulties of setting up a multi-hop ad hoc 802.11 and cellular network, and carried out very intriguing experiments to derive the performance limits provided by the existing networking technologies, particularly switching latencies between networking interfaces and network throughput. We enu-

Time(s)	Source	Destination	Protocol
0	IntelCor_22:31:65	Broadcast	ARP
1.078281	192.168.2.132	192.168.2.255	NBNS
1.828219	192.168.2.132	192.168.2.255	NBNS
2.578254	192.168.2.132	192.168.2.255	NBNS
3.328302	192.168.2.132	192.168.2.255	NBNS
4.078672	192.168.2.132	192.168.2.255	NBNS
4.828383	192.168.2.132	192.168.2.255	NBNS
5.578421	192.168.2.132	192.168.2.255	NBNS
6.328462	192.168.2.132	192.168.2.255	NBNS
14.816363	IntelCor_22:31:65	Broadcast	ARP
14.817905	SmcNetwo_0a:29:88	IntelCor_22:31:65	ARP
14.81791	192.168.2.132	134.160.38.1	TCP
14.950414	134.160.38.1	192.168.2.132	TCP
14.950447	192.168.2.132	134.160.38.1	TCP
14.950872	192.168.2.132	134.160.38.1	HTTP
15.085513	134.160.38.1	192.168.2.132	TCP

Table 3: Cingular GSM/GPRS Connection Switching to 802.11 Connection

merated many roadblocks to setting up our test including driver support, outdated hardware and software, limited documentation, and non-configurable private cellular connections. It is encouraging to realize the integration of data networks based on 802.11 and the cellular networks in order to provide the ubiquitous connectivity for mobile devices through our experiments.

## Acknowledgments

Prof. Lichun Bao has been the mentor of this research, and supplied the USR2450 802.11 router for the duration of the summer. The California Alliance for Minority Participation (CAMP) Summer Scholars Program sponsored this summer research.

## References

- [1] 802.11. en.wikipedia.org/wiki/802.11, 2006.
- [2] AODV-UU. core.it.uu.se/core/index.php/aodv-uu, 2006.
- [3] BROCH, J., MALTZ, D. A., JOHNSON, D. B., HU, Y.-C., AND JETCHEVA, J. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *Mobile Computing and Networking* (1998), pp. 85–97.
- [4] ETHEREAL. ethereal.cc, 2006.
- [5] FALL, K. Network emulation in the VINT/NS simulator. In *Proceedings of the fourth IEEE Symposium on Computers and Communications* (1999).
- [6] GPRS. en.wikipedia.org/wiki/gprs, 2006.
- [7] HAMACHI. Applied networking inc., hamachi.cc, 2006.
- [8] HSIEH, H.-Y., AND SIVAKUMAR, R. Performance comparison of cellular and multi-hop wireless networks: a quantitative study. In *SIGMETRICS/Performance* (2001), pp. 113–122.
- [9] JETCHEVA, J. G., HU, Y.-C., PALCHAUDHURI, S., SAHA, A. K., AND JOHNSON, D. B. Design and evaluation of a metropolitan area multitier wireless ad hoc network architecture. *wmsa 00* (2003), 32.
- [10] JOHNSON, D., MALTZ, D., AND BROCH, J. *DSR The Dynamic Source Routing Protocol for Multihop Wireless Ad Hoc Networks*. Addison-Wesley, 2001, ch. 5, pp. 139–172.
- [11] KE, Q., MALTZ, D., AND JOHNSON, D. B. Emulation of multi-hop wireless ad hoc networks. In *The 7th International Workshop on Mobile Multimedia Communications (MoMuC 2000)* (2000).
- [12] KERNEL-AODV. National institute of standards and technology, wireless communication technologies group, kernel aodv, w3.antd.nist.gov/wctg/aodv\_kernel/, 2006.
- [13] KURKOWSKI, S., CAMP, T., AND COLAGROSSO, M. MANET simulation studies: The incredibles. *ACM's Mobile Computing and Communications Review* 9, 4 (2005), 50–61.
- [14] LINUXAP-EH. olotwireless.net/catala/linuxap.htm, 2006.
- [15] LUO, H., RAMJEE, R., SINHA, P., LI, L., AND LU, S. Ucan: a unified cellular and ad-hoc network architecture, 2003.
- [16] NDISWRAPPER. ndiswrapper.sourceforge.net/, 2006.
- [17] OPENAP. devicescape.com/opensource/, 2006.
- [18] PERKINS. Performance comparison of two on-demand routing protocols for ad hoc networks, 2001.
- [19] PERKINS, C. Ad hoc on demand distance vector (aodv) routing, 1997.
- [20] SMITH, K. Linuxap: openap.ksmith.com, 2006.
- [21] SPEAKEASY. speakeasy.net/speedtest, 2006.
- [22] UOBWINAODV. aodv.org, 2006.