Changing Communication Environments in MosquitoNet

Mary G. Baker
Computer Science and Electrical Engineering Departments - Stanford University
Stanford, CA 94305-2140

Abstract

MosquitoNet is a new project at Stanford University for investigating operating system and application issues in mobile and wireless computing. Our initial goals focus on providing the appearance of continuous network connectivity for mobile hosts. To achieve these goals we must first enable portable computers to move seamlessly from one communications medium to another, for example from an Ethernet connection to a wireless modem, without rebooting or restarting applications. Second, we must determine how to manage the resulting dynamic changes in network characteristics, either transparently or through a simple interface between the network and the applications. Other goals include developing power management policies for network devices and experimenting with file data consistency algorithms for networks in which disconnection may be economically desirable, but seamless network re-connection is almost always possible.

1 Introduction

Lighter-weight, more powerful sub-notebook computers, combined with the availability of low-cost, higher-bandwidth wireless communications services have vastly increased the potential of mobile and wireless computing. These technological advances make it attractive to build networks that include both traditional workstations and portable computers of equal standing. Sub-notebook computers weighing well under three pounds have crossed the threshold that makes them easily portable for the majority of users. As these machines get lighter, they require less conscious effort for users to carry them almost anywhere. They can then be treated as commodity items, such as pagers or cellular phones, whose connection status should not require much attention. In addition, wireless service providers are quickly making Internet connection possible over large geographical areas [33][35]. These advances together should greatly increase the mobility of networked portable computers.

Unfortunately, portables cannot be equivalent to stationary hosts unless they too can remain connected to the network, and there are still many problems that make this difficult. One of these problems is the ability to switch between different communications media while maintaining all current network conversations. Making such switches as seamless as possible is required in order to maintain a nearly continuous connection while choosing the highest performance communication device available at a given time or place.

Addressing this and other problems that prevent the appearance of continuous network connectivity is the primary goal of MosquitoNet, a new project at Stanford University for investigating operating system and application issues in mobile and wireless computing on the Internet. As part of this project we are also studying how best to deal dynamically with changes in network characteristics such as bandwidth, latency, connectivity and connection price structure. We are investigating the trade-offs between making some of these changes transparent to the applications by handling them in the system, versus exposing the changes to applications using a simple interface. We will also explore power management policies for network devices and data consistency algorithms for networks in which disconnection may occur, but reconnection is almost always possible.

2 Connectivity Problems for Mobile Hosts

There are at least four problems aside from battery lifetime that currently prevent continuous network connection: lack-of-service disconnections, economic disconnections, disconnections due to switching between communications media, and intermittent disconnections due to poor channel quality. As described below, we believe the first two types of disconnection will improve over time, but the last two will not. Thus we propose to address the last two types in MosquitoNet.

The MosquitoNet project is funded in part by the Center for Telecommunications at Stanford University, the Reid and Polly Anderson Faculty Scholar Fund, and the Xerox Palo Alto Research Center.

0-8186-6345-6/95 $04.00 © 1995 IEEE

Lack-of-service disconnections occur when there is no available communications media. For several years it will still be possible to leave wireless service areas, as when heading to remote areas to do fieldwork. Disconnection is inevitable under these circumstances. Economic disconnections occur when users choose to disconnect to save money, for example to avoid connect-time-based access charges. Over time, though, these two conditions should improve for three reasons: satellite cellular telephones and self-contained receivers are becoming smaller and more affordable, service providers will cover larger areas, and more service providers will offer fees that are not based on connection time [15][33].

The third type of disconnection, due to switching between communications media, results from attempting to get the highest possible network performance. For example, while in his or her office, a user may have an Ethernet connection available for a portable computer and would prefer to use this path for the higher bandwidth it provides. But when he leaves the office and unhooks the machine from the Ethernet, he would like it to switch seamlessly to whichever other communication device is available and provides the best service at that time - a wireless radio modem perhaps. It may even be the case that two or more communication devices are available and appropriate simultaneously, perhaps for a single application's use. Currently such seamless switches are not possible, since the user must shut down the applications and sometimes even reconfigure the system to use the new connection. Solving this problem is therefore one of MosquitoNet's goals, as described in the following section.

Finally, some communication channels may provide only weak connectivity. For example, the high latency and burstiness of some radio links leads to connections that do not appear to be continuous to most software or users. This condition may get better as wireless service providers improve their services, but it can also get worse as radio cells get more crowded. MosquitoNet will also address this last type of disconnection through software techniques, as described in Section 4.

3 Providing Seamless Connectivity

Enabling transparent switches between communication devices requires protecting the applications and upper layers of the TCP/IP [23] protocol stack from these changes. Each communication device, or network interface, has its own IP address. Thus the IP address of the source or destination changes whenever the source or destination host switches to a new network interface. The transport and application layers should be protected from such changes in addressing, but the transport layer needs to know the IP address of the destination with which it is corresponding. To make address changes transparent to the transport layer and above, we must modify the network (IP) layer so that it allows the layers above to continue to use static IP addresses, even if the network interfaces and actual IP addresses of the source or destinations change. This new adaptable IP layer must thus manage these address changes with some new and modified tasks:

- **Keeping track of destination host interfaces -** The adaptable network layer must be able to choose the correct interface for sending out packets according to the interface in use by the destination address (or the first hop router). This layer will therefore need access to state information about the available interfaces in use at other hosts as those interfaces change.
- **Keeping track of local interfaces -** When an interface on the local host is shut down, the adaptable network layer will need to figure out which new one can be used instead. Thus this layer will need to maintain and update state information about available local interfaces.

There are several potential problems with this scheme. One is how to detect or announce network address changes for the routing layer. A second problem is the added latency incurred when a communication device changes on a host and packets are lost or delayed while the new interface is put into use. Some of these problems are similar to those already encountered in work on mobile IP routing protocols [9][16][20][29]. Some of these problems are therefore amenable to similar solutions, including the use of outside agents for monitoring and managing connection state information and for streamlining routing changes. Finally, there are other unsolved problems, such as choosing the best communication device when more than one is available. For example, given a choice between a satellite connection and an Ethernet connection, we may be faced with choosing between high bandwidth with high latency versus low latency with lesser bandwidth. The best choice of interface depends on the application's behavior, and different applications may wish to use different devices concurrently. It can even be advantageous for one application to make use of two network interfaces simultaneously.

4 Mobile Awareness for Operating Systems and Applications

Solving the connectivity problems described in Section 2 will require more than merely making it possible to swap physical devices on-the-fly. Even where such device switching is possible, various communications media have such different properties that some applications will not perform correctly without further work. For example, applications that transfer large images over a network connection for a graphical user interface may perform well over an Ethernet connection, but they will not perform as well over a phone line or a low-bandwidth infrared link. Applications with certain retransmission policies (TCP) will perform poorly on a channel with high latency variance even if the channel has reliable packet delivery [3]. Other applications may not perform well with only intermittent connectivity. They may assume a link is down and abandon the connection even if the link is only temporarily unusable. Finally, the pricing structure for various communications media may differ by vendor. Services that charge by connect time require different approaches than those that accept a flat rate for however much connection time is used.

MosquitoNet will investigate how best to manage these changing connection properties. The most attractive approach initially is to shield applications from the changes by handling them in the system or a system library layer. For example, if one is charged for connect time, one would like applications to connect only when necessary. However, it may not be reasonable to expect applications such as a distributed file system to worry about connecting and reconnecting as necessary. Instead, a library interface between the application and the network could connect only when the application finds it must transfer data. The library can maintain state that makes the connection appear continuous to its applications. Making these changes transparent to applications means that application designers need not concern themselves with this issue.

Unfortunately, we do not believe that the transparent approach will provide sufficient performance for all classes of applications. A second approach provides a simple interface that exposes communication changes to applications that require some knowledge of them. Many applications may find it sufficient if they can query or receive information from the system about only a few types of changes: latency, throughput, variance in latency and throughput, connectivity, and connection price structure. For example, a Mosaic-like application could automatically cease transferring image data if its connection switches to too low a bandwidth. Some previous work in this area concentrates on providing a more general-purpose RPC mechanism for specifying any dynamically changing environment variables, especially for applications that are mobile in the sense that they may migrate from one machine or display to another [26][31]. MosquitoNet will focus at first on the simpler problem of conveying only information about changing network resources. This approach will not work without modification of some applications, as it assumes that applications can respond to the changes in their communication environments.

The third approach places most of the burden on applications rather than the system. This involves structuring applications so they are as undemanding as possible with regard to their network connections. For example, many projects have separated some part of the graphical user interface from the rest of the application [10][21][22][28][34]. Most of the interface processing is done on the mobile host, so that less data is transferred over the network. When data must be transferred between the separated portions of applications, the applications can use a communication protocol designed for low-bandwidth. While this approach has the potential to handle weak connections more effectively than the other approaches, it also means that the applications may not perform to their full potential when higher performance connections are available.

Finally, a related problem is to make it possible to move various parts of applications dynamically to the location where their connection, data, or processing requirements are best satisfied. For example, an application that requires a lot of processing power may provide better performance if it is allowed to migrate to a high-performance processor. Thus we may want a graphical user interface to run on the portable, while the processing engine of the application runs on a CPU server. But if a large amount of data must be transferred, it may prove beneficial to move the processing engine to the portable temporarily so that it and the graphical interface need not communicate over an off-machine link. While this approach is more complex, it allows for applications that are more adaptable to their network environments [26]. Solving this last problem also requires a standard environment that the application can understand as it moves various pieces of code or scripts around to different platforms [5][19][21].

5 Power Management for Network Devices

Another problem that currently restricts host mobility is a physical one: battery lifetimes for most portables are still limited to just a few hours. More continuous network connection will only aggravate this problem since communication devices consume energy. (Use of a PCMCIA Ethernet card on a Gateway 2000 Handbook, for example, seems to reduce the battery lifetime by almost half.) As part of the MosquitoNet project we would like to develop intelligent policies for reducing the use of communication devices while still providing the appearance of continuous connection.

Turning off communication devices while a host is not transmitting (or receiving responses from its transmissions) does not necessarily limit its effective connectivity. For most client workstations, applications will function as if continuously connected as long as they are able to initiate communication whenever desired. Unless a computer functions as a server, for instance by providing on-line file archives or a World-Wide-Web site, it may not be necessary for it to remain continuously accessible to outside machines. For example, the portable need not be able to receive electronic mail continuously, because mail may be delivered to another host that the portable periodically polls. Although it is not currently possible to "power-down" many communication devices, this situation will change if the energy savings prove to be significant. We will examine usage patterns in our mobile computing testbed to help develop these power management policies.

6 Data Consistency Algorithms

Besides battery life problems, there are other issues related to providing the appearance of continuous connection without actually requiring it. One of these issues is file and data consistency. Files and data stored remotely may be cached or replicated on portable computers to increase performance and reduce necessary communication over the network. Mechanisms for monitoring consistency problems between the local and remote data are highly desirable.

Much of the work on data consistency for mobile hosts assumes that there will be significant periods of mandatory disconnection from the network [1][7][13][24][30]. When a portable reconnects to the network, the user may find that conflicting updates have been made to remote copies of files he has edited locally. Ensuring consistency under these circumstances is usually considered impractical, since it may prevent others from modifying or perhaps even accessing remote copies of data while it is on a mobile host. Instead, these techniques tend to detect inconsistencies and to attempt to handle them once they have occurred. This approach trades data consistency for data availability.

In contrast, our goal in MosquitoNet is to create an environment in which seamless re-connection is possible whenever desired. If re-connection is possible, then a mobile host has the option of checking and updating consistency information when necessary. This should allow us to develop algorithms that provide improved levels of consistency while still permitting higher availability. However, the burden is on the mobile host to manage the consistency problems, since servers will not be able to contact the mobile host with consistency-related callbacks when the mobile host is not actually connected [17].

7 Project Platform and Status

MosquitoNet is a brand new project, and thus its status is very preliminary. We have spent most of our time so far investigating the areas we wish to address and equipping and testing our experimental platform.

As a prototype of what we believe future systems will look like, we are aiming for light-weight but fully functional computers equipped with several possible communications devices, including a relatively high-bandwidth wireless network. We have chosen to install the Linux operating system [14] on the Gateway Handbook. We chose Linux for several reasons: 1) it supports dynamically loadable kernel modules and "hot swap" of devices, an important feature to us since we will be switching communication devices while on-line, 2) it includes drivers for the PCMCIA cards that we are using, 3) it is small and performs well on portable computers, and 4) it is free and there are no source code restrictions. This gives us a familiar UNIX and X Windows [6] environment in under 3 pounds.

The other technological advance in our geographic area is the micro-cellular data network service, Ricochet, that Metricom is installing around the Stanford campus and in cities nearby [4]. Metricom's network costs only $30 a month for at least a 50 kilobit-per-second wireless link. The flat fee per month means there is no reason to avoid using the service once it has been purchased. Since Metricom is blanketing the Stanford campus and local communities with pole-top transceivers, some of which provide wired access points to BARRNet, we will soon have reasonably wide-spread wireless access to the Internet in our area.

Metricom's radio modems give us one communication device, but since MosquitoNet must address switching between devices, we have also equipped our setup with PCMCIA Ethernet cards and PCMCIA modems.

8 Conclusions

In summary, the issues we plan to address are 1) switching seamlessly between different communications devices, 2) shielding applications, where possible, from the resulting changes in latency, throughput, latency and throughput variance, connectivity, and connection price structure, 3) developing a good interface for conveying these changes to applications that must adapt to them and for allowing applications to determine what network services are available, 4) restructuring applications so they can easily handle connection changes, 5) migrating portions of applications as appropriate considering their connection, data, and processing requirements, 6) automatically choosing the best communication devices available for an application at a given time and place, 7) developing communication device power management policies, and 8) experimenting with file consistency techniques for use when network re-connection is almost always possible.

Our goal is to provide seamless mobility for sub-notebook sized computers based upon the UNIX operating system and Internet protocols. New wireless radio service providers in our area and Linux on a sub-notebook computer make it possible to start the project using mostly off-the-shelf hardware and software.

9 Acknowledgments

We gratefully acknowledge mobile computing discussions with Doug Terry, Mark Weiser, David Cheriton, Steve Deering, Owen Densmore, Behfar Razavi, Arthur Keller, Barry Leiner, M. Satyanarayanan, and Dan Siewiorek. We thank Alan Demers, Mendel Rosenblum, Jonathan Stone, Stuart Cheshire, Xinhua Zhao, and Wendell Baker for their comments on this paper, and also Vaughan Pratt for his help with Linux on the Gateway. We also thank Metricom for their generosity in providing us with lots of radio modems and technical support.

10 References

[1] Daniel Barbara and Hector Garcia-Molina, "Replicated Data Management in Mobile Environments: Anything New Under the Sun?" *Proceedings of the IFIP Conference on Applications in Parallel and Distributed Computing*, April 1994.

[2] Trevor Blackwell et al., "Secure Short-Cut Routing for Mobile IP." *Proceedings of the 1994 Summer USENIX Conference*, June 1994.

[3] R. Caceres and L. Iftode, "The Effects of Mobility on Reliable Transport Protocols." *Proceedings of the 14th International Conference on Distributed Computing Systems*, IEEE Computer Society, June 1994.

[4] George Flammer, Metricom, Personal communication, October, 1994.

[5] General Magic Information Binder, General Magic Inc., 1994.

[6] R. Scheifler and J. Gettys, "The X Window System." *ACM Transactions on Graphics*, volume 5, number 2, April 1986.

[7] Larry B. Huston and Peter Honeyman, "Disconnected Operation for AFS." *Proceedings of the USENIX Mobile and Location-Independent Computing Symposium*, August 1993.

[8] John Ioannidis, Dan Duchamp, and Gerald Q. Maguire Jr., "IP-based Protocols for Mobile Internetworking." *Proceedings of the SIGCOMM '91 Conference: Communications Architectures & Protocols*, September 1991.

[9] David B. Johnson, "Scalable and Robust Internetwork Routing for Mobile Hosts." *Proceedings of the 14th International Conference on Distributed Computing Systems*, IEEE Computer Society, June 1994.

[10] C. Kantarjiev et al., "Experiences with X in a Wireless Environment." *Proceedings of the USENIX Mobile and Location-Independent Computing Symposium*, August 1993.

[11] R. H. Katz, "Adaptation and Mobility in Wireless Information Systems." *IEEE Personal Communications*, volume 1, number 1, 1994.

[12] Kimberly Keeton et al., "Providing Connection-Oriented Network Services to Mobile Hosts." *Proceedings of the USENIX Mobile and Location-Independent Computing Symposium*, August 1993.

[13] James Kistler and M. Satyanarayanan, "Disconnected Operation in the Coda File System." *Proceedings of the 13th SOSP*, October 1991.

[14] The Linux Journal, P.O. Box 85867, Seattle, Washington, 98145. Editor Phil Hughes, publisher Robert F. Young. volume 1, edition 1, March 1994.

[15] Mackay Communications, Marketing brochure on ABB Nera's Saturn MiniPhone, Summer 1994. Contact address: Raritan Center, 300 Columbus Circle, P.O. Box 7819, Edison, NJ 08818.

[16] Mobile IP Working Group, "Routing Support for Mobile Hosts." *Internet Draft*, Dec. 1993.

[17] L. Mummert and M. Satyanarayanan, "Large Granularity Cache Coherence for Intermittent Connectivity." *Proceedings of the 1994 Summer USENIX Conference*, June 1994.

[18] B. D. Noble and M. Satyanarayanan, "An Empirical Study of a Highly Available File System." *Performance Evaluation Review*, volume 22, number 1, May 1994.

[19] John Ousterhout, Personal Communication, August 1994.

[20] Charles E. Perkins, Andrew Myles, and David B. Johnson, "The Internet Mobile Host Protocol (IMHP)." *Proceedings of INET '94, The Annual Conference of the Internet Society*, June 1994.

[21] Karin Petersen, "Tcl/Tk for a Personal Digital Assistant." *To appear in the Proceedings of the USENIX Symposium on Very High Level Languages*, October 1994.

[22] Rob Pike, "The Blit Programmer's Manual." AT&T Bell Laboratories, Computing Science Technical Report Number 121, 1985.

[23] J. Postel, "Internet Protocol." RFC 791, September 1981.

[24] Peter Reiher et al., "Resolving File Conflicts in the Ficus File System." *Proceedings of the 1994 Summer USENIX Conference*, June 1994.

[25] M. Satyanarayanan et al., "Coda: A Highly Available File System for a Distributed Workstation Environment." *IEEE Transactions on Computers*, April 1990.

[26] B. Schilit, M. Theimer, and B. Welch, "Customizing Mobile Applications." *Proceedings of the USENIX Symposium on Mobile and Location-Independent Computing*, August 1993.

[27] S. Sheng, A. Chandrakasan, and R. W. Brodersen, "A Portable Multimedia Terminal." *IEEE Communications Magazine*, volume 30, number 12, December 1992.

[28] Bernhard Strassl, Electronic posting to comp.os.linux.announce, announcing TIP - a client/server Toolkit Interface Protocol, August 10, 1994.

[29] Fumio Teraoka, Keisuke Uehara, Hideki Sunahara, and Jun Murai, "VIP: A Protocol Providing Host Mobility." *Communications of the ACM*, volume 37, number 8, August 1994.

[30] Douglas Terry, Alan Demers, Karin Petersen, Mike Spreitzer, Marvin Theimer, and Brent Welch, "Session Guarantees for Weakly Consistent Replicated Data." *Proceedings of the Third International Conference on Parallel and Distributed Information Systems*, September 1994.

[31] Marvin Theimer, Alan Demers, and Brent Welch, "Operating System Issues for PDAs." *Proceedings of the Fourth Workshop on Workstation Operating Systems*, October, 1993.

[32] Hiromi Wada, Tatsuya Ohnishi, and Brian March, "Packetforwarding for Mobile Hosts." Internet Draft, Matsushita Electric Industrial Co., Ltd., Japan, November 1992.

[33] "Metricom Network to Increase Size, Speed of Wireless Connection." *Wall Street Journal*, June 14, 1994.

[34] Terri Watson, and Brian Bershad, "Local Area Mobile Computing on Stock Hardware and Mostly Stock Software." *Proceedings of the USENIX Symposium on Mobile and Location-Independent Computing*, August 1993.

[35] Editor, "Hot Stuff." *Wireless*, volume 3, number 3, May/June 1994, pp 6-7.