Secure Mobile Computing
Final Report: 1993-97

Research Documents Group
C. Roy Taylor, Editor
9 September 1997
CMU-CS-97-168

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213-3890

Abstract

Research under the ‘‘Secure Mobile Computing’’ project has developed middleware and system-level solutions that provide access to shared data and that apply computing resources to such data in a manner that preserves system scalability and security. The work has also established connections with potential users of secure, mobile computing technology and produced portable software systems that will support its further development.

This report summarizes work during the 1993-97 contract period and guides the reader into the literature.

This research is sponsored by DARPA/CSTO, through ARPA Order A700, and issued by ESC/ENS under contract F19628-93-C-0193. Views and conclusions contained in this document are those of the authors and should not be interpreted as representing official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or of the United States Government.

Keywords

Mobile computing, application-transparent adaptation, weakly-connected operation, data consistency, I/O latency, application-aware adaptation, mobile network protocol, system security, secure remote execution,

Mobile computing systems, viewed from a software perspective, differ fundamentally even from their distributed, nonmobile kin:

- Mobile elements are *resource-poor* relative to static elements. For a given cost and level of technology, mobile elements will be slower and have less memory and disk space than static elements. Weight, power, and size constraints will continue conspiring to preserve this inequity.
- The *trustworthiness and reliability* of mobile elements is low. A Wall Street stockbroker can lose her laptop through foul play on Manhattan streets much more easily than an intruder could subvert her office workstation.
- Mobile elements must operate effectively under broad range of *connectivity and failure* conditions. While a desktop workstation can typically rely on LAN or WAN connectivity, a laptop in a hotel room may have, at best, modem or ISDN connectivity. Outdoors, a laptop with a cellular modem may find itself temporarily beyond the range of the nearest relay antenna.

In addition, as portable computers become ubiquitous, they will compose increasingly large, dynamic computing systems and raise significant issues of *scalability*. Taken together, these constraints, intrinsic to mobility, go beyond mere artifacts of current technology and present significant technical challenges.

Research under the ‘‘Secure Mobile Computing’’ project has addressed these challenges in several broad thrusts, developing middleware and system-level solutions that provide access to shared data and that apply computing resources to such data in a manner that preserves system scalability and security. Our work has also established connections with potential users of secure, mobile computing technology and produced portable software systems that will support its further development.

This report summarizes work during the 1993-97 contract period and guides the reader into the literature.

1. Adaptivity at the Middleware Level

Ideally, mobility should be completely *transparent* to users. Attaining that transparency requires middleware support that is *adaptive*, to cope with frequently changing environments, and *extensible*, to allow different adaptation for different applications. To provide an evolutionary path for applications widely used today, such support must also be upward-*compatible* with existing interfaces. Our work has developed a software infrastructure that supports mobility while simultaneously attending to *security* issues.

Mobile computing's unique constraints [Satyanarayanan 96a] pose significant challenges to the design of distributed systems. Research opportunities include five important topics: caching metrics, semantic callbacks and validators, resource revocation, analysis of adaptation, and estimating global parameters from local observations.

Our research approach derives from an examination of the fundamental forces at work in mobile computing systems and the way they constrain mobile information access. *Adaptivity*, we conclude, represents a crucial requirement of mobile clients. Our analysis has produced a taxonomy of adaptation strategies [Satyanarayanan 96b], including *application-transparent* and *application-aware* variants.

The former strategy allows applications to remain unchanged, while the latter, complementary approach views adaptation as a collaborative partnership between applications and the system [Noble and Satyanarayanan 95]. Our research proceeds on both fronts, investigating application-transparency in the context of the Coda file system. Section 2 also presents our work toward application-awareness within the Odyssey system.

1.1 Computational Tools and Infrastructure

Advancing distributed-file-system frontiers requires both empirical and synthetic data to guide and confirm progress. For this purpose, we developed a tool to collect and analyze long-term file reference data in a distributed, UNIX workstation environment [Mummert and Satyanarayanan 96]. Our unique, DFSTrace system pays particular attention to efficiency, extensibility, and the logistics of long-term trace collection. In operation DFSTrace has been virtually unnoticeable, degrading performance merely 3-7%, and the resulting data have proven useful for a variety of purposes. To simulate user data, we developed a synthetic file-reference generator, SynRGen, that operates at the system-call level and models a wide variety of usage environments [Ebling and Satyanarayanan 94]. The system achieves realism through trace-inspired micromodels — each a parameterized code fragment that captures an application's distinctive signature — and flexibility by combining micromodels stochastically. SynRGen can emulate real users closely, within 20% on key system variables, and has facilitated extensive stress-testing of Coda.

One critical piece of infrastructure is the RVM system [Mashburn and Satyanarayanan 94, Satyanarayanan et al. 94a], which provides an efficient, portable, and easily used implementation of recoverable virtual memory for UNIX environments. Recoverable virtual memory refers to regions of a virtual address space on which transactional guarantees are offered. RVM also allows independent control over the transactional properties of atomicity, permanence, and Serializability. That control, in turn, enhances RVM's flexibility and extends the range of applications that can exploit transactions.

The RPC2 package [Satyanarayanan et al. 95] provides another capability essential to our work, a highly portable, extensible, remote procedure call facility. It supports streaming file transfer, parallel RPC calls, and IP-level multicast for UNIX. RPC2 and its utilities — a stub generator and coroutine-based lightweight process package — all run at user level. We have also ported the software to various architectures and to several variants of the UNIX operating system.

1.2 Application-transparent Adaptation

Modern, distributed file systems gain considerable power by exploiting the strong connectivity available from today's cheap, fast, and reliable networks. The Andrew File System (AFS), which originated at Carnegie Mellon, illustrates this point. AFS has grown into a wide-area, distributed file system spanning well over 100 organizations worldwide, with each typically containing many tens or hundreds of clients. Our usage study of AFS [Spasojevic and Satyanarayanan 94] indicates that AFS provides robust and efficient data access and represents a viable a design point for wide-area distributed file systems. More recently, the World Wide Web has further extended the wide-area information access concept and simultaneously pushed network and server resources to their limits. A comparative study of AFS and the Web [Satyanarayanan and Spasojevic 96] reveals that each offers technologies and lessons valuable to the successful growth of global, mobile networks.

At the other connectivity extreme, during server or network failures or deliberate disconnection, file system clients can only use data in their own caches. Such *disconnected operation* imposes numerous limitations: Updates are invisible to other clients; cache misses may impede progress; updates are at risk from client loss or damage; and the danger of update conflicts increases with disconnection duration.

1.2.1 Weakly-connected operation

Weakly-connected operation, due to intermittent, low-bandwidth communication or to expensive networks, lies between these two conditions and provides an opportunity to recover from disconnected operation. To exploit this opportunity, we have evolved the Coda file system, identifying hidden assumptions about strong connectivity and systematically eliminating them by introducing application-transparent adaptivity [Mummert, Ebling, and Satyanarayanan 95, Mummert 96]. In Coda, the system bears full responsibility for coping with the demands of weak connectivity. Applications can run unchanged, so this approach offers upward compatibility. The evolution has affected numerous system characteristics, including communication, cache validation, update propagation and cache-miss handling. As a result, Coda provides good performance, even when network bandwidth varies over four orders of magnitude — from modem to LAN speeds.

Coda, like its ancestor, AFS, provides location-transparent access to a shared UNIX file namespace that maps onto a collection of dedicated file servers. Coda represents a substantial improvement over AFS, however, because it offers considerably greater availability in the face of server and network failures [Noble and Satyanarayanan 94, Satyanarayanan, Ebling, and Raiff 95]. This improved availability derives from two, complementary mechanisms: server replication and disconnected operation.

Coda's communications layer provides the foundation for adaptivity, deriving and suppling information on network conditions to higher system layers through three key mechanisms: *Rapid cache validation* enables the system to recover quickly in intermittent environments; *trickle* reintegration insulates the user from poor network performance by propagating updates to servers asynchronously; *cache-miss handling* alerts the user to potentially lengthy service times and provides opportunities for intervention.

Mobile computing will place additional stress on increasingly strained networks. While system designers have generally treated the network as an inexhaustible resource, mobility will highlight the network, rather than CPU or storage usage, as a bottleneck [Ebling, Mummert, and Steere 94]. Coda addresses these challenges by treating the network as a first-class resource, expending computational and storage resources to use it intelligently, employing, techniques of prescient caching and smart scheduling.

1.2.2 Exploiting weak connectivity

In a distributed file system, particularly one with mobile components, situations of weak connection — limited bandwidth and frequent connectivity changes — offer opportunities to recover from periods of disconnection. Our research explored several approaches to exploit this opportunity.

A conventional strategy for reducing the impact of intermittent connectivity minimizes client-server communication with callback-based, cache-coherence schemes. This technique, however, assumes a fast and reliable underlying network, since clients must revalidate files whenever connections are re-established. Our work introduced an alternative, *large granularity callbacks* [Mummert and Satyanarayanan 94a], that trades precision of invalidation for speed of validation after connectivity changes. Measurements indicate that, at 9.6 Kbps, large-granularity callbacks can validate Coda user caches 5-25 times faster than two competing strategies [Mummert and Satyanarayanan 94b]. A formal analysis [Mummert, Wing, and Satyanarayanan 94] applied the notion of ''belief'' to reason about the three cache-coherence protocols. Using an extended subset of a logic of authentication, we were able to identify, and subsequently repair, design flaws in our large-granularity protocol.

Optimistic replication strategies offer the possibility of increased data availability in distributed systems. To be practical, they must also handle potentially conflicting, partitioned updates, which occur frequently in mobile computing environments. Our work developed mechanisms that transparently resolve diverging replicas in the Coda file system [Kumar 94]. Coda provides a framework for invoking customized pieces of code called application-specific resolvers (ASRs) that encapsulate the knowledge needed for file resolution [Kumar and Satyanarayanan 95]. When resolution succeeds, the user notices nothing more than a slight performance delay. Should automatic resolution fail, observed in less than 1% of the cases studied, a repair tool helps the user merge file replicas. The combination of both automatic and system-assisted manual mechanisms enables Coda to provide high data availability with minimal impact on usability, scalability, security, and performance.

1.2.3 Maintaining consistency

The UNIX file system provides data consistency through its shared-memory model, but its lack of concurrency control makes it susceptible to unexpected read/write sharing between different processes. In practice such conflicts have been rare. With the advent of mobile computing, however, they become a realistic threat to data integrity. To preserve upward compatibility for the large body of existing UNIX software while offering improved consistency in a mobile computing environment, we developed and implemented a new transaction model *isolation only transactions* (IOT) [Lu and Satyanarayanan 94]. IOT imposes serializability-based isolation requirements on partitioned transaction executions [Lu and Satyanarayanan 95, Lu 96]. Transactions executed on a disconnected client remain tentative until the client regains connection to the relevant servers. The system then validates each transaction and commits it to the servers. Invalidated transactions are automatically or manually resolved to ensure global consistency. Empirical results [Lu and Satyanarayanan 97] show that IOT's resource-conservation techniques do indeed result in modest demands on a mobile client's three critical resources: CPU and I/O usage, disk space, and RVM space. Overall, our measurements confirm that even a severely resource-constrained mobile client can benefit from the improved consistency offered by the IOT mechanism.

1.2.4 Reducing I/O latency

Today's operating systems face, among others, three challenges: reducing the performance impact of I/O latency, coping with the variable bandwidth and latency of mobile communication environments, and efficiently locating target files in distributed systems. Our work addresses all three problems with a single unifying abstraction, *dynamic sets* [Steere and Satyanarayanan 94, Steere and Satyanarayanan 95].

Dynamic sets are short-lived, unordered object collections, that search applications create to hold query results. An object's membership in a set *hints* of future access, informing the I/O subsystem that prefetching the object could improve performance. In addition to providing opportunities to exploit prefetching and parallelism aggressively, dynamic sets support associative naming, which complements the hierarchical naming of typical file systems.

To understand the varying degrees of inconsistency clients might be willing to tolerate and to understand the tradeoff between providing strong consistency guarantees and implementing weak sets efficiently, we specified the abstraction's behavior formally [Steere and Wing 95]. In addition, empirical evaluation shows that dynamic sets can substantially reduce I/O latency for search on both wide- and local-area distributed systems and on local file systems. Replaying traces of real users searching on the WWW, for example, shows that sets can reduce latency by over an order of magnitude across a range of factors [Steere 95, Steere 96, Steere 97].

2. Application-aware Adaptation

Application-aware adaptation, we contend, forms an essential capability for mobile clients. Our research has developed this technology in the context of the *Odyssey* architecture [Satyanarayanan et al. 94b]. Odyssey splits previously monolithic functionality between the operating system and individual applications. The operating system senses external events and monitors and allocates scarce resources. Individual applications adapt to changing conditions by exploiting system-provided information and resources.

Coping with wide variations in network conditions and local resource availability, mobile clients need an ability to retrieve and present diverse data types at varying degrees of fidelity. Odyssey has has demonstrated it can do this job for both video and map data [Noble, Price, and Satyanarayanan 95].

Through collaborative partnership between operating system and applications, application-aware adaptation offers the most general and effective approach to mobile information access. Odyssey's *agility* represents a key attribute of adaptive systems, and we describe how to quantify and measure it in [Noble et al. 97a]. This preliminary evaluation indicates up to fivefold performance improvements on a benchmark of three, concurrent applications, using remote services over a network with highly variable bandwidth.

3. Low-level System Support for Mobility

One of the fundamental resources for research in mobile computing is a genuinely mobile computing environment. To support our work, we have built such an infrastructure. The initial design [Hills and Johnson 96] specifies two network types: a low-bandwidth, wide-area system and a high-bandwidth, local-area system. Each provides provides campus community members seamless access to the University's computer network from anywhere within the greater Pittsburgh area.

Our larger effort in this area is developing networking protocols and protocol interfaces that allow truly seamless wireless and mobile host networking. This work includes protocol design, implementation, performance evaluation, and usage-based validation, spanning areas ranging roughly from portions of the ISO Data Link layer (Layer 2) through the Presentation layer (Layer 6) [Johnson and Maltz 96a].

Current internetworking protocols, such as the Internet's IPv4, do not support host movement. Our work has analyzed this problem and considered how to support large numbers of mobile hosts transparently within a large internetwork, such as the Internet [Johnson 96].

We developed the concept of an ‘‘ad hoc network,’’ a collection of wireless mobile hosts that form a temporary network without the aid of any established infrastructure or centralized administration. In such an environment one mobile host may enlist the aid of others in forwarding a packet to its destination. Our *dynamic source routing* protocol for this process adapts quickly to routing changes when hosts move frequent, yet requires little or no overhead during periods in which hosts remain stationary [Johnson and Maltz 96b].

Working with the Internet Engineering Task Force (IETF), we have implemented our dynamic source routing ideas under a new internet protocol, IPv6, that will replace IPv4. Specifically, we have designed enhancements for IPv6, known as Mobile IPv6, that support transparent routing of packets to mobile nodes [Perkins and Johnson 96, Johnson and Perkins 96a, Johnson and Perkins 96b]. Mobile IPv6 provides mobility IPv6 hosts by allowing them to leave their home subnet while transparently maintaining all of their present connections and remaining reachable to the rest of the Internet [Broch 97].

At a higher level, mobile networks are both poorly understood and difficult to study empirically. To address this challenge, we are developing tools for mobile network tracing, tools that both advance our understanding of wireless channels and enable us to build realistic, repeatable testbeds for mobile software and systems. We have suggested to the technical community a candidate format for collecting mobile network traces [Noble et al. 96]. The RFC also describes tools for generating analytic models of mobile network behavior from network traces and trace-modulation tools that allow emulating wireless channel latency, bandwidth, loss, and error rates on private, wired networks.

Our designs for trace-modulation techniques derive from analyses of loss behavior in our in-building wireless interface, an AT&T WaveLAN system. Results indicate that the network averages two to three percent packet-error rate and that errors are not independent. We model these phenomena using distributions of the errorful and error-free packet stream lengths. Modulating both traces and our models in a simulator, we observed average TCP throughput agreed within five percent [Nguyen et al. 96].

Trace modulation faithfully recreates observed, end-to-end characteristics of real, wireless networks in a controlled and repeatable manner. It is also transparent to applications and accounts for all network traffic sent or received by the system under test [Noble et al. 97b]. We have found trace modulation an indispensable tool for building and evaluating mobile computing systems. The essence of the technique is transparent, real-time, trace-driven emulation of a target network. Although conceptually simple, trace modulation strikes an attractive balance between the conflicting demands of realism, ease-of-use, and reproducibility. We expect trace modulation to play a central role in developing future mobile computing systems [Satyanarayanan and Noble 97]

4. Security Issues in Mobility

Work in this area emphasized secure remote execution, a critical technology for mobile applications and one that focuses on two issues: protecting a remote host from Trojan horses and protecting remotely executing applications from attack by the remote host. The latter ability allows security-critical software to execute on a remote machine. To implement this functionality, we have investigated using small, secure coprocessors [Yee 94, Tygar and Yee 94, Yee and Tygar 95] as well as portable, secure, coprocessing devices, such as smart cards [Gobioff et al. 96].

Our application of this technology for the US Postal Service allows mobile, disconnected devices to print postage and verify cryptographically encoded postal indicia [Tygar, Yee, and Heintze 96]. We also explored the use of this technology to support electronic commerce [Bahreman and Tygar 94, Camp, Sirbu, and Tygar 95, Camp et al. 96, Su and Tygar 96, Tygar 97]. We designed an efficient, secure, time function that permits applications to run securely, even when they depend on causality relations implied by potentially compromised time sources [Smith et al. 94, Smith and Tygar 94]. Finally, we built a system for secure, private access to distributed databases, a mechanism that allows databases to reside on physically accessible devices [Camp and Tygar 94], and explored flexible specification mechanisms for access control [Heydon and Tygar 94].

5. Robustness and Portability Enhancements

A significant portion of our effort concentrated on evolving our technology for secure, mobile computing from an academic research tool into a robust and widely portable platform for continued development in other and more commercial environments.

We ported the Coda system from AT&T **C**$^{++}$, in which it was originally written, to g^{++} in an important step toward facilitating distribution and use outside Carnegie Mellon. An agreement negotiated with IBM freed the source code from the last vestiges of encumbrance and allowed distributing Coda via anonymous FTP on the Internet and Web. We have now ported the entire Coda system to the NetBSD and Linux platforms, developing a special MiniCache for NetBSD in the process.

Our demonstration videotape, highlighting Coda's capabilities, supports technology-transfer and education. Our comprehensive *Coda User Manual* and detailed documentation of Coda server internals will help users beyond Carnegie Mellon understand and use the system more easily and effectively.

Specific arrangements with industrial users illustrate Coda's progress toward commercial viability:

- A licensing agreement with Novell, Inc., allows them to use Coda in their commercial products and represents an important step in transferring Coda technology.
- CTA, Inc. of Rockville, MD, had planned to incorporate a modified version of the Coda file system into their ''Warfighter's Associate'' (WFA), which they proposed to develop for DARPA's ''Battlefield Awareness and Data Dissemination'' initiative.

6. References

[Bahreman and Tygar 94]
Bahreman, A., and J.D. Tygar.
Certified Electronic Mail.
In *Proceedings of 1994 Symposium on Network and Distributed System Security, and Exhibition*, pages 3-19. The USENIX Association, January, 1994.

We propose two new families of protocols for certified electronic mail. Certified electronic mail enables two mutually suspicious users to exchange a receipt for electronic mail. One family of protocols, the believers' protocols, use a trusted third party. The second family, the skeptics' protocols, use no third party. Our protocols are secure in a very strong sense; the probability of one party cheating can be made arbitrarily small. The protocols provide a practical example of the use of various innovative cryptographic techniques, including digital signatures, bit-commitment, and zero-knowledge interactive proofs. These protocols can be implemented in modern communication networks.

[Broch 97] Broch, J.G.
An Implementation and Evaluation of Mobile IPv6.
Master's thesis, Information Networking Institute, Carnegie Mellon University, March, 1997.

An increasing demand for mobility coupled with the popularity of the Internet has created the need for a protocol that allows Internet nodes to roam while still maintaining transport-level connectivity to the Internet. IPv6 (IPng) is the new version of the Internet Protocol that is presently being standardized by the Internet Engineering Task Force (IETF). Mobile IPv6 provides mobility for IPv6 hosts by allowing them to leave their home subnet while transparently maintaining all of their present connections and remaining reachable to the rest of the Internet.

In Mobile IPv6 each node is always identified by its home address, regardless of its current point of attachment to the Internet. While a host it not on its home network, it is associated with a care-of address that indicates it present location. Most packets destined for a mobile node are routed directly to the mobile node's care-of address by the sending node.

[Camp and Tygar 94]
Camp, J., and J.D. Tygar.
Protecting Privacy while Preserving Access to Data.
*The Information Society*10(1), January, 1994.

Historically, there has been tension between performance and privacy of information systems because of the crucial role of collection of usage data. In this paper, we examine how a number of different architectures approach this tension. We present both enhancements to traditional software architectures and an architecture that resolves this conflict. We discuss a cryptographic technique called secret counting that allows us to collect aggregate data. We discuss these architectures and techniques in the context of the specific application of library information systems.

[Camp et al. 96] Camp, L.J., M. Harkavy, J.D. Tygar, and B. Yee.
Anonymous Atomic Transactions.
In *Proceedings of the Second USENIX Workshop on Electronic Commerce*, pages 123-133. The USENIX Association, November, 1996.

We show here an example of a protocol that satisfies anonymity properties while providing strong ACID (atomic, consistent, isolated, durable) transactional properties, resolving an open question. This allows us to provide electronic commerce protocols that are robust even in the event of message loss and communication failures. We use blind signature tokens to control values. We use a separate transaction log to reduce trust assumptions between the merchant, the consumer, and the bank.

[Camp, Sirbu, and Tygar 95]
Camp, L.J., M. Sirbu, and J.D. Tygar.
Token and Notational Money in Electronic Commerce.
In *Proceedings of the USENIX Workshop on Electronic Commerce*. The USENIX Association, July, 1995.

What properties of money are important for electronic commerce? We argue that both transactional and privacy properties distinguish electronic commerce systems. We provide a quick overview of the history of money. We then consider privacy provided by different forms of money, and socially desirable disclosure of information as specified by legal reporting requirements. We classify electronic and traditional commerce systems into two categories:

- Token systems, which exchange markers representing value
- Notational systems, where value is stored as notations in a ledger or computer.

We analyze different forms of traditional money based on the degree to which they protect the privacy and preserve transactional ACID (atomicity, consistency, isolation, durability) properties. Finally we apply our evaluation criteria to two proposed electronic commerce systems: Digicash, a token-based system; and NetBill, a notational system.

[Ebling and Satyanarayanan 94]
Ebling, M.R., and M. Satyanarayanan.
SynRGen: An Extensible File Reference Generator.
In *Proceedings of the 1994 ACM SIGMETRICS Conference*. ACM, May, 1994.
Nashville, TN.

SynRGen, a synthetic file reference generator operating at the system call level, is capable of modeling a wide variety of usage environments. It achieves realism through trace-inspired micromodels and flexibility by combining these micromodels stochastically. A micromodel is a parameterized piece of code that captures the distinctive signature of an application. We have used SynRGen extensively for stress testing the Coda File System. We have also performed a controlled experiment that demonstrates SynRGen's ability to closely emulate real users — within 20% of many key system variables. In this paper we present the rationale, detailed design, and evaluation of SynRGen, and mention its applicability to broader uses such as performance evaluation.

[Ebling, Mummert, and Steere 94]
Ebling, M.R., L.B. Mummert, and D.C. Steere.
Overcoming the Network Bottleneck in Mobile Computing.
In *Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications*. IEEE, December, 1994.
Santa Cruz, CA.

System designers have traditionally treated the network as an inexhaustible resource, focusing their efforts on optimizing CPU and storage usage. For instance, the popular NFS file system supports diskless operation, thereby avoiding use of local secondary storage at the expense of increased network usage. But in mobile computing it is the network, rather than CPU or storage, that will be the scarce resource. The time has come when we must treat the network as a first-class resource, expending the CPU and storage resources necessary to use it intelligently. In this paper we argue that prescient caching and smart scheduling are key techniques for overcoming the network bottleneck. We use the Coda file system as a case study to substantiate our position.

[Gobioff et al. 96] Gobioff, H., S. Smith, J.D. Tygar, and B.S. Yee.
Smart cards in hostile environments.
In *Proceedings of the Second USENIX Workshop on Electronic Commerce*, pages 23-28. The USENIX Association, November, 1996.
Oakland, CA. An earlier version available as technical report CMU-CS-95-188.

One often hears the claim that smart cards are the solution to a number of security problems, including those arising in point-of-sale systems. In this paper, we characterize the minimal properties necessary for the secure smart card point-of-sale transactions. Many proposed systems fail to provide these properties: problems arise from failures to provide secure communication channels between the user and the smart card while operating in a potentially hostile environment (such as a point-of-sale application.) Moreover, we discuss several types of modifications that can be made to give smart cards additional input/output capacity with a user, and describe how this additional I/O can address the hostile environment problem. We give a notation for describing the effectiveness of smart cards under various environmental assumptions. We discuss several security equivalences among different scenarios for smart cards in hostile environments.

[Heydon and Tygar 94]
Heydon, A., and J.D. Tygar.
Specifying and Checking UNIX Security Constraints.
*Computing Systems*7(1):91-112, 1994.

We describe a system called Miro for specifying and checking security constraints. Our system is general because it is not tied to any particular operating system. It is flexible because users express security policies in a formal specification language, so it is easy to extend or modify a policy simply by augmenting or changing the specification for the current policy. Finally, our system is expressive enough to describe many relations on file system configurations; however, it is not expressive enough to describe more subtle security holes like Trojan horses or weaknesses in the passwords chosen by the system's users.

This paper is a case study of the Miro languages and tools. We show how to represent various UNIX security constraints — including those described in a well known paper on UNIX security — using our graphical specification language. We then describe the results we obtained from running our tools to check an actual UNIX file system against these constraints.

[Hills and Johnson 96]
Hills, A., and D.B. Johnson.
A Wireless Data Network Infrastructure at Carnegie Mellon University.
*IEEE Personal Communications*3(1):56-63, February, 1996.

In order to support mobile computing research, including the development of software which will allow seamless access to multiple wireless data networks, we are building a wireless data network infrastructure at Carnegie Mellon University. This infrastructure will allow researchers and other members of the campus community to use mobile computers to gain access to data networks while they are on-campus or while they are off-campus in the greater Pittsburgh area. The infrastructure will initially include two different types of wireless networks, a low-bandwidth wide area system and a high-bandwidth local area system, each of which will provide access to our campus computer network. Since our campus network is called ‘‘Andrew’’ (after Andrew Carnegie and Andrew Mellon), the new wireless infrastructure has been dubbed ‘‘Wireless Andrew.’’ This article describes the Wireless Andrew infrastructure we are building.

[Johnson 96] Johnson, D.B.
Scalable Support for Transparent Mobile Host Internetworking.
Mobile Computing.
In T. Imielinski and H. Korth,
Kluwer Academic Publishers, 1996, pages 103-128, Chapter 3.

This paper considers the problem of providing transparent support for very large numbers of mobile hosts within a large internetwork such as the Internet. The availability of powerful mobile computing devices and wireless networking products and services is increasing dramatically, but internetworking protocols such as IP used in the Internet do not currently support host movement. To address this need, the Internet Engineering Task Force (IETF) is currently developing protocols for mobile hosts in the Internet. This paper analyzes the problem to be solved, reviews the current state of that effort, and discusses its scalability to very large numbers of mobile hosts in a large internetwork.

[Johnson and Maltz 96a]
Johnson, D.B., and D.A. Maltz.
Protocols for Adaptive Wireless and Mobile Networking.
*IEEE Personal Communications*3(1):34-42, February, 1996.

The goal of the Monarch Project at Carnegie Mellon University is to develop networking protocols and protocol interfaces to allow truly seamless wireless and mobile host networking. The scope of our efforts includes protocol design, implementation, performance evaluation, and usage-based validation, spanning areas ranging roughly from portions of the ISO Data Link layer (layer 2) through the Presentation layer (layer 6). In this article, we give a status report of our current work in the Monarch Project, placing it in the context of broader efforts by the Internet mobile networking community.

[Johnson and Maltz 96b]
Johnson, D.B., and D.A. Maltz.
Dynamic Source Routing in ad hoc Wireless Networks.
Mobile Computing.
In T. Imielinski and H. Korth,
Kluwer Academic Publishers, 1996, pages 153-181, Chapter 5.
Invited paper.

An ad hoc network is a collection of wireless mobile hosts forming a temporary network without the aid of any established infrastructure or centralized administration. In such an environment, it may be necessary for one mobile host to enlist the aid of other hosts in forwarding a packet to its destination, due to the limited range of each mobile host’s wireless transmissions. This paper presents a protocol for routing in ad hoc networks that uses dynamic source routing. The protocol adapts quickly to routing changes when host movement is frequent, yet requires little or no overhead during periods in which hosts move less frequently. Based on results from a packet-level simulation of mobile hosts operating in an ad hoc network, the protocol performs well over a variety of environmental conditions such as host density and movement rates. For all but the highest rates of host movement simulated, the overhead of the protocol is quite low, falling to just 1% of total data packets transmitted for moderate movement rates in a network of 24 mobile hosts. In all cases, the difference in length between the routes used and the optimal route lengths is negligible, and in most cases, route lengths are on average within a factor of 1.01 of optimal.

[Johnson and Perkins 96a]
Johnson, D.B., and C.E. Perkins.
Mobility Support in IPv6.
Internet-Draft, Internet Engineering Task Force.
November, 1996. Replaces June, 1996, version.

This document specifies the operation of mobile computers using IPv6. Each mobile node is always identified by its home address, regardless of its current point of attachment to the Internet. While situated away from its home, a mobile node is also associated with a care-of address, which provides information about the mobile node's current location. IPv6 packets addressed to a mobile node's home address are transparently routed to its care-of address. The protocol enables IPv6 nodes to cache the binding of a mobile node's home address with its care-of address, and to then send packets destined for the mobile node directly to it at this care-of address.

[Johnson and Perkins 96b]
Johnson, D.B., and C.E. Perkins.
Route Optimization in Mobile IP.
Internet-Draft, Internet Engineering Task Force.
November, 1996. Replaces February, 1996, version.

This document defines extensions to the operation of the base Mobile IP protocol to allow for optimization of datagram routing from a correspondent node to a mobile node. Without Route Optimization, all datagrams destined to a mobile node are routed through that mobile node's home agent, which then tunnels each datagram to the mobile node's current location. The protocol extensions described here provide a means for correspondent nodes that implement them to cache the binding of a mobile node and to then tunnel their own datagrams for the mobile node directly to that location, bypassing the possibly lengthy route for each datagram to and from the mobile node's home agent. Extensions are also provided to allow datagrams in flight when a mobile node moves, and datagrams sent based on an out-of-date cached binding, to be forwarded directly to the mobile node's new binding.

[Kumar 94] Kumar, P.
Mitigating the Effects of Optimistic Replication in a Distributed File System.
PhD thesis, Computer Science Department, Carnegie Mellon University, December, 1994.
Available as technical report CMU-CS-94-215.

Optimistic replication strategies can significantly increase availability of data in distributed systems. However, such strategies cannot guarantee global consistency in the presence of partitioned updates. The danger of conflicting partitioned updates, combined with the fear that the machinery needed to cope with conflicts might be excessively complex has prevented designers from using optimistic replication in real systems.

This dissertation puts these fears to rest by showing that it is indeed practical and feasible to use optimistic replication in distributed file systems. It describes the design, implementation and evaluation of the mechanisms used to transparently resolve diverging replicas in the Coda file system. Files and directories are resolved using orthogonal mechanisms due to the difference in their structure and semantics. A server-based mechanism that uses operation logging is utilized to resolve directories, while a client-based mechanism that uses application support is utilized to resolve files. When automatic resolution fails a repair tool in conjunction with standard UNIX utilities aids the user in merging the diverging replicas. The combination of these mechanisms allows the system to provide high data availability with minimal impact on its usability, scalability, security and performance.

This dissertation makes four significant contributions: a design of simple yet novel automatic resolution techniques; a formalization of the UNIX file system model and proof of correctness of the resolution methods; implementation of these methods in a system with a real user community; and measurements, showing the efficacy of the approach.

[Kumar and Satyanarayanan 95]
Kumar, P., and M. Satyanarayanan.
Flexible and Safe Resolution of File Conflicts.
In *Proceedings of the USENIX Winter 1995 Technical Conference*. The USENIX Association, January, 1995.
New Orleans.

In this paper we describe the support provided by the Coda File System for transparent resolution of conflicts arising from concurrent updates to a file in different network partitions. Such partitions often occur in mobile computing environments. Coda provides a framework for invoking customized pieces of code called application-specific resolvers (ASRs) that encapsulate the knowledge needed for file resolution. If resolution succeeds, the user notices nothing more than a slight performance delay. Only if resolution fails does the user have to resort to manual repair. Our design combines a rule-based approach to ASR selection with transactional encapsulation of ASR execution. This paper shows how such an approach leads to flexible and efficient file resolution without loss of security or robustness.

[Lu 96] Lu, Q.
Improving Data Consistency for Mobile File Access Using Isolation-Only Transactions.
PhD thesis, Computer Science Department, Carnegie Mellon University, May, 1996.
Available as technical report CMU-CS-96-131.

Disconnected operation based on optimistic replication has been demonstrated as an effective technique enabling mobile computers to access shared data in distributed file systems. To guard against inconsistencies resulted from partitioned data sharing, past research has focused on detecting and resolving write/write conflicts. However, experience shows that undetected read/write conflicts pose a subtle but serious threat to data integrity in mobile file access. Solving this problem is critical for the future success of mobile computing.

This dissertation shows that *isolation-only transaction* (IOT), an upward compatible transaction mechanism for the UNIX File System, is a viable solution to this problem. The central idea of the IOT model is imposing serializability-based *isolation* requirements on partitioned transaction executions. Transactions executed on a disconnected client stay in a tentative state until the client regains connection to relevant servers. They are committed to the servers as soon as they pass consistency validation. Invalidated transactions are automatically or manually resolved to ensure global consistency. Powerful resolution mechanisms such as automatic transaction re-execution and application specific resolver invocation can transparently resolve conflicts for many common UNIX applications. In addition, a concise conflict representation scheme enables application semantics to be smoothly integrated for only conflict resolution and consistency validation.

The main contributions of this thesis research are the following: the design of an isolation only transaction model specialized for improving mobile file consistency while preserving upward compatibility with existing UNIX applications; the development of a working IOT implementation in the Coda file system; experimentation and evaluation demonstrating the feasibility and practicality of the IOT model.

[Lu and Satyanarayanan 94]
Lu, Q., and M. Satyanarayanan.
Isolation-only Transactions for Mobile Computing.
*Operating Systems Review*28(2):81-87, April, 1994.

The UNIX File System (UFS) has historically offered a *shared-memory* consistency model. The lack of concurrency control makes this model susceptible to *read/write conflicts*, i.e., unexpected read/write sharing between two different processes. For example, the update of a header file by one user while another user is performing a long-running make can cause inconsistencies in the compilation results. In practice, read/write conflicts are rare for two reasons. First, the window of vulnerability is relatively small because read/write conflicts only occur when the executions of two processes overlap. Second, they are often prevented via explicit user-level coordination.

However, the advent of *mobile computing* makes read/write conflicts a realistic threat to data integrity. Mobile computing is characterized by periods of *disconnection* and *intermittent connectivity*. Such communication disturbances greatly widen the window of vulnerability from the life span of a process to the duration of a disconnection. They also significantly reduce the effectiveness of explicit user-level coordination, especially when disconnections are made transparent to users.

How can we preserve upward compatibility with the large body of existing UNIX software, while offering improved consistency in a mobile computing environment? This position paper puts forth our solution, a new transaction model called *isolation only* transaction (IOT).

[Lu and Satyanarayanan 95]
Lu, Q., and M. Satyanarayanan.
Improving Data Consistency in Mobile Computing Using Isolation-Only Transactions.
In *Proceedings of the Fifth IEEE HotOS Topics Workshop*. IEEE, May, 1995.
Orcas Island, WA.

Disconnected operation is an important technique for providing mobile access to shared data in distributed file systems. However, data inconsistency resulting from partitioned sharing remains a serious concern. This paper presents a new mechanism called isolation-only transaction (IOT) that uses serializability constraints to automatically detect read/write conflicts. The IOT consistency model provides a set of options for automatic and manual conflict resolution. In addition, application specific knowledge can be incorporated to detect and resolve conflicts. To preserve upward UNIX compatibility, the IOT mechanism is provided as an optional file system facility and its flexible interfaces allow any existing UNIX application to be executed as an IOT. This paper describes high level system design and implementation and concludes with related work and current status.

[Lu and Satyanarayanan 97]
Lu, Q., and M. Satyanarayanan.
Resource Conservation in a Mobile Transaction System.
*IEEE Transactions on Computers*46(3), March, 1997.

This paper addresses the problem of providing transactional support for improved data consistency in mobile file access, while paying careful attention to the resource constraints of mobile clients. We present data on resource consumption from an implementation of the isolation-only transaction (IOT) mechanism of the Coda File System. The data shows that the resource conservation techniques used by the IOT mechanism do indeed result in modest demands on the three critical resources on a mobile client: CPU and I/O usage, disk space and RVM space. Overall, our measurements confirm that even a

severely resource-constrained mobile client can benefit from the improved consistency offered by the IOT mechanism.

[Mashburn and Satyanarayanan 94]
Mashburn, H., and M. Satyanarayanan.
RVM User Manual, Release 1.3
School of Computer Science, Carnegie Mellon University, 1994.

RVM provides an unstructured recoverable virtual memory. The recoverable storage is represented by UNIX files or disk partitions that applications can map at page granularity into the address space of a process. Simple, non-nested atomic transactions guarantee permanence of changes to recoverable storage across system crashes. Applications can schedule transaction logging actions to enhance performance.

The design stresses simplicity, ease of use, and high performance. UNIX compatibility is standard, while optional Mach-specific extensions are supported for additional flexibility and performance. RVM has been extensively used in the clients and servers of the Coda File System, and in the Venari system.

[Mummert 96] Mummert, L.B.
Exploiting Weak Connectivity for Mobile File Access.
PhD thesis, Computer Science Department, Carnegie Mellon University, December, 1996.
Available as technical report CMU-CS-96-195.

Weak connectivity, in the form of intermittent, low-bandwidth, or expensive networks is a fact of life in mobile computing. For the foreseeable future, access to cheap, high-performance, reliable networks, or *strong connectivity* will be limited to a few oases, such as work or home, in a vast desert of weak connectivity. The design of distributed file systems has traditionally been based on an assumption of strong connectivity. Yet, to provide ubiquitous data access, it is vital that distributed file systems make effective use of weak connectivity.

This dissertation describes the design, implementation, and evaluation of weakly connected operation in the Coda File System. Coda's strategy for weakly connected operation is best characterized as *application-transparent adaptation*. The system bears full responsibility for coping with the demands of weak connectivity. This approach preserves upward compatibility by allowing applications to run unchanged. Coda provides several mechanisms for weakly connected operation motivated by actual experience.

A quantitative evaluation of these mechanisms, based on controlled experimentation and empirical data gathered from the deployed system in everyday use, shows that Coda is able to provide good performance even when network bandwidth varies over four orders of magnitude — from modem speeds to LAN speeds.

[Mummert and Satyanarayanan 94a]
Mummert, L.B., and M. Satyanarayanan.
Variable Granularity Cache Coherence.
*Operating Systems Review*28(1):55-60, January, 1994.

Weak connectivity is characterized by slow or intermittent networks. Distributed file systems using weak connections must function in spite of limited bandwidth and frequent connectivity changes. Callback-based cache coherence schemes were designed to minimize client-server communication, but with an underlying assumption that the network is fast and reliable (i.e., a LAN). This paper presents large granularity callbacks as a way to reduce the client-server communication necessary to maintain file cache coherence. Large granularity callbacks trade off precision of invalidation for speed of validation after connectivity changes.

[Mummert and Satyanarayanan 94b]
Mummert, L.B., and M. Satyanarayanan.
Large Granularity Cache Coherence for Intermittent Connectivity.
In *Proceedings of the Summer USENIX Conference*. The USENIX Association, June, 1994.

To function in mobile computing environments, distributed file systems must cope with networks that are slow, intermittent, or both. Intermittence vitiates the effectiveness of callback-based cache coherence schemes in reducing client-server communication, because clients must validate files when connections are reestablished. In this paper we show how maintaining cache coherence at a large granularity alleviates this problem. We report on the implementation and performance of large granularity cache coherence for the Coda File System. Our measurements confirm the value of this technique. At 9.6 Kbps, this technique takes only 4-20% of the time required by two other strategies to validate the cache for a sample of Coda users. Even at this speed, the network is effectively eliminated as the bottleneck for cache validation.

[Mummert and Satyanarayanan 96]
Mummert, L.B., and M. Satyanarayanan.
Long-Term Distributed File Reference Tracing: Implementation & Experience.
*Software Practice and Experience*26(6), June, 1996.
Also available as technical report CMU-CS-94-213.

DFSTrace is a system to collect and analyze long-term file reference data in a distributed UNIX workstation environment. The design of DFSTrace is unique in that it pays particular attention to efficiency, extensibility, and the logistics of long-term trace data collection in a distributed environment. The components of DFSTrace are a set of kernel hooks, a kernel buffer mechanism, a data extraction agent, a set of collection servers, and post-processing tools.

Our experience with DFSTrace has been highly positive. Tracing has been virtually unnoticeable, degrading performance 3-7%, depending on the level of detail of tracing. We have collected file reference traces from approximately 30 workstations continuously for over two years. We have implemented a post-processing library to provide a convenient programmer interface to the traces, and have created an on-line database of results from a suite of analysis programs to aid trace selection.

Our data has been used for a wide variety of purposes, including file system studies, performance measurement and tuning, and debugging. Extensions of DFSTrace have enabled its use in applications such as field reliability testing and determining disk geometry. This paper presents the design, implementation, and evaluation of DFSTrace and associated tools, and describes how they have been used.

[Mummert, Ebling, and Satyanarayanan 95]
Mummert, L.B., M.R. Ebling, and M. Satyanarayanan.
Exploiting Weak Connectivity for Mobile File Access.
In *Proceedings of the Fifteenth ACM Symposium on Operating Systems Principles*. December, 1995.
Copper Mountain Resort, CO.

Weak connectivity, in the form of intermittent, low-bandwidth, or expensive networks is a fact of life in mobile computing. In this paper, we describe how the Coda File System has been evolved to exploit such networks. The underlying theme of this evolution has been the identification of hidden assumptions about strong connectivity, and their systematic elimination through the introduction of adaptivity. Many aspects of the system, including communication, cache validation, update propagation and cache miss handling have been affected. As a result, Coda is able to provide good performance to users even when network bandwidth varies over four orders of magnitude — from modem speeds to LAN speeds. Users are affected by network quality mainly in the promptness with which updates are propagated, and in the degree of transparency with which cache misses are handled.

[Mummert, Wing, and Satyanarayanan 94]
Mummert, L.B., J.M. Wing, and M. Satyanarayanan.
Using Belief to Reason About Cache Coherence.
In *Proceedings of the 13th ACM Conference on Principles of Distributed Computing*. ACM, August, 1994.
Los Angeles.

The notion of belief has been useful in reasoning about authentication protocols. In this paper, we show how the notion of belief can be applied to reasoning about cache coherence in a distributed file system. To the best of our knowledge, this is the first formal analysis of this problem. We used an extended subset of a logic of authentication to help us analyze three cache coherence protocols: a validate-on-use protocol, an invalidation-based protocol, and a new large granularity protocol for use in weakly connected environments. In this paper, we present two runs from the large granularity protocol. Using our variant of the logic of authentication, we were able to find flaws in the design of the large granularity protocol. We found the notion of belief not only intuitively appealing for reasoning about our protocols, but also practical given the optimistic nature of our system model.

[Nguyen et al. 96] Nguyen, G., B. Noble, R. Katz, M. Satyanarayanan.
A Trace-Based Approach for Modelling Wireless Channel Behavior.
In *Proceedings of the Winter Simulation Conference*. December, 1996.

The loss behavior of wireless networks has become the focus of many recent research efforts. Although it is generally agreed that wireless communications experience higher error rates than wireline, the nature of these lossy links is not fully understood. This paper describes an effort to characterize the loss behavior of the AT&T WaveLAN, a popular in-building wireless interface. Using a trace-based approach, packet loss information is recorded, analyzed, and validated. Our results indicate that WaveLAN experiences an average packet error rate of two to three percent. Further analysis reveals that these errors are not independent, making it hard to model them with a simple two-state Markov chain. We derive another model based on the distributions of the error and error-free length of the packet streams. For validation, we modulate both the error models and the traces in a simulator. Simulations based on traces yield an average TCP throughput of about five percent less than simulations based our best error model.

[Noble and Satyanarayanan 94]
Noble, B., and M. Satyanarayanan.
An Empirical Study of a Highly Available File System.
In *Proceedings of the 1994 ACM SIGMETRICS Conference*. ACM, May, 1994.

In this paper we present results from a six-month empirical study of the high availability aspects of the Coda File System. We report on the service failures experienced by Coda clients, and show that such failures are masked successfully. We also explore the effectiveness and resource costs of key aspects of server replication and disconnected operation, the two high availability mechanisms of Coda. Wherever possible, we compare our measurements to simulation-based predictions from earlier papers and to anecdotal evidence from users. Finally, we explore how users take advantage of the support provided by Coda for mobile computing.

[Noble and Satyanarayanan 95]
Noble, B., and M. Satyanarayanan.
A Research Status Report on Adaptation for Mobile Data Access.
SIGMOD Record 24(4), December, 1995.
Mobility demands that systems be *adaptive*. One approach is to make adaptation transparent to applications, allowing them to remain unchanged. An alternative approach views adaptation as a collaborative partnership between applications and the system. This paper is a status report on our research on both fronts. We report on our considerable experience with *application-transparent* adaptation in the Coda File System. We also describe our ongoing work on *application-aware adaptation* in Odyssey.

[Noble et al. 96] Noble, B., G. Nguyen, M. Satyanarayanan, and R. Katz.
Mobile Network Tracing.
September, 1996. Draft Informational RFC, revised.
Mobile networks are both poorly understood and difficult to experiment with. This RFC argues that mobile network tracing provides both tools to improve our understanding of wireless channels, as well as to build realistic, repeatable testbeds for mobile software and systems. The RFC is a status report on our work tracing mobile networks. Our goal is to begin discussion on a standard format for mobile network tracing as well as a testbed for mobile systems research. We present our format for collecting mobile network traces, and tools to produce from such traces analytical models of mobile network behavior.
We also describe a set of tools to provide network modulation based on collected traces. Modulation allows the emulation of wireless channel latency, bandwidth, loss, and error rates on private, wired networks. This allows system designers to test systems in a realistic yet repeatable manner.

[Noble et al. 97a] Noble, B., M. Satyanarayanan, D. Narayanan, J.E. Tilton, J. Flinn, K. Walker.
Agile Application-Aware Adaptation for Mobility.
In *Proceedings of the 16th ACM Symposium on Operating System Principles*. ACM, October, 1997.
St. Malo, France. To appear.
In this paper we show that application-aware adaptation, a collaborative partnership between the operating system and applications, offers the most general and effective approach to mobile information access. We describe the design of Odyssey, a prototype implementing this approach, and show how it supports concurrent execution of diverse mobile applications. We identify agility as a key attribute of adaptive systems, and describe how to quantify and measure it. We present the results of our evaluation of Odyssey, indicating performance improvements up to a factor of five on a benchmark of three applications concurrently using remote services over a network with highly variable bandwidth.

[Noble et al. 97b] Noble, B., M. Satyanarayanan, G.T. Nguyen, and R. Katz.
Trace-Based Network Emulation.
In *Proceedings of ACM SIGCOMM'97*. ACM, September, 1997.
Cannes, France. To appear.
Subjecting a mobile computing system to wireless network conditions that are realistic yet reproducible is a challenging problem. In this paper, we describe a technique called *trace modulation* that recreates the observed end-to-end characteristics of a real wireless network in a controlled and repeatable manner. Trace modulation is transparent

to applications and accounts for all network traffic sent or received by the system under test. We present results that show that it is indeed capable of reproducing wireless network performance faithfully.

[Noble, Price, and Satyanarayanan 95]
Noble, B., M. Price, and M. Satyanarayanan.
A Programming Interface for Application-Aware Adaptation in Mobile Computing.
Computing Systems 8(4), Fall, 1995.
An early version appeared in the *Proceedings of the Second USENIX Symposium on Mobile and Location-Independent Computing*, held at Ann Arbor in April, 1995, and was nominated to the journal as an Outstanding Paper.

Mobile clients face wide variations in network conditions and local resource availability when accessing remote data. Coping with this uncertainty requires the ability to retrieve and present data at varying degrees of *fidelity*. In this paper we present *application-aware adaptation* as a solution to this problem. The essence of our solution is a collaborative partnership between applications and the operating system. We describe the *Odyssey* API for application-aware adaptation and demonstrate its use in accessing two types of data: video and maps.

[Perkins and Johnson 96]
Perkins, C.E., and D.B. Johnson.
Mobility Support in IPv6.
In *Proceedings of the Second Annual ACM International Conference on Mobile Computing and Networking (MobiCom'96)*. November, 1996.
Rye, NY.

IP version 6 (IPv6) is being designed within the IETF as a replacement for the current version of the IP protocol used in the Internet (IPv4). We have designed protocol enhancements for IPv6, known as Mobile IPv6, that allow transparent routing of IPv6 packets to mobile nodes, taking advantage of the opportunities made possible by the design of a new version of IP. In Mobile IPv6, each mobile node is always identified by its home address, regardless of its current point of attachment to the Internet. While away from its home IP subnet, a mobile node is also associated with a care-of address, which indicates the mobile node's current location. Mobile IPv6 enables any IPv6 node to learn and cache the care-of address associated with a mobile node's home address, and then to send packets destined for the mobile node directly to it at this care-of address using an IPv6 Routing header.

[Satyanarayanan 96a]
Satyanarayanan, M.
Fundamental Challenges in Mobile Computing.
In *Proceedings of the Fifteenth ACM Symposium on Principles of Distributed Computing*. ACM, May, 1996.
Philadelphia. Invited paper.

This paper is an answer to the question: *"What is unique and conceptually different about mobile computing?"* The paper begins by describing a set of constraints intrinsic to mobile computing, and examining the impact of these constraints on the design of distributed systems. Next, it summarizes the key results of the Coda and Odyssey systems. Finally, it describes the research opportunities in five important topics relevant to mobile computing: *caching metrics, semantic callbacks and validators, resource revocation, analysis of adaptation, and global estimation from local observations.*

[Satyanarayanan 96b]
Satyanarayanan, M.
Mobile Information Access.
*IEEE Personal Communications*3(1), February, 1996.
The ability to access information on demand when mobile will be a critical capability in the 21st century. In this paper, we examine the fundamental forces at work in mobile computing systems and explain how they constrain the problem of mobile information access. From these constraints, we derive the importance of *adaptivity* as a crucial requirement of mobile clients. We then develop a taxonomy of adaptation strategies, and summarize our research in *application-transparent* and *application-aware adaptation* in the ***Coda*** and ***Odyssey*** systems respectively.

[Satyanarayanan and Noble 97]
Satyanarayanan, M., and B. Noble.
The Role of Trace Modulation in Evaluating Mobile Computing Systems.
In *Proceedings of the Sixth Workshop on Hot Topics in Operating Systems*. May, 1997.
Cape Cod, MA. To appear.
In this paper we put forth the view that trace modulation is an indispensable technique for building and evaluating mobile computing systems. The essence of our solution is transparent, real-time, trace-driven emulation of a target network. Although conceptually simple, trace modulation strikes an attractive balance between the conflicting demands of realism, ease-of-use, and reproducibility. It provides three important benefits. First, it allows control of the complexity of the network environment to which mobile software is exposed. Second, it enables mobile clients to be subjected to reproducible yet realistic network performance. Third, it allows easy exploration of a mobile system in the context of hypothetical mobility patterns and network infrastructures. These benefits suggest that trace modulation will play a central role in the development of future mobile computing systems.

[Satyanarayanan and Spasojevic 96]
Satyanarayanan, M., and M. Spasojevic.
AFS and the Web: Competitors or Collaborators?
In *Proceedings of the SIGOPS96 Workshop*. September, 1996.
Connemara, Ireland.
Over the last few years, the World-Wide Web has risen to dominance as a mechanism for wide-area information access. Its success, unfortunately, has also exposed many of its limitations, such as a tendency to overload the network and servers, limited ability to control access to sensitive data, lack of mechanisms for data consistency, and susceptibility to network and server failures. It is now widely recognized that these problems must be solved if the Web is to continue growing.
With much less fanfare, another world-wide information system, AFS, has also been operating on the Internet. AFS was originally designed to support the file-sharing needs of a campus-sized community, five to seven thousand workstations. In current parlance, AFS was conceived as an information-*sharing* mechanism for an organization's *intra*net. Since then, AFS has evolved to function effectively over the Internet, as well. Many organizations have become part of a single, distributed, UNIX file-name space supported by AFS. As of 1994, this system spanned well over 100 organizations world-wide, with each typically containing many tens or hundreds of clients. Measurements confirm that AFS does indeed function effectively at this scale.
This paper critically compares these two global, information-access mechanisms, beginning with the following questions:

- What are the relative strengths and weaknesses of the two mechanisms?
- Which of these differences are superficial, and which are deep?
- Why is the Web so much more visible and popular?

Our comparison shows that, rather than being competitors, the Web and AFS actually represent complementary technologies that can be used together for mutual advantage. We present real-life examples to confirm that this potential can indeed be realized in practice.

[Satyanarayanan et al. 94a]
Satyanarayanan, M., H. Mashburn, P. Kumar, D.C. Steere, and J.J. Kistler.
Lightweight Recoverable Virtual Memory.
*ACM Transactions on Computer Systems*12(1), February, 1994.

Recoverable virtual memory refers to regions of a virtual address space on which transactional guarantees are offered. This paper describes RVM, an efficient, portable, and easily used implementation of recoverable virtual memory for UNIX environments. A unique characteristic of RVM is that it allows independent control over the transactional properties of atomicity, permanence, and serializability. This leads to considerable flexibility in the use of RVM, potentially enlarging the range of applications than can benefit from transactions. It also simplifies the layering of functionality such as nesting and distribution. The paper shows that RVM performs well over its intended range of usage even though it does not benefit from specialized operating system support. It also demonstrates the importance of intra- and inter-transaction optimizations.

[Satyanarayanan et al. 94b]
Satyanarayanan, M., B. Noble, P. Kumar, and M. Price.
Application-Aware Adaptation for Mobile Computing.
In *Proceedings of the 6th ACM SIGOPS European Workshop*. ACM, September, 1994.
Dagstuhl, Germany.

This paper identifies *application-aware adaptation* as an essential capability of mobile clients, and provides an overview of *Odyssey*, an architecture that supports this capability. Functionality that has hitherto been implemented monolithically must now be split between the operating system and individual applications. The role of the operating system is to sense external events, and to monitor and allocate scarce resources. In contrast, the role of individual applications is to adapt to changing conditions by using the information and resources provided by the operating system.

[Satyanarayanan et al. 95]
M. Satyanarayanan (ed.) et al.
RPC2 User Guide and Reference Manual
School of Computer Science, Carnegie Mellon University, 1995.

This manual describes the programming interface and internal design of RPC2, a highly portable and extensible remote procedure call package for UNIX. RPC2 runs on top of the IP/UDP protocol, and provides support for streaming file transfer, parallel RPC calls, and IP-level multicast. The manual also describes two other packages used by RPC2: RP2Gen, a stub generator, and LWP, a coroutine-based lightweight process package. All the software described in this manual runs at user-level on the UNIX 4.3BSD interface; no kernel modifications are necessary. The software has been ported to a variety of machine architectures (such as IBM-RT, MIPS, Sun2, Sun3, SPARC, and i386) and to variants of the UNIX operating system (such as Mach, SunOS, and AIX).

[Satyanarayanan, Ebling, and Raiff 95]
Satyanarayanan, M., Ebling, M.R., Raiff, J. (Eds).
Coda File System: User & System Administrator's Manual
School of Computer Science, Carnegie Mellon University, 1995.
The Coda File System is a descendant of the Andrew File System. Like AFS, Coda offers location-transparent access to a shared UNIX file namespace that is mapped on to a collection of dedicated file servers. But Coda represents a substantial improvement over AFS because it offers considerably higher availability in the face of server and network failures. The improvement in availability is achieved using the complementary techniques of server replication and disconnected operation. Disconnected operation has proved to be especially valuable in supporting portable computers. This document is a reference manual for Coda users and system administrators.

[Smith and Tygar 94]
Smith, S.W., and J.D. Tygar.
Security and Privacy for Partial Order Time.
Technical Report CMU-CS-94-135, School of Computer Science, Carnegie Mellon University,
April, 1994.
Partial order time expresses issues central to many problems in asynchronous distributed systems, but suffers from inherent security and privacy risks. Secure partial order clocks provide a general method to develop application protocols that transparently protect against these risks. Our previous *Signed Vector Timestamp* protocol provides a partial order time service with some security: no one can forge dependence on an honest process. However, that protocol still permits some forgery of dependence, permits all denial of precedence, and leaks private information. This paper uses *secure coprocessors* to improve the vector protocol: our new *Sealed Vector Timestamp* protocol detects both the presence and absence of causal paths even in the presense of malicious processes, and protects against some privacy risks as well. By solving these previously open security problems, our new protocol provides a foundation for incorporating security and privacy into distributed application protocols based on partial order time.

[Smith et al. 94] Smith, S.W., D.B. Johnson, and J.D. Tygar.
Asynchronous Optimistic Rollback Recovery using Secure Distributed Time.
Technical Report CMU-CS-94-130, School of Computer Science, Carnegie Mellon University,
March, 1994.
In an asynchronous distributed computation, processes may fail and restart from saved state. A protocol for *optimistic rollback recovery* must recover the system when other processes may depend on *lost* states at failed processes. Previous work has used forms of partial order clocks to track potential causality. Our research addresses two crucial shortcomings: the rollback problem also involves tracking a second level of partial order time (potential knowledge of failures and rollbacks), and protocols based on partial order clocks are open to inherent security and privacy risks. We have developed a *distributed time* framework that provides the tools for multiple levels of time abstraction, and for identifying and solving the corresponding security and privacy risks. This paper applies our framework to the rollback problem. We derive a new optimistic rollback recovery protocol that provides *completely asynchronous* recovery (thus directly supporting concurrent recovery and tolerating network partitions) and that enables processes to take full advantage of their maximum potential knowledge of orphans (thus reducing the worst case bound on asynchronous recovery after a single failure from exponential to at

most one rollback per process). By explicitly tracking and utilizing both levels of partial order time, our protocol substantially improves on previous work in optimistic recovery. Our work also provides a foundation for incorporating security and privacy in optimistic rollback recovery.

[Spasojevic and Satyanarayanan 94]
Spasojevic, M., and M. Satyanarayanan.
A Usage Profile and Evaluation of a Wide-Area Distributed File System,
In *Proceedings of the USENIX Winter Technical Conference*. The USENIX Association, January, 1994.
San Francisco. A slightly revised version also appeared as *An Empirical Study of a Wide-Area Distributed File System*, a Transarc Corporation technical report, November 1994.

The evolution of the Andrew File System (AFS) into a wide-area distributed file system has encouraged collaboration and information dissemination on a much broader scale than ever before. In this paper, we examine AFS as a provider of wide-area file services to over 80 organizations around the world. We discuss usage characteristics of AFS derived from empirical measurements of the system, and from user responses to a questionnaire. Our observations indicate that AFS provides robust and efficient data access in its current configuration, thus confirming its viability as a design point for wide-area distributed file systems.

[Steere 95] Steere, D.C.
Using Dynamic Sets to Speed Search in World Wide Information Systems.
Technical Report CMU-CS-95-174, School of Computer Science, Carnegie Mellon University,
March, 1995.

Search on wide area distributed systems is plagued by the high latencies inherent in remote access. A solution is to prefetch information before it is requested by the searcher to hide latency. But this raises the problem of knowing what to prefetch, since fetching data that will not be used can actually hurt performance. This paper proposes extending the UNIX file model to support *dynamic* sets, short-lived and unordered collections of objects created by searchers to hold the results of queries. An object's membership in a set is a *hint* of future access, informing the system that prefetching that object can improve performance. An additional benefit of using set membership as the hint is that it allows the system to determine the order in which objects are returned to the searcher, further increasing the opportunity for performance improvement. This paper presents the design of SETS, a system extension to UNIX to provide dynamic sets. A performance evaluation of SETS shows dynamic sets offer substantial opportunity to reduce the aggregate latency to fetch a group of objects. Experiments on existing world wide information systems show as much as a factor of 8 performance improvement from using sets.

[Steere 96] Steere, D.C.
Using Dynamic Sets to Reduce the Aggregate Latency of Data Access.
PhD thesis, School of Computer Science, Carnegie Mellon University, November, 1996.
Available as technical report CMU-CS-96-194.

Many users of large distributed systems are plagued by high latency when accessing remote data. Latency is particularly problematic for the critical application of search and retrieval, which tends to access many objects and may suffer a long wait for each object accessed. Existing techniques like caching, inferential prefetching, and explicit prefetching are not suited to search, are ineffective at reducing latency for search applications, or greatly increase the complexity of the programming model. This dissertation shows that extending the file system interface to support a new abstraction called *dynamic sets* can address the problem of latency for search without incurring the penalties of the other techniques.

A dynamic set is a lightweight and transitory collection of objects with well-defined semantics. An application creates a dynamic set on-demand to hold the objects it wishes to process. Adding dynamic sets to the system's interface results in two benefits. First, creation of a set discloses the application's interest in the set's members to the system. This allows the system to reduce the aggregate I/O latency of search through prefetching and reordering of requests. Second, dynamic sets provide direct support for accessing and manipulating sets of objects. Thus dynamic sets improve performance and functionality without unduly increasing the complexity of the programming model.

This dissertation describes the design of the dynamic sets abstraction, an implementation which adds dynamic sets to the 4.3 BSD file system interface, and an evaluation of the implementation. The evaluation shows that dynamic sets can substantially reduce I/O latency for search on wide and local area distributed systems and on local file systems. For example, replaying traces of real users searching on the WWW shows that sets can reduce latency by over an order of magnitude across a range of factors.

[Steere 97] Steere, D.C.
Exploiting the non-determinism and asynchrony of set iterators to reduce aggregate file I/O latency.
In *Proceedings of the 16th ACM Symposium on Operating System Principles*. ACM, October, 1997.
St. Malo, France. To appear.

A key goal of distributed systems is to provide prompt access to shared information repositories. The high latency of remote access is a serious impediment to this goal. This paper describes a new file system abstraction called dynamic sets — unordered collections created by an application to hold the files it intends to process. Applications that iterate on the set to access its members allow the system to reduce the aggregate I/O latency by exploiting the non-determinism and asynchrony inherent in the semantics of set iterators. This reduction in latency comes without relying on reference locality, without modifying DFS servers and protocols, and without unduly complicating the programming model. This paper presents this abstraction and describes an implementation of it that runs on local and distributed file systems, as well as the World Wide Web. Dynamic sets demonstrate substantial performance gains — up to 50% savings in runtime for search on NFS, and up to 90% reduction in I/O latency for Web searches.

[Steere and Satyanarayanan 94]
Steere, D.C., and M. Satyanarayanan.
A Case for Dynamic Sets in Operating Systems.
Technical Report CMU-CS-94-216, Computer Science Department, Carnegie Mellon University,
November, 1994.

Recent trends have exposed three key problems in today's operating systems. The first is the emergence of I/O latency as the dominant factor in the performance of many applications. The second is the need to cope with mobile communication environments where bandwidth and latency may be highly variable. The third is the importance of search activity to locating files of interest in a distributed system. In this paper we describe a single unifying abstraction called *dynamic sets*, which can offer substantial benefits in the solution of these problems. These benefits include greater opportunity in the I/O subsystem to aggressively exploit prefetching and parallelism, as well as support for associative naming to complement the hierarchical naming in typical file systems. This paper motivates dynamic sets, presents the design of a system that embodies this abstraction, and evaluates a prototype implementation of the system via measurements and an analytical model.

[Steere and Satyanarayanan 95]
Steere, D.C., and M. Satyanarayanan.
Using Dynamic Sets to Overcome High I/O Latencies During Search.
In *Proceedings of the Fifth IEEE HotOs Conference*. IEEE, May, 1995.
Orcas Island, WA.

In this paper we describe a single unifying abstraction called *dynamic sets* which can offer substantial benefits to search applications. These benefits include greater opportunity in the I/O subsystem to aggressively exploit prefetching and parallelism, as well as support for associative naming to complement the hierarchical naming in typical file systems. This paper motivates dynamic sets and presents the design of a system that embodies this abstraction.

[Steere and Wing 95]
Steere, D.C., and J. Wing.
Specifying Weak Sets.
In *Proceedings of the 15th International Conference on Distributed Computing Systems*. June, 1995.
Vancouver, Canada.

We present formal specifications of a new abstraction, weak sets, that can be used to alleviate high latencies when retrieving data from a wide-area information system such as the World Wide Web. In the presence of failures, concurrency, and distribution, clients performing queries may observe behavior that is inconsistent with the stringent semantic requirements of mathematical sets. For example, an element retrieved and returned to the client may be subsequently deleted before the query terminates. We chose to specify formally the behavior of weak sets because we wanted to understand the varying degrees of inconsistency clients might be willing to tolerate and to understand the tradeoff between providing strong consistency guarantees and implementing weak sets efficiently. Our specification assertion language uses a novel construct that lets us model reachability explicitly; with it, we can distinguish between the existence of an object and its accessibility. These specifications were instrumental in understanding the design space, and we are currently implementing the most permissive of the specifications in several types of UNIX systems.

[Su and Tygar 96] Su, J., and J.D. Tygar.
Building blocks for atomicity in electronic commerce.
In *Proceedings of the Sixth USENIX Security Symposium*, pages 97-104. The USENIX Association, July, 1996.
San Jose, CA.

Atomicity is clearly a central problem for electronic commerce protocols — we can not tolerate electronic commerce systems where money is arbitrarily created or destroyed. Moreover, these atomicity properties should be retained in the event of component failures in distributed systems. In this paper, we enumerate several classes of atomic protocols. We then give two fundamental building blocks for building atomic electronic commerce protocols: encryption-based atomicity and authority-based atomicity. We then illustrate these building blocks by considering variations of payment-server based protocols that use these different building blocks. The results give a contrast to the class of protocols that we have previously examined in our work with NetBill.

[Tygar 97] Tygar, J.D.
Atomicity in electronic commerce.
In P. Denning and D. Denning (editors), *Networks Under Attack*. ACM Press and Addison-Wesley, 1997.
Early versions appeared in *Proceedings of the Fifteenth Annual ACM Symposium on Principles of Distributed Computing*, May, 1996, as a Keynote Paper and as technical report CMU-CS-96-112.

There is a tremendous demand for the ability to electronically buy and sell goods over networks. Electronic commerce has inspired a large variety of work. Unfortunately, much of that work ignores traditional transaction processing concerns — chiefly atomicity. This paper discusses the role of atomicity in electronic commerce. It then briefly surveys some major types of electronic commerce pointing out flaws in atomicity. We pay special attention to the atomicity problems of proposals for digital cash.

[Tygar and Yee 94]
Tygar, J.D., and B.S. Yee.
Dyad: A System for Using Physically Secure Coprocessors.
Journal of the IMA 1(1), January, 1994.

Physically secure coprocessors, as used in the Dyad project at Carnegie Mellon University, provide easily implementable solutions to perplexing security problems. This paper presents solutions to five problems: (1) protecting the integrity of publicly accessible workstations; (2) tamperproof accounting/audit trails; (3) copy protection; (4) electronic currency without centralized servers; and (5) electronic contracts.

[Tygar, Yee, and Heintze 96]
Tygar, J.D., B.S. Yee, N.C. Heintze.
Cryptographic Postage Indicia.
In *Proceedings of ASIAN'96: The Asian Computer Science Conference*. December, 1996.
Keynote paper, Springer-Verlag. Earlier versions appeared in *Proceedings of SECURICOM'96* and as technical report CMU-CS-96-113.

We apply cryptographic techniques to the problem of fraud in metered mail. We describe a mail system that combines off-the-shelf barcode technology, tamper-proof devices, and cryptography in a fully-integrated secure franking system. This system provides protection against:

1. Tampering with postage meters to fraudulently obtain extra postage
2. Forging and copying of stamps
3. Unauthorized use of postage meters
4. Stolen postage meters.

We provide detailed justification for our design, and discuss important tradeoffs involving scanning strategies, encryption technology and 2-D barcode technology. The US Postal Service' recent Information Based Indicia Program (IBIP) announcement adopted the principal design features of our model.

[Yee 94] Yee, B.S.
Using Secure Coprocessors.
PhD thesis, School of Computer Science, Carnegie Mellon University, May, 1994.
Available as technical report CMU-CS-94-149.

How do we build distributed systems that are secure? Cryptographic techniques can be used to secure the communications between physically separated systems, but this is not enough: we must be able to guarantee the privacy of the cryptographic keys and the integrity of the cryptographic functions, in addition to the integrity of the security kernel and access control databases we have on the machines. Physical security is a central assumption upon which secure distributed systems are built; without this foundation even the best cryptosystem or the most secure kernel will crumble. In this thesis, I address the distributed security problem by proposing the addition of a small, physically secure hardware module, a secure coprocessor, to standard workstations and PCs. My central axiom is that secure coprocessors are able to maintain the privacy of the data they process.

This thesis attacks the distributed security problem from multiple sides. First, I analyze the security properties of existing system components, both at the hardware and software level. Second, I demonstrate how physical security requirements may be isolated to the secure coprocessor, and showed how security properties may be bootstrapped using cryptographic techniques from this central nucleus of security within a combined hardware/software architecture. Third, I demonstrate the feasibility of the secure coprocessor approach, and report on my implementation of this combined architecture on top of prototype hardware. Fourth, I design, analyze, implement, and measure performance of cryptographic protocols with super-exponential security for zero-knowledge authentication and key exchange. Last, I show how secure coprocessors may be used in a fault-tolerant manner while still maintaining their strong privacy guarantees.

[Yee and Tygar 95]
B. Yee and J.D. Tygar.
Secure Coprocessors in Electronic Commerce Applications.
In *Proceedings of the USENIX Workshop on Electronic Commerce*. The USENIX Association, July, 1995.

Many researchers believe electronic wallets (secure storage devices that maintain account balances) are the solution to electronic commerce challenges. This paper argues for a more powerful model — a secure coprocessor — that can run a small operating system, run application programs, and also keep secure storage for cryptographic keys and balance information.

We have built a system called Dyad, on top of a port of the Mach 3.0 microkernel to the IBM Citadel secure coprocessor. This paper describes the abstract architecture of Dyad and a general discussion of secure coprocessor implementations of a variety of electronic commerce applications:

1. Copy protection for software
2. Electronic cash (including a critique of proposed solutions for point-of-sale electronic wallet systems)
3. Electronic contracts
4. Secure postage