
A new model in the use of computer and communications technology

Vision, Issues, and Architecture for Nomadic Computing

**RAJIVE BAGRODIA, WESLEY W. CHU, LEONARD KLEINROCK
AND GERALD POPEK**

The pervasive nature of computing and communications is bringing about a paradigm shift in the way these technologies are being used. The shift is in the direction of nomadic computing and communications. *Nomadcity* refers to the system support needed to provide a rich set of computing and communication capabilities and services to the nomad as he or she moves from place to place in a transparent, integrated and convenient form. This new paradigm is already manifesting itself as users travel to many different locations with laptops, personal digital assistants (PDAs), cellular telephones, pagers, and so on.

In this article we discuss a vision of nomadcity as well as open issues, and architecture and design concerns that must be addressed as we bring about the system support necessary for nomadcity.

Since the beginnings of telecommunication technology, we have witnessed a number of major shifts in the application of communications to the needs of our society and industry. In that process, we have seen the marriage of wireline and wireless technologies, of analog and digital technologies, of voice, data, video, image, fax, graphics, and so on to create a computer communications infrastructure that spans the globe and serves billions of people.

We are now in the midst of an accelerating groundswell in this field of computer-communications in its largest sense (i.e., not simply the wires and networks, but also the infrastructure, the middleware, the applications, the uses and users of the technology) [1]. Witness the fact that the Internet is now a household word. The use of the Worldwide Web (WWW) is growing faster than any other application we have ever witnessed in 25 years of networking (from the day the ARPANET was born at UCLA in September 1969 up to the present); and the WWW is still in its infancy!

Most people function in a world where their desktop computing appliance is connected through a corporate or private network to a server located "someplace else" (possibly in a dungeon in some mysterious basement). It is usually assumed that the connectivity provided by this network is reliable and

of high bandwidth (typically megabits per second). In fact, however, many of us are *nomads*, moving between office, home, airplane, hotel, automobile, branch office, conference room, bedroom, and so forth. In so doing, we often find ourselves decoupled from our "home base" computing and communications environment. As we move around, we find enormous variations in the computing platform to which we have access (advanced workstation, Pentium-class PC, laptop, palmtop), in the quality of available printers and displays, in the communication device we use (Ethernet attachment, PCMCIA card, analog modem card, CDPD wireless data channel), as well as in the communication bandwidth that is available to us (wireless at 9.6 kb/s, analog modem at 28.8 kb/s, ISDN at 128 kb/s, Ethernet at 10 Mb/s, ATM at 25 or 155 Mb/s). In addition, we may choose to do computing and/or communicating while we are on the move.

The variety of portable *computers* is impressive, ranging from laptop computers, to notebook computers, to PDAs, to smart credit card devices, to wristwatch computers, and so on. In addition, the *communication* capability of these portable computers is advancing at a dramatic pace from high-speed modems, to PCMCIA modems, to e-mail receivers on a card, to spread-spectrum handheld radios, to CDPD transceivers, to portable GPS receivers, to gigabit satellite access, and so on.

The combination of portable computing with portable communications is changing the way we think about information processing. We now recognize that access to computing and communications is necessary not only from one's "home base," but also while one is in transit and when one reaches one's destination [2].¹

Even without portable computers or communications, there are many who travel to numerous locations in their business and personal lives, and who require that access to computers and communications be available at their destination when they arrive there. Indeed, a move from one's desk to a conference table in one's office constitutes a nomadic move since the computing platforms and communications capability may be considerably different at the two locations.

A fundamental way in which nomadic computing differs from conventional operation is the huge variability in connectivity to the rest of one's computing environment. That level

This work was supported by the Advanced Research Projects Agency, ARPA/CSTO DABT-63-C-0080 "Transparent Virtual Mobile Environment-Travler"; N00174-91-C-0107, "Large Scale Filing and Data Management Environments"; ARPA/ISO F30602-94-C-0207A1, "Scalable and Extensible Cooperative Information Systems."

¹ Moreover, one may have more than a single "home base"; in fact, there may be no well-defined "home base" at all.

of connectivity often includes extended periods of low bandwidth or no communication at all. Since many users and programs make intermittent, but nevertheless essential, use of "off-machine" information and services, they will be unable to operate effectively unless extraordinary steps are taken. The goal of "transparent virtual networking" is precisely to permit users and programs to be as effective as possible in this environment of uncertain connectivity without changes to the manner in which they operate. That is, transparent virtual networking makes the sometimes-connected computer operate in the same way and as effectively in standalone operation as when it is connected to the organization's information network.

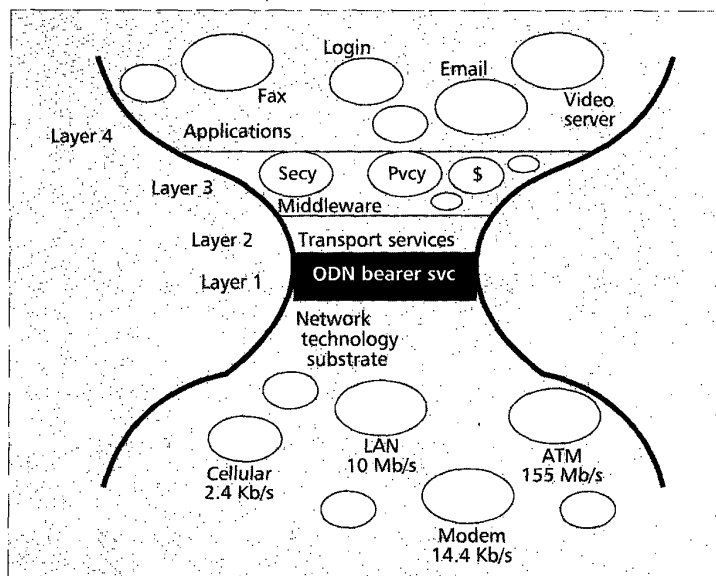
These ideas form the essence of a major shift to *nomadicity* (nomadic computing and communications) that we choose to address in this article. As stated earlier, nomadicity may be defined as the system support needed to provide a rich set of computing and communication capabilities and services to nomads as they move from place to place in a transparent, integrated and convenient form.

In this article we discuss our vision of the nomadic computing environment from a systems perspective, with specific attention to the software system issues that must be addressed in the design of a transparent virtual networking interface to support the nomadic user. The following section proposes a nomadic system architecture and reference model. The third and fourth sections present an overview of the file systems and database issues in supporting nomadicity, and the fifth section describes the relevant performance issues.

Nomadic Computing

We are interested in those capabilities that must be put in place to support nomadicity.² The desirable characteristics for nomadicity include independence of location, motion, computing platform, communication device, and communication bandwidth, and the widespread presence of access to remote files, systems and services. The notion of independence here does not refer to the quality of service one sees, but rather to the perception of a computing environment that *automatically adjusts* to the processing, communications and access available at the moment. For example, the bandwidth for moving data between a user and a remote server could easily vary from a few bits per second (in a noisy wireless environment) to hundreds of megabits per second (in a hard-wired ATM environment); or the computing platform available to the user could vary from a low-powered PDA while in transit to a powerful supercomputer in a science laboratory. Indeed, today's systems treat radically changing connectivity or bandwidth/latency values as *exceptions* or *failures*; in the nomadic environment, these must be treated as the *usual* case. Moreover, the ability to accept partial or incomplete results is an option that must be made available due to the uncertainties of the informatics infrastructure.

Many people think of wireless communications as the enabler, or even the characterization, of nomadicity. The view we take is far



■ Figure 1. The open data network architecture.

broader than that. It is true that wireless communications may be a component of nomadicity, but it is not a necessary one. When people travel across the country and check into a hotel, they have made a nomadic move as they attempt to connect their laptop via a wireline analog modem to the network infrastructure; no wireless communication is involved there. Indeed, we emphasize that much of the action for nomadic computing takes place at the middleware level of the commonly accepted layered architecture. One version of that layered architecture, as put forward in [1], is shown in Fig. 1. In this figure we clearly see the middleware level between the applications and transport levels. Most of the functionality we describe will be found at this level. However, some of the interfaces to the network technology substrate refer to functionality at the bearer service level.

System Parameters

Some of the key system parameters with which one must be concerned include bandwidth, latency, reliability, error rate, delay, storage, processing power, interference, interoperability, user interface and so forth. These are the usual concerns for any computer-communication environment, but what makes them of special interest for us is that the values of these parameters change dramatically (and sometimes suddenly) as the nomad moves from location to location. In addition, some totally new and primary concerns arise for the nomad such as weight, size and battery life of his/her portable devices. And the bottom line consideration in many nomadic applications is cost.

Why Nomadic Computing Is of Interest

There are a number of enchanting reasons why nomadicity is of interest. These include:

- Nomadicity is clearly a *newly emerging technology* with which users are already surrounded.
- We judge it to be a *paradigm shift* in the way computing will be done in the future.

² Some of the ideas presented in this section were developed with the Nomadic Working Team (NWT) of the Cross Industrial Working Team (XIWT); one of the present authors (LK) is the chairman of the NWT [3]. In addition, portions of this section were described in [4].

Mode of operation	Number of updates	Number of conflicts	Conflict rate in percent	User-visible conflict rate
Disconnected, shared	1,114,855	273	.0245%	.0052%
Disconnected, private	387,523	106	.0274%	.0012%
Office, shared	6,316,331	66	.0010%	.0005%
Office, private	6,286,754	44	.0007%	.0003%
Network, shared	36,778	0	0%	0%

■ **Table 1.** Conflict statistics for different types of volumes.

- Information technology trends are *moving in this direction*.
- Nomadic computing and communications is a *multidisciplinary* and *multi-institutional* effort.
- It has a huge *potential for improved capability* and convenience for the user.
- At the same time, it presents at least as huge a problem in *interoperability* at many levels; nomadicity makes all the usual problems much harder.
- Contributions from many investigations of nomadicity will be at the *middleware* level.
- The products that are beginning to roll out have a *short-term focus*; however, there is an enormous level of interest among vendors (the computer manufacturers, the networking manufacturers, the carriers, etc.) for long-range development and product planning, much of which is *now underway*.
- Whatever work is accomplished now will certainly be of *immediate practical use*.

Systems Research Issues

There are fundamental new research problems that arise in the development of a nomadic architecture and system. Let us consider a sampling of such problems.

Nomadic System Architecture – One key problem is to develop a full *system architecture and set of protocols* for nomadicity. These should provide for a transparent view of the user's dynamically changing computing and communications environment. The protocols must satisfy the following kinds of requirements:

- Enable interoperation among many kinds of infrastructures (e.g., wireline and wireless)
- Deal with unpredictability of user behavior, network capability and computing platform
- Provide for graceful degradation
- Scale with respect to heterogeneity, address space, quality of service (QoS), bandwidth, geographical dimensions, number of users, and so on
- Provide the user with an indication of the QoS he or she is currently receiving, the size of files about to be downloaded and so on
- Provide for integrated access to services
- Allow for ad hoc access to services
- Deliver maximum independence between the network and the applications from the users' viewpoint as well as from the development viewpoint
- Relieve the user from reconfiguring or rebooting each time the mode of communication access changes

- Match the nature of what is transmitted to the bandwidth availability (i.e., compression, approximation, partial information, etc.)
- Enable cooperation among system elements such as sensors, actuators, devices, network, operating system, file system, middleware, services, applications and so forth

Some of the system developments that can help in providing these requirements are:

- An integrated software framework which presents a common virtual network layer
- Appropriate replication services at various levels
- File synchronization
- Predictive caching
- Consistency services
- Intelligent (adaptive) database management
- Location services (to keep track of people and devices)
- Discovery of resources

The first five items are discussed further in the third section, and the last three in the fourth section below.

Nomadic Reference Model – A second research problem is to develop a reference model for nomadicity which will allow a discussion of its attributes, features and structure in a consistent fashion. This should be done in a way that characterizes the view of the system as seen by the user, and the view of the user as seen by the system. The dimensions of this reference model might include:

- System state consistency (i.e., is the system consistent at the level of e-mail, files, database, applications, etc.)
- Functionality (this could include the bandwidth of communications, the nature of the communication infrastructure, the quality of service provided, etc.)
- Locality or awareness (i.e., how aware is the user of the local environment and its resources, and how aware is the environment of the users and their profiles)

Modeling of the Nomadic Environment – A third research problem is to develop *performance models* of the nomadic environment. These models will allow one to study the performance of the system under various workloads and system configurations as well as to develop design procedures.

In the fifth section below, we elaborate on simulation as a tool for the modeling and evaluation of nomadic systems.

The Multidisciplinary Nature of Nomadicity

As mentioned above, the area of nomadic computing and communications is multidisciplinary. A list of the disciplines that contribute to this area are (in top-down order):

- Advanced applications, such as multimedia or visualization
- Database systems
- File systems
- Operating systems
- Network systems
- Wireless communications
- Low-power, low-cost radio technology
- Micro-electro-mechanical systems (MEMS) sensor technology

- MEMS actuator technology
- Nanotechnology

The reason that the last three items in this list are included is that we intend the nomadic environment to include the concept of an *intelligent room*. Such a room has embedded in its walls, furniture, floor and so on all manner of sensors (to detect who and what is in the room), actuators, communicators, logic, cameras and other tools. Indeed, one would hope to be able to speak to the room and say, for example, "I need some books on the subject of spread-spectrum radios," and perhaps three books would reply. The replies would also offer to present the table of contents of each book, as well, perhaps, as the full text and graphics. Moreover, the books would identify where they are in the room, and, if such were the case, might add that one of the books is three doors down the hall in a colleague's office!

Other Issues

There are numerous other systems issues of interest that we have not addressed here. One of the primary issues is that of security, which involves privacy as well as authentication. Such matters are especially difficult in a nomadic environment since the nomad often finds that his computing and communication devices are outside the careful security walls of his home organization. This basic lack of physical security exacerbates the problem of nomadicity.

Another entire set of nomadic issues we have not addressed is that of wireless networking. Access to wireless communications provides two capabilities to the nomad. First, it allows the nomad to communicate from various (fixed) locations without being connected into the wire-line network. Second, it allows the nomad to communicate while traveling. These issues are discussed in detail in a number of recent publications [5-8].

Transparent Virtual Networking, Information Storage, and Mobility

The emergence of mobile computing from tethered operation promises to alter the way computing is done to a similar degree that the emergence of workstations from time-sharing changed the computational experience.

Clearly, the first step toward accomplishing the goal of transparent virtual networking is to ensure that the information which the user needs to compute is accessible to the mobile machine when needed. If the necessary information is not accessible, there is little more that can be done. Thus, we view the provision of information as a key prerequisite for a successful transparent virtual network. In this section, then, we concentrate on what can be done to supply the user's desired information ahead of time, so that poor or nonexistent communication does not interfere with continued operation.

A variety of investigations, most recently at Locus Computing Corporation [9], have demonstrated that for most users, today's PC disks are

Environment	CPU					
	0.00	(0.01)	0.050	.02	(0.03)	0.07
Productivity	0.00	(0.01)	0.050	.02	(0.03)	0.07
Programming	0.01	(0.02)	0.060	.11	(0.09)	0.28
Commercial	4.35	(4.75)	16.29	12.32	(8.57)	24.57

■ Table 2. Total conflict rates in Locus study of commercial users.

quite capable of storing the information and tools the user will need for the next several weeks. An important question, of course, is *which* information to place in local copies. Furthermore, once copies are made of the desired information, it is obviously desirable that the user be able to manipulate the local copy as if it were the original, including making changes. This update of replicated information leads to the potential for conflicting updates, which must be effectively reconciled if transparent virtual networking is to be successful.

In fact, the need to replicate information storage introduces a number of issues to address. Below, we survey some of those issues, and briefly consider approaches to their solution.

Optimistic Update

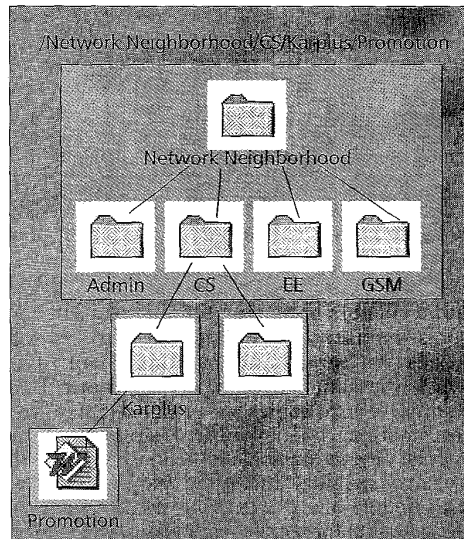
It is a conventional wisdom that, at least for many files and applications, concurrent update is quite rare [9-13]. If this pattern is generally true in practice, and those rare problems which do occur can be corrected by reasonable means, then this entire approach has promise. Fortunately, data collected in the studies at UCLA and Locus indicate that conflicts are generally rare in both academic and commercial applications; and that those which do occur can often be reconciled automatically, without any user intervention. Systems that support replication, independent update, and subsequent reconciliation are called "optimistic replication systems." The UCLA results reflect data access patterns collected over a nine-month period from an optimistically replicated file system employed for all of their work by the researchers who developed it. Patterns of use were reasonably representative of academic activity: group software development, e-mail, document preparation, games, Internet use and the like. The Locus data reflects three-month-long traces of network file access patterns by a 12-person finance and administration department. Tables 1 and 2 summarize some of the UCLA and Locus results.

Overall, .0035 percent of all updates to non-directory files resulted in conflicts; .0018 percent of all name creations led to name conflicts. Our statistics did not collect the number of potential directory conflicts fixed internally by automatic resolution code.

These results are quite encouraging, lending credence to the overall approach. Nevertheless, it is still necessary to actually cause the desired data to reside on the mobile machine, a problem to which we now turn.

Predictive File Replication (Caching)

The problem being considered is to *anticipate* what information the user will require, while



■ Figure 2. Network naming navigation.

connectivity is available, so that the information to be needed in the future can be transferred. In some respects, this problem is similar to that addressed by predictive caching methods used in virtual memory systems and databases, which bring information into main memory from rotating storage in anticipation of its being required in order to improve performance and response time.

However, in this case, the scope of the problem is quite different. First, the cost of a fault is far greater. In the main memory case, a page fault can be satisfied in a few tens of milliseconds. In the mobile case, if information is lacking it may be hours or days before it can be

We expect that an effective predictive caching system can be developed, which together with the user can effectively decide what files should be replicated for the mobile user.

acquired, leaving little choice but to terminate the program. Hence, there is an enormous premium on effective prediction. Fortunately, there are a lot of hints available, since the mobile machine can record many interesting aspects of the user's behavior over a lengthy period of time, even noting periodic behavior (e.g., reviewing financial information monthly). It is relatively easy to predict, if a user touches a particular spreadsheet, that a family of related spreadsheets (which may be linked together) are likely to be of interest too. Hence, collections of files may be clustered, and when a few of the members are touched, the system can gather them all together. However, this clustering approach cannot help when the user starts a new task. Fortunately, the studies cited earlier indicate that such *attention shifts* are also relatively rare. If each cluster of information is viewed as being associated with a user task, the job at hand is to identify the members of a task and collect the relevant tasks for local storage.

Of course, it is critical that the system permit the user to indicate interest in certain tasks, as the user will always know better what his or her intent is likely to be. Note, however, that the user is *not* best able to specify the content of a task's information cluster, as users are often not even aware of what files, utilities, linked information, and the like are in use on his or her behalf [11]. Fortunately, the system can help considerably in this respect [14]. Therefore, we expect that an effective predictive caching system can be developed, which together with the user can effectively decide what files should be replicated for the mobile user.

Network Naming Navigation

Once the desired files are resident on the mobile computer, the issue arises of how they are to be named. In the example below, the file named Promotion is stored on a file server called CS. If that file were replicated on Karplus' mobile machine, it should be named as in Fig. 2.

Unfortunately, the shaded portion of the naming hierarchy in Fig. 2 is not really stored anywhere. When the user selected the Network Neighborhood on the Windows 95 mobile machine, for example, network queries were dynamically conducted to "materialize" the list of available servers, which were then represented as icons in a folder. When the mobile machine is not connected to the network, the server folders won't be available. Nevertheless, the file Promotion must be named in the same manner (i.e., by the same four-part path name). Otherwise, applications (as well as the user!) may not know how to refer to it.

Obviously, file structure information, database metadata, and other information will need to be captured by the mobile system to provide transparent virtual network support for replicated storage.

Operation Queueing

Clearly there are numerous operations which cannot be conducted without established connectivity. E-mail and printing support are oft-cited examples. Today, many e-mail systems are capable of storing copies of messages on the mobile machine, allowing users to read, sort, and answer their mail while disconnected. Those replies are actually sent when connectivity is available. Windows 95 permits one to queue print requests, and with suitable networking support will automatically place those requests upon reconnection.

A general-purpose queueing service is an essential part of a transparent virtual network so that the wide set of requests which may be issued during disconnected operation [15] can be subsequently completed upon reconnection: faxes, database queries and updates, a variety of structured transactions and so forth. However, there is more to a general queue service than is apparent at first. Client-server applications must be tolerant of their server being temporarily unavailable. Some of the underlying tools which make such a queueing service robust are reminiscent of batch job control languages, a level of complexity to be avoided.

Nevertheless, the goal here is for the user to be able to conduct many operations without

regard for the fact that some of them will take place later, when connectivity allows. The e-mail and printing examples illustrate how valuable such a transparent model is to mobile users.

Optimistic Database Fragments

In the earlier discussion, we said that concurrent update was demonstrated to be rare in the academic and commercial cases we examined. However, there is one major exception to that generalization: shared on-line databases. If the database is viewed as a single object, there would certainly be many conflicting updates if it were replicated. In many cases, such replication is infeasible anyway, due to sheer size.

It is well known, of course, that databases exhibit well-defined hotspots of activity. Furthermore, early evidence suggests that users tend to employ a small, well-defined, predictable portion of such databases. As a result, one has the opportunity once again to simulate the availability of large data collections for the user by predictively storing relevant database fragments on the mobile machine in a local database, so long as the tool the user employs to access that data is the same whether accessing the local copy or the corporate version. We wish to give the user the illusion shown on the left in Fig. 3, while the actual situation is indicated on the right.

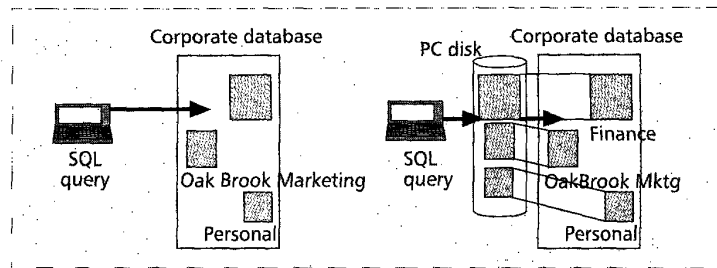
That is, the standard language query (SQL), perhaps issued by a convenient graphical user interface tool, actually queries the information stored on the PC disk; only later are updates posted to the corporate database.

Mobile Systems Management

Systems administration and management have been recognized as dominant components of overall computing cost. Novell recently estimated that 70 percent of the total cost of ownership of computers that are part of a corporate network lay in their administrative support rather than the hardware or software purchases themselves [16]. Even for higher-priced workstations, Sun Microsystems estimated that at least half of the cost is administrative in nature [17].

Mobile computing exacerbates this problem. Unlike tethered networks, the mobile machine may often be disconnected when problems or administrative actions arise. The problem once again concerns access to information. In this case, the user's environment (various settings, program versions, etc.), stored on the mobile machine, is not available to the administrator to interrogate, diagnose, or update. Conceptually, this information should be optimistically replicated as well, stored in the corporate network so that the administrator can examine and alter it. Upon reconnection, those changes will be posted to the user and his environment updated. Changes that the user initiated would be transmitted in the other direction. Conflicting changes would be dealt with on an exception basis.

While the specifics of mobile systems management are of course more complex than these comments indicate, the point once again is that a great deal can be accomplished if the relevant information is replicated and thus available where needed. Without it, solution is much more difficult.



■ Figure 3. Optimistic databases: user view and system operation.

Objects and Multimedia

Multimedia presents special problems. Imagine the simple case of an embedded video clip in the first of a set of e-mail messages queued waiting for receipt by a traveling user. The intended receiver's machine subsequently connects, in this case over a low-speed line, to collect that mail. Unfortunately, the machine spends all available resources attempting to retrieve the first message. Nothing of value is achieved, as transmission of the video clip exceeds available connect time.

Obviously, priorities are essential in this case, so the clip is transmitted last, if at all. The task is somewhat more difficult, however, than it may at first appear. Only the relevant application can understand where in the e-mail message the video clip begins and ends; indeed, even where the boundaries among different messages fall. There may be little the unaided system can do.

The more the structure of the data is visible to the underlying system, the more likely that automatic solutions can be provided. The visibility of the modular structure of information, and associated access and manipulation routines, is one of the attractions of modern object-oriented programming methods. In the example above, one could choose to skip the clip, or transmit a degraded image, and/or omit frames.

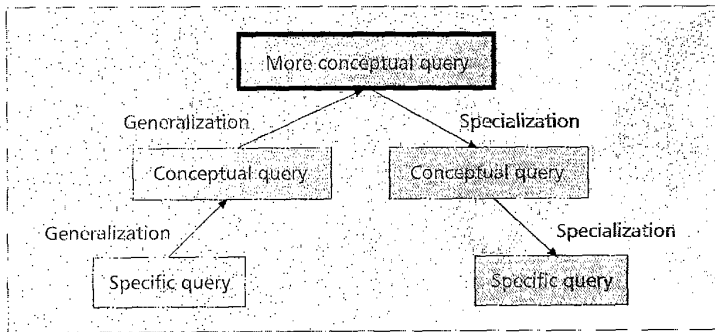
Summary

The purpose of the discussion in this section was not to suggest that transparent virtual networking is a reality. Instead, we merely wish to convey to the reader that there is ample basis for belief that it may well be feasible to place the information the user needs on the mobile machine, in large measure automatically. The presence of that information permits the development of systems architectures and modes of operation that allow mobile machines to behave as if they were fully and continually connected. We believe that such a transparent virtual networking paradigm is an exceedingly valuable model for mobile computing in general.

Intelligent Databases

Intelligent Databases to Support Nomadicity

The vast number of clients and servers interconnecting in a nomadic information system changes the operating environment (e.g., bandwidth, accessing heterogeneous data sources) and also causes frequent intermittent connectivity. As a result, new problems occur in accessing data from different database servers due to:



■ Figure 4. Query generalization and specialization process.

- Data granularity: Different servers present data in different formats.
- Data consistency: Different servers have different data consistency requirements.
- Data schema: Different servers are organized in different schema and abstractions.
- Bandwidth limitation: Different operating environments support different bandwidths.

Database servers in a nomadic environment need to have a resource discovery facility (RDF) and a set of intelligent agents to provide searching and adaptive matching of resources to satisfy the query conditions and current user operating environment (e.g., bandwidth, computing resources). Because there are a large number of servers in the system, it is almost impossible for users to be familiar with the data schemas at all of them. The user may not be able to reach the requested server because of connectivity; the user often may not even know the availability and existence of all the servers in the system. Therefore, based on context, user profile and current user operating environment, adaptive agents must search through the RDF and match the best servers to process the query due to data consistency or granularity problems. The answer may be approximate. Furthermore, the user may only get a summary answer if there is bandwidth limitation. In the following sections, we discuss the functionality of the RDF and the characteristics of the adaptive agents.

The Resource Discovery Facility

An RDF is one of the key components in a nomadic information system. The RDF consists of an intelligent dictionary/directory (IDD) and a collection agent to maintain the contents of the RDF. The IDD contains such information as resource characteristics, locations and user profiles of all the clients, servers, and middleware in the information system. Clearly, this is a vast collection of data and knowledge that covers hardware (processors, peripheral devices, modems, networks, etc.), software (operating systems, databases, knowledge bases), and operating environment (stationary, mobile, airplane, space, underwater, etc.). The system characteristics can be channel bandwidth, protocol used, processor or operating system types and so forth. Locations can be specified by network and/or physical addresses. User profiles can be specified according to the user's profession, affordability, requirements (e.g., security, speed, accuracy) and so on. Clearly, the acquisition, logical organiza-

tion, and allocation of this information affects the efficiency and timely retrieval of the desired information for processing and transferring to the clients.

The RDF can be organized into a collection of intelligent distributed dictionaries/directories with their information stored in databases and knowledge bases. One way to organize the information is to store it according to classifications (e.g., hardware, software, environment and user profile), which eases maintenance and updates.

Acquiring Dictionary/Directory Databases and Knowledge Bases – An information-collecting agent in each resource discovery facility collects all the interoperability relevant information in the following ways:

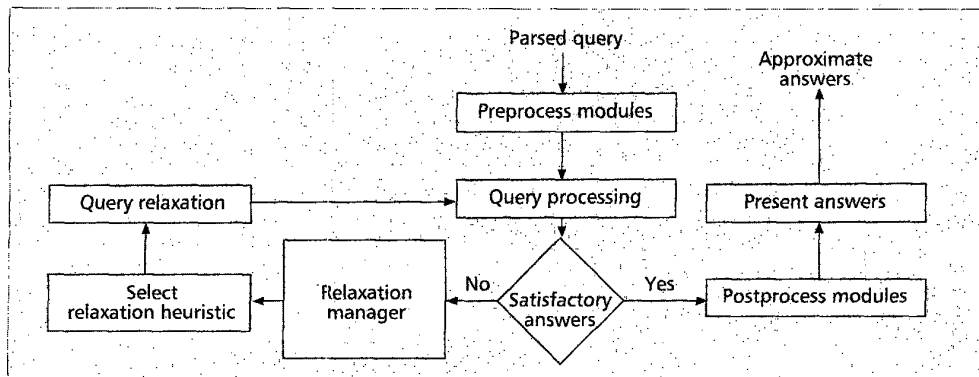
1. By determining the set of interoperability information for all the resources (hardware, software, files, databases, etc.), the information can be queried by the agent periodically or reported from the clients, servers and middleware whenever updates occur.

2. The profiles of clients, middleware and servers can be downloaded to the repository center and the collection agent automatically. The hardware and software as well as the user profile are precoded in a terminal as the user logs on the system. This information is automatically downloaded to the collection agent. The user can review the profiles and edit them to fit his or her requirements and environment.

3. The relevant interoperability information can be derived from a variety of documentation sources.

Conventional data dictionaries/directories include data descriptions and definitions, and their respective storage locations. However, when we interconnect with different servers or multiple heterogeneous autonomous databases, mismatch problems are created. The problems include instance mismatch, unit conversion (value interpolation), context and structure mismatch, and semantic representation mismatch. Although there are commercial dictionary/directory products available (e.g., Gateway for database integration), they are for specific database applications and inadequate for use in a nomadic information system. An IDD uses lexical and synonym transformation to provide meta-thesaurus information that resolves such conflicts. Furthermore, different users and applications require varying levels of information and presentation. Adaptive agents can be used to perform the approximate matching and/or query relaxation for users when the exact answer is not available. Current commercial data dictionaries/directories have very limited functionality and are restricted to a particular fixed application or system. It is essential in nomadic systems to extend and add the additional approximate matching capability so that the dictionary/directory from different sources can be dynamically integrated.

Resources Discovery Facility Architecture – For availability and response time reasons, the RDF should be replicated; however, increasing replication also increases the cost of update maintenance. Therefore, one could organize the data



■ Figure 5. Query (e.g., resource characteristics) relaxation process.

item in the RDF not only according to their functionality but also according to their update rates. To reduce communication and update costs, one should minimize the number of copies for those data items that have high update rates. To increase the availability and response time, one should increase the replications of those data items that are frequently asked but seldom updated (e.g., locations of the RDF, database servers, version software). Furthermore, this will also reduce the communication traffic. As a result, the RDF should be organized into a hybrid directory structure. The level of replication should be different for different data items and be based on the data update rate/access rate [18].

Adaptive Agents

Adaptive agents [19–22] provide approximate matching and/or query relaxation, association and explanation during the information integration and materialization process. The *relaxation agent* performs query relaxation and/or approximate matching of information at different abstraction levels from heterogeneous sources. The *association agent* provides relevant information to the users or applications that the user does not explicitly (or does not know how to) ask. The *explanation agent* [23] provides explanation to the user of the relaxation process and the quality (security, accuracy, etc.) and cost (time, price, etc.) of the answers. The agents have interactive capabilities that enable the user to dialogue with them to derive the user's goal.

Relaxation Agent – The relaxation agent utilizes a knowledge structure — type-abstraction hierarchies (TAHs) — that provides multilevel knowledge representation. For each database attribute type, its TAH is a classification hierarchy of its possible values [24]. Operations are provided to traverse the hierarchy, such as generalization (up) and specialization (down). The agent is able to transform lower-level, heterogeneous distributed data and knowledge into a higher level of abstraction, as shown in Fig. 4. Query condition values are relaxed to their semantic neighbors until an approximate answer is produced.

Explicit relaxation operations can also be provided by the user such as *near-to*, *similar-to*, approximate range of attribute value and so on. The *near-to* operation is for numerical attribute

values (e.g. *near-to* location x) while *similar-to* can be applied to a set of attributes for both numerical and nonnumerical values. Weights can be assigned to each attribute in accordance with its relative importance among the set of attributes. The weighted mean error can be used as the similarity measure.

Knowledge Discovery from Databases – The TAH can be generated automatically from relational databases for both numerical and nonnumerical attribute values [19, 25]. It can be applied to structure data (e.g., relational database) as well as to feature-based image data (e.g., spatial and temporal features). The computation time required to generate the TAH

It is essential in nomadic systems to extend and add the additional approximate matching capability so that the dictionary/directory from different sources can be dynamically integrated.

increases approximately on the order of the square of the number of tuples. The TAHs can be based on multiple attributes to classify multiple dependent attributes. Furthermore, the generated TAH can be edited (e.g., add, delete, naming) by the domain expert via the knowledge editor. We have generated the TAHs for an entire transportation database which consists of 94 relations. The size of these relations range from 84 attributes and 8464 tuples to 12 attributes and 195,000 tuples, and takes less than a few hours of CPU time on a Sun Sparc 20 Workstation.

Relaxation Control – The purpose of the relaxation manager is to provide efficient, accurate and flexible query relaxation control. The relaxation manager transforms the original query into a relaxed query by traversing the TAHs for the attributes listed in the condition clause of the query. Relaxation query operators and default relaxation control strategies provide information to assist the relaxation manager in TAH traversal and subsequent relaxed query formation (Fig. 5).

Relaxation control is necessary to control the quality of the answers. For example, if each TAH is traversed all the way to the root node,

then all the tuples in the database for those attributes will be returned. The following list of relaxation control operators provides flexible relaxation control to reduce processing time and increase answer accuracy.

- Relaxation order: specifies the order in which attributes should be relaxed.
- Unacceptable Lists: enables the user to control the list of attributes that cannot be relaxed. The user can either specify this list in the query or interactively with the agent. An experienced user can modify (trim) the TAH.
- Preference lists: The user specifies preferred values for the attribute. If no answers are found, then the query is relaxed in the order given in the preference list.
- Relaxation levels: The user may specify the level to which an attribute should be relaxed before relaxation of other attributes is considered.

The adaptive agents are able to assess the users' operating environment and set the relaxation control parameters to satisfy his/her information requirements (e.g., provide summary answers if the complete answer is either too time-consuming to transmit or unavailable).

Association Agent – Associate query answering provides relevant information that the user does not know specifically or does not know how to request. This additional information reduces the number of queries the user must make to reach the desired answer. Thus, associative query answering is useful for operating in a low-bandwidth environment. Furthermore, the additional information provided by the association agent

Associate query answering provides relevant information that the user does not know specifically or does not know how to request. This additional information reduces the number of queries the user must make to reach the desired answer.

allows users to interact with the information system. Because of the changing operating environment and connectivity in nomadic computing, such additional information allows the user to make subsequent queries and reach his/her query goal. The association agent uses patterns (query conditions) to link relevant subjects. Such an approach provides a match with a finer granularity than that obtained using type. We use a case-based reasoning approach to match the user query with previous cases for association [26]. To reduce the number of cases stored in the case memory, similar cases are summarized in accordance with the TAH. Thus, the user query may be matched with similar cases in the case memory. The list of associated subjects are ranked based on context, user profile, and previous relevant feedback from the same type of users. A ranking based on threshold is used to select and limit the number of associated subjects.

Explanation Agent – Explanations may occur at many points during the relaxation and/or association process. Both the relaxation and associa-

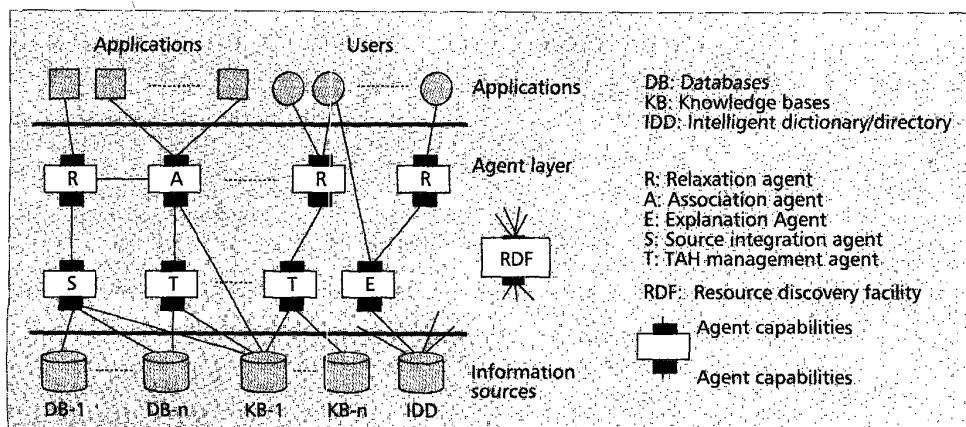
tion agents generate their action traces for input to the explanation agent. The explanation agent takes the action traces of the operations and puts the explanation goals (e.g., accuracy, cost, time) on the trace object. The invocation rules determine when explanation events occur and can be generated automatically based on the context and user type. The explanation operators determine the content of explanations. The explanation templates are associated with types, not instances, and hence have good reuse properties. Based on the user profiles, the agent generates the appropriate explanation text of the relaxation and association processes as well as the quality in terms of accuracy, and the cost of the answer to the user [20, 23].

The agents have uniform logical interfaces and can be interconnected to achieve a common task. Each agent can use the output of other agents as input. Standard agent protocol (e.g., KQML [27]) can be used to communicate with each other. New performances as well as algorithms need to be developed for performing negotiations, process binding, recovery, and commitments. Agents may be replicated and reused; our proposed cooperative agent architecture has a scaleable infrastructure, as shown in Fig. 6 [20], to support multi-adaptive agent processing for nomadic information systems.

Integrated Performance Evaluation

The design of a virtual networking layer to provide transparent support for nomadic computing raises a number of fundamental performance issues that address the interaction between the nomadic computing element, the operating system, and networking and other components of the nomadic computing infrastructure. Performance evaluation is needed for off-line consideration of alternative mechanisms to support a specific service as well as to monitor, in real time, the operation of a nomadic computing environment. An example of the type of issue that must be addressed is the following: given a replicated file system, parts of which reside on mobile machines, for which traffic patterns and network topologies is peer-to-peer reconciliation superior to master-slave reconciliation? Another example is in the selection of a subset of files to cache predictively on the mobile disk given a network connectivity with specific bandwidth and cost characteristics — for instance, given an unreliable and expensive connection (e.g., cellular), minimal caching or reconciliation protocol may be appropriate, with the quality of reconciliation and caching improving as the connectivity improves. In a dynamic environment, it may be appropriate to have an on-line monitoring facility that anticipates connectivity characteristics and adaptively selects appropriate reconciliation and caching protocols.

Another example of the types of issues that must be addressed relates to the selection of appropriate protocols at the internetworking or transport layers based on the type of traffic (data, audio, multimedia, etc.), desired quality of service (best effort, guaranteed real-time, etc.)



■ Figure 6. A scalable and extensible agent-based information system architecture.

and other parameters. Once again, the selection process can use table lookups either based on off-line simulations identifying the best alternative for a predetermined scenario or using an on-line monitoring and prediction capability that selects and dynamically adapts to its environment.

Although models are needed to evaluate the performance of the proposed environment, they will also be an important component of the operational system. Many of the services that we envisage, (e.g., predictive caching or adaptability of the network interface) may require regular predictions on the future state of a system component under an estimated workload and system configuration. The inclusion of a simplified modeling capability in the operational system will allow the system to dynamically and continuously evaluate the impact of various policies and select the most suitable alternative.

Given the complexity of the system software needed to support transparent virtual networking, a central role of the performance evaluation research must be to predict the impact of the interaction between the networking (e.g., connectivity, communication substrate, network protocols) and file management (e.g., predictive caching, replication) components on the desired quality of service for the user. The performance monitoring and evaluation environment must be able to decompose the components from their execution environment to allow maximum flexibility in experimentation with alternative implementations of a given functionality (e.g., file reconciliation strategies) as well as to support a "plug-and-play" capability that generates composite prototypes constructed from pieces which model system components at widely differing levels of detail.

In the remainder of this section we present our view of the primary challenges that must be addressed in studying the performance of a nomadic computing environment and outline an architecture for such a capability.

Simulation Environment

Analytic performance models can yield useful insights into the performance of many of the tasks envisaged as part of the nomadic computing domain. However, simulations allow the analyst to examine the interactions among different

components of a virtual networking environment, for instance, the impact of a specific transport protocol on the performance of a reconciliation algorithm in a nomadic environment. Simulations also permit an evaluation of the impact of realistic workloads, including traces derived from real-world environments on the choice of available design alternatives. As the simulation models for many of these components can become complex and detailed, large execution time for the models may become a

Given the complexity of the system software needed to support transparent virtual networking, a central role of the performance evaluation research must be to predict the impact of the interaction between the networking and file management components on the desired quality of service for the user.

serious hindrance to their inclusion. As large networks, perhaps comprising between hundreds to thousands of mobile machines, are expected to be commonplace in the future, it is desirable to design the performance evaluation environment keeping scalability issues at the forefront. Parallel execution of simulation models, particularly discrete-event simulation models, has shown a significant potential for reducing the execution time of large models. A simulation environment to study nomadic issues should certainly incorporate the potential for parallel execution of its models.

In studying the impact of changes in one subsystem of the nomadic environment (e.g., file systems) on another (e.g., transport layer protocols), it may be necessary to design detailed simulation models for the proposed subsystem and its interfaces with the rest of the system and its execution environment, including the applications. This is clearly an onerous task — detailed simulation models of complex heterogeneous systems are themselves complex and resource-intensive. An alternative is to use *hybrid models* — a partially implemented design where some components exist as simulation or analytic models and others as operational subsystems realized

in hardware or in software [28]. Use of hybrid models allows an analyst to ascertain the impact of design changes in one subsystem without developing detailed simulation models of the entire system; instead, certain system modules can be directly replaced by simulation models of alternative designs, considerably reducing the modeling overheads. The hybrid models that are developed must be parallel because the operational system from which they are derived is itself parallel.

The preceding ideas are being incorporated in the design of a general-purpose *parallel heterogeneous* simulation environment called *Collage* that is being developed for the simulation and prototyping of virtual networking components. The proposed modeling environment will support a *parallel, hybrid, real-time simulation*

Event scheduling constructs in *Maisie* are integrated with the send and receive primitives. Thus, transmission delays in a physical network can be modeled simply by incrementing the time stamp on the message when it is sent from the source to the destination node.

environment built on *Maisie*, a discrete-event simulation language developed at UCLA [29, 30]. The environment will support parallel execution of simulation models and the operational prototype of the proposed virtual networking framework in an integrated environment. Deadline-based schedulers will be incorporated to support real-time and interactive model execution.

Parallel Model Execution — Parallel execution of a simulation model requires that the model be partitioned among the available processors, and events in each partition be executed in their causal order. Causality is most commonly defined on the basis of the event timestamps. Three primary synchronization mechanisms have been used to ensure that events in a parallel simulation are eventually executed in their causal order: the conservative protocols [31], the optimistic protocols [16], and the recently proposed adaptive protocol that combines the hitherto disparate protocols [20, 32]. None of the preceding mechanisms have been found to be efficient for all types of models. A continuing challenge is to develop efficient simulation testbeds that can be used to evaluate and prototype nomadic computing environments.

One approach that is being explored for this purpose is based on the use of the *Maisie* simulation language. *Maisie* supports a diverse set of simulation protocols and parallel architectures. It is a process-based simulation language that uses message communications to schedule events and provides a natural paradigm for modeling distributed systems. Event scheduling constructs in *Maisie* are integrated with the send and receive primitives. Thus, transmission delays in a physical network can be modeled simply by incrementing the timestamp on the message when it is sent from the source to the destination node. A receive statement can optionally specify a time-out interval, where time is measured using the simulation clock; such a statement may be

used to simulate the passage of time corresponding to activities like servicing of a job in a model, or transmission of a message in a network.

Maisie has been implemented on networks of workstations, on scaleable MPP platforms like the distributed memory IBM SP2, and on symmetric multiprocessor (SMP) platforms like the SparcStation 1000 using both conservative and optimistic protocols. It has been used for the simulation of applications in diverse areas, including very large scale integrated (VLSI) designs, parallel programs, ATM networks, and high-speed communication switches and protocols. The diversity of applications, architectures, and synchronization protocols supported by *Maisie* is significant as it makes it possible to use parallel execution for modeling at many levels in the nomadic computing domain.

Hybrid Models

Hybrid models have a number of advantages over simulation models, particularly for on-line monitoring. For domains like network protocols, operational software is typically faster than a detailed simulation model; integrating operational subsystems in the model can lead to better model execution times. Second, every simulation model is an abstraction of a system and is hence only an approximate representation of the corresponding subsystem. Inclusion of operational subsystems typically implies an improvement in the overall accuracy of the model. Lastly, hybrid models are particularly useful when evaluating complex systems, parts of which have already been implemented. Rather than design simulation models of the existing subsystems, the operational system may itself be integrated into the model. In the monitoring of nomadic systems, hybrid models can be useful in predicting hot-spot activities. For instance, actual traffic activity from numerous points in the network may be collected dynamically, fed into a hybrid model to detect trends and identify potential sources of congestion, such that corrective actions may be taken to avoid the bottleneck. Hybrid models may also be useful in evaluating the ability of an operational network to handle anticipated file caching or multiple-copy replication activity that may be projected in the near future. In the event that the model detects insufficient network capacity for the anticipated workload, either the projected file transfer activity can be delayed, or alternative protocols better suited to the network constraints may be invoked.

A language, synchronization algorithms and environment have been developed at UCLA [28] to support the execution of parallel hybrid models. This environment is being used to evaluate the feasibility of hybrid models in the evaluation of the interactions among the nomadic network protocols and operating system components of a nomadic computing environment.

Real-Time Simulation

A real-time simulation facility can significantly aid the adaptive components of a virtual networking environment. Consider predictive caching as an example: the caching heuristics can identify a set of resources that are needed to be cached at a remote site over the available net-

work. Given that the entire set of identified resources cannot be transmitted, an "optimal" subset may have to be identified in a dynamic and time-critical manner. In general, it is hard to devise a static algorithm to identify such a subset. An alternative is to use a model that "executes" the caching heuristics for a set of anticipated workloads to identify the most useful subset, given the network constraints. In the extreme case, the characteristics of the network over which the files will be transmitted may also change dynamically. Running the model for the modified environment is probably the only alternative to adapting the predictive caching. Such an on-line modeling facility can be used periodically to evaluate the suitability of the "working set" identified by the predictive caching facility for anticipated changes in the workload or network environment. This will permit the interface to adapt rapidly to changes as they actually begin to occur in the operational system.

Such a real-time modeling facility must comprise both analytical and simulation models. For the analytical models, we expect to have good estimates of the execution time in the models for different workloads and system parameters; given an approximate resource requirement for the analytical components will greatly facilitate its scheduling in the on-line modeling capability.

Real-time simulation differs from traditional simulation in two primary ways:

- It executes concurrently with an operational system.
- The completion of specific (or all) subtasks in the real-time simulation is subject to physical time constraints.

In a traditional simulation program, causality is enforced by executing events in strict order of their time stamps. In contrast, in real-time simulations we execute events according to their real-time deadlines without violating causality constraints. This implies that some events may be processed in an order that is different from that implied by their time-stamps. The central problem in designing an on-line or real-time simulation facility is to develop parallel simulation algorithms that can satisfy the causality and physical time constraints of an on-line simulation.

Examples

We conclude the discussion of the performance evaluation issues relevant to the design of a nomadic computing environment with a brief outline of models for two of the primary components of such an environment. The first example, from the network protocol domain, is a *multimedia instant infrastructure protocol* to create a cluster-based time division multiplexed (TDM) infrastructure which can support instant deployment and nomadicity and enable guaranteed bandwidth for voice and video traffic. The second example, from the operating system domain, is the simulation of the reconciliation strategy proposed for a user-level replicated file system called Rumor.

Instant Infrastructure Protocol – The creation of an instant network infrastructure that is able to maintain communication, perhaps via a topological reconfiguration in the face of mobility,

node failures and environmental changes, is a central feature of a nomadic computing environment [33]. Such protocols typically use a clustering algorithm to dynamically group and assign code division multiple access (CDMA) codes to groups of users and to provide local synchronization among the nodes. They can also support multimedia (voice, video and data) traffic through special protocols which provide bandwidth allocation via virtual circuits. The bandwidth allocation is done by dynamically reserving time division multiple access (TDMA) slots throughout the network. The underlying communication mechanism uses wireless CDMA multi-hop packet routing. Hybrid models are a perfect vehicle for evaluating such protocols and monitoring their operation.

Hierarchical models were developed for the preceding protocol using different components of the Collage simulation environment. One set of detailed models were used to measure aggregate link failure rates due to loss of TDM frame synchronization as a result of CPU overload, control packet loss due to channel propagation effects or interference, and radio mobility. The aggregate link failure rates estimated from the preceding models were used in the coarse-grained models to study the impact of different link failure rates on system performance [34]. Whereas the detailed models were executed and validated for networks of small numbers of nodes, the coarse-grained models could be used

Hybrid models are particularly useful when evaluating complex systems, parts of which have already been implemented. Rather than design simulation models of the existing subsystems, the operational system may itself be integrated into the model.

to quickly predict the impact of these failure rates on larger networks of many hundreds of nodes. The abstract models are being modified such that they can be used in an on-line manner, where link failure rates are monitored in the system and fed to the model to continually evaluate their impact on overall system performance. Hybrid models can also be used which monitor the link failure rate, in real time, in the nomadic environment to predict their impact on system performance and suggest, for example, an alternative media access control protocol that is more suitable for the dynamic execution environment.

File Replication – A file replication service called Rumor is being developed at UCLA with the eventual goal of supporting consistency management of files that may be replicated across a network, some of whose nodes may be mobile. The user operates on the file system loaded on the mobile machine as if it were connected to the rest of the network. In a perfect environment, the nomadic node will, in fact, be continuously connected to the network, to allow the location of the accessed files to be transparent to the user. However, in the presence of intermittent connectivity, particularly when the file sys-

tem is being accessed by many users, the accesses may create conflicts. Such conflicts can be minimized by resorting to file reconciliation when it is feasible. An example of a conflict is when two users update the same file during an interval when both of them are disconnected from each other and the rest of the system. As the updates to the file do not happen at the same global time, if all users were connected, presumably no conflict would have resulted. However, in the presence of unpredictable connectivity, it is possible that both the updates are made to the local replicas, resulting in a conflict. The eventual goal of the replication service is to adjust the update and reconciliation frequency based on available network bandwidth and connectivity, user access patterns, and desired level of availability such that these types of conflicts can be minimized.

A simple model of the replication service, which includes the algorithms used to generate the replicas, the different types of reconciliation algorithms used, the types of network connectivity envisaged to be available for the reconciliation, the type of workload anticipated, and so on, can be used to estimate the cost of reconcili-

The creation of an instant network infrastructure that is able to maintain communication, perhaps via a topological reconfiguration in the face of mobility, node failures and environmental changes, is a central feature of a nomadic computing environment.

ation at different frequencies and to measure its impact on file availability. Parallel executions of the model could be used to explore scalability issues of the algorithms as the number of nomads increases dramatically. The model of the replicated file system can be combined with an abstract model of the instant infrastructure protocols described in the previous section to predict system performance in a truly nomadic environment where the communication infrastructure must first be established. Such scenarios are common in applications ranging from disaster relief to covert military operations. In the operational system, a parallel hybrid model of the system can be developed by linking the model of the replicated file system service to the actual network protocols that are available in a nomadic computing environment to validate the model and to evaluate its performance over network protocols that are difficult or tedious to model. Lastly, an abstract model of the service can be used in the operational nomadic environment to dynamically adapt the reconciliation service to anticipated changes in network connectivity or bandwidth characteristics.

Conclusion

In this article we have presented nomadicity as a new paradigm in the use of computer and communications technology and have laid down a number of challenging problems. We presented our vision of nomadicity, and exam-

ined some of the salient issues and architecture considerations.

Nomadcity is an emerging fact of life for a significant fraction of users. The needs are real. The field is current, exciting, draws from many disciplines, and offers a variety of kinds of problems whose solutions are of immediate importance. Nomadicity makes all the problems in computing and communications harder, but the potential payoffs for improved functionality are huge. In a word, one cannot ignore the challenge of nomadicity.

References

- [1] Computer Science and Telecommunications Board, Realizing the Information Future: The Internet and Beyond, [National Academy Press, Washington, DC, 1994].
- [2] M. Weiser, "The Computer for the 21st Century," *Scientific American*, Sept. 1991, pp. 94-104.
- [3] Nomadic Working Team of the Cross Industrial Working Team, "Nomadicity: Characteristics, Issues, and Applications," 1995.
- [4] L. Kleinrock, "Nomadic Computing - An Opportunity," *ACM SIGCOMM Computer Communication Review*, vol. 25, no. 1, Jan. 1995, pp. 36-40.
- [5] A. Alwan et al., "Adaptive Mobile Multimedia Networks," submitted to *IEEE Personal Commun. Mag.*, Oct. 1995.
- [6] R. Jain, et al., "PC-notebook Based Mobile Networking: Algorithms, Architectures and Implementations" to appear in *Proc. ICC '95*, June 1995.
- [7] R. H. Katz, "Adaptation and Mobility in Wireless Information Systems," *IEEE Personal Commun. Mag.*, vol. 1, no. 1, (First Quarter, 1994), pp. 6-17.
- [8] J. Short, R. Bagrodia, and L. Kleinrock, "Mobile Wireless Network System Simulation," *Proc. of the 1995 ACM International Conference on Mobile Computing and Networking*, Berkeley, CA, Nov. 1995, (to appear).
- [9] G. H. Kuenning, "Design of the SEER predictive caching system," *Proc. of the Workshop on Mobile Computing Systems and Applications*, Santa Cruz, CA, Dec. 1994.
- [10] J. S. Heidemann et al., "Primarily disconnected operation: Experiences with Ficus," *Proc. of the Second Workshop on Management of Replicated Data*, IEEE, Nov. 1992, pp. 2-5.
- [11] J. Kistler, Disconnected Operation in a Distributed File System, Ph.D. dissertation, Carnegie-Mellon University, May 1993.
- [12] P. Kumar and M. Satyanarayanan, "Supporting application-specific resolution in an optimistically replicated file system," *Proc. of the Fourth Workshop on Workstation Operating Systems*, Napa, CA, Oct. 1993 IEEE, pp. 66-70.
- [13] P. Reiher et al., "Resolving file conflicts in the Ficus file system," *USENIX Conference Proceedings*, USENIX, June 1994, pp. 183-95.
- [14] J. Misra, "Distributed Discrete-Event Simulation," *ACM Computing Surveys*, vol. 18, no. 1, pp. 39-65, March 1986.
- [15] M. Satyanarayanan et al., "Experience with disconnected operation in a mobile computing environment," *Proc. of the USENIX Symposium on Mobile and Location-Independent Computing*, Cambridge, MA, August 1993, USENIX, pp. 11-28.
- [16] K. Rekhi, private communication.
- [17] D. Rosenthal, private communication.
- [18] W. W. Chu, "Performance of Directory Systems for Databases in Star and Distributed Networks," *AFIPS Conf. Proc.*, 1976.
- [19] W. W. Chu and K. Chiang, "Abstraction of High Level Concepts from Numerical Values in Databases," *Proc. of the AAAI Workshop on Knowledge Discovery in Databases*, Seattle, Washington, July 1994.
- [20] W. W. Chu et al., "CoBase: A Scaleable and Extensible Cooperative Information System," to appear in *J. Intelligent Info. Sys.*, [Kluwer Academic Publishers, 1996].
- [21] W. W. Chu and Q. Chen, "A Structured Approach for Cooperative Query Answering," *IEEE Trans. Knowledge and Data Eng.*, vol. 6, no. 5, Oct. 1995.
- [22] W. W. Chu, M. Merzbacher, and L. Berkovich, "The Design and Implementation of CoBase," *Proc. SIGMOD '93*, May 1993, pp. 517-22.
- [23] M. Minock and W. W. Chu, "Interactive Explanation for Cooperative Information Systems," *UCLA Computer*

Science Technical Report, 1995.

- [24] W. W. Chu, Q. Chen, and R. Lee, "Cooperative Query Answering via Type Abstraction Hierarchy," Cooperative Knowledge Based System, S. M. Dean, ed., [Elsevier Science Publication Inc., 1991].
- [25] M. Merzvaeh and W. W. Chu, "Pattern Based Clustering of Database Attribute Values," *Proc. All Work-shop on Knowledge Discovery in Databases*, Washington, DC, 1993.
- [26] G. Fouque, W. W. Chu, and H. Yau, "A Case-Based Reasoning Approach for Associative Query Answering," *Proc. of the 8th International Symposium on Methodologies for Intelligent Systems*, Charlotte, North Carolina, Oct. 1994.
- [27] T. W. Finin *et al.*, "Specification of the KQML agent-communication language, plus example agent policies and architectures," Prepared by the ARPA Knowledge Sharing Initiative External Interfaces Working Group, June 1993.
- [28] R. Bagrodia and C. C. Shen, "MIDAS: Integrated Design and Performance Evaluation of Distributed Systems," *IEEE Transactions on Software Engineering*, Oct. 1991.
- [29] R. Bagrodia, "Designing Efficient Simulations Using Maisie," *Proc. 1991 Winter Simulation Conference*, Phoenix, AZ, Dec. 1991, pp. 243-247.
- [30] R. Bagrodia and W. Liao, "Maisie: A Language for design of Efficient Discrete-Event Simulations," *IEEE Transactions on Software Engineering*, April 1994.
- [31] D. J. Jefferson, "Virtual Time," *ACM TOPLAS*, 1970, pp. 404-425.
- [32] V. Jha and R. Bagrodia, "A Unified Framework for Conservative and Optimistic Distributed Simulation," *Proc. 1994 Workshop on Parallel and Distributed Simulation*, Edinburgh, July 1994.
- [33] M. Gerla and J. Tsai, "Multicluster Mobile Multimedia Network," *ACM, Baltzer Journal of Wireless Networks*, Aug. 1995.
- [34] R. Bagrodia *et al.*, "A Hierarchical Simulation Environment for Mobile Wireless Networks," to appear at Winter Simulation Conference, Dec. 1995.

Bibliography

- G. H. Kuenning *et al.*, "An analysis of trace data for predictive file caching in mobile computing," *USENIX Conf. Proc.*, USENIX, June 1994, pp. 291-306.
- C. D. Tait *et al.*, "Intelligent file hording for mobile computers," *Proc. MobiCom '95: 1st int'l. Conf. Mobile Computing and Networking*, Berkeley, CA, Nov. 1995.

Nomadcity is an emerging fact of life for a significant fraction of users. The needs are real. It makes all the problems in computing and communications harder, but the potential payoffs for improved functionality are huge.

Biographies

RAJIVE L. BAGRODIA received the B.Tech. degree in electrical engineering from the Indian Institute of Technology, Bombay, in 1981, and the M.A. and Ph.D. degrees in computer science from the University of Texas at Austin in 1983 and 1987, respectively. He is currently an associate professor in the Computer Science Department at UCLA. His research interests include parallel languages, parallel simulation, distributed algorithms, and software design methodologies. He was selected as a 1991 Presidential Young Investigator by the National Science Foundation (NSF).

WESLEY W. CHU [F] received the B.S. and M.S. degrees from the University of Michigan, Ann Arbor, in 1960 and 1961, respectively, and the Ph.D. degree in electrical engineering from Stanford University in 1966. He worked at General Electric, IBM, and AT&T Bell Laboratories prior to joining UCLA in 1969. He served as a chair of the UCLA Computer Science Department from 1989 to 1991, and is currently a professor in the department. His current research interests are in the areas of distributed processing, distributed databases, and knowledge-based systems.

LEONARD KLEINROCK [F] received the B.S. degree in electrical engineering from City College of New York in 1957 and the M.S.E.E. and Ph.D.E.E. degrees from the Massachusetts Institute of Technology in 1959 and 1963, respectively. He is currently a professor in the Computer Science Department at UCLA. His research interests focus on performance evaluation and design of many kinds of networks (e.g., packet-switching, packet radio, LANs, MANs, broadband, and gigabit networks) and of parallel and distributed systems.

GERALD POPEK has been a professor of computer science at UCLA since 1973. He has been the principal investigator for the ARPA distributed systems contract since 1977. He is best known for his work on secure systems, the design of the Locus distributed UNIX system, and most recently the Ficus large-scale replicated filing environment and mobile computing architectures. His academic background includes a doctorate in computer science from Harvard University.