

Adapting to a highly variable and unpredictable environment

Adaptive Mobile Multimedia Networks

**A. ALWAN, R. BAGRODIA, N. BAMBOS, M. GERLA, L. KLEINROCK,
J. SHORT, AND J. VILLASENOR**

Consider a networking environment in which the users are mobile, the topology changes, code division multiple access (CDMA) provides multiple wireless channels, the bandwidth of a given link is unpredictable and possibly very low, the error rates are extremely high and variable, major interference occurs when multiple transmissions take place over (possibly different) links on the same or different codes, real-time multimedia traffic must be supported as well as datagram traffic, there is no stable communication infrastructure, and there is no central control! This is the environment that is addressed by the research described in this article.

We consider the problem of developing a design prototyping methodology, performance evaluation techniques, and networking algorithms to support a rapidly deployable radio network for such an environment. The network must be capable of providing guaranteed quality of service (QoS) to real-time multimedia traffic in a mobile, wireless, multihop radio network with no fixed infrastructure (e.g., no base stations). This last comment is worth emphasizing because much of the research in wireless communications has exploited the existence of central control emanating from a base station; we deal with no such central system support in this research.

Another element of the environment with which we deal is that of multihop communications. This means that, due to transmitted power constraints, not all radios are within range of each other, and may need to be relayed from one radio to another on the way to their intended destinations. One reason for considering multihop communication is that when we are forced to deploy a radio infrastructure rapidly, we may not have the luxury of placing all radios within range of each other. A second reason is that by carefully limiting the power of radios, we conserve battery power. Furthermore, we also cause less interference to other transmissions further away; this gives the additional benefit of "spatial reuse" of channel spectrum, thus increasing the capacity of the system. Of course, multihop systems are more complex than single-hop centrally controlled systems, and that is part of the challenge we faced in this system design.

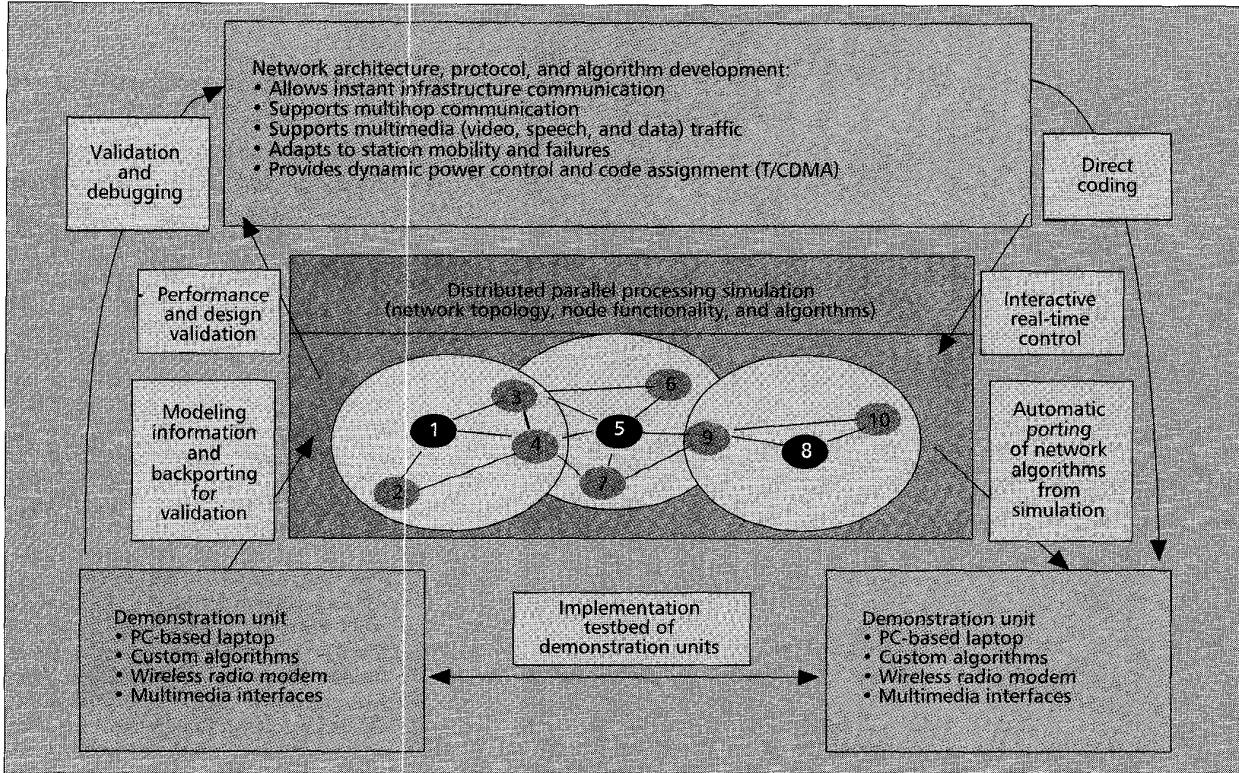
The kinds of application scenarios that motivate this

research include many that require instant infrastructure network support and multimedia network support. These include military applications (special operations, battlefield scenarios, etc.), disaster relief (fire, earthquake, flood), law enforcement situations, short-term scenarios such as public events, and so forth.

The salient features of this environment may be described as the "3M" environment: real-time Multimedia, Multihop, and Mobile. In the past, there have been studies and implementation of systems that combined any two of these Ms, but not all three. For example, real-time Multimedia plus Multihop has been studied in certain satellite systems (e.g., Iridium); real-time Multimedia plus Mobile is pervasive in cellular radio systems and Berkeley's dynamic hand-off with "hints" for multimedia, Mobile, cellular (i.e., single-hop) networks [1]; and Multihop plus mobile was well studied in the 1970s Advanced Research Projects Agency (ARPA) packet radio project.¹ It is the three-way combination that provided the challenges addressed in this article. Our approach to the 3M environment has been to address its key systems issues and to provide a methodology for its system performance evaluation (based largely on simulation), development, and implementation.

Given the diverse set of factors that can influence the performance of network protocols in this domain, we decided to adopt an integrated design, evaluation, and prototyping methodology from the outset. The methodology is illustrated in Fig. 1: new protocols were typically simulated prior to implementation. A modular simulation testbed for mobile sys-

¹ A challenge similar to ours was faced in the 1970s when ARPA funded the Packet Radio Network project [2, 3]. That project developed mobile, multihop wireless networks. Indeed, the results there showed that such a network could be deployed, albeit with extremely low channel efficiency and with bulky radio data terminals. Today we enjoy the fruits of two decades of technology improvement in many dimensions (bandwidth, processing power, size, weight, power consumption, battery power, etc.); some of these technologies have followed Moore's Law and have provided improvements in excess of a factor of 10,000. In view of these technology gains since the 1970s project, ARPA felt that it was time to reexamine these problems and extend them to real-time multimedia; hence the creation of ARPA's Wireless Adaptive Mobile Information System (WAMIS) initiative. As part of this initiative, our group is examining various systems issues regarding adaptive rapidly deployable wireless mobile multimedia networks.



■ Figure 1. Design methodology.

tems simulation was developed for this purpose. In order to preserve the significant investments in model development, the simulation capability was designed to permit the network algorithms to be directly ported from the simulation to the physical domain, where the algorithms could be executed on the physical testbed.

This article addresses the algorithmic digital signal processing issues for video and speech as well as the networking issues that arise in such an environment of rapid deployment. The key issues in multimedia digital signal processing revolve around delivering a suitable QoS in the face of changing and limited bandwidth as provided by the underlying network. The network focus is on the system design and algorithms that have been developed to cope with changing topology, variable error rate, cross-channel interference, and the requirements of the multimedia traffic. The kinds of algorithms we describe include those of power control, routing control, admission control, code assignment, dynamic topology configuration, spatial reuse, video and speech compression, and more. In addition, we discuss a set of simulation tools that have been developed at the University of California-Los Angeles (UCLA) to evaluate the performance of these many algorithms.

Architecture Overview

In networks that are set up dynamically and adaptively, one cannot rely on the convenience of a backbone or predetermined topology. Supporting the rapidly deployable and adaptive network introduces high overhead and low bandwidth utilization for the multimedia application traffic. Adaptive multimedia (speech and video) compression algorithms make efficient use of the available application bandwidth.

In Fig. 2 we show the five basic components of our architecture. Each of these components are described in more detail in the following subsections. These components are:

- Multimedia applications
- Multimedia compression algorithms
- Standard network algorithms
- Wireless adaptive network algorithms
- Wireless communications substrate

One notes the flow of QoS measurements from lower (network) levels up to the multimedia compression algorithms that use these measurements to adjust the parameters of the compression algorithms. Moreover, we see the exchange of the lower-level radio control and measurements between the wireless communication substrate and wireless adaptive network algorithms layers.

Multimedia Applications

In the spirit of supporting multimedia traffic at the application level with a decentralized control structure, we recognize that the *details of the infrastructure support* should be hidden from the user and the applications. However, it is desirable to have the *multimedia support* appear at the application level. In the past, network support for applications has been text-based, for example, remote login (telnet) and file transfer protocol (FTP). There has recently been an emergence of static multimedia capabilities for stored images (Joint Photographic Expert Group images — JPEGs — and MOVs) and pre-coded speech/audio (WAVs and SNDs). The network traffic created by these static multimedia applications has exploded with the emergence of the World Wide Web (WWW) and the corresponding WWW browsers such as Netscape and Mosaic which allow for real-time multimedia as well. In order to support the real-time aspect of these multimedia application requirements, a new type of network support is being developed which can enable control such as QoS. Real-time multimedia streams must be captured, coded, transferred over the network, decoded, and presented to the user. This should take place with minimal delay while making efficient use of the available multimedia application bandwidth. Not only do

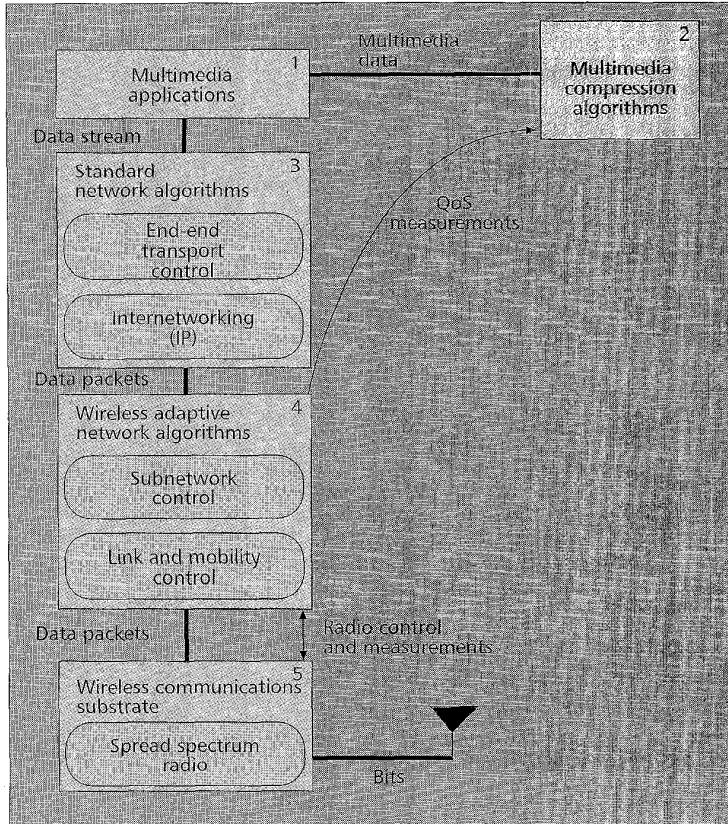


Figure 2. Adaptive multimedia system components.

applications need to be able to synchronize the data, video, and voice streams, but the network (e.g., transport layers) needs to directly support these real-time multimedia requirements while remaining compatible with the transport and internetworking infrastructures that exist today. In order to accomplish this, the multimedia applications negotiate resource allocation with the network and adapt to the changing network condition by using adaptive compression and secure encoding techniques. This support is especially critical in a wireless mobile environment. This is because the wireless mobile segment of an internet path is subject to continuous changes in bandwidth and channel quality. The QoS information fed back to the multimedia layer (Fig. 2) permits dynamic renegotiation of parameters so as to adjust to these changes.

Multimedia Compression Algorithms

Supporting multimedia traffic in these networks requires efficient use of the available bandwidth. As was seen in the packet radio projects [2, 3], the overhead in maintaining connectivity and routing information can be high. The source coders described below are able to efficiently use and flexibly adapt to the available bandwidth. As the topology and other network attributes change, the bandwidth and reliability of individual links will change. To respond to these changes, both the voice and video source coding algorithms must support rate-adaptive compression. In addition, the source coding must be performed using algorithms that represent the compressed source data in a manner that provides or facilitates error protection. For example, multiresolution decompositions are used in which different portions of the encoded bitstream can be assigned appropriate priority levels. This information is then used at the higher network layers in decisions regarding

the ways in which packets containing voice and video information are handled.

Standard Networking Algorithms

Transport, and especially internetworking, must support large-scale connectivity across multiple communication substrates (fiber, coax, satellite, and radio). The Internet Protocol (IP) of the Transmission Control Protocol/IP (TCP/IP) protocol suite is a time-proven protocol, which can support this internetworking for static (nonmobile) hosts. In order to accommodate mobility while remaining compatible with the internet, the wireless network architecture we use supports TCP/IP. TCP/IP views any infrastructure as a virtual network (thus providing a level of abstraction from the wireless link). To support large-scale mobility (movement of nodes from one area of the internet to another), protocols like Mobile IP [4] can be used. While most of the Mobile IP proposals address data traffic only, some recent contributions, such as Berkeley's "handoff with hints" scheme [1], consider the multimedia traffic case as well, and propose solutions for the maintenance of virtual circuits in the face of mobility. To support applications, such as "search and rescue," which must function without connectivity to the internet and rely on an instant infrastructure, we must depend on the wireless subnet to control local connectivity. This wireless subnet is viewed as another network on the internet where independent and specific algorithms such as multihop routing and topology creation can be run independent of constraints imposed by other infrastructures, networks, and communication substrates.

Wireless Adaptive Networking Algorithms

The adaptive wireless subnetwork control algorithms implement specialized functions to support the rapidly deployable, wireless multimedia requirements. The necessary functionality includes reconfiguration and dynamic control over the topology to instantly set up an infrastructure when new nodes start up, and to adapt to the network configuration when nodes move or fail. This wireless subnet is viewed as a virtual topology which is set up among the nodes by selecting or creating a set of links to local neighbors. Various algorithms, some of which are described below, can determine when and how to create new links using various techniques such as clustering [5], power minimization [6], or highest reliability [7] where the best code and power levels are chosen to create reliable links among neighbors in support of the network's connectivity and multimedia requirements (e.g., virtual circuits) in the face of mobility. These algorithms are described in greater detail in the "Wireless Subnetwork" section.

Wireless Communication Substrate

There are numerous wireless spread spectrum radio modems now commercially available [8], each of which has its own set of characteristics, capabilities, and built-in link-level control algorithms. Our goal is to be able to support various types of wireless communication substrates in our rapidly deployable multimedia network, and utilize the spread spectrum CDMA technology capabilities for maximizing bandwidth, increasing spatial reuse, and resisting interference and noise. In order to accomplish this, the adaptive wireless network control algorithms react to measurements from the radio and

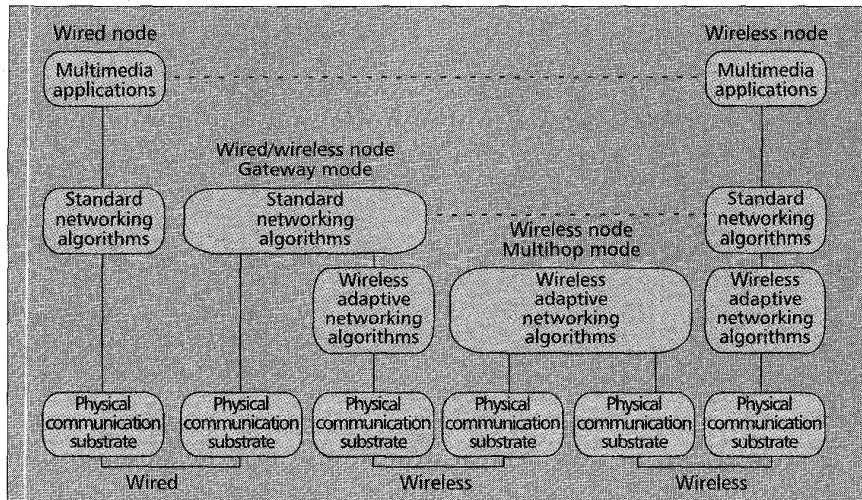
dynamically control various parameters.

Many of the commercial spread spectrum radio modems we have examined do not support the adaptive control parameters and provide feedback as desired by the adaptive networking algorithms. We have used a direct sequence spread spectrum radio modem developed at UCLA [9] which is capable of supporting various control parameters and providing performance and measurement feedback. In the fourth section, we look at the interaction between the adaptive networking algorithms and the communication substrates and consider several application programming interfaces (APIs) that are needed.

Returning now to Fig. 2, one notes that if we remove boxes 2, 4, and 5, we have a system architecture, as illustrated in the reference model in Fig. 3, that can support multimedia in a stationary wired infrastructure (as in much of today's Internet, for example). If we now add box 5, we extend the multimedia support to that of today's wireless (e.g., cellular or satellite) stationary environment, which again presents a stable infrastructure as regards bandwidth. However, once we add mobility and multihop, we are required to add boxes 2 and 4 to deal with the rapidly changing infrastructure and topology. Since the elements of these last two boxes (2 and 4) represent new components of the architecture, we devoted our efforts to the domains they represent. In the following section, we begin with the crucial elements of box 2 (rate-adaptive compression for multimedia) and then move on to the fourth section, where we consider the elements of box 4 (the wireless adaptive network algorithms). Next, we describe the simulation environment that "embraces" the system design in terms of providing the essential performance evaluation needed for the methodology. Finally, we describe in the last section those portions of the system that we have implemented as well as the tools we have developed to "automate" the implementation process.

Video and Speech Encoding

A key requirement for mobile wireless multimedia systems is the ability for the source coders to make efficient and flexible use of the available bandwidth. As the topology and other network attributes evolve, the bandwidth and reliability of individual links will change. To respond to these changes, both voice and video source coding algorithms have been designed to support rate-adaptive compression. Namely, the source can adjust the rate based on the QoS advertised by the network. In addition, the source coding is performed in a manner that permits "on-line" rate adjustment in the network based on changing path conditions. For example, multiresolution decompositions are used in which different portions of the encoded bitstream are assigned different priority levels. This information is used at lower-level network layers in decisions regarding the ways in which packets containing voice and video information are handled (e.g., if bandwidth becomes scarce, low-priority packets are dropped). The following section describes the design of our video and voice coders/decoders (codecs) and focuses on the features specifically required to handle a rapidly changing wireless environment, namely error protection and adaptive rate adjustment.



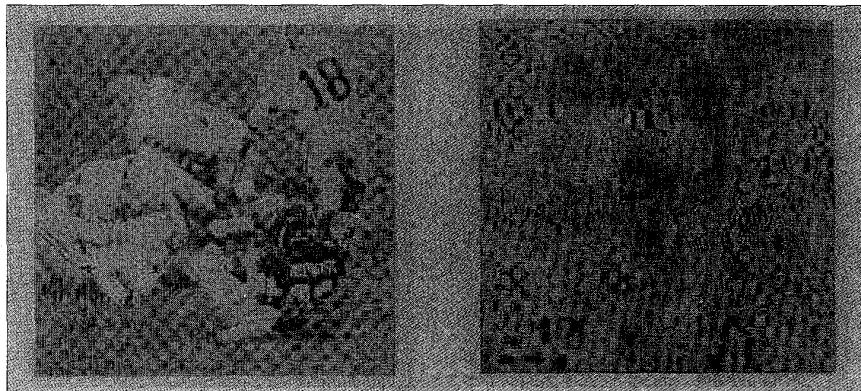
■ Figure 3. Network control reference model.

Video Coding Algorithms: Design Issues in Wireless Environments

In a wireless environment, the channel quality and the codec cost/power/complexity design constraints differ substantially from those of a wireline system. Thus, video coding techniques will differ from those that have been developed and refined for wireline applications. For example, in order to reduce complexity and increase robustness, algorithms for wireless video place a lower emphasis on motion compensation and a higher emphasis on intraframe coding. These algorithms also employ low-complexity transforms and quantization schemes, and utilize coding and synchronization techniques, allowing not only protection from random or bursty bit errors, but recovery from deep fades in which the signal is lost for several seconds or more. The algorithms adapt to changing bandwidth allocations from the network and to changing error characteristics of the channel. Finally, error protection, packetization, and other transfer protocol aspects affect both the video coding and the network, and are designed to enable optimal performance in both.

Under the general umbrella of wireless video systems there are a range of distinct applications that place different requirements on the video coding algorithms and hardware. One application consists of remote monitoring in which the wireless unit transmits but does not receive video. By contrast, in a portable transceiver to be used for transmitting and receiving video it is necessary to use a higher frame rate, and to design an algorithm in which both the encoding and the decoding can be performed with low complexity. Khansari *et al.* [10] have proposed a modified version of the International Telecommunications Union standard H.261 video coding algorithm in which a dual rate bitstream is used to achieve robustness in a microcellular CDMA environment. Like H.261, this approach utilizes motion compensation and involves the same image quality/complexity trade-off mentioned above. A third application consists of a wireless terminal that is optimized for receiving video (as opposed to supporting fully symmetric two-way video). In the Infopad project at UC Berkeley, a team led by Brodersen [11] has taken advantage of this asymmetry to place much of the computational complexity burden at the encoder, leading to a vector quantization scheme in which video decoding can be performed with extremely low complexity using a table lookup.

The above examples illustrate that no single coding algorithm is likely to be optimal across all wireless video applications. Each of the cited approaches reflects a different set of



■ Figure 4. Illustration of the importance of error protection in wireless multimedia communications. These images were compressed to .46 b/pixel (about 17:1) using the wavelet scheme described earlier. This corresponds to a frame rate of 15 frames/s if the video bandwidth is 450 kb/s. The channel bit error rate used in this example is .003. In the image on the left, some of the bandwidth was allocated to error protection using punctured convolutional codes that exploit the hierarchical nature of the wavelet decomposition. The image on the right is unprotected; that is, the full bandwidth was used for source coding. The optimal allocation of bits to source or channel coding is an important aspect of a wireless multimedia system, and requires close interaction between the source coding and network layers.

requirements on bandwidth, complexity, robustness and functionality. In our project we are addressing interactive multimedia applications (e.g., search and rescue) and therefore are concerned with algorithms that enable both high functionality and low complexity in a network of wireless transceivers. We have performed a design in which the complexity constraint is dominant, but in which two-way, rate-adaptive coding is performed at low bit rates ranging from 60 to 600 kb/s.

One of the most basic design decisions in a wireless image transmission system concerns the use of motion compensation. From an image quality standpoint, the advantages of motion compensation are large. While the precise coding gain due to motion compensation depends on the video content and the frame rate, one can expect between a factor of three and ten improvement in compression efficiency when the redundancy between successive frames in a video sequence is utilized. Unfortunately, this image quality gain comes at the price of a large increase in complexity. For example, if spatial-domain block-matching using a full search is used, the number of additions the encoder must perform will increase by approximately two orders of magnitude. Reducing the number of block-to-block comparisons by adopting a fast search approach is only of limited value because memory accesses will still be several times greater than what would be needed in a system with no motion compensation.

For low-power wireless systems in which both the encoding and decoding must be performed on a portable transceiver in which power consumption must be minimized, this complexity increase is problematic. Motion-compensated video has the additional disadvantage in wireless systems of being more vulnerable to transmission errors because the effects of errors can propagate across several frames. Given the choice between maximizing compression efficiency at the price of high complexity or minimizing complexity at the cost of sacrificing some bandwidth efficiency, we have emphasized low complexity. As a result, the algorithm we have designed and implemented is based on pure intraframe coding.

Wavelet-Transform-Based Rate-Adaptive Video Compression

The transform constitutes one of the most important elements of a compression algorithm. While transforms themselves do

not perform any compression, they furnish an alternative representation of image data that can be efficiently compressed in the subsequent quantization step. The block discrete cosine transform (DCT) is the transform employed in the JPEG [12] and Motion Picture Experts Group (MPEG) [13] compression standards, as well as in video conferencing standards such as H.261. The DCT performs well when low compression ratios are used, but at high compression ratios only the lowest-frequency DCT coefficients from each 8-pel by 8-pel block are retained, resulting in severe blocking artifacts in the reconstructed image. Wavelet transforms offer an alternative to the DCT and can lead to very efficient still image compression if the right filters are used. The principle behind the wavelet transform, as

elaborated in a number of recent papers [14, 15], is to hierarchically decompose an input signal into a series of successively lower-resolution reference signals and their associated detail signals. This hierarchical representation is very useful to our lower networking layers, which use the range of available priorities in decisions on packet delivery.

Our coding scheme consists of a six-level discrete wavelet transform (DWT) followed by scalar quantization, run-length coding (RLC), and Huffman coding, with error protection provided by convolutional codes. Each of these steps has been modified to increase its functionality in the wireless system. Since the DWT represents the most computationally intensive step in an intraframe subband coder, the length and nature (floating point versus integer) of the filter coefficients will have a significant effect on the overall power requirements. Using new filter evaluation methods based on the impulse and step responses of the DWT, we performed a study of filter banks and identified short (a total of eight low-pass and high-pass taps) integer-coefficients filters [16] that require approximately two-thirds less chip area than the more commonly used 16-tap 9/7-wavelet filter [7]. Measured in terms of peak signal-to-noise ratio (PSNR), the compression performance of the filter we are using is within 0.5 dB of that of the 16-tap filter at the data rates of interest.

We are using scalar quantization, performed adaptively on a subband-specific basis based on a "Q-factor" similar to that used in the JPEG standard. The quantizer is simplified for low-power implementation by choosing the quantization step sizes Q_i for subband i as a function of the subband average absolute values A_i . After normalization by the user "Q-factor," the Q_i are rounded to a power of two so that a shift rather than a multiply performs the actual quantization. The resulting quantizer implementation is multiplierless, and requires only one add and seven shifts per pixel. The average reduction in image quality due to the simplifying approximations is about 0.3 dB PSNR. Coding rate adaptivity in response to changing network allocation is achieved by varying the quantizer Q-factor, which normalizes all subband quantizer step sizes. We use a table of 32 Q-factors capable of achieving bit rates of between 0.1 and 0.8 b/pixel (bpp) over a wide range of 8-bit greyscale images; rate control to within 0.02 bpp is achieved by feedback from previously transmitted frames. To protect against decoder sync errors when the variable-

Given the choice between maximizing compression efficiency at the price of high complexity or minimizing complexity at the cost of sacrificing some bandwidth efficiency, we have emphasized low complexity. As a result, the algorithm we have designed and implemented is based on pure intraframe coding.

length code words are corrupted by channel noise, we use a fixed length end-of-frame marker to maintain frame sync, and an end-of-subband (EOS) symbol to prevent sync errors from propagating across subbands. Detection of the correct EOS is aided by the self-synchronizing properties of Huffman codes [18], which lead to rapid self-resynchronization if the bit error rate (BER) in the received bitstream is below about .001. To ensure a low BER in the received bitstream, we use punctured convolutional codes and unequal error protection of the subbands.

This image algorithm, including the integer wavelet filters, the simplified quantizer, and the entropy, achieves 31.4 dB PSNR on the 512 × 512 greyscale "lena" image at 0.201 bpp, and delivers intelligible video at channel BERs up to 0.02. An example of a coded image from this algorithm is shown in Fig. 4. As mentioned earlier, this system has been designed to minimize complexity while still permitting two-way transmission over error-prone channels. The result is a bitstream that requires more bandwidth than an interframe approach such as H.261, but does not utilize expensive (in terms of power) motion compensation and retains intelligibility in high BER environments. The compression algorithms (wavelet transform, quantization, run-length coder) described here have been implemented on a Xilinx FPGA, and integrated with the networking protocols and radio to support real-time video at 8 frames/s at a frame resolution of 256 × 256.

Speech Coding

As mentioned earlier, the wireless multimedia network supports speech as well as video coding. As with video coding, the design of high-quality speech coders for wireless and mobile networks is a challenging task; good quality must be maintained with low power consumption under time-varying channel conditions and limited bandwidth. The design must account for a number of parameters such as bit rate, delay, power consumption, complexity, and quality of coded speech. For the wireless and mobile network described in this article, noise robustness, hierarchical, or embedded, encoding and low delay are critical. Noise robustness is important because wireless channels are typically characterized by fading and interference. Embedded encoding allows for rate adaptation, which is required for mobile networks with time-varying channel conditions; as the bit rate is reduced, the degradation in speech quality should be graceful. Low delay is important for two-way communication (without echo suppression) because delays of 30 ms or higher are noticeable to human listeners.

In the past, speech codec design has mostly been driven by bandwidth efficiency considerations, the target application being telephonic where the channel does not vary considerably with time and the SNR is relatively high. For example, code-excited linear prediction (CELP)-based coders are popular because of their low bit rates. The performance of these coders, however, deteriorates significantly in the presence of background noise, and their complexity is rather high.

We developed a speech and audio compression scheme with the following characteristics:

- Low delay
- Adaptive bit allocation and quantization
- Noise-robust

The coder consists of four components: analysis/synthesis filter banks, a perceptual metric, an adaptive bit allocation scheme, and a quantizer. The subband coder processes input frames of 20 ms at 8 kHz. A detailed description of the coder can be found in [19, 20].

Low Delay – The analysis/synthesis filter banks we designed achieve the lowest delay reported in the literature. The low delay results from using computationally efficient filter banks which are infinite impulse response (IIR) quadrature mirror filters (QMFs). We designed and implemented an eight-channel tree-structured IIR QMF bank with seventh-order elliptic filters. These filters provide over 60 dB attenuation in the stop bands; a minimum of 40 dB stop-band attenuation is typically used in speech coding applications [21]. It should be noted that filter order has been optimized such that phase distortions are not audible.

Adaptive Bit Allocation and Quantization – Once the speech signal is processed through the analysis filter bank, it is split into different frequency bands. A number of bits are then used to quantize the samples in each subband. Instead of assigning a fixed number of bits per band, we adaptively change the number based on the signal characteristics. The adaptive bit allocation is achieved by exploiting the masking property of the human auditory system. Masking is a phenomenon in which one signal becomes inaudible, or masked, by the presence of another (masker) signal. The minimum number of bits per band is chosen such that the speech signal masks the quantization noise in that band. If the number of bits available is insufficient to mask out the quantization noise, we allocate bits in proportion to the prescribed SNR required for masking. Our scheme provides better perceptual quality when compared to the more commonly used reverse water-filling scheme, such as that employed in the audio coding scheme of MPEG III [22].

Finally, we designed a quantizer which takes into account the statistical distribution of the input signal. Analysis was performed on a number of subband speech signals, and the signals were then normalized using the subband signal variances. The average subband speech sample distribution resembled a Gaussian; hence, our scalar quantizer uses a Gaussian table. The quantizer was made embedded by considering the effect of truncating the least significant bits (LSBs) of the index. The quantizer is optimum at full rates, and yields optimum reconstruction values if bits are lost or packets dropped in transmission. The embedded bitstream allows the coder output to be scalable from high-quality at higher bit rates to lower-quality at lower rates, supporting a wide range of services. The network takes advantage of this by dropping the LSBs first when congestion or reduction of bandwidth affects the virtual circuit.

Performance Evaluation – Traditionally, the performance of encoding systems has been evaluated using the SNR criterion. For encoded speech and audio signals, however, SNR is a poor indicator of distortion in the coded signals. Signals with high SNR may contain significant levels of audible distortion, whereas signals with moderate to low SNR may contain noise levels that are not perceptible [23, 24]. Hence, subjective tests were used to evaluate our coder. In these tests, signals were presented monaurally to simulate audition with a telephone handset or a portable radio unit. The sentences were presented at levels ranging between 80 and 90 dB sound pressure

Layer	Alg	Clustering	SWAN	Virtual Net	Token CDMA
Network	Call management with QoS acceptance, fast VC, datagram				
	QoS routing (loop-free B-F)	Probing, power adj., minimum power routing	Source routing	QoS routing (loop-free B-F)	
Link control	ACKs back pressure, priorities, voice/video rate adjustment				
	Priority drop	Adaptive power adj.	Power control	Low-priority drop	
MAC	TDMA, CDMA PRMA (voice/video) Slotted CSMA (data)	TDMA, CDMA with channel probing	TDMA, CDMA	Token, CSMA, CDMA	
Connectivity management	Lowest-ID clusterhead election	Link probing, adaptive power control	Virtual topology maintenance	Highest-degree clusterhead election	

■ Figure 5. Alternative network architectures.

level (SPL) with additive road noise measured at 85 dB SPL. Results show that our coder, at 12 kb/s and higher, provides good speech quality. Informal subjective tests were conducted to compare the performance of our coder to that of the QCELP coder — the speech service option for wideband spread spectrum digital cellular system, Electronics Industry Association/Telecommunications Industry Association/International Standard 96 (EIA/TIA/IS-96) — at 8 kb/s and a Global Mobile System (GSM) 6.10 coder (a standardized lossy speech compression algorithm employed by most European wireless telephones) at 13 kb/s. In error-free conditions, our coder at 12 kb/s had similar performance to both GSM and QCELP. In the presence of road noise, however, the average quality rating score (referred to as Mean Opinion Score — MOS) dropped by 70 percent for QCELP and by 10 percent for GSM, while the MOS score for our coder was not affected.

The coder has been implemented in software. In addition, a simplified version of the coder has been implemented on a TMS320C50 DSP board, controlled by a PC, which allows for real-time processing. Protocol routines have been developed on both the DSP and the PC for both speech coding and decoding.

While CELP-based coders may be adequate for telephonic applications, future applications such as multimedia personal communication systems demand low-complexity high-quality speech coders under varying channel conditions, which can be supported by the coder described here.

The Wireless Subnetwork: Bandwidth Allocation and Topology Control

The previous section has discussed efficient video/voice coding strategies to handle multimedia traffic in a wireless mobile network. Several assumptions were made about the services that the subnet layer can provide, and the QoS information which must be passed from the subnet layer to upper layers. In this section, we define the functions provided by the subnet layer, and describe their implementation in specific network algorithms. The main objective is to support efficient, reliable transfer of multimedia traffic in a mobile, multi-hop, self-configuring environment. As mentioned earlier, the systematic design of networks, including all these aspects and providing QoS support for multimedia, is still an unexplored problem.

The goal of the wireless subnet design team has been to explore various approaches to the solution of this novel problem, and to assess their performance as well as their impact on radio design. Several network architecture concepts were generated, as shown in Fig. 5.

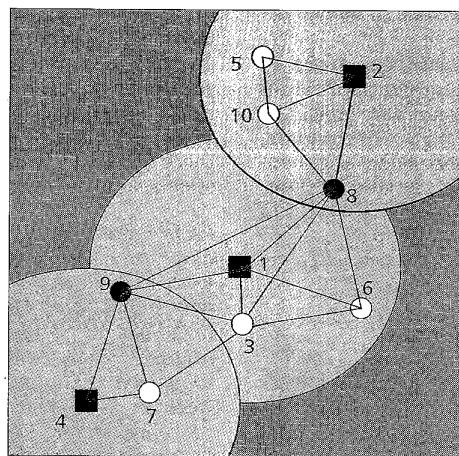
These architectures place different demands on the radios and use very disparate techniques for channel sharing, ranging from time division multiple access (TDMA) to CDMA to token protocol. However, they share some key concepts, most notably:

- The notion of clustering for resource accountability
- The dynamic virtual circuit scheme (for voice/video support) with resource reservation along the path
- QoS routing
- Dynamic transmit power control

In this article we will focus on the cluster architecture [5], which is perhaps the simplest to implement and thus was targeted for early demonstration in our testbed. For the other architectures the reader is directed to several published references [25, 26]. One should bear in mind, however, that many of the design concepts of the cluster scheme are shared by the other schemes.

A major challenge in multihop multimedia networks is the ability to account for resources so that bandwidth reservations (in a deterministic or statistical sense) can be placed on them. We note that in cellular (single-hop) networks such accountability is made easy by the fact that all stations learn of each other's requirements, either directly or through a control station (e.g., base station in cellular systems).

This solution can be extended to multihop networks by creating clusters of radios in such a way that access can be controlled and bandwidth can be allocated in each cluster. The notion of cluster has been used before in packet radio nets, but mainly for hierarchical routing rather than for resource allocation [27]. Two of the schemes reported in Fig. 4, namely Cluster and Token CDMA, make direct use of clusters. The remaining schemes, SWAN and Virtual Net, rely more heavily on CDMA for channel sharing and essentially define clusters of two nodes which communicate with a properly chosen code. While in Cluster and Token CDMA the clusters are spatially separate, in SWAN



■ Figure 6. Example of clustering.

and Virtual Nets the two-node clusters are typically overlapped. Regardless of the clustering method, a common requirement is the ability to dynamically reconfigure clusters in the face of mobility and failures. In the following subsection we describe the dynamic clustering scheme used in the Cluster architecture, evaluate the self-configuring properties, and illustrate the bandwidth allocation mechanism based on virtual circuits (VCs). We also show how VC bandwidth allocation is coupled with routing, more specifically with the QoS information conveyed by the routing algorithm.

Another challenge in mobile multimedia networks is to reroute VCs and reallocate bandwidth when radios move. This is particularly critical in multihop networks where not only sources and/or destinations move (as in cellular-type nets), but any node along the path can move. In a later section, we propose a "soft state" VC scheme coupled with fast reservations a la packet reservation multiple access (PRMA) [28] to attack this problem.

Power control is essential in multihop networks in order to achieve efficient spatial reuse. It is particularly critical in CDMA channels in order to mitigate the near-far effects [29]. The multi media support section describes a distributed power control solution suitable for SWAN as well as for Cluster.

Clustering and Virtual Circuits for Multimedia Support

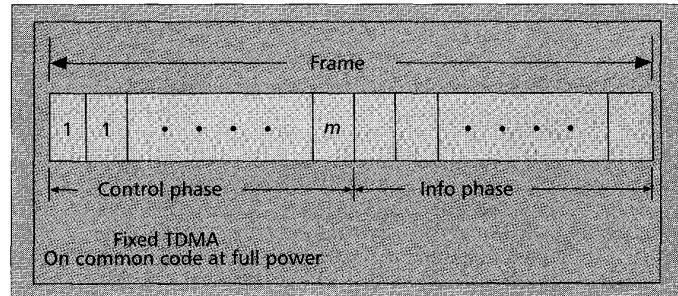
In order to support multimedia traffic, the wireless network layer must guarantee QoS (bandwidth and delay) to real-time traffic components. Our approach to providing QoS to multimedia consists of the following three steps:

- Partition the multihop network into clusters so that controlled, accountable bandwidth sharing can be accomplished in each cluster.
 - Establish VCs with QoS guarantee
 - Use QoS routing to keep track of QoS (bandwidth, SIR, etc.) along each path.

In this section we describe the implementation of these steps in the Cluster architecture, reminding the reader that similar functions are also present in the other architectures.

The objective of the clustering algorithm is to partition the network into several clusters. Optimal cluster size is dictated by the trade-off between spatial reuse of the channel (which drives toward small sizes), and delay minimization (which drives towards large sizes). Other constraints also apply, such as power consumption and geographical layout [5]. Cluster size is controlled through the radio transmit power, which may vary from cluster to cluster and should be dynamically adjusted as the node layout changes. For the Cluster algorithm, we have so far assumed that transmit power is fixed and is uniform across the network. Work is underway on the development of dynamic power and cluster size control.

Within each cluster, nodes can directly communicate with a clusterhead and can communicate with each other in at most two hops. The clustering algorithm is based on a distributed, lowest ID clusterhead election scheme which works as follows. Each node is assigned a distinct ID. Periodically, each node broadcasts the list of IDs it can hear (including its own) and elects the lowest ID node as its clusterhead. As a result of this "election," a node becomes either a *clusterhead* ■ or a *gateway* ● (i.e., a node connecting two or more clusters) or an *ordinary node* ○. In Fig. 6, nodes 1, 2 and 4 are clusterheads for their



■ Figure 7. Channel access frame.

coverage areas, respectively; nodes 8 and 9 are gateways. The clustering algorithm converges very rapidly, on average in the order of one broadcast cycle. In the worst case, the convergence is linear in the total number of nodes, N .

Within each cluster, the medium access control (MAC) layer is implemented using a time-slotted frame. That is, the transmission time scale is organized in frames, each containing a fixed number of time slots. The entire network is synchronized on a frame and slot basis using an elastic synchronization scheme [30]. The frame is divided into two phases, namely the control phase and information phase, as shown in Fig. 7.

The control phase consists of m control minislots where $m \leq N$ (note that $m = N$ in the simplest, most conservative implementation of the Cluster algorithm) [5]. This phase supports the communications needs of all control functions, such as slot and frame synchronization, clustering, routing, power measurement and control code assignment, VC reservation, and acknowledgments (ACKs). By exchanging connectivity information between "neighbors," each node elects its cluster-head and updates its routing tables. Clusterheads broadcast ACKs (for datagrams) as well as slot reservations and code assignment for each VC within their covering area. Each real-time connection is assigned to a VC. The VC is an end-to-end path along which slots have been reserved. The number of slots per frame assigned to a VC is determined by the bandwidth required by the VC. As we shall later see, the path and slots of a VC may change dynamically during the lifetime of a connection.

The information phase must support both VC and datagram traffic. Since real-time traffic (which is carried on a VC) needs guaranteed bandwidth during its active period, slots must be allocated to the VC at call setup time, as depicted in Fig. 8. The remaining (free) slots can be accessed by datagram traffic using a random access scheme (e.g., slotted ALOHA). To increase efficiency, carrier sense multiple access (CSMA) probing of VC slots by datagram stations can be employed in order to reuse the silent VC slots. While VC packets are neither ACKed nor retransmitted (because of real-time constraints), datagrams are retransmitted if an ACK is not

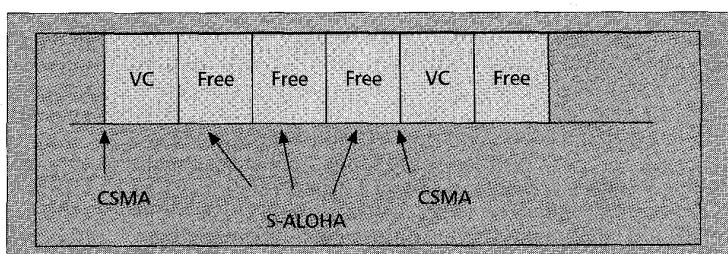


Figure 8. Slot allocation and channel access in the information phase sub-frame.

received. ACKs are cumulatively transmitted by the cluster-head in the control subframe.

To reduce interference between neighboring clusters a different spreading code is used in each cluster. Gateway nodes pseudorandomly retune (on a slot-by-slot basis) their receivers to the codes of the neighboring clusters in order to maintain connectivity with all neighbors. Within a cluster, CDMA can be used to increase channel capacity. Namely, two or more VCs can share the same time slot using different codes. In this case, transmit power must be carefully adjusted slot by slot in order to yield an acceptable signal-to-interference ratio (SIR), as discussed previously.

We should point out that the use of spread spectrum modulation (in our case, direct sequence spread spectrum, DS-SS) is required by the Federal Communications Commission (FCC) in the ISM band in order to limit the interference among different networks operating within the same frequency range. The use of spread spectrum for CDMA within each individual network, however, is optional. For example, many wireless local area networks (LANs) use only one code; furthermore, the leading wireless LAN MAC-layer protocol, IEEE 802.11 [31], does not exploit CDMA. In our Cluster scheme, we do use CDMA in conjunction with TDMA to improve performance, at the expense of additional protocol and implementation complexity. Thus, we use CDMA to allow transmissions to coexist in the same time slot in different clusters, which is required for proper gateway operation. Furthermore, in selected cases we allow VCs to share the same slot within a cluster via CDMA. Simulation results have shown that network throughput more than doubles when intracluster CDMA is used [5]. Finally, we mention that other architectures such as SWAN and Virtual Net make much more aggressive use of CDMA, still in conjunction with TDMA.

Routing is based on a distributed distance vector scheme [5]. Each node knows the next leg of the minimum hop path to each destination. Intercluster routing is restricted to pass through gateways. For instance, in the Fig. 6 example the route from 7 to 10 is {7, 9, 8, 10}. The gateway transit constraint stems from the fact that different clusters use different codes, and only gateways can "translate" codes. The basic Distance Vector [32] routing scheme has been augmented with QoS support. That is, bandwidth availability and various other channel attributes (e.g., SRI, packet drop rate, cyclic redundancy check — CRC — failure) can be propagated along with minimum hop distance information. The source node can make judicious use of the advertised QoS on the path by blocking calls with infeasible rate requests, by adjusting the rates to the advertised bandwidth, and by optimizing the video rate/channel coding strategy, as discussed previously.

Multimedia Support in the Presence of Mobility

As nodes move, routes must be updated and clusters must be dynamically reconfigured. Dynamic route updating is intrinsic in the distance vector scheme and therefore is automatically supported. Some extra care has been devoted to the suppression of temporary loops which occur during reconfiguration [33]. The distributed cluster algorithm can reconfigure clusters very quickly and efficiently in the face of mobility, as reported in [5].

More critical is the rerouting of VCs following node movements. In fact, a conventional VC setup scheme is suitable only for a static multihop network. In a highly mobile environment, the time required to set up a new VC (i.e., acquire bandwidth before data transmission) may be comparable to the interval between path changes. Thus, the conventional VC reconfiguration scheme (i.e., reset the VC end-to-end when the path fails) cannot keep up with station movements. To

Transmission range	35	40	50	60
Packets received	7357	8946	9900	9991
Packets dropped	2640	1051	98	7
Out of sequence	325	273	184	86
Number of loops	572	365	84	15
Mean delay	3.18	2.80	2.04	1.70
Average hops	2.53	2.31	1.84	1.60
Mean #. of clusters	5.45	4.59	3.50	2.89

■ **Table 1.** Maximum movement between route updates = 5 units (average speed = 54 km/hr).

handle mobility, we have developed a VC setup based on *fast reservations*.

In the fast reservation scheme, each packet in the VC stream is routed individually, based on its destination address (very much like a data packet). The difference, however, is that the first packet in the VC stream, upon successfully capturing a slot in the info subframe, will *reserve* it for all subsequent frames. If the slot remains unused for a certain number of frames, it is declared free by the cluster head and is returned to the free slot pool. This scheme, which was inspired by the PRMA protocol [28], allows the VC stream to dynamically select a new path to the destination when the old path fails. In this regard, the fast reservation VC is very different from the X.25 or asynchronous transfer mode (ATM) models where the VC is permanently associated with a path via fixed entries in VC tables at each intermediate node. Our solution is more akin to the "soft state" connection and flow ID concept proposed in RSVP [34]. Of course, each path change will cause some disruption (i.e., possible out of sequencing, delay to acquire a slot on the new path, lack of free slots, possible looping if the routing algorithm is not strictly loop-free, etc.). However, the disruption is much less severe than if a new path had to be entirely reconstructed end to end each time. This dynamic VC rerouting resembles the dynamic handoff and VC rerouting "with hints" in mobile radio networks proposed in [1]. The "hint" in our case is the change in routing table, which forces the immediate rerouting of the VC (without waiting for end-to-end service disruption and VC reinitialization). Our solution, however, is more general in that it handles multihop mobility, where any arbitrary node along the path may move, as opposed to [1], where only the end node is mobile.

The disruption caused by rerouting can be mitigated by exploiting the rate-adaptive, hierarchically encoded voice/video compression schemes presented earlier. Namely, "least significant" packets within a hierarchically encoded video or voice stream can be dropped during rerouting. This will alleviate congestion and reduce reservation delay if bandwidth on the new path is scarce, albeit at the expense of temporary signal quality degradation. Quality is restored automatically after rerouting, to the degree that there is sufficient bandwidth on the new path. Quality can also be enhanced by using rate adaptation at the voice/video source. Source rate control is based on adaptive quantization and coding, and is thus much more "bandwidth-efficient" than low-priority packet dropping. It has, however, a much slower response time because it must rely on network and/or destination feedback. Thus, low-priority packet dropping and source rate control ideally complement each other, the first providing fast reaction to network changes and the second ensuring long-term bandwidth efficiency.

The fast reservation scheme was evaluated using the parallel simulation package Maisie, developed at UCLA. The simulator is described in the next section. The simulated environment consists of 20 nodes randomly placed in a 100 x 100 square. Each node moves after each time division multiplexed (TDM) frame interval by x units with probability $P = 0.1$. For our applications, a realistic value for the TDM frame period (which is also the routing update period) is 100 ms, and for the unit of distance, $x = 10$ ft.

We assume that only one VC is present in the network, and monitor the performance on this VC as nodes move for various values of node speed and transmission range. Tables 1 and 2 report the results. In Table 1, the maximum speed is 5 units/frame with probability 0.1 (i.e., average speed of 50 ft/s or 54 km/hr). This corresponds to a relatively fast moving vehicle. A total of 10,000 packets are transmitted. Of these, some are dropped if the destination has become temporarily unreachable. Some are delivered to the destination out of sequence, partly because of dropping and partly because of route changes or temporary loops. Number of loops (caused by routing table inconsistencies during transitions), average end-to-end delay (measured in number of time frame periods), average number of hops covered, and average number of clusters are also reported. Note that the performance is quite sensitive to transmission range. In this case, a tx range = 50 seems to be the minimum acceptable value in order to support voice (with < 2 percent packet loss).

In Table 2, sensitivity to mobility is explored. At an average speed of 25 units/frame (equivalent to 270 km/hr), the number of out-of-sequence packets is 5 percent, thus rendering voice quality very poor. Clearly, this speed is well beyond the limits of our basic scheme; a more robust routing algorithm must be used which exempts, for example, fast-moving nodes (e.g., aircraft) from store-and-forward routing responsibilities, or which resorts to flooding. Such an algorithm is the subject of our current investigations.

Power Control for Bandwidth Reuse and Mobility Management

By controlling the transmitter powers, we can mitigate interference in a wireless network and improve spatial reuse of the radio channel, increasing network capacity. Transmitted signals reach their intended receivers while generating minimal interference on others sharing the same channel and minimizing the power needed to maintain transmission quality (hence increasing the battery life of mobile units). In particular, during the operation of the Clustering Algorithm power control allows for the reuse of the bandwidth resources by multiple transmission links throughout the network. We have developed a suite of algorithms for power control, user admission, and mobility management in this project. Its key objective is the maintenance of the QoS of operational links (transmissions) in the network at acceptable levels as new links try to establish themselves. At the core of this suite of protocols is the Distributed Power Control with Active Link Protection (DPC/ALP) algorithm, which updates the powers of the transmitters of the links based on measurements of the SIRs at their intended receivers.

We briefly describe the

DPC/ALP algorithm below and discuss its key properties. In the interest of clarity, we employ the simplest possible setup in this discussion, considering the network as a collection of links (one-hop communications from transmitters to their intended receivers). Strings of consecutive links correspond to multihop network connections for communication sessions, but we can still consider them a collection of individual links for our purposes. The particular implementations in this project and the relevant mathematical analysis of the algorithms can be found in [7, 29, 35].

The DPC/ALP algorithm facilitates the efficient spatial reuse of the communication bandwidth resource by links of given QoS, allowing more links to coexist in the channel satisfying their SIR requirements. It is fully distributed (as seen later), and provides active link protection in the sense of maintaining the SIRs of currently operational links above their required threshold values at all times while new interfering links try to access the channel and establish communication by achieving their SIR target levels. However, if this is not possible, new links are simply suppressed and rejected without hurting the operational ones in the process. Guaranteed QoS through ALP is very important in multimedia networks where traffic has stringent QoS requirements.

For links to be interfering, they must use the same bandwidth resource (space, time, spectrum). The DPC/ALP algorithm powers up a new link by increasing its transmitter power by a small, fixed factor of δ in each power update. This gives all the operational links enough time to adjust their power in order to accommodate the new link without compromising their own QoS. Let the power updates be indexed by k in $\{0, 1, 2, 3, \dots\}$ (corresponding to time in some appropriate time units) and the links by i in L (where L is the set of all interfering links). Let $R_i(k)$ be the SIR measured at the receiver of the i th link after the k th power update. This link has a target SIR r_i that it needs to maintain at all times in order to sustain a certain QoS. We call the i th link *active* or *operational* during the k th step if $R_i(k)$ is larger or equal to r_i ; if it is less, we call the link *inactive* or *new*. Finally, let $P_i(k)$ be the transmitter power of the i th link during the k th step. The DPC/ALP algorithm updates the transmitter powers $P_i(k+1)$ at the $(k+1)$ th step according to the following rule:

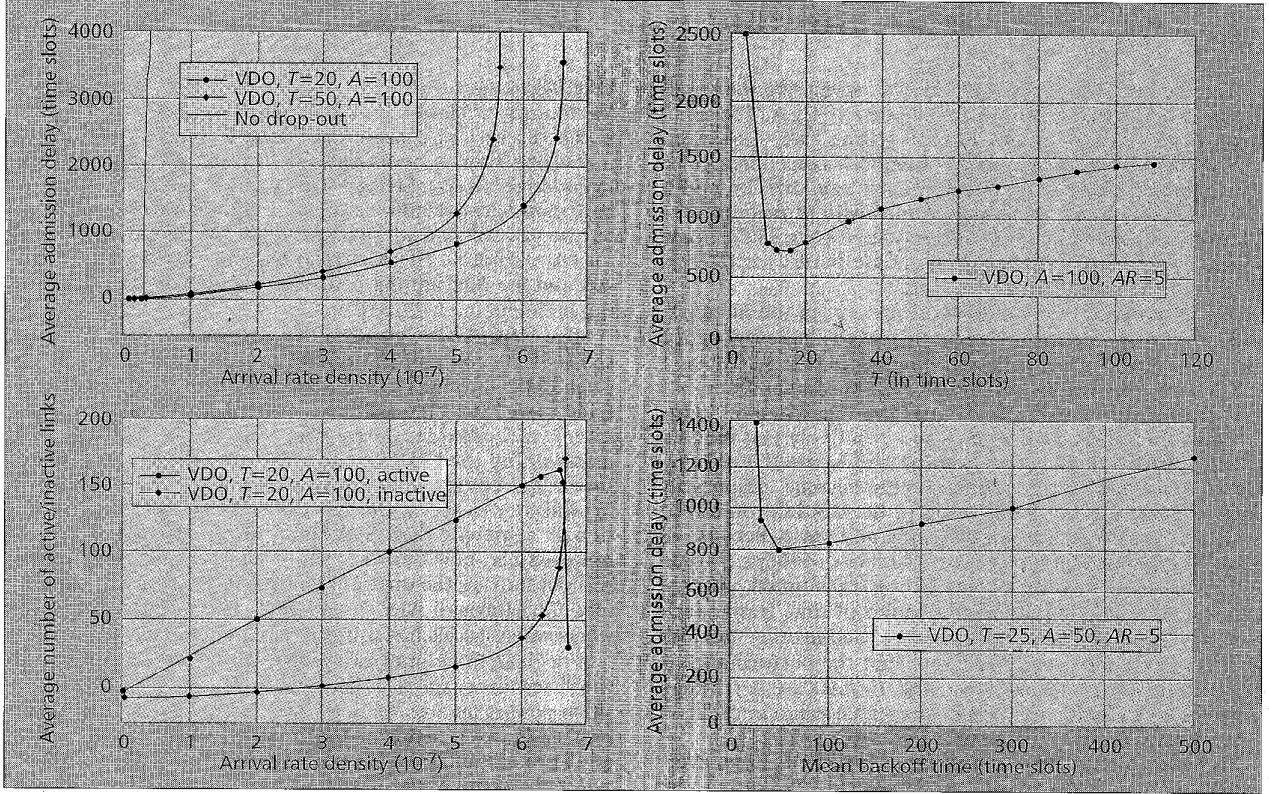
$$P_i(k+1) = \begin{cases} \frac{r_i \delta}{R_i(k)} P_i(k), & \text{if link } i \text{ is operational} \\ R_i(k) & \\ 8P_i(k) & , \end{cases} \quad \text{if link } i \text{ is a new link.} \quad (1)$$

As mentioned above, $R_i(k)$ is the SIR of the link measured at the k th iteration, and r_i is the target SIR for the i th link. It is easy to see that the algorithm is fully distributed since the power updates at any link are simply based on the SIR values measured at this link alone. Note that various links may have different SIR thresholds r_i corresponding to different QoS, BERs, and so on, which are specified by traffic type (data, voice, video) and the requirements of particular connections.

The DPC/ALP algorithm forces new links in the network to power up gradually by a factor of δ . Moreover, it boosts the target SIRs of all active links also by a factor of δ . This provides a protection margin for active links to absorb the

Maximum movement	1	5	25
Packets received	9995	9991	9767
Packets dropped	3	7	232
Out of sequence	24	86	474
Number of loops	2	15	137
Mean delay	1.91	1.70	1.73
Average hops	1.89	1.60	1.59
Mean #. of clusters	2.87	2.89	3.23

■ Table 2. Transmission range = 60 units.



■ Figure 9. Simulation performance curves of power control dynamics.

degrading effect of new links powering up in their vicinity without seeing their SIRs drop below their target values. This provable property of the algorithm is what we refer to as *active link protection*. Hence, active links remain active forever, or until they complete their intended communications and depart. On the other hand, another key property of the algorithm is that new links see their SIR increasing in every iteration. Therefore, there are two possible evolution scenarios for a new link:

- It may eventually see its SIR rising above its target threshold and becoming active (gaining admission to the network).
- It may see its SIR saturating below its target value.

In the latter case the link knows that it cannot be accommodated in the network at this time; hence it can voluntarily back off and try again later for admission. By backing off it reduces the interference on other links competing for admission, giving them a better chance to succeed. This dynamic process eventually forces a more efficient (dense) use of the communication resources increasing the network capacity.

We present here a simple simulation model and some performance results regarding the use of power control to regulate link admission in the network and hence to control congestion. Link arrivals refer to the initial requests to establish single-hop connections between nodes. The network scenario assumes that links arrive in the network at random times, experience some delay until gaining network admission, transmit for some time (service time), and then depart. The simulations demonstrate that it pays for a new link to *voluntarily* drop out when its objective of reaching some required SIR target r seems unrealizable. This mitigates congestion and may help other new links gain admission. The drop-out link can retry for admission later. We call this concept *voluntary drop-out* or *VDO*. When a new link i arrives at the network, it starts its local clock (counting steps k), begins transmitting at

a very low power $P_i(0)$ (see how low in [29]), and powers up according to the DPC/ALP scheme, implementing the following strategy.

The new link initially sets a *time-out horizon* T . If it has not gained admission by time T , it computes a *drop-out horizon* D as a decreasing function of $(r - R(T))$, for example, $D = Ae^{-\alpha(r - R(T))}$ (where $A, \alpha > 0$). The link keeps trying for admission until time $T + D$. Note that links closer to their SIR target r tend to try longer. Throughout D the new link hopes that some other link dropping out (or naturally departing) may boost its chance of admission. If that has not occurred by $T + D$, the link voluntarily drops out, setting its power to zero. After a new link drops out (backs off), it lies dormant for some time B and then retries for admission by starting to power up once again.

In Fig. 9, we see the performance curves of network dynamics obtained by simulation of the following specific scenario (physical units are normalized). Links arrive according to a Poisson process of arrival rate density AR (links per square length-unit), and are uniformly placed (in position and direction) on a square region 500×500 . Link lengths are exponentially distributed with mean 10. The power attenuation follows the inverse fourth power law (urban environment with multipath fading, log-normal shadowing, etc.), and the normalized noise floor is 10^{-9} . The SIR threshold is set to $r = 5$ (rather low, but necessary to achieve high active link density and speed up the simulation by inducing fast statistical mixing). VDO is used with $T, D = Ae^{-0.23(5 - R(T))}$. Unless otherwise specified on the graph, the standard operational point (around which we vary the parameters) is $\delta = 1.1$, mean back-off time $B = 100$, mean call time $S = 1000$, arrival rate density $AR = 5$. In the upper left graph, we see that when VDO is used the throughput increases more than 10 times (!) compared to when no drop-out is allowed; this highlights the importance of VDO for mitigating congestion by diffusing it

temporarily. The remaining curves show results of experiments to evaluate trade-offs among other parameters of the DPC/ALP/VDO scheme.

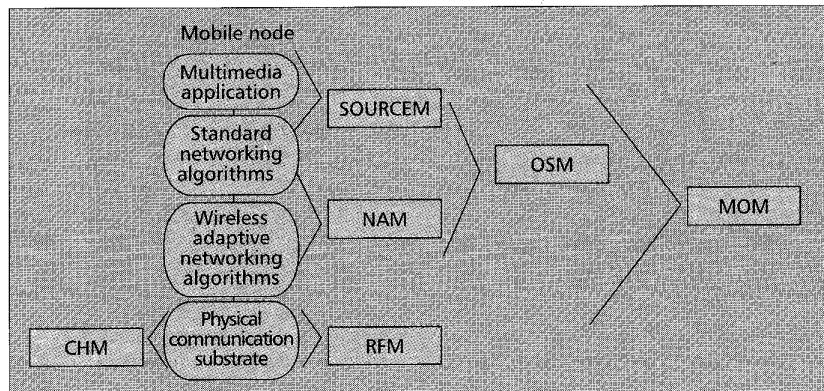
Another area where the DPC/ALP algorithm is useful is in mobility management. Since the nodes may continuously move around, their mutual interference changes dynamically in time with position. By adjusting the powers we can maintain the operational links and hence sustain the essential structure of the network, despite the fact that its actual topology is gradually stretched and twisted due to mobility. Hence, the DPC/ALP algorithm provides for considerable *network elasticity* in maintaining connections in the presence of mobility. Of course, power control can maintain existing operational links up to a point. Eventually, severe network deformation, due to changing node positions, may result in the total lack of a feasible power configuration to sustain all active links, which have now become incompatible. In this case, some links will have to be torn down and reconfigured in order to establish again the needed communication sessions throughout the network. This is a radical action which may be quite costly in terms of overhead. The network elasticity provided by the DPC/ALP algorithm makes this drastic action quite infrequent for reasonable degrees of node mobility and network loads.

Network Simulation Environment

As discussed in the previous sections, a diverse set of parameters and trade-offs must be considered in the design of adaptive network protocols to support multimedia applications. The performance of the protocols depends on a combination of factors that include traffic characteristics (for instance, video or audio coder complexity, quantization scheme, hierarchical or embedded encoding), mobility patterns (static, random, constant drift, variable velocity, etc.), application objectives (e.g., maximize throughput or minimize noise/loss), processor characteristics (central processing unit — CPU — speed, network interface, etc.), and radio or channel characteristics (bandwidth, power, etc.), to name a few. Although analytical models are often appropriate for studying isolated trade-offs in a specific domain, a detailed evaluation of the interactions among the multiple components of these protocols often requires the use of simulation.

A number of simulators, including commercial packages, have been designed to study network protocols in the wired domain. These simulators were found to be inadequate for our research for a number of reasons. First, most simulators provide limited support for node mobility and wireless channel models. Second, most existing tools only support sequential execution of the models. The execution time for models involving even a moderate number of mobile nodes can become excessive, making it infeasible to undertake scalability studies that involve thousands of mobile nodes. Last, with many existing simulators, the simulation models are of little use in generating protocol implementations. Because substantial effort must be invested in the design of detailed simulation models, it is desirable for the modeling environment to provide a path whereby the simulation models can easily be transformed into operational code.

A modular and hierarchical simulation environment is being developed at UCLA that addresses the preceding issues. We use the simulation environment to:



■ Figure 10. Mapping of the reference model to simulation models.

- Predict the performance of multimedia traffic transmissions over a wireless network
- Estimate performance of protocols at different layers in the mobile networking infrastructure described in the previous section
- Provide a link between simulation and implementation for code porting

Our eventual goal is to develop an environment in which video or audio coding quality can be evaluated as a function of the QoS guaranteed by the network, measured in terms of mean delay, packet loss rate, and similar metrics. We envisage running scripts or traces of the video/audio traffic against network models spanning much of the design space outlined in the preceding sections to identify, using appropriate heuristics, the optimal parameters for the protocols, radios, and video and speech codecs. We begin with a description of the simulation environment and subsequently illustrate its use in performance modeling of multimedia network protocols.

Mobile System Simulation Environment

The simulator is being built on an existing parallel discrete-event simulation language called Maisie [36]. Maisie provides a natural paradigm, based on message passing, for modeling networks. A Maisie program can be executed, with minor changes, using both sequential and parallel simulation algorithms. When a Maisie program is compiled, the analyst can indicate the specific simulation algorithm that is to be used to synchronize execution of the model: sequential, parallel conservative [37], or parallel optimistic [38]. A variety of parallel simulation algorithms are supported because no single algorithm has been found to offer better parallel performance across a wide set of models. Parallel Maisie implementations offer the possibility of evaluating much larger networks than would be feasible using only sequential implementations. Lastly, Maisie is built around a message-passing kernel called MPC. The MPC kernel can be used to implement general-purpose parallel programs. The primary difference between Maisie and MPC is that Maisie programs use a virtual clock and hence execute in logical time, whereas MPC programs use a physical clock and hence execute in physical time. As both models and programs are written using essentially the same set of constructs, in principle this notation may be used to design hybrid models that consist of simulation models of the subsystem interspersed with actual implementations of other subsystems [39]. The commonality of constructs in Maisie and MPC also supports the conversion of a simulation model into operational code. The process of converting a network model to operational code is discussed at the end of this section.

The simulation environment, called WAMISSIM, is decomposed into components that mimic the reference model

Propagation model	Free space Rate control	Fading Rate control	Free space No rate ctrl.	Fading No rate ctrl.
Packets received	99.8%	89.4%	81.2%	45.9%
Packets dropped	0.2%	10.6%	18.8%	55.1%
Out of sequence	1.0%	18.4%	0.6%	6.0%
Fraction of time when rate was reduced by half	0.7%	16.4%	N/A	N/A
Mean delay	1.17	2.70	3.11	9.22

Table 3. Cluster algorithm simulation experiment: impact of hierarchical encoding on VC performance (20 nodes in a 300m x 300m square; transmission range = 150 m; maximum speed = 11 km/hr, average speed = 1.1 km/hr).

described previously. This decomposition allows the environment to support a “plug-and-play” capability that generates composite prototypes constructed from pieces that model system components at widely differing levels of detail. The primary components of the WAMISSIM environment are:

- Operating system models (OSMs)
- Application traffic models (SOURCEMs)
- Network algorithm models (NAMs)
- Wireless radio models (RFMs)
- Channel models (CHMs)
- Mobility models (MOMs)

In Fig. 10, we see the relationship of the modeling components to the layers in the reference model as shown earlier in Fig. 3. The OSM simulates the relevant portion of the operating system kernel that is involved in interfacing with the application (e.g., delivery of incoming messages) or with the network (e.g., transmission of a remote message). The OSM components include multitasking process scheduling, packet manipulation routines, time control, and interfacing, such as between the SOURCEM and NAM and between NAM and RFM. The SOURCEM components can be broken down into the source and destination streams (e.g., hard disk, keyboard, camera, screen, microphone, or speaker) corresponding to the voice, video, and data traffic, the control of these streams via the application, and the transport mechanism (e.g., TCP, UDP, or VCs) the application chooses to use.

The NAM components are broken down into internetwork models such as IP, and subnetwork control such as clustering. The subnetwork layer models are used to model the topology creation (instant infrastructure), reconfigurability, adaptive channel assignment (CDMA), and wireless multihop routing. The data link layer models are used to provide mobility and link-level control such as power control (utilizing various power levels available on the radio and adapting the SIR measurement), media access control via a TDMA-based time frame, and logical link control such as providing a hop-by-hop acknowledgement scheme such as that described in [40].

The MOM components are responsible for node movement patterns such as speed and

direction. The CHM components are responsible for the transmission media, including the range over which two nodes are able to communicate, and environmental effects such as multipath fading, shadowing, and interference. The RFM components are responsible for the physical layer modeling of the radio frequency modem and include the raw channel bandwidth, modulation techniques, and acquisition delays.

Simulation Results

A number of models have been designed using the preceding simulator, including models of the instant infrastructure protocols [41] and a realistic channel fading model [42]. Several results produced by such models were reported previously.

Those models addressed the behavior of specific network algorithms. Here, we present an additional set of models and experiments that go beyond the specific algorithms and are aimed at evaluating the interaction between network and voice/video characteristics. The experiments demonstrate the effectiveness of hierarchical encoding of voice/video traffic. For a given network scenario, the simulations measured packet loss and mean delay with and without hierarchical encoding. We recall that when hierarchical encoding is used, high-resolution packets are dropped when bandwidth is scarce (e.g., during VC rerouting). Different radio channel propagation models were used (free space and fading), thus showing the impact of the channel model on network performance.

Four experiments were run, as reported in Table 3. The first two experiments (two left columns) assume hierarchical encoding and therefore reduce the voice/video rate by a factor of two when the VC is rerouted onto a new path and bandwidth is scarce. The first experiment uses a free-space propagation model, while the second experiment uses a channel-fading model [42]. Note the dramatic effect of fading on packet loss performance. With a free space channel, the loss is only 0.2 percent, yielding very good quality. With the more realistic fading model, the performance (10 percent packet loss) is inadequate. The second pair of experiments (two columns to the right) assume no hierarchical encoding. Thus, the packet loss following a rerouting is much more severe than in the first set of experiments. We note that even when the free space model is used, the packet loss rate (18.8 percent) is very severe. The mean delay measurements are consistent with the packet loss results. Namely, mean delays (expressed in time frame units) are about three times lower in the rate reduction case than in the no reduction case. This type of experiment has been very helpful in assessing the interaction between source coding and network algorithms. The results guide us in setting the proper priorities for the various voice/video substreams. More experiments are in progress to evaluate the impact of making path bandwidth and QoS information available to the voice/video source, and to design efficient strategies for source rate

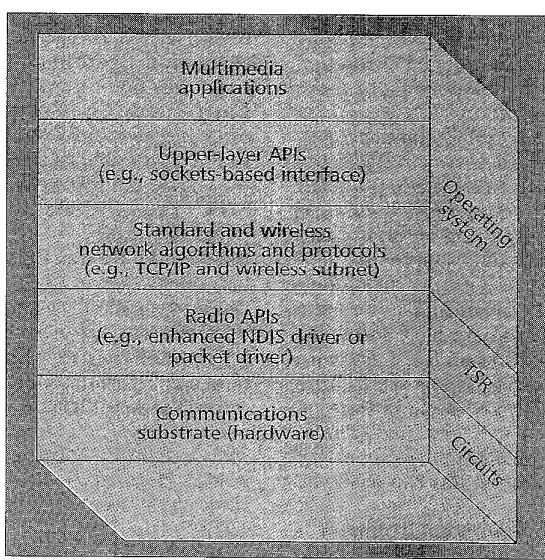


Figure 11. Testbed interfaces and components.

adjustment, dynamic channel encoding, and low-priority packet dropping. The above experiments have also clearly shown the impact of channel characteristics on performance, prompting us to consider revisions to algorithms and radios (e.g., antenna diversity) in environments in which multipath fading is predominant.

From Simulation Models to Implementation

To preserve the significant investment in developing detailed simulation models, it is desirable to support the transition of the simulation models into operational software. Not only does such a port allow effective reuse of code, it also ensures consistency between the model and the eventual implementation. As discussed earlier, it is relatively easy to transform Maisie simulation models into operational code because of the common set of message-passing primitives used by both environments. However, to support the automatic conversion of network models to implementations, the simulation environment must provide a modular interface to the network operating system. This interface allows the code generated from the model to be directly composed with the rest of the operating system.

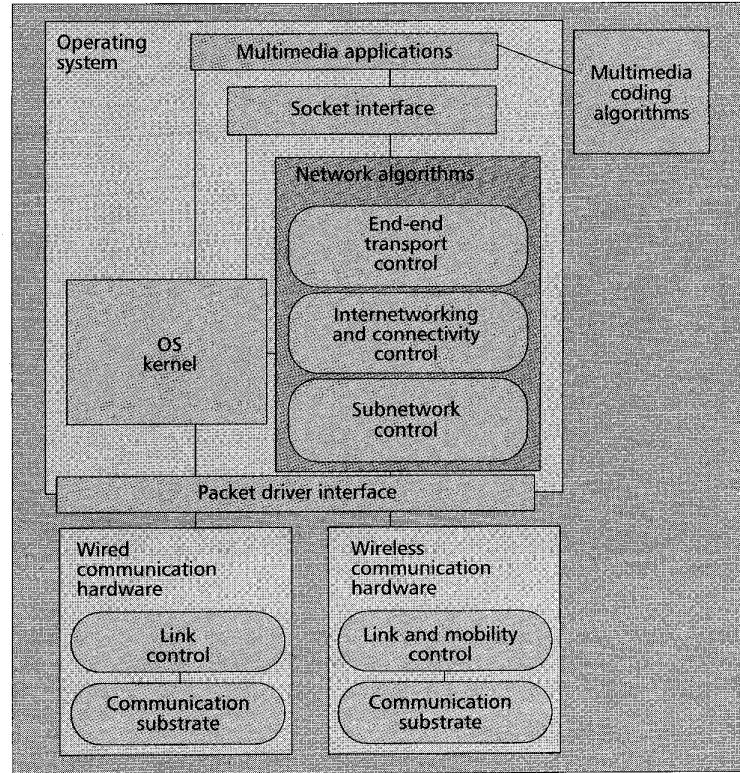
In the existing environment, we have developed such an interface between the simulator and an existing network operating system called Wireless Adaptive Mobile Information System Network Operating System (WAMISNOS). WAMISNOS is based on DOS, and the initial interface was designed primarily to demonstrate the automatic capability to transform the simulation models into operational code. Interfaces to link the simulation environment to Linux are in progress, and similar interfaces to commercial laptop operating systems such as Windows are envisaged in the near future.

The existence of an interface between the simulation models and the operating system also allows existing protocol implementations to be incorporated in the simulation model. This is desirable if the functionality needed in the simulation model is already available within some protocol implementations. Inclusion of existing protocol implementations within the simulation model is referred to as *backporting*. To support automatic porting of models and backporting, the interfaces among the various components of the simulator must be designed with care so that it is easily possible and practical to replace the simulation model at the interface with the corresponding code.

System Implementation

A testbed has been developed to carry out numerous networking experiments to test out the various network control algorithms and their interaction with other system components developed in the simulation environment. A video conferencing application (along with other node and network testing tools) was developed in WAMISNOS which provided performance statistics and demonstrated the integration of video and data (TCP/IP) in the multihop wireless and dynamic network connectivity environments.

The migration of the model towards an actual protocol implementation has been demonstrated in the context of the instant infrastructure protocols. A model of this protocol was initially developed using our simulator, as discussed in the



■ Figure 12. System component integration.

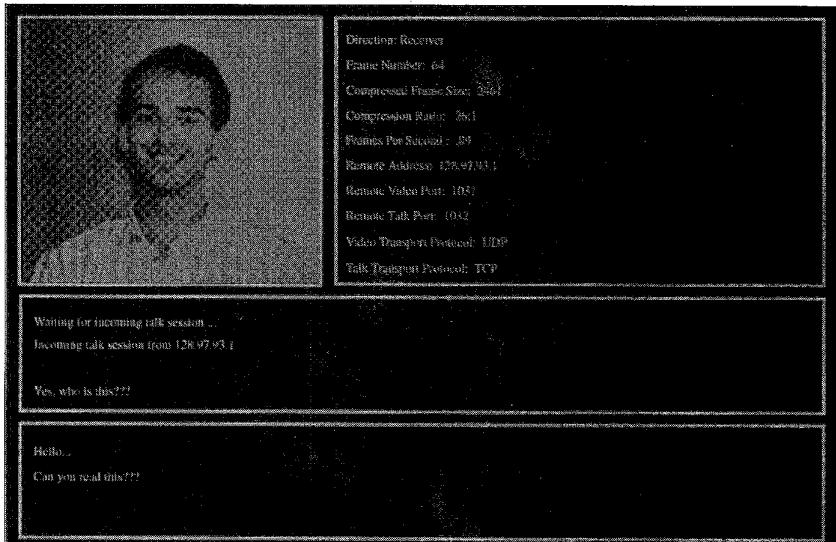
previous section. The model was used to evaluate the impact of mobility and CPU loading on network connectivity [41]. Subsequently, the model was transformed into an operational implementation on a network of four laptops running WAMISNOS.

This rapid prototyping of networking algorithms allows the designer to use and do real-time testing of the node and the network. Manual coding and implementation is not only time-consuming, but introduces too many variations in the functionality of the node and the network for which typical simulations today do not account. This environment can be extended to provide additional functionality, such as bringing together the simulation environment and implementation system to perform hybrid simulations of the entire system with real system components. This simulation and automatic implementation environment not only bridges the gap between design, simulation, and implementation, but it also simplifies the process and reduces the time to market for such system designs.

System Components Interfaces

In order to integrate the various system components together, we utilize a set of standard interfaces, as shown in Fig. 11. The socket-based interface between the network control algorithms/protocols and the multimedia application is widely supported on many different platforms. There are also numerous proposals for ways to enhance the socket-based interface to better support multimedia networking, such as with QoS measurements and control parameters. We base our multimedia application on the standard TCP/IP-based sockets and provide enhancements as necessary to support the adaptive nature of our project.

A clear and concise definition of the interface, in Fig. 3, between the wireless subnet layer and upper and lower layers is essential for modular design and for interoperability of networking and radio modules. While specific interface require-



■ Figure 13. Video Talk conferencing program.

ments vary depending on the choice of wireless network architecture (e.g., Cluster vs. SWAN), the following APIs were found to be common to most schemes:

Wireless Subnet/Radio APIs –

- *Power control*: ability to control transmit power dynamically (on a packet-by-packet basis) over a properly defined range; ability to measure received power. These features are required for connectivity/topology management.
- *SIR measurements*: The SIR information is necessary to optimize transmit power in a CDMA environment.
- *Dynamic code selection*: ability to change the code (and possibly the chip rate) on a packet-by-packet basis. This feature is required for intercluster communications.

Wireless Subnet/Upper-Layer APIs –

- *QoS reporting*: The subnet makes available the following measurements: for each destination: available bandwidth; SIR value (the lowest value observed along the path); BER, or CRC failure rate. Available bandwidth information is used by the upper layers to block a call (when the bandwidth is insufficient), or to prompt voice/video sources to automatically adjust input rate (at call setup or dynamically during the life of the connection). Furthermore, SIR and BER measurements may be used by voice/video sources to optimize combined source and channel-coding strategies.
- *Bandwidth request*: The upper layer can specify the bandwidth (e.g., in slots per TDM frame) required for a given real-time connection.
- *Multiple-priority support*: The wireless subnet supports multiple priorities. These priorities are used to differentiate the various substreams in a hierarchically encoded voice/video stream, and to selectively drop the high-resolution substreams in case of internal network congestion, bandwidth contention after rerouting, or multicast delivery to heterogeneous destinations.

Our network control algorithms/protocols and communications substrate interface is based on FTP's Packet Driver specification. This interface allows various network interface cards (like the PI Card, which we use to interface to the UCLA Direct Sequence Spread Spectrum radio modem) to be used in place of one another without having to change the details of the network operating system. A packet driver is loaded which corresponds to the correct network interface card and networking equipment used. There are several other interface

standards, such as NDIS and ODI, which could be used as alternatives to or in conjunction with the FTP Packet Driver specification. These various interfaces and components are implemented in the testbed, as shown in Fig. 12.

Operating System

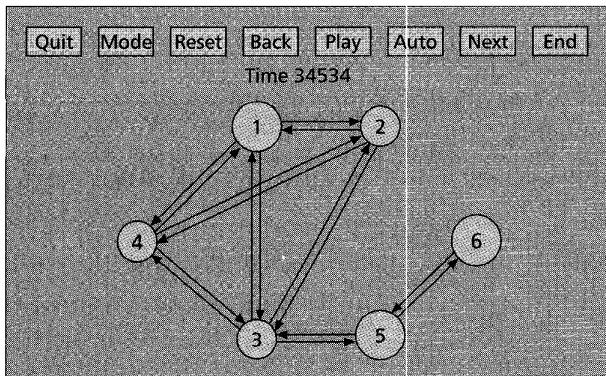
To integrate the various aspects of a mobile, instant infrastructure with the multimedia networking components, an operating system is desired such as that seen in Fig. 12. These components (applications, network algorithms and protocols, communications hardware, and multimedia compression hardware) are integrated together through the operating system. There are numerous commercial operating systems available today, including UNIX, PC-DOS, Microsoft Windows

95, and Windows NT, as well as network-based operating systems like Phil Karn's KA9Q NOS [43] and Novell Netware. A cross-platform implementation just for validation and experimentation is not feasible, so one usually has to start with a particular operating system. We initially chose DOS-based NOS as a starting point, due to its source code availability, low system requirements (640K), support for all NDIS and Packet-Driver-based communication hardware, and compatibility with all the PC-DOS-based laptops. WAMISNO \bar{S} , as defined earlier, is an enhancement of NOS to support the multimedia algorithms and subnetwork algorithms described earlier. Now, with the capability of direct porting between simulation and implementation described previously, porting to operating systems such as Linux and Windows can easily be achieved.

WAMISNOS is similar to Linux and Windows in that it includes its own multitasking scheduler. Since all of these operating systems, as well as the Maisie Simulation Language, support multitasking and synchronization, common interface routines are used to enable the porting between simulation and implementation. Various operating system semantics useful for protocol development are supported in the simulation and implementation environment. Each of the algorithms or protocols can be developed as its own process. Each multitasking kernel allows these algorithms and protocols to multi-task, sharing the CPU, and yet provide semantics such as wait and signal semaphores for interprocess (interalgorithm) communication. Time processing routines, such as TDMA, are able to sleep processes for a defined period of time, and can be used to allow other protocols and algorithms to run without halting or bogging down the CPU. Memory buffers (mbufs), also found in BSD UNIX system buffers, are used to minimize the overhead in copying packets in memory by linking together memory buffers for performing encapsulation, packetization, and so forth.

Video Talk – A Multimedia Experiment

In addition to the algorithm development and implementation, we have fielded a small testbed to demonstrate the performance of the wireless infrastructure using these algorithms. Indeed, we have measured network and link performance, experimented with live network configuration graphics, and supported video transmission while moving laptop-based terminals around in a laboratory environment [9]. In order to experiment with the wavelet-based video coding algorithm



■ Figure 14. Subnetwork topology analyzer program (TOPO)

described previously, we have developed a multimedia application called Video Talk. Video Talk is used to demonstrate the capabilities of adaptive multimedia data streams by displaying video and text (data) simultaneously on the same screen between any pair of nodes in the network. This video-conferencing-type program is illustrated in Fig. 13. With the current version of this program, video is sent using the connectionless, nonguaranteed delivery (but error-free) transport protocol UDP (User Datagram Protocol.) As we saw back in Fig. 2, the multimedia compression algorithm, here the wavelet-based video compression algorithm, is able to adapt to several QoS parameters. The Video Talk application thus sends out data and video through the standard networking algorithms to the subnetwork control algorithms at an acceptable rate. In Fig. 13, we see the specification on the rate and size of the video image as accepted and various statistics about the link and video rate in the upper righthand corner of the screen. In the bottom half of the screen are two boxes. The top box displays what the remote user types, and the bottom box displays what the local user types. This data is sent using TCP, thus using standard flow control techniques to determine the rate of data transfer. Of course there are numerous proposed improvements to the standard TCP/IP protocol, specifically proposed for wireless networks [44], which could have been utilized in this experiment. However, we chose not to modify the standard TCP/IP protocol stack since the goal was to promote interoperability with QoS measurements rather than to optimize TCP performance.

The video coding for the Video Talk application is performed in hardware on a Xilinx Field Programmable Gate Array (FPGA) and supports real-time video transmission at 8 frames/s using the algorithm described previously. The video images are captured using the Data Translation Frame Grabber Card located in the docking station attached to the laptop. The video frame images are 8-bit grayscale and 256 x 256 pixels. The codec implementation in a 10,000-gate FPGA was made possible by the low complexity of the video algorithms. By contrast, implementation of a commercial video standard would have required approximately an order of magnitude more gates. The upper-bound limitations due to the node implementation (such as the bus bandwidth limitation) is discussed in more detail in [9]. The maximum achievable frame rate (available user bandwidth) in a multihop environment is limited by the subnet control overhead, as mentioned earlier, due to the radio's capabilities and limitations. More specifically, with reference to the TDM frame shown in Fig. 7, we found that the adaptive 32 kb/s radios developed as part of this project require a control slot time of at least 100 ms and an info slot time of at least 200 ms for stable operation, due to spread code acquisition time constraints. This leads to a frame time of 1.2 s (i.e., 1 video frame/s) for a four-station network. Commercially available 1.6 Mb/s Proxim radios were

also tested, but did not yield much better performance in spite of the much higher channel rate. This is because the built-in retransmission scheme at the link layer caused a large variance in packet delays, and thus forced us to choose a very conservative (around 200 ms) slot time. With 2 Mb/s AT&T WaveLAN radios [45], with no link-level retransmissions, we were able to reduce total frame time to 400 ms (i.e., 3 video frames/s.). The above results clearly indicate that the radios are currently one of the main obstacles in achieving high performance in a multimedia multihop environment. Leveraging on this experience, a second generation of radios are now being developed at UCLA which will overcome these limitations [9].

Cluster/Topology Demonstration

One of our goals in a multihop mobile multimedia network is to dynamically adapt the subnet topology as nodes move around. In order to demonstrate how the subnetwork forms as nodes move, the subnetwork clustering algorithms were ported into WAMISNOS from simulation, and a topology analyzer program was developed to graphically illustrate the current topology. A typical output of the topology analyzer program, called TOPO, is shown in Fig. 14. This program runs under both WAMISSIM and WAMISNOS, so topology analysis can be done first in simulation and verified later in implementation using TOPO. In the screen dump shown in Fig. 14, we find that there are currently six nodes in the wireless subnetwork which are partitioned in two different clusters. Nodes 1 and 5 are clusterheads, as determined by the algorithm in the section on clustering and VCs and highlighted in Fig. 14 by the larger circles. Nodes 1, 2, 3, and 4 are fully connected in the node 1 cluster. Node 3 belongs to both the node 1 and node 5 clusters. Node 6 can only reach the other nodes in the wireless subnet by multihopping through nodes 5 and 3. Each laptop displays its own version of the subnetwork topology using TOPO. Thus, we are able to view the subnetwork topology in real time on each node as nodes move. Changes are shown on the screen. In [41], we document link failure rate measurement (the rate at which a link drops between two nodes) and the rate at which clusterheads change. The TOPO application buttons shown in Fig. 14 are used to control the current topology display; for example, one can move backward or forward in time to step through a clusterhead formation.

Conclusion

The ARPA-supported WAMIS project has offered a unique opportunity to carry out a complete design of a wireless, rapidly deployable multimedia network. We have developed a methodology for overall systems design, and have implemented various layers of the network architecture, from the video/voice applications to the physical (radio) layer. This article has focused on two layers of key importance in multimedia wireless network design, namely compression algorithms and adaptivity in the voice/video applications layer, and network algorithms at the wireless subnet layer. Various design choices have been presented and evaluated. The notion of quality of service was introduced, and was shown to cut across the various layers, providing a means for applications and subnet layers to effectively interwork. Simulation tools were used to evaluate our design as well as to provide a path toward their implementation in software.

An important component of the WAMIS project has been the design of radios and DSPs. These developments have been reported elsewhere [9]. Simple demonstrations jointly testing the hardware, software, and network algorithms have been carried out.

In the course of this research, several lessons were learned, including:

- The importance of interaction (via APIs) of voice/video coding strategies and subnet algorithms and protocols
- The difficulty of multihop network design when rapid deployment, mobility, and multimedia are required. Among the key problems was the need to define "clusters of accountable bandwidth" and to carry out QoS routing and power control.
- The difficulty of building radios that provide the proper APIs and the desired performance
- The importance of simulation, both for performance evaluation of wireless mobile protocols and in developing protocol implementation

Research is progressing in many directions. In the voice/video layers, apart from the search for ever more efficient encoding schemes, major emphasis will be dedicated to the interaction with the subnet layer. In the subnet algorithm area, various algorithms are being revised and redesigned to address new challenges such as: mix of fast/slow nodes, cluster power adjustment, datagram flow and congestion control, and others. The simulation tool is being extended to operate in a parallel, multiprocessor environment (to handle large node populations) and to implement more realistic radio channel models. The laptop operating system environment and underlying radios are also evolving to provide a more efficient testbed for new algorithms and applications. Many of the lessons and results derived from this project can be directly applied to other wireless environments such as mobile computing, personal communications systems, and microsensor networks.

Acknowledgments

The authors would like to thank everyone on the UCLA WAMIS research project as well as others who contributed to the work described in this article. The authors are indebted to the anonymous reviewers for their thorough and constructive comments.

References

- [1] K. Keeton et al., "Providing Connection-Oriented Network Services to Mobile Hosts," *Proc. USENIX Assn. Mobile and Location-Independent Comp. Sys.*, Aug. 1993, pp. 83-102.
- [2] R. E. Kahn et al., "Advances in Packet Radio Technology," *Proc. IEEE*, vol. 66, no. 11, Nov. 1978, pp. 1468-96.
- [3] B. Leiner, D. Nielson, and F. Tobagi, Special Issue on Packet Radio Networks, *Proc. IEEE*, Jan. 1987.
- [4] IP Mobility Working Group, "Routing Support for IP Mobile Hosts," Internet Engineering Task Force (IETF), Internet Draft, ver. 12, 1995.
- [5] M. Gerla and T.-C. Tsai, "Multicluster, Mobile, Multimedia Radio Network," *ACM Baltzer J. Wireless Networks*, vol. 1, no. 3, 1995, pp. 255-65.
- [6] S. Chen, N. Bambos, and G. Pottie, "On Power Control with Active Link Quality Protection in Wireless Communication Networks," *Proc. 28th Annual Conf. on Info. Sci. and Sys.*, Princeton University, Princeton, NJ, 1994.
- [7] N. Bambos, S. Chen, and G. Pottie, "Radio Link Admission Algorithms for Wireless Networks with Power Control and Active Link Quality Protection," *Proc. IEEE INFOCOM '95*, Boston, MA.
- [8] D. C. Cox, "Wireless Personal Communications: What Is It?" *IEEE Personal Commun.*, Apr. 1995.
- [9] R. Jain et al., "PC-Netbook Based Mobile Networking: Algorithms, Architectures and Implementation," *Conf. Rec. IEEE Int'l. Conf. on Commun.*, vol. 2, June 1995, pp. 771-77.
- [10] M. Khansari et al., "Approaches to Layered Coding for Dual-Rate Wireless Video Transmission," *IEEE Int'l. Conf. on Image Processing*, Austin, TX, Nov. 1994.
- [11] A. Chandrakasan, A. Burstein and R. Brodersen, "A Low Power Chipset for Portable Multimedia Applications," *Proc. IEEE Int'l. Solid-State Circuits Conf.*, San Francisco, CA, Feb. 1994, pp. 82-83.
- [12] G. K. Wallace, "The JPEG Still Picture Compression Standard," *Comm. ACM*, vol. 34, 1991, pp. 30-45.
- [13] D. Le Gall, "MPEG: A Video Compression Standard for Multimedia Applications," *Comm. ACM*, vol. 34, 1991, pp. 46-58.
- [14] C. Chien et al., "A 12.7 Mchips/sec All-Digital BPSK Direct-Sequence Spread-Spectrum IF Transceiver," *IEEE J. Solid-State Circuits*, vol. 29, no. 12, Dec. 1994.
- [15] R. Stedman et al., "Transmission of Subband-Coded Images via Mobile Channels," *IEEE Transmiltomm. Circuits Sys. Video Tech.*, vol. 3, Feb. 1993, pp. 15-20.
- [16] J. Villasenor, B. Belzer, and J. Liao, "Wavelet Filter Evaluation for Image Compression," *IEEE Trans. on Image Proc.*, vol. 4, Aug. 1995, pp. 1063-70.
- [17] M. Antonini et al., "Image Coding Using Wavelet Transform," *IEEE Trans. on Image Proc.*, vol. 1, 1992, pp. 205-20.
- [18] D. A. Lelewer and D. S. Hirshberg, "Data Compression," *ACM Comp. Surveys* 19, 1987, pp. 261-96.
- [19] A. Shen, "Perceptually-Based Subband Coding of Speech Signals," Master's thesis, Electrical Engineering Dept., UCLA, June 1994.
- [20] A. Shen et al., "A Robust and Variable-Rate Speech Coder," *Proc. Int'l. Conf. Acoustic Speech Sig. Proc. (ICASSP)*, vol. I, 1995, pp. 249-52.
- [21] R. Cox et al., "New Directions in Subband Coding," *IEEE JSAC*, vol. 6, Feb. 1988.
- [22] K. Brandenburg and T. Sporer, "NMR and Masking Flag: Evaluation of Quality Using Perceptual Criteria," *Audio Eng. Soc. Test and Measurement Conf.*, 1992.
- [23] S. Furui, *Digital Speech Processing, Synthesis, and Recognition*, New York: Marcel Dekker, 1989.
- [24] A. Gershoff, "Advances in Speech and Audio Compression," *IEEE Proc.*, June 1994.
- [25] C. Lin and M. Gerla, "A Distributed Control Scheme in Multi-Hop Packet Radio Networks for Supporting Voice/Data Traffic," *Proc. IEEE Int'l. Conf. on Commun.*, 1995.
- [26] K. Scott and N. Bambos, "The Self-Organizing Wireless Network (SWAN) Protocol For Communication Among Mobile Users," *Proc. IEEE GLOBECOM '95*, Nov. 1995, pp. 355-59.
- [27] J. Jubin and J. D. Tornow, "The Daprà Packet Radio Network Protocols," *Proc. IEEE*, vol. 75, no. 1, Jan. 1987.
- [28] D. J. Goodman and S. X. Wei, "Efficiency of Packet Reservation Multiple Access," *IEEE Trans. on Vehicular Tech.*, vol. 40, no. 1, 1991, pp. 170-76.
- [29] N. Bambos and G. Pottie, "On Power Control in High Capacity Cellular Radio Networks," *Proc. IEEE GLOBECOM*, vol. 2, 1992, pp. 863-67.
- [30] Y. Ofek, "Generating a Fault Tolerant Global Clock Using High-Speed Control Signals for the MetaNet Architecture," *IEEE Trans. on Commun.*, vol. 42, no. 5, 1994, pp. 2179-88.
- [31] IEEE, "Wireless Lan Medium Access Control (MAC) and Physical Layer (PHY) Specifications," July 1995.
- [32] D. Bertsekas and R. Gallager, *DATA NETWORK*, 1987.
- [33] S. Murthy, and J. J. Garcia-Luna-Aceves, "A Routing Protocol For Packet Radio Network," *Proc. ACM MobiCom '95*, Nov. 1995, pp. 86-95.
- [34] L. Zhang et al., "RSVP: A New Resource Reservation Protocol," *IEEE Network*, 1993, pp. 8-18.
- [35] N. Bambos, S. Chen, and G. Pottie, "Channel Access Algorithms with Active Link Protection for Wireless Networks with Power Control," *Tech. Rep. UCLA-ENG-95-114*, UCLA School of Engineering and Applied Science, 1995.
- [36] R. Bagrodia and W. Liao, "Maisie: A Language for the Design of Efficient Discrete-Event Simulations," *IEEE Trans. on Software Eng.*, vol. 20, 1994, pp. 225-38.
- [37] V. Jha and R. Bagrodia, "Transparent Implementation of Conservative Algorithms in Parallel Simulation Languages," *Proc. 1993 Winter Simulation Conf.*, Dec. 1993, pp. 677-86.
- [38] R. Bagrodia, K. M. Chandy, and W-T. Liao, "A Unifying Framework for Distributed Simulations," *ACM Trans. on Modeling and Comp. Simulation*, vol. 1, no. 4, 1991, pp. 348-85.
- [39] R. Bagrodia et al., "Parallel Logic Level Simulation of VLSI Circuits," *Proc. 1994 Winter Simulation Conf. - WSC '94*, pp. 1354-61.
- [40] J. Short, R. Bagrodia, and L. Kleinrock, "Mobile Wireless Network System Simulation," *Proc. 1995 ACM Int'l. Conf. on Mobile Comp. and Networking*, Berkeley, CA, Nov. 1995.
- [41] R. Bagrodia et al., "A Hierarchical Simulation Environment for Mobile Wireless Networks," *Proc. Winter Simulation Conf.*, 1995.
- [42] E. Wu, J. Tsai, and M. Gerla, "A Spread Spectrum DS/CDMA Channel Fading Simulation Environment for Mobile Network Systems," to appear in *Proc. WMC '96*.
- [43] The KA9Q WWW server, <http://inorganic5.chem.ufl.edu/KA9Q>, July 1995.
- [44] H. Balakrishnan et al., "Improving TCP/IP Performance over Wireless Networks," *Proc. MobiCom '95*, Nov. 1995.
- [45] AT&T WaveLAN home page, <http://www.ncr.com:80/product/wavelan/>, the Internet, last updated Feb. 16, 1996.

Biographies

ABEER ALWAN received her Ph.D. in electrical engineering from MIT in 1992. Since then she has been an assistant professor in the Electrical Engineering Department at UCLA where she established the Speech Processing and Auditory Perception Laboratory. Her research interests include modeling speech production and perception mechanisms and application of these

models in speech coding, synthesis, and recognition systems. She is the recipient of the NSF Research Initiation Award (1993), the NSF Career Development Award (1995), the NIH FIRST Career Development Award (1994), and the UCLA-TRW Excellence in Teaching Award (1994). Dr. Alwan is a member of the New York Academy of Sciences, Eta Kappa Nu, Sigma Xi, and Tau Beta Pi. She has been a member of the Acoustical Society of America technical committee on speech communication since 1993 and has served as a member of the technical program committees of several conferences. E-mail: aber@icsl.ucla.edu.

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. E-mail: rajive@cs.ucla.edu.

NICHOLAS BAMBOS received his Ph.D. in electrical engineering and computer sciences from the University of California at Berkeley in 1989. He is currently an associate professor in the Electrical Engineering Department of the University of California at Los Angeles. He has received a National Young Investigator Award from NSF. His research interests are in the area of communication networks, protocol design, performance analysis and queueing modeling. E-mail: bambos@ee.ucla.edu.

MARIO GERLA received his graduate degree in electrical engineering from Politecnico di Milano, Italy, in 1966 and his M.S. and Ph.D. degrees in computer science from University of California, Los Angeles in 1970 and 1973, respectively. His current research projects cover topology design and bandwidth allocation in ATM networks; design and implementation of optical interconnects for supercomputer applications; and; network protocol design and implementation for a mobile, integrated services wireless radio network. E-mail: gerla@cs.ucla.edu

LEONARD KLEINROCK [F] received the B.S. degree in electrical engineering from the 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 networks, packet radio networks, local area networks, metropolitan area networks, and broadband and gigabit networks) and of parallel and distributed systems. He has had over 190 papers published and is the author of five books. Dr. Kleinrock is a member of the National Academy of Engineering and a Guggenheim Fellow. He is also the recipient of the Ericsson Award and the Marconi International Fellowship. E-mail: lk@cs.ucla.edu

JOEL SHORT [M] received the B.S. degree in computer science from the California State University, Chico, in 1992 and the M.S. degree in computer network modeling and analysis from the University of California, Los Angeles in 1995. He is currently a Ph.D. candidate in the Computer Science Department at UCLA, studying wireless and nomadic system simulation and rapid prototyping. His research interests also include development of software and hardware to support nomadic computing and communication. E-mail: jshort@cs.ucla.edu

JOHN VILLASENOR received the B.S. degree from the University of Virginia in 1985, the M.S. from Stanford University in 1986, and the Ph.D. from Stanford in 1989, all in electrical engineering. He has held postdoctoral appointments at Stanford and at CERFACS in Toulouse, France. From 1990 to 1992 Dr. Villasenor was with the Radar Science and Engineering section of the Jet Propulsion Laboratory in Pasadena, California, where he developed interferometric terrain mapping and classification techniques using synthetic aperture radar data. He is currently assistant professor of electrical engineering at the University of California, Los Angeles, and is conducting research on configurable computing and on wireless image and video communications. Email: villa@icsl.ucla.edu

We'd like to introduce you to the best connections in your industry

HNS Internet Address: resumes@hns.com

HNS WWW Home Page:

<http://www.hns.com>

National Staffing Center

Fax: (301) 428-1699

One of the world's fastest-growing telecommunications companies, Hughes Network Systems (HNS) provides wide area network solutions that set the pace in the industry. Connect with HNS in Germantown, Maryland or San Diego, California in one of the following areas:

Hardware Engineering

- Digital modulation
- Power systems design and distribution
- Digital and analog monitor and controls
- RF hardware design
- Frequency synthesizers
- ATM, frame relay
- ASIC, MMIC
- Datacom networking hardware development

Firmware Engineering

- DSP algorithm design, real-time DSP implementation
- Source (speech, image) and channel (FEC) coding
- C simulation



Demodulation

Hardware Interface

Senior Hardware Design

- Experience with high-volume consumer electronic product design
- Specification, design, implementation, test, documentation and production

Software Engineering

- C, Unix
- Real-time embedded microprocessors
- X-Windows, Motif, GUI
- Protocols (TCP/IP, SS7, ISDN, X.25)
- AMPS, D-AMPS, IS-54, TDMA, CDMA, GSM, PCS
- Interrupt processing/service routines, device drivers
- Network management (SNMP, Solstice)
- Telephone and data protocols (ISDN, SS7)
- Communication network simulation with OpNet
- Real-time call management using Unix
- i960, Vxworks, C, Assembly

- Configuration management
- System integration, test and verification

Use one of the connections listed above to forward your resume or find out more information about HNS. Please indicate location and technical discipline of interest. Or mail your resume to the location of your choice: Hughes Network Systems, Dept.

ATRADE, 11717 Exploration Lane, Germantown, MD 20876 or for San Diego: Hughes Network Systems, Dept. ATRADE, 10450 Pacific Center Court, San Diego, CA 92121. An equal opportunity employer.

HUGHES
NETWORK SYSTEMS

A HUGHES ELECTRONICS COMPANY

Wide area networks. Wide open opportunities.