Information Survivability for Mobile Wireless Systems

Thomas M. Parks and Clifford J. Weinstein

Mobile wireless networks are more vulnerable to cyber attack and more difficult to defend than conventional wired networks. In discussing security and survivability issues in mobile wireless networks, we focus here on group communication, as applied to multimedia conferencing. The need to conserve resources in wireless networks encourages the use of multicast protocols for group communication, which introduces additional security concerns. We point out the need for rate-adaptation techniques to simultaneously support multiple receivers that each experience different network conditions. The security properties associated with a number of approaches to rate adaptation are compared. We also identify several security issues for reliable group communication, providing examples of denial-of-service attacks and describing appropriate security measures to guard against such attacks. We examine the costs of these security measures in terms of network efficiency and computational overhead. Finally, we introduce a survivability approach called dynamically deployed protocols, in which the effects of an information attack are mitigated by dynamically switching to a new protocol to evade the attack. We suggest that this dynamic protocol deployment can be achieved effectively by transmission of in-line mobile code.

Information survivability encompasses many aspects of security and reliability for computers and networks. Information survivability is more than just preventing computer break-ins and protecting secret information from prying eyes. Systems must be robust; they must be able to continue to operate despite successful attacks that may compromise portions of the system.

Maintaining the security and survivability of mobile wireless systems is a complex challenge. Wireless communication links typically have lower bandwidth and reliability than the wire or fiber-optic links used in conventional networks because they are more susceptible to communication errors caused by noise and interference. Radio links can fade as computers in the network move in and out of communication range with their neighbors. Communication can be cut off altogether if computers move behind obstacles, such as buildings or hills.

Mobile wireless networks are more vulnerable to attack and more difficult to defend than conventional networks. Passive attacks, such as eavesdropping, are easy to accomplish on a broadcast radio link. It is much more difficult to gain access to a wire or fiber-optic cable that is embedded inside a wall of a locked building. Active attacks, such as interfering with communication or transmitting deceptive messages, are also easier to carry out on a wireless network.

Computer networks in general are vulnerable to a number of cyber attacks that include spoofing [1], hijacked connections [2], routing attacks [3], and various denial-of-service attacks [4–6]. In a spoofing attack, one computer assumes the identity of another computer in order to gain special privileges or to obfuscate the source of an attack. In a hijacking attack, one computer takes over a connection already established by another computer, bypassing security checks such as password authentication. In a routing

attack, a computer sends misleading information to routers in order to disrupt the normal flow of network traffic. In a denial-of-service attack, an adversary exploits flaws in network protocols or simply floods the network with useless data to consume resources and degrade the level of service provided to other computers in the network. The portable computers in a wireless system are vulnerable not only to these network intrusions, but also to theft and loss. The highly dynamic nature of a mobile wireless network further complicates the task of defending the network by making it difficult to distinguish a normal degradation in service from a malicious attack.

The security measures for conventional, wired networks are not always appropriate for mobile wireless networks. Network firewalls [7] and other similar measures can provide protection against many active attacks. However, these defenses depend on the existence of a bottleneck through which all traffic entering or leaving a network must pass. The changing topology of a mobile wireless network limits the effectiveness of such bottleneck defenses.

Computation and Communication Costs of Security Protocols

Computers communicate with each other over the network by exchanging discrete messages called packets. Each packet consists of a header and a payload. The packet header contains information such as the addresses for the source and destination computers, similar to the return address and destination address on the envelope of a letter sent through the mail. The payload contains the data to be delivered to the destination, similar to the letter inside an envelope.

Security protocols that use encryption [8–10] can protect against eavesdropping and, to a lesser extent, traffic analysis. Encryption prevents an adversary from reading the payload of intercepted packets, but any parts of the header left unencrypted, such as the destination address, can potentially provide useful information to an adversary. The protection provided by security protocols comes at a cost in terms of computation and communication. Although this cost can be insignificant in high-speed wired networks, it can be prohibitive in mobile wireless networks.

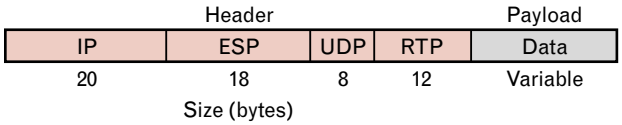

FIGURE 1. A packet is divided into a header, which carries control information such as addresses, and a payload, which carries data. Several protocols contribute to the size of a packet header. The Internet Protocol (IP) header is typically 20 bytes, though it can be longer with options. The Encapsulating Security Payload (ESP) header is typically 18 bytes, but it can be longer. The User Datagram Protocol (UDP) header is 8 bytes. The Real-Time Transport Protocol (RTP) header is 12 bytes. The payload has variable length. Here it is scaled to be 18 bytes, but it can reach a length of over 65,000 bytes.

Some encryption algorithms impose an excessive computational burden on low-power devices. Also, because many encryption algorithms operate on blocks of data, it may be necessary to append a few bytes to the end of a packet before encrypting it in order to pad its length to the next block boundary. A typical block size is 8 bytes, so it may be necessary to add up to 7 bytes of padding, depending on the length of the packet. Security protocols further reduce network efficiency by increasing the size of packet headers. These computation and communication costs increase power consumption, thereby shortening the battery life for mobile devices in a wireless network.

Traffic analysis is still possible even if encryption is used. Addresses and other identifying information in the packet headers can be left unencrypted, so that packets can be forwarded to the proper destination. Even if all header information is encrypted, an adversary can still observe the size and frequency of packets. To prevent such traffic analysis, it would be necessary to transmit continuously, filling gaps with random, meaningless data where the channel would otherwise be idle. Because all information would be encrypted, an adversary could not determine the boundaries between packets or distinguish the meaningful information from the meaningless. Achieving this level of protection may be prohibitively expensive in mobile wireless networks because of the need to conserve power and battery life.

Individual packets sent across the Internet typically have a 28-byte header for the Internet Procotol (IP)

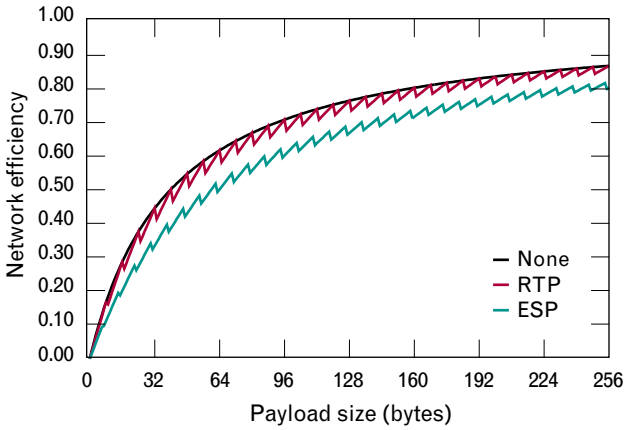

FIGURE 2. Reduction in network efficiency associated with security protocols. With no security protocols, the combined IP, UDP, and RTP headers produce 40 bytes of overhead, so that 50% efficiency is reached when the payload size is 40 bytes. If encryption is used with RTP, the size of the header remains 40 bytes, but up to 7 bytes of padding must be added for block encryption algorithms. If ESP is used instead, the header grows to 58 bytes and up to 7 additional bytes are required for padding.

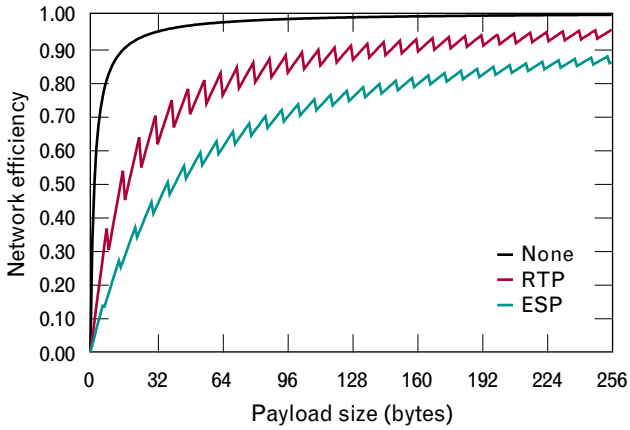

FIGURE 3. Improvement in network efficiency associated with header compression. With no security protocols, a 40-byte packet header can be reduced to 2 bytes, yielding high efficiency for all but the smallest packets. If encryption is used with RTP, 12 bytes of the packet header are encrypted, but the remaining 28 bytes of the header can be reduced to 2 bytes, yielding a 14-byte header. If encryption is used with ESP, 30 bytes of the header are encrypted, but the remaining 28 bytes can be reduced to 2 bytes, yielding a 32-byte header. In addition, up to 7 encrypted bytes of padding are required when using a block encryption algorithm with RTP or ESP.

and the User Datagram Protocol (UDP). Audio and video applications often use the Real-Time Transport Protocol (RTP), which adds 12 bytes to the header [11]. Using a security protocol such as the Encapsulating Security Payload (ESP) protocol adds another 18 bytes to the header [10]. Packets in real-time, interactive audio streams can carry as few as 18 bytes of data in the payload. Thus headers of 28, 40, or 58 bytes can introduce significant overhead compared to the size of the packet's payload. Figure 1 compares the sizes of these headers to an 18-byte payload.

Figure 2 shows network efficiency, which we define as the ratio of payload size to total packet size (payload plus headers), as a function of payload size for no encryption, encryption by using RTP, and encryption by using ESP. The efficiency decreases with the addition of padding and security protocol headers, and the decrease is most severe when the payload size is small.

A technique called header compression [12, 13] can be used to improve efficiency. Header fields that are constant or change predictably from one packet to the next, such as addresses or counters, may be omitted. Receivers can reconstruct the missing header fields by using information from earlier packets that had full, uncompressed headers. It is necessary to occasionally transmit packets with full headers to keep the receivers synchronized with the sender.

Unfortunately, encryption limits opportunities for header compression. Header fields that are scrambled by encryption change unpredictably from one packet to the next, so they must be transmitted in their entirety with each packet. For example, without encryption a typical 40-byte header can be compressed to 2 bytes. With encryption via RTP, 12 bytes of the 40-byte header are encrypted so the resulting compressed header is now 14 bytes. Figure 3 shows network efficiency with header compression as a function of payload size for no encryption, encryption by using RTP, and encryption by using ESP. Header compression improves efficiency dramatically when no encryption is used, but the improvement is more modest when portions of the header are encrypted.

Group Communication

In our work we have chosen to focus on group-communication applications because they pose difficult challenges for security and survivability in mobile wireless networks. There are many kinds of group-

communication applications, and each makes different demands on the network. Communication with small or large group sizes can be one to many or many to many. Some applications sacrifice some degree of reliability to reduce delays, while other applications require messages to be delivered reliably and in sequence. Several applications require receivers to acknowledge the receipt of messages.

Multimedia conferencing, which includes interactive audio, video, text, and drawings, can be extremely sensitive to network delays. In contrast, noninteractive applications such as E-mail can tolerate delays of hours or days in the delivery of messages. Audio is especially sensitive to delays. Users notice an audio delay of approximately 250 msec, which is associated with transmission via geosynchronous satellite. They find that longer delays interfere with the ability to carry on a conversation. Interactive audio is so sensitive to delays that it is preferable to tolerate occasional dropouts of data and reduced sound quality due to network errors rather than suffer the delays required to retransmit missing audio data. Interactive video can tolerate a moderate error rate because video data are transient: one video image is replaced by the next a fraction of a second later. Any imperfections in the video display will last only a short time, so there is little need to retransmit missing video data. In contrast, the transmission of text and drawings can be very sensitive to errors. The delay required to request the retransmission of missing data may be tolerated in exchange for reliability.

Multicast Transmission

Most network communication uses unicast transmission: each packet is delivered to a single destination. Group communication can be achieved through the use of multiple unicast transmissions: a separate copy of each packet is sent to each destination. For example, a participant in a video conference can transmit a stream of video packets to a server via unicast. This server can then retransmit a separate copy of each video packet to each of the other participants by using multiple unicast streams [14]. This approach wastes bandwidth, which is a precious resource in a wireless network, because the network links near the server carry multiple copies of the same video stream, as illustrated in Figure 4. This inefficiency may be acceptable for groups with only a few participants, but the problem grows worse as the group size increases.

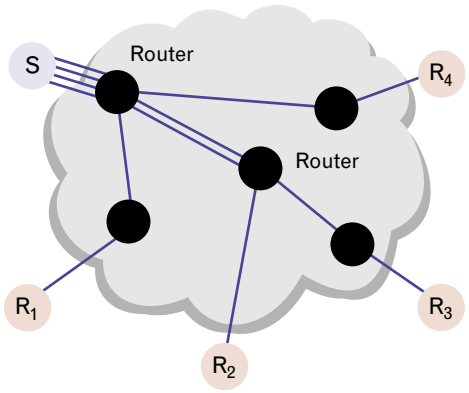

FIGURE 4. Unicast transmission from a single source S to multiple receivers R. The nodes in the figure (circles) represent computers and the lines represent data streams flowing along network communication links. Nodes within the network that are connected to several communication links are called routers because they must determine the route that traffic follows as it flows through the network. Unicast transmission consumes bandwidth because the network links near the source carry multiple copies of the same data stream.

Multicast transmission, which delivers a packet to multiple destinations, uses network bandwidth more efficiently. Routers within the network construct a tree that reaches all desired destinations. Packets are routed such that copies never traverse the same network link twice, as shown in Figure 5. Such network efficiency is especially important in mobile wireless networks. A video server could use multicast to deliver a video stream to all participants simultaneously, or the participants could use multicast to deliver video directly to one another without using an intermediate server.

Multicast Transmission Security

Several aspects of multicast transmission, as it is implemented in the Internet, lead to increased vulnerabilities [15]. In general, anyone can join a multicast group because standard Internet multicast protocols [16, 17] distribute the task of group management among the receivers rather than placing the entire burden upon the sender. Receivers simply send a subscription message to their local multicast router, and traffic for that group will be forwarded to them.

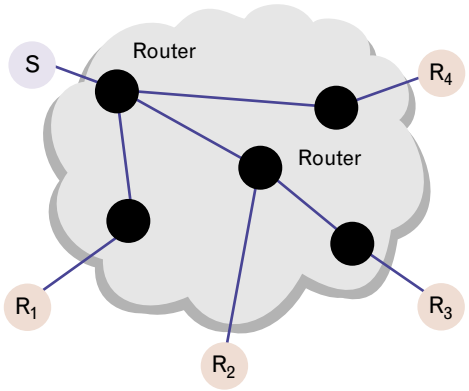

FIGURE 5. Multicast transmission from a single source S to multiple receivers R. The nodes in the figure (circles) represent computers and the lines represent data streams flowing along network communication links. Multicast transmission uses network bandwidth more efficiently than unicast transmission because only one copy of the data stream flows along any path.

This arrangement makes passive attacks such as eavesdropping and traffic analysis possible, even from remote locations without direct access to the local network where the traffic originates. Also, anyone can send traffic to a multicast group—even a nonsubscriber—making active attacks possible. Because multicast routers make copies of packets to forward them to multiple destinations, brute-force denial-of-service attacks that flood the network with packets benefit from a multiplier effect. With unicast transmission these packets steal bandwidth from only one receiver, while with multicast transmission these packets steal bandwidth from many receivers.

Adaptive Multicast Transmission

Distributing multimedia data, such as audio and video, to multiple receivers in a heterogeneous network environment such as the Internet requires applications that can adapt to changing network conditions. This requirement is especially important for mobile wireless networks where conditions change frequently and vary widely from one receiver to the next.

Several sources of variability make it necessary for applications to adapt. When multimedia data are distributed to multiple receivers, there is a different path from a sender to each receiver. In a heterogeneous network that includes fast, reliable fiber-optic links and slow, error-prone wireless links, paths to the receivers can have vastly different bandwidth limits. The available bandwidth along any one path from sender to receiver also changes with time. This situation may be due to such causes as competing network traffic or time-varying capacity and error rates of wireless links.

To compensate for this variability, receivers can provide feedback to the sender indicating packet loss rate or other measures of network conditions. In response, the sender can adjust the parameters of its compression algorithms to increase or decrease the rate at which it transmits data. Several such feedback control protocols have been developed [18–20].

Previous discussions of adaptation techniques for multimedia distribution have focused on performance: providing each receiver with the best quality possible for the given network conditions. The security implications of each of these adaptation techniques has only recently been discussed [21]. We review three different adaptation approaches: transcoding, simulcast, and layered coding. We summarize the advantages and disadvantages of each approach from a performance perspective. Then we discuss the security implications of these approaches.

Adaptation Techniques

Ideally, each receiver should receive the best possible media quality, given the available bandwidth along the path leading to it from each sender. This scenario could be achieved by having a sender transmit a separate media stream to each individual receiver via unicast, as in Figure 4. Each stream could be adapted independently with feedback from the receiver. However, transmitting multiple redundant streams wastes network bandwidth and does not scale to large numbers of receivers. At the other extreme, a single media stream could be multicast to all receivers in order to conserve network bandwidth and improve scalability, as in Figure 5. However, any approach that distributes a single data stream will be inherently unfair because a single data rate cannot simultaneously satisfy users with different available network bandwidth.

Transmitting data at a high rate to provide the best possible media quality to receivers along high-bandwidth paths will induce packet loss due to congestion,

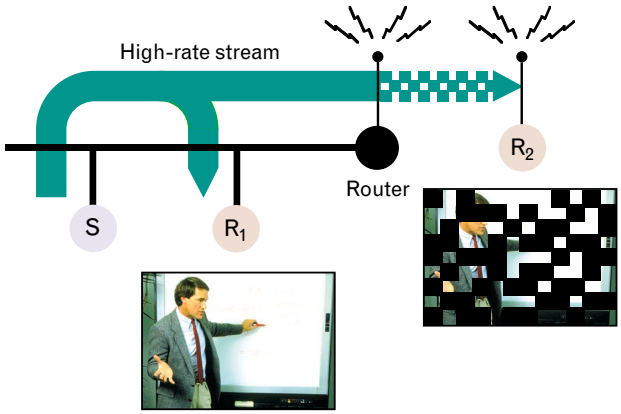

FIGURE 6. Packet loss and reduced media quality due to congestion. Low-bandwidth paths that include wireless links suffer network congestion, with a corresponding reduction in media quality, when the transmission rate is too high.

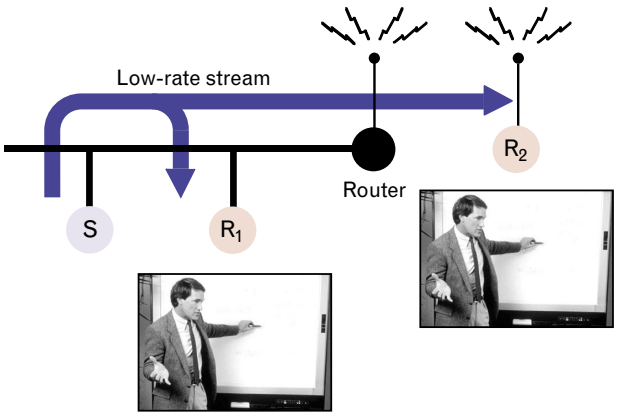

FIGURE 7. Effect of reducing the transmission rate to accommodate receivers along low-bandwidth paths. High-bandwidth paths are underutilized when the transmission rate is too low.

as illustrated in Figure 6. When packets arrive at the router faster than they can be forwarded to the low-bandwidth wireless network, some of the packets must be discarded, resulting in reduced media quality. Reducing the transmission rate to accommodate low-bandwidth receivers will unfairly reduce media quality for high-bandwidth receivers, as illustrated in Figure 7. Applications should be designed to support a broad range in network bandwidth and to dynamically adapt to time-varying network conditions in order to provide high-quality media to high-bandwidth receivers without causing congestion for low-bandwidth receivers.

Transcoding. A transcoder [22] can be employed to accommodate low-bandwidth receivers without sacrificing quality for high-bandwidth receivers, which is accomplished by translating media streams from one encoding to another more compact encoding, as illustrated in Figure 8. The sender transmits a high-quality media stream at a high rate matched to the bandwidth of the local network. A receiver on the same network can receive this high-quality stream. A transcoder, located at the boundary between the high-bandwidth network and a low-bandwidth wireless network, receives the high-quality media stream, translates it to a lower-quality stream, then retransmits this new stream at a lower rate matched to the bandwidth of the wireless network. A receiver on the wireless network can receive this translated stream. This technique provides the best possible media quality to each receiver. The quality for the receiver on the high-bandwidth network is similar to that in Figure 6, while the quality for the receiver on the low-bandwidth network is similar to that in Figure 7. The lowering of media quality is performed in a controlled way by the transcoder, unlike the situation in Figure 6, where packets are dropped in an uncontrolled way by the router when congestion occurs.

Simulcast. Simulcast [23] is an alternative to transcoding. To accommodate receivers with differing available bandwidth, a sender uses multiple encodings and transmits each version of the media stream to a different multicast group. Figure 9 shows the sender transmitting both high-quality and low-quality media streams on its local network. A receiver on the same network chooses to receive the high-quality stream. A receiver on the low-bandwidth wireless network chooses to receive only the low-quality stream. Because there are no subscribers on the wireless network for the mulitcast group with the high-quality media stream, it is not forwarded to the wireless network by the router. The primary disadvantage of simulcast is that it wastes bandwidth. However, the bandwidth overhead of transmitting to multiple groups affects only participants in high-bandwidth regions of the network, where bandwidth is plentiful.

Layered Coding. With layered coding [24–27], as with simulcast, media are distributed as multiple streams. Unlike simulcast, where data streams are redundant, these layered streams provide cumulative

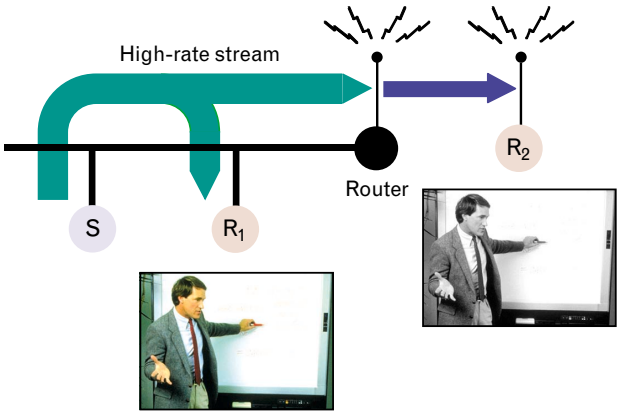

FIGURE 8. A transcoder translates data from one encoding to another more compact encoding for retransmission to receivers on low-bandwidth paths.

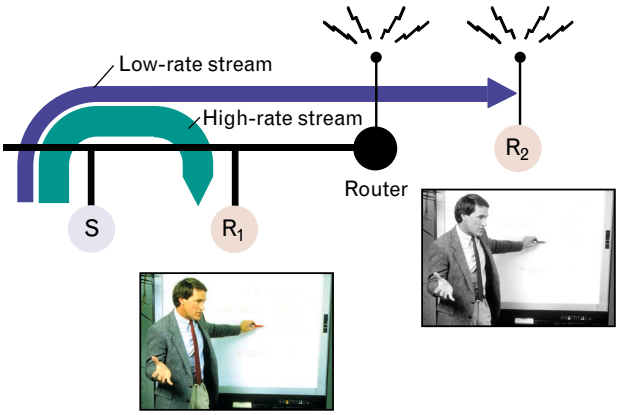

FIGURE 9. Example of simulcast transmission. The sender uses multiple encodings and simulcasts the data to different multicast groups.

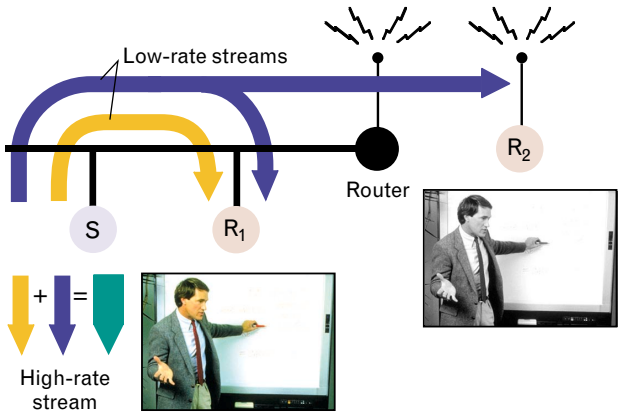

FIGURE 10. Example of layered encoding. The sender uses a layered encoding and transmits each layer to a different multicast group. Receivers can combine layers to obtain better quality.

information: receiving more streams provides progressively better media quality. Receivers react to changing network conditions by selectively subscribing to more or fewer streams to achieve the best quality for the available bandwidth.

Figure 10 shows an example of this approach. The sender transmits two streams on its local network. A receiver on the same network chooses to receive both streams, which it combines to form a high-quality version of the media. A receiver on the wireless network chooses to receive only one stream.

Layers may be independent or hierarchical. When the layers are hierarchical, the decoding of higher layers depends on having properly received and decoded all lower layers; thus transmission of hierarchical layers can be inefficient on lossy networks, such as mobile wireless networks. If a network packet for the lowest layer is lost or corrupted, the now useless packets for the other layers are still delivered, consuming network bandwidth. On the other hand, if the layers are independent, then these remaining packets still carry useful information.

If we assume that packets are lost with probability p, then the probability that a packet is received is $1-p$. When the layers are hierarchical, a packet is useful only when all lower layers have also been received. Thus the probability that a packet in layer n is useful is $(1-p)^n$.

If we define network efficiency to be the ratio of useful packets to total packets received, then the efficiency for a particular layer n is

$$\frac{(1-p)^n}{(1-p)} = (1-p)^{n-1}.$$

The combined network efficiency of transmitting N hierarchical layers is then the average of the efficiencies for each layer:

$$\frac{1}{N}\sum_{n=1}^{N}(1-p)^{n-1} = \frac{1-(1-p)^N}{Np}.$$

Figure 11 shows network efficiency as a function of p, the probability of packet loss, and N, the number of layers. Network efficiency decreases as more layers are added and as the loss rate increases. For example,

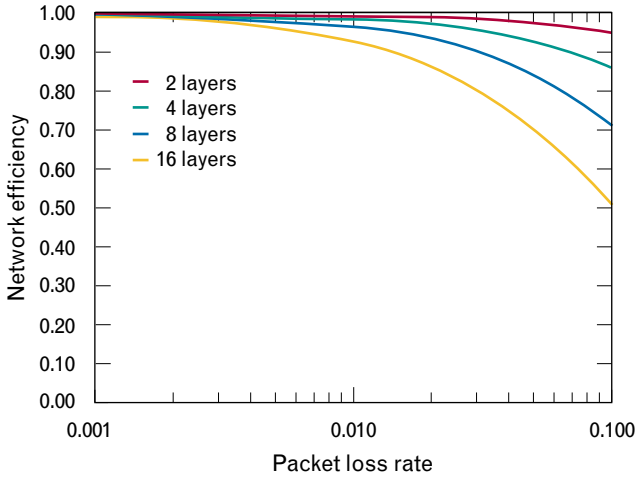

FIGURE 11. Use of hierarchical layers causes network efficiency to decrease as packet loss rate increases. The inefficiency is due to the fact that packets for higher layers are useless if packets for lower layers are lost.

with a 10% packet loss rate (p = 0.1) and 8 hierarchical layers (N = 8), the network efficiency is approximately 70%. This efficiency implies that approximately 30% of packets received are useless because one or more packets from lower layers have been lost. From this we conclude that independent rather than hierarchical layers should be used in error-prone networks.

Security Implications

Each of the adaptation techniques described has different performance characteristics. They also have different security characteristics. Thus a total solution involves trade-offs between performance and security. Transcoding has the disadvantage that it does not preserve end-to-end security. In order to translate media streams, transcoders must have access to encryption keys. Security is no longer provided on an end-to-end basis because data are available in the clear at a midpoint in the network. On the other hand, both simulcast and layered encoding preserve end-to-end security because each media stream can be separately encrypted on an end-to-end basis [25].

Layered coding has the additional advantage that it may not be necessary to encrypt every media layer. If the layers are hierarchical, then encrypting only the lowest layer may provide adequate security [28, 29]. Information in higher layers may be useless without first decrypting lower layers. The bandwidth overhead of incompressible security protocol headers and the computational overload of encrypting and decrypting data can thus be reduced. However, this reduction must be balanced by the fact that packet losses for the lowest layer contribute to the wasted bandwidth of delivering useless packets for the higher layers.

This comparison of security and performance considerations for the various adaptation techniques leads to an unusually advantageous solution. Rather than having to make a trade-off between security and performance, the user can, by using layered coding, obtain simultaneously both the best network performance and the most secure operation for adaptive multicast transmission of multimedia data.

Reliable Multicast Transmission

Some group-communication applications, such as interactive audio and video conferencing, can tolerate some data loss, but other applications cannot. Shared text and drawing applications used in a multimedia conferencing session require data to be delivered reliably to all group members, which can be achieved with a reliable multicast protocol. Other applications, such as distributed interactive simulation [30] or replicated file servers, can benefit from reliable multicast transmission, but may have very different reliability requirements. Careful design of reliability protocols is especially important in wireless networks with their higher error rates and lower bandwidth.

The Transmission Control Protocol (TCP) [31] meets the general requirements for reliable, sequential delivery of packets for unicast transmission. No such general-purpose protocol exists for multicast transmission. Because group-communication applications have widely varying reliability requirements, many different reliable multicast protocols have been developed; none have emerged as a dominant standard, as TCP is for reliable unicast. Most reliable multicast protocols are optimized for performance and are robust to common faults, such as lost packets or failures suffered by one or more group participants. Few protocols are designed to withstand intentional malicious attacks. As experimental protocols become candidates

for standardization, their vulnerabilities to such attacks will be an important evaluation criterion [32]. We now discuss characteristics of several reliable multicast protocols, their vulnerabilities to denial-of-service attacks, and protective measures that can be taken.

Sender-Initiated Reliability Protocols

A sender-initiated reliability protocol places the burden of loss detection on the sender. A positive acknowledgment (ACK) is required from every receiver for every packet sent. A lost packet is detected when an ACK fails to arrive from one or more receivers. When a loss is detected, the packet is retransmitted and the sender again waits for an ACK from every receiver.

Such sender-initiated protocols suffer from ACK implosion [33, 34], in which a flood of acknowledgments arrives in response to each packet sent, as shown in Figure 12. Increasing amounts of network bandwidth and processing time are consumed as the number of receivers in the group increases, which limits the group size that such protocols can accommodate. Denial-of-service attacks that produce similar packet implosions have been reported [6].

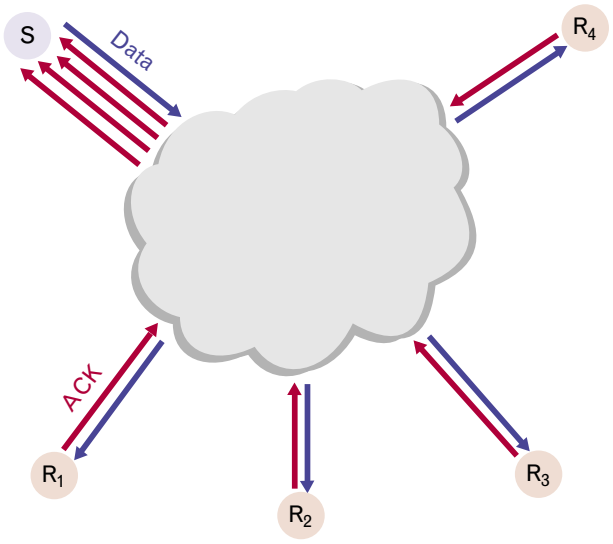

FIGURE 12. Acknowledgment (ACK) implosion associated with sender-initiated reliability protocols. ACK implosion occurs when many receivers simultaneously send unicast acknowledgments to the sender.

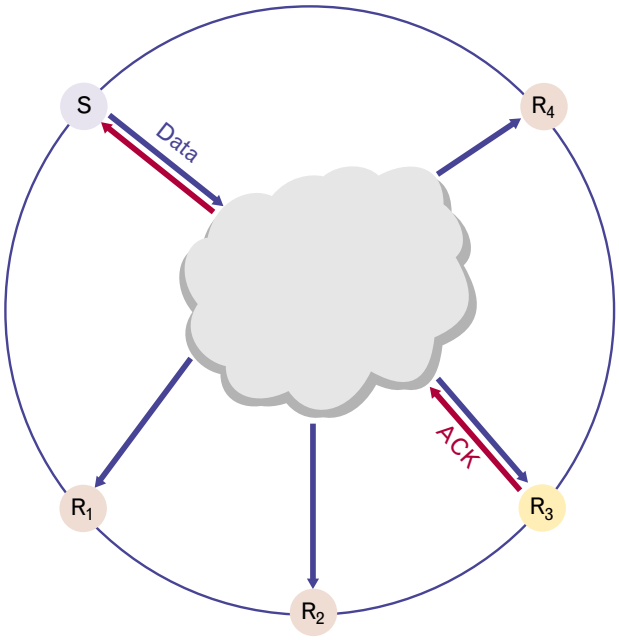

FIGURE 13. Members of a multicast group organized into a logical ring. This arrangement reduces the number of acknowledgments that must be processed by any one group member. In a ring, a token is passed from one member to the next, and only the current token holder, R_3 in this example, sends acknowledgments.

Some protocols avoid ACK implosion by organizing the group into a logical tree [35, 36] or ring [37, 38], as shown in Figure 13. This arrangement reduces the number of acknowledgments that must be processed by any one group member. In a tree, each group member sends an acknowledgment to its parent after receiving acknowledgments from each of its children. In a ring, a token is passed from one member to the next and only the current token holder sends an acknowledgment. The virtual links connecting the nodes of the tree or ring need not correspond to direct network links between group participants. Both tree and ring protocols require that the group membership be known. Thus there must also be a protocol for reliably distributing the membership list to all group participants.

The Reliable Multicast Protocol (RMP) [38] is a sender-initiated protocol in which group members are organized into a ring. This ring serves several purposes. Acknowledgment implosion is eliminated by allowing only the current token holder to acknowledge packets. A global ordering of packets from dif-

ferent sources is determined by the order in which they arrive at the token holder. This order is indicated by a sequence number in each acknowledgment. A protocol is defined to be stable if the sender receives acknowledgment that all packets have been delivered to all receivers. Stability is provided in RMP by the circulation of the token around the ring, because a member will not accept the token from its neighbor until it has received all acknowledged packets.

Receiver-Initiated Reliability Protocols

A receiver-initiated reliability protocol places the burden of loss detection on the individual receivers. Receivers generate a negative acknowledgment (NAK) when they detect a lost packet. The packet is retransmitted in response to one or more NAKs. NAK implosion is reduced because a NAK is sent only when a loss is detected, rather than for every packet. NAK implosion can still be a problem if a large number of receivers lose the same packet. Suppression mechanisms can minimize the number of duplicate NAKs produced when such correlated losses occur. Similar suppression mechanisms can prevent a flood of retransmissions when any member with the appropriate data is allowed to respond to a NAK [39, 40].

The Scalable Reliable Multicast (SRM) [39] protocol is an example of a receiver-initiated protocol. Receivers send NAKs to indicate that a packet has been lost. Periodic status messages from every receiver, which announce the highest sequence number received from each source, serve as a form of positive acknowledgment. In order to suppress excess NAKs in response to a correlated loss, receivers each send their NAK to the entire multicast group, as shown in Figure 14. However, if a receiver first hears another NAK, then it will suppress its own NAK. Because the NAK is sent to the entire group, any participant with a copy of the appropriate data can respond with a retransmission. In order to suppress duplicate retransmissions, participants responding to a NAK send their retransmission to the entire group, and cancel their own retransmission if they hear another one.

Security Implications

It may be tempting simply to encrypt data within an application before transmission with a reliable multicast protocol. When only the payload within each packet is encrypted, header compression is more effective. This improved efficiency can be important in bandwidth-constrained wireless networks. This approach may be adequate if confidentiality is the only security requirement, but it leaves open the possibility of denial-of-service attacks. An adversary could interfere with the reliable multicast protocol in subtle ways to prevent correct delivery of packets.

The authors have developed several experimental denial-of-service attacks that exploit the reliability mechanisms of reliable multicast protocols. We exploit the sequence numbers of acknowledgments in RMP to disrupt the ordering of packets. We exploit the NAK suppression mechanism in the SRM protocol to prevent retransmissions of lost packets. Similar attacks could be devised against other reliable multicast protocols that use similar reliability mechanisms.

In RMP, the current token holder in the ring establishes the ordering of packets by including sequence numbers in the acknowledgments for these packets. An attacker who observes these acknowledgments can

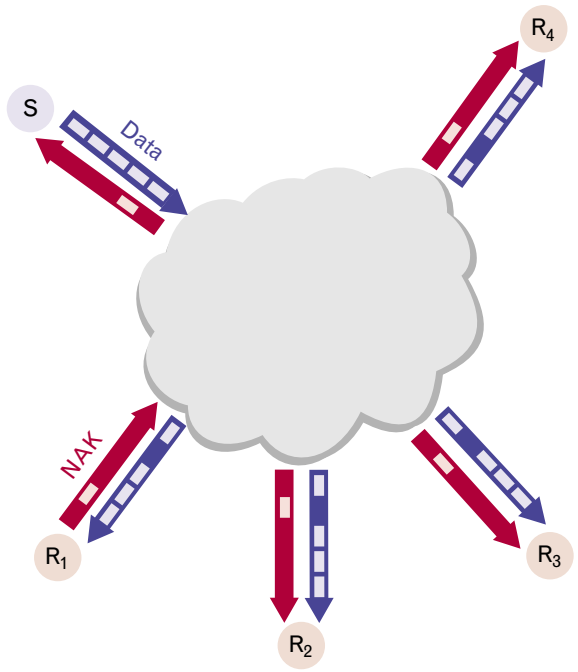

FIGURE 14. Avoiding negative acknowledgment (NAK) implosion with receiver-initiated reliability protocols. Receivers can send multicast negative acknowledgments (NAKs) to the entire group to suppress duplicate NAKs and avoid NAK implosion.

anticipate the sequence numbers that will be assigned to packets in the near future. By generating an acknowledgment for a packet that has not yet been transmitted, the attacker can cause an incorrect sequence number to be associated with that packet. Other members of the group will process the packet in the incorrect order. The acknowledgment with the correct sequence number will be discarded as a duplicate because it arrives after the acknowledgment sent by the attacker.

In the SRM protocol, NAKs serve two purposes. Their primary purpose is to trigger the retransmission of a missing packet. Their secondary purpose is to suppress duplicate NAKs from other receivers. This suppression is accomplished by sending each NAK to the entire group. When it reaches the sender, the NAK will trigger a retransmission. When it reaches another receiver, the NAK will suppress that receiver's own NAK.

An attacker can use one of several methods to send a bogus NAK to a subset of the group. This action suppresses NAKs from the members that receive the bogus NAK without triggering a retransmission from the other members of the group. Similarly, an attacker can generate a retransmission that is delivered to a subset of the group. This action will cancel legitimate retransmissions from other group members without reaching the members that requested the retransmission. These attacks interfere with retransmissions, which means that they are effective only when losses occur. Thus these attacks are more effective in wireless networks, which are more prone to losses, or when used in combination with brute-force attacks, such as network flooding, which can induce losses through congestion.

These denial-of-service attacks can be prevented by encrypting the sequence numbers in the headers of all packets, including acknowledgments, to prevent an adversary from observing them. A simple authentication protocol [41] that uses a shared secret key could also be used to prevent an adversary from constructing a packet that would be accepted by legitimate group members. If a member of the group is compromised, in effect becoming an adversary, then it would be necessary to distribute a new key or resort to more costly public-key authentication protocols.

A few secure reliable multicast protocols exist [42, 43]. These protocols can survive attacks even if up to one-third of the group participants have been compromised. They include mechanisms whereby honest group participants can detect other group participants that exhibit malicious behavior or otherwise fail to properly execute the reliable multicast protocol. However, these protocols rely on digital signatures to authenticate critical control packets. Because of the complexity of public-key algorithms, about ten digital signatures can be generated per second, and about one hundred signatures can be verified per second on today's computers. This cryptographic processing overhead limits the performance and scalability of such protocols, but the performance penalty may be acceptable in high-security applications.

Dynamically Deployed Protocols

For a multimedia conferencing session to succeed, all participants must be capable of decoding the audio and video formats being used. Because different participants may choose to transmit media in formats best suited to each of their local computing and communication capabilities, several incompatible media formats may be in use simultaneously. One approach to interoperability is to include built-in support for many different network protocols and media formats. For example, the vic video-conferencing application, available in the MASH Multimedia Toolkit [44], includes three different network protocols, and coders and decoders for six different video formats.

As new protocols and media formats become available over time, applications must grow to include additional coders and decoders. Those wishing to participate in multimedia conferencing must update their software as new versions become available. This approach to interoperability represents a barrier to the adoption of new protocols and media formats. There may be a long delay from the time a new protocol or format is introduced to the time when a sufficient number of participants have updated software that is capable of taking advantage of it. Until there is a sufficiently large audience, a new protocol or format is of limited use.

Dynamically deployed protocols, implemented by using mobile code, ease the adoption of new network

protocols and media formats. With mobile code, media streams become independent objects with methods for encoding and decoding their own data. It is no longer necessary to have all possible decoders built into the application in anticipation of specific media formats. Rather, the executable implementation of the decoder is provided as part of the data stream using mobile code. In a sense, an arriving media stream will know how to decode itself. This concept can be taken a step further with objects that also know how to transport themselves across the network. An object representing an audio or video stream may use RTP [11], while an object representing a shared text document may use SRM [39]. Objects requiring security could carry an executable implementation of their own security protocol, such as Secure Socket Layer (SSL) or Transport Layer Security [8]. Such a mobile-code framework will make it possible to rapidly develop and deploy new protocols.

With the ability to dynamically deploy new protocols using mobile code, it would be possible to reconfigure a communication session in reaction to an attack [45]. We have discussed several attacks that target specific mechanisms in specific protocols. If it becomes possible to switch to a different protocol, then the communication session as a whole can survive such attacks. Rather than simply reacting to an attack after it occurs, such protocol switching could be used proactively, similar to the concept of frequency hopping in spread-spectrum radio transmission in which frequencies are changed constantly to thwart attempts to intercept or interfere with communication. If an undetected attack disrupts a session, communication will improve at the next protocol switch. This kind of active protocol switching can also help detect attacks. If the cause of a disruption is a benign one, such as network congestion, its effects should be apparent with all protocols in use. A malicious disruption that targets a specific protocol would soon be noticed.

Example Scenario of Wireless System Attacks and Countermeasures

We now illustrate how some vulnerabilities of mobile wireless information systems might be exploited in a fictional scenario, and how some of the security and survivability techniques described in this article might be used to maintain network operations during attacks.

Consider the following peacekeeping mission: headquarters has been set up in a command center with high-speed wired networks. Remote patrols include vehicles with reasonably high-powered radios, and individuals on foot carry small battery-powered computers and low-powered radios that provide low-speed network connections. The terrain may be urban, where communication can sometimes be blocked by a variety of structures. There are enemy forces in the area, some operating covertly. The enemy forces may be small in number but technically sophisticated.

A remote patrol maintains contact with commanders back at headquarters via an ongoing multimedia conferencing session. The patrol is sending reconnaissance video back to the commanders. All are participating in interactive audio and video, and a shared drawing tool is being used to exchange maps and annotate them.

A layered encoding is used for the reconnaissance video and conferencing session. This encoding makes it possible to accommodate the heterogeneous mixture of high-speed wired links and low-speed wireless links as well as the variability of wireless links. The lowest video layer is encrypted to provide end-to-end security. Higher layers need not be encrypted if a hierarchical layering is used, because those higher layers are useless without the lowest layer. This strategy conserves network bandwidth by making header compression more effective. A reliable multicast protocol is used to deliver map images and drawing commands for the shared drawing tool. Because wireless links are subject to relatively high error rates, it is necessary to retransmit any lost packets.

As the patrol drives through city streets, the vehicle's radio channel suffers from fading and other impairments. In response to the increased network errors, the higher video layers are no longer transmitted between the patrol and headquarters while the vehicle is in motion. When the vehicle comes to a stop at the destination, network conditions improve and additional layers are transmitted again.

Several members of the patrol leave the vehicle, but

remain connected to the network. The radios they carry on their backs connect them to their vehicle, which relays packets between them and headquarters. Because of the reduced network bandwidth with the portable radios, the patrol members on foot choose not to transmit or receive video. They continue to participate in the audio conferencing session and continue to update and annotate maps by using the shared drawing tool. The transmission of reconnaissance video and the interactive video conferencing session continue among the commanders at headquarters and the patrol members remaining with the vehicle.

One member of the patrol is separated from the others and is captured by the enemy. A month earlier, the enemy obtained a computer identical to the one carried by the captured patrol member. Since that time, they have reverse-engineered all the software on the computer, discovered a weakness in the reliable multicast protocol used for the shared drawing tool, and developed a subtle denial-of-service attack that exploits that weakness. Using the captured computer, which contains all the cryptographic keys currently in use for the multimedia conferencing session, the enemy launches their denial-of-service attack. This attack interferes with acknowledgments and retransmissions to prevent the correction of network errors due to lost packets.

The commanders at headquarters and the other patrol members are not aware that one member has been captured and do not immediately realize that an attack is in progress. Eventually the commanders notice a problem and use mobile code to dynamically deploy a different reliable multicast protocol, one for which the enemy does not have an attack prepared. At this point, communication improves dramatically, indicating that the earlier problems were likely due to a malicious attack rather than more benign communication difficulties.

Conclusion

We have shown a number of ways in which mobile wireless networks complicate security and survivability by creating additional vulnerabilities and making defenses more difficult. Our focus has been on group communication, particularly multimedia conferencing, which is a stressing application for mobile wireless networks. Multicast transmission is essential for efficient group communication in wireless networks, but it introduces additional security concerns.

There are several techniques to adapt the multicast delivery of audio and video to different receivers. Layered multicast gives good efficiency and preserves end-to-end security. However, there is a trade-off between hierarchical layers and independent layers. Using hierarchical layers can yield savings when only the lowest layer is encrypted, making header compression more effective for the higher layers. However, using independent layers may be more efficient in the presence of errors because all packets that are delivered are useful. Further work is required to determine optimal combinations of media coders and network protocols for mobile wireless networks.

Reliable multicast protocols are vulnerable to sophisticated denial-of-service attacks unless critical control information in each packet is protected through encryption or authentication. Security protocols that use a secret key shared by all members of a group can prevent attacks from those outside the group. Preventing attacks from inside the group is more difficult, often requiring the use of complex public-key encryption algorithms. Further work is required to find an appropriate balance between security and efficiency for mobile wireless networks.

Because mobile networks are so dynamic, a network protocol that is appropriate one instant may no longer be appropriate the next instant. Dynamically deployed protocols offer a solution to this problem. An object-oriented communication framework in which objects contain not only data but also executable code makes it possible to react to changing network conditions.

We believe that techniques such as layered coding, secure reliable multicast, and dynamically deployed protocols offer promise, but that much research remains to be done. Reliance on wireless networks will continue to increase, making such networks more tempting targets for attack.

Acknowledgments

The authors would like to acknowledge the valuable contributions of David Kassay of the Information

Systems Technology group for implementing and testing some of the ideas described in this paper. The authors would also like to acknowledge the contributions of MIT graduate research assistants Chad Jones, Subbu Sthanu, and Lara Karbiner, whose thesis projects in the Information Systems Technology group made valuable contributions to the work described here. This work was sponsored by the Defense Advanced Research Projects Agency.

REFERENCES

1. R.T. Morris, "A Weakness in the 4.2BSD Unix TCP/IP Software," Technical Report 117, AT&T Bell Laboratories, Murray Hill, N.J., 1985, <ftp://ftp.research.att.com/dist/internet_security/117.ps.Z>.
2. "IP Spoofing Attacks and Hijacked Terminal Connections," CERT Advisory CA-95:01, CERT Coordination Center, 3 Jan. 1995, rev. 23 Sept. 1997, <ftp://ftp.cert.org/pub/cert_advisories/CA-95%3A01.IP.spoofing>.
3. S.M. Bellovin, "Security Problems in the TCP/IP Protocol Suite," *Comput. Commun. Rev.* **19** (2), 1989, pp. 32–48, <ftp://ftp.research.att.com/dist/internet security/ipext.ps.Z>.
4. B. Guha and B. Mukherjee, "Network Security via Reverse Engineering of TCP Code: Vulnerability Analysis and Proposed Solutions," *IEEE Network* **11** (4), 1997, pp. 40–48.
5. "TCP SYN Flooding and IP Spoofing Attacks," CERT Advisory CA-96:21, CERT Coordination Center, 19 Sept. 1996, rev. 24 Aug. 1998, <ftp://ftp.cert.org/pub/cert_advisories/CA-96.21.tcp_syn_flooding>.
6. "'Smurf' IP Denial-of-Service Attacks," CERT Advisory CA-98:01, CERT Coordination Center, 5 Jan. 1998, rev. 24 Aug. 1998, <ftp://ftp.cert.org/pub/cert_advisories/CA-98.01.smurf>.
7. W.R. Cheswick and S.M. Bellovin, *Firewalls and Internet Security: Repelling the Wily Hacker* (Addison-Wesley, Reading, Mass., 1994).
8. T. Dierks and C. Allen, "The TLS Protocol Version 1.0," Request for Comments 2246, IETF Network Working Group, Jan. 1999, <ftp://ietf.org/rfc/rfc2246.txt>.
9. S. Kent and R. Atkinson, "Security Architecture for the Internet Protocol," Request for Comments 2401, IETF Network Working Group, Nov. 1998, <ftp://ietf.org/rfc/rfc2401.txt>.
10. S. Kent and R. Atkinson, "IP Encapsulating Security Payload," Request for Comments 2406, IETF Network Working Group, Nov. 1998, <ftp://ietf.org/rfc/rfc2406.txt>.
11. H. Schulzrinne, S.L. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," Request for Comments 1889, IETF Audio-Video Transport Working Group, Jan. 1996, <ftp://ietf.org/rfc/rfc1889.txt>.
12. M. Degermark, B. Nordgren, and S. Pink, "IP Header Compression," Request for Comments 2507, IETF Network Working Group, Feb. 1999, <ftp://ietf.org/rfc/rfc2507.txt>.
13. S.L. Casner and V. Jacobson, "Compressing IP/UDP/RTP Headers for Low-Speed Serial Links," Request for Comments 2508, IETF Network Working Group, Feb. 1999, <ftp://ietf.org/rfc/rfc2508.txt>.
14. T. Dorcey, "CU-SeeMe Desktop Video Conferencing Software," Connexions **9** (3), 1995, <http://www.cu-seeme.net/squeek/tech/DorceyConnexions.html>.
15. T. Ballardie and J. Cowcroft, "Multicast-Specific Security Threats and Counter-Measures," *Proc. Symp. on Network and Distributed Systems Security, San Diego, 16–17 Feb. 1995,* pp. 2–16, <ftp://cs.ucl.ac.uk/darpa/IDMR/mcast-sec isoc.ps.Z>.
16. B. Cain, S. Deering, and A. Thyagarajan, "Internet Group Management Protocol, Version 3," Internet-Draft, IETF InterDomain Multicast Routing Working Group, June 2000, <ftp://ftp.ietf.org/internet-drafts/draft-ietf-idmr-igmp-v3-04.txt>.
17. D. Waitzman, C. Partridge, and S. Deering, "Distance Vector Multicasting Routing Protocol," Request for Comments 1075, IETF Network Working Group, Nov. 1988, <ftp://ietf.org/rfc/rfc1075.txt>.
18. H. Kanakia, P.P. Mishra, and A.R. Reibman, "An Adaptive Congestion Control Scheme for Real-Time Packet Video Transport," *IEEE/ACM Trans. Networking* **3** (6), 1995, pp. 671–682.
19. E. Amir, S. McCanne, and R. Katz, "Receiver-Driven Bandwidth Adaptation for Light-Weight Sessions," *ACM Multimedia Conf., Seattle, Wash., 5–9 Nov. 1997,* pp. 3248, <http://www.cs.berkeley.edu/~elan/pubs/papers/scuba-acm-mm97.ps>.
20. J.-C. Bolot and T. Turletti, "Experience with Control Mechanisms for Packet Video in the Internet," *Comput. Commun. Rev.* **28** (1), 1998, pp. 4–14, <http://www.acm.org/sigcomm/ccr/archive/1998/jan98/ccr-9801-bolot.html>.
21. T.M. Parks, D.A. Kassay, and C.J. Weinstein, "Security Implications of Adaptive Multimedia Distribution," *IEEE INFOCOM: Conf. on Communications* **3**, *Vancouver, B.C., Canada, 6–10 June 1999,* pp. 1563–1567, <http://www.ll.mit.edu/IST/pubs/icc99-parks>.
22. E. Amir, S. McCanne, and H. Zhang, "An Application Level Video Gateway," *ACM Multimedia Conf., San Francisco, 9–13, Nov. 1995,* pp. 255–265, <http://www.cs.berkeley.edu/~elan/pubs/papers/vgw.ps>.
23. X. Li and M.H. Ammar, "Bandwidth Control for Replicated-Stream Multicast Video Distribution," *IEEE Int. Symp. on High Performance Distributed Computing, Syracuse, N.Y., 6–9 Aug. 1996,* pp. 356–363, <http://www.cs.gatech.edu/~lixue/bw.ps.gz>.
24. N. Shacham, "Multipoint Communication by Hierarchically Encoded Data," *IEEE INFOCOM: Conf. on Computer Communications* **3**, *Florence, Italy, 4–8 May 1992,* pp. 9A.4.1–9A.4.8.
25. P. Haskell and D.G. Messerschmitt, "Some Research Issues in a Heterogeneous Terminal and Transport Environment for Multimedia Services," *Workshop on Adaptive Systems, Intelligent Approaches, Massively Parallel Computing and Emergent Techniques in Signal Processing and Communications, Bayona, Spain, Oct. 1994*, <http://www.eecs.berkeley.edu/~messer/PAPERS/94/Spain.pdf>.
26. S. McCanne, V. Jacobson, and M. Vetterli, "Receiver-Driven Layered Multicast," *Proc. 1996 ACM Conf. on Applications, Technologies, Architectures, and Protocols for Computer Communications, Stanford, Calif., 26–30 Aug. 1996,* pp. 117–130, <ftp://ftp.ee.lbl.gov/papers/mccanne-sigcomm96.ps.gz>.
27. L. Wu, R. Sharma, and B. Smith, "Thin Streams: An Architecture for Multicasting Layered Video," *Int. Workshop on Network and Operating Systems Support for Digital Audio and Video, St. Louis, Mo., 19–21 May 1997,* pp. 173–182, <http://www.cs.cornell.edu/zeno/papers/ThinStreams/ts.pdf>.

28. I. Agi and L. Gong, "An Empirical Study of Secure MPEG Video Transmissions," *Proc. Symp. Network and Distributed System Security*, *San Diego, 22–23 Feb. 1996,* pp. 137–144, <http://java.sun.com/people/gong/papers/mpegsecurity.html>.
29. C. Shi and B. Bhargava, "A Fast MPEG Video Encryption Algorithm," *Proc. 6th ACM Multimedia Conf.*, *Bristol, U.K., 12–16 Sept. 1998,* pp. 81–88, <http://www.acm.org/sigmm/MM98/electronic_proceedings/shi>.
30. *Proc. IEEE* **83** (8), 1995 (Special Issue on Distributed Interactive Simulation, J.S. Dahmann and D.C. Wood, eds.).
31. J. Postel, "Transmission Control Protocol," Request for Comments 793, Internet Engineering Task Force, Sept. 1981, <ftp://ietf.org/rfc/rfc0793.txt>.
32. A. Mankin, A. Romanow, S. Bradner, and V. Paxson, "IETF Criteria for Evaluating Reliable Multicast Transport and Application Protocols," Request for Comments 2357, IETF Network Working Group, June 1998, <ftp://ietf.org/rfc/rfc2357.txt>.
33. S. Pingali, D. Towsley, and J.F. Kurose, "A Comparison of Sender-Initiated and Receiver-Initiated Reliable Multicast Protocols," *SIGMETRICS Conf. on Measurement and Modeling of Computer Systems*, *Nashville, Tenn., 16–20 May 1994,* pp. 221–230, <ftp://gaia.cs.umass.edu/pub/Ping94:Multicast.ps.Z>.
34. B.N. Levine and J.J. Garcia-Luna-Aceves, "A Comparison of Reliable Multicast Protocols," *ACM Multimedia Systems J.* **6** (5), 1998, pp. 334–338, <http://www.cse.ucsc.edu/research/ccrg/publications/brian.mmsj.ps.gz>.
35. B.N. Levine, D.B. Lavo, and J.J. Garcia-Luna-Aceves, "The Case for Reliable Concurrent Multicasting Using Shared Ack Trees," *ACM Multimedia Conf., Boston, 18–22 Nov. 1996,* pp. 365–376, <http://www.cse.ucsc.edu/research/ccrg/publications/brian.mm96.pdf>.
36. S. Paul, K.K. Sabnani, J.C. Lin, and S. Bhattacharyya, "Reliable Multicast Transport Protocol (RMTP)," *IEEE J. Sel. Areas Commun.* **15** (3), 1997, pp. 407–421, <http://www.bell-labs.com/user/sanjoy/rmtp2.ps>.
37. J.-M. Chang and N.F. Maxemchuk, "Reliable Broadcast Protocols," *ACM Trans. Computer Syst.* **2** (3), 1984, pp. 251–273.
38. B. Whetten, T. Montgomery, and S. Kaplan, "A High Performance Totally Ordered Multicast Protocol," Technical Report TR-94-069, International Computer Science Institute, Berkeley, Calif., Dec. 1994, <ftp://ftp.icsi.berkeley.edu/pub/techreports/1994/tr-94-069.ps.gz>.
39. S. Floyd, V. Jacobson, C.-G. Liu, S. McCanne, and L. Zhang, "A Reliable Multicast Framework for Light-Weight Sessions and Application Level Framing," *IEEE/ACM Trans. Networking* **5** (6), 1997, pp. 784–803, <http://www-nrg.ee.lbl.gov/floyd/srm-paper.html>.
40. B. Sabata, M.J. Brown, B.A. Denny, and C.-H. Heo, "Transport Protocol for Reliable Multicast: TRM," *Proc. IASTED Int. Conf. on Networks*, *Orlando, Fla., 8–10 Jan. 1996,* pp. 143–145, <http://www.erg.sri.com/people/sabata/paper/isted.ps>.
41. H. Krawczyk, M. Bellare, and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication," Request for Comments 2104, IETF Network Working Group, Feb. 1997, <ftp://ietf.org/rfc/rfc2104.txt>.
42. M.K. Reiter, "Secure Agreement Protocols: Reliable and Atomic Group Multicast in Rampart," *Proc. 2nd ACM Conf. on Computer and Communications Security*, *Fairfax, Va., 2–4 Nov. 1994,* pp. 68–80, <http://www.research.att.com/~reiter/papers/ccs2.ps.gz>.
43. K.P. Kihlstrom, L.E. Moser, and P.M. Melliar-Smith, "The SecureRing Protocols for Securing Group Communication," *Thirty-First Hawaii Int. Conf. on System Sciences* **3**, *Kona, Hawaii, Jan. 1998,* pp. 317–326, <http://alpha.ece.ucsb.edu/ftp/SMP/hicss98.ps.gz>.
44. S. McCanne, E. Brewer, R. Katz, L. Rowe, E. Amir, Y. Chawathe, A. Coopersmith, K. Mayer-Patel, S. Raman, A. Schuett, D. Simpson, A. Swan, T.-L. Tung, D. Wu, and B. Smith, "Toward a Common Infrastructure for Multimedia-Networking Middleware," *Proc. IEEE 7th Int. Workshop on Network and Operating Systems Support for Digital Audio and Video, St. Louis, Mo., 19–21 May 1997,* pp. 39–49, <http://www-mash.cs.berkeley.edu/dist/mash/papers/mash-nossdav97.ps.gz>.
45. S.R. Sthanu, T.M. Parks, and S.R. Lerman, "Survivability through Dynamic Reconfiguration," *2nd Federated Laboratory Symp., College Park, Md., 2–6 Feb. 1998,* pp. 65–69, <http://www.ll.mit.edu/IST/pubs/atirp98-sthanu>.

THOMAS M. PARKS
is a former staff member in the Information Systems Technology group. He is now a professor in the Computer Science Department at Colgate University in Hamilton, New York. He earned B.S.E. and Ph.D. degrees in electrical engineering and computer science from Princeton University and the University of California, Berkeley, respectively. After receiving his undergraduate degree in 1987, he worked in the Speech Systems Technology group at Lincoln Laboratory. There he designed and implemented multiprocessor architectures for real-time signal processing, and he contributed to research projects in low-rate speech coding and continuous speech recognition. In graduate school, he studied the semantics of delay-free loops in timed discrete-event simulations, real-time scheduling theory and its applications to multirate signal processing systems described by dataflow programs, formal properties of dataflow languages, and the decidability of certain dataflow scheduling problems. After receiving his doctorate in 1995, he rejoined Lincoln Laboratory in the Information Systems Technology group, where his research involved security and survivability of mobile information systems.

CLIFFORD J. WEINSTEIN
leads the Information Systems Technology group and is responsible for initiating and managing research programs in speech technology, machine translation, and information assurance. Cliff joined Lincoln Laboratory as an MIT graduate student in 1967, and became leader of the Speech Systems Technology group (now Information Systems Technology group) in 1979. From 1986 to 1998, he was U.S. technical specialist on the NATO RSG10 (now IST-01) Speech Research Group, authoring a comprehensive NATO report and journal article on opportunities for applications of advanced speech technology in military systems. From 1989 to 1994, he chaired the coordinating committee for the DARPA Spoken Language Systems Program, the major U.S. research program in speech recognition and understanding. In 1999, he was appointed to the Defense Advanced Research Projects Agency (DARPA) Information Sciences and Technology (ISAT) Panel, a group that provides DARPA with assessments of the state of advanced information science and technology, and its relationship to Department of Defense issues. In 1993, Cliff was elected as a Fellow of the IEEE for technical leadership in speech recognition, packet speech, and integrated voice/data networks. He received S.B., S.M., and Ph.D. degrees in electrical engineering from MIT.