MRTP: A Multiflow Real-Time Transport Protocol for Ad Hoc Networks

Shiwen Mao, *Member, IEEE*, Dennis Bushmitch, *Member, IEEE*, Sathya Narayanan, *Member, IEEE*, and Shivendra S. Panwar, *Senior Member, IEEE*

***Abstract*—Real-time multimedia transport has stringent quality of service requirements, which are generally not supported by current network architectures. In emerging mobile ad hoc networks, frequent topology changes and link failures cause severe packet losses, which degrade the quality of received media. However, in such mesh networks, there usually exist multiple paths between any source and destination nodes. Such path diversity has been demonstrated to be effective in combating congestion and link failures for improved media quality. In this paper, we present a new protocol to facilitate multipath transport of real-time multimedia data. The proposed protocol, the *multiflow real-time transport protocol* (MRTP), provides a convenient vehicle for real-time applications to partition and transmit data using multiple flows. We demonstrate through analysis that data partitioning, which is an essential function of MRTP, can effectively reduce the short-range dependence of multimedia data, thus improving its queueing performance in underlying networks. Furthermore, we show that a few flows are sufficient for MRTP to exploit most of the benefits of multipath transport. Finally, we present a comprehensive simulation study on the performance of MRTP under a mobile ad hoc network. We show that with one additional path, MRTP outperformed single-flow RTP by a significant margin.**

***Index Terms*—Ad hoc networks, multipath transport, real-time transport protocol, traffic partitioning, video communications.**

I. Introduction

AD HOC networks are wireless mobile networks without an infrastructure. It is very challenging to provide multimedia service (e.g., video communications) in such networks. We find that what makes video transport over the Internet and certain wireless networks successful is the existence of a relatively reliable path from the source to the receiver, such that packet losses and delays are within a predictable range, which can be effectively dealt with by applying appropriate error control and error concealment techniques [1]. However, such an assumption hardly holds true in mobile ad hoc networks, within which a path usually consists of multiple wireless links that are highly fragile due to fading, interference, and node mobility. In addition, due to contention in the MAC layer, the capacity of a wireless link is further constrained by neighboring transmissions. For example, using the IEEE 802.11 MAC, the maximum throughput of a chain topology may be only a seventh of the channel bandwidth [2]. Such limited capacity makes congestion more frequent and persistent in ad hoc networks.

In our previous work [3]–[5], we studied the problem of image and video communications in mobile ad hoc networks. We found that using multiple paths concurrently for a multimedia session, termed *multipath transport* throughout this paper, is highly effective in combating fragile paths and frequent congestion in such networks. With multipath transport, as long as link/node failure events on different paths are not entirely correlated, the receiver can always receive some data and apply appropriate error control/concealment techniques to reduce the damage caused by lost packets. Although multipath transport has been previously applied in wireline networks for load balancing and bandwidth aggregation [6], we feel that it has more potential in mobile ad hoc networks. The mesh topology of such networks is highly amenable to multipath routing [7]–[9].

In this paper, we present a new protocol, named *multiflow real-time transport protocol* (MRTP), to facilitate multipath multimedia transport. Considering the substantial ongoing research on using multipath transport for real-time multimedia streaming (see Section II-A), such a protocol is both timely and of importance. MRTP is a transport protocol implemented in the application layer. Given multiple paths maintained by a multipath routing protocol, MRTP and its companion control protocol, the *multiflow real-time transport control protocol* (MRTCP), provide essential support for real-time multimedia applications, including session and flow management, data partitioning and reassembly, traffic dispersion, data framing, timing, sequence numbering, and quality-of-servie (QoS) feedback. This enables application developers to fully explore the potential of multipath transport. We provide a detailed description of MRTP, including a formal definition, design considerations and protocol operations, as well as three usage scenarios that are typical in multimedia applications using multipath transport.

Second, we present two performance studies of the proposed protocol. In the first performance study, we examine the impact of traffic partitioning, which is an essential function module of MRTP, on the queueing performance of multimedia data. We

Manuscript received February 8, 2004; revised April 22, 2005. This work was supported by the National Science Foundation under Grants ANI-0081375 and CNS-0435228, and the New York State Center for Advanced Technology in Telecommunications (CATT) at Polytechnic University, Brooklyn, NY. Part of this work was presented at the IEEE 2003-Fall Vehicular Technology Conference, Orlando, FL, October 2003. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Madjid Merabti.

S. Mao is with the Department of Electrical and Computer Engineering, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061 USA (e-mail: smao@ieee.org).

D. Bushmitch and S. Narayanan are with the Panasonic Digital Networking Laboratory (PDNL), Panasonic Technologies, Inc. (PTI), Princeton, NJ 08540 USA (e-mail: db@research.panasonic.com; sathya@research.panasonic.com).

S. S. Panwar is with the Department of Electrical and Computer Engineering, Polytechnic University, Brooklyn, NY 11201 USA (e-mail: panwar@catt.poly.edu).

Digital Object Identifier 10.1109/TMM.2005.864347

analytically demonstrate that traffic partitioning can improve the queueing performance of multimedia data, resulting in less congestion, smaller delay, and higher utilization of bottleneck link bandwidth. This result is quite interesting by itself. The common belief is that self-similarity/long-range dependence, which has a dominant effect on the asymptotic behavior of the queueing system, is quite persistent. That is, it cannot be reduced by traffic shaping, buffering, or multiplexing [10]. However, we show that although long-range dependence cannot be reduced (i.e., the partitioned flows will have the same Hurst parameter H as the original flow), the level of short-term correlation can be effectively reduced by traffic partitioning. By applying the Bahadur–Rao Asymptotic [11], we are able to quantitatively demonstrate the improvement on queueing performance achieved by traffic partitioning. This result is highly relevant to MRTP, which is designed to partition and transmit real-time data over multiple flows, and to real-time multimedia traffic (e.g., video), which usually has tight decoding deadlines and has been shown to be long-range dependent [12].

Furthermore, we show analytically that most of the performance improvement can be achieved with a few paths (e.g., two or three paths), while only marginal improvement is gained by a further increase in the number of paths. Although similar trends has been observed in previous work where additional paths are only used for error recovery when the primary path fails [13], this has not been demonstrated analytically before for the case where all paths are used concurrently. This is highly relevant and important to MRTP. It shows, for example, if a node can establish a few routes in a mesh ad hoc network, or if an institutional network has two or more access routers, MRTP can be deployed to improve received media quality.

The second performance study is an OPNET simulation of MRTP for video streaming in a mobile ad hoc network. In this simulation study, lower layer details, including node mobility, multipath routing, and MAC layer dynamics are modeled and simulated. This study provides a realistic view of the impact of these factors on the MRTP performance. In the simulations, we used the multiple description motion compensation (MDMC) codec for video coding [29], which is among the latest advances in multiple description (MD) coding [36]. We find that although the paths used by MRTP may not always be independent (e.g., paths may share nodes or links, or interfere with each other, and there may not exist multiple source-destination paths in some topologies), MRTP can still effectively improve the received video quality over existing single-flow approaches.

Finally, it is worth noting that although MRTP has been developed in the context of mobile ad hoc networks, it can be applied in other types of networks for multipath multimedia transport, such as the Internet and infrastructure-based wireless networks. Many institutional networks nowadays have more than one access router to their service providers for fault tolerance purposes. As our analytical results show, even an increase from one access router to two will yield some improvement in queueing performance at the access routers, which are usually the bottleneck of an end-to-end multimedia session. See, for example, an experimental study of voice streaming over different ISP networks in [14]. Moreover, it is typical to have multiple access points in many infrastructure-based wireless networks (e.g., wireless LANs or "soft" hand-off of mobile nodes in cellular networks). A mobile node can use MRTP to access multiple access points in parallel for improved media quality (see [15] for an interesting study).

The rest of this paper is organized as follows. In Section II, we present the background and preliminaries. In Section III, we formally define MRTP/MRTCP and present their usage scenarios. The two performance studies of MRTP are presented in Section IV. Section V concludes the paper.

II. Background and Preliminaries

A. Multipath Transport of Multimedia Data

Recently, there has been considerable advances in multimedia coding and communications [16]–[18]. In particular, substantial research effort has been focused on using multiple paths for multimedia communications in the Internet [14], [19]–[28], infrastructure-based wireless networks [15], and mobile ad hoc networks [3]–[5], [7]–[9], [29] for unicast video transport [3]–[5], [8], [15], [19], [24], [26]–[28], [30], multicast video communications [7], [21], and multiple-server video streaming [9], [20], [22], [23].

The idea of dispersity routing was first introduced in [31]. Multipath transport has then been applied in various settings for data communications, including load balancing, bandwidth aggregation, and failure/error recovery [6], [32]–[35]. The recent interest in multipath transport for multimedia applications concentrates on exploring path diversity for robustness against transmission errors. It has been shown that traditional error control mechanisms, such as forward error correction (FEC) and automatic repeat request (ARQ), could be more efficient when combined with multipath transport [5], [30]. The recent advances in MD coding have made it highly suitable for multipath transport in various types of networks [5], [21], [23], [36]. In MD video, the correctly received streams from reliable paths provide information that enables improved recovery of a corrupted stream [4], [23], [29].

Although demonstrating the benefits of using multipath transport for real-time multimedia, to the best of our knowledge, none of the existing work addresses the protocol design aspect of this research. Multipath and real-time transport related functionalities are performed in different ways in existing work. We propose to abstract and augment these common functionalities and make a single generic protocol for multipath real-time transport, thus relieving applications of such burdens as traffic partitioning and mapping flows to a varying number of paths.

B. Multipath Routing

In multimedia communications, the received media quality is affected by link quality metrics, such as bandwidth, delay, jitter, average loss rate, and the loss pattern. In multipath transport, path correlation is another important factor that affects the received media quality. Generally, path diversity (i.e., the independence of failure events on different paths) is strongest when paths are disjoint, which usually yields better error resilience performance. However, focusing solely on disjointedness may cause the use of low quality links, which may give a worse

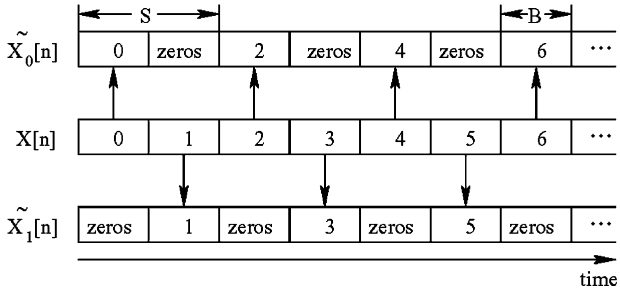


Fig. 1. Thinning example (thinning parameter $S = 2$).

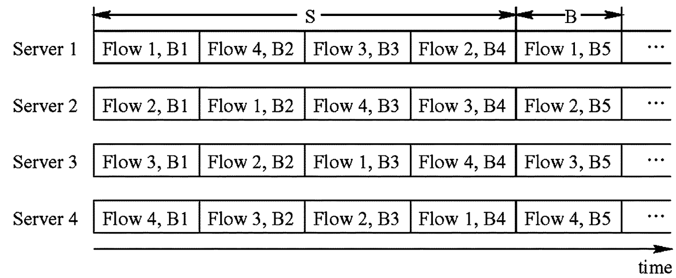


Fig. 2. Striping example (striping parameter $S = 4$).

quality than sharing some "good" links. These factors should be jointly considered in path selection for multimedia sessions.

In the area of wireless ad hoc networks, many existing routing protocols are multipath-capable (e.g., TORA [37] and DREAM [38]). In addition, many ad hoc routing protocols (e.g., DSR [39]) can be extended to support multipath routing (e.g., [13] and [5]). For multimedia-centric routing, several recent papers have provided models for the distortion of received video as a function of link quality metrics and path correlation [8], [23], [28]. Therefore, multipath routing becomes a cross-layer optimization problem, where the received video distortion is minimized. Various heuristic algorithms have been proposed to derive near-optimal solutions [7]–[9], [23], [26], [28].

Once computed, the set of paths could be established in several ways. If source routing is supported by the underlying network (e.g., DSR, IPv4, and IPv6), the sender can store the entire route in the packet headers. Each intermediate node will simply examine the header of a received packet, and forward it to the next node as indicated in the source route. Second, for multihomed hosts, an application can use the stream control transmission protocol (SCTP) sockets [32] to set up multiple connections via different interfaces. Third, multipath routing can be performed via an overlay approach (e.g., by application-level multipath routing). That is, every participating node runs an application layer source routing module. All participating nodes will form an overlay network, within which each logical link may consists of one or more physical wireless links. Thus, multipath routing and packet forwarding can be implemented in the application layer without changing the underlying network architecture and operation [40].

C. Traffic Partitioning and Multistream Coding

In order to use multiple paths, the original multimedia stream should be divided into several substreams (or flows), one for each path used. We introduce several traffic partitioning processes in the following, all of which are supported by MRTP.

With block-based *thinning* [41], a video sequence $X[n]$ is first divided into blocks of equal-sized temporal length B. The block size is expressed in the number of video or audio frames, or some other application-specific temporal payload units. Then, a thinned sequence $\tilde{X}_i[n]$ is assembled by picking blocks periodically from the original blocks in an increasing order, while blocks of zero bits are inserted in place of those skipped blocks. The distance in blocks between two consecutive nonzero blocks is called the *thinning parameter* and denoted as S. An example with $S = 2$ is given in Fig. 1, where two thinned streams are produced from the original stream in a round-robin manner. The thinning parameter S is used to select blocks from the original stream, and the block size B determines the granularity of the partition.

Striping is a technique for data storage and retrieval in distributed systems [20], [22]. With striping, data is partitioned and then stored on multiple *storage elements* (SEs, or servers). A client can stream data in parallel from the SEs. Thus user requests are more evenly distributed among the servers, resulting in a better scalability and lower delay. Striping also makes data downloading more robust to single server failures. Consider the output port of a server, where N flows belonging to N different clients are multiplexed. The multiplexed stream, $\hat{X}[n]$ consists of blocks of data for different clients, as illustrated in Fig. 2. We also use B to denote the block size and S to denote *Striping Parameter*, which is the number of blocks between two consecutive data blocks for the same client.

Furthermore, a *multistream coder* using layered coding or MD coding can produce multiple compressed media flows. In layered coding, a flow is either the base layer or one of the enhancement layers; in MD coding, a flow typically consists of packets from a description. In practice, different coding techniques may have different requirements on network transport. For example, for best performance, layered video requires a reliable path (i.e., unequal error protection) for the base layer. For MD video, strong path diversity is desirable in order to avoid simultaneously losing all the descriptions.

Finally, the MPEG-4 standard uses an object-oriented coding approach, within which various objects (i.e., video or audio components of MPEG-4) are coded individually and can be easily manipulated [42]. Furthermore, objects are placed in so-called elementary streams (ES). A separate flow could be set up for each ES, or the FlexMux tool in MPEG-4 could be used to multiplex several ESs into one flow [42]. The MPEG-7 metadata also provides great flexibility for identifying, retrieving, and manipulating objects and streams [43].

D. Related Transport Protocols

In the following, we briefly review several transport protocols that are closely related to MRTP from the perspective of transporting real-time data or using multiple paths.

The real-time transport protocol (RTP) is basically a framing, timing, and QoS reporting protocol designed to provide generic support for multimedia applications [44]. It has been widely implemented in existing media frameworks, e.g., the Java Media

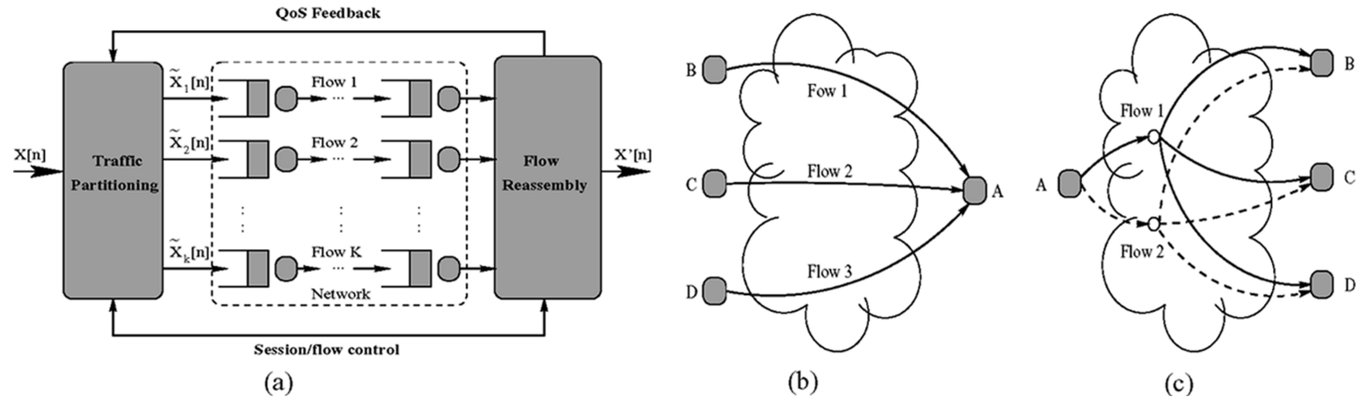


Fig. 3. Usage scenarios of MRTP. (a) Point-to-point multipath video streaming, (b) parallel video streaming, and (c) video multicast with two trees.

Framework (JMF) of Sun Microsystems, Inc. Although not prohibiting the use of multiple paths, RTP may not be amenable to multipath transport. For example, RTP sequence numbers are defined for the overall original stream (rather than per-flow-based), making it difficult for the receiver to infer packet losses or out-of-order events on each individual path. Moreover, RTP does not support traffic partitioning and reassembly, as well as session/flow management for a dynamic environment. To use RTP over multiple paths, an application has to implement these functions by itself. As discussed, a generic protocol that abstracts and augments these common functions for multipath transport will relieve applications of such burdens.

Another closely related protocol is SCTP [32], a message-based transport layer protocol. Although initially designed for reliable signaling in the Internet (e.g., out-of-band control for voice over IP (VoIP) call setup or teardown), SCTP has the potential of replacing UDP and TCP in the transport layer for elastic and real-time data [32]. SCTP has the attractive features of multihoming and multistreaming, where multiple network interfaces or multiple streams can be used in an SCTP session. However, we find that it may be difficult to use SCTP directly for multimedia applications, due to the lack of functions required for real-time services, including framing, timing, and QoS reporting. In addition, although SCTP supports multihoming and multistreaming, it actually sends all traffic on the specified primary path, while all other available paths are used only rarely (e.g., for retransmission, for sending ACKs when duplicated packets are received, or when the primary path fails) [32]. As a result, path diversity is not fully utilized in SCTP.

In a recent work [35], Hsieh *et al.* present a receiver-centric transport protocol called reception control protocol (RCP). RCP is a TCP clone in its general behavior, but allows for better congestion control, loss recovery, and power management mechanisms compared to sender centric approaches. The authors also present a multistate extension of RCP, called $\mathrm{R^2CP}$ for multihomed mobile hosts, which can achieve seamless handoffs, server migration, and bandwidth aggregation [35]. Essentially, $\mathrm{R^2CP}$ is a TCP-type protocol. Its congestion control and flow control may cause large throughput fluctuations, which may be undesirable for multimedia streaming. In addition, it is not clear how to support multicast applications using the proposed protocol.

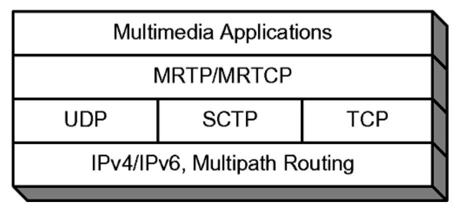


Fig. 4. MRTP protocol stack architecture.

III. MULTIFLOW REAL-TIME TRANSPORT PROTOCOL

A. An Overview

MRTP provides a convenient vehicle for real-time applications to frame, partition, and transmit data using the *association* of multiple flows. A companion control protocol, MRTCP, provides the essential session/flow control, traffic transport engine, and QoS feedback mechanisms.

Fig. 3(a) illustrates a point-to-point MRTP session. At the sender side, the original real-time data $X[n]$ is partitioned into multiple flows $\tilde{X}_1[n], \tilde{X}_2[n], \ldots, \tilde{X}_K[n]$. Applications can choose a data partitioning method and its associated parameters according to their particular requirements. Then, each flow is assigned to a path which is possibly disjoint from those used for other flows. The receiver reassembles the received flows using a resequencing buffer. The reconstructed flow, $X'[n]$, is then retrieved from the buffer to be decoded and displayed.

In addition to unicast applications, MRTP can be applied to group communications as well. Fig. 3(b) depicts a many-to-one MRTP session, where a client streams different portions of a video from three servers (or proxies) concurrently. A one-to-many MRTP session is illustrated in Fig. 3(c), where two source trees are used for video multicast.

The protocol stack architecture of MRTP is depicted in Fig. 4. MRTP is very flexible in utilizing services provided by the underlying protocols. For example, MRTP uses the UDP datagram service or the SCTP transport service for data and control. In addition, the session and control management function (which will be discussed in Section III-C) can also be implemented over the session initiation protocol (SIP) [45], which can use either the TCP or UDP service at the transport layer. An underlying multipath routing protocol maintains multiple paths for an MRTP session.

B. Definitions

In the following, we introduce MRTP/MRTCP messages, extension headers, and other supplemental documents. We first define several terms that are used in the description of the proposed protocol.

- *MRTP Flow*: A designation of real-time data packets transferred from the sender to the receiver. This is similar to an RTP flow.
- *MRTP Session*: An association of one or more MRTP flows, intended to carry a single real-time multimedia stream between the sender and receiver.

MRTP/MRTCP uses three types of packets for data and control, namely, MRTP data packets, MRTCP QoS report packets, and MRTCP session/flow control packets. It also provides the flexibility of defining new extension headers or payload format specifications for emerging multimedia applications.

1) MRTP Data Packet: The format of an MRTP data packet is similar to that of RTP, with several additional fields [46].

- *Session ID*: Identifier of the MRTP session that generated this data packet.
- *Flow ID*: Identifier of the flow that this data packet is transmitted on.
- *Flow Sequence Number*: Sequence number of this packet, indicating its relative position in the stream of packets sent on this flow.

The Session ID is randomly generated when the session is established, and is carried by all the packets belonging to this session. Similarly, both the flow ID and the initial flow sequence number are randomly generated when the flow is first established in the session.

An MRTP data packet is always associated with one of the flows in the session, as indicated by its Flow ID. Within each flow, a packet is assigned a unique flow sequence number (increased by one for each packet transmitted in this flow). The flow sequence number facilitates detection of packet loss (or out-of-order events) within a flow. The receiver uses the flow sequence number, flow ID and timestamp in the packet header to reassemble received packets.

2) MRTCP QoS Reports: An MRTP end point measures QoS statistics of each flow and reports the statistics to other end point(s). A *Sender Report* (SR) is the QoS report generated by a sender, while a *Receiver Report* (RR) is the QoS report generated by a receiver. In the MRTP sessions shown in Fig. 3(a) and (b), the role of an end point, sender or receiver, is fixed as long as it stays in the session. In the MRTP session shown in Fig. 3(c), an end point is a sender if it transmits real-time data during the last reporting period; otherwise, it is a receiver.

MRTCP SR and RR have similar formats as those of RTP, with the following differences [46]: 1) Additional fields specifying which flow the report is for and 2) Flow specific statistics. In order to increase the reliability of feedback, RRs or SRs may be sent on the best path or sent on multiple paths. In the latter case, the timestamp carried in the RR/SR packet can be used to filter out obsolete or duplicated reports.

3) MRTCP Session/Flow Control Messages: MRTCP control messages include the messages used to establish or tear down an MRTP session, to manage the set of flows, and to describe a participant of the session. We introduce these messages here while their usage will be presented in Section III-C.

a) Session control messages: The *HelloSession* message is sent by either a sender or a receiver (called the *initiator*) to initiate an MRTP session. A *HelloSession* message has a common MRTP header, followed by a randomly generated session ID and the total number of flows proposed in this session. Next is a number of flow maps, each associating a flow ID to the corresponding source/destination sockets, i.e., IP address and port number pairs. A randomly generated initial sequence number follows each flow map [46].

An *ACKHelloSession* message is sent to acknowledge the reception of a *HelloSession* message or another *ACKHelloSession* message. Its format is similar to the *HelloSession* message format, but with the initial flow sequence number field replaced by a flow status field [46]. A value of SUCCESS for this field indicates that the proposed flow has been confirmed by the remote end point, while a value of FAIL indicates that the flow was denied.

MRTCP *ByeSession* and *ACKByeSession* messages are used to terminate an MRTP session. Either end point could transmit a *ByeSession* message to terminate the session. The session is terminated after an *ACKByeSession* message is received.

b) Flow control messages: Due to frequent link failures and congestion in ad hoc networks, a path may be broken or congested during transmission, or a new path may be found by the underlying multipath routing protocol. Flow control messages, namely, *AddFlow*, *DeleteFlow*, *ACKAddFlow* and *ACKDeleteFlow*, are used to add or delete flows from an MRTP session dynamically [46].

c) Participant descriptions: As in RTP, *SourceDescription* messages are used in MRTP to describe the source and CNAME (or canonical name) to identify an end point [44].

4) Header Extensions: MRTP uses header extensions to support additional functions not supported by the common header. Each extension has a common *extension header*, followed by extension-specific data. The Type field indicates what kind of extension it is, while the Length field indicates its total length in bytes [46].

In the following, we introduce several examples of MRTP header extensions. New header extensions providing other services can be defined and incorporated into MRTP as well.

- Authentication header extension: It provides a simple authentication mechanism using an ID field and a Password field encrypted with application specific encryption schemes (e.g., *public key encryption*). Alternatively, the MD5 algorithm could be used to generate a digest for the packet [47]. The authentication header extension can be used in session/flow control packets to validate the operations requested, or in an RR or SR to authenticate the report.
- Striping header extension: It consists of fields carrying striping related parameters, such as S and B (see Section II-C). A client can use the striping header extension to inform each server which blocks of the video clip it wishes to download.

MRTP data packet common header (ext = 1)	Source routing extension header (ext = 1)	Authentication extension header (ext = 0)	Data

Fig. 5. Daisy-chain of MRTP header extensions.

- Thinning header extension: As with the striping header extension, the thinning header extension carries thinning related parameters. Thus the sender and receiver can negotiate how thinning is to be performed.
- Source routing header extension: Since multiple paths are used in MRTP, strict or loose source routing can be used to specify the route for each packet. However, if source routing is not supported by the network layer, application-level source routing can be implemented in an overlay manner by defining a source routing extension header to carry the source route (see Section II-B).

We borrowed the idea of *daisy-chain headers* from IPv6 [48]. A one-bit Extension field in the MRTP/MRTCP common header and all the header extensions is used for this purpose. If this bit is set to 1, there will be another header following the current header. This provides the flexibility to combine different extension headers for a specific application. As illustrated in Fig. 5, an MRTP data packet has a common header with $\mathrm{Ext} = 1$, a source routing header extension with $\mathrm{Ext} = 1$, and an authentication header extension with $\mathrm{Ext} = 0$, followed by the multimedia data payload. Thus this MRTP data packet is authenticated, and uses application-level source routing to get to its destination.

It is worth noting that the MRTP header extensions are optional, i.e., an application can decide whether to use it or not. For example, if the underlying IPv6 routing header or authentication header is enabled, the application may disable the corresponding MRTP header extensions, in order to avoid redundancy. MRTP provides such services in the application layer, making it possible for an application to choose services according to its needs and without affecting other applications.

5) MRTP Profiles: MRTP provides various extensions and options in the main protocol (e.g., different methods for traffic partitioning and various header extensions), in order to support various applications with diverse requirements. An application can then make appropriate choices according to its needs. Such choices are specified in the corresponding MRTP profile and payload format specifications for the application.

MRTP supports all existing RTP profiles and specifications. We intend to develop new profiles and payload format specifications for emerging or existing applications, in order to adapt them to multipath transport.

C. Basic Elements of MRTP

In the following, we discuss basic elements of the proposed protocol. Fig. 6, depicts the typical operation of an MRTP session.

1) Connection Establishment and Termination: MRTP is a session-oriented protocol in the sense that an MRTP session needs to be established before data transfer begins. Either the sender or receiver (the *initiator*) can initiate a session with a three-way handshake of *HelloSession* and *ACKHelloSession* messages (see the first three message exchanges in Fig. 6).

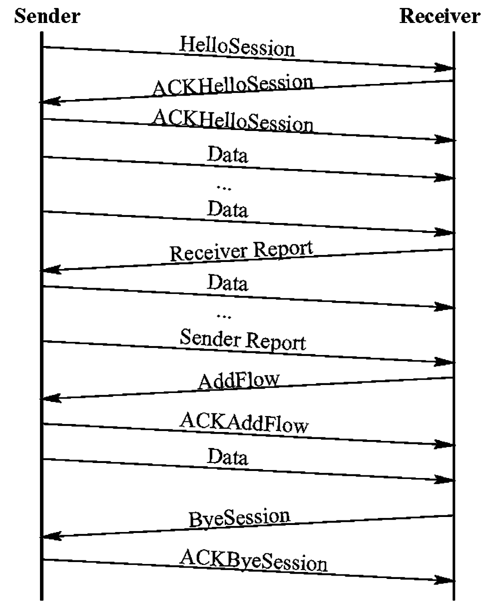


Fig. 6. Operation of an MRTP/MRTCP session.

The three-way handshake gives both end points a chance to choose which flows to use. For example, the initiator may propose to use several flows (carried in the flow mapping fields of the *HelloSession* message). Then, the other end point could choose which proposed flows to use in the returned *ACKHelloSession* message: the Status field of an agreed upon flow is set to SUCCESS, while the Status field of a denied flow is set to FAIL. The initiator further acknowledges this *ACKHelloSession* with another *ACKHelloSession* message. Then, data transmission begins on the confirmed flows.

When data transmission is over (or during the transmission), a participating node may decide to leave the session. This is achieved in different ways for different usage scenarios. For the *point-to-point* multimedia streaming case [see Fig. 3(a)], either of the end points can send an MRTP *ByeSession* message to terminate the MRTP session. The session is terminated after the remote end point responds with an *ACKByeSession* message (see Fig. 6). For a *many-to-one* type application [see Fig. 3(b)], the receiver may send a *ByeSession* message to all the servers, and get acknowledged by each server, to terminate the session. However, a participating server can only send a *DeleteFlow* message if it wishes to leave the session. For multicast application [see Fig. 3(c)], a departing member will send a *ByeSession* to the sender. After receiving the acknowledgment from the sender, the departing member should remove itself from the multicast group using its multicast management protocol [e.g., the Internet group management protocol (IGMP)]. The session is over when there is no participant left.

MRTP uses retransmission timers for control messages to cope with the unreliable UDP/IP service. If there is no response when the timer expires, the control message is retransmitted. The maximum number of retransmissions allowed is set by the application. The timeout value is determined by RTT estimations. A measured RTT sample is computed using the timestamp, Last Report Received, and Delay since Last Report in a received RR or SR, as in RTP [44]. Furthermore, the measured sample is smoothed and the timer value is updated, e.g., using the TCP RTT measurement algorithm [49]. Note that the RTT estimation process will be restarted when there is a path change (e.g., a new flow is added); and be stopped when the corresponding flow is deleted from the session.

MRTCP allows applications to choose how control messages are routed. For example, control messages can be transmitted on all of the paths (with duplicate copies) for better error resilience. On the other hand, control messages can also be transmitted on the best path in terms of quality (inferred from QoS feedback). The first approach is suitable for high loss environments, while the second approach introduces less control traffic overhead.

2) Flow Management: During an MRTP session, some flows may be unavailable. For example, an intermediate node may crash, be congested, or move out of range. In these cases, a receiver will observe excessive packet losses in the corresponding flow. It will then send a *DeleteFlow* message to remove the flow from the session. Packets originally assigned to this flow will be redistributed to other flows. When a new path is found, a new flow can be added to the session by sending an *AddFlow* message. These mechanisms enable MRTP to quickly adapt to topology changes and congestion in the network.

3) Traffic Partitioning and Data Transmission: In MRTP, a traffic allocator partitions real-time data into multiple flows. A basic traffic partitioning scheme is provided in MRTP, which assigns packets to multiple flows using the *round-robin* algorithm. This simple assignment may not be optimal for some applications and can be overridden in such situations. Traffic can be assigned to multiple flows with a granularity of a packet, frame, group of pictures, or substream. Possible traffic partitioning schemes are discussed in Section II-C. After an MRTP session is established, MRTP packets carrying multimedia data are transmitted on the multiple flows associated with the session. Each packet carries a sequence number that is local to its flow and a timestamp that can be used by the receiver to synchronize the flows.

The core implementation of MRTP does not guarantee the reliable delivery of application data. Rather, MRTP relies on lower layers for QoS guarantees. However, MRTP is flexible in supporting various error control schemes. For example, redundancy can be introduced at the traffic allocator when assigning packets to flows, or in a multistream video encoder when compressing the video stream (see Section II-C). Both open-loop error control schemes (e.g., FEC [30] and MDC [29]) and closed-loop error control schemes (e.g., ARQ [5]) can be incorporated into MRTP for better error resilience. It has been shown in previous work that the use of multiple flows makes these error control schemes more effective [4], [5], [23], [28], [30].

4) QoS Feedback: As in RTP, MRTP generates QoS reports periodically. An MRTP SR or RR carries both per-flow and session statistics. In RTP, QoS reports are transmitted at a rate of one report per $T = \max\{T_d, 5\}$ seconds, where T_d is dynamically computed according to the current number of participants in the multicast group and the bandwidth used by the session [44, App. A.7]. This algorithm effectively keeps the bandwidth used by feedback to a relatively constant ratio of the total session bandwidth. However, such a feedback rate may not be frequent enough for the sender to adapt to congestion fluctuations or the rapidly changing topology of an ad hoc network.

Unlike RTP, MRTP SR and RR can be sent at an interval specified by the application. For point-to-point and parallel downloading applications [see Fig. 3(a) and (b)], RR and SR could be sent for each frame since the number of the participants are relatively small. Such timely QoS reports are necessary for highly dynamic ad hoc network environments, so that the sender can quickly adapt to transmission errors. For example, the encoder could change its coding parameters or encoding mode for the next frame [50], introducing more (or less) redundancy for error resilience, or the traffic allocator can assign packets to other paths rather than the error-prone one.

5) Reassembly at the Receiver: When multipath transport is used, there are two types of jitter, i.e., the jitter within each flow, and the jitter across the flows (since each path may have different delays). The MRTP receiver uses a reassembly buffer to absorb jitter and reorder received packets. The receiver first restores the order of each flow using the flow sequence numbers. Then, it examines the head-of-line packets of the flows, and orders them according to their timestamps. Recall that the timestamp of a packet is the sampling instance of the first byte in its payload, and thus is unique in the entire session.

There is a rich literature on the resequencing delay analysis, e.g., see [51] and [52]. Previous work shows that both the resequencing delay and buffer requirements are moderate if the traffic allocator is adaptive to the path conditions inferred from the QoS feedbacks. In [27], Mao *et al.* present an analysis on the optimal traffic partitioning that minimizes the end-to-end delay and its practical implementation using a number of leaky buckets.

D. Usage Scenarios

1) Unicast Video Transport: This is a *point-to-point* scenario as illustrated in Fig. 3(a). Consider a wireless sensor network deployed to monitor, e.g., wildlife, in a remote region [53]. Some sensors carry a video camera, and others are simple relays that pass the captured video to the base. On-demand source routing [5], or other advanced routing algorithms [8], could be used for finding and establishing multiple paths.

A camera sensor initiates an MRTP session to the base. The captured video is transmitted using multiple flows going through different relays. In this way, a relatively high rate video can be spread over multiple paths, each being bandwidth-limited. Redundancy could be introduced by transmitting a more important substream using multiple flows.

Some sensors may be damaged or may run out of power. In this case, the underlying multipath routing protocol informs MRTP about the path changes. Either the sender or the base can delete a failed flow, or add a fresh flow to the session. The server at the base maintains a resequencing buffer for each flow, as well as enforcing a deadline for each packet expected to arrive.

2) Parallel Video Streaming: This is a *many-to-one* scenario, as illustrated in Fig. 3(b). Consider an ad hoc network, where each node maintains a cache for recently downloaded files. When a client A wishes to stream a video file, it would be more efficient to search the caches of nearby nodes first before going directly to a remote server. Suppose the video file is found

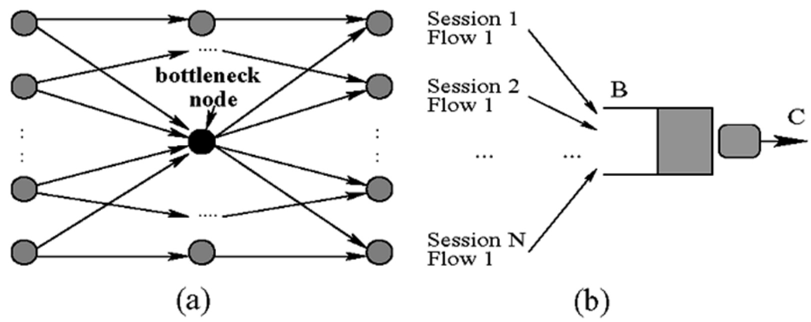


Fig. 7. Performance analysis model for Section IV-A. There are N MRTP sessions, each having a flow traversing the bottleneck node. (a) A bottleneck node in a mesh network and (b) the output buffer of the bottleneck node, multiplexing N flows.

in the caches of nodes B, C, and D. Node A then initiates an MRTP session to these nodes, streaming a piece of the video file simultaneously from each of them. The striping technique introduced in Section II-C is used and the striping header extension (see Section III-B4) is used in each flow to negotiate striping parameters. There are three flows in this session. Node A uses a resequencing buffer to reorder the packets, using flow sequence numbers and timestamps in received packet headers.

During the transmission, if Node D moves out of the network, Node A would delete the flow from D and adjust the striping parameters used in the other two flows. The portion of the video initially chosen for D will be streamed from B and C instead, by sending *AddFlow* messages to B and C with updated striping parameters. On the other hand, node A may broadcast probes periodically to find new neighbors with the video file and replace a flow having a high packet loss rate. MRTP provides the flexibility for applications to implement these schemes.

Such parallel streaming is robust to network partition due to mobility [9]. Further, in order to recover from packet losses, each flow could be protected by a channel coding scheme (e.g., FEC [25]). Combined with multistream video coding schemes, e.g., layered coding with unequal protection of the base layer [5] or multiple description coding [29], error resilience can be greatly improved. In these schemes, the video encoders and traffic allocators can adapt to transmission errors and topology changes inferred from MRTCP QoS reports. An interesting study of video streaming in content delivery networks (CDN) using multiple servers is presented in [23]. The algorithms presented in [9] can be applied to select near-optimal video servers and paths to the servers in ad hoc networks. This usage scenario also applies to peer-to-peer networks when a client streams a video from multiple peers for a lower delay and better error resilience [54].

3) Real-Time Multimedia Multicasting: This is a multicast application similar to RTP-based video teleconferencing. However, MRTP uses multiple multicast trees for a session, as illustrated in Fig. 3(c). In this example, two source trees are used by the sender, each carrying a flow. Since there may be a large number of participants in the session, QoS feedback should be suppressed as in RTP, in order to avoid feedback explosion [44]. A flow's QoS metrics could be averaged over all the receivers of this flow at the sender side.

An example of this usage scenario is reported in [21], where multiple independent multicast trees are used with MD coding to deal with "flash crowds" in the Internet. In [7], Mao *et al.* present a multicasting scheme for MD video in mobile ad hoc networks. The authors also present an efficient genetic algorithm-based scheme for computing a pair of source trees such that the received video quality is optimized.

IV. MRTP Performance Studies

In this section, we present two performance studies of the proposed protocol. Note that the most important feature of MRTP is its multipath transport capability. Consequently, our studies focus on how trafic partitioning and multipath transport improves received media quality. Specifically, we first investigate the impact of traffic partitioning on the queueing performance of multimedia flows. We then present a simulation study of MRTP using OPNET, comparing MRTP with single-flow RTP for video streaming in a mobile ad hoc network.

A. The Impact of Traffic Partitioning

In the following, we focus on the impact of traffic partitioning on congestion at bottleneck links. We show, analytically, *how* traffic partitioning improves the queueing performance of multimedia flows, *how much* improvement could be achieved, and *how many* MRTP flows are needed to achieve most of the potential improvement.

1) Analytical Model: Consider a bottleneck node in the network. There are N flows, from different MRTP sessions, that traverse this node, as illustrated in Fig. 7(a). Assuming that the flows are independent and homogeneous, the output buffer of the bottleneck node can be modeled as a multiplexer of N *i.i.d.* flows, as depicted in Fig. 7(b). A multimedia data stream, e.g., variable bit rate video, is thinned (or striped) with various thinning (striping) parameters S and fixed block size $B = 1$ sample (see Section II-C). In the following, we examine the impact of thinning and striping on the queueing performance of the flows at this bottleneck node.

Previous work on large deviation techniques shows that the buffer overflow probability of a queue having buffer size B and service capacity C and fed by N homogeneous sources can be approximated by the *Bahadur–Rao Asymptotic* [11]:

$$\Psi(c, b, N) \approx \exp[-NI(c, b) + g(c, b, N)] \tag{1}$$

where

$$\begin{cases} I(c, b) = \inf_{m \geq 1} \frac{[b+m(c-\mu)]^2}{2V(m)} \\ g(c, b, N) \approx -\frac{1}{2} \log[4\pi NI(c, b))]. \end{cases} \tag{2}$$

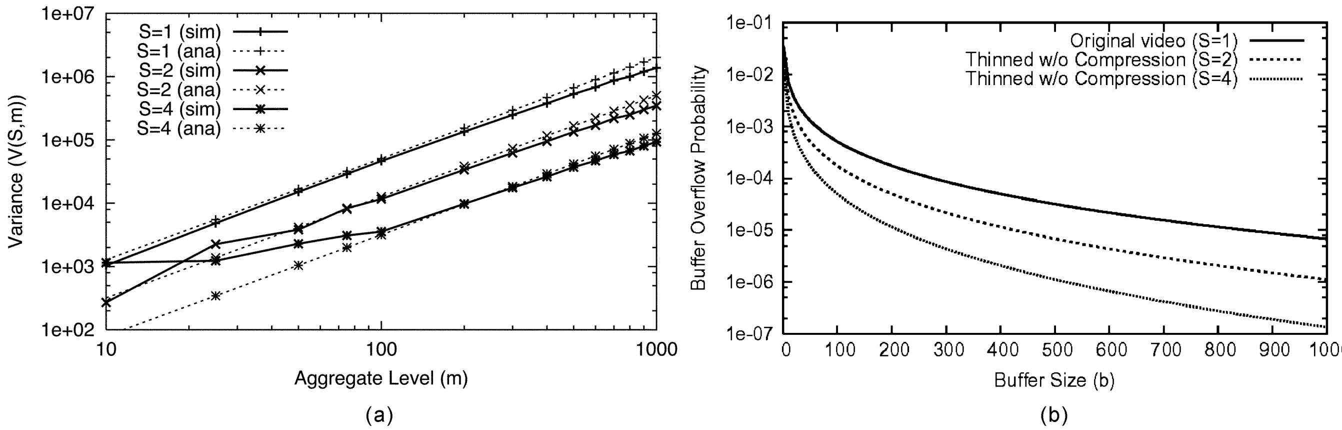


Fig. 8. Traffic partitioning reduces the short-range dependence in a multimedia flow and reduces buffer overflow probability. (a) Variance $V(S, m)$ with different aggregation level m and (b) buffer overflow probability of a queue fed by 100 video flows.

In (1) and (2), c and b are the per-flow capacity and buffer assignments, respectively (i.e., $c = C/N, b = B/N$), and μ is the average rate of a source. $V(m)$ is the variance of a single source with a temporal aggregation level m. That is, given the original sequence $\{X_i, i = 1, 2, \ldots\}$, we can derive a new sequence $\{Y_k = \sum_{i=km}^{(k+1)m-1} X_i, k = 1, 2, \ldots\}$ with aggregation level m. Then, $V(m)$ is the variance of $\{Y_k\}$, i.e., $V(m) = \mathrm{var}\{Y_k\}$ Note that $V(1) = \sigma^2$, which is the variance of the source.

It is well-known that variable bit rate video traffic is *Long Range Dependent* (LRD) [12]. For a second-order exactly or asymptotically self-similar source with Hurst parameter H, $0.5 < H < 1$, the variance $V(m)$ can be approximated as [55]:

$$V(m) \approx \sigma^2 m^{2H}, \text{ for large } m. \tag{3}$$

Substituting this $V(m)$ into (2), we have

$$I(c, b) = \frac{(c - \mu)^{2H} b^{2-2H}}{2\sigma^2 \kappa^2(H)} \tag{4}$$

where $\kappa(H) = H^H (1 - H)^{1-H}$.

For a thinned video stream with thinning parameter S, we have [41]

$$\tilde{V}(S, m) \approx S^{-2}\sigma^2 m^{2H}, \text{ for large } m. \tag{5}$$

Given the same system parameters B and C, and the same number of flows N, we can derive the following for the queueing system fed by the thinned flows:

$$\tilde{I}(S, c, b) = \frac{(Sc - \mu)^{2H} (Sb)^{2-2H}}{2\sigma^2 \kappa^2(H)}. \tag{6}$$

Note that when $S = 1, \tilde{I}(S, c, b)$ reduces to $I(c, b)$ in (4). According to (6), thinning reduces the short-term auto-correlation of a flow, which is equivalent to a larger per-flow capacity and buffer assignment.

Similarly, for a striped flow with striping parameter S, we have [41]:

$$\hat{V}(S, m) \approx S^{-1}\sigma^2 m^{2H}, \text{ for large } m. \tag{7}$$

We can then derive the following for the queueing system fed by the stripped flows:

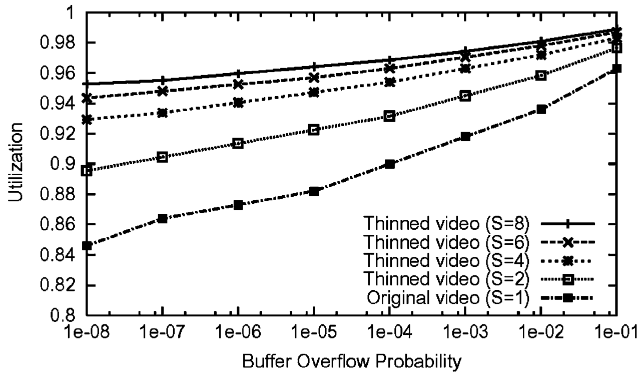
$$\hat{I}(S, c, b) = \frac{S(c - \mu)^{2H} b^{2-2H}}{2\sigma^2 \kappa^2(H)}. \tag{8}$$

When $S = 1, \hat{I}(S, c, b)$ reduces to $I(c, b)$ in (4).

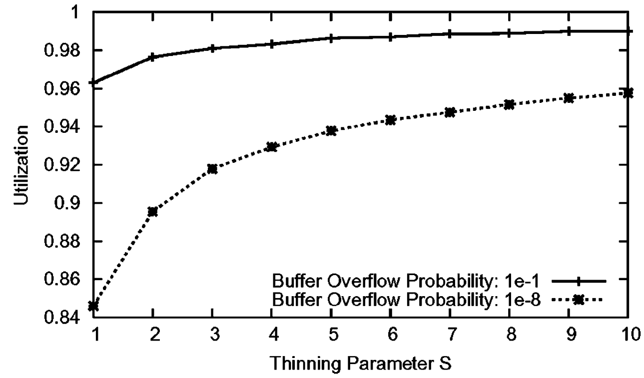
2) Improvement on Buffer Overflow Probability: In order to illustrate how traffic partitioning improves queueing performance, we generate a self-similar traffic trace that is the aggregate of 128 identical on-off sources with Pareto distributed on and off periods. An on-off source generates one unit of data per time slot when it is on, and is idle in the off state. The generated self-similar trace has a mean of 64, a variance of 32, and a Hurst parameter of 0.8 or 0.88. We used a trace of 10 000 000 samples for the results reported in this section.

In Fig. 8(a), we plot $V(S, m)$ computed from the original synthesized data and the thinned data when $S = 2$ and 4, respectively. Note that all the three curves increase linearly with m in the log-log plot, indicating that both the original trace and the thinned traces are self-similar. It can be easily verified that the slopes of the three curves are all equal to $2H$. Therefore, the long-term correlation in the trace is not affected by the thinning process. This observation is consistent with the common belief that long-range dependence (or self-similarity) is quite persistent: it cannot be reduced by traffic shaping or queueing [10]. However, the amplitude of $V(S, m)$ is effectively reduced by thinning. The analytical results using (5) are also plotted in Fig. 8(a), which match the simulation results for large m.

The reduced $V(S, m)$ results in improved queueing performance. Fig. 8(b) plots the buffer overflow probabilities of a queue having $c = 1$ and $\mu = 0.9$. The queue is fed by 100 thinned sources having Hurst parameter $H = 0.88, \sigma^2 = 1$. Three curves for the cases of $S = 1, S = 2$, and $S = 4$ are plotted. For fair comparisons, the system loads for the three curves are kept the same by reducing the service capacity of the queue by a factor of 2 and 4 for the $S = 2$ and $S = 4$ curves, respectively. We observe that with thinning, the buffer overflow probability is greatly reduced. The improved buffer overflow probability translates to a smaller delay, a lower packet lost rate, and a smaller jitter. We have observed the same trend for striping, but omit the corresponding results for brevity.



(a)



(b)

Fig. 9. Impact of thinning on bandwidth utilization. (a) Bandwidth utilizations for different requirements on buffer over flow probability and (b) bandwidth utilization versus the thinning parameter S.

3) Improvement on Bandwidth Utilization: To further illustrate the achievable improvements, consider a bottleneck queue having capacity $C = 100$, buffer $B = 40\,000$, and fed by N flows, each having Hurst parameter $H = 0.88, \sigma^2 = 1$, and $\mu = 0.9$. Fig. 9(a) plots the bandwidth utilization, defined as the sum of the average rates of all the flows over the server capacity, i.e., $\rho = N \cdot \mu / C$, for different buffer overflow requirements and thinning parameters. As expected, the bandwidth utilization increases as the buffer overflow requirement gets less stringent (e.g., from 10^{-7} to 10^{-1}). However, the bandwidth utilization also increases as S increases, i.e., as more flows are used in each MRTP session. For a buffer overflow requirement of 10^{-8}, there is an 11.2% increase in ρ when S increases from 1 to 8 [also see Fig. 9(b)].

In Fig. 9(b), we plot the utilizations of the system for different thinning parameters, when the buffer overflow probability is 10^{-1} and 10^{-8}, respectively. We observe that the largest improvement is achieved when S increases from 1 to 2 in both curves. When S increases further, the improvement gets progressively smaller. For example, when the buffer overflow probability requirement is 10^{-8}, there is a 5.0% improvement in bandwidth utilization when S increases from 1 to 2, while the improvement is only 0.3% when S increases from 9 to 10. This implies that only a few MRTP flows are needed to get the most benefits of multipath transport.

In Fig. 10, we vary the server capacity C and the buffer size B of the system, and plot the improvement in bandwidth utilization when S increases from 1 to 8. The buffer overflow probability requirement is fixed at 10^{-6}. In other words, each point on the surface in Fig. 10 is the difference between the bandwidth utilizations when $S = 8$ and $S = 1$. The improvement in bandwidth utilization ranges from 6.8% to 13.1%. In addition, we found that the improvement is higher when either C or B decreases. This trend can be well explained by the results in [56]. Choe and Shroff show in [56] that the Central Limit Theorem (CLT) comes into play for a multiplexer with a large number of flows. When the buffer size is very large, the buffer overflow probability behaves as if the queue is driven by uncorrelated input processes, regardless of the correlation structure of the input processes. We observe similar trends when striping is used. This trend implies that MRTP could be helpful in combating congestion in mobile ad hoc networks, which, as discussed (see Section I), are usually bandwidth limited, and in the Internet, where bandwidth limited access routers or access points for wireless access networks are usually the bottleneck of an end-to-end session.

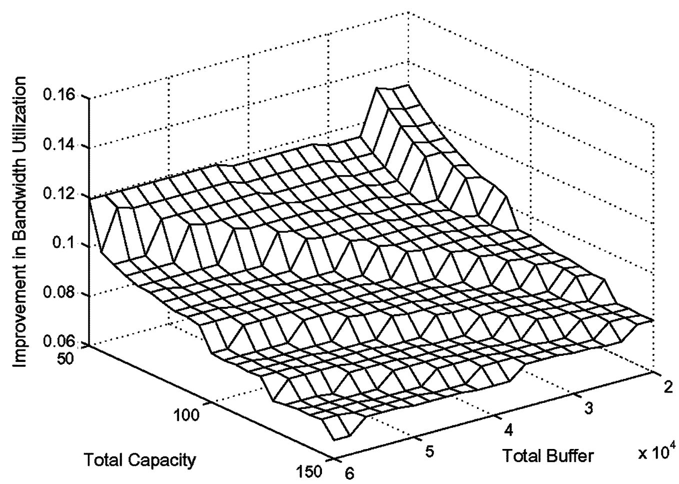


Fig. 10. Improvement in bandwidth utilization for different server capacities and buffer sizes when thinning parameter S is increased from 1 to 8.

B. Video Transport Over Ad Hoc Networks

In this section, we present an MRTP performance study using OPNET simulations. We model most layers of the protocol stack (including part of the physical layer, Wireless LAN MAC, multipath routing, MRTP, and the application layer), as well as the two key characteristics of mobile ad hoc networks: mobility and multihop wireless communications. With such simulations, we examine the performance of MRTP in a realistic ad hoc network environment.

We simulate an ad hoc network in a square region, while each node is randomly placed in the region initially. The popular *Random Waypoint* mobility model is used [57], with *constant* nodal speed and pause time. For the results reported in this section, the network consists of 16 nodes in a $600\ \text{m} \times 600\ \text{m}$ region. The node speed is 5 m/s and the pause time is 2 s. We use the IEEE 802.11 protocol in the MAC layer working in the DCF mode. The channel bandwidth is 1 Mbps and the transmission range is 250 meters.

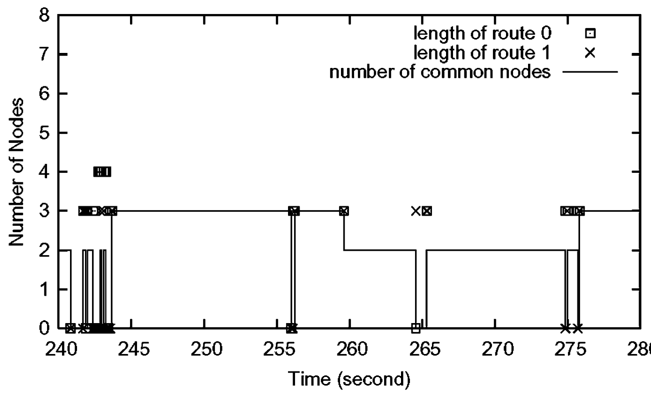


Fig. 11. Path dynamics in a 16-node mobile ad hoc network with node speed of 5 m/s and pause time of 2 s. The traces of the two routes are plotted, from the video source node to the video sink node used by MRTP.

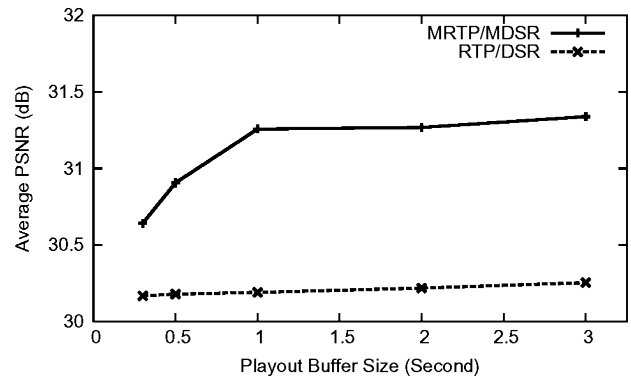


Fig. 12. Average PSNR for different playout buffer sizes.

The quarter common intermediate format (QCIF) (176×144 Y pixels/frame, 88×72 Cb/Cr pixels/frame) sequence "Foreman" (200 even frames from the original 30 fps sequence) is encoded at 10 fps and used in our simulations. For long simulations, the same 200-frame sequence is repeatedly used. The multiple description motion compensation (MDMC) video codec is used [29] to generate two video flows (or descriptions), each with a bit rate of 59 Kbps, and to decode received video frames. The encoder uses a 5% macroblock level intra-refreshment, which has been found to be effective in suppressing error propagation for the range of the packet loss rates considered [29]. Each group of blocks (GOB) is packetized into a single MRTP data packet, to make it independently decodable (i.e., nine packets for each frame).

In each simulation, one node is randomly chosen as the video source and another as the video receiver, where a playout buffer is used to absorb received packet jitter. The MRTP session uses two routes. The multipath routing DSR (MDSR) model [5] (a multipath extension of the dynamic source routing (DSR) protocol [39]), is deployed to maintain two maximally node-disjoint routes for the MRTP session. All other nodes generate background traffic for a randomly chosen destination. The inter-arrival time of the background packets is exponentially distributed with a mean of 0.2 second. The background packets have a constant length of 512 bits.

In Fig. 11, we plot the dynamics of the two paths used by an MRTP session during a simulation. The length of a route is denoted by the total number of nodes the route traverses, including the source and destination nodes. Furthermore, each point in the figure indicates a route update: either a better route is found, or a route in use is broken. We find that the routes are highly dynamic (e.g., see the path changes between the 240th and 245th second). The MRTP session/flow management and QoS reporting functions will be helpful for applications to handle the dynamic flows in such environments. We also plot the number of common nodes, which includes the source and destination nodes, between the two paths.

Next, we compare the MRTP performance with that of RTP, using the same MDMC codec and the same video clip. For the RTP simulations, we used the NIST DSR routing model that maintains a single path to a destination. The two descriptions are interleaved and transmitted on this path. The MRTP session uses two flows, each transmitted on one of the two paths found by MDSR. The receiver decodes received frames (corrupted by transmission errors) and computes their peak signal to noise ratios (PSNR). In Fig. 12, we present average PSNR values of received frames for different playout buffer sizes. Each PSNR value in the figure is the average of 8000 decoded frames. As expected, the average PSNR increases with playout buffer size for both RTP and MRTP simulations. This is because with a larger playout buffer, packets that temporarily experience large delays can still be used for decoding. In the MRTP case, when the playout buffer size increases from 300 ms to 1 s, there is a 0.63 dB gain in average PSNR of decoded frames, which is significant in terms of visual quality. However, the improvement gets smaller for further increases in playout buffer size, implying that most remaining packet losses are due to path failure (i.e., dropped at a failed link), rather than congestion. Furthermore, we find that the two-flow MRTP outperforms the single-flow RTP in all of the cases, with average PSNR improvements ranging from 0.66 dB to 1.29 dB.

Although Fig. 12 demonstrates improved video quality in the average sense, it would be more interesting to examine the quality of individual frames. In Fig. 13, we plot the PSNR traces obtained from a two-flow MRTP simulation and a single-flow RTP simulation. The playout buffer is set to 1 s for both simulations. We discard the first 1000 frames in order to eliminate the impact of initial node placements. The same random seed is used for the random number generators in both simulations, such that a node has the same trajectory in both simulations. As a result, the deep valleys of the two PSNR curves occur at about the same time instances (e.g., when the source and destination nodes are far apart from each other, or one of them is in a hot-spot).

For the MRTP simulation, the average packet loss ratio is 3.3%, with an average burst length of 19 packets. For the RTP simulation, the average loss ratio is 9.2%, with an average burst length of 213 packets. Recall that there are nine packets for each frame. Clearly, using two flows not only reduces the average loss ratio, but also makes packet losses more random. We observe that the degradation in PSNR is more persistent in Fig. 13(b), since when the single flow is down, both video descriptions are lost. In the MRTP case, when one flow fails, it is likely that the other flow is still up, since the paths are maximally node-

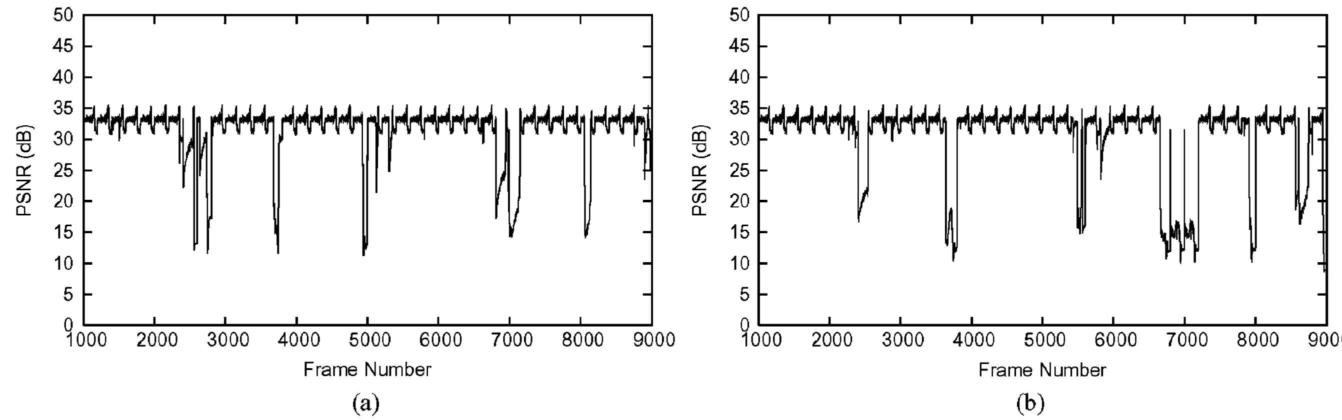


Fig. 13. PSNRs of the decoded frames from the two-flow MRTP and the single-flow RTP simulations (5 m/s nodal speed and 2 s pause time). (a) MRTP/MDSR with two flows and (b) RTP/DSR with one flow.

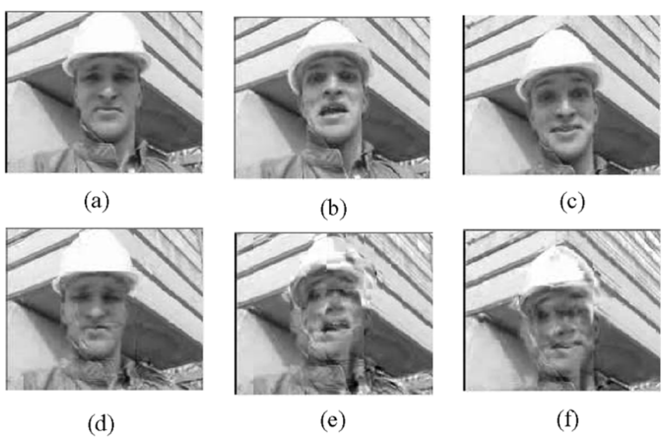


Fig. 14. Decoded frames from the MRTP and RTP simulations. (a) MRTP: frame 8630, (b) MRTP: frame 8635, (c) MRTP: frame 8640, (d) RTP: frame 8630, (e) RTP: frame 8635, and (f) RTP: frame 8640.

disjoint. The MDMC decoder can use received information in a description to recover information carried in the lost description [29], and therefore, can recover more quickly from a burst of packet losses (e.g., see the valleys around 7000 s in both figures). The average PSNR for the MRTP case in Fig. 13(a) is 31.45 dB, while the average PSNR for the RTP case in Fig. 13(b) is 30.19 dB. A significant gain of 1.26 dB is achieved when two MRTP flows are used.

For multimedia applications, the ultimate performance measure is perceived QoS at the receiver. In order to visually demonstrate the received quality, we plot several decoded frames from the RTP and MRTP simulations in Fig. 14. Note that when there is no large burst of packet losses, e.g., as around 6000 s in Fig. 13(a) or around 5000 s in Fig. 13(b), the decoded frames from the two simulations have similar perceived quality. However, during a large burst of packet losses, a long sequence of frames will have low quality in the RTP case, while MRTP can recover from the losses more quickly with a smaller number of frames affected. The decoded frames in Fig. 14 are Frames 8630, 8635, and 8640. During this period, all three frames are affected by a loss burst in the RTP case (frame 8630 is the first frame affected by this burst), and have high distortions as compared to the corresponding frames from the MRTP simulation.

V. Conclusion

We presented the MRTP protocol for real-time multimedia transport over mobile ad hoc networks. Our proposal was motivated by the recent considerable research effort on multimedia communications with path diversity. MRTP is an extension of existing real-time transport protocols (i.e., RTP/RTCP) that incorporates the multipath transport capability; it is also complementary to existing data-centric transport protocols (i.e., SCTP) for real-time multimedia applications.

We also presented two performance studies of the proposed protocol. The first study focused on the impact of traffic partitioning on the queueing performance of real-time flows at a bottleneck node. We showed that the bandwidth utilization of a bottleneck node can be greatly improved when MRTP is used. In addition, the improvement can be achieved with a relatively small number of MRTP flows. The second study focused on the error resilience aspect of MRTP. Through OPNET simulations, we showed that MRTP achieves significant improvements on received video quality over single-path RTP.

Although MRTP has been developed in the context of mobile ad hoc networks, we believe that MRTP can be applied in the Internet, when an institutional network has multiple access routers, for infrastructure-based wireless networks, when multiple base stations can be accessed in parallel, and for multimedia data sharing in P2P overlay networks for improved media quality.

Acknowledgment

The authors are grateful to the three anonymous reviewers whose comments helped improve the quality of this paper. They also thank Prof. Y. Wang of Polytechnic University, Brooklyn, NY, and Dr. S. Lin of Harmonic, Inc., for providing the MDMC codec, and the Wireless Communications Technologies Group of the National Institute of Standards and Technology for making the OPNET DSR model available in the public domain.

References

[1] Y. Wang and Q.-F. Zhu, "Error control and concealment for video communication: A review," *Proc. IEEE*, vol. 86, no. 5, pp. 974–997, May 1998.

[2] J. Li, C. Blake, D. S. J. de Couto, H. I. Lee, and R. Morris, "Capacity of ad hoc wireless networks," in *Proc. ACM MobiCom 2001*, Rome, Italy, Jul. 2001, pp. 61–69.

[3] N. Gogate and S. S. Panwar, "Supporting video/image applications in a mobile multihop radio environment using route diversity," in *Proc. IEEE ICC*, Jun. 1999, pp. 1701–1706.

[4] N. Gogate, D. Chung, S. S. Panwar, and Y. Wang, "Supporting image/video applications in a multihop radio environment using route diversity and multiple description coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 9, pp. 777–792, Sep. 2002.

[5] S. Mao, S. Lin, S. S. Panwar, Y. Wang, and E. Celebi, "Video transport over ad hoc networks: Multistream coding with multipath transport," *IEEE J. Sel. Areas Commun.*, vol. 21, no. 10, pp. 1721–1737, Dec. 2003.

[6] E. Gustafsson and G. Karlsson, "A literature survey on traffic dispersion," *IEEE Network*, vol. 11, no. 2, pp. 28–36, Mar. 1997.

[7] S. Mao, X. Lin, Y. T. Hou, and H. D. Sherali, "Multiple description video multicast in wireless ad hoc networks," *Mobile Networks Applicat. J. (MONET)*, vol. 11, no. 1, pp. 63–73, Jan. 2006.

[8] S. Mao, Y. T. Hou, X. Cheng, H. D. Sherali, and S. F. Midkiff, "Multipath routing for multiple description video over wireless ad hoc networks," in *Proc. IEEE INFOCOM*, Miami, FL, Mar. 2005, pp. 740–750.

[9] S. Mao, X. Cheng, Y. T. Hou, H. D. Sherali, and J. H. Reed, "Joint routing and server selection for multiple description video in wireless ad hoc networks," in *Proc. IEEE ICC*, Seoul, Korea, May 2005.

[10] A. Erramilli, O. Narayan, and W. Willinger, "Experimental queueing analysis with long-range dependent packet traffic," *IEEE Trans. Networking*, vol. 4, no. 2, pp. 209–223, Apr. 1996.

[11] M. Montgomery and G. De Veciana, "On the relevance of time scales in performance oriented traffic characterization," in *Proc. IEEE INFOCOM*, San Francisco, CA, Mar. 1996, pp. 513–520.

[12] J. Beran, R. Sherman, M. S. Taqqu, and W. Willinger, "Long-range dependence in variable-bit-rate video traffic," *IEEE Trans. Commun.*, vol. 43, no. 2/3/4, pp. 1566–1579, Feb./Mar./Apr. 1995.

[13] A. Nasipuri and S. R. Das, "On-demand multipath routing for mobile ad hoc networks," in *Proc. IEEE ICCCN*, Boston, MA, Oct. 1999, pp. 64–70.

[14] Y. J. Liang, E. G. Steinbach, and B. Girod, "Real-time voice communication over the internet using packet path diversity," in *Proc. ACM Multimedia 2001*, Ottawa, ON, Canada, Sept./Oct. 2001, pp. 431–440.

[15] A. Miu, J. G. Apostolopoulos, W.-T. Tan, and M. Trott, "Low-latency wireless video over 802.11 networks using path diversity," in *Proc. IEEE ICME*, Baltimore, MD, Jul. 2003, pp. IL-441–IL-4444.

[16] R. Chandramouli, R. Shorey, P. K. Srimani, X. Wang, and H. Yu, "Special issue on recent advances in wireless multimedia," *IEEE J. Sel. Areas Commun., Special Issue on Recent Advances in Wireless Multimedia*, vol. 21, no. 10, Dec. 2003.

[17] W. Zhu, M.-T. Sun, L.-G. Chen, and T. Sikora, *Proc. IEEE, Special Issue on Advances in Video Coding and Delivery*, vol. 93, no. l, Jan. 2005.

[18] B. Girod, I. Lagendijk, Q. Zhang, and W. Zhu, *IEEE Wireless Commun., Special Issue on Advances in Wireless Video*, Jun. 2005.

[19] A. Banerjea, "Simulation study of the capacity effects of dispersity routing for fault tolerant realtime channels," in *Proc. ACM SIGCOMM*, Stanford, CA, Aug. 1996, pp. 194–205.

[20] M. Reisslein, K. Ross, and S. Shrestha, "Striping for interactive video: Is it worth it," in *Proc. IEEE International Conference on Multimedia Computing and Systems*, Florence, Italy, Jun. 1999, pp. 7–11.

[21] V. N. Padmanabhan, H. J. Wang, P. A. Chou, and K. Sripanidkulchai, "Distributing streaming media content using cooperative networking," in *Proc. ACM NOSSDAV*, Miami, FL, May 2002, pp. 177–186.

[22] J. Y. B. Lee, "Staggered push—A linearly scalable architecture for push-based parallel video servers," *IEEE Trans. Multimedia*, vol. 4, no. 2, pp. 423–433, Dec. 2002.

[23] J. G. Apostolopoulos, T. Wong, W. Tan, and S. Wee, "On multiple description streaming in content delivery networks," in *Proc. IEEE INFOCOM*, New York, June 2002, pp. 1736–1745.

[24] J. Chakareski, S. Han, and B. Girod, "Layered coding vs. multiple descriptions for video streaming over multiple paths," in *Proc. ACM Multimedia 2003*, Nov. 2003, pp. 422–431.

[25] T. Nguyen and A. Zakhor, "Multiple sender distributed video streaming," *IEEE Trans. Multimedia*, vol. 6, no. 2, pp. 315–326, Apr. 2004.

[26] Z. Ma, H.-R. Shao, and C. Shen, "A new multi-path selection scheme for video streaming on overlay networks," in *Proc. IEEE ICC*, Paris, France, June 2004, pp. 1330–1334.

[27] S. Mao, S. S. Panwar, and Y. T. Hou, "On optimal traffic partitioning for multipath transport," in *Proc. IEEE INFOCOM*, Miami, FL, Mar. 2005, pp. 2325–2336.

[28] A. C. Begen, Y. Altunbasak, O. Ergun, and M. M. Ammar, "Multi-path selection for multiple description encoded video streaming," in *EURASIP Signal Process.: Image Commun.*, vol. 20, Jan. 2005, pp. 39–60.

[29] Y. Wang and S. Lin, "Error resilient video coding using multiple description motion compensation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 6, pp. 438–452, Sept. 2002.

[30] T. Nguyen and A. Zakhor, "Path diversity with forward error correction (PDF) system for packet switched networks," in *Proc. IEEE INFOCOM*, San Francisco, CA, Apr. 2003, pp. 663–672.

[31] N. F. Maxemchuk, "Dispersity routing in high speed networks," *Comput. Networks ISDN Syst.*, vol. 25, pp. 645–661, Jan. 1993.

[32] R. Stewart, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, and V. Paxson, Stream Control Transmission Protocol, Oct. 2000. IETF RFC 2960.

[33] D. S. Phatak and T. Goff, "A novel mechanism for data streaming across multiple ip links for improving throughput and reliability in mobile environments," in *Proc. IEEE INFOCOM*, New York, June 2002, pp. 773–782.

[34] H.-Y. Hsieh and R. Sivakumar, "A transport layer approach for achieving aggregate bandwidths on multi-homed mobile hosts," in *Proc. ACM MobiCom*, Atlanta, GA, Sep. 2002, pp. 83–95.

[35] H.-Y. Hsieh, K.-H. Kim, Y. Zhu, and R. Sivakumar, "A receiver-centric transport protocol for mobile hosts with heterogeneous wireless interfaces," in *Proc. ACM Mobicom*, San Diego, CA, Sept. 2003, pp. 1–15.

[36] V. K. Goyal, "Multiple description coding: Compression meets the network," *IEEE Sigal Process. Mag.*, vol. 18, pp. 74–93, Sept. 2001.

[37] V. D. Park and M. S. Corson, "A highly adaptive distributed routing algorithm for mobile wireless networks," in *Proc. IEEE INFOCOM*, Kobe, Japan, Apr. 1997, pp. 1405–1413.

[38] S. Basagni, I. Chlamtac, V. R. Syrotiuk, and B. A. Woodward, "A distance routing effect algorithm for mobility (DREAM)," in *Proc. ACM MobiCom*, Dallas, TX, Oct. 1998, pp. 76–84.

[39] D. B. Johnson, D. A. Maltz, and Y.-C. Hu, The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR), Apr. 2003. IETF Internet Draft draft-ietf-manet-dsr-09.txt.

[40] J. Liebeherr, J. Wang, and G. Zhang, "Programming overlay networks with overlay sockets," in *5th COST 264 Workshop on Networked Group Communications (NGC 2003), LNCS 2816*, Munich, Germany, Sep. 2003, pp. 242–253.

[41] D. Bushmitch, "Thinning, striping and shuffling: Traffic shaping and transport techniques for VBR Video," Ph.D. dissertation, Polytechnic Univ., Brooklyn, NY, Jan. 2004.

[42] R. Koenen, Ed., *Overview of the MPEG-4 Standard*, May/June 2000. ISO/IEC 14496.

[43] *MPEG-7 Overview (Version 9)*, J. M. Martinez, Ed., Mar. 2003. ISO/IEC JTC1/SC29/WG11 N5525.

[44] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Realtime Applications,", July 2003. IETF Request For Comments 3550.

[45] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "SIP: Session Initiation Protocol,", June 2002. IETF Request For Comments 3261.

[46] S. Narayanan, D. Bushmitch, S. Mao, and S. S. Panwar, "MRTP: A multi-flow real-time transport protocol", Aug. 2004. work in progress, IETF Internet Draft draft-narayanan-mrtp-00.txt.

[47] R. Rivest, "The MD5 message-digest algorithm", Apr. 1992. IETF Request For Comments 1321.

[48] C. Huitema, *IPv6: The New Internet Protocol*. Upper Saddle River, NJ: Prentice-Hall, 1998.

[49] W. R. Stevens, *TCP/IP Illustrated, Volume 1: The Protocols*. Reading, MA: Addison-Wesley, 1994.

[50] B. Girod and N. Farber, "Feedback-based error control for mobile video transmission," *Proc. IEEE*, vol. 87, no. 10, pp. 1707–1723, Oct. 1999.

[51] N. Gogate and S. S. Panwar, "On a resequencing model for high speed networks," in *Proc. IEEE INFOCOM*, Toronto, ON, Canada, June 1994, pp. 40–47.

[52] Y. Nebat and M. Sidi, "Resequencing considerations in parallel downloads," in *Proc. IEEE INFOCOM*, New York, 2002, pp. 1326–1335.

[53] Y. Li, S. S. Panwar, S. Mao, and S. Burugupalli, "A mobile ad hoc biosensor network," in *Proc. 2005 IEEE Int. Conf. Communications (ICC)*, Seoul, Korea, May 2005.

[54] X. Xu, Y. Wang, S. S. Panwar, and K. W. Ross, "A peer-to-peer video-on-demand system using multiple description coding and server diversity," in *Proc. IEEE ICIP 2004*, Singapore, Oct. 2004, pp. 1759–1762.

[55] J. Beran, *Statistics for Long-Memory Processes*. New York, NY: Chapman & Hall, 1994.

[56] J. Choe and N. B. Shroff, "A Central-Limit-Theorem-based approach for analyzing queue behavior in high-speed networks," *IEEE Trans. Networking*, vol. 6, no. 5, pp. 659–671, Oct. 1998.

[57] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," in *Proc. ACM/IEEE Inter. Conf. Mobile Comp. and Networking*, Dallas, TX, Oct. 1998, pp. 85–97.

Shiwen Mao (S'99–M'04) received the B.S. degree in electrical engineering and the B.E. degree in business management from Tsinghua University, Beijing, China, in 1994, the M.S. degree in electrical engineering from Tsinghua University in 1997, the M.S. degree in systems engineering from Polytechnic University, Brooklyn, NY, in 2000 , and the Ph.D. degree in electrical and computer engineering from Polytechnic University in 2004.

He was a Research Member at IBM China Research Laboratory, Beijing, from 1997 to 1998. From 1998 to 2003, he was a Research Fellow at the New York State Center for Advanced Technology in Telecommunications. In the summer of 2001, he was a research intern at Avaya Labs-Research, Holmdel, NJ. Currently, he is a Research Scientist in the Bradley Department of Electrical and Computer Engineering, Virginia Polytechnic Institute and State University, Blacksburg. His research interests include multimedia networking, optimization, algorithms, and performance analysis of wireless networks. He co-authored the textbook *TCP/IP Essentials: A Lab-Based Approach* (Cambridge University Press, 2004).

Dr. Mao is a co-recipient of the 2004 IEEE Communications Society Leonard G. Abraham Prize in the Field of Communications Systems.

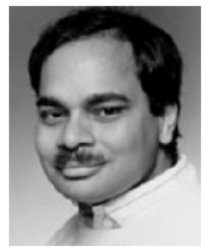
Dennis Bushmitch (S'91–M'03) received the B.S. degree in electrical engineering (summa cum laude) from New York Institute of Technology (NYIT) in 1994 and the M.S. and Ph.D. degrees in electrical engineering from Polytechnic University, Brooklyn, NY, in 1997 and 2003 respectively.

In 1994, he joined Bell Communications Research as a Research Scientist, and in 1997, moved to Panasonic Information and Networking Technologies Laboratory, Princeton NJ, where he is currently a Senior Scientist. His research interests include service discovery, session management, and media streaming middleware and protocols for mobile and peer-to-peer networks, media transport protocols for mobile ad hoc networks, multimedia traffic characterization and network performance analysis, QoS in mobile network architectures and admission control, real-time reliable multimedia streaming, broadcast DTV—home-mobile internetworking convergence architectures, and computer and network security. He holds 11 U.S. patents, with some 40 pending, and has published a number of journal and conference papers in these areas. He is active in DLNA, IETF, UPnP and OSGi standardization forums, authoring several professional standards for home networking.

Dr. Bushmitch is a member of the IEEE Communications Society, the Association of Computing Machinery (ACM), and the Society of Motion Picture and Television Engineers (SMPTE).

Sathya Narayanan (M'01) received the M.S. degree in computer science from Temple University, Philadelphia, in 1998. He is currently pursuing the Ph.D. degree at Polytechnic University, Brooklyn, NY.

Since 1999, he has been with the Panasonic Digital Networking Laboratory, Princeton, NJ. His research interests include networking protocol design, IPv6 mobility, 802.11 MAC and co-operative communications. He is active in IETF working groups related to IPv6 mobility.



Shivendra S. Panwar (S'82–M'85–SM'00) received the B.Tech. degree in electrical engineering from the Indian Institute of Technology, Kanpur, in 1981, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Massachusetts, Amherst, in 1983 and 1986, respectively.

He is a Professor in the Electrical and Computer Engineering Department at Polytechnic University. He is currently the Director of the New York State Center for Advanced Technology in Telecommunications (CATT). He spent the summer of 1987 as a Visiting Scientist at the IBM T. J. Watson Research Center, Yorktown Heights, NY, and has been a Consultant to AT&T Bell Laboratories, Holmdel, NJ. His research interests include the performance analysis and design of networks. Current work includes video systems over peer-to-peer networks, switch performance and wireless networks. He is a co-editor of two books: *Network Management and Control, Vol. II* (Plenum, 1994), and *Multimedia Communications and Video Coding* (Plenum: 1996), and co-authored the textbook *TCP/IP Essentials: A Lab-Based Approach* (Cambridge University Press, 2004).

Dr. Panwar has served as the Secretary of the Technical Affairs Council of the IEEE Communications Society (1992–1993) and is a member of the Technical Committee on Computer Communications. He is a co-recipient of the 2004 IEEE Communications Society Leonard G. Abraham Prize in the Field of Communications Systems.