The Case for Multipath Multimedia Transport over Wireless Ad Hoc Networks

Yihan Li†, Shiwen Mao‡, Shivendra S. Panwar†
†Department of Electrical and Computer Engineering
Polytechnic University, Brooklyn, NY 11201
‡the Bradley Department of Electrical and Computer Engineering
Virginia Tech, Blacksburg, VA 24061
yli@photon.poly.edu, smao@vt.edu, panwar@catt.poly.edu

Abstract

Real-time multimedia transport has stringent bandwidth, delay, and loss requirements. Supporting this application in current wireless ad hoc networks is a challenge. Such networks are characterized with frequent link failures, as well as congestion. Consequently, data packets are dropped when a link fails or congestion occurs, resulting in low received quality. In addition, a realtime multimedia service may be unavailable when a particular server is unreachable. In this article, we make the case for using multipath transport for realtime multimedia services in wireless ad hoc networks, which provides a unified solution to the above problems. We review existing work on multipath multimedia transport, and discuss the advantages, as well as related issues, of using multipath transport for realtime multimedia transport.

1 Introduction

With the recent advances in wireless technologies, wireless networks are becoming a significant part of today's access networks. Ad hoc networks are wireless mobile networks without an infrastructure, where mobile nodes cooperate with each other to find routes and relay packets. Such networks can be deployed instantly in situations where infrastructure is unavailable or difficult to install. It is maturing as a means to provide ubiquitous untethered communication. With the increase in the bandwidth of wireless channels and the computing power of mobile devices, it is expected that multimedia service will be offered over ad hoc networks in the near future.

Due to its realtime nature, realtime multimedia transport has stringent bandwidth, delay, and loss requirements. Even though some packet loss is generally tolerable, the quality of reconstructed video will be impaired and errors will propagate to consecutive frames because of the dependency introduced among adjacent frames. However, the current best-effort network architecture does not offer any quality of service (QoS) guarantees for video transport. The Transmission Control Protocol (TCP) is mainly designed for reliable data traffic. It is not suitable for realtime multimedia data because

- The delay and jitter caused by TCP retransmissions may be intolerable.
- TCP *slow-start* and *congestion avoidance* are not suitable for realtime multimedia transport.
- TCP does not support multicast.

The User Datagram Protocol (UDP), typically used in almost all realtime multimedia applications, only extends the best-effort, host-to-host IP service to the process-to-process level. When congestion occurs, an unlimited amount of UDP datagrams may be dropped since UDP is non-adaptive. Realtime multimedia applications must implement additional rate control and error control mechanisms in order to cope with network congestion.

In ad hoc networks, a wireless link have high transmission error rate because of shadowing, fading, path loss, and interference from other transmitting users.

Proceedings of the First International Conference on Broadband Networks (BROADNETS'04)
0-7695-2221-1/04 $ 20.00 IEEE

An end-to-end path found in ad hoc networks has an even higher error rate since it is the concatenation of multiple wireless links. Moreover, user mobility makes the network topology change constantly. Ad hoc networks also need to reconfigure themselves when users join or leave the network. The frequent link failures and route changes cause packet losses and reduce the received video quality. This is different from wireline networks, where packet loss is mainly caused by congestion and buffer overflow. To provide an acceptable received video quality in ad hoc networks, there should be effective error control to reduce packet losses to a certain level. Traditional error control techniques, including Forward Error Correction (FEC) and Automatic Repeat Request (ARQ), have been adapted to take link failures into consideration [23, 27, 39, 40].

In this paper, we examine the problem of using *multipath transport*, by which multiple paths are used to transfer data, for a realtime multimedia session in order to cope with the above problems, and review related issues and techniques. The rest of the paper is organized as follows. In Section 2 we present the general application scenarios, as well as the benefits of using multipath transport in realtime multimedia applications. We discuss related issues in following sections, including multipath routing in section 3, transport layer protocols in section 4, traffic partitioning techniques in section 5, and other related issues in section 6, when multipath transport is used. Section 7 concludes this article.

2 Multipath Realtime Multimedia Transport

2.1 Application Scenarios

Figure 1 illustrates the general architecture for the multipath transport of realtime multimedia data, using video as an example. At the sender, a raw video is first compressed by a video encoder. The encoder may generate a single compressed video flow, or multiple compressed video flows. In the latter case, we call it a *multistream coder*. Then the flows are partitioned and assigned to the multiple paths by a *traffic allocator*. These paths are maintained by a *multipath routing protocol*. When the flows arrive at the

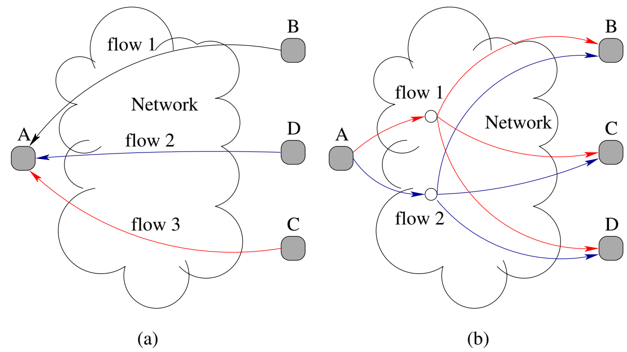

Figure 2. The architecture for using multipath transport for many-to-one and one-to-many types of applications: (a) Parallel downloading, (b) Multicasting using multiple trees.

receiver, they are first put into a *resequencing buffer* to restore the original order. Finally, the video data is extracted from the buffer to be decoded and displayed. The underlying network could be a wireline network, such as the Internet, or a wireless ad hoc network. We assume the underlying network has a mesh-structure where multiple paths exist between source-destination pairs. As will be discussed in the following, better error resilience may be achieved if the paths in such networks are maximally disjoint to each other [1, 18, 32].

The point-to-point architecture in Fig. 1 can be extended to more general cases. We call this broader class *generalized multipath transport*. For example, an architecture for the many-to-one type of applications is shown in Fig. 2(a), where a node downloads a video clip from multiple servers in parallel [2]. The server node, e.g., Node B, C, or D is a mobile node in an ad hoc network, that has the target video in its cache or public directory. A multicast-based architecture is shown in Fig. 2(b), where a source multicasts realtime multimedia data to a group of nodes using two multicast trees [36].

2.2 Advantages of using Multipath Transport

The advantages of using multipath transport in wireline and wireless networks have been reported in many previous works, e.g., see [1, 2, 8, 10, 13, 14, 18, 19, 23, 24, 27, 36, 40] and the references therein. We briefly summarize these advantages in the following.

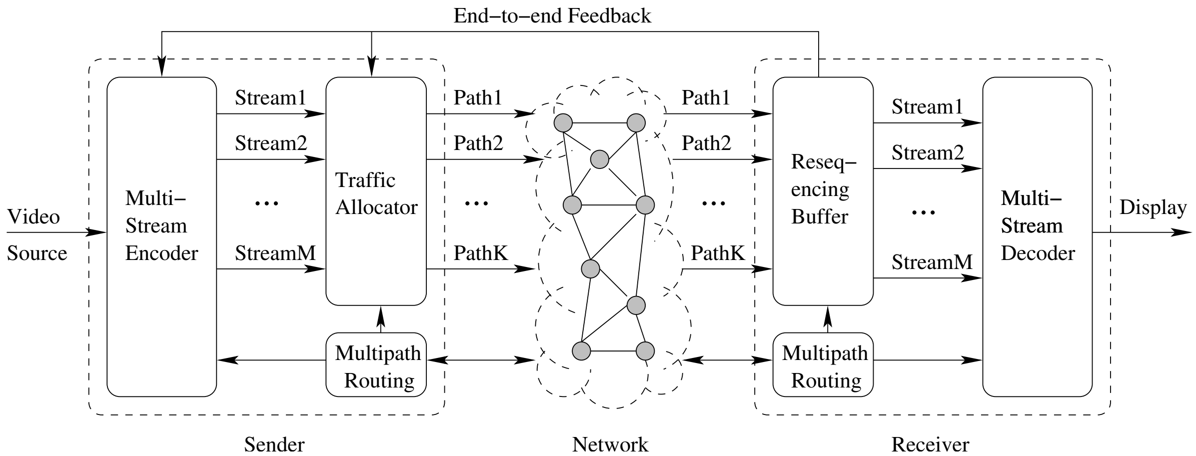

Figure 1. The general architecture for the multipath transport of realtime multimedia applications.

First, multipath transport distributes traffic load in the network more evenly. For example, a large burst of data, e.g., an Intra or I video frame, can be partitioned into several smaller bursts, each transmitted on a different path. A high rate video flow can be partitioned into several subflows, each with a lower rate and sent on a different path. Such balanced load results in less congestion inside the network [5,6]. Thus the video packet losses caused by router buffer overflow can be effectively reduced. Although a recent work [9] shows that multipath routing does not necessarily achieve better load balancing than shortest path routing in wireless ad hoc networks unless the number of paths in use is high, this result is obtained based on the assumption of extremely high node density, such that the shortest path between the source and destination node is very close to the line segment connecting these two nodes. This assumption is more valid for wireless sensor networks, but not for the typical application scenarios of wireless ad hoc networks.

Second, multipath transport provides a larger aggregate capacity for a multimedia session. In an ad hoc network, since the available link bandwidth may be limited and time varying, a high rate flow may not find enough available capacity on a single path. With multipath transport, the flow can be partitioned into several thinner subflows, each of which can be accommodated by a path.

Third, if a set of disjoint paths are used in multipath transport, losses experienced by the subflows may be independent to each other. When a path is down because of a link failure, which happens more often in an ad hoc network than in a wireline network, it is likely that some other paths are still in good condition. Thus the receiver can always receive some data during any period [1,23,40][1]. With proper error concealment schemes applied, the display will not be interrupted by link failures, although a certain degradation in the video quality will be observed. Furthermore, with path diversity, error control schemes can be designed jointly with the traffic allocator, making traditional error control schemes more effective and resulting in better error resilience. Examples of such techniques will be presented in Section 6.2.

Fourth, multipath transport facilitates load balancing for the servers. As shown in Fig. 2(a), a client can download video from multiple servers when multipath transport is used. A high rate session can be partitioned into several lower rate ones, each with a smaller server processing time. The smaller granularity of per user loads allows for more even load balancing, which can translate into either more clients supported or lower response times [2,28].

To summarize, the use of multipath transport for realtime multimedia applications in ad hoc networks can effectively reduce packet losses, provide better scalability, and provide un-interrupted display of video even

[1]Except when all the paths are down simultaneously, which we assume only occurs rarely. For example, if path i fails with probability p_i, $i = 1, 2, \cdots, N$, then the probability of simultaneous failure of all of the N paths is $p_1 \times p_2 \times \cdots \times p_N$, which is much smaller than any of the p_i's, assuming that the path failures are independent.

with the presence of frequent link failures.

3 Multipath Routing

As illustrated in Fig. 1, in order to use multipath transport, the underlying routing protocol must provide and update the multiple paths between the source and the destination node.

3.1 Types of Multipath Routing

The idea of *dispersity routing* was first presented in [24] for wireline networks. Since Maxemchuck's seminal work [24], multipath transport has been applied in various settings, e.g., load balancing, achieving a higher aggregate capacity, and path redundancy for failure recovery [13]. There are two types of multiple path routing protocols, as illustrated in Fig. 3. A set of *braided* paths is shown in Fig. 3(a), where each node maintains a backup path to the destination node [8, 26]. Fig. 3(b) shows two *node disjoint* paths, i.e., there is no common nodes between these paths, except for the source and destination nodes. A relaxed type of disjoint paths is *link disjoint* paths, where sharing of nodes, but not links, is allowed. The braided multipath routing is a relaxed version of disjoint multipath routing, since the latter may be more difficult to implement or unavailable in some network topology. However, the benefits of using multipath transport are generally maximized when disjoint paths are in use. For example, when a common node, shared by two paths, is congested or is unavailable, both paths will fail and the receiver video display will be interrupted until a new set of paths are found (see footnote 1).

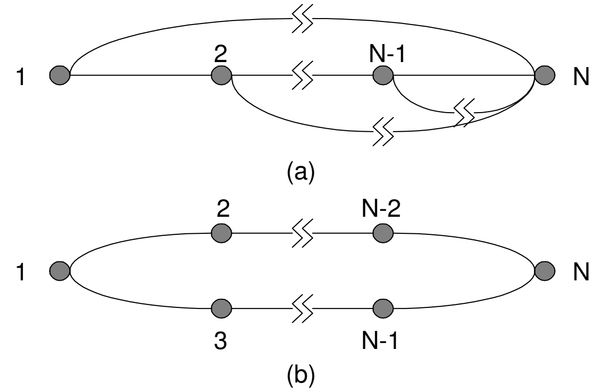

Figure 3. Two types of multipath routing: (a) Braided multiple paths, (b) Node disjoint multiple paths.

3.2 Finding Multiple Routes

In order to find multiple disjoint paths, multipath routing protocols may be used [21, 29, 38]. Many routing protocols designed for ad hoc networks are multipath routing protocols, such as the Temporally Ordered Routing Algorithm (TORA) [30], the ticket-based QoS routing scheme in [7], and the Distance Routing Effect Algorithm for Mobility (DREAM) [3].

In addition, many other protocols are potentially capable of, and can be extended to, multipath routing, such as the Dynamic Source Routing (DSR) protocol [15], Ad hoc On-demand Distance Vector routing (AODV) [33], and the Zone Routing Protocol (ZRP) [31]. For example, when a proactive routing protocol is used, a node learns the entire topology from the routing information updates. Then, it can compute the shortest path and an additional path which is most disjoint to the shortest one. On the other hand, when reactive routing protocols are used, a node learns multiple paths to a destination from the route discovery process. Then, it can choose multiple disjoint ones (e.g., a shortest one and a maximally disjoint one) from the discovered paths. DSR [34] has been extended to disjoint multipath routing in [23] and [17], where multiple maximally disjoint paths are maintained, and is extended to braided multipath routing in [26], where each intermediate node along the shortest path between the source and destination maintains one or more backup routes to the same destination.

The performance improvment achieved by multipath transport is at the cost of a slightly increased routing overhead. In the proactive routing case, the additional cost is low since nodes have learnt the topology information. It only needs to do some additional computation locally. In the reactive routing case, more route replies are transmitted than the original DSR. These additional costs in either computation or traffic load are limited, and result in better video quality [26].

The recent work in [21] proposed an efficient genetic algorithm-based multipath routing approach for multiple description (MD) video transport over wireless ad hoc networks. The multipath routing is formulated as a constrained combinatorial optimization problem, aiming to minimize the distortion of the received video. The formulated problem is shown to be NP-hard and solved by a genetic algorithm-based metaheuristic approach. Simulation results illustrate significant gain in video quality achieved by this approach over tradition network-centric multipath routing approaches, mainly due to the fact that the application layer performance metric, i.e., video distortion, is optimized. This approach was extended to support multiple concurrent video sessions in [22], and to multicasting MD video (using multiple trees) in [20].

3.3 Deploying Multiple Routes

When multiple paths are found, they can be used in several ways. If source routing is supported by the underlying network, the sender can store the entire route in the headers of multimedia data packets. Each intermediate node will simply examine the header of a received packet, and forward it to the next node. Source routing is supported both in IPv4 and IPv6, and the very popular ad hoc network routing protocol, DSR, is also based on source routing. Another way of using the multiple paths is by using the Stream Control Transmission Protocol (SCTP) [39], which has the built-in features of multi-streaming, where a flow is partitioned and transmitted as multiple streams, and multi-homing, where an SCTP endpoint can use multiple network interfaces. SCTP sockets can be used to set up multiple streams, and then the traffic allocator can assign the realtime multimedia data to the streams.

If the underlying network do not support source routing and SCTP, multipath routing can be performed via an overlay approach, which we call *application-level multipath routing*. That is, all participating nodes of the multimedia application will form an overlay network, in which each logical link may consists of one or more links in the underlying network. Multipath routing and packet forwarding can then be easily implemented in the application layer without changing the underlying network architecture and operation [4].

4 Transport Layer Protocols for Multipath Transport

There have been several new transport protocols proposed to facilitate multipath transport of multimedia data. A transport layer protocol, called meta-transmission control protocol (Meta-TCP), was presented in [11] to maintain multiple TCP connections for a data session. Meta-TCP was designed to focus on general elastic data transport using TCP. For realtime multimedia data, the Multi-flow Realtime Transport Protocol (MRTP), was presented in [19, 25] to support the general architecture using multiple paths shown in Figures 1 and 2.

There are two existing protocols that meet some, but not all, of the design goals of MRTP. The first one is Realtime Transport Protocol (RTP) and its companion control protocol Realtime Transport Control Protocol (RTCP) [37]. RTP/RTCP is a multicast-oriented protocol for Internet realtime applications. RTP/RTCP itself does not support the use of multiple flows. An application could implement multipath realtime transport using RTP, but it would have to perform all the overhead functions of managing multiple flows [10]. Compared with RTP, MRTP provides more flexible data partitioning and uses multiple paths for better queueing performance and better error resilience. Furthermore, RTP focuses on multicast applications, where feedback is suppressed to avoid feedback explosion [37]. RTP Receiver Reports (RR) or Sender Reports (SR) are sent at least 5 s apart, which may be too infrequent for the sender to react to congestion or path failures. With MRTP, since only a few routes are in use (Fig. 1 and Fig. 2(b)), it is possible to provide more timely feedback, enabling the source encoder and the traffic allocator to quickly adapt to congestion or path failures.

The other closely related protocol is SCTP [39], which we have already mentioned in the previous section. SCTP is a message-based transport layer protocol initially designed for reliable signaling in the Internet (*e.g.*, out-of-band control for Voice over IP (VoIP) call setup or teardown). It has the attractive features of multi-homing and multi-streaming, where multiple network interfaces or multiple streams can be used in a SCTP session [14]. SCTP cannot be applied directly for multimedia data because it lacks functions required for realtime services. With MRTP, the design is focused on supporting realtime applications, with timestamping and QoS feedback as its essential mode of operation. Moreover, since SCTP is a transport layer protocol and is implemented in the system kernel, it is hard, if not impossible, to make changes to it. A new multimedia application, with a new coding format or a new transport requirement, could only with difficulty be supported by SCTP. MRTP is largely an application layer protocol and is implemented in the user space as an integral part of an application. New multimedia services can be easily supported by defining new profiles and new extension headers. Indeed, MRTP is complementary to SCTP in that it supports realtime multimedia services using multiple paths. MRTP can establish multiple paths by using SCTP sockets, taking advantage of the multi-homing and the multi-streaming features of SCTP.

MRTP/MRTCP, presented in [19, 25], is a natural extension of RTP/RTCP, exploiting path diversity to combat frequent congestion and link failures. Although such functionalities provided by MRTP/MRTCP can be implemented in the application layer by each application independently, it would be valuable to abstract and package the realtime multipath transport-related functions as a single generic protocol that can be used by many different applications, and thus relieve multimedia applications of such burdens. MRTP is a transport protocol usually implemented in the application layer. Given multiple paths maintained by an underlying multipath routing protocol, MRTP and its companion control protocol, the Multi-flow Realtime Transport Control Protocol (MRTCP), provide essential support for multiple path realtime transport, including session and flow management, data partitioning, traffic dispersion, timestamping, sequence numbering, and Quality of Service (QoS) feedback.

The protocol stack with MRTP is shown in Fig. 4. MRTP uses the UDP datagram service or the multi-homing/multi-streaming transport service of SCTP for data and control. Note that the session/flow management function can also be performed using the Session Initiation Protocol (SIP) [35] over TCP. An underlying multipath routing protocol maintains multiple paths from the source to the destination. When SCTP is used in the transport layer, SCTP sockets can be used to set up multiple flows.

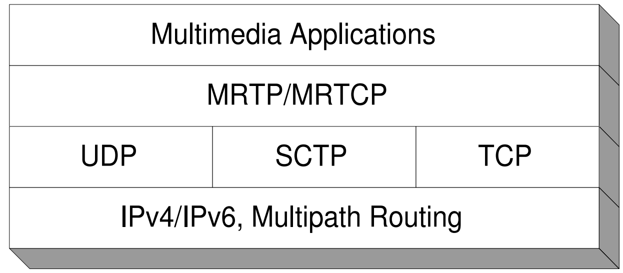

Figure 4. The protocol stack using MRTP.

Unlike RTP, MRTP is a session-oriented protocol. An MRTP session should be established first by MRTCP, where two end nodes exchange information such as available paths, session/flow IDs, and initial sequence numbers. During data transmission, a new flow may be added to the session when a better path is found, and a stale flow may be removed from the session based on QoS reports.

With MRTP, a traffic allocator partitions and disperses the realtime multimedia traffic to multiple flows. A basic traffic partitioning and dispersion scheme is provided in MRTP, which assigns the packets to the multiple flows using the *round-robin* algorithm. This simple assignment may not be optimal for some applications and can be overridden in such situations.

As in RTP, MRTP generates QoS reports periodically. An MRTP SR or RR carries both the per-flow statistics and session statistics. Unlike RTP, the MRTP SR and RR can be sent at an interval set by the application. For point-to-point and parallel downloading applications (see Fig.1 and Fig.2(a)), RR and SR could be sent for each frame since the number of participants are relatively small. Timely QoS reports enable the sender to quickly adapt to transmission errors. For example, the encoder can change the coding parameters or encoding mode for the next frame, introducing more (or less) redundancy for error resilience, or the traffic allocator can avoid the use of a stale path and disperse packets to other better paths.

5 Multimedia Traffic Partitioning

As discussed in Section II, realtime multimedia traffic needs to be partitioned into multiple flows in order to be transmitted over multiple paths. Such a traffic partitioning can be performed in different layers [10]. For stored media, traffic partition schemes such as thinning can be performed, while for live media applications, a multistream encoder can produce multiple streams on the fly. We discuss these schemes in this section.

5.1 Traffic Partitioning

As illustrated in Fig. 1, on the sender side, the traffic allocator is responsible for partitioning the application data, i.e., dispatching application data packets to the multiple paths in use. The traffic partitioning strategy is affected by a number of factors, such as the auto-correlation structure of the application data flow, the number of available paths, and the QoS parameters of the paths (e.g., the bandwidth, delay, and loss characteristics of each path). Usually the path parameters can be inferred from feedback, so that the traffic allocator can adjust its strategy to adapt to changes in the network [18].

For stored video, a partitioning technique called block-based traffic *thinning* can be used [19]. With block-based thinning, a video sequence is first divided into equal-sized blocks of length B. From the application's perspective, the blocks consist of a number of video frames or audio frames or some other application-specific temporal payload units. Then, in the simplest case, the blocks are assigned to the paths using the *round robin* scheme.

In addition, multimedia traffic partitioning can also be performed by a multistream coder, especially in interactive multimedia applications, as discussed in the following section.

5.2 Multistream Video Coding

The multistream coder should be carefully designed to generate substreams so that the loss in one substream does not adversely affect the decoding of other substreams. Furthermore, this relative independence between the substreams should not be obtained at the expense of a significant decrease in coding efficiency. Therefore, the multistream encoder should aim to achieve a good trade-off between coding efficiency and error resilience.

Obviously, one way to generate multiple substreams is to use a standard video codec and split the resulting bitstream into multiple substreams. An intelligent splitting scheme is needed to split the bit stream at the boundary of independently decodable units, otherwise a lost substream will make the received ones from other paths useless. A simple way to accomplish this is to send the frames to the paths in a round robin manner, e.g., all odd frames are sent to path 1 and all even frames are sent to path 2. In order to completely avoid the dependency between sub-streams, the frames sent on one path should be predictively coded with respect only to the frames on the same path. This method is in fact an option available in the H.263+ standard (Video Redundancy Coding (VRC)) [41]. However, compared to predicting a frame from its immediate neighbor, VRC requires significantly higher bit rates. Also, although this method can prevent the loss in one path from affecting frames in the other path, error propagation still exists within frames in the same path. In [23] [18], a feedback based reference picture selection method was presented to circumvent these two problems.

Another natural way of generating multiple streams is by using *layered video coding*, which is very useful in coping with the heterogeneity of user access rates, in network link capacities, and in link reliability. A layered coder encodes video into several layers. The base layer (BL), which includes the crucial part of the video frames, guarantees a basic display quality. Each enhancement layer (EL) correctly received improves the video quality. But without the BL, video frames cannot be reconstructed sufficiently. Usually, EL packets may be dropped at a congested node to protect BL packets, and BL packets are better protected with FEC or ARQ [16]. When combined with MPT, it is desirable to transmit the BL substream on the best route. The source may sort the paths according to their loss characteristics, inferred from QoS feedback (e.g., Receiver Report in RTP/RTCP [37]). Alternatively, the multipath routing layer may organize the route cache according to some performance metrics (number of hops, mean loss rate in the last time window, etc.).

Instead of generating substreams that are unequal in their importance, *Multiple Description Coding* (MDC) generates multiple equally important streams, each giving a low but acceptable quality. A high-quality reconstruction is decodable from all bit streams together, while a lower, but still acceptable quality reconstruction is achievable if only one stream is received. The correlation among the substreams introduced at the encoder makes it possible to partially recover lost information of one substream, using information carried in other correctly received substreams. However, such a correlation limits the achievable coding efficiency, as compared to a conventional coder designed to maximize it. An excellent review of the theoretical bounds and proposed MDC algorithms can be found in [12]. In designing a MCP-based MD video codec, a key challenge is how to control the mismatch between the reference frames used in the encoder and those used in the decoder caused by transmission errors. With MDC, the transport layer design can be simpler than with layered coding. Because all the descriptions are equally important, the transport layer does not need to protect one stream more than another. Also, because each description alone can provide a low but acceptable quality, no retransmission is required, making MDC more suitable for applications with stringent delay requirements.

In [23] [18], three representative video coding schemes from the above three classes of coding are chosen and adapted to multipath transport for video transport over wireless mobile ad hoc networks. Impressive gains in the received video quality obtained by using these schemes were presented.

6 Other Related Issues

In this section, we discuss two related issues on multipath multimedia transport in wireless ad hoc networks, which need to be carefully addressed in order to achieve the best performance of using multiple paths.

6.1 Resequencing Buffer and Delay

One major concern when using multipath transport is the additional resequencing delay. Since packets sent on different paths suffer different delays, they may arrive at the receiver out of order. The receiver needs to use a resequencing buffer to temporarily store the received packets and put them in order. In reliable transport protocols for data traffic, e.g., TCP, a packet may stay in the resequencing buffer for a long time waiting for a missing packet with a smaller sequence number. In realtime multimedia applications, the resequencing buffer is mainly used to absorb jitter in arriving packets. Since the receiver displays the video or audio continuously, each packet is associated with a deadline D_l, which is the difference between the time when it is extracted from the resequencing buffer to be decoded and played out, and the time when it was transmitted by the sender. A packet may be lost either because of transmission errors or because it is overdue. Such deadlines impose a smaller time window for multimedia transport and limit the efficacy of traditional error control schemes, such as ARQ.

Traffic partitioning and resequencing delay are closely related to each other. In [18], the optimal traffic partitioning problem was investigated for realtime applications using network calculus in a deterministic setting. The bottleneck link of each path was modelled as a queue with a deterministic service rate. The contribution of all other links and the propagation delay are lumped into a fixed delay element. Moreover, with the assumption that the source flow is regulated by a $\{\sigma, \rho\}$ leaky bucket, traffic can be split into multiple flows, each conforming to a $\{\sigma_i, \rho_i\}$ regulator, by using *deterministic* traffic partitioning. Under these assumptions, a constrained optimization problem on minimizing the total end-to-end delay was formulated. The simplicity of the model results in a compact formulation. A closed-form solution was derived and simple guidelines on minimizing end-to-end delay and path selection was provided. The path set chosen using the analysis is optimal in the sense that it is *the minimum set of paths required to achieve the minimum delay*; adding any rejected path to the set will only increase the total end-to-end delay.

This path selection scheme is useful, since although it is always desirable to use a path with a higher bandwidth and a lower fixed delay, it is impossible to order the paths *consistently* according to their bandwidth or fixed delay in many cases. A brute force optimization testing all the feasible combinations of the paths would have exponential complexity [40]. Using this analysis, the performance metrics can be translated to the end-to-end delay and the set of paths can be easily determined with $\mathcal{O}(N)$ complexity, where N is the number of paths available. Thus this algorithm is suitable for the cases where the paths are highly dynamic. The *exact* optimal partitioning, rather than a heuristic, can be quickly computed and applied to a snapshot of the varying network. We also present a design to enforce the optimal partitioning using a number of cascaded leaky buckets, one for each path.

6.2 Error Control

To combat transmission losses, redundancy can be introduced into the streams by the traffic allocator (see Fig. 1). One extreme of doing so is just partitioning the original stream with no redundancy introduced. When packets are lost, the receiver may copy the corresponding blocks in the previous frame to conceal the error. The other extreme is transmitting multiple copies of each original packet. If feedback is available, the amount of redundancy introduced can be adaptive to the congestion or losses in the network. A good strategy should be positioned between these two extremes and provide a good trade-off between error resilience and bandwidth requirements.

Multipath transport makes the traditional error control schemes more effective. One of the most common FEC codes are Reed-Solomon (RS) codes. $RS(n, k)$ codes consist of k source packets and $n - k$ redundant packets. The reception of any k packets from the n transmitted ones allows the reconstruction of all the k source packets. If the loss characteristics of the paths are known, e.g., estimated from feedbacks, the traffic allocator can assign the n packets to the paths in a way that maximizes the probability that at least k, out of n, packets are correctly received [27, 40].

When ARQ is used, a retransmitted packet may be assigned to a different path [23]. Since a lost packet on a path indicates either congestion or link failure in that path, sending the retransmitted packet on the same path only intensifies the congestion, and the retransmitted packet may well be dropped again. Path diversity implies that the losses or congestion periods of the paths are independent. Thus the success probability is higher if a lost packet is retransmitted on a different path [23, 39].

7 Conclusions

In this paper, we review the case for using multipath transport for realtime multimedia applications in wireless ad hoc networks. Multipath transport can reduce congestion in the network, as well as at the servers. In addition, path diversity enables effective error control, resulting in stronger error resilience. These benefits come at the cost of a limited increase in computational complexity and traffic load. The literature on related issues of multipath multimedia transport, including multipath routing, multipath transportation layer protocols, multimedia traffic partitioning, resequencing and error control, is reviewed.

References

[1] J. G. Apostolopoulos. Reliable video communication over lossy packet networks using multiple state encoding and path diversity. In *Proc. SPIE VCIP*, Jan. 2001.

[2] J. G. Apostolopoulos, T. Wong, W. Tan, and S. Wee. On multiple description streaming with content delivery networks. In *Proc. IEEE INFOCOM*, June 2002.

[3] S. Basagni, V. R. S. I. Chlamtac, and B. A. .Woodward. A distance routing effect algorithm for mobility (dream). In *Proc. MobiCom98*, pages 76–84, 1998.

[4] A. C. Begen, Y. Altunbasak, and O. Ergun. Multipath selection for multiple description encoded video streaming. In *Proc. IEEE ICC*, pages 1583–1589, May 2003.

[5] D. Bushmitch. *Queueing performance of realtime traffic under thinning, striping, and shuffling*. PhD thesis, Polytechnic University, Brooklyn, NY, January 2004.

[6] D. Bushmitch, S. S. Panwar, and A. Pal. Thinning, striping and shuffling: Traffic shaping and transport techniques for variable bit rate video. In *Proc. IEEE GLOBECOM 2002*, pages 1485–1491, November 1975.

[7] S. Chen and K. Nahrstedt. Distributed quality-of-service routing in ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 17(8):1488–1505, August 1999.

[8] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin. Highly-resilient, energy-efficient multipath routing in wireless sensor networks. *ACM Mobile Computing and Communications Review*, 1(2):1–13, Feb. 2002.

[9] Y. Ganjali and A. Keshavarzian. Load balancing in ad hoc networks: Single-path routing vs. multi-path routing. In *Proc. IEEE INFOCOM 2004*, pages 352–355, March 2004.

[10] 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 and Systems for Video Technology*, 12(9):777–792, September 2002.

[11] N. Gogate and S. S. Panwar. Supporting applications in a mobile multihop radio enviroment using route divesity: I. nonrealtime data. In *Proc. IEEE ICC*, pages 802–806, June 1998.

[12] V. K. Goyal. Multiple description coding: compression meets the network. *IEEE Sigal Processing Mag.*, 18:74–93, Sept. 2001.

[13] E. Gustafsson and G. Karlsson. A literature survey on traffic dispersion. *IEEE Network*, 8(5):28–36, March/April 1997.

[14] H.-Y. Hsieh and R. Sivakumar. A transport layer approach for achieving aggregate bandwidths on multihomed mobile hosts. In *Proc. ACM Inter. Conf. Mob. Comp. Networking*, pages 83–95, September 2002.

[15] D. B. Johnson, D. A. Maltz, and Y.-C. Hu. The dynamic source routing protocol for mobile ad hoc networks (dsr), April 2003. IETF Internet Draft, draft-ietf-manet-dsr-09.txt.

[16] M. Khansari, A. Jalali, E. Dubois, and P. Mermelstein. Low bit-rate video transmission over fading channels for wireless microcellular systems. *IEEE Trans. Circuit Syst. Video Technol.*, 6(1):1–11, Feb. 1996.

[17] S.-J. Lee and M. Gerla. Split multipath routing with maximally disjoint paths in ad hoc networks. In *Proc. IEEE ICC*, pages 3201–3205, June 2001.

[18] S. Mao. *Multimedia transport using multiple paths*. PhD thesis, Polytechnic University, Brooklyn, NY, January 2004.

[19] S. Mao, D. Bushmitch, S. Narayanan, and S. S. Panwar. Mrtp: A multi-flow realtime transport protocol for ad hoc networks. In *Proc. IEEE VTC 2003-Fall*, pages 1736–1745, Oct. 2003.

[20] S. Mao, X. Cheng, Y. T. Hou, and H. D. Sherali. Multiple description video multicast in wireless ad hoc networks. In *Proc. IEEE/ACM BroadNets*, October 2004.

[21] S. Mao, Y. Hou, X. Cheng, H. D. Sherali, and S. F. Midkiff. Multi-path routing for multiple description video over wireless ad hoc networks. Technical report, the Bradley Dept. of ECE, Virginia Tech, April 2004.

[22] S. Mao, Y. Hou, X. Cheng, H. D. Sherali, and S. F. Midkiff. Optimal routing for multiple concurrent video sessions in wireless ad hoc networks. Technical report, the Bradley Dept. of ECE, Virginia Tech, April 2004.

[23] S. Mao, S. Lin, S. S. Panwar, Y. Wang, and E. Celebi. Video transport over ad hoc networks: Multistream coding with multipath transport. *IEEE Journal on Selected Areas in Communications*, 21(10):1721–1737, Dec. 2003.

[24] N. F. Maxemchuck. Dispersity routing. In *Proc. IEEE ICC*, pages 10–41, June 1975.

[25] S. Narayanan, D. Bushmitch, S. Mao, and S. S. Panwar. Mrtp: A multi-flow real-tim transport protocol, August 2004. work in progress, IETF Internet Draft draft-narayanan-mrtp-00.txt.

[26] A. Nasipuri and S. R. Das. On-demand multipath routing for mobile ad hoc networks. In *Proc. IEEE ICCCN*, pages 64–70, 1999.

[27] T. Nguyen and A. Zakhor. Path diversity with forward error correction (pdf) system for packet switched networks. In *Proc. IEEE INFOCOM*, April 2003.

[28] V. N. Padmanabhan, H. J. Wang, P. A. Chou, and K. Sripanidkulchai. Distributing streaming media content using cooperative networking. In *Proc. ACM NOSSDAV*, May 2002.

[29] P. Papadimitratos, Z. Haas, and E. Sirer. Path set selection in mobile ad hoc networks. In *Proc. ACM Mobihoc 2002*, May 2002.

[30] V. D. Park and M. S. Corson. A highly adaptive distributed routing algorithm for mobile wireless networks. In *Proc. IEEE/ACM INFOCOM97*, 1997.

[31] M. R. Pearlman and Z. J. Haas. Determining the optimal configuration for the zone routing protocol. *IEEE Journal on Selected Areas in Communications*, 17(8):1395–1414, August 1999.

[32] M. R. Pearlman, Z. J. Haas, P. Scholander, and S. S. Tabrizi. On the impact of alternate path routing for load balancing in mobile ad hoc networks. In *ACM MobiHOC'2000*, August 2003.

[33] C. Perkins, E. Belding-Royer, and S. Das. Ad hoc on-demand distance vector (aodv) routing, July 2003. IETF RFC 3561.

[34] C. E. Perkins, editor. *Ad Hoc Networks*. Addison-Wesley, Reading, MA, 2001.

[35] J. Rosenberg and et al. Sip: Session initiation protocol, June 2002. IETF RFC 3261.

[36] S. Sajama and Z. Haas. Independent-tree ad hoc multicast routing (itamar). *ACM MONET journal*, 8(5):551–556, October 2004.

[37] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. Rtp: A transport protocol for realtime applications, July 2003. IETF RFC 3550.

[38] D. Sidhu, R. Nair, and S. Abdallah. Finding disjoint paths in networks. In *Proc. ACM SIGCOMM*, pages 43–51, Sept. 1991.

[39] R. R. Stewart and Q. Xie, editors. *Stream Control Transmission Protocol: A Reference Guide*. Addison-Wesley, Reading, MA, 2001.

[40] A. Tsirigos and Z. Haas. Analysis of multipath routing - part i: The effect on the packet delivery ratio. *IEEE Transactions on Wireless Communications*, 3(1):138–146, January 2004.

[41] S. Wenger. Video redundancy coding in h.263+. In *Proc. Workshop Audio-Visual Services for Packet Networks*, Sept. 1997.