A Multipath Video Streaming Testbed for Ad Hoc Networks

Shiwen Mao, Shunan Lin, Shivendra S. Panwar, and Yao Wang
Department of ECE, Polytechnic University, Brooklyn, NY 11201

Abstract— **Enabling video transport over ad hoc networks is more challenging than over other wireless networks. The wireless links in an ad hoc network are highly error prone and can go down frequently because of node mobility, interference, channel fading, and the lack of infrastructure. However, the mesh topology of ad hoc networks implies the existence of multiple paths between two nodes. Indeed, multipath transport provides an extra degree of freedom in designing error resilient video coding and transport schemes.**

In our previous work, we propose schemes combining multistream coding with multipath transport, to show that path diversity provides an effective means to combat transmission error in ad hoc networks. In this paper, we report the implementation of an multiple path video streaming testbed using notebook computers and IEEE 802.11b cards, to validate the viability and performance advantages of these schemes. We implemented a layered coding with selective ARQ scheme and a multiple description motion compensation coding scheme in the testbed. The experimental results show that video transport is viable in ad hoc networks given careful cross-layer design. Combining multistream coding with MPT improves video quality, as compared to traditional schemes where a single path is used.

I. Introduction

Ad hoc networks are multi-hop wireless networks without a pre-installed infrastructure. They can be deployed instantly in situations where infrastructure is unavailable (e.g., disaster recovery), or where infrastructure is difficult to install (e.g., battlefields). It is maturing as a means to provide ubiquitous untethered communication. With the increase both in the bandwidth of wireless channels and in the computing power of mobile devices, it is expected that video service will be offered over ad hoc networks in the near future. Examples of the need for video service in such networks are: (1) In disaster recovery, e.g., firefighters carrying cameras and wireless transceivers can track each other's whereabouts and send the fire scene back to the command center; (2) A sensor network can be deployed to monitor wildlife in an inaccessible region, where live video is captured and relayed back to the base by the sensors scattered in the region.

Ad hoc networks pose a great challenge to video transport. There is no fixed infrastructure and the topology is frequently changing due to node mobility. Therefore, links are continuously established and broken. The availability and quality of a link further fluctuates due to channel fading and interference from other transmitting users. In addition, an end-to-end path consists of a number of wireless links. Thus transmission loss in ad hoc networks is more frequent than that in wireless networks with single hop wireless paths connecting nodes to the wireline infrastructure. In the data link layer, the most popular Media Access Control (MAC) scheme, the Carrier Sensing Multiple Access/Collision Avoidance (CSMA/CA) scheme [1], is designed for best-effort data. It provides no hard guarantees for a session's bandwidth and delay. Although bandwidth reservation is possible with MAC schemes based on Time Division Multiple Access (TDMA) or Code Division Multiple Access (CDMA), practical implementations of these schemes are non-trivial because of the synchronization or code assignment problems when node mobility is allowed [2].

Video transport typically requires stringent bandwidth and delay guarantees. However, it is very hard to maintain an end-to-end route which is both stable and has enough bandwidth in an ad hoc network. Furthermore, compressed video is susceptible to transmission errors. For example, a single bit error often causes a loss of synchronization when Variable Length Coding (VLC) is used in the multiplexed video bit-streams. Moreover, the motion compensated prediction (MCP) technique is widely used in modern video coding standards. In MCP, a frame is first predicted from a previous coded frame (called reference picture) and then the prediction error is encoded and transmitted. Although MCP achieves high coding efficiency by exploiting the temporal correlation between adjacent frames, it makes the reconstruction of a frame depend on the successful reconstruction of its reference picture. Without effective error protection and concealment, a lost packet in a frame can cause not only error within this frame, but also errors in many following frames, even when all the following frames are correctly received [3].

Given the error-prone nature of ad hoc network paths and the susceptibility of compressed video to transmission errors, effective error control is needed. Traditional techniques, including Forward Error Correction (FEC) and Automatic Repeat Request (ARQ), must be adapted to take into consideration of the delay constraint and the error propagation problem [4]. In ad hoc networks, wireless links break down the traditional concept of topology, which is not constrained by physical cable connections anymore. Although user mobility makes links volatile, it provides variability of topology. On the one hand, a link may break when nodes move away from each other. On the other hand, it is possible to quickly find new routes formed in a new topology. Furthermore, the mesh topology of ad hoc networks implies the existence of multiple routes between two nodes. Given multiple paths, a video stream can be divided into multiple substreams and each substream is sent on one of the paths. If these paths are

0-7803-7954-3/03/$17.00

disjoint, the losses experienced by the substreams would be relatively independent. Better error resilience can be achieved when traffic dispersion is performed appropriately and with effective error control for the substreams. In a manner similar to multi-antenna diversity that improves the capacity of wireless networks, path diversity can also be exploited to improve the capacity of ad hoc networks. Indeed, multipath transport (MPT) provides an extra degree of freedom in designing video coding and transport schemes.

In our previous work [5]-[8], we proposed to combine multistream video coding and MPT for enabling video transport over ad hoc networks. Some of the schemes we examined are:

1) A layered coding (LC) with selective ARQ scheme (LC with ARQ) [6];
2) A multiple description motion compensation coding scheme (MDMC) [7].

We studied the performance of the proposed schemes using Markov link models [9] and OPNET Modeler [10]. To further validate the viability and performance advantages of these schemes, we implemented a video streaming testbed using notebook computers with IEEE 802.11b cards. The LC with ARQ scheme and the MDMC scheme are implemented. Two node-disjoint paths (each has two hops) are used for a video session. The video codecs closely interact with the video transport layer to exploit the benefits of path diversity. The results of our experiments show that satisfactory video quality is achievable in an ad hoc network given careful cross-layer design. Combining multistream coding with MPT improves video quality, as compared to traditional schemes where a single path is used.

Theoretical model of ad hoc networks is still an open research problem. On the other hand, a testbed, built using off-the-shelf devices, is very convenient for extensive testing of new algorithms under a realistic setting. A number of ad hoc testbeds have been built recently [11][12]. These mainly focus on the performance of ad hoc routing protocols, physical layer characteristics, scalability issues, and integration of ad hoc networks with the Internet for data transport. In [13], a firewall is inserted between the source and destination, which drops video packets according to a Markov channel model [9]. So far as we know, the testbed we developed is the first effort in combining multistream video coding and MPT for video transport in ad hoc networks.

The remainder of the paper is organized as follows. In Section II, the video coding/transport schemes implemented in the testbed are presented. Section III presents the video codec implementation and the testbed setup. Experimental results are given in Section IV. Section V conclusions the paper.

II. The Video Coding and Transport Schemes

A. Layered Coding with Selective ARQ

This is a scheme using layered video coding. With this scheme, a video sequence is coded into two layers: a BL and an EL. We follow the SNR profile in the H.263 standard [14] when generating the layers. A BL frame is encoded using the standard predictive video coding technique. Note that because the BL coding uses only the previous BL picture for prediction, this coding method has a lower coding efficiency than a standard single layer coder. This loss in coding efficiency is, however, justified by increased error resilience: a lost EL packet will not affect the BL pictures. Good quality is thus guaranteed if the BL packets are delivered error-free or at a very low loss rate.

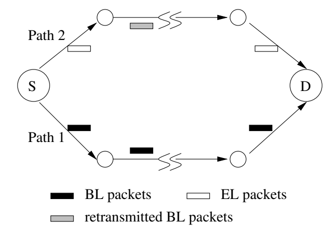


Fig. 1. The two-path layered video transmission model with end-to-end ARQ for BL packets.

Given two paths, a traffic allocator sends the BL packets on one path (the better path in terms of loss probability when the two paths are asymmetric) and the EL packets on the other path. The receiver sends selective ARQ requests to the sender for lost BL packets. To increase the reliability of the feedback, ARQ requests are sent on both paths. When the sender receives an ARQ request, it retransmits the requested BL packet on the EL path, as illustrated in Fig.1. The transmission bit rate for the EL may vary with the bit rate spent on BL retransmission. For video streaming applications where video is typically pre-encoded off-line, a simple rate control method is used for the EL path: when a BL packet is retransmitted on the EL path, one or more EL packets are dropped to satisfy the target transmission rate on the EL path.

We assume the sender continuously estimates the states of the paths based on received ARQ requests or QoS reports [16]. For example, a burst of ARQ requests received at the sender implies that the BL path is in a "bad" state. If the inferred EL path state is better, the sender may switch the paths.

B. Multiple Description Motion Compensation

Unlike LC with ARQ, MDMC is a multiple description coding scheme which does not depend on the availability of feedback channels. Because paths in ad hoc networks change between "up" and "down" state very often, each description experiences bursty packet losses. Therefore, we employ the *packet-loss mode* of MDMC presented in [10]. It uses a linear superposition of two predictions from two previously coded frames. In the MDMC encoder, the central prediction is obtained by

$$\hat{\psi}(n) = a_1\tilde{\psi}_e(n-1) + (1-a_1)\tilde{\psi}_e(n-2), \qquad (1)$$

where $\tilde{\psi}_e(n-1)$ and $\tilde{\psi}_e(n-2)$ are motion compensated predicted signals constructed from two previously encoded frames $\psi_e(n-1)$ and $\psi_e(n-2)$ respectively. The central prediction error $e_0(n) = \psi(n) - \hat{\psi}(n)$ is quantized by quantizer $Q_0(\cdot)$ to $\tilde{e_0}(n)$. The quantized prediction error and motion vectors for even frames are sent on one path, and those for odd frames are sent on another path. In the decoder, if frame $n-1$ is received, frame n is reconstructed using

$$\psi_d(n) = a_1\tilde{\psi}_d(n-1) + (1-a_1)\tilde{\psi}_d(n-2) + \tilde{e}_0(n). \quad (2)$$

where $\tilde{\psi}_d(n)$ represents motion compensated prediction from decoded frame n.

If frame $n-1$ is damaged but frame $n-2$ is received, the decoder only uses the reconstructed frame $n-2$ for prediction. To circumvent the mismatch between the predicted frames used in the encoder and the decoder, the signal $e_1(n) = \tilde{\psi}_e(n-2) - a_1\tilde{\psi}_e(n-1) - (1-a_1)\tilde{\psi}_e(n-2) - \tilde{e}_0(n)$ is quantized by another quantizer $Q_1(\cdot)$, which is typically coarser than $Q_0(\cdot)$, and the output $\tilde{e}_1(n)$ is sent along with other information on frame n. Now when frame $n-1$ is damaged, the side decoder reconstructs frame n using

$$\psi_d(n) = \tilde{\psi}_d(n-2) + \tilde{e}_0(n) + \tilde{e}_1(n). \quad (3)$$

In addition, the lost frame $\psi(n-1)$ is estimated using

$$\tilde{\psi}_d(n-1) = \frac{1}{a_1}\left(\psi_d(n) - (1-a_1)\tilde{\psi}_d(n-2) - \tilde{e}_0(n)\right). \quad (4)$$

The MDMC codec offers a trade-off between redundancy and distortion over a wide range by varying the coder parameters (the predictor coefficient a_1 and the quantization parameter of $Q_1(.)$). No feedback is needed for this scheme.

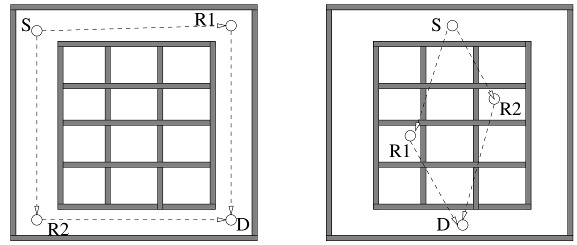
III. Testbed Implementation

A. Video Codec Implementations and Parameters

We implemented in software the LC with ARQ and the MDMC video codec on top of the public domain H.263+ codec [15]. For LC with ARQ, we added a simple rate control algorithm for EL for video streaming applications, where video is typically pre-encoded off-line. When a BL packet is retransmitted on the EL path, one or more EL packets are dropped to satisfy the target transmission rate on the EL path. For MDMC, the codec was modified to produce both central and side predictions in the INTER mode, and encodes both central and side prediction errors using quantization parameters QP_0 and QP_1 respectively.

Error concealment is performed in the decoders when packets are lost. In the LC decoder, if the BL is lost, the *copy-from-previous-frame* method is used. If the EL is lost but the BL is received, the frame is reconstructed using the BL only. In the MDMC decoder, the lost information can be recovered partially from the other received description [7].

We use the Quarter Common Intermediate Format (QCIF, 176×144 Y pixels/frame, 88×72 Cb/Cr pixels/frame) sequence "Foreman" (first 200 frames from the original 30 fps sequence) encoded at 10 fps in the testbed experiments. The encoder generates two substreams with a bit rate of 59Kbps each. In MDMC, h_1 is set as 0.9, and the quantization parameter (QP_0,QP_1) is fixed at (8,15) [7], which achieves approximately the above bit rates for each substream. A 5% macroblock level intra-refreshment is used, which has been found to be effective in suppressing error propagation for the range of the packet loss rates considered. Each group of blocks (GOB) is packetized into a single packet, to make each packet independently decodable.



(a) Line-of-sight (b) Behind the walls

Fig. 2. Experiment scenarios for the testbed

B. The Testbed Setup

The testbed consists of four IBM Thinkpad notebooks equipped with 802.11b cards. Fig.2 shows the topology of the testbed. The notebooks were placed at (or moved around) various locations in the Library/CATT building (about 30m$\times$60m) at Polytechnic University. IBM High Rate Wireless LAN cards are used working in the DCF mode with a channel bandwidth of 11Mbps. In the building, there is interference from IEEE 802.11 access points (AP) and other electronic devices (e.g., microwave ovens). Nodes S and D are, respectively, the video source and receiver, while nodes R_1 and R_2 are the relays. Since there are only four nodes in this network, we use static routing to force the use of two-hop routes. Dynamic routing will be implemented in a future version.

The system is built on Microsoft Windows 2000. Fig.3 shows the nodal architecture of nodes S and D. We implemented the timestamping, sequence numbering, and QoS feedback functions in the application layer [16]. For LC with ARQ, limited end-to-end retransmission for BL is implemented in the application layer as well. UDP sockets are used at the transport layer. A traffic allocator at the sender dynamically allocates packets to the two paths. Source routing, where each packet carries the end-to-end route in its header, is used to route the packets to the video receiver. The video receiver maintains a playout buffer, using a hash table data structure. Typically video streaming applications use a playout buffer of a few seconds to smooth the jitter in incoming packets. We chose a playout buffer of 2 seconds for this network. To support interactive applications, we also experimented with a 300ms playout delay. Note that the playout buffer also resequences the packets of each substream. For both the LC with ARQ and the MDMC schemes, the testbed performs off-line encoding, i.e., the video transmitted is pre-encoded. But the received video frames, possibly impaired by packet losses, are decoded in real time and displayed on the screen of node D. Statistics about the substreams (e.g., loss rates, jitter, and bit rates) are collected at D and sent to S. These can be used to infer the path conditions at the sender.

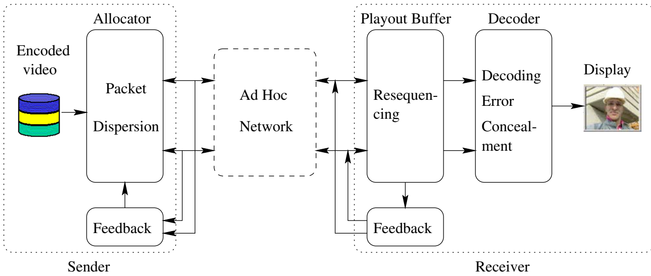


Fig. 3. The nodal architecture of the testbed.

The two relay nodes forward downstream video packets to D and upstream feedback packets to S. We also implemented a packet dropper in the relay nodes, which can be set to drop packets according to the following theoretical models: (1) the random packet loss model, where each packet is dropped with a probability P_d; (2) the burst packet loss model [9], where a two-state Markov chain, with states "good" and "bad", is maintained at the relay, and the "bad" state has a higher packet loss probability (P_b) than the "good" state (P_g). Packets can also be dropped manually by selecting the "Drop Packets" checkbox on the relayer GUI. These are very useful to force high packet loss rates when the link quality is very good.

IV. Experimental Results

We examined the performance of the testbed in the scenarios shown in Fig.2. Figure 4 is a screenshot at D of the testbed captured during a MDMC experiment. The upper left part of the GUI displays the received and decoded video, as well as the network view of the testbed. The transport related statistics (loss rates, jitter, receiver buffer occupancy, etc.) and the video codec related attributes (frame rate, bit rates, format, etc.) are displayed on the upper right part of the GUI. The two windows in the center display the packet loss traces of the two substreams for each frame. The lower part of Fig.4 is the PSNR trace of the received video, which illustrates how video quality is impaired by packet losses of both substreams, and how the MDMC decoder recovers from the packet losses.

Figure 5 shows the PSNR trace of the received video frames obtained from a LC with ARQ testbed experiment. The packet loss traces for the two paths are also plotted using the right y axis. Note that the two lost BL packets are successfully recovered by ARQ, resulting in a very stable PSNR curve as compared with that of MDMC in Fig.4.

The average PSNRs of the received frames using the two schemes are presented in Table I and Table II, respectively. Each PSNR value in the tables is the average over an experiment lasting for 10 to 15 minutes. In all the LC with ARQ experiments, the better path is used by the BL.

In the experiments, LC with ARQ performs better than MDMC, except the very low loss cases. This is consistent with the performance of the schemes with Markov channel models and OPNET Modeler [8]. However, LC with ARQ achieves the better performance by using a feedback channel and unequal protection of the BL packets. When the delay constraint is tight and the traffic load is high, or when feedback is infeasible (e.g., multicast applications), LC with ARQ may have worse performance. In these situations, MDMC is suitable since no feedback is needed. For the test scenario in Fig.2(a), when the loss rates for both substreams are very low, MDMC has a higher average PSNR (33.11 dB in Table I) than LC with ARQ (32.24 dB in Table II). This also confirms the simulation results using Markov channel models in [8].

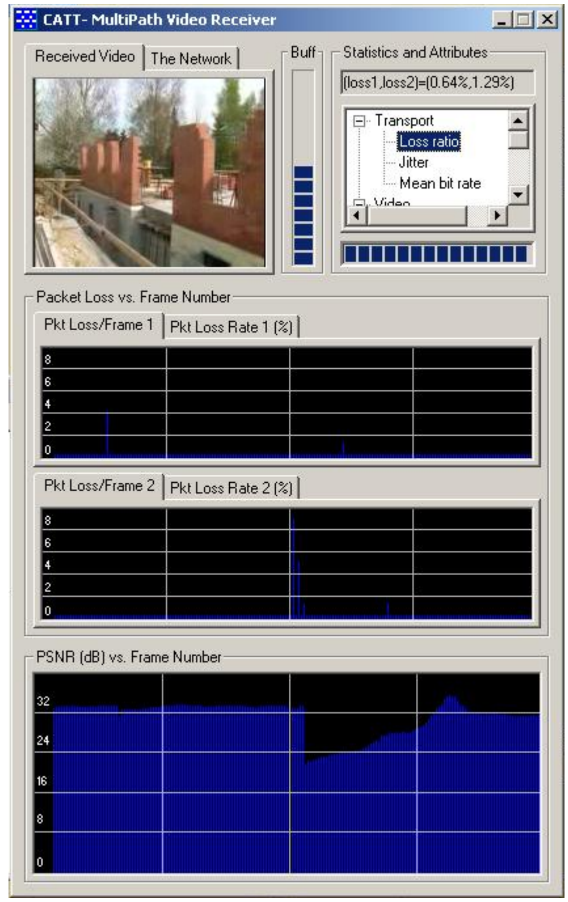


Fig. 4. A screenshot of the testbed GUI during a MDMC experiment.

TABLE I
AVERAGE PSNRS OF DECODED FRAMES: MDMC EXPERIMENTS

Scenario	Fig.2(a)	Fig.2(b)	Fig.2(b)	Fig.2(b)
Playout delay	2s	2s	2s	300ms
Pkt loss rate 1	0.41%	6.14%	8.46%	8.13%
Pkt loss rate 2	0.75%	11.96%	7.52%	7.97%
Mean Burst len. 1	1.75	3.79	3.67	6.08
Mean Burst len. 2	4.76	3.08	3.33	2.34
Ave. PSNR	33.11 dB	27.53 dB	28.65 dB	28.16 dB

As shown in Table II, ARQ effectively reduces the BL packet loss rate in all the experiments. This reduction accounts for improved video quality. For example, all lost BL packets in the test of Fig.2(a) are successfully recovered, resulting in a BL loss rate of 0%. The average PSNR for this test is the highest among all the LC with ARQ experiments. However, this is not true for the mean burst length. Since each BL packet

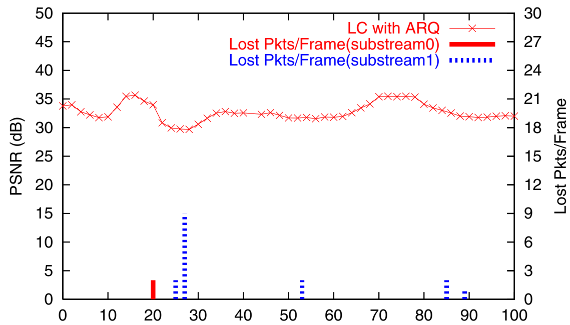


Fig. 5. A PSNR plot for the LC with ARQ scheme obtained from a testbed experiment shown in Fig.2(a). Plotted using the right y axis is the number of packets lost for each frame.

TABLE II

AVERAGE PSNRS OF DECODED FRAMES: LC WITH ARQ EXPERIMENTS

Scenario	Fig.2(a)	Fig.2(b)	Fig.2(b)	Fig.2(b)
Playout delay	2s	2s	2s	300ms
Ori. BL loss	0.06%	5.95%	7.98%	7.94%
BL loss	0.00%	2.49%	2.25%	5.37%
EL loss	0.38%	12.22%	8.14%	8.16%
Ori. BL burst len.	3.64	4.80	3.94	4.25
Mean BL burst len.	0	10.23	6.33	8.58
Mean EL burst len.	3.29	4.18	3.94	3.16
ARQ succ. ratio	100%	58.0%	71.8%	32.4%
Ave. PSNR	32.34 dB	30.64 dB	30.14 dB	30.13 dB

has its deadline imposed by the playout delay, ARQ is more effective in recovering short error bursts. Short error bursts are more likely to be recovered, reducing the number of error bursts. But for an error burst longer than the playout delay, only a portion of it is successfully retransmitted. Therefore, sometimes the mean burst length of BL increases when ARQ is used. During the LC with ARQ experiments, we observed short periods of badly corrupted frames, followed by a long period of high quality frames. However, results shown in [8] show that given the same average loss rate, PSNR improves when the mean burst length increases. The increased mean burst length, combined with reduced average BL loss rate, contributes to the increased average video quality.

The ARQ success ratio in the table is the ratio of the number of successfully retransmitted BL packets to the number of all lost BL packets. This ratio decreases as the loss rates of both paths increase, and as the playout delay decreases.

For the experiments reported here, the packet delay and delay jitter are both very low, because there is low background traffic and the bit rates of the video substreams are also low. For this reason, few packets are dropped because of lateness even with 300 ms playout delay, and the video quality obtained with 300 ms playout delay is similar to that with 2s playout delay, with both schemes. When the system load is higher, the 300ms playout delay experiment is likely to yield worse performance.

V. CONCLUSIONS

Enabling video transport over ad hoc networks is challenging both because ad hoc paths are highly unstable and compressed video is susceptible to transmission errors. However, multiple paths in an ad hoc network can be exploited as an effective means to combat transmission errors.

In this paper, we report the implementation of an multipath video streaming testbed using four notebook computers equipped with IEEE 802.11b wireless cards. Our experiments show that acceptable video quality is achievable with both LC with ARQ and MDMC, for the range of video bit rates, background traffic, motion speed, and the indoor environment examined. The testbed demonstrates the viability and performance advantages of using multistream coding and multipath transport for video transport over ad hoc networks.

ACKNOWLEDGMENT

This work is supported by the National Science Foundation under Grant ANI 0081375, the New York State Center for Advanced Technology in Telecommunications (CATT) and the Wireless Internet Center for Advanced Technology (WICAT) at Polytechnic University, Brooklyn, NY, USA.

REFERENCES

[1] IEEE 802.11, Draft Standard for Wireless LAN: Medium Access Control (MAC) and Physical Layer (PHY) Specifications IEEE, July 1996.
[2] C. R. Lin and M. Gerla, "Asynchronous multimedia multihop wireless networks," in *Proceedings of IEEE/ACM INFOCOM'97*, 1997.
[3] Y. Wang, S. Wenger, J. Wen, and A. K. Katsaggelos, "Error resilient video coding techniques," *IEEE Signal Magazine*, Vol.17, pp.61-82, July, 2000.
[4] Y. Wang and Q.-F. Zhu, "Error control and concealment for video communication: a review," in *Proceedings of the IEEE*, vol.86, issue.5, pp.974-997, May 1998.
[5] S. Lin, S. Mao, Y. Wang, and S. S. Panwar, "A reference picture selection scheme for video transmission over ad hoc networks using multiple paths," in *Proc. IEEE International Conference on Multimedia and Expo 2001*, 2001.
[6] S. Mao, S. Lin, S. S. Panwar, and Y. Wang, "Reliable transmission of video over ad hoc networks using Automatic Repeat Request and multipath transport," in *Proceedings of the IEEE Fall Vehicular Technology Conference*, Vol.2, pp.615-619, October 2001.
[7] Y. Wang and S. Lin, "Error resilient video coding using multiple description motion compensation," *IEEE Transaction on Circuits and Systems for Video Technology*, Vol.12, No.6, pp.438-452, 2002.
[8] S. Mao, S. Lin, S. S. Panwar, and Y. Wang, "Video transport over ad hoc networks: Multistream coding with multipath transport," to appear, IEEE JSAC Special Issue on Recent Advances in Wireless Multimedia.
[9] E. N. Gilbert, "Capacity of a bursty-noise channel," *Bell Syst. Tech. J.*, vol.39, no.9, pp:1253-1265, September 1960.
[10] OPNET Tech., Inc. *OPNET Modeler*. Homepage: `http://www.mil3.com`.
[11] D. A. Maltz, J. Broch, and D. B. Johnson, "Lessons from a full-scale multi-hop wireless ad hoc network testbed," in *IEEE Personal Communications*, vol.8, issue.1, pp.8-15, February 2001.
[12] H. Lundgren, *et al*, "A large-scale testbed for reproducible ad hoc protocol evaluations," in *Proceedings of IEEE Wireless Communications and Networking Conference*, vol.1, pp.412-418, 2002.
[13] W. Kellerer, E. Steinbach, P. Eisert, and B. Girod, "A real-time internet streaming media testbed,", in *Proc. IEEE ICME*, pp.453-456, Aug. 2002.
[14] ITU-T Recommendation H.263, *Video coding for low bit rate communication*, 1998.
[15] University of British Columbia, "H.263+ codec," used to be available from: *ftp://dspftp.ece.ubc.ca*.
[16] S. Mao, D. Bushmitch, S. Narayanan, and S. S. Panwar, "MRTP: A multi-flow real-time transport protocol for ad hoc networks," to appear, IEEE Fall VTC, October, 2003.