VIDEO MULTICAST OVER FAIR QUEUEING NETWORKS

Sergio D. Servetto Martin Vetterli

Laboratoire de Communications Audiovisuelles
Ecole Polytechnique Fédérale de Lausanne. CH-1015 Lausanne, Switzerland.
http://lcavwww.epfl.ch/~{servetto,vetterli}/

ABSTRACT

We consider the problem of video multicast over networks that enforce fair bandwidth allocations in the routers. State-of-the-art layered IP multicast systems perform well for moderately sized sessions, but suffer from three basic problems that prevent their massive deployment: (a) lack of fairness among different multicast sessions, (b) lack of fairness to competing TCP flows, and (c) high complexity requirements to scale up to potentially large numbers of users and sessions. In this work we present an entirely different approach to the design of these systems, in which sources obliviously inject packets into the network (disregarding congestion), whereas routers obliviously drop the amount of bandwidth that exceeds the fair share of a flow (disregarding packet contents). We model our system as a broadcast channel, with receivers connected to the source via erasure channels of different capacities, and we design a video coder to operate in this environment. Our results suggest that a combination of fair queueing routers and appropriate coding is indeed able to overcome certain drawbacks of current IP multicast systems. Furthermore, we also find that such systems would benefit significantly from being able to re-encode the video signal at internal nodes of the multicast tree.

1. INTRODUCTION

1.1. Drawbacks of Layered IP Multicast

Most work on video multicast over IP has focused on *layered* techniques, where the video source is sliced into a number of multiresolution layers, each layer is transmitted over a separate multicast group, and different receivers adapt to bandwidth fluctuations by adjusting the number of layers to which they subscribe. This is the approach pioneered by McCanne in his thesis [12], and of much of the recent work in this area (e.g., [4, 19]).

Despite its many successes, layered multicast suffers from three basic deficiencies:

- *Lack of fairness among different multicast sessions.* Without some additional machinery in the network (e.g., RED gateways [8]), it is in general not possible to guarantee that all receivers in all sessions will subscribe to all the layers their fair share of bandwidth would allow them to [13].
- *Lack of fairness to competing TCP flows.* When a number of multicast flows are present in steady-state in a network, and then a new TCP flow is added, the bandwidth that this TCP flow converges to may be less that one half of its fair share [13, 18].
- *High complexity requirements to scale up to potentially large numbers of users and sessions.* For each multicast session, the network is required to maintain a significant amount of state information in the routers [12]. And since this amount grows with the number of sessions as well as with the number of receivers in a session, serious concerns are raised about how well these schemes could scale up.

If enough bandwidth is available, from a practical point of view issues with moderate unfairness are not enough to question the use of layered multicast. However, the scalability issue is, since in this case the network may turn out to be unable to provide service in important practical situations (e.g., under the volumes that would be handled by public TV broadcasts).

1.2. Multicast with Fair Bandwidth Allocations

A problem of active research currently is the design of *low-complexity* fair queueing routers, i.e., routers capable of performing fair bandwidth allocations among competing flows without the requirement of maintaining per-flow state information. *Core-stateless* fair queueing is an example of a network architecture based on these ideas [18]. Networks based on these routers are expected to become an important part of the future QoS enabled Internet.

Under the assumption that network routers can be designed in a way such that any flow will only be allocated its fair share of bandwidth irrespective of its injection rate (i.e., the extra bandwidth is trimmed off by the network without affecting the performance of other sources), then it makes sense for a source to inject any number of packets above its fair share. In this case, the flow control task becomes trivial (packets are randomly dropped upon congestion), there is no longer a need for maintaining multiple multicast groups, neither to perform a large number of join/leave operations. Therefore, all the problems mentioned above for layered IP multicast are overcome.

But the solution to the scalability problem does not come for free: the problem is partially transferred to the source. This is so because the resulting communications channel becomes significantly more unpredictable in this case: packets are randomly dropped disregarding their content, at loss rates potentially much higher (e.g., if the injection rate is way above the fair share on a given link). Furthermore, a number of information-theoretic results establish that data compression algorithms robust to impairments of the nature introduced by this network cannot attain the performance of algorithms designed for better behaved channels, much less operate anywhere close to the applicable rate-distortion bounds for all possible delivered rates [7, 9]. We

M. Vetterli is also with the Dept. of EECS, UC Berkeley.

0-7803-6297-7/00/$10.00

see therefore that communicating over this channel poses a major signal processing challenge: how to encode video information, to make it robust to the expected wide range of impairments. The resulting modeling and coding tasks coming up in the context of video multicast over networks that implement fair queueing are therefore the subject of this work.

1.3. Main Contributions and Paper Organization

Since the way in which we deal with heterogeneity among receivers is by letting the network drop at random packets that cannot be delivered, an efficient communications system necessarily has to be based on a video coder capable of tolerating a potentially high rate of packet losses. Our contributions in this paper are: (a) the formulation of the problem of multicast over networks that implement fair queueing as one of transmission over broadcast erasure channels, and (b) the design of a robust video coding algorithm suitable for such environments.

The rest of this paper is organized as follows. In Section 2 we formulate the problem of video multicast over networks that implement fair queueing as one of communicating over broadcast erasure channels, in Section 3 we present the design of a suitable video coder, and in Section 4 we present experimental results. Conclusions are presented in Section 5.

2. CODES FOR BROADCAST CHANNELS

2.1. Non-Ergodic Multicast

We propose to model multicast transmission over a network that implements fair queueing as a problem of communication over broadcast erasure channels [6, Ch. 8.1&14.6]. Indeed, since with fair queueing routers the loss rate induced on each flow is proportional to the rate of that flow, it seems only natural to model this transmission as that of communicating over an erasure channel whose loss parameter is different for each receiver.

In our model, a transmitter is connected to 2 receivers via erasure channels with parameters p_1, p_2.[1] The loss parameters are assumed known to the transmitter, but are otherwise arbitrary: specifically, we do not assume a known probability measure on the family of channels, leading in our case to a non-ergodic formulation.[2] Our task therefore is that of encoding a message X to be sent simultaneously to both receivers, in a way such that the resulting distortions $d(X, Y_k)$ are small, as illustrated in Fig. 1.

2.2. PET Systems and Broadcast Channels

Priority Encoding Transmission (PET) is a technique proposed by Albanese et al., to encode hierarchically structured messages for transmission over a packet network which does not differentiate packets based on their content [2]. In a PET system, the user assigns to each segment a priority value. Based on their priorities, these segments are encoded into a set of packets, with each priority determining the least number of packets required at the receiver to be able to decode a segment.

PET systems have been proposed as an interesting engineering solution to the problem of communicating over packet networks under real-time delay constraints. However, recent work by Boucheron and Salamatian has established a remarkable property for these systems: using interleaved erasure-resilient codes of different strengths (as in the PET approach), it is possible to attain all points in the capacity region for broadcast erasure channels with a degraded message set [3]. Now, whereas that work is essentially of an information-theoretic nature, it has deep implications for the design of practical systems:

- First and foremost, it provides a "pseudo" separation theorem for the broadcast erasure channel. Consider a system design based on first generating a hierarchical encoding of the source, and then multicasting all these messages using a PET system. If PET systems can attain all points in the capacity region of a broadcast erasure channel, then the only suboptimality involved in this approach is due to the fact that the source may not be successively refinable [7]. But if it is, this system is indeed optimal.
- Recently, some heuristic approaches have been proposed for the design of Multiple Description (MD) coders based on PET systems [14]. Whereas these heuristics have been found to perform reasonably well in practice, from a more fundamental point of view they are somewhat deficient: in all known cases, their performance remains bounded away from the applicable MD rate/distortion bounds [10]. However, those same heuristics for the MD problem turn out to be the optimal thing to do in the problem of transmission over broadcast erasure channels.

Motivated by these observations, we define in the next section a video coder built on top of a PET system.

2.3. Construction of Nested Codes

Our basic code construction is defined as follows. Given:

- a message X_{coarse} of length k_1, for both receivers,
- a message X_{refine} of length k_2, for the good receiver,

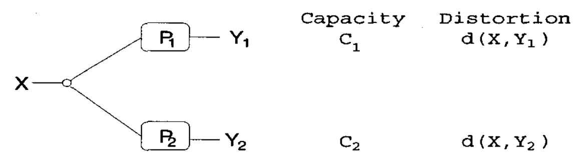

Figure 1: Non-ergodic multicast. Receivers joining and leaving a multicast session are modeled as the selection of a channel from the source to that receiver, which remains fixed and is to be used throughout its entire transmission.

[1] The extension to more receivers, although notationally cumbersome, is conceptually straightforward. We focus here on the case of two receivers only, to keep our notation simple by working with a small number of parameters.

[2] This is a major difference with the approach of Chou et al. [5], where the multicast problem is essentially reduced to a point-to-point problem for an *average* channel, where the average is taken relative to a probability measure on all possible channels.

- a pair of codes $\mathcal{C}^1$ and $\mathcal{C}^2$ (with parameters $[n_1, k_1, d_1]$ and $[n_2, k_2, d_2]$, $d_1 > d_2$),

we encode X_{coarse} using $\mathcal{C}^1$, X_{refine} using $\mathcal{C}^2$, and output the concatenation of the resulting codewords. In this construction, if a receiver loses between d_2 and $d_1 - 1$ packets then it can recover X_{coarse} only, whereas if it loses less than d_2 packets then it can decode X_{refine} as well.

3. VIDEO CODING FOR MULTICAST

3.1. Motion Compensation and 3D Subbands

Motion compensation is a technique which, although highly successful in the context of "monolithic" video coding (meaning, coding for a single target bit rate and for transmission/storage in error-free media), is inherently difficult to adapt to the requirement of robustness to a wide range of packet loss rates mandated by our application.[3] Two features of motion compensation contribute to this situation:

- *Error propagation.* For a given motion field, motion compensation is essentially a linear predictor with gain factor 1. Hence, the resulting filter is only marginally stable, and so errors do not vanish over time. Plus, if errors affect the motion field itself, the error signal is only amplified.
- *Lack of support for rate-scalability.* Rate-scalability is inherently difficult in the context of predictive coding, essentially because it is unclear what data will be available to the decoder to compute predictions.

Conversely, 3D subband coding does not suffer from any of these deficiencies: error propagation is limited to the length of the filters used, and work has been done on scalable subband video coding [11, 20]. Therefore, we see that for our application, 3D subband coding is a better choice.

3.2. 3D Subband Coding of Video for Multicast

Following a strategy closely related to that employed in [15, 16] mostly for still images, here we take the following steps:

1. Decompose the input video signal into subbands.
2. Quantize each subband twice, with quantizers of step-size Δ and $n\Delta$ (for some positive integer n).
3. For each subband form a causal local variance estimate, and based on it select one of two possible probability models to use in the entropy coder. Encode the field of coarse quantizer outputs, and then encode the field of fine quantizer outputs, but conditioned on the values of the coarse field.
4. Apply nested codes as described above, to the encoded fields of coarse and fine outputs of each subband.

[3] We should point out that this situation is very specific to multicast. Other applications involving transmission over "difficult" channels (e.g., video over wireless) have successfully applied motion compensated coders. The key difference is that in those cases, a feedback channel goes a long way towards making the problem more tractable, something we do *not* have in this case.

3.3. Optimization of Parameters

The final stage in the development of our system consists of defining an algorithm for setting parameters of the encoder. That is, given (a) the channels to both receivers, and (b) a constraint on the maximum distortion in the signal delivered to the low rate receiver, our goal is to maximice the quality of the signal delivered to the high rate receiver as a function of the quantizers and codes used. Formally, we want to solve

$$\min_{\{[\Delta,n]_k,[\mathcal{C}^1,\mathcal{C}^2]_k\}} \sum_{k=1}^{N} \mathcal{D}_{P_1}(s_k, [\Delta, n]_k, [\mathcal{C}^1, \mathcal{C}^2]_k)$$

subject to the constraint

$$\Sigma_{k=1}^{N} \mathcal{D}_{P_2}(s_k, [\Delta, n]_k, [\mathcal{C}^1, \mathcal{C}^2]_k) \leq D_{\text{budget}}$$

where $[\Delta, n]$ is a nested quantizer, $[\mathcal{C}^1, \mathcal{C}^2]$ is a nested code, s_k is the k-th subband in the current decomposition, and $\mathcal{D}_{P_i}$ is the average distortion in the signal delivered over the channel P_i $(i = 1, 2)$.

This optimization problem is a standard one of resource allocation under constraints, whose solution can be obtained exactly via Dynamic Programming (DP), or almost exactly (at greatly reduced complexity) via Lagrange multiplier methods [17]. Specific details about how exactly we find good allocations will be presented elsewhere.

4. EXPERIMENTAL RESULTS

To illustrate some of the issues that make this problem interesting, in this section we present a number of results obtained in experiments. Consider a broadcast channel with two receivers: 4 Mbits/sec are available from the transmitter to the good receiver, and 2 Mbits/sec available to the bad receiver: if the encoder injects 4 Mbits/sec, then the good receiver will see no packet losses, and the bad receiver will see a 50% packet loss rate. Now consider the following possible coding strategies:

- Encode the source at a rate of 4 Mbits/sec. In this case, the good receiver will see a high quality signal, the bad receiver will see only a black signal. This is an extreme case, in which the signal quality delivered to the good receiver is maximized.
- Encode the source at a rate of 2 Mbits/sec, add another 2 Mbits/sec of parity bits using an MDS code (e.g., Reed-Solomon).[4] In this case, both receivers will decode a 2 Mbits/sec signal. This is another extreme case, in which the signal quality delivered to the bad receiver is maximized.
- In between the previous two extremes there is a number of other options. By spending a fraction of time $0 \leq \lambda \leq 1$ optimizing for the bad receiver and $1 - \lambda$ optimizing for the good receiver, a number of other options are available as well:
  1. Encode a coarse version of the source, at a rate of 2λ Mbits/sec.

[4] MDS codes are codes in which each extra parity bit added permits correcting one more erasure.

2. Add another 2λ Mbits/sec of MDS parity bits.
3. Encode a *refinement* of the source, at a rate of $4(1-\lambda)$ Mbits/sec. In this case, the bad receiver gets a signal encoded at 2λ Mbits/sec, whereas the good receiver gets $2\lambda + 4(1-\lambda)$ Mbits/sec.

Note that as simple minded as this approach might seem, the result of Boucheron and Salamatian proves that this system is essentially optimal, modulo suboptimalities due to the source may be not being successively refinable [3, 7].

We computed the resulting PSNR tradeoffs for a preliminary version of our proposed video coder. For this purpose, we used YUV sequences of size 352x240 pixels (30 frames/sec), and Haar filters.[5] PSNR plots are shown in Fig. 2, for the standard football sequence, and for channels in which the bad receiver gets 1 or 2 or 3 Mbits/sec.

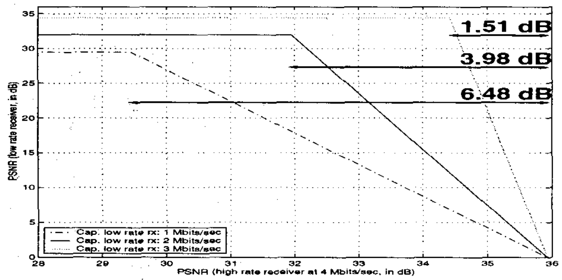

Figure 2: PSNR results for different channels. The numbers in the upper right corner indicate the *minimax* gap: the performance drop suffered by the good receiver when operating at the maximum rate at which information can be delivered reliably to the bad receiver.

A most interesting observation to make about Fig. 2 is that, when the difference in channel capacities between receivers is large, there seems to be a steep price to pay when trying to improve the PSNR delivered to the good receiver using time-sharing, in terms of PSNR delivered to the bad receiver. This is illustrated for example by the fact that, when the bad receiver has a capacity of 1 Mbit/sec, each dB gained by the good receiver is payed for with 5 dB taken off the bad receiver.

5. CONCLUSIONS

In this paper, we considered the problem of video multicast over a network capable of enforcing fair bandwidth allocations in its routers. We explained that in the future the Internet is most likely going to support some form of packet differentiation at the routers (e.g., based on the core-stateless approach [18]), and proposed a method by which some of the known drawbacks of current layered IP multicast could be overcome. To do so, we formulated the multicast problem as one of transmission over broadcast channels, where each receiver sees an erasure channel with different loss probabilities. Then we designed a video coder for this channel, and presented performance results.

The main conclusion we draw from our results is that there is a clear need for introducing coding at internal nodes in the multicast tree. Observe Fig. 2 again. When the bad receiver has capacity 3 Mbits/sec (a bit rate gap of only 1 Mbit), the minimax gap is about 1.5 dB: for video coded at such high rates, this difference is visually imperceptible. However, when the bad receiver has capacity 1 Mbit/sec (a bit rate gap of 3 Mbits), not only the minimax gap is much larger and visible (about 6.5 dB), in order to reduce it by 1 dB it is necessary to accept a drop of 5 dB for the bad receiver, resulting in extremely poor performance in the latter case. This is probably close to the best one can do with time-sharing, and time-sharing is indeed the best one can do for broadcast erasure channels [3]. Therefore, whenever an encoded signal reaches a node in which the capacities of receivers down the tree differ significantly, the only way to deliver good performance to all these receivers simultaneously is by re-encoding the signal.

Note also that the need for coding at the nodes is consistent with the recent results of Ahlswede et al. [1]. In that work it is essentially shown that there is an inherent suboptimality in regarding information to be multicast as a "fluid" which can be only routed or replicated: by permitting coding as well at the nodes, they prove that higher data rates can be achieved. Our current work focuses on the design of a protocol suite for *receiver-driven* multicast (similar to [13]), in which instead of having internal routers re-encode the signal it is the receivers who distribute among themselves the task of re-encoding when necessary.

[5] Haar filters are the ones we found empirically to provide the best performance. Most other filters we tried induce visually (very) annoying ringing artifacts, due to their length.

6. REFERENCES

[1] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung. Network Information Flow. *IEEE Trans. Inform. Theory*, to appear.

[2] A. Albanese, J. Blömer, J. Edmonds, M. Luby, and M. Sudan. Priority Encoding Transmission. *IEEE Trans. Inform. Theory*, 42(6):1737–1744, 1996.

[3] S. Boucheron and M. R. Salamatian. About Priority Encoding Transmission. *IEEE Trans. Inform. Theory*, 46(2):699–705, 2000.

[4] P. A. Chou, A. E. Mohr, A. Wang, and S. Mehrotra. FEC and Pseudo-ARQ for Receiver-Driven Layered Multicast of Audio and Video. Technical Report MSR-TR-99-86, Microsoft Research, 1999. Available from http://www.research.microsoft.com/~pachou/.

[5] P. A. Chou and K. Ramchandran. Clustering Source/Channel Rate Allocations for Receiver-Driven Multicast under a Limited Number of Streams. In *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, New York, NY, 2000.

[6] T. M. Cover and J. Thomas. *Elements of Information Theory*. John Wiley and Sons, Inc., 1991.

[7] W. Equitz and T. M. Cover. Successive Refinement of Information. *IEEE Trans. Inform. Theory*, 37(2):269–275, 1991.

[8] S. Floyd and V. Jacobson. Random Early Detection for Congestion Avoidance. *IEEE/ACM Trans. Networking*, 1(4):397–413, 1993.

[9] A. A. El Gamal and T. Cover. Achievable Rates for Multiple Descriptions. *IEEE Trans. Inform. Theory*, IT-28(6):851–857, 1982.

[10] V. K. Goyal. *Transform Coding: Classical and Multiple Description*. SIAM, 2000.

[11] B.-J. King, Z. Xiong, and W. Pearlman. Very Low Bit Rate Embedded Video Coding with 3D Set Partitining in Hierarchical Trees (3D SPIHT). *IEEE Trans. Circ. Syst. Video Tech.*, to appear.

[12] S. R. McCanne. *Scalable Compression and Transmission of Internet Multicast Video*. PhD thesis, University of California, Berkeley, 1996.

[13] S. R. McCanne, V. Jacobson, and M. Vetterli. Receiver-Driven Layered Multicast. In *Proc. ACM SIGCOMM*, Stanford, CA, 1996.

[14] R. Puri and K. Ramchandran. Multiple Description Source Coding Using Forward Error Correcting Codes. In *Proc. Asilomar Conf. Signals Syst. Comp.*, Pacific Grove, CA, 1999.

[15] S. D. Servetto. *Compression and Reliable Transmission of Digital Image and Video Signals*. PhD thesis, University of Illinois at Urbana-Champaign, 1999.

[16] S. D. Servetto, K. Ramchandran, V. A. Vaishampayan, and K. Nahrstedt. Multiple Description Wavelet Based Image Coding. *IEEE Trans. Image Proc.*, 9(5):813–826, 2000.

[17] Y. Shoham and A. Gersho. Efficient Bit Allocation for an Arbitrary Set of Quantizers. *IEEE Trans. Acoust. Speech Signal Proc.*, 36(9):1445–1453, 1988.

[18] I. Stoica, S. Shenker, and H. Zhang. Core-Stateless Fair Queueing: a Scalable Architecture to Approximate Fair Bandwidth Allocations in High Speed Networks. In *Proc. ACM SIGCOMM*, 1998.

[19] W.-T. Tan and A. Zakhor. Error Control for Video Multicast using Hierarchical FEC. In *Proc. IEEE Int. Conf. Image Proc. (ICIP)*, Kobe, Japan, 1999.

[20] D. S. Taubman. *Directionality and Scalability in Image and Video Compression*. PhD thesis, University of California, Berkeley, 1994.

ROBUST H.263+ VIDEO FOR REAL-TIME INTERNET APPLICATIONS

J.T.H. Chung-How and D.R. Bull

Image Communications Group, Centre for Communications Research
University of Bristol, Woodland Road, Bristol BS8 1UB, U.K.

ABSTRACT

Any real-time interactive video coding algorithm used over the Internet needs to be able to cope with packet loss, since the existing error recovery mechanisms are not suitable for real-time data. In this paper, a robust H.263+ video codec suitable for real-time interactive and multicast Internet applications is proposed. Initially, the robustness to packet loss of H.263 video packetised according to the RTP-H.263+ payload format specifications is assessed. Two techniques are proposed to minimise temporal propagation – *selective FEC of the motion information* and the use of *Periodic Reference frames*. It is shown that when these two techniques are combined, the robustness to loss of H.263+ video is greatly improved.

1. INTRODUCTION

The transmission of real-time video over the Internet is becoming increasingly desirable for videoconferencing, distance learning and other applications. The Internet was designed mainly for non-time critical data, and is ill-suited for the transmission of time-critical data such as interactive video. Packets can be lost or dropped when intermediate links or routers become congested due to excess traffic. Reliable transport protocols like TCP/IP recover from loss by using acknowledgements and retransmissions. However, the resulting latency is generally too large for real-time interactive applications, where late packets are effectively lost. As a result, real-time multimedia applications typically use UDP/IP, which provides an unreliable packet delivery service. The Real-time Transport Protocol (RTP) [1] was defined to enable real-time multimedia applications over the Internet. A payload format for H.263+ video has been defined for use with RTP [2]. The new payload format for H.263+ that has been specified can also be used with the original version of H.263. Video transmission over the Internet has received considerable attention recently. The most popular scheme for providing error-resilience in a packet video transport system is scalable or layered coding combined with some form of prioritisation, forward error correction (FEC) [3,4] or receiver-based rate control which is particularly well suited for the heterogeneous nature of the Internet [5,6].

In this paper, the problem of robust transmission of H.263+ video over the Internet is addressed. In order to make our solution applicable to the widest range of situations requiring video over IP, we assume that there is *no* feedback channel from the decoder to the encoder, as is the case for a typical multicast application because of the feedback implosion problem. We also assume a real-time environment with strict end-to-end delay requirements, such as in a typical two-way videoconferencing application. Further improvements are possible if these constraints are relaxed, e.g. in a video-streaming application where delay is not so critical. First, the robustness to packet loss of RTP-H.263+ video is investigated. The main problem is caused by temporal error propagation and two ways of minimising this propagation are proposed - the *selective use* of FEC on the motion information and the use of *periodic reference* frames. Using *periodic reference* frames is shown to be more efficient and robust than periodic intraframe coding when used with FEC. These two algorithms are then combined, resulting in a very efficient robust video-coding scheme that provides graceful degradation as the packet loss rate increases.

Generally, Internet packet loss rates vary widely and losses may occur in bursts, i.e. a lost packet is more likely to be followed by another loss [7,8]. However, no simple method exists for modelling the typical loss patterns likely to be seen over the Internet since the loss depends on so many untractable factors. Therefore, in all the loss simulations presented in this paper, random loss patterns are used. This is believed to be the best solution since our test sequences are of short duration and burst losses can be modelled as very high random loss.

2. RTP-H263+ WITH PACKET LOSS

In order to transmit H.263+ video over the Internet, the H.263+ bitstream must be packetised according to the RTP-H.263+ payload format specification [2], and then transmitted as RTP packets. In addition to the RTP/UDP/IP headers, the RTP packetisation also generates a RTP-H.263+ payload header, which is normally 16 bits. To minimise packetisation header overhead, each RTP packet should be as large as possible. In practice, to avoid IP fragmentation, the size of the packet must be kept below the maximum transmission unit (MTU) of the network, which is 1500 bytes for Ethernet. On the other hand, for maximum robustness to loss, packet size should be kept to a minimum.

In our experiments, the slice-structured mode was not used and packetisation was always performed at GOB boundaries, i.e. each RTP packet contains one or more complete GOBs. Since every packet begins with a picture or GOB start code, the leading 16 zeros are omitted in accordance with RFC 2429 [2]. The packetisation overhead then consists only of the RTP/UDP/IP headers, which is typically 40 bytes per packet. This overhead can be quite significant at low bit-rates.

The header overhead associated with using 1,3 and 9 GOBs/packet for QCIF images at 12.5 fps is given in Table 1. Results for one simulation with the foreman sequence (QCIF, 125 frames, 12.5 fps) coded at 62 kbps (excluding RTP/UDP/IP header overhead) with 10% random packet loss are shown in Fig 1. Only the first frame was intracoded. The RTP sequence number enables the decoder to detect lost packets, so that the