

Issues with multicast video distribution in heterogeneous packet networks

Thierry Turletti Jean-Chrysostome Bolot
INRIA
B. P. 93
06902 Sophia-Antipolis Cedex
France
{turletti, bolot}@sophia.inria.fr

1 The problem

We consider the problem of distributing a VBR video stream over a multicast tree (i.e. from a source to N destinations) where the branches of the tree are non-homogeneous. In practice, such a tree appears in a packet network such as the Internet in which branches have different characteristics (bandwidths, buffer space, etc). Typical VBR video sources used in the Internet (e.g. the video coders in IVS [14] and NV [7]) can control the rate at which they send packets into the network by adjusting parameters in the video coder. The problem then is for a video source to choose an appropriate output rate (by ‘output rate’, we mean either the maximum output rate or some average output rate). This output rate characterizes the bandwidth requirements of the source. Clearly, the choice of the output rate will impact the level of congestion, expressed e.g. in terms of delay and loss patterns, experienced on the tree. Since the network is heterogeneous, we can expect this level of congestion to be widely different on different branches. The source might then choose its output rate so that only a small fraction of the branches of the tree are congested. For example, if the fraction is equal to 10%, then the output rate is set so that no more than 10% of the receivers are congested. If the fraction is equal to 0%, then the output rate is set low enough so that no part of the tree is congested. This solution is

easy to implement, but both choices of the fraction above have limitations. In the first case, the video delivered to some of participants will be of low quality (since packet losses and/or delays will be high on some branches of the tree). The second case is safest, but it results in disturbing the bulk of the participants.

In practice, it is not clear how to choose an adequate value for this fraction. In IVS, we set the fraction to a few percent [2, 3]. Ideally, however, the source should be able to single out the parts of the multicast tree that experience congestion. In order not to disturb the bulk of participants in the tree, these branches should be treated separately. Two possible solutions include *video gateways*, and using some form of *layered coding*.

Video gateways or layered coding schemes are not new ideas. Our contribution is to identify and discuss the issues associated with using these techniques in the Internet. We illustrate our points with the H.261[10] software coder of IVS.

2 Video gateways

Video gateways take as input a video flow encoded using some scheme, say X , and forward this flow down the multicast tree encoded using some other scheme Y with lower bandwidth requirements than X .

Recently, a gateway that converts video from the CU-SeeMe [5] to the NV format has been ad-

vertised. However, its goal is apparently less to avoid congestion on branches of a multicast tree than to allow interoperability of different kinds of codecs (no reference about this work is available yet). We will also be developing at INRIA a gateway that converts video from the MPEG[1] to the H.261 format used in IVS. A simple way to do this is to remove the B frames from the MPEG stream to obtain a lower-bandwidth stream. The video gateway is then similar to a selective filter as described in [13].

In both examples above, different coding schemes are used for flow X and flow Y. However, it is also possible to use the same coding scheme for X and Y provided different parameters are used. Example parameters for a H.261/MPEG coding scheme include the quantizer value and the movement detection threshold. Adjusting these parameters makes it possible to forward a lower rate video flow down the multicast tree, although at the expense of a lower image quality. If the precise rendition of individual images (as opposed to the precise rendition of movement) is important to the destinations, then it is also possible for the gateway to decode incoming frames and forward some of them re-encoded in INTRA mode (i.e. using intraframe coding).

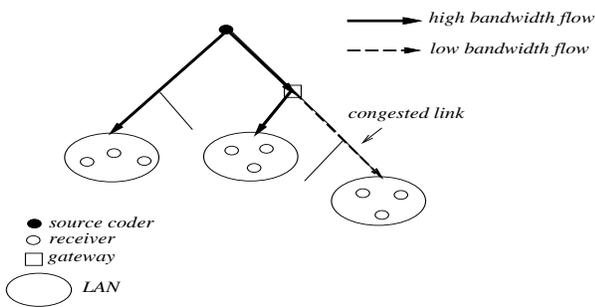


Figure 1: A video gateway in a multicast tree

3 Layered coding for the Internet

A second solution to the problem of multicast video distribution over heterogeneous networks is to split the video flow generated by the source into

multiple flows, each one with different bandwidth requirements than the original flow. The obvious way to do this is to encode the video with a layered or hierarchical coding scheme. In the two layer case, the baseflow typically includes the low resolution information, and it can be decoded into a meaningful service. The other flow includes enhancement information. The idea then is to transmit both streams over the non-congested branches of the multicast tree, but to transmit only the base stream over the congested branches.

The H.261 scheme used in IVS, as well as the MPEG scheme, do not lend themselves very easily to hierarchical coding. One way to do it is to encode the first n DCT coefficients (ranked according to their significance using the zigzag scan technique) in the base flow, and the remaining coefficients in the other flow. However, it turns out that the base flow decoded by itself yields a low quality picture. Another way is to encode the n DCT coefficients with the highest energy in the base flow. Another way is to encode in the base flow only those coefficients with energy higher than some threshold. Yet other ways are described in [9, 4, 8]. Recent work indicates that the energy threshold approach yields good quality video flows yet requires little CPU power [6].

However, layered coding is useful if the nodes in the multicast tree are able to selectively forward one or both flows to a specific branch. Furthermore, it should also be possible to assign packets from the base flow a higher priority than that assigned to packets of the other flow. Using a higher priority for packets from the base flow guarantees a minimal video quality even during periods of network congestion. If one such packet is lost, then corresponding data sent in the other flow is useless, and the quality of the video delivered to the destinations is disappointing [9]. Unfortunately, current routers in the Internet are not smart enough to handle selective forwarding and/or priority queues.

Furthermore, we note that video flows encoded with layered coding might not be compatible with hardware codecs. Consider for example the frequency truncation scheme described above in

which the first n DCT coefficients (i.e. the low frequency coefficients) are encoded in one flow and the remaining coefficients in another flow. Only the low frequency flow can be decoded by hardware codecs. To decode both flows, a multiplexer/resequencer must be added before the hardware codec. The multiplexer Huffman decodes both flows, merges the decoded DCT coefficients from both flows into blocks, and then Huffman encodes these blocks to obtain a new flow.

In conclusion, layered coding schemes appear attractive, but several networking issues make them premature for the current Internet environment.

4 Multiple coding for the Internet

The layered coding schemes described above are non-overlapping, in the sense that the information in different flows (corresponding to different layers) is non overlapping. It is also possible to use 'not-quite-layered' coding schemes, i.e. schemes in which the information sent by the source in the different flows is overlapping¹.

With H.261 coding for example, when the frame rate is more important than the rendition quality of the image, one approach consists of encoding each video frame with two different quantizer values. The frame encoded with the coarser quantizer value (i.e. most compressed) is sent on the low-fidelity video flow. Unlike pure layered coding techniques, this method provides full compatibility with hardware codecs.

In the opposite case, i.e. when the rendition quality of the image is of primary importance, we propose the following implementation: As with MPEG encoding, enforce INTRA mode for one out of every N frames². Send this INTRA frame

¹Such schemes are similar to simulcast techniques described in [11]

²Note that the H.261 recommendation only specifies that each block should be INTRA encoded at least one time out of 132 in order to limit error accumulation in the inverse cosine transform.

along with the INTER/INTRA frames in the high quality video flow but also save it in memory at the coder side. Now send this saved INTRA frame periodically in the low fidelity flow (e.g 10% of the high fidelity output rate), until the next INTRA encoded frame. The periodic resending is used for packet loss recovery. Note that the low-quality flow is less sensitive to packet loss since it uses PCM encoding (INTRA mode) instead of DPCM encoding (INTER mode). The main drawback of this method is the jerky frame rate which may result : each time a full INTRA encoding is done, the video frame rate is decreased since all blocks of the image must be encoded, this requires more cpu time both to encode and decode it. Using simulcast, only one flow will be decoded in each case; this suppresses the need for multiplexer/resequencer but increases the total bandwidth used in most parts of the multicast tree.

5 Joining and quitting flows

In all the above techniques, multicast route pruning³ is used so that packets flow over only those links necessary to reach active receivers; this reduces the traffic level and allows the use of layered coding. Low-fidelity and high-fidelity video flows must be sent with different multicast addresses since pruning does not work for individual ports. Priority mechanisms, which should be available in the future, could be used to selectively discard packets at congestion points (e.g. by giving more priority to the low-fidelity video flow). But even when such priority mechanisms are used, the joining/quitting procedures will not be redundant. Indeed, the priority mechanisms cause high fidelity packets to be discarded at the bottleneck, but quitting the corresponding multicast group allows that traffic to be discarded even earlier. In other terms, priority mechanisms will handle the immediate needs of the congestion

³Pruning mechanisms have been implemented since version 3.1 of mrouter

avoidance⁴.

Decoders should not join/quit a high fidelity traffic flow too often since the join/leave mechanism is expensive. A long sampling interval is required in order to avoid highly responsive joining and leaving procedures. The problem is to determine how to pick and adapt the loss sampling rate interval and the re-join interval. Typically, all participants will join the high-fidelity video flow by default except if some problems are known before (e.g. low-rate link in the multicast tree path or known congestion problems at some period). In the latter case, a manual switch is more efficient.

In the scalable congestion control scheme [3] implemented in version 3.3 of IVS, receivers periodically send the current network state⁵ observed to the coder, and the coder sends the global estimate of all network state received to the group. If a receiver sends a CONGESTED state to the coder and if the coder continues to send a different state (i.e. LOADED or UNLOADED), then the receiver can conclude that the bulk of the group have better network conditions. Since the coder will not react by reducing its output rate, the congested receiver will switch to the low-fidelity flow in order to reduce the loss rate observed. Now, if this participant observes an UNLOADED network state for sufficient time (e.g. 2 minutes), it could try to switch again to the high-fidelity flow. Note that the sampling interval could be increased after each unfruitful attempt in order to reduce the useless join/leave attempts.

6 Conclusion

We have discussed the issues associated with the distribution of a VBR video stream over a multicast tree. This problem is important because this kind of traffic is increasing and rapidly taking a larger share of Internet resources. We

⁴Some of these issues have been discussed in the IDMR[12] working group of the IETF.

⁵UNLOADED, LOADED and CONGESTED generic variables are used to characterize the current network state and correspond to pre-defined loss rate thresholds.

illustrated our points with the H.261 software coder of IVS, available by anonymous ftp from zenon.inria.fr:rodeo/ivs.

References

- [1] R. Aravind et al., "Image and video coding standards", *AT&T Tech. Journal*, pp. 67-89, Jan/Feb. 1993.
- [2] J-C. Bolot, T. Turletti, "A rate control scheme for packet video in the Internet", *Proc. IEEE Infocom '94*, Toronto, Canada, pp. 1216-1223, June 1994.
- [3] J-C. Bolot, T. Turletti, I. Wakeman, "Scalable feedback control for multicast video distribution in the Internet", to appear in *Proc. ACM Sigcomm '94*, London, UK, Sept. 1994.
- [4] Y.C. Chen, K. Sayood and D.J. Nelson, "A robust low-rate coding scheme for packet video", *Communication, Control and Signal Processing*, Elsevier Science Publishers B.V., pp. 1490-1498, 1990.
- [5] R.Cogger, "CU-SeeMe videoconferencing tool", Unix Manual Page, Cornell University.
- [6] B. DeCleene et al., "Comparison of priority partition methods for VBR MPEG", *Proc. IEEE Infocom '94*, Toronto, Canada, pp. 689-696, June 1994. R.Cogger, "CuSeeMe videoconferencing tool", Unix Manual Page, Cornell University.
- [7] R. Frederick, "nv", Unix Manual Page, Xerox PARC, 1992.
- [8] H. Gharavi, "Multilayer subband-based video coding", *IEEE Trans. on Communications*, vol. 39, no 9, pp. 1288-1291, Sept. 1991.
- [9] M. Ghanbari, "Two-layer coding of video signals for VBR networks", *IEEE JSAC*, vol. 7, no 5, pp. 771-781, June 1989.
- [10] "Recommendation H.261: Video codec for audiovisual services at p*64 kb/s", CCITT White Book, 1990.
- [11] D. Hoffman, M. Speer, G. Fernando, "Network support for dynamically scaled multimedia data", *Proc. NOSSDAV '93*, Lancaster, UK, pp. 269-278, Oct. 1993.
- [12] idmr@cs.ucl.ac.uk mailing list. Archive available by anonymous FTP from cs.ucl.ac.uk:/darpa/idmr-archive.Z
- [13] J. Pasquale et al., "Filter propagation in dissemination trees: Trading off bandwidth and processing in continuous media networks", *Proc. NOSSDAV '93*, Lancaster, UK, pp. 269-278, Oct. 1993.
- [14] T. Turletti, "H.261 software codec for videoconferencing over the Internet", INRIA Research Report 1834, Jan. 1993.