Watercasting: Distributed Watermarking of Multicast Media

Ian Brown Colin Perkins Jon Crowcroft
{I.Brown|C.Perkins|J.Crowcroft}@cs.ucl.ac.uk
Department of Computer Science
University College London
Gower Street
London WC1E 6BT
United Kingdom

11 February 1999

Abstract

This paper outlines a scheme that allows encrypted multicast audiovisual data to be watermarked by lightweight active network components in the multicast tree. Every recipient receives a slightly different version of the marked data, allowing those who illegally re-sell that data to be traced. Groups of cheating users or multicast routers can also be traced. There is a relationship between the requirements for the scheme proposed here, the requirements for reliable multicast protocols, and the need for mechanisms to support layered delivery of streamed data.

1 Introduction

When talking about multicast in the Internet, the loosely coupled Mbone [3] model often causes information providers some disquiet. There are a number of reasons for this, but the one of interest to this paper is the relative anonymity of the receivers: senders send to the group address, receivers inform their local router that they are interested in traffic addressed to that group, and the routers conspire to deliver the traffic [20]. Traffic forwarding and group membership are local issues, and at no point is the complete group membership known unless an application level protocol is used to provide an approximate list of members (for example RTCP [1]). This anonymity implies that it is a simple matter to eavesdrop on, and record, a stream in an undetectable manner.

There have been various proposals for protecting multicast data using public key authentication and encryption techniques, together with some suggestions for providing scalable key distribution for such services [4, 5]. These schemes rely on the cost of re-transmitting media data being large enough to deter paying customers from re-selling on content they have received. However, the Internet provides an environment where miscreants can easily re-transmit data without detection and on a large scale. There is also the problem, that even if illegal copies are found, it may be hard to find which of the legitimate receivers has "turned coat" and carried out the misdeed.

One way to provide an audit trail of the origin of a copy is to use a technique known as *watermarking*. This entails embedding additional, virtually undetectable, information in the data to distinguish one copy from another. There is, however, a problem with using watermarking with broadcast or multicast media: the addition of distinguishing marks to the data stream is contrary to the bandwidth saving of being able to distribute the same data efficiently to multiple recipients.

The proposal in this paper is to make use of lightweight active network components at branch points in an IP multicast network (something that is hard or impossible to achieve in a broadcast network) to modify the stream so that recipients are delivered unique versions of the sourced data. It turns out that some of the mechanisms being proposed to provide distributed support for reli-

able multicast data delivery and for delivery of scalable, layered, streamed multicast data (such as video and audio) have very similar properties to those that we need here. In fact, there is a family of end-to-end services that can make use of a distributed, heterogeneous and dynamic set of filters in a multicast distribution tree.

In the next section we describe related work on watermarking and multicast security. Following that, we outline the salient points of our approach. Then we analyse its efficiency and look at possible threats to its effectiveness. Finally we look at what needs to be done to make the system work in practice.

2 Background and Related Work

2.1 Watermarking

Typically, in broadcast networks, cryptographic techniques are sufficient to give a sufficient level of protection against dishonest reception. The major problem is legitimate users illegally selling their keys to others. Schemes have been designed to make keys effectively as large as the content they protect, and hence uneconomic to sell [6], or to identify the source of stolen keys [7] even when legitimate users collude to try and cover their trail. Such schemes are ineffective against large- scale fraud and so still require trusted decoders and rigorous identification of subscribers. They are also no protection against retransmission of content rather than keys.

With the Internet available as an extremely efficient retransmission mechanism, further protection is essential. Watermarking allows content providers to trace subscribers who have illegally sold on data by indelibly but invisibly marking the data sent to each user with their details.

The simplest watermarking schemes use the low bits in an audio or image file to embed information such as subscriber ID. These are easily defeated by altering these bits. More complex schemes select a subset of bits to alter using a secret key, or use spread spectrum or complex transformations of the data to make removal of the watermark more difficult.

Anderson and Manifavas describe an ingenious scheme that allows a single broadcast ciphertext to be decrypted to slightly different plaintexts by users with slightly different keys. Unfortunately the scheme is extremely vulnerable to collusion between users. Five or more users can together produce plaintext (or keys for installation in pirate decoders) that cannot be traced. Shamir has pointed out that increasing collusion resistance in all of these schemes requires exponential work from the defender to cost the attacker linearly more effort [8].

Regrettably this is the only scheme proposed so far that allows efficient marking of data being supplied to large numbers of users. While others are not hugely computationally intensive, they would not scale well. A content provider would need huge computing resources to be able to watermark data being sent to typical live event audiences. An enormous amount of bandwidth would also be used up sending a different version of the data to each viewer. This motivates our approach, which distributes the processing needed throughout the multicast tree used to efficiently deliver data.

2.2 Multicast Security

The IP multicast model allows for *any* receiver to become a sender. Unlike traditional broadcast networks, the effort needed to re-multicast data is small. Further, any host can receive multicast traffic, with the decision to route data to that host being made close to that host with no reference to the original source of the data. It becomes clear that, in addition to marking data to trace those who illegally redistribute it, we also need to protect data in transit to prevent unauthorised access by eavesdroppers.

By encrypting IP packets, we ensure only those possessing the necessary decryption key can access content. This encryption is typically performed at the application level, although in the future it may be integrated into the IP forwarding mechanisms [16]. Whilst the problem of key management for multicast data is not yet completely solved, proposals to provide this functionality [18] are moving forward and could build on content providers' systems for authenticating users. With rapid re-keying, content providers could remove pirates from groups even quicker than current pay-per-view systems.

If access control is not employed, spoofing traffic to a multicast group is relatively simple. In the main, hosts do not authenticate themselves to their local multicast

router (or to the core in routing schemes such as PIM-SM or CBT), so source-address spoofing is possible. The reverse-path forwarding check present in some routing protocols limits the scope of this (can only spoof sources on the reverse path), but does not eliminate the problem entirely.

It is possible to limit multicast traffic to a specific region of the network, using administratively scoped addressing [21]. This relies on border routers of the administrative region being correctly configured to prevent traffic sent to certain address ranges leaking out of the region. It provides an effective means of limiting the flow of traffic if correctly configured, but does not prevent unauthorized reception of data by hosts within the region. It is also difficult to configure and use, although future protocol developments may ease these problems [22].

To summarise, we note that it is almost impossible to limit access to multicast data; we must rely on encryption and good key management to prevent intercepted traffic being decoded, and watermarking to trace authorised users who illegally redistribute content.

2.3 Multicast Loss Characteristics

A number of studies have been conducted into the performance and loss characteristics of the Mbone [10, 11]. These have shown a large amount of heterogeneity in the reception quality for multiple receivers in a single session, posing a challenge to the designers of both resilient multicast streaming protocols [12, 13] and reliable multicast transport protocols [14].

The *loss signature* of a receiver is used by a number of protocols to identify subsets of receivers which belong, at least symptomatically, to shared subtrees. This signature has two components: the temporal pattern of packet loss, and the correlation between the position of a receiver in the multicast distribution tree and the observed loss.

The temporal correlation of packet loss has been noted by a number of authors. Bolot [23] noted that packet loss is not independent (it is more likely that a packet is dropped if the previous packet was also lost), and derived a simple Bernoulli model for such loss. More recent work [24, 25] notes that this model is not sufficient in many cases, and that higher-order Markov models are more accurate.

Correlation is also noticable at longer time-scales. For example, Handley [10] and Bolot [23] have noted bursts of loss with a 30 second period (possibly due to router bugs) and the authors have noted similar effects.

Packet loss is also correlated between receivers, such that many receivers see the same patterns of loss [11]. This is clearly due to lossy links within the distribution tree which cause loss for all leaf nodes below them. Packet loss correlation is a therefore a good predictor for the shape of the multicast distribution tree [26].

One significant drawback of these techniques is that whilst loss signatures match the network topology to a fairly high accuracy, they do not allow the topology to be discovered directly, although recent work has shown that it is possible to infer the *logical* network topology based on packet loss measurements [27, 28].

2.4 Reliable Multicast: Router Support

It has long been noted that the problem of reliably multicasting a packet to a large group of receivers becomes noticably simpler if the network acts to ensure reliability. Recently, a number of proposals have been made to add such reliability into the network. One of these, PGM [2], provides a close fit for our requirements for a watermarking scheme.

PGM is "a reliable transport protocol for applications that require ordered, duplicate-free, multicast data delivery from multiple sources to multiple receivers". To achieve reliability, receivers send negative acknowledgements (NAKs) which are reliably propogated up the multicast distribution tree towards the sender, with the aid of the routers. Retransmissions of lost data are provided by the sender or by designated local retransmittors.

Two mechanisms are incorporated to prevent NAK implosion: on detecting loss, receivers employ a random backoff delay before sending a NAK with suppression of their NAK is one is received from another receiver in this time. In addition, routers which receive duplicate NAKs from multiple downstream links eliminate the duplicates, sending only a single NAK up towards the source.

The result is a timely and efficient means by which NAKs can be returned to the source of multicast data,

allowing either retransmission of that data or addition of FEC to the stream to ensure reliable delivery.

In addition to providing an efficient NAK delivery and summarisation service, PGM offers a number of end-to-end options to support fragmentation, sequence number ranges, late joins, time-stamps, reception quality reports, sequence number dropout and redirection.

Of interest to us is the sequence number dropout option. This allows placement of "intermediate application-layer filters" in routers. Such filters allow the routers to selectively discard data packets and convey the resulting sequence number discontinuity to receivers such that sequencing can be preserved across the dropout, and to suppress NAKs for those packets intentionally discarded. The operation of these filters is not defined by PGM. In later sections of this paper, we describe semantics for these filters suitable for watermarking multicast streams.

2.5 Reliable Multicast: Layering & FEC

The use of packet-level FEC to recover from loss is well-known. For every k data packets, $n - k$ FEC packets are generated, for the transmission of n packets over the network. For every transmission group of n packets it is necessary to receive only a subset to reconstruct the original data.

There are a number of means by which these FEC packets may be transmitted. The three primary means are by piggy-backing them onto previous packets, sending them as part of the same stream but with a different payload type indicator or sending them as a separate stream.

Sending FEC packets within the same stream as the original data has the advantage of reducing overheads (routers only need keep state for a single stream), but forces all receivers to receive the FEC data in addition to the original data. If a receiver is not experiencing loss, this is clearly wasteful.

Sending the FEC data on a different stream has greater overhead (because routers need keep state for multiple flows), but allows for greater flexibility. Those receivers which are not experiencing loss do not join the multicast group transporting the FEC stream, and hence do not receive the FEC data; varying amounts of FEC can be supplied, layered over a range of groups, giving different levels of protection; or enhancement layers can be provided – not FEC but additional data to improve the quality of the stream for those on high capacity links who are not experiencing loss.

This use of layered transmission to provide either FEC or differing quality has been studied by a number of authors [29, 30] and shown to perform well.

3 Overview of the Watercasting Protocol

Given that the loss signature of a receiver corresponds to its position in the network, it should be possible to use this as a simple form of digital watermark. The pattern of degradation in a stream, caused by lost packets, will be different for most receivers provided there is a non-zero packet loss rate in the network (see section 2.3).

There are four problems with this technique:

1. A receiver may neglect to send a loss signature back to the sender, escaping notice by the watermarking scheme entirely.
2. Lost packets cause degradation of the delivered stream. A network which drops enough packets to make this watermarking technique successful will likely provide insufficient quality for most uses.
3. A receiver may collude with another receiver to repair the loss, hence defeating the watermarking scheme.
4. A receiver may easily defeat the watermarking scheme by dropping additional packets (possibly transforming the stream to match that received by another receiver).

Ensuring that a receiver returns its loss signature to the sender is clearly an impossible task in the traditional Internet environment with smart end-points and dumb routers. However, if an active network is assumed it becomes possible for the last-hop router before the receiver to return a loss signature to the sender. If the installation of this active element into the network forms part of the multicast tree setup procedure, we may ensure

that the loss signature of all receivers is returned to the source.

Further, the active network elements can conspire to ensure that all receivers see unique loss patterns, rather than leaving this to chance. Instead than relying on the loss signature on a particular branch in the multicast forwarding tree being unique, the position of a node in the tree is used as input to a random sequence generator, to determine which packets to drop in order to ensure a unique loss pattern for each node.

The proposed use of active network components is not unique to our scheme; a number of reliable multicast protocols have been developed which would benefit from support within the network (whether within the router or switch fabric, or in application level relays). This support typically takes the form of filtering, summarisation and subcasting abilities: exactly the requirements for our scheme (see section 2.4).

The assumption of an active network leaves three barriers to the development of an effective watermarking solution: degradation of the stream by packet loss, collusion attacks by multiple receivers to repair the stream, and the ease of breaking the protection by dropping additional packets. These three problems are related, and have a common solution.

A typical counter to the problem of packet loss in a multicast network is to add forward-error correction (FEC) data to a media stream (section 2.5). This allows repair of the stream if some fraction of the packets are lost. We modify this approach by sending FEC data which is subtly different to the original data, such that a stream repaired using this FEC data will differ based on the observed loss pattern, but will not be noticably degraded.

This is altering of the media stream is generally straightforward, although content specific. For example, in an H.261/H.263 stream, we could alter the DC component by different amounts per packet and send additional intra-coded blocks. In audio, we may frequency shift some packets by a small amount, or mask out certain small frequency ranges. It is vital that the set of transformed packets resulting from one packet cannot be used to recreate the original packet, otherwise a collusion attack could produce a non- watermarked version of the data. Likewise, the watermark must be resistant to a wide range of transforms, such as the introduction of jitter or re-sampling [9].

The active network elements therefore *subtract* FEC packets. Rather than ensuring a unique loss pattern at each receiver, they ensure a unique pattern of packets is received. This may be implemented using the PGM sequence number dropout option and application layer filters, as described in section 2.4.

This solves the quality degradation problem, since some version of each packet is received by every participant the reception quality is no worse than that provided by the underlying network, although each receiver sees a slightly different stream. Receivers can no longer collude to generate a stream without the watermark (the result will simply be a combination of their watermarks, enabling identification of the conspirators). Finally, discarding additional packets simply results in a degraded stream with the watermark still present.

The result is a relatively simple means of watermarking multicast data: the source sends multiple, subtly different, copies of each packet. The branch points in the network discard packets, such that the stream delivered to each receiver is unique.

4 Implementation Strategy

Our watercast protocol operates in two parts. A client wishing to join a protected multicast group first identifies itself to the source and joins the group, updating the router state in the tree. The routers accordingly filter the data passing down that tree.

A client wishing to receive content from a server first performs a unicast authentication with that server. After convincing the server they are a valid subscriber the server gives the client the current session keys for the requested media and a receiver identification key.

The receiver identification key is supplied to the last hop router by the receiver when it joins the session. Each router in the distribution tree must pass up to the source its location in the tree, and addresses of its downstream routers. *encrypt to keep topology secret as far as possible?* This information is necessary to allow the source to determine the correct watermark for each receiver. There is no need for the source to be informed of the presence of receivers.

When this information reaches the source, it can check that new routers on the path back to the receiver are not on a blacklist of previously-discovered traitors?

For a multicast distribution tree of depth d, the source generates a total of n copies of each packet; such that $n > d$. Each group of n alternate packets is termed a transmission group.

On receiving the packets which form a transmission group, a router forwards all but one of those packets out of each downstream interface on which there are receivers. The choice of which packet to discard is made at random, with the random sequence keyed by the position of the router in the distribution tree and the interface address.

The last hop router in the distribution tree will receive $n-d$ packets from each transmission group. Exactly one of these packets will be forwarded onto the subnet with the receiver(s). The choice of which packet is forwarded is determined randomly, keyed with the position of the router in the tree, the interface address and the receiver identification key.

This filtering process is illustrated in figure 1. In this example, the source will generate $n > 4$ distinct versions of each packet to form a transmission group; label these ABCDE for example. At router 0 these are filtered, passing ABDE to receiver 00 and ACDE to router 01. At router 01 the packets are filtered again, with ACE being passed to router 010. Since this is the last hop router before receiver R2, router 010 does not just filter out a single packet, it randomly selects one packet, E, to pass to the receiver.

It is clearly possible that multiple receivers receive the same version of a particular transmission group. An analysis of how likely this is, and how many transmission groups must be received before the signature of received packets is unique is made in section 5.

The media payload is encrypted. The receiver also receives a decryption key from the source by the same means that it receives the receiver identification key. Media packet headers are *not* encrypted, since routers need to use the sequence numbers in the packets to determine which packets to discard.

The encryption of the media payload prevents unauthorised receivers from snooping on the packets. The

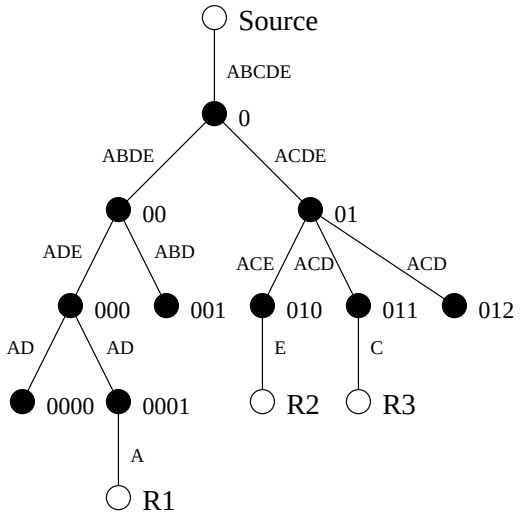

Figure 1: Filtering transmission groups to obtain a unique watermark

watermark may be used to detect illegal redistribution of the decrypted payload by legitimate receivers.

In effect, the tree topology is the secret used by the source to do the watermarking. Participating routers should therefore refuse requests to reveal any part of that topology. Even if some routers and clients collude, they would need a conspiracy from a client right up to the source to discover anything useful.

The selective discard function aims to provide the multicast routers and their clients the minimum degree of freedom possible in order to facilitate the later tracing of cheating routers or users. Every router in the tree indelibly affects the stream by dropping certain packets; thus a cheating router should be identifiable as every user downstream from that router would need to collude to reproduce the watermarked data passing through the router. The higher up the tree, the less likely this is; the lower down the tree, the easier to eliminate targets from an investigation.

We keep as much of the processing as possible at the source to simplify the router protocol. For each watercast stream a router is processing, it only needs to store the sequence state to allow it to drop the appropriate packet once each transmission group. Routers rapidly increasing in processing power, and more sophisticated versions of our filter could produce far more complex wa-

termarks. With access to the plaintext of the scheme, they could do a lot more than just drop packets. But the corresponding drop in confidentiality is likely to outweigh the benefits of such a scheme.

We place a slightly heavier burden on the server. It needs to appropriately modify the outgoing stream and inject sufficient redundant packets into it to maintain quality for all clients while allowing enough packets to be dropped to watermark each client's stream uniquely. It also needs to store enough information to enable it to later reconstruct the path to a watermark found in a recovered media clip.

Need an analysis of how, given a particular watermark, we can determine which receiver this stream came from...

5 Analysis

The length in packets of a watermark in a stream will be

$$\text{length} = \log_2 \max \Sigma(fanout + slack)$$

This can vary continuously according to the topology of the multicast tree at the time of transmission of the marked data, something known only to the server. This makes it difficult even for conspiracies of users and routers to remove the watermark or alter it to implicate someone else.

Generally, the longer the media clip of interest, the harder it will be for enough watermark data in it to be damaged to reduce the possibility of detection to an acceptably low value for the pirate. Our ultimate design criterion must be the ability to convince a court of law that a piece of marked media with no other links to a suspect originated from that suspect.

Our protocol is inherently scalable. The resource requirements on the server are roughly $O(\log n)$, while demands on routers from new clients are linear but only up to the limit of the number of interfaces they are willing to serve. *is the demand on the network therefore also $O(\log n)$?*

The most efficient stage of multicast might appear to be a problem for watercast. When a multicast packet's final destination is a subnet with broadcast capability (such as an ethernet LAN), any host on that network can receive the same packet with no extra effort. Non-subscribers on a network can intercept such packets, but do not have the decryption key needed to read them. But two legitimate subscribers on the same LAN will receive the same watermarked data.

We would content that this is a small problem. It is unlikely there would be more than two or three customers for exactly the same programme on the same network in the brave new multi-channel world of the future. If one of them illegally resold data, it would be traceable to those few people, which should be sufficient for the normal police investigative process to begin. Pirates are also likely to repeat their crimes, and a set of different resold programmes should allow the identification of the one miscreant who was receiving those signals on that network at those times. The network adminstrator may also be able to provide further information to help in the investigation.

Collaborative conspiracies are always a difficult problem for watermarking schemes. Groups of users can attempt to combine their different watermarked versions of the same piece of data in a way that removes or at least damages the watermark. The simplest way to do this is perform 'bit voting': set each bit in the reconstructed piece of data to be that which is most prevelant in the same bit in the set of watermarked files. This is usually fatal to simple schemes, and can damage more sophisticated watermarks.

We have used signal processing techniques to mark data in an attempt to defeat collaborative and individual attacks such as introducing jitter and re-sampling [9]. But the main contribution of our paper is that an active network can perform part of the watermarking function. Even if the specific transforms we have used can be defeated, we hope that it should be possible to simply plug in others more resistant to attacks, preserving the validity of our approach.

6 Conclusions and Future Work

We have outlined a general idea for watermarking AV data using active network components, and a specific method to perform that task. Our method leverages schemes such as PGM that are being developed to pro-

vide other services such as reliable multicast and filtering in active networks.

We are now performing simulations to model the performance of our method. Factors such as the size of transmission groups and complexity of filtering algorithms will have large effects on the performance of the system, and we hope to use our simulations to fine tune these parameters. We will also investigate optimisations such as increasing capacity near the multicast root for given sizes of receiver sets and the depth and breadth of the tree.

Watercasting has wide applicability to the protection of any data that is distributed to a large number of people via a network. While we have focussed on audiovisual data, other information such as stock prices could be equally covered with the development of appropriate transforms. Indeed, an appropriate software architecture would allow small pieces of transform code to be plugged-in to our system to extend it with minimum effort.

The authenticity of such data may be more important to clients than that of a video broadcast. It would be trivial to put a public-key Authentication Header in each packet [17] and so assure clients of the information's integrity and origin.

Watermarking technology is still in its infancy. Petitcolas et al. hoped that their attacks on first-generation algorithms would lead to an improved second generation, and so on [9]. We hope our system is reasonably resistant to the attacks they designed, but we will no doubt see further ones developed.

Our design criteria are slightly less robust than those of, for example, the International Federation for the Phonographic Industry (IFPI), who required a watermark that could not be removed or altered "without sufficient degradation of the sound quality as to render it unusable" [31]. Our definition of unusable is not unlistenable or even unsellable, but simply perceptually intrusive enough to justify paying for the original rather than a pirated copy. We plan to use tools developed for measuring subjective audio quality [32] to assess the impact of removing the watermarks we develop.

We also believe that the best use for our system is the transmission of 'live' data. As Barlow observed [19], the value of such transmissions drop rapidly as they age. The live TV rights to a World Cup soccer final are worth many tens of millions of pounds, but drop dramatically once the game is finished and everyone knows the result.

Attacks on complex marking schemes such as echo hiding can require computationally intensive brute-force calculations [9]. A useful watermark would allow arbitrary variation of the processing power needed to remove it, a factor that could be tuned according to how quickly protected information aged.

Therefore, even if our watermarking scheme can be defeated, as long as it takes a reasonable amount of time to do so it will have achieved its main objective — to prevent large profits being available from the illegal redistribution of content.

7 Acknowledgements

Thanks to the IRTF Reliable Multicast Research Group and the network multimedia research group at UCL for discussion that led to some of these ideas.

References

[1] H. Schulzrinne, S. Casner, R. Frederick and V. Jacobson, "RTP: A Transport Protocol for Real-time Applications", IETF Audio/Video Transport Working Group, RFC1889, January 1996.

[2] T. Speakman, D. Farinacci, S. Lin and A. Tweedly, "PGM Reliable Transport Protocol Specification", Work in Progress, February 1999.

[3] M. Macedonia and D. Brutzman, "Mbone Provides Audio and Video Across the Internet", IEEE Computer, Vol.27 No.4, April 1994, pp. 30-36.

[4] A. Ballardie and J. Crowcroft, "The Importance of Security in Wide-Area Multicast Communication", In proceedings of the 1994 ISOC Security Conference.

[5] A. Ballardie, "Scalable Multicast Key Distribution", RFC 1949, May 1996.

[6] C. Dwork, J. Lolspiech, M. Naor, "Digital Signets: Self Enforcing Protection of Digital Information", in Proceedings 28th Annual ACM Symposium on the Theory of Computing, 1997.

[7] M. Naor and B. Pinkas, "Threshold Traitor Tracing", in proceedings CRYPTO'98.

[8] R.J. Anderson and C. Manifavas. Chameleon – A New Kind of Stream Cipher. Fourth Workshop on Fast Software Encryption, 1267:107-113, January 1997.

[9] F. A. P. Petitcolas, R. J. Anderson and M. G. Kuhn, "Attacks on copyright marking systems", Second Workshop on Information Hiding, Portland, Oregon, April 15-17, 1998.

[10] M. Handley, "An Examination of Mbone Performance", USC/ISI Research Report ISI/RR-97-450, April 1997.

[11] M. Yajnik, and J. Kurose, and D. Towsley, "Packet Loss Correlation in the Mbone Multicast Network", IEEE Global Internet Conference, November 1996.

[12] G. Carle and E. Biersack, "A Survey of Error Recovery Techniques for IP-Based Audio-Visual Multicast Applications", IEEE Network, November/December 1997.

[13] C. Perkins, O. Hodson and V. Hardman, "A Survey of Packet Loss Recovery Techniques for Streaming Audio", IEEE Network, September/October 1998.

[14] B. Levine and J. J. Garcia-Luna-Aceves, "A Comparison of Reliable Multicast Protocols", ACM Multimedia Systems, August 1998.

[15] C. Kurak and J. McHugh, "A cautionary note on image downgrading", Computer Security Applications Conference, San Antonio, Texas, 153–159, December 1992.

[16] S. Kent and R. Atkinson, "IP Encapsulating Security Payload", RFC 2406, November 1998.

[17] S. Kent and R. Atkinson, "IP Authentication Header" RFC 2402, November 1998.

[18] T. Hardjono, B. Cain and N. Doraswamy, "A Framework for Group Key Management for Multicast Security", Work in progress, July 1998.

[19] J. P. Barlow, "The economy of ideas", Wired 2(3) 85, March 1994.

[20] V. Jacobson, "Multimedia Conferencing on the Internet", Tutorial slides, ACM SIGCOMM'94, London, August 1994.

[21] D. Meyer, "Administratively Scoped IP Multicast", IETF Mbone Deployment Working Group, RFC2365, July 1998.

[22] M. Handley and D. Thaler, "Multicast-Scope Zone Announcement Protocol", IETF Mbone Deployment Working Group, Work in progress, October 1998.

[23] J.-C. Bolot and A. Vega-García, "The Case for FEC-Based Error Control for Packet Audio in the Internet", ACM Multimedia Systems.

[24] M. Yajnik, S. Moon, J. Kurose and D. Towsley, "Measurement and Modelling of the Temportal Dependence in Packet Loss", To appear in IEEE Infocom, 1999.

[25] S. Moon, J. Kurose, P. Skelly and D. Towsley, "Correlation of Packet Delay and Loss in the Internet", Technical Report 98-11, Department of Computer Science, University of Massachusetts, Amherst, MA 01003, USA.

[26] B. Levine, S. Paul and J. J. Garcia-Luna-Aceves, "Organizing Multicast Receivers Deterministically by Packet-Loss Correlation", Preprint, University of California, Santa Cruz.

[27] R. Cáceres, N. G. Duffield, J. Horowitz, D. Towsley and T. Bu, "Multicast-Based Inference of Network-Internal Characteristics: Accuracy of Packet Loss Estimation", in proceedings IEEE Infocom, 1999.

[28] S. Ratnasamy and S. McCanne, "Inference of Multicast Routing Trees and Bottleneck Bandwidths using End-to-end Measurements", in proceedings IEEE Infocom, 1999.

[29] L. Vicisano, L. Rizzo and J. Crowcroft, "TCP-like congestion control for layered multicast data transfer", in proceedings IEEE Infocom, 1998.

[30] S. McCanne, V. Jacobson and M. Vetterli, "Receiver-driven Layered Multicast", in proceedings ACM SIGCOMM'96, August 1996, Stanford, CA.

[31] International Federatioin of the Phonographic Industry. Request for proposals — Embedded signalling systems issue 1.0, June 1997.

[32] A. Watson and M. A. Sasse. Measuring Perceived Quality of Speech and Video in Multimedia Conferencing Applications. Proceedings of ACM Multimedia '98, Bristol, England, 55-60, September 1998.