Deployment Issues for Multi-User Audio Support in CVEs

Milena Radenkovic, Chris Greenhalgh, Steve Benford
School of Computer Science and IT
University of Nottingham
Nottingham
NG8 1BB, UK
{mvr, cmg, sdb}@cs.nott.ac.uk

ABSTRACT

We describe an audio service for CVEs, designed to support many people speaking simultaneously and to operate across the Internet. Our service exploits a technique called Distributed Partial Mixing (DPM) to dynamically adapt to varying numbers of speakers and network congestion. Our DPM implementation dynamically manages the trade-off between congestion and audio quality when compared to the approaches of peer-to-peer forwarding and total mixing in a way that is fair to the TCP protocol and so operates as a "good Internet citizen". This paper focuses on the large-scale deployment of DPM over wide area networks. In particular we raise and examine the issues when deploying DPM within the context of large dynamic environments. We argue that DPM paradigm remains feasible and desirable in such environments.

Categories and Subject Descriptors

C.2.3 Network Operations [COMPUTER-COMMUNICATION NETWORKS]: *Network management, Network monitoring.*

C.2.4 Distributed Systems [COMPUTER-COMMUNICATION NETWORKS]: *Client/server, Distributed applications.*

General Terms

Algorithms, Management, Measurement, Performance, Design, Reliability, Human Factors, Theory.

Keywords

CVEs, real-time audio, simultaneous speakers.

1. INTRODUCTION

This paper is concerned with supporting 'natural' audio communication in Collaborative Virtual Environments (CVEs) across the Internet. Recent experience with CVEs, for example to support large on-line communities and highly interactive social events, suggest that in the future there will be applications in which many users speak at the same time (see section 2). This paper describes scaleable audio service for CVE applications that can support situations in which relatively large numbers of mutually aware users (potentially tens or even hundreds) are speaking simultaneously, but where available resources, especially

bandwidth, are limited and dynamically changing. This service is based on a method called DPM. This paper focuses on DPM deployment over wide area networks. In particular we raise and examine the issues when deploying DPM within the context of large dynamic environments. We argue that DPM paradigm remains feasible and desirable in such environments. This paper promotes the idea of congestion control over multiple audio streams as a necessary complement to existing techniques for single stream adaptation.

Section 2 explains the problem domain and identifies some key design and use decisions that have shaped our approach. Section 3 discusses related work and section 4 describes Distributed Partial Mixing as the general approach to multi-party audio distribution. Section 5 discusses issues in large scale deployment of DPM. Sections 6 and 7 propose two deployment schemes for DPM. Section 8 gives final conclusions.

2. MULTI SPEAKER AUDIO

There are many situations in everyday life in which large groups of people "speak" at the same time. Audiences and crowds of spectators at performances and sports events provide the most extreme examples, with thousands of people engaging in activities such as cheering, chanting and singing together. However, any relaxed social setting that involves sizeable groups of people is also likely to involve significant numbers of simultaneous speakers; for example, interruptions, non-verbal speech cues [4], shouting out answers to questions, dynamically forming conversing sub-groups within a large space [11].

Furthermore, collaborative virtual environments (CVEs) can support large on-line communities and highly interactive social events such as multi-player games and inhabited television [6]. These applications have fast pace of interaction, have M-to-N distribution of audio streams where M is almost equal to N. For example, recent work on inhabited television has focused on enabling public participation in on-line TV shows within shared virtual worlds [6]. As part of a public experiment in inhabited TV called Out of This World (OOTW), patterns of user activity, including audio activity, were studied by statistically analysing system logs [5]. The results showed that overlapping audio transmissions from several participants were common for this event. Indeed, during a 45 minute show, there were several minutes when all 10 of the participants were generating audio traffic at the same time. Periods of high activity included teams shouting instructions to one another during action games and also shouting out answers to questions (OOTW was a gameshow). Similar analyses of patterns of audio activity in other CVE applications and platforms, for example virtual teleconferencing in

the DIVE system [3], have also revealed significant periods when several participants are simultaneously generating audio traffic.

Not all of this activity is necessarily verbal; it may also include non-verbal utterances and background noise. However, we argue that non-verbal utterances form a significant part of human communication. For example, studies of social interaction in CVEs have noted how communication in a virtual world can be influenced by events in participants' local physical environments [1]. Obtaining awareness of these events through overheard background audio might help participants account for actions in the virtual world. Similarly, when discussing patterns of usage for the Thunderwire audio media-space, Hindus *et al.* note that it might be better to consider the number of live microphones, rather than the number of active participants [10].

For these reasons we have rejected approaches to audio management that restrict the use of the audio channel, including explicit and implicit floor control, and explicit control actions (e.g. push to talk). Instead we assume that our ideal audio service will use silence suppression or even continuous transmission, and we have designed our audio service to cope with potentially many simultaneous audio streams.

3. RELATED WORK

There is a range of techniques that are concerned with efficient distribution of single audio streams. These include network supported multicasting, e.g. as used for various multiparty audio and video tools (e.g. RAT [7]), layered multicast, which is based on layered media encodings [14], and various established compression algorithms. There are also various proposals that – like our own – include processing elements at intermediate points within the media distribution tree. These include filter and proxy mechanisms proposed in [2],[15],[21]. These techniques affect only the bandwidth of a single stream and do not operate across multiple streams. Therefore, the amount of audio data and processing for these techniques rises proportionally with the number of simultaneous speakers. Furthermore, [12] argue that although layered (or adaptive) encoding is possible for sampled speech, the transmission bandwidth range is significantly more restricted than for video, limiting the potential for bandwidth adaptation of a single audio stream.

There are also a number of approaches that are based on some form of mixing to reduce the number of audio streams in the network. Mixing multiple streams involves digitally summing the corresponding audio samples from the incoming streams and then normalising the result. In general, performing mixing (e.g. in a server, filter or proxy) in the network is the only mechanism that can actually decrease the number of audio streams. Mixing can thus drastically reduce network traffic for many simultaneous speakers (sources). This makes it efficient in terms of its support for efficient distribution of audio streams in the network. Mixing also imposes no constraints on individual speakers (compared with floor control's "gagging" of users).

However there are also negative aspects to mixing, which mainly concern the user's point of view including: increased noise of the mixed signal, additional delays compared to direct communication, more components in the network with corresponding hardware, organisational and real-time control requirements. A further major disadvantage of mixing for CVEs is that mixing introduces loss of potential spatialisation and other aspects of individual listener control over individual audio streams. Previous research has shown that providing spatialised audio is a significant factor in engendering a sense of presence in a virtual environment [9]. Note that mixing is irreversible and the individual streams cannot be split at a later stage.

In a centralised (total) mixing architecture every audio stream is sent to a centralised audio mixer that mixes multiple streams into a single stream that is then redistributed to each listener. However, this prevents each listener from creating their own personal mix (e.g. for spatialisation or distance attenuation). Also, the server becomes a potential bottleneck in the system, since it (and its nearby network) has to deal with, mix and distribute $O(n)$ streams.

Client-specific mixing has been proposed in the SPLINE system [20]. One or more low-bandwidth users connect to a specialised access server that interfaces to the main system. This access server provides a customised audio mix for each connected user, which is sent directly to them. This approach provides a separate mix to each listener, giving greater flexibility, though at the cost of more work for the access servers (since sending two copies of the same mix is easier than computing and sending two different mixes). However, this approach still requires (like total mixing) that the access servers deal with and mix all of the available audio streams, and it assumes that the network between the access servers can cope with all of the audio streams. This approach can be viewed as a subset of our own approach.

We argue that these approaches perform 'excessive' mixing, in that they mix audio streams even in the presence of spare bandwidth. They also lack explicit support for network monitoring and adaptation and so may cause network congestion and excessive packet loss.

4. DISTRIBUTED PARTIAL MIXING

Consequently our design has to respond to three primary challenges. First, how can the audio streams be mixed in a way that it is adaptive and friendly towards other traffic in the network, supports heterogeneous networks and minimises the packet loss experienced? Second, how can we do no more mixing than is absolutely necessary i.e. keeping the maximum possible number of independent streams (and thus allowing maximum flexibility of audio presentation to the end user) while having an efficient aggregation of audio streams? Third, how can (if at all) such a service be deployed over the current Internet infrastructure? In this section we summarise the approach of Distributed Partial Mixing [16]. In the remainder of the paper we consider its potential wide-area deployment.

4.1 Distributed Partial Mixing

Consider a simple Internet scenario is assumed comprising two LANs and dial-up users connected via WAN connections with unknown properties as illustrated in Fig 1. A Collaborative Virtual Environment (CVE) application is being used over this network, and a number of people (clients) distributed around both LANs and their homes are connected to the CVE, each person having an open microphone and being able to hear the audio from the rest of the people at any time.

Distributed Partial Mixing requires that a number of Partial Mixing components also exist in the network (e.g. as dedicated servers, or as roles performed by a subset of the clients). The clients typically communicate via these Partial Mixers, sending their audio to – and receiving other clients' audio from – a 'nearby' partial mixer. The partial mixers in turn form a fully connected network (e.g. a tree) to achieve end-to-end distribution of the audio streams.

Unlike total mixing (which is based on mixing the whole set of received streams into a single output stream) partial mixing dynamically chooses to mix only a subset of the available audio streams at any given time and place, and forwards this along with the rest of the un-mixed streams. In this way instead of producing a single output stream in all cases, partial mixing produces varying numbers of streams in different situations.

Figure 1 illustrates a scenario in which there are multiple partial mixing stages between the end systems. The figure shows the streams that get mixed at different stages in the system. For example, the network between Partial Mixer (PM) A and B is experiencing low loss rates, and so all five streams (*a-e*) are forwarded without mixing. However the network between PM C and D is experiencing higher loss rates, and is mixing the three audio streams (*f-h*) into a single combined stream. Client *j* is receiving a single fully mixed stream from PM E, perhaps because of tail-link congestion, or because of client processing limitations. Whereas client *i* is receiving maximally separated audio streams (only mixed where this was forced earlier in the network). Note that in this example partial mixers A and B are client machines, and partial mixers C, D and E are dedicated servers.
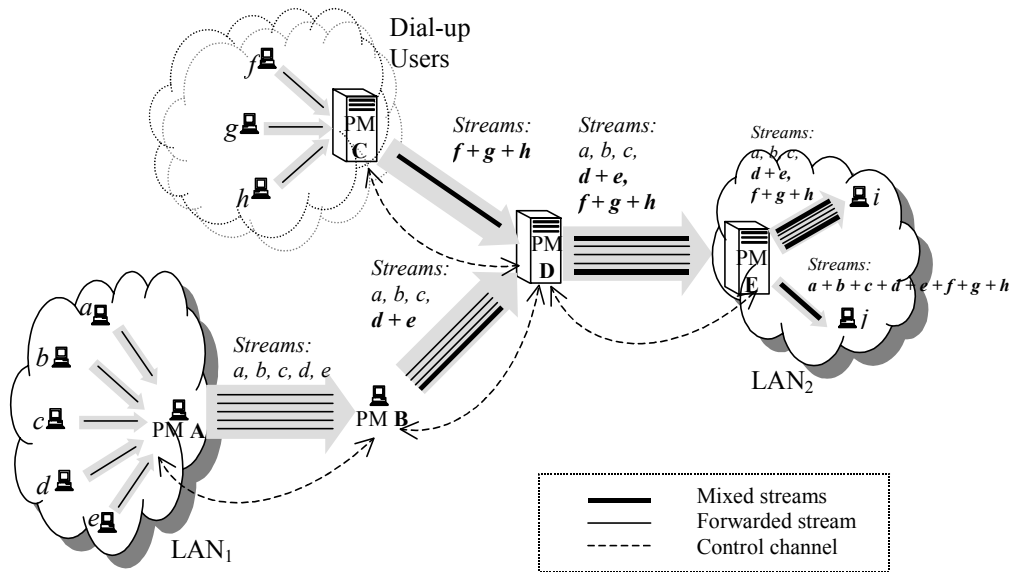

Figure 1. Sample scenario of DPM in use.

4.2 DPM Over a Single link

DPM was prototyped in MASSIVE-3 [7] CVE system. An extensive set of experiments over single link was done to demonstrate the behaviour of distributed partial mixing against non-adaptive traffic and TCP traffic.

Two quantifiable aspects of audio quality were considered: the level of packet loss experience, and the degree of spatialisation available to the end user. The packet loss determines the quality of the heard audio. The degree of spatialisation is determined by the number of independent audio streams that are delivered to the listener, and that can therefore be independently localised within their own subjective audio mix.
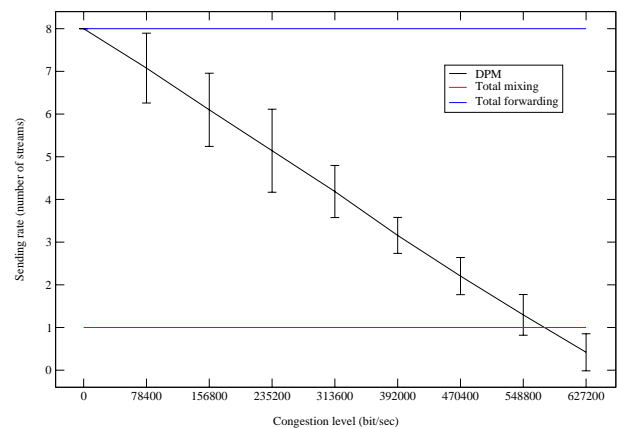

Figure 2. Degree of spatialisation for various distributed audio schemes (number of independent audio streams) against competing traffic

At low, moderate and high steady state congestion levels, distributed partial mixing has a relatively high mean throughput and small variations of throughput (Fig. 2), while preserving the packet loss under 5% (Fig. 3). Fig. 2 shows not only mean value of the number of independent streams sent by DPM mixing source

within congestion level but also variations in bandwidth with two standard deviations. Note that these graphs are updates from [17] to reflect refinements in rate control, stable packet loss measurement and TCP fairness in DPM design choices.
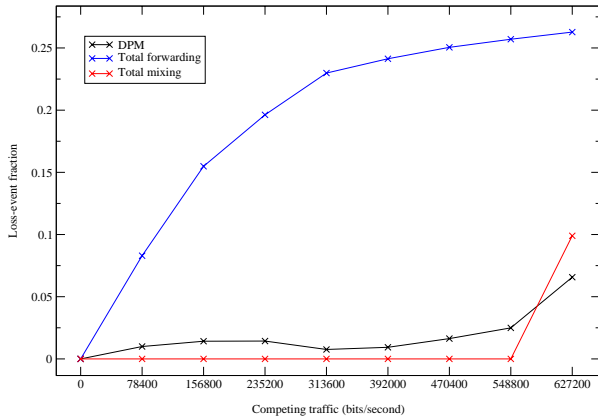

Figure 3. Packet loss rates for various distributed audio schemes

Experiments against adaptive traffic demonstrate the dynamic behaviour of DPM. The results demonstrated that distributed partial mixing co-exists acceptably well when sharing congested links with other adaptive traffic: DPM consumes its fair share of the bandwidth when competing with multiple TCP flows and other DPM flows. More extensive results and more detailed discussion about them can be found in [17].

5. DEPLOYMENT

The main new contribution of this paper is to consider the potential large-scale deployment of distributed partial mixing over wide area networks. The goal of this section is to raise and examine the issues when deploying DPM within the context of large dynamic environments. The fundamental question is whether or not the DPM paradigm remains desirable or even feasible in such environments.

5.1 Steady state DPM deployment

In order to deploy DPM over a wide area network we must establish a fully connected topology of DPM clients and partial mixing components that is compatible with the underlying network topology and that avoids loops and echoes, minimises the delays to the these tolerable by the application and minimise fan in/out of each server to mach server processing and I/O capabilities.

We propose the use of a shared tree of DPM servers and end users, where the nodes of the tree can perform distributed partial mixing as an effective basis for wide area DPM topology. The major benefits of shared trees are that they support many-to-many communication, low overheads and simple to build. Such topology can achieve global optimisations in terms of the four addressed issues, i.e. it copes with loops and echoes and can minimise the delays and fan in/out using a variety of heuristics. Even thought basic algorithms for shared trees do not guarantee bounds on height and breadth of the tree, there has been some work emerging to address these problems for real time applications [13], [19], [22].

Note that shared trees have traditionally been used in multicast and they suffer from traffic concentration on the links shared by multiple senders. Using distributed partial mixing in the nodes enables the use of shared trees even for large number of senders that send large volumes of data possibly along the same network links without the risk of traffic concentration and congestion

5.2 Issues in realising large scale DPM

To actually establish and maintain such a configuration of DPM clients and servers we must address the following issues:

- Deployment domain
- DPM placement
- DPM initiation
- DPM discovery and advertisement
- DPM topology determination
- Managing the process of topology determination
- Support for re-configuration

We consider each of these issues in turn. We then describe two contrasting solutions that make radically different choices for each of these issues. Note that other intermediate approaches are also possible.

5.2.1 Deployment Domain

At a general level we can distinguish two main kinds of potential deployment domain: managed and unmanaged. Specifically, these domains differ in the kind of access rights they allow, the amount of available knowledge about the network and other systems, as well as the (assumed) stability and reliability of the network links. The target deployment domain will affect all of the other issues presented in this section.

5.2.2 DPM Placement

DPM placement refers to where in the network DPM functionality is placed (how, and on which machines). General requirements for a DPM system are that DPM servers are highly available and easily accessible to the clients, but also positioned near to problematic links where they can perform effective congestion control. However this is very challenging when no prior knowledge of the client distribution and network is available. For example, placing too many DPM servers in the network might result in under-utilisation and inefficient use of resources, and vice versa. Also, only placing DPM servers near links with no congestion will not result in efficient congestion control of the whole DPM system.

5.2.3 DPM discovery and advertisement

This refers to the question of how a DPM server learns about (and becomes known to) the entire system (or the part(s) of the system that are relevant to it). This also includes the question how much knowledge a single DPM server should have about the entire system. This has a direct impact on the storage space and memory of each DPM, as the knowledge that each has about the system has to be stored and processed by it. The dynamics of DPM discovery and advertisement can influence the efficiency of the DPM system as a whole.

5.2.4 DPM initiation

DPM initiation refers to activating a DPM server, e.g. to start monitoring the neighbouring links and to performing adaptive partial mixing. More specifically, a DPM server can become

active in various ways and at various times, e.g. on-demand or pre-configured. Ideally, DPM servers should be activated promptly when they are needed and by those who need them. If, on the other hand, DPM servers are not active when they are needed or this activation process takes a long time then the responsiveness of the whole DPM system may be compromised. The quantity of session initiation traffic can affect the congestion control of the whole system because it might cause additional congestion itself.

5.2.5 DPM topology determination

DPM topology determination refers to deciding on a suitable topology of DPM servers (i.e. topology does not need to be based on trees). The problem is twofold: Firstly, how are the connections between DPM servers and clients and other DPM servers determined i.e. what are the metrics for determining the cost of a connection? This refers to the underlying aggregation metrics/techniques used by schemes that determine the topology. Secondly, are global or local optimisations of the topology made when determining it?

5.2.6 Topology Control

This refers to how the overall process of managing the topology is realised. There are two extreme approaches for realising this process: centralised and distributed. Efficiency, scalability and robustness of the controlling process are of great importance for the stability and overall performance of the DPM system. The level of overheads (e.g. controlling messages, that include initiation messages and various update messages) in this process can also be an issue – ideally the controlling process should scale with the number of clients and DPM servers and the updates should be sufficiently frequent to allow accurate decision making. Robustness of the DPM controlling process to various faults in the system will directly impact the stability of the whole DPM system.

5.2.7 Support for re-configuration

This issue refers to whether and how the system can respond to changes in the network and user membership. At the most general level, the topologies of DPM servers can fall into two categories: non-adaptive and adaptive. This is important given that today's networks are not very stable, either in terms of user membership (session members can be expected to continually join and leave during the session) or in terms of network link reliability (network links can be expected to fluctuate in link quality).

In the next two sections we describe two contrasting approaches of dealing with the issues described in this section.

6. FULLY CENTRALISED DEPLOYMENT OVER MANAGED NETWORKS

This section illustrates one particular approach that can be used in a fully managed scenario with a centralised coordinator (manager) that assumes full knowledge of the network, DPM servers and clients, and has all the necessary access rights to configure them.

For example we might assume that an Inhabited TV show [6] (a popular soap opera, fashion show, or football game) is shown over cable TV, and that only the subscribed viewers can view it and participate in it. Some viewers choose to become more active participants in the virtual world that they are seeing on their televisions ("inhabit" the TV show), and it is further assumed that all the network communication with the other participants will be done over fully managed corporate network. Similar assumptions

can hold for a large teleconferencing application that has to be scheduled in advance, and where participants have to be registered (subscribed) before the beginning of the session.

The centralised controller plays the role of an ideal observer that has centralised global knowledge of the whole system at every point of time. This knowledge is available to the centralised controller via multiple databases including topological and user databases, and it enables the controller to create connections among the nodes and their topology. This is illustrated in Figure 5.
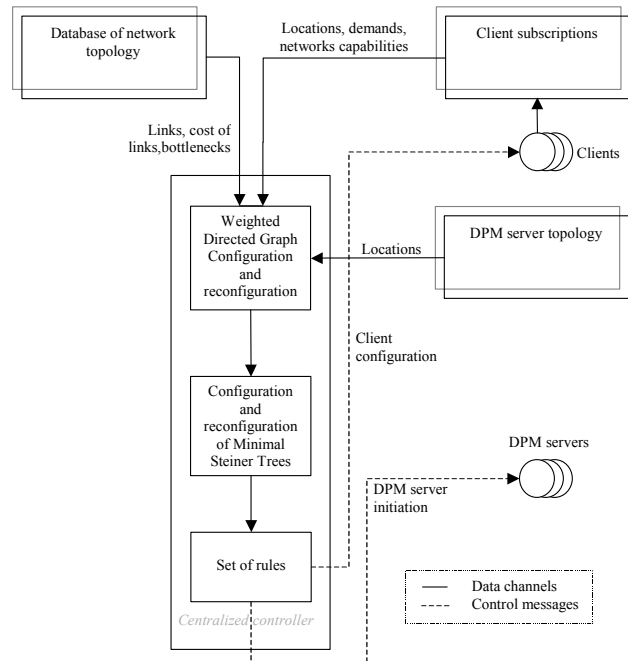

Figure 5. Static centralised DPM service

The network topology database contains complete information about network topology including link reliability, link bandwidth, link cost, possibility of bottlenecks, time delays, topological delay, throughput. The users' database contains complete information about subscribed clients, their locations and other properties. The DPM server database contains the locations of the DPM servers and their connections. A fixed set of dedicated DPM servers is placed in well-suited positions in the network. The controller then creates a weighted graph with all this information and processes it into a spanning tree according to some minimisation cost (e.g. bounding latency, bounding fan in and fan out of the DPM servers) using one of the algorithms referred to in section 5.1. Once all the interconnections between the nodes are determined, the centralised controller produces a set of rules (where each rule applies to one or more DPM server and clients), and disseminates these rules to each node for execution. These rules instruct each DPM server and client what to connect to and when to become active.

7. SELF-ORGANISED DPM OVER UNMANAGED NETWORKS

The example architecture illustrated in the previous section has several drawbacks. In particular it has very limited support for dynamic user membership and changes in the topology. DPM system should dynamically and gracefully respond to users joining

and leaving the application at any time as well as changes in the distribution topology and varying network conditions.

In contrast to the previous section (where distributed partial mixing servers were assumed to be started on pre-determined machines and be possibly idle before the establishment of a DPM session), in this example the distributed partial mixing service is started on demand, potentially at any time and in any part of the network. Moreover, the distributed partial mixing service does not depend on dedicated servers, as any client machine can potentially act as a DPM server. This example is modelled on the self-organised transcoding (SOT) approach proposed by [12]. The basic idea adopted is to use self-organisation to form groups out of co-located receivers, which share loss events, and to provide local adaptation though the use of a DPM service. The idea behind self-organisation usually is to have the end clients self-organise into a multi-level hierarchy of multicast groups, where each individual group corresponds to the homogeneous regions within the heterogeneous multicast tree regions. In this way the resulting topology is congruent with the underlying multicast tree, and one large heterogeneous and difficult problem is reduced to many small, homogeneous and simple sub-problems. The metric used to identify groups in this particular proposal is based on shared loss patterns among end clients. This and other similar metrics are discussed in greater detail in [12], [18].

Self-organised DPM can be described in the following way. It is assumed that all of the clients joining the session belong to a common (prearranged) multicast group (G_0) over which they can exchange various session messages and audio streams. When a client starts detecting packet loss, it requests a DPM server by multicasting a query to the whole multicast group. In this request message, the requesting client (requestor) gives its loss pattern (i.e. packet loss history from each source) and its own identity.

Each client that shares the same loss pattern as the requestor decides to join the requestor and use the same distributed partial mixing server (members of the same loss group are likely to be behind the same bottleneck link(s) and therefore likely to need the same rate adaptation server, i.e. the same DPM server).

All of the other clients that receive this query determine whether they have *better* reception by comparing the received loss pattern with their own loss pattern. If they have better reception, they send a response (via unicast) to the requestor, indicating that they could act as a DPM server for the requesting client. Their response message contains their loss pattern as well as their distance to the requestor (e.g. TTL, time).

Once the requestor has gathered all of the available responses, it determines the best one and selects that client as its distributed partial mixer. It then multicasts the address of the selected distributed partial mixer in order to both to notify the offering client that is going to provide the distributed partial mixing service, and to instruct all of the members in the same loss group to switch to the new multicast group (G_1) to which this new distributed partial mixer will send.

The newly elected DPM server can then perform partial mixing and multicast the resulting audio streams to the new loss group that it is serving.
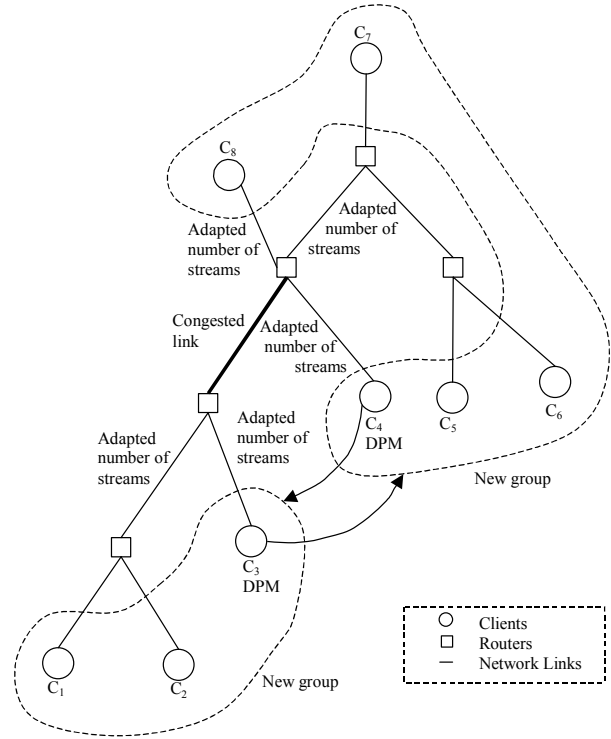

Figure 6. Self-organising DPM service

Figure 6 shows the scenario when a problematic link is congested in both directions and there are two groups of clients affected by two different problems (C_{1-3}) and (C_{4-8}). Clients C_{1-3} receive the audio streams from C_{4-8} via C_{4DPM} on the new multicast group G_1. This new multicast groups is also used for local communication among clients C_{1-3}. We assume that C_3 is elected to perform DPM functionality for the group of clients on the other side of the link C_{4-8}. Two new multicast groups are used for achieving this communication. Clients C_{1-3} use G_1 for receiving audio from C_{4-8} via C_{4DPM}, and for communicating among each other. Similarly, clients C_{4-8} use G_2 for receiving audio from C_{1-3} via G_{3DPM}, and for communicating among each other. The figure also shows systematically how the original multicast group (G_0) gets partitioned into multicast islands (G_1 and G_2) each with their own DPM server (C_{3DPM} and C_{4DPM}). Each member in an island sends audio to its local multicast group, and one member forwards the audio to the other multicast island. Multicast partitioning is important because a new member joining the original multicast group would otherwise pull down all traffic across the congested link.

This self-organisation approach gracefully accommodates new clients joining and old clients leavening. When a new client joins, it can probe for existing loss groups and existing distributed partial mixers in its network neighbourhood before it joins one. If the requesting client leaves, some of the other clients in the groups must schedule a message to take over the role of the requestor. A distributed partial mixer can also detect the departure of the last member and stop sending to the group.

This approach also copes gracefully with new bottlenecks, resolution of the old ones and even link and router downtime. For example, introduction of new bottleneck links will cause some members of the loss group to detect different loss patterns to those of the requestors, and start the process of discovering a new distributed mixer again.

8. CONCLUSIONS

We have prototyped and tested distributed partial mixing over a simple two LAN situation and established that its can be effective in balancing spatialisation and loss rate, while achieving fairness to competing TCP traffic.

We have not beep able to deploy a larger DPM network 'for real'. However, in this paper we show how a large-scale deployment of DPM would be possible in contrasting situations. We argue that these scenarios and designs show that DPM is both feasible and desirable even for large scale and wide area applications such as future Inhabited TV systems. In particular, the self-organising scheme shows how large scale DPM might be realised in a flexible and adaptive way on current best effort networks, such as the Internet (or at least that subset of the Internet which supports native multicast).

8. ACKNOWLEDGEMENTS

9. REFERENCES

[1] Bowers, J., Pycock, J. & O'Brien, J., Talk and Embodiment in Collaborative Virtual Environments, in Proceedings of CHI'96, 1996.

[2] Fox, A., Gribble, S.D., Brewer E.A, Amir, E., Adapting to Network and Client Variablility via On-Demand Dynamic Distillation, In Proceedings of the Seventh International Conference on Architectural Support for Programming Languages and Operating Systems, pages 160--170, Cambridge, MA, October 1996.

[3] Frécon, E., Greenhalgh, C. & Stenius, M., The DiveBone: An Application-Level Communication Infrastructure for Internet-Based CVEs, in Proceedings of VRST'99 (London, 1999), 58-65.

[4] Goodwin, C., Notes on story structure and the organization of participation, in Atkinson, J. & Heritage, J. (eds.), Structures of Social Action: Studies in Conversation Analysis, 1984, 225-46, CUP.

[5] Greenhalgh, C., Benford, S. & Craven, M., Patterns of Network and User Activity in an Inhabited Television Event, in Proceedings of VRST'99 (London, Dec 20-22 1999), ACM, 58-65.

[6] Greenhalgh, C., Benford, S., Taylor, I., Bowers, J., Walker, G. & Wyver, J., Creating a Live Broadcast from a Virtual Environment, in Proceedings of SIGGRAPH'99 (1999), 375-384.

[7] Greenhalgh C., Purbrick J., & Snowdon D., Inside MASSIVE-3: Flexible Support for Data Consistency and World Structuring, in Proceedings of CVE'2000 (San Francisco, 2000), ACM, 119-127.

[8] Hardman, V., Sasse, M., Handley A. & Watson, Reliable Audio for use over the Internet, in Proceedings of INET'95 (Hawaii), 1995.

[9] Hendrix, C., Barfield, W. Presence within virtual environments as a function of visual display parameters, PRESENCE, 5(3), 274-289, 1996, MIT Press

[10] Hindus, D., Ackerman, M., Mainwaring, S. & Starr, S., Thunderwire: A Field Study of an Audio-Only Media Space, in Proceedings of CSCW'96 (USA), ACM Press, 238-247

[11] Kendon, A., A description of some human greetings, in Conducting interaction: Patterns of behavior in focused encounters, 153-207, 190, Cambridge University Press

[12] Kouvelas, I., Hardman, V., Crowcroft, J., Network Adaptive Continuous-Media Applications through Self-Organised Transcoding, In Proceedings of Network and Operating Systems Support for Digital Audio and Video, 1998.

[13] Krumke, S., Noltemeier, H., Marathe, M., Ravi, R., Ravi, S. S., Improving Steiner trees of a network under multiple constraints, Technical Report LA-UR 96-1374, Los Alamos National Laboratory, Los Alamos, New Mexico, USA, 1996

[14] McCanne. S., Jacobson, V. & Vetterli, M., Receiver-driven Layered Multicast, Computer Communication Review, in Proceedings of ACM SIGCOMM, vol. 26, no 4, Oct, 1996

[15] Pasquale, G., Polyzos, E., Kompella, V., Filter Propagation in Dissemination Trees, In Network and Operating Systems Support for Digital Audio and Video, November 1993.

[16] Radenkovic M, Greenhalgh C., Patent published: Audio Data Processing, International Publication Number: WO 02/17579 A1, 28 February 2002

[17] Radenkovic M, et al "Scaleable and Adaptable Audio Service for Supporting Collaborative Work and Entrainment over the Internet", in the Proceedings of International Conference on Advances in Infrastructures for e-Business, e-Education, e-Science and e-Medicine in Internet (SSGRR 2002), L'Aquila, Italy, January, 2002.

[18] Ratnasamy, S., McCanne, S., Inference of Multicast Routing Tree and Bottleneck Bandwidth Using End-to-end Measurements, in Proceedings IEEE Infocom. '99, New York, March 1999.

[19] Salama, H. F., Reeves, D. S., Viniotis, Y., Delay-Constrained Shared Multicast Trees, Technical Report 96/46, November 1996

[20] Waters, R., Anderson, D., Barrus, J., Brogan, D., Casey, M., McKeown, S., Nitta, T., Sterns, I. & Yerazunis, W., Diamond Park and Spline: a social virtual reality system with 3D animation, spoken interaction, and runtime extendibility, PRESENCE, 6 (4), 461-481, 1997, MIT.

[21] Yeadon, N., Quality of Service Filters for Multimedia Communications, Ph.D. Thesis, Lancaster University, Lancaster, U.K., May 1996.

[22] .Zhu, Q., Parsa, M., Dai, W.W.M., An Iterative Approach for Delay-Bounded Minimum Steiner Tree Construction, Technical Report UCSC-CRL-94-39, UC Santa Cruz, 1994.