

Towards (Perceived) Latency Minimisation in Audio-Conferencing Systems over Application-Level Multicast*

Nick Blundell Laurent Mathy
Lancaster University, UK
{n.blundell, laurent}@comp.lancs.ac.uk

Abstract – *In this paper, we propose a design for a scalable and dynamically-adaptive application-level multicast (ALM) routing protocol, designed specifically for audio-conferencing systems over the Internet.*

Currently proposed ALM protocols try to optimise receiver-delay for the whole group of participating nodes as the overlay-network is built and during overlay-network maintenance, which, when using standard packet flooding, can result in a number of the participants experiencing unacceptably-high latencies, unsuitable for real-time audio communication; whereas we propose to dynamically prioritise routing for those participants who are currently in conversation (i.e. those who require the lowest latencies in order to react to conversational cues) and allow higher latencies for participants who simply listen to the conversation without taking an active part in it at that particular moment in time. Thus, we aim to provide low perceived latency for all of the audio-conference participants without any support from the underlying network.

Keywords: Application-level multicast (ALM), audio conferencing, voice over IP (VoIP).

1 Introduction

The term *audio conferencing* refers to multi-party communication among participants who share access to a common audio-communication channel. Having been the focus of much recent research, driven by interests in Internet telephony, the problems of audio transmission over packet networks have largely been solved; however, the enablement of audio conferencing over the Internet for groups of any reasonable size has proved to be more problematic.

As a result of faster computer processors; full-duplex sound devices; efficient, low-bandwidth audio-compression codecs; packet-loss concealment techniques; and more recently with the increase in network bandwidth accessible to the home, point-to-point audio communication of toll-standard or higher quality can be more-or-less achieved over the Internet, provided that network delay and packet loss do not exceed their tolerable thresholds (see section 2.1).

Table 1: A Critique of Proposed Group-Communication Models for Internet Audio-Conferencing Systems

Model	Description	Comments
Network-Level Multicast	A member joins a multicast group and transmits audio frames to the group address. Packet duplication is done efficiently in the network such that the other members receive audio frames in unicast trip time [5].	+ Minimal delay. + High scalability. – Not widely deployed.
Multiple Unicast	A member transmits a duplicate of an audio frame to each of the other members.	+ Minimal delay. – Poor scalability. – Wasteful of network resources.
Central Server	A member transmits an audio frame to an audio-conference server which duplicates the frame to the other members. The server may also mix audio frames generated by simultaneously speaking members to produce a combined stream suitable for low-bandwidth clients [16].	+ Supports low-bandwidth clients. – Wasteful of network resources. – Server position: delay okay for a LAN but not for the Internet.
Application-Level Multicast (ALM)	Members organise themselves into one or more overlay tree(s) by strategically adding data links between each other. Acting as routers at the application level, members flood audio frames over the overlay tree(s) akin to multicast routers in network-level multicast.	+ Reasonable scalability. – Scalability at the cost of increased delay.

Group communication for real-time multi-media applications, on the other hand, has posed more challenging in the way of availability, group scalability, and of communication-channel quality: network-level multicast was proposed over a decade ago [5] as a solution for efficient, large-scale group communication over the Internet but wide-scale deployment of the service has since been hampered due to the various technical and administrative issues that surround it [3]. In response to the lack of a group-communication service in the network, various techniques have been proposed and adopted that implement this service at the application level; the main models of which, along with the network-level multicast model, are described and compared in table 1.

Application-level multicast (ALM) has been proposed as

a solution for audio conferencing [3], however, as an ALM group grows in size there is, inevitably, an increased imbalance in the degrees of latency (end-to-end delay) experienced by different, communicating node pairs within the group such that, and with regard to studies into user tolerance of latency in audio systems [9], a significant number of participant pairs will begin to experience difficulty in communicating with each other due to excessive latency in the audio channel.

The remainder of this paper is organised as follows: firstly, in section 2, we describe related work that has influenced our design rationale; next, in section 3, we present observations of conversation patterns (using samples of actual conversation) in a preliminary study of conversation turn-taking before, in section 4, describing the proposed application-level network audio-conferencing (ALNAC) routing protocol. Finally, in sections 5 and 6, we give concluding remarks on the paper and describe future directions of this work.

2 Related Work

In this section, we describe three areas of particular relevance to the proposed work, namely: Internet packet-audio transmission, application-level multicast (ALM) techniques, and conversation analysis — the study of conversation.

2.1 Internet Packet-Audio Systems

After being sampled from the microphone by the soundcard, during the *packetisation* phase, audio samples are stored into an audio-frame buffer. The size of the audio frame determines the packetisation delay — the delay for a single audio sample, after its capture, before it can be transmitted; a small audio-frame is therefore preferred for minimising overall latency (end-to-end delay), though having a frame size that is too small will result in inefficient packet processing by the CPU and by the network. As a compromise, audio-frame sizes which are multiples of 20 ms are commonly used.

When an audio frame becomes full of captured audio samples it is (if required) compressed and then transmitted on the network. If silence suppression is used — which typically is the case for voice — audio frames containing only silence will be dropped before transmission. Silence suppression results the generation of traffic bursts, known as talk spurts, when the user is speaking and thus conserves bandwidth while they are silent.

On reaching the destination network-node, an audio frame is uncompressed and then placed into a *playout buffer*, which serves to absorb variation in inter-packet arrival times before the audio frame is passed to the soundcard for playback; if no artificial delay were introduced by a playout buffer, the next packet, on experiencing a slightly higher network delay, would arrive after all of the samples in the previous audio frame had been played, resulting in an audible gap and thus reducing the listener’s perceived quality of the communication channel.

Network packet loss results in the same kind of disruption as does excessive jitter, though numerous techniques, such as forward-error correction (FEC) and lost-packet substitution, have been proposed that are able to mask this from the listener up to a degree of packet loss commonly experienced on the Internet [15].

The latency components of audio transmission, therefore, include: packetisation delay, pre-processing delay (silence-suppression and compression), network-transmission delay, network-propagation delay, uncompression delay, and finally, playout-buffering delay; where network-propagation delay is typically the least-predictable and most-dominant component for transmission of audio over the Internet.

An abundance of studies into user tolerance of round-trip latency in audio-communication systems has been conducted; these studies generally agree on the following levels of tolerance: excellent, 0–300 ms; good, 300–600 ms; poor, 600–700 ms; quality becomes unacceptable for a round-trip latency in excess of 700 ms [9].

As latency increases, it is miss-interpreted by the user as extended pause in speech, causing confusion when they do not get an immediate response, and inevitably results in their loss of synchronisation with the conversation.

2.2 Application-Level Multicast

Application-level multicast (ALM) is a technique for establishing a group-communication service that is reliant only on the formation of participating nodes into an overlay network, upon which they perform multicast-like routing of packets at the application level.

Generally speaking, the common goal of existing ALM techniques is to construct one or more overlay trees such that they (1) evenly distribute responsibility for data-packet duplication among the group members and (2) optimise paths on the tree for some cost metric (e.g. delay, or bandwidth).

Table 2 gives a brief description, and examples, of each of the four ALM protocol classes.

Summarising characteristics of ALM protocols: mesh-first protocols tend to produce more-optimal trees in comparison to the other techniques due to global dissemination of inter-node measurements among group nodes, which, as a direct result, limits the scalability of such mesh-first approaches; generally, the other protocols have the advantage of scalability at the cost of producing less-optimal trees.

2.3 Conversation Analysis

Conversation analysis is the study of verbal communication between people, with an emphasis on how that communication is structured and how it is affected by scenarios (e.g. formal or social settings, the cultural backgrounds of participants, and so on) [14].

An area of conversation analysis of particular relevance here is the study of *turn taking*: the basic form of organisation in conversation. In conversation, people naturally organise their spoken contributions (utterances) into turns, where

Table 2: Classification of Application-Level Multicast Protocols

Class	Description
Centralised	A central server is responsible for calculating an optimised overlay tree among registered nodes based on reports of inter-node cost measurements that are received periodically from the nodes. To realise the calculated tree, the server strategically instructs specific nodes to add or drop data-links to other nodes in the group. Examples of this protocol class include: ALMI [10] and HBM [12].
Mesh-First	Nodes arrange themselves into a well-connected mesh through periodic exchanges of cost-metric probes, on top of which they run a protocol similar to DVMRP — as used by network-multicast routers. Narada is an example of this protocol class [3].
Tree-First	On joining the overlay, nodes connect immediately into a tree structure, trying to optimise their position with regard to local cost information that is discovered during the join process. Examples of this protocol class include: NICE [2], TBCP [8], and HTMP [17].
Coordinate Systems	Protocols in this class use the notion of coordinates to find optimal routing paths among nodes: either as geometric coordinates in a virtual plane or as the result of applying a distributed hash table (DHT) to key-value pairs. Examples of this protocol class include: CAN [11], SCRIBE [13], and delaunay triangulation [7].

each person silently waits, listening to the current speaker, for their turn to speak [14].

Overlapping speech occurs rarely in conversation, since one person *must* remain silent to effectively listen to what another person is saying. In situations where overlapping speech *does* occur, however, the conversation is quickly repaired by all-but-one of the competing speakers backing down to let a single person speak; typically, the competitor who speaks the loudest, or who spoke first, will be allowed to continue and take up the turn [14].

A large part of verbal communication among any number of participants consists of turns that are somehow related to each other, known as *adjacency pairs*. Examples of adjacency pairs include: where one participant provokes an answer from another participant by posing a question, where people exchange greetings, and where people seek clarification of previously-received information that they did not fully understand [14].

In conversation, adjacency pairs may be nested such that the response to one question may be another question which must be answered before the original question can be properly answered [14]. The organisation of conversation turns into adjacency pairs naturally leads to a degree of localisation, where, over a given interval of time, a small number of those participants who are present exchange turns with one another.

Figure 1 gives a dialog example of everyday conversation, illustrating the nested relationship between turns, and, more generally, illustrating how people become engaged in specific discussions only of interest to each other.

Neil: Would you like to go out tonight, Jane? (question)
Jane: Where to? (response and question)
Neil: The cinema. (response)
Jane: What film is on tonight? (response and question)
Neil: “Big”, with Tom Hanks. (response)
Jane: Sounds good, would you like that, Issac? (re-routed to another person)
Issac: Yes! (response)
Jane: Yes, I would like that, Neil, if Issac is coming too. (response)
Neil: Right then, lets get ready! (non-adjacency-pair)

Figure 1: An example of nested adjacency pairs in conversation, leading to temporal localisation of turns between three people who are potentially in the presence more people who are listening.

3 Preliminary Study of Next-Speaker Predictability

This section presents an initial study of the patterns observed in actual conversation, helping to support our design rationale in the following section.

To examine the extent of turn localisation (see section 2.3) that occurs in conversation, as indicated by observations in conversation analysis, we performed a simple analysis of two audio-conference trace files and of two public-meeting transcripts.

The trace files, which contained time-stamps of talk spurts produced by participants, were logged from a basic, locally developed audio-conferencing client that used multiple-unicast transmission over a LAN. One session included ten players of an online game, and the other session was an informal discussion between twelve people, where both sessions lasted between twenty and twenty-five minutes.

To represent larger group sizes in the study, for which sizes we were unable to trace on the audio-conference system, we analysed the turn-taking patterns of participants in two long meeting transcripts, involving 38 [4] and 42 [1] speaking participants, both of which were publicly available on the Internet.

We calculated the probability that one of a backlog of n previous, distinct speakers would speak next after the current speaker: That is, the probability that the last person to speak before the current speaker would speak next; and then the probability that one of the last two people to speak, before the current speaker, would speak next, and so on.

In the analysis, we considered it impossible to predict the next participant to speak, based on their previous activity, if they had not yet spoken and so omitted such predictions from the results.

Table 3: Accuracy of next-speaker prediction when considering a backlog of n distinct speakers.

Back-Log Size	1	2	3	4	5	6	7
Accurate Predictions (%)	58	73	82	88	92	94	95

Table 3 displays the mean probabilities of the four analysed sessions. The figures show that, by considering only the last speaker, the next speaker can be predicted with a chance of 58%, suggesting — the obvious — that many of the consecutive turns in the sessions were exchanged between the same pairs of people.

The probability of predicting the next speaker increases rapidly against a small increase in the previous-speaker backlog size, which suggests the localisation of participants' turns: where the sessions consisted mainly of temporal discussions between small numbers of participants who shared a common interest over a given interval of time.

4 Dynamic Overlay-Routing Protocol Design

The design for the proposed application-level network audio-conferencing (ALNAC) routing protocol is derived from both our understanding of a user's perception of audio-channel quality (see section 2.1) and from our observation of patterns in conversation (see section 2.3).

From these, we surmise that, if a pair of overlay nodes are joined with minimal distance only for the duration that their respective participants are talking to each other, and are otherwise allowed to become further apart (with respect to overlay distance) when simply listening and not talking, then they will perceive a constant minimal delay whilst reaping the good scalability of ALM provided that, (1) the unicast distance is not excessively high and (2) path adaptation is responsive to their changing levels of interactivity.

Thus, the general strategy of the ALNAC routing protocol is to enable the transfer of audio packets from a speaking participant to the other group participant(s) who are most likely to respond, by also speaking, in the (assumed) minimum time that is achievable (unicast trip time) whilst ensuring group scalability, potentially to a large number of participants.

Audio conferencing requires any-source group communication, whereby all members may transmit audio data to each of the other group members; ALM protocols achieve any-source routing in one of two ways: either, by using a shared tree, whereby data from all of the members is forwarded along paths of a single tree; or by using per-source trees, where separate delivery trees are generated for each sender.

For clarity, we describe the ALNAC routing protocol in terms of a single, shared tree, on top of which we perform uni-directional routing (uni-directional routing provides lower maximum hop counts on any-source delivery paths than does bi-directional routing), whereby a transmitting node sends data packets down the tree via its children and supplies the rest of the tree via the tree-root node; it then follows that the proposed protocol can be applied to per-source trees by considering a sending node who is the root.

To achieve these stated goals, the ALNAC routing protocol will extend upon the technique of nodes simply flooding

data-packets over an overlay tree by allowing nodes to, at their discretion, forward packets to non-child and non-root nodes whilst ensuring that no loops will be formed in delivery paths of the packets and thus avoiding duplication.

An ALNAC routing-protocol node will obtain a set of child nodes and a parent node from some overlay-tree construction protocol, and, where available, to optimise routing, will make use of additional inter-node distance information collected by the overlay-construction protocol.

Therefore, when an audio-conference participant begins to speak, the ALNAC routing protocol will make it a priority to get packets containing the speaker's audio frames, to the other group participants who are most likely to engage, or be currently engaged in conversation with the speaking participant.

The following subsection, section 4.1, describes the overall priority-routing process of the ALNAC routing protocol before remaining subsections, sections 4.2, 4.3, 4.4, and 4.5, describe specific processes of the protocol.

4.1 Priority Routing of Data Packets

On transmitting an audio packet, an ALNAC node will not simply flood the packet to its children and to root node — as performed in typical overlay-tree flooding — but will choose a set of *target nodes*, nodes to whom it will directly send the packet, to include a number of nodes hosting participants who have recently spoken, such that they will receive the audio packet in unicast trip time. Section 4.2 describes the target-node selection process in detail.

Note that, to prevent the introduction of intra-talk-spurt jitter (see section 2.1) at the receiving nodes, the target-node set will change only between, and not during, a participant's transmitted talk spurts: For example, if a participant is speaking and, at that same time, begins to receive packets from a new speaker, then the original speaker will defer inclusion of the new speaker in the target-node set until the original speaker's next talk spurt begins.

By directly routing packets to nodes that host recently-speaking participants, a transmitting node, of constrained out-degree (the maximum number of nodes to whom it may simultaneously send an audio frame), may deprive some, or all, of its dependant tree-nodes (children and the root node) from receiving the packet directly from itself; in which case, we propose that a node can temporarily delegate its responsibility for such nodes to those nodes to which it is directly sending the packet, thus ensuring that the deprived nodes will be eventually supplied with the packet, albeit after a longer duration of time. The selection of suitable delegates is described in more detail in section 4.3.

To ensure responsiveness of the ALNAC routing protocol to the changing set of recently-speaking participants, whilst ensuring consistency in routing state among the tree (that is constructed upon a network susceptible to packet loss), we chose to put such instructions of delegation into the packet headers of each audio packet sent to delegates such that deviations from tree routes are done on a per-packet basis.

The packet header will simply carry an additional list of delegated-node addresses which have been assigned to the delegate by the packet sender.

To avoid duplication as a result of a transmitting node supplying a packet directly to a non-child and non-tree-dependant node, where the directly-supplied node will, eventually, receive a second copy of the packet from its parent, the address of the supplied node will be included, and remain, in the header of all packets containing that audio frame forwarded over the overlay tree; that is, except in the packet sent to the directly-supplied node, who, by default, will know that it has been supplied. Thus, a parent of a supplied node will choose simply not to forward an audio packet to a child that is indicated to have been already supplied in the supplied-node list of the audio-packet header.

Figure 2 illustrates this adaptive process of the routing protocol, where a node, S , with an out-degree limit of five, transmits an audio packet directly to three recently-speaking participant nodes, A_1 , A_2 , and A_3 , and to the tree-root node, R . S also sends the audio packet to one of its children, D , nominating D as the most suitable delegate for S 's three other children, who have been deprived of directly receiving the packet from S . Consequently, D deprives two of its own children, allowing it to supply directly two of S 's children, and the third child, indirectly, through a chained delegation (see section 4.5).

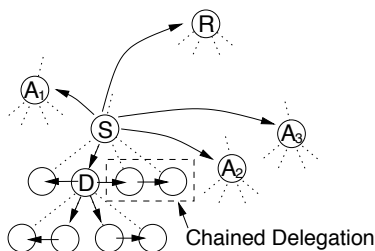


Figure 2: An example of an audio frame being directly routed to three recently-speaking participant nodes, displaying also the adapted delegation paths of the packet which will ensure the audio frame still reaches all members of the tree, albeit on deviated paths.

4.2 Target Node Selection

The selection of target nodes (the nodes that will receive an audio frame directly from the transmitting node) is ultimately constrained by the maximum out-degree of the transmitting node and, therefore, there is a trade-off between the number of recently-speaking participants who can be directly sent the audio frame and the degree of disruption to tree routes caused by child and root-node deprivation and the resulting delegations.

We propose a simple control for this optimisation trade-off by enforcing that a minimum number, n , of the transmitting-node's dependent nodes are always included in the target-node set; where a value for n can be chosen between zero and

the transmitting-node's maximum out-degree — at which value, routing would become standard tree-flooding.

Observations of localisation, both in conversation analysis (see section 2.3) and in our short analysis of transcript and trace files (see section 3), indicate that, if we prioritise delivery of audio packets to three, or more, nodes hosting those participants who have spoken previously, then there will be a good chance of achieving minimal overlay-network delay between participants who exchange turns (talk) with each other.

Where target-node selection allows the inclusion of either the tree-root node or a child node, the tree-root node will be included to ensure that the packet spreads first-most towards the whole tree, rather than to a subtree.

Where target-node selection allows the inclusion of a subset of children, and where local, inter-node distance information is available from the overlay-construction protocol, the central-most children will be selected, from among the other children, as approximations to the most suitable delegates.

In the case where no distance information is available among a subset of the target nodes but where consistency of target-node selection is nevertheless required, an arbitrary selection will be made in favor of children with the lowest network addresses.

4.3 Delegate Nomination

The transmitting node will nominate some, or all, of its target nodes as delegates depending on the existence of inter-node distance information between them and the deprived nodes: If this distance information is available for a subset of the target nodes, they will be nominated as delegates; otherwise, where no distance information is available, *all* of the target nodes will be nominated as delegates, in favor of a balanced number of delegations per delegate.

4.4 Assignment of Delegations

Once a transmitting node has determined its sets of target nodes, delegate nodes, and deprived nodes, it will attempt to assign each deprived node to its nearest nominated delegate, whilst ensuring a balanced distribution of delegations per delegate. If no distance information is available, the assignment will be made arbitrarily, based on the distance between network addresses values.

4.5 Delegation and Delegation Chaining

As a transmitting node will prioritise direct delivery of audio packets to recently-speaking participant nodes, a delegate node will do so for the deprived nodes for which it has been delegated responsibility. In an effort to supply the delegated deprived-nodes in the least time possible, a delegate node may deprive some of its own children from directly receiving the audio frame. Note that, unlike a transmitting node, a delegate will deprive no more than half of its number of children from directly receiving the packet; this will cause path disruption to be absorbed higher in the tree and will therefore result in more-consistent delay penalties for the group

(i.e. more nodes will be consistently affected by the delay penalty as appose to a few nodes being greatly affected).

As a further optimisation to the delegation process, and where inter-node distance information is available to the transmitting node between a delegate node and each of its assigned delegated nodes, a transmitting node will give each delegate sufficient information in its delegate list to allow efficient chaining of delegations such that, if a delegate has more delegations than it is able to directly supply (after also depriving some of its children), it can efficiently chain delegations by delegating delegated nodes to other delegated nodes.

The transmitting node will calculate delegation chains for a delegate node as follows:

- start with n optimal chains (paths) from the delegate to each of its assigned delegated nodes (see figure 3(a))
- append the two chains that produce the shortest path, producing $n - 1$ chains (see figure 3(b))
- store the address of the first node in the chain that was joined to the appended chain, such that the chain can later separated again at that point.
- continue to join the chains and to store the address of the joining node until a single chain remains that spans all of the delegates (see figure 3(c)–(d))

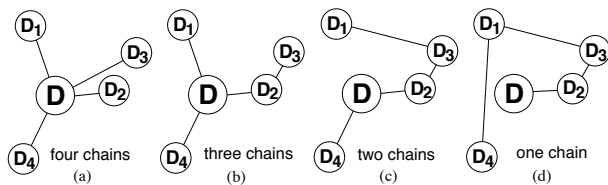


Figure 3: Construction of a single chain through all of the delegated nodes that are being assigned to the delegate node, D , such that it can be broken into efficient paths through the delegated nodes.

The result of this process is that, we have a single chain from the delegate through each delegated node, and also the order in, and points at, which chains were joined together. Thus, when a delegate node receives this information it is able to break the chain into as many optimal chain fragments as it is able to directly serve.

5 Conclusions

In this paper, we have justified the case, and presented a design, for an application-level multicast routing protocol that can dynamically adapt audio-packet routing to reflect the changing patterns in application usage such that we can improve upon the perceived quality of Internet audio-conferencing systems whilst maintaining good scalability.

The approach we have taken essentially fuses the benefits of multiple-unicast and ALM group-communication techniques, through consideration of patterns observed in conversation and through an understanding of a user’s perception of audio-channel quality; in effect, we make the audio-conference participants less sensitive to the group’s size than they otherwise would be for standard ALM flooding.

6 Further Work

We are in the process of simulating the ALNAC routing protocol, using trace-files of actual audio-conferences, to examine the effects that dynamic routing and delegation have on network stress and on the overall group delay.

Next, we plan to implement an ALM plug-in for the Robust-Audio Tool (RAT) client [6] — designed originally for audio conferencing over network multicast — with which we will conduct subjective user trials of the ALNAC routing protocol.

Such trials will help us to better understand the limitations of the routing technique, such as: when group size increases, how does the exchange of utterances between a pair of participants sound to (1) a recently-speaking participant and (2) to a listening-only participant — at what point, if so, do exchanged utterances begin to loose synchronisation for the listening-only participant; also, to what size can a group scale before competition for turns among participants becomes too unfair for those who have not spoken recently.

References

- [1] Athelstan. Transcript of a Forty-Two-Member Meeting in the Corpus of Spoken Professional American English (CSPA). <http://www.athel.com/sample.html>.
- [2] S. Banerjee, B. Bhattacharjee, and C. Kommareddy. Scalable Application Layer Multicast. In *ACM SIGCOMM*, Aug 2002.
- [3] Y-H. Chu, S. Rao, and H. Zhang. A Case for End System Multicast. In *ACM SIGMETRICS*, pages 1–12, Santa Clare, CA, USA, June 2000.
- [4] Competition Commission. Lloyds TSB / Abbey National Merger Inquiry Open-Meeting Transcript. <http://www.competition-commission.org.uk/inquiries/completed/2001/lloyd%20tsb/lloydstran.htm>.
- [5] S. Deering and D. Cheriton. Multicast Routing in Datagrams Internetworks and Extended LANs. *ACM Trans. Comp. Syst.*, 8:85–110, May 1990.
- [6] Vicky Hardman, Martina Angela Sasse, and Isidor Kouvelas. Successful Multiparty Audio Communication over the Internet. *Communications of the ACM*, 41(5):74–80, may 1998.

- [7] J. Lieberherr, M. Nahas, and W. Si. Application-Layer Multicast with Delaunay Triangulations. In *IEEE GLOBECOM*, Nov 2001.
- [8] L. Mathy, R. Canonico, and D. Hutchison. An Overlay Tree Building Control Protocol. In *Proc. of Intl. workshop on Networked Group Communication (NGC)*, pages 76–87, Nov 2001.
- [9] Princy C. Mehta and Sanjay Udani. Overview of VoIP, Technical Report MS-CIS-01-31. Technical report, University of Pennsylvania, February 2001.
- [10] D. Pendarakis, S. Shi, D. Verma, and M. Waldvogel. ALMI: an Application Level Multicast Infrastructure. In *3rd USENIX Symposium on Internet Technologies*, San Francisco, CA, USA, Mar 2001.
- [11] S. Ratnasamy, M. Handley, and S. Shenker. Application-Level Multicast using Content Addressable Networks. In *Proc. of Intl. workshop on Networked Group Communication (NGC)*, Nov 2001.
- [12] V. Roca and A. El-Sayed. A Host-Based Multicast Solution for Group Communications. In *IEEE Intl. Conf. Networking*, Jul 2001.
- [13] A. Rowstron, A-M. Kermarrec, M. Castro, and P. Druschel. SCRIBE: the Design of a Large-Scale Event Notification Infrastructure. In *Proc. of Intl. workshop on Networked Group Communication (NGC)*, Nov 2001.
- [14] H. Sacks. *Lectures on Conversation*. Blackwell, Oxford, UK, 1992.
- [15] H. Sanneck, A. Stenger, K. Younes, and B. Girod. A New Technique for Audio Packet Loss Concealment. In *In Proceedings of IEEE Global Internet (London, England)*, pages 48–52, Nov 1996.
- [16] Kundan Singh, Gautam Nair, and Henning Schulzrinne. Centralized Conferencing using SIP. In *Proceedings of the 2nd IP-Telephony Workshop*, April 2001.
- [17] B. Zhang, S. Jamin, and L. Zhang. Host Multicast: a Framework for Delivering Multicast to End Users. In *IEEE Infocom*, Jun 2002.