# MutualCast: A Serverless Peer-to-Peer Multiparty Real-Time Audio Conferencing System

Jin Li
Communication and Collaboration Systems, Microsoft Research
One Microsoft Way, Bld. 113, Redmond, WA 98052
Email: jinl@microsoft.com

## Abstract

We describe MutualCast, a serverless peer-to-peer (P2P) multiparty real-time audio conferencing system. In MutualCast, the peers form a fully connected clique. During the conferencing session, each peer takes turn to mix and redeliver the compressed audio. The audios are split into frames, and the number of frames mixed and redelivered by a certain peer is proportional to the available resource of the peer, e.g., the upload bandwidth or the computation power. MutualCast balances the serving load (network bandwidth and computation power) needed for the mixing among all participant peers. It enables a multiparty conferencing session without any powerful server.

**Keywords:** Real-time audio conferencing, peer-to-peer (P2P), fully connected network, MutualCast content distribution.

## 1. Introduction

A multiparty audio conferencing system enables a group of people to engage in a real-time audio communication session, with possibly multiple people speaking at the same time. Although video and data sharing are nice additional features, audio with sufficient quality remains to be the necessary condition for almost all collaboration scenarios that involve multiple groups of people. In addition to the components of an ordinary two-party audio conferencing system, e.g., audio capture, acoustic echo cancellation (AEC), automatic gain control (AGC), audio/speech compression, the multiparty audio conferencing system poses unique challenges in audio mixing and network delivery.

Let us assume that $n$ peers are engaged in a multiparty conference session, with possible multiple concurrent speakers. Let us further assume that each stream of audio requires a bandwidth of $bw$. The conferencing system may be formed with a variety of topologies and mixing strategies, as shown in Figure 1. One popular topology is the star topology, as in Figure 1a. A sever $s$ receives the audio streams from all peers, mixes them, and sends the mixed and re-encoded audio back to all peers. The advantage of the star topology is that each peer is the same as that of a two-party conferencing system, and thus needs no modification. Only the server needs to be redesigned to support multiparty conferencing. The star topology is a popular choice for commercial multiparty conferencing solutions, such as [1]. The shortcoming is that the burden on the server can be heavy, as it needs to receive $n$ streams of compressed audio (with $n \cdot bw$ download bandwidth), decode, mix and re-encode them, and send the mixed audio back to $n$ peers ($n \cdot bw$ upload bandwidth).

A second common topology is a fully connected unicast network, e.g., [2], as shown in Figure 1b. The peers do not perform any audio mixing. Each speaker simply sends the compressed audio to every other peer. In such a topology, each peer needs $(n-1) \cdot bw$ upload bandwidth to send the audio to the rest of the peer, and a maximum

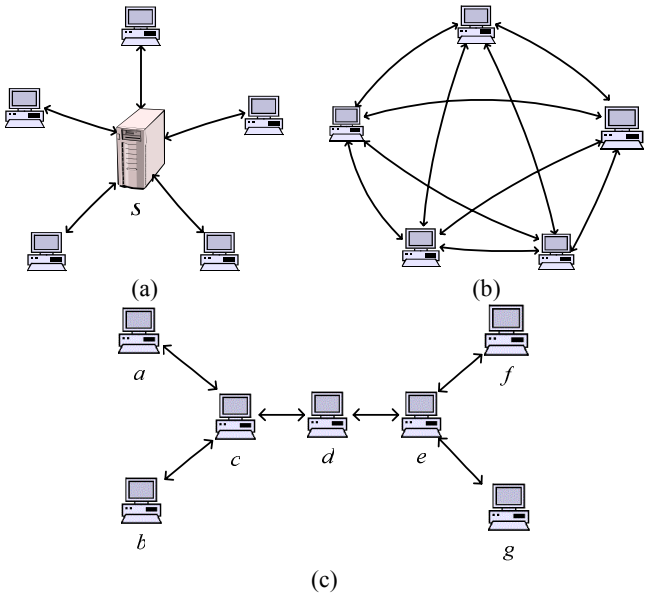


Figure 1. Multiparty conferencing network topology: a) the star topology, b) the fully connected graph via unicast, c) the generic graph, with peers $c$, $d$, and $e$ being the gateway nodes.

of $(n-1) \cdot bw$ download bandwidth to receive the incoming audio. This places a big burden on each peer and the entire network.

A third possible topology is to form a generic graph, and use end system mixing, as shown in Figure 1c. A sample system can be found in [3]. In this example, peers $a$, $b$, $f$ and $g$ are leaf nodes, and do not perform any mixing operations. The peers $c$, $d$ and $e$ serve as a gateway node, which mixes and redelivers the audio for the nearby peers. In general, a gateway node with $m$ neighbors requires $m \cdot bw$ upload and download bandwidth to receive and redeliver the audio. Since $m$ is usually much smaller than $n$, the design scales well to a large conferencing session. Nevertheless, the burden on the gateway node can be heavy. As the chain of gateways becomes long, the latency in audio delivery increases. The audio may also lose synchronization along the chain of delivery.

A network level solution to further reduce the traffic in an audio conference session is through IP multicast. For example, in the star topology shown in Figure 1a, the peer may still send the compressed audio to the server via unicast, but the server can multicast the mixed and re-encoded audio back to $n$ peers. A sample implementation of such system can be found in [4]. The upload bandwidth of the server is reduced to $bw$. Nevertheless, the requirement on the download bandwidth of the server remains unchanged at $n \cdot bw$. In the fully connected network of Figure 1b, each speaker may also multicast the compressed audio to every other peer in the network. Again, the upload bandwidth of the peer is reduced to $bw$, but the download bandwidth of the peer remain unchanged at $(n-1) \cdot bw$. Moreover, the de-

0-7803-9332-5/05/$20.00 ©2005 IEEE

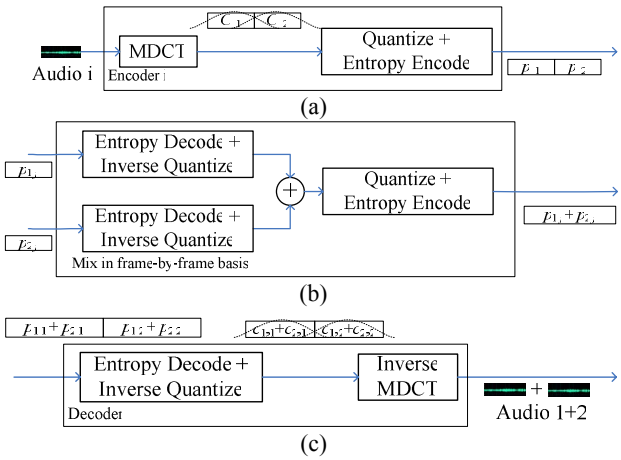


Figure 2. Audio components in a multiparty conferencing systems: a) the encoder, b) the mixer, and c) the decoder.

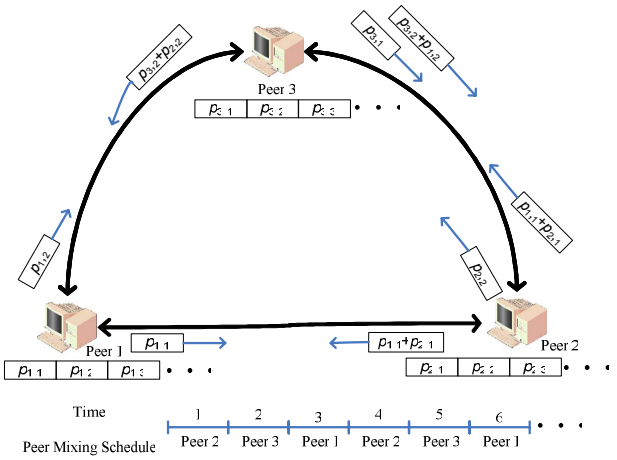


Figure 3. Audio mixing in MutualCast multiparty conferencing system.

ployment of IP multicast is slow in the real world because of issues such as inter-domain routing protocols, ISP business models (charging models), congestion control along the distribution tree, security and so forth. As a result, except certain limited university/corporate subnet and network test bed, e.g., Internet2, native IP multicast support is not widely available.

In all existing multiparty audio conferencing systems, the mixing/redelivery role played by the peer/server is fixed by the network topology. In this work, we propose an alternative form of multiparty audio conferencing system called MutualCast. It can be considered as an extension of the MutualCast file distribution system proposed in [5]. A key characteristic of MutualCast is that the mixing and redelivering task are rotated among the peers. Using the special property of the waveform coded audio that the audio mixing can be performed on the transform domain and on a frame-by-frame basis, MutualCast rotates the mixing and redelivering tasks among the participant peers, thereby sharing the network bandwidth and computation load. MutualCast may conduct a multiparty audio conferencing session without a powerful server.

The rest of the paper is organized as follows. The operation of a MutualCast clique is examined in Section 2. The MutualCast may also serve as a super gateway or a super server, and its operation is described in Section 3. We analyze the bandwidth and computation load requirement for the MutualCast peer node in Section 4. A conclusion is given in Section 5.

## 2. Mutualcast Multiparty Audio Conferencing System

The basic operations of MutualCast are as follows. The compressed audio is split into frames. At each frame, one peer node is selected to mix and redelivery the audio for the rest peers. The number of frames mixed and redelivered by a certain peer is proportional to its available resource, e.g., the upload bandwidth. In the follows, we examine a number of MutualCast components. They are 1) the audio mixing unit, 2) the MutualCast mixing strategy, 3) the allocation of the mixing tasks, and 4) the delay.

### 2.1 Mixing of Waveform Coded Audio

We encode audio with a waveform codec, such as Siren/G.722.1 [6]. The operation of the encoder is shown in Figure 2a. The audio waveform is first split into frames (say 20ms), transformed by a MDCT[1] (modified discrete cosine transform) module into coefficient blocks $c_{i,j}$, quantized and finally entropy encoded into packet $p_{i,j}$. We use subscript $i$ to index the peer, and use subscript $j$ to index the frame.

Since MDCT is a linear operation, we observe that the waveform coded audio can not only be mixed in the transform domain, but also be mixed on a frame-by-frame basis. The operation of mixing a certain frame of two compressed audio is shown in Figure 2b. First, the compressed audio packets $p_{1,j}$ and $p_{2,j}$ are entropy decoded and inverse quantized. The resultant MDCT transform coefficients are then added together, quantized and entropy re-encoded to generate the frame of the mixed audio $p_{1,j}+p_{2,j}$. No audio packets of the other frames are accessed during the mixing process.

At the receiver, the mixed packets can be decoded normally, as shown in Figure 2c. The mixed audio $p_{1,j}+p_{2,j}$ are feed into the decoder, with each frame being entropy decoded, inverse quantized and inverse MDCT transformed. The result is the mixed audio waveform from peer 1 and 2.

### 2.2 MutualCast: Rotating Audio Mixing

A MutualCast clique consists of a small number of nodes that form a fully connected mesh. Using the property that the waveform coded audio can be mixed on a frame-by-frame basis, MutualCast rotates the mixing and redelivery operation among the peers, thus distributes the bandwidth and computation load.

A sample MutualCast audio mixing session for three peer nodes 1, 2 and 3 is shown in Figure 3. As shown by the audio mixing schedule at the bottom of the figure, the peer nodes 1, 2 and 3 are in charge of audio mixing and redelivery at frames $3k$, $3k+1$ and $3k+2$, respectively. At the 1st frame, the peer 2 mixes and redelivers the audio packets. The peers 1 and 3 send their coded audio $p_{1,1}$ and $p_{3,1}$ to the peer 2. The incoming audio packets are then entropy decoded and inverse quantized back to the MDCT coefficients $c_{1,1}$ and $c_{3,1}$. The peer 2 adds its own coefficients $c_{2,1}$, and send back the mixed audio. In order to avoid echo, the source audio is not mixed and sent back. Thus the peer 2 adds together coefficients $c_{1,1}$ and $c_{2,1}$, quantizes and entropy encodes the sum of the coefficients, and sends the mixed packet $p_{1,1}+p_{2,1}$ to the peer 3. Similarly, the mixed packet $p_{3,1}+p_{2,1}$ is

[1] Because there is a 50% overlap in-between frames, the total algorithmic delay is 40 ms, which doubles that of the frame. Nevertheless, each quantized and entropy coded frame coefficients is still 20ms.

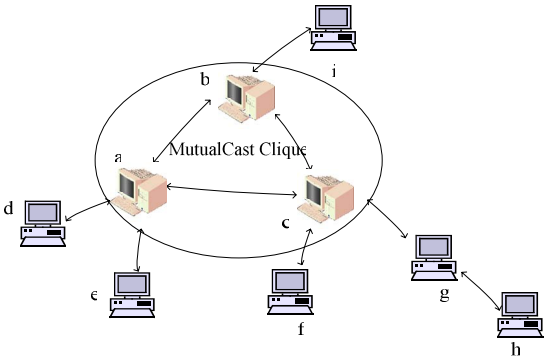


Figure 4. The MutualCast clique (nodes server as a gateway node in a generic multiparty conference graph.

sent to the peer 1. At the destination, the mixed audio packets from different peers are sorted, entropy decoded, inverse quantized and inverse MDCT transformed for play back. At the 2nd frame, the peer 3 takes the mixing role. The peers 1 and 2 send their compressed audio packets at the 2nd frame $p_{1,2}$ and $p_{2,2}$ to the peer 3. The peer 3 mixes the incoming audio packets with its own coefficients $c_{3,2}$, and sends the mixed packet $p_{3,2}+p_{2,2}$ to the peer 1, and the mixed packet $p_{3,2}+p_{1,2}$ to the peer 2. At the 3rd frame, the peer node 1 becomes the mixing node. By time-sharing the mixing and redelivery task, the bandwidth and computational cost of the mixing is distributed among the peers. As a result, a group of less powerful peers can conduct a multiparty audio conferencing session without a server.

In a MutualCast clique of $n$ nodes, each peer sends and receives $2(n\text{-}1)$ packets every $n$ frames. Among them, $(n\text{-}1)$ packets are sent and received during the $n$-1 frames that it does not perform the mixing operation. Also, $(n\text{-}1)$ packets are sent and received during the frame that it performs the mixing and redelivery operation. The upload/download bandwidth required is thus $(2\text{-}2/n)\cdot bw$. It may also be calculated that on average, (2-2/n) streams of audio are decoded and re-encoded by each peer. During the mixing, the peer performs $(n\text{-}1)$ entropy decoding and inverse quantization operations, and $(n\text{-}1)$ forward quantization and entropy encoding operations. An alternative strategy of mixing is for the peer to mix the audio of all peers, i.e., to get $m_j=p_{1,j}+p_{2,j}+p_{3,j}$ for frame $j$, and then send back the same mixed audio to all peers. To get rid of echo, each peer then subtracts its own audio from the mixed audio. For example, peer $i$ subtracts $p_{i,j}$ from $m_j$, which is a mixing operation with subtract instead of addition. The pro of this approach is that the peer only needs to perform a single forward quantization and entropy encoding operation during the mixing. Moreover, if IP multicast is supported among all the peers, the mixing peer may also multicast the mixed packet to the rest of the peers. The con is that since the mixed audio needs to be quantized and entropy coded, the component audio $p_{i,j}$ in the mixed packet $m_j$ is different from the audio $p_{i,j}$ hold by the peer $i$, thus residual echo may persist. The residual echo is more obvious with the increase of the number of the peers and/or the decrease in the coding bitrate of the mixed audio. Due to residual echo, we do not consider mixing of all audio packets as a preferred implementation strategy for MutualCast.

## 2.3 The Allocation Of the Mixing Tasks

MutualCast can allocate the mixing and redelivery task on a frame-by-frame basis. We may assign more mixing tasks to the peers with more resources, and fewer mixing tasks to the peers with few resources.

The paramount resource considered is the upload bandwidths of the peers. In increasingly common networks, the total upload bandwidths of the P2P network are much smaller than the total download bandwidths. This is especially true for end-user nodes on the cable modem and ADSL networks, for which the balance is asymmetrically skewed towards larger download bandwidth. Even for the user nodes on the university/corporate networks, the download bandwidth can still be larger than the available upload bandwidth as the user caps the upload bandwidth. Therefore, it is advantageous to allocate more mixing and redelivery tasks to the peer with higher available upload bandwidth, and fewer tasks to the peer with lower upload bandwidth.

The second resource considered is the peak upload bandwidth (or the physical link bandwidth) of the peer. During the mixing, the MutualCast peer receives and sends out $(n\text{-}1)$ audio packet to $(n\text{-}1)$ peers, the traffic characteristics of the MutualCast peer is bursty. It is helpful to assign more mixing and redelivery tasks to the peer with a faster physical link, or the peer that is connected to the router with a relatively large token bucket, so that the delay caused by sending packets to multiple peers can be reduced.

Normally, the download bandwidths and the computation resources of the peers are not a bottleneck. Nevertheless, we may also take these into consideration in the allocation as well[2].

## 2.4 Delay

Let the network transmission delay between the peer node $i$ and $j$ be $d_{i,j}$. Assuming the delay caused by the mixing operation is negligible, the delay of the peer $i$ to receive an audio frame mixed by peer $k$ amounts to:

$$D_{i,k} = \left(\max_{j\neq i,k} d_{j,k}\right) + d_{k,i}. \quad (1)$$

The maximum delay of the peer $i$ can be calculated as:

$$D_i = \max_{k\neq i}\left(d_{k,i} + \left(\max_{j\neq i,k} d_{j,k}\right)\right), \quad (2)$$

while the maximum delay of the MutualCast is:

$$D_{\max} = 2\max_{i\neq j} d_{i,j}, \quad (3)$$

which is two times the network delay of the farthest peer pair in the MutualCast clique.

# 3. Mutualcast As Super Gateway or Super Server

Because a MutualCast clique must be formed by a set of fully connected nodes, and the MutualCast delay increases as the clique grows large, the number of nodes in a MutualCast clique shouldn't be very large. A good number is between 3 to 7. Nevertheless, the MutualCast clique can serve as a super gateway or super server, and thus function in a much larger network.

## 3.1 As a Super Gateway Node

One strategy is to let the MutualCast clique serves as a "super" gateway node in a large multiparty conference graph. In this case, the rest of the nodes form a generic graph, and use end system mixing, as shown in Figure 1c. This configuration is particularly suited for a multiparty conferencing session with a large number of peer nodes, where a small number of close-by nodes are fully connected and form a MutualCast clique. For example, in Figure 4, the MutualCast clique formed by the nodes $a$, $b$ and $c$ serves as a "super" gateway node for nodes $d$, $e$, $f$, $g$ and $i$. Each peer node in the MutualCast clique serves as a gateway for the nodes attached. For example, the peer node $a$ is attached to two nodes $d$ and $e$ outside of the MutualCast clique. Thus, within the MutualCast clique, the node $a$ merges the audio of $d$ and $e$

[2] If the slow/less powerful nodes are allowed to deliver fewer packets, they become leeches of the faster, more powerful peers. Whether to allow such leech behavior is a design choice between better conferencing performance vs. fairness on contribution.

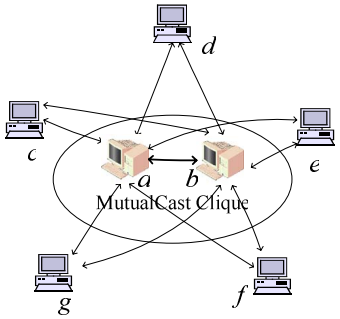

Figure 5. The MutualCast clique as a super server in a multiparty conference session with star-topology.

with its own, and delivers the combined audio ($a$+$d$+$e$) to the nodes $b$ and $c$ in MutualCast fashion. At frames that the node $a$ is mixing for the MutualCast clique, the combined audio ($a$+$d$+$e$) is mixed with the input from the node $b$ (audio $b$+$i$) and sent to the node $c$. Likewise, the combined audio ($a$+$d$+$e$) is mixed with the input from the node $c$ (audio $c$+$f$+$g$+$h$) and sent to the node $b$. The node $a$ also mixes the inputs from the nodes $b$ and $c$, combined it with its own,

$$m = a+b+i+c+f+g+h. \quad (4)$$

and sends $m$+$d$ to node $e$, and sends $m$+$e$ to node $d$. At frames that the node $a$ is not mixing, the combined audio ($a$+$d$+$e$) is sent to the mixing node ($b$ or $c$) at the moment. The node $a$ also receives the mixed input from the node $b$ or $c$, combined it with its own to form the mixed frame $m$, and sends $m$+$d$ to node $e$, and sends $m$+$e$ to node $d$.

## 3.2 As a Super Server

The MutualCast clique can also serve a "super" server. In this case, the rest of the nodes are the client nodes in a star topology, and the MutualCast clique may consist as fewer as two nodes. An example is shown in Figure 5, where the peer nodes $a$ and $b$ form a two node MutualCast clique, which serves as a super server for the rest client nodes $c$, $d$, $e$, $f$ and $g$. The peer nodes in the MutualCast clique again form a fully connected mesh. In addition, each client node outside is connected to all peer nodes in the MutualCast clique. This configuration is more suited for small-to-medium size networks, say 4-16, where there are a few powerful broadband nodes that share to serve as the server. Another scenario involved this configuration is due to the existence of NAT (network address translator) or firewall. The client nodes may be behind NAT/firewall. They can connect to the nodes that are directly connected to the Internet, i.e., those of the MutualCast clique. However, they can not connect to each other. The mixing and redelivery operation of such network is very similar to the MutualCast operation described in Section 2.2. The only difference is that the client nodes are exempted from the mixing/redelivery task. At each frame, one peer of the MutualCast clique mixes and redelivers the audio packets for the rest of the peers, both inside and outside the MutualCast clique.

# 4. Bandwidth and Computation Load Analysis

A prototype MutualCast multiparty audio conferencing system is being built. In this section, we calculate the bandwidth requirement of a few MutualCast scenarios, and compared those with the solutions not using MutualCast.

First, we consider an $n$-party conferencing session. If all peers are of equal bandwidth, the MutualCast requires the upload/download bandwidth of $(2\text{-}2/n)\cdot bw$ for each and every peer node. In particular, for a three node MutualCast clique, the bandwidth required is $1.34bw$. The same three node conferencing session needs a node with bandwidth at least $2bw$ to conduct a multiparty conferencing session with either star topology, or generic graph. Thus, MutualCast may conduct a multiparty conferencing sessions even when all peer nodes are less resourceful.

Second, we consider the case that the MutualCast clique serves as a super gateway in a large graph. Let the gateway node be connected to $m$ nodes. Normally, the gateway node needs $m\cdot bw$ upload and download bandwidth to mix and redeliver the audio traffic. By replacing the gateway node with an $n$-node MutualCast clique, and assuming $m/n$ nodes are attached to each MutualCast node, we may reduce the upload/download bandwidth requirement of each node in this super gateway to:

$$\left(2+\frac{m-2}{n}\right)bw. \quad (5)$$

As an example, let $m$=6 and $n$=3, MutualCast reduces the bandwidth requirement from $6bw$ to $3.34bw$. The use of the MutualCast clique as a super gateway may reduce the bandwidth requirement of the gateway nodes.

Finally, we consider the case that the MutualCast clique is used as a super server. We again assume that there are $m$ clients. Without MutualCast, the server needs $m\cdot bw$ upload and download bandwidth to serve the $m$ clients. By using a MutualCast clique of $n$ nodes, it can be calculated again that on average, each peer node in the MutualCast clique needs only $\left(2+\frac{m-2}{n}\right)bw$ upload/download bandwidth. As an example, let $m$=5 and $n$=2, MutualCast reduces the bandwidth requirement of the server node from $5bw$ to $3.5bw$.

It can be easily deduced that MutualCast reduces the computation load of the peer, the super gateway node and the super server node by the same proportion as well.

# 5. Conclusions

We propose MutualCast, a serverless P2P multiparty audio conferencing system. MutualCast rotates the role of audio mixing and redelivery among the constituent peers. At each instance (frame), one peer gathers the compressed audio from the rest of the peers, performs the mixing operation, and sends the mixed audio back to the rest of the peers. The mixing and redelivery task may be assigned in proportional to the resources of the peers. By sharing the network bandwidth and computation load among the peers, MutualCast may conduct a multiparty audio conference among less resourceful peers. The MutualCast clique may also be used as a super gateway node or a super server, and share the network bandwidth and the computation load of the gateway/server among the peers.

# 6. References

[1] K. Singh, G. Nair and H. Schulzrinne, "Centralized Conferencing using SIP", *In Proc of the 2nd IP-Telephony Workshop*, April 2001.

[2] J. Lennox, H. Schulzrinne, "A protocol for reliable decentralized conferencing", *In Proc of 13th international workshop on network and operating systems support for digital audio and video*, (NOSSDAV'2003), pp. 72-81, 2003, Monterey, California.

[3] M. Radenkovic, C. Greenhalgh and S. Benford, "Deployment issues for multi-user audio support in CVEs", *In Proc. ACM Symp. On Virtual Reality Software and Technology,* pp. 179-185, 2002, Hong Kong, China.

[4] "ConferenceXP: wireless classrooms, collaboration and distance learning", http://www.conferencexp.net/ .

[5] J. Li, P. A. Chou and C. Zhang, "Mutualcast: An Efficient Mechanism for Content Distribution in a Peer-to-Peer (P2P) Network", MSR-TR-2004-100, Sept. 2004.

[6] G.722.1, "Coding at 24 and 32 kbit/s for hands-free operation in systems with low frame loss".