2001-07

A Feedback Control Model for Managing Quality of Service in Multimedia Communications

Bordetsky, Alex

Telecommunication Systems; Jul 2001, Volume 17, Issue 3
http://hdl.handle.net/10945/43658

Telecommunication Systems 17:3, 349–371, 2001

A Feedback Control Model for Managing Quality of Service in Multimedia Communications

ALEX BORDETSKY and KEVIN BROWN {bord, kbrown}@csuhayward.edu
TELCOT Institute, California State University, Hayward, CA 94542, USA

LEANN CHRISTIANSON leann@csuhayward.edu
Department of Math and Computer Science, California State University, Hayward, CA 94542, USA

Abstract. Advances in Internet Protocol (IP) multicasting provide a rich background for support of IP multipoint collaborative communications. IP multicast applications have traditionally been deployed over wired networks, however, new wireless technologies such as Local Multipoint Distribution Service (LMDS) are currently emerging as transport mechanisms for wireless video, voice, and application sharing environments. For multimedia wireless services to effectively evolve, management tools that can support Quality of Service (QoS) adaptation of increasingly complex network resources and customer application profiles are needed. In this paper, we present a control model which provides response time and bandwidth requirement adaptation of audio, video, and application sharing multipoint IP teleconferences for emerging wireless multimedia communications. Our study is innovative in that it integrates feedback controls between the application and network layers. Our model is based on revealing feedback controls for multimedia call preparation and subsequent real-time connection control. Case-based reasoning memory is used to match real-time congestion (connection) controls with call preparation controls and user profiles for improved QoS. Network agents are used to capture user and multimedia teleconference application call profiles at the application layer and transfer them to the case memory. Real Time Protocol (RTP) statistics are used to identify connection management feedback controls at the network layer. Receiver-based, real-time adaptation at the network layer and above is possible through the use of a hierarchical coding technique. The proposed adaptive management architecture is based upon a case memory representation of call preparation feedback controls, RTP feedback controls for providing audio stream bandwidth adaptation, and configuration descriptions for integrated experiments. We conclude that implementation of these techniques should lead to improved QoS of wireless IP multipoint teleconferences.

Keywords: network management, quality of service, bandwidth adaptation

1. Introduction

Current advances in IP multicasting and MBone technologies provide a rich background for support of IP multipoint collaborative communications. By means of multipoint video, voice, and data communication, IP multicasting technology enables project managers and system analysts to access necessary human resources at any time. By means of application sharing and white board processing, it enables rapid transfer and sharing of knowledge.

Recently most of the IP multipoint multimedia applications have been restricted to experimental high-speed wired networking solutions. Introduction of Local Multi-

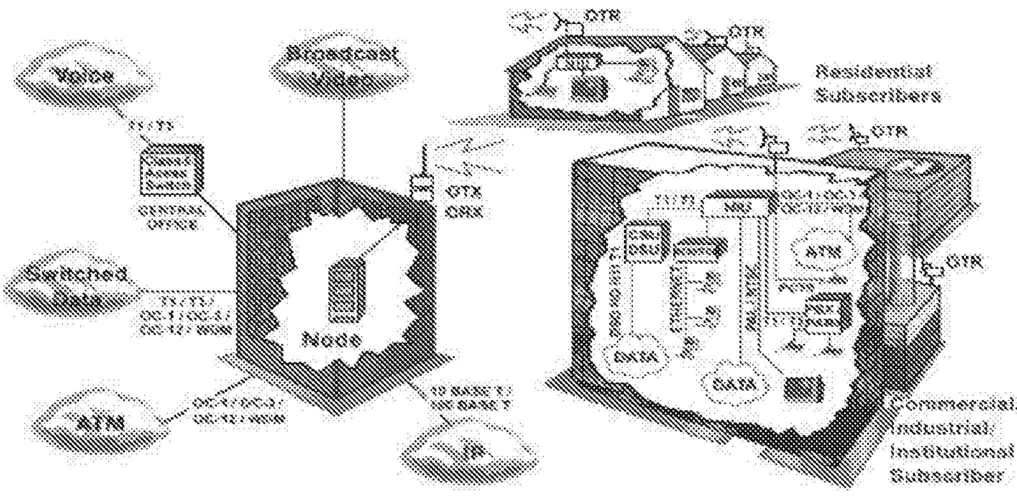

Figure 1. Typical LMDS architecture.

point Distribution Service (LMDS) represents one of several emerging opportunities to develop wireless multipoint video, voice, and application sharing environments. LMDS is a new wireless cell-based technology for interactive multimedia networks combining telephony, video services, high-speed data and integrated applications. The LMDS operates in the 28–30 GHz frequency range and is designed to operate in overlapping cells approximately 10 km in diameter. A typical LMDS (see figure 1) application can provide downlink throughput of 51.48–155.52 Mbps (SONET OC-1 to OC-3 speeds) and a return link of 1.544 Mbps (T1 speed). LMDS is protocol neutral, and can support ATM, TCP/IP, and other standards. Actual service carrying capacity depends on how much bandwidth is allocated to video versus voice and data applications.

In order for LMDS-based multimedia wireless services to effectively evolve, service managers need management tools that can support Quality of Service (QoS) adaptation to increasingly more complex networking resources and customer application profiles. This would include response time management, rapid re-configuration, and in some cases (e.g., IP over ATM) dynamic bandwidth allocation in accordance with content and customer communication profiles.

Our proposed model is based on tying TMN [1] Service Level Management functionality (see figure 2) to the fundamental concept of system coordination which involves identifying critical relationships [11] by revealing associated feedback controls. The process of adaptive control and coordination in our proposed architecture is based on capturing feedback controls, storing them in an agent's awareness memory, and delivering multimedia knowledge-sharing conferences via an ensemble of bridging, routing, and gateway agents-facilitators. In structuring the agents as agents-facilitators with bridging, routing, and gateway functionality we follow the evolving KQML concept [13] of agent communication models [6]. We expand the bridging, routing and gateway functionality into the agents' integration with case memory. Case memory supports the learn-

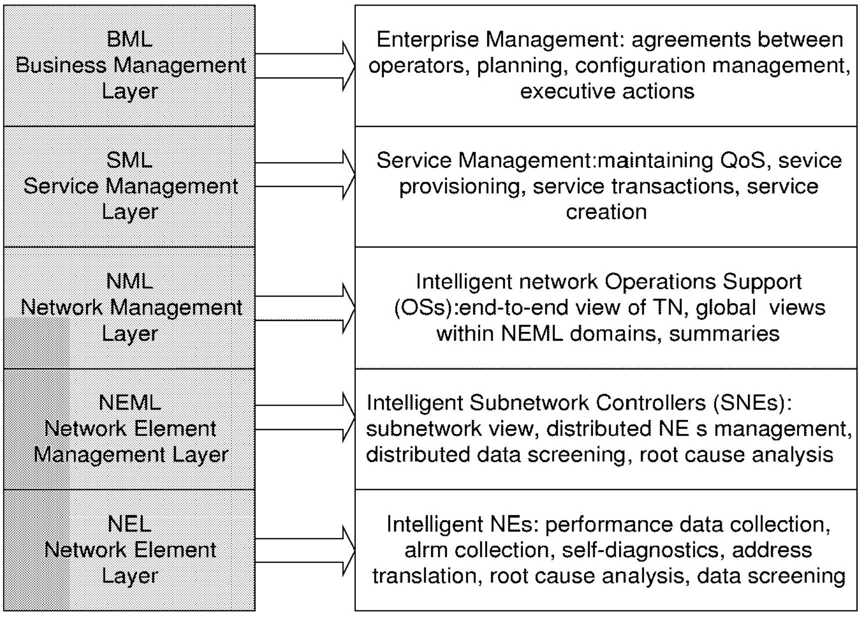

Figure 2. TMN intelligent management architecture.

ing of feedback control relationships and adaptive management of QoS requirements by utilizing a case-based reasoning technique [4,10] for indexing, capturing, and retrieving the feedback structures associated with IP multipoint multimedia conferencing events and QoS constraints.

Packet-switched networks in use today typically do not offer guarantees on minimum bandwidth or maximum delay. *Real-time* applications such as audio and video conferencing and shared application control, however, have stringent requirements regarding maximum delay and minimum bandwidth. A reduction in available bandwidth will result in loss of video frames, dropouts in audio streams, possible loss of synchronization between streams, and difficulties in shared control of applications as timing requirements may be exceeded. Hence in many cases, it is necessary for applications to *adapt* to the bandwidth available. Applications which are asynchronous in nature can adapt naturally, leading only to changes in response time. Real-time applications, however, may choose to reduce the quality of the data stream to reduce bandwidth needs.

On a single-application basis, work has been done in bandwidth adaptation for video applications (see section 5.1), and we describe a method to allow real-time audio application adaptation in section 4. When we consider multiple applications running simultaneously, lower-priority applications may be required to adapt to lower bandwidth usage or to be switched off entirely to free up bandwidth for higher priority applications. In this paper, we also propose a method for tracking user preferences and using that information to manage the bandwidth needs for multiple interacting applications in future conference sessions. We respectively consider two layers of feedback controls: *Call Preparation Control* (CPC) and *Connection Control* (CC). Call preparation control integrates feedback gathered from previous conferencing sessions to make informed decisions regarding connection setup and bandwidth tradeoffs in future sessions. Connection control reflects ongoing performance measurement and adaptation throughout the length of the call.

2. Layers of feedback control

Call preparation control requirements to support multimedia multipoint applications include the following:

- A call must establish, modify, execute, and terminate voice, video, and application sharing communication between multiple users.
- A call involves coordination between parties to satisfy their response time, bandwidth, and other QoS requirements.
- A call contains relationships between user profiles, media and system resources. These relationships may be dynamically modified during a call.
- Users can request resources individually.
- A call allows negotiations between different sites for system resources.

Connection control requirements could be summarized as follows:

- Supervision of provided QoS parameters.
- Provision of flow control, congestion control, routing, reservation, and renegotiation of resources.
- Modification and release of connections.

In terms of the length of a change's effects, call preparation control adaptation could be referred to as long-term adaptation, mainly associated with allocating resources for the entire length of a multimedia call. Conversely, connection control adaptation would deal with short-term adaptation, which might be required many times during a single call. Application adaptation to very short-term bandwidth changes (on the order of milliseconds) has been shown to be ineffective and possibly detrimental to connection quality. The problem is that the adaptation mechanism cannot keep up with the rate of change in the allocated bandwidth. There are, however, many opportunities to capitalize on *course-grained* bandwidth adjustments. Course-grained adaptation attempts to match application bandwidth usage to available bandwidth when changes last seconds or minutes, rather than milliseconds. As an example, consider the following scenario:

An Internet telephony application user is connected to the Internet via a microcellular wireless data network. In such a network, wireless devices common to a microcell must *share* the bandwidth available there. As such, user movements in and out of the cell, as well as user actions such as launching or terminating applications will cause the number of active connections within the cell to vary. As the number of connections varies, the bandwidth available for each will also vary. The bandwidth changes will occur at intervals of several seconds or longer, however, as they are the result of human interaction.

3. Call preparation adaptation: Application layer feedback controls and case memory

The architecture of the proposed adaptive management mechanism is represented by three components: a case-based reasoning memory, agents-facilitators, and collaborative feedback controls (see figure 3). The layers of case memory are structured according to the feedback control relationship for a multipoint teleconferencing service:

$$\text{SLM event}(t) = \{U(t), X(t), P(t), I(t)\}, \tag{1}$$

where SLM event(t) stands for a Service level management event, $X(t)$ is a set of SLM process state variables (QoS constraints such as response time and bandwidth), $U(t)$ is a set of user input controls (e.g., desktop video conferencing calls, links to knowledge sources), $P(t)$ is a set of service process outputs (e.g., the content of an electronic commerce transaction), and $I(t)$ describes the environmental impact to the service management process.

In accordance with the layered memory architecture of agents-facilitators, agents are divided into bridge or router agents which operate with different combinations of feedback control layers.

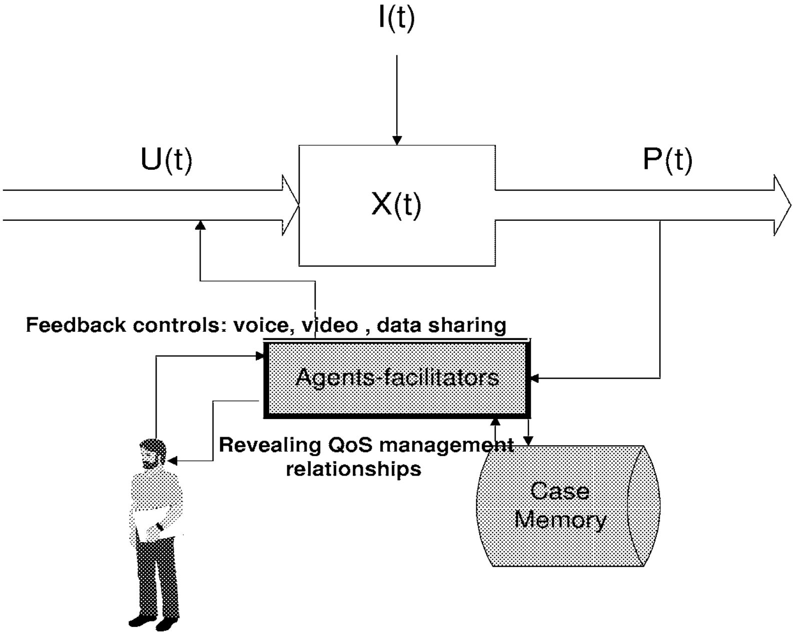

Figure 3. Adaptive management architecture: providing feedback controls.

In structuring the agents as agents-facilitators with bridging and routing functionality we follow the evolving KQML concept of agent communication models [6] by expanding the bridging and routing functionality into agents integration with case memory. This enables agents-facilitators to inform the agents on QoS profiles (bridging) and to associate QoS profiles with feedback control enabling processes (router functionality).

In our architecture, a Bridge Agent typically provides multicasting of $P(t)$ content and/or *X(t)* information in accordance with expression (1). In other words, a Bridge Agent simply informs the other agents about the state of the process it handles by posting the $\{X(t), P(t)\}$ information into the shared segments of case memory. Unlike the Bridge Agent, the Router Agent provides the link to the multipoint teleconferencing feedback controls $U(t)$ agents via the SLM event frame captured into the case memory:

$$\{U(t), \text{User View } (\text{SLM\_Event}(t))\}. \tag{2}$$

The feedback controls $U(t)$ could represent the calls to an RTP tool, recommendations to a human network operator, or calls to other agents. Correspondingly the Router Agent plays a major role in providing feedback controls and adaptation in service management. Its "routing table" is a case memory frame that associates user and application call preparation requirements with connection control processes and their agents. The Router Agent also provides user-memory transactions, supports capturing of communication parameters and the gathering of personal, document, and task profiles. It enables the location of appropriate human sources of knowledge and manages desktop video conference calls. It provides training and capturing of QoS management knowledge in case memory.

The knowledge retrieval model is a hierarchy of case memory layers (see figure 4), in which each interface between layers (from the bottom-up) is an association based on

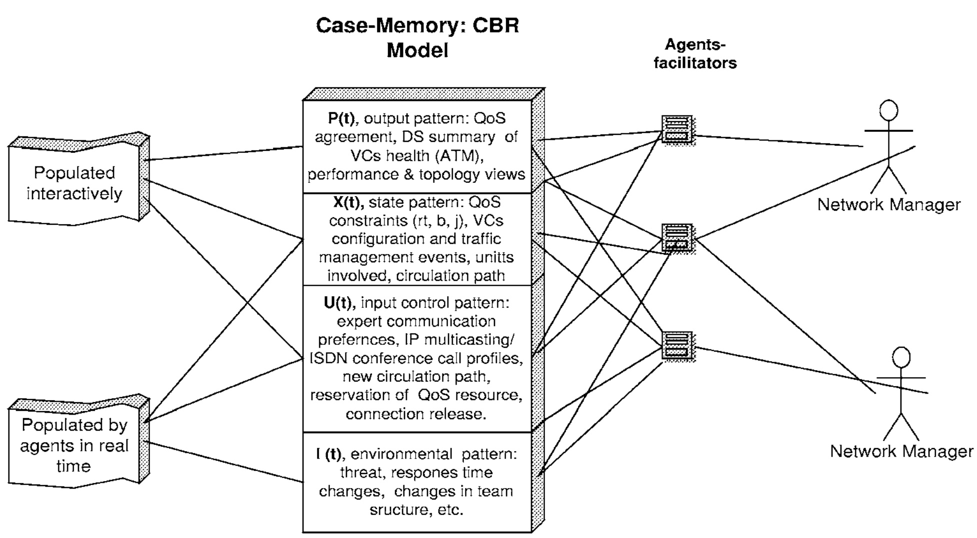

Figure 4. Feedback control model in case memory: associating $P(t)$ and $X(t)$ with $U(t)$.

the underlined feedback structure. The content profiles and user response time requirements are captured in real-time and populate the lower segment of the case memory stack. A sequence of application calls (content profile) and time stamps captured by an agent (see figure 5) are converted into response time and bandwidth requirements that populate the QoS segment of a case memory frame. Conversion is based on the QoS segment rules. As an example of QoS segmentation, consider the example illustrated in figure 5. The call to the shared Earth View 1 for Chains indicates sharing of a 2D Earth map with animated evaluation of LEO satellite constellation orbital performance. Such a view allows visualization of the access capabilities of selected terrestrial gateways, an important consideration when purchasing satellite services. The consumer and seller would typically discuss potential service scenarios by remotely sharing the controls of the Earth-View-Chains window. This would typically require about 0.2 Mbps of bandwidth between the two conferencing sites. In many cases, such a rate could be satisfied by an IP multicasting videoconference without voice. Voice would need to be diverted to a separate dial-up channel in order to free enough bandwidth for this exchange.

The addition of a STK/VO module into the shared environment seller/consumer multimedia conference (the next item in figure 5) would require more significant changes in order to keep up with the response time requirements. (For example, less than 3 second end-to-end delay.) The new component, named STK/VO, allows 3D views of satellite operations and ground station and in-orbit inter-satellite access link operation. It

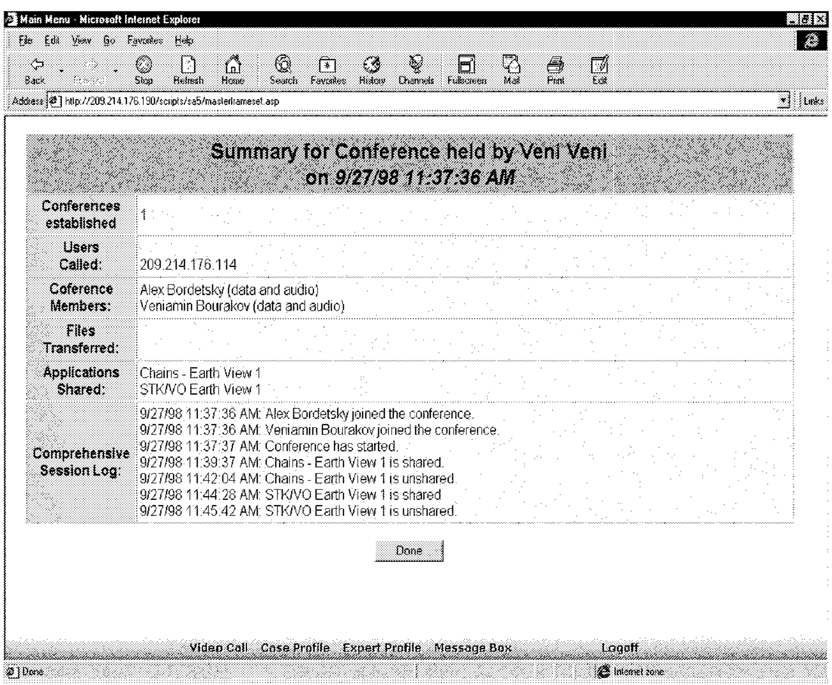

Figure 5. Mapping $X(t)$ to $U(t)$: capturing response time requirements and content profiles into the agents case memory.

dramatically improves the consumer's understanding of what to expect from the purchased service, but requires an extra 2 Mbps or more for each two-way, point-to-point IP conferencing channel. In order to satisfy the expected response time requirements and maintain a reasonable quality conference with the consumer, the agent may need to alternate between shared window monitoring and video stream display. Alternatively, the agent may switch to a different access technique, such as integrating the IP STK/VO application sharing by re-routing the video stream to an ISDN point-to-point link (if available).

If more than two participants are engaged in the seller/consumer conference, then satisfying the content profile (see figure 5) could require even more substantial changes in the communication resources distribution for an SLM event. For example, sharing of an Earth-View-Chains map would in this case require reservation of at least $(n(n-1)/2) \times 2$ Mbps, where n is the number of participants. Depending on the Internet access rate that is available at each conference site, the whole system of QoS constraints could become infeasible, or could require multiple alternations of the SLM event stream forwarding. An alternative to alleviate this situation is the following. Suppose that an SLM event profile is described by:
$QoS(1)$ = preferred bandwidth for voice,
$QoS(2)$ = preferred bandwidth for video,
$QoS(3)$ = preferred bandwidth for white board, and
$QoS(4)$ = preferred bandwidth for application sharing.
According to such a profile, each conferencing node has associated voice, video, whiteboard, and application sharing delivery trees. Switching between these delivery trees could help to satisfy otherwise infeasible response time requirements.

Correspondingly, the QoS segment of the case memory is expanded by rules and heuristics that allow the generation of non-dominated minimal spanning trees based on the measures, such as the following one, suggested by B. Peltsverger:

$$w_{i,j}(k) = w_{i,j}(k-1) - q_\tau(k)QoS(k), \tag{3}$$

where $k \in \{1, 2, 3, 4\} = K_\tau \subseteq \Phi$, K_τ – a set of IP conferencing tasks that are used on an interval of time $\tau(mt_0 \leqslant \tau \leqslant (m+1)t_0,\ m = 0, 1, 2, \ldots)$, Φ – a set of possible multimedia multipoint conferencing tasks, $w_{i,j}(0)$ – the initially available bandwidth, and each pair $(i, j) \in E$ identifies the seller/consumer conferencing nodes.

A Router Agent is integrated with feedback control associations $\{P(t), X(t), U(t)\}$ via the case memory. This data model for integration will be illustrated in section 5.3. The functionality of the case memory is provided by web integrated dynamic case frames, a case-based reasoning inference engine, and database tables. The database management system is used to keep actual input, output, and state attributes of QoS profiles that are captured and adopted by the case memory. Figure 5 illustrates how the event log that an agent provides reveals the response time requirements and content profiles that are captured into the lower segment of the case memory stack (see figure 4) associated with the seller/consumer conferencing transaction.

4. Connection control adaptation

As described in section 2, connection control requirements include:

- supervising QoS parameters,
- providing flow control, congestion control, routing, reservation and renegotiation of services,
- modifying and releasing connections, and
- notifying applications to allow them to adapt.

As opposed to the call preparation control, in which decisions are made *before* the call is made, connection control is done on an ongoing basis throughout the duration of the call. Feedback regarding network conditions must be continuously collected and processed in order to allow the applications in use to adapt. The most dynamic network resource in wired and wireless networks is allocated channel bandwidth. This is where we concentrate our efforts in network layer feedback controls.

In a multicast environment, each participant in a call may be connected via a different access media and may be allocated varying amounts of bandwidth, perhaps differing in orders of magnitude (e.g., LMDS vs. a standard modem). Hence, it is not reasonable for the source of a data stream to attempt to adapt to the bandwidth used by the stream. A bandwidth usage solution which is acceptable to one participant may well result in a connection of unacceptable quality for others. In the multicast environment then, the *destination* of a data stream must be responsible for monitoring its own network resources and for driving adaptation of its received input stream based on the link bandwidth available. What is required is a standard mechanism for communicating the receiver's current network status to the applications in use for the current call. There are numerous in-band and out-of-band possibilities, but a commonly used mechanism is the Real Time Protocol (RTP).

4.1. The real time protocol

At the transport layer, the real time protocol [16] is used to support multimedia traffic on the Internet. Some of the benefits of using RTP are that it does not require changes to existing routers or gateways, it may be implemented on top of UDP/IP or ATM, and it can take advantage of the multicast backbone to provide efficient delivery of data. RTP is made up of two components: a real-time data transfer protocol (RTP) and a control protocol (RTCP). RTP does not assume virtual circuits at the network layer, and prepends an RTP header including a sequence number to each data packet to allow re-ordering at the receiver. This header also includes a timestamp, and a Synchronization Source (SSRC) field. The SSRC field may be used to identify the media source independently of the transport protocol used (for instance to differentiate data streams received on the same UDP port). Data marked with the same SSRC is grouped together for playback at the receiver.

The Real Time Control Protocol (RTCP) performs quality of distribution monitoring, intermedia synchronization, and participant identification. Quality of distribution monitoring is done via sender and receiver status reports, (see figures 6 and 7) which each participant generates periodically and multicasts to the other participants of the RTP session. Sender reports (SR) include the SSRC ID for the data source and the total number of packets and octets sent since the source started transmitting. Receiver reports (RR) are generated by each receiver to indicate its current loss ratio, jitter, and highest sequence number received from the source. These reports allow the call participants to detect reception problems in the network and to possibly adapt in some way to compensate.

<table>
<tr><td>Ver</td><td>Pad</td><td>RC</td><td>PT</td><td>Length</td></tr>
<tr><td colspan="5">SSRC of Sender</td></tr>
<tr><td colspan="5">NTP Timestamp, Most Significant Word</td></tr>
<tr><td colspan="5">NTP Timestamp, Least Significant Word</td></tr>
<tr><td colspan="5">RTP Timestamp</td></tr>
<tr><td colspan="5">Sender's Packet Count</td></tr>
<tr><td colspan="5">Sender's Octet Count</td></tr>
</table>

Figure 6. Format of an RTP Sender Report (SR).

<table>
<tr><td>Ver</td><td>Pad</td><td>RC</td><td>PT</td><td>Length</td></tr>
<tr><td colspan="5">SSRC of Sender</td></tr>
<tr><td colspan="5">SSRC_1 (SSRC of First Source)</td></tr>
<tr><td colspan="3">Fraction Lost</td><td colspan="2">Cumulative # of Packets Lost</td></tr>
<tr><td colspan="5">Extended Highest Sequence Number Received</td></tr>
<tr><td colspan="5">Interarrival Jitter</td></tr>
<tr><td colspan="5">Last SR (LSR)</td></tr>
<tr><td colspan="5">Delay Since Last SR (DLSR)</td></tr>
</table>

Figure 7. Format of an RTP Receiver Report (RR).

4.2. Design

Given these RTP reports as a mechanism for reporting network performance, we need to provide a means of adaptation for applications which experience dynamic bandwidth conditions. We will concentrate on an audioconferencing application as representative of the types of applications commonly used in a multicast teleconference. Bandwidth adaptation of the received data stream may be achieved in the following manner:

(1) The data source *hierarchically* encodes the audio stream and separates the levels of encoding into n separate data streams.

(2) Each stream is multicast to a separate multicast address.

(3) Receivers determine their current bandwidth allocation and subscribe only to a number of data streams which they can feasibly receive.

(4) The individual streams are reassembled and played back.

Hierarchical encoding has been used successfully in many image and video applications. It is useful as it allows each user to choose their own acceptable level of quality. Data transmission is curtailed when the desired level has been reached. Hierarchical image encoders often use transformation techniques such as the Discrete Cosine Transform (DCT) or wavelets to transform the digitized samples into a new representation. Larger magnitude transform coefficients represent average or coarse characteristics while smaller coefficients add detail. Hierarchical encoding involves organizing the transform coefficients based on overall importance to reconstruction quality. The coefficients which contribute most to reconstruction, generally those representing average characteristics, are transmitted first with detail coefficients following.

Note that the receiver may well choose not to accept all of the components of the original data stream (due to bandwidth limitations). In this case, the reconstructed stream will offer lower dynamic range than the original. It will however be continuous and will not suffer from dropouts and long silence periods. Adaptation to current network conditions may be achieved by subscribing to more of the available data streams when bandwidth is plentiful and dropping subscriptions from a number of streams when bandwidth is restricted.

For this study, hierarchical encoding of audio samples was accomplished by creating 4 groups of 4 bits each from the original 16 bit sample (see figure 8). Group 1 is the base group and consists of the upper 4 bits (15–12). It is the lowest resolution

A 16 bit Audio Sample

15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Group 1	**Group 2**	**Group 3**	**Group 4**

Figure 8. An example of how the audio sample is divided into groups representing different resolutions.

group and is required by all receivers. The next 4 bits (11–8) represent group 2, followed by group 3 bits (7–4), and the lowest 4 bits (3–0) represent group 4. Samples are packed together on a per-group basis and sent as separate data streams to the destination where they are re-assembled for playback. Hence, the data source multicasts 4 separate streams corresponding to the 4 groups. As a receiver subscribes to more groups, they receive increased resolution and should expect higher quality audio.

Consider that multiple adaptation models are possible. Adaptation may be controlled by the user, based on an indication of network performance. Adaptation may also be done automatically by the application based on RTP statistics. We chose to provide automatic adaptation based on the loss ratio signaled in the RTP receiver reports. Thresholds for maximum acceptable loss and minimum detectable loss were set. We then chose a very simple adaptation algorithm. If three consecutive RTCP receiver reports are produced by this receiver indicating that the current loss rate is above the specified maximum loss rate, then the number of subscribed groups is reduced by one. Groups continue to be dropped until only the base group remains or the loss ratio improves. If the loss ratio improves such that it is less than the minimum detectable loss ratio (possibly due to an increase in allocated bandwidth), then groups are added, up to the maximum number of available groups. Three consecutive reports were required before an adaptation in either direction in an effort to reduce oscillations between group levels. This is similar to the approach used for adaptation of video bandwidth in the ivs videoconferencing tool as presented by Bolot and Turletti [2]. Note that these adaptation control mechanisms are not stable, however, due to the finite set of bandwidth levels supported by the multimedia tools. Should the actual available bandwidth lie between two supported levels, oscillation will occur. A more sophisticated control mechanism is required which can recognize and control oscillations as well as differentiate temporary fluctuations in available bandwidth from continuing trends. Such a mechanism is proposed in section 5.

4.3. Implementation

As the basis for our development effort, we chose to use the rat (robust audio tool) audioconferencing tool developed by Vicky Hardman and Isidor Kouvelas at the University College London [7]. Rat supports both multicast and unicast modes and uses the RTP protocol on top of UDP/IP. Rat provides many options for improving audio transmission quality such as forward error correction implemented by sending redundant packets. Adaptive scheduling protection is also provided. Receiver based repair of damaged audio streams is supported through packet repetition, silence substitution, and pattern matching.

Enhancements to the rat application were required to provide support for hierarchical encoding of data streams at the source, support for multiple multicast streams at the source and destination, and reconstitution of individual streams at the receiver. The receiver was given the option of specifying thresholds for minimum and maximum packet loss. If thresholds are specified, the number of subscribed groups may change over time.

The number of subscribed groups will decrease if network conditions at the receiver indicate that the current loss rate is greater than the maximum loss threshold, and it will increase when network conditions improve (loss rates drops) beyond the minimum loss threshold.

Note that the source will send all four groups regardless of the receiver's subscriptions. It is the multicast routers, acting on the receiver's wishes, which filter the data streams and forward only the requested streams. This means that any adjustment of groups will occur completely at the receiver and does not require any actions on the part of the data source.

As data packets from each audio stream or group reach the destination, they are combined with the corresponding data packets from the other streams or groups. A composite RTP packet is created for decoding purposes, which contains data from each of the subscribed groups. The RTP header byte of this packet indicates the number of groups within the packet. The possibility exists that a packet from one particular data stream or group will not be received in time to be combined with the others. In this case, the data packet is not combined with the others, and the number of groups in the RTP header is decreased to reflect the change. Groups must be present in numerical sequence, and the base group (1) is always required. For example, if the receiver has subscribed to 4 groups, but only data from groups 1, 2, and 4 are present at the time the data needs to be passed to the decoder, the number of subscribed groups will be changed temporarily to 2 for decoding purposes of this particular packet. If all packets from each data stream are received in time at the next interval, the number of subscribed groups will again be 4. When the decoder receives the new RTP packet, it retrieves the number of groups present from the header and pulls data in 4 bit increments from each group, combining the information into samples of the appropriate size, and sending them to the audio device for playback. If groups are missing, or the receiver has chosen not to subscribe to them, those portions of the 16 bit sample will be set to 0.

4.4. Performance results

Testing was performed between a 300 MHz Pentium PC running RedHat Linux 4.2 and a 150 MHz Pentium PC also running RedHat Linux 4.2. These machines were at a distance of approximately 0.5 miles from each other. All transmission and reception from these machines was executed in unicast mode and took place in the early evening hours. In order to simulate restricted bandwidth, the rat *drop* option was used. This option allows the user to choose a particular packet loss (drop) rate. Packets are then randomly dropped at this rate, and are therefore not received at the destination. Tests were performed to observe the bandwidth adaptation process which added and subtracted groups.

In figure 9, we see an example where the receiver has subscribed to 4 groups (128 Kbps). Allocated bandwidth was restricted to 64 Kbps and maximum allowable packet loss was set to 5%. Initially, the loss rate was high (55%). After three consecutive receiver reports indicating loss above the maximum allowable value, the number of groups was dropped to three. Loss was reduced but was still too high, therefore, another

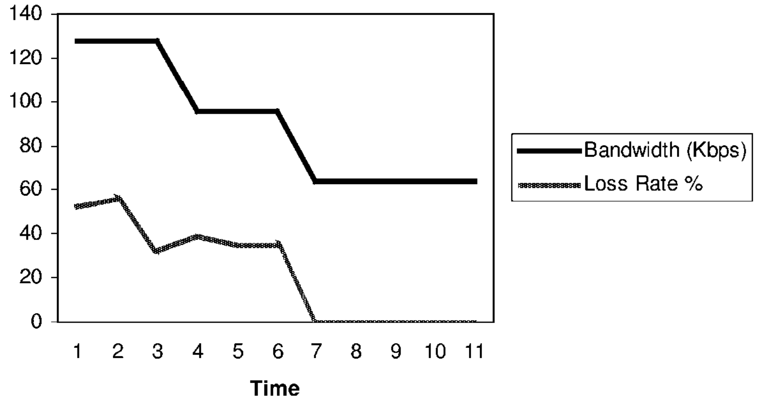

Figure 9. Downward bandwidth adaptation. Initial data rate is 128 Kbps with four subscribed groups. Final data rate is 64 Kbps with 2 subscribed groups.

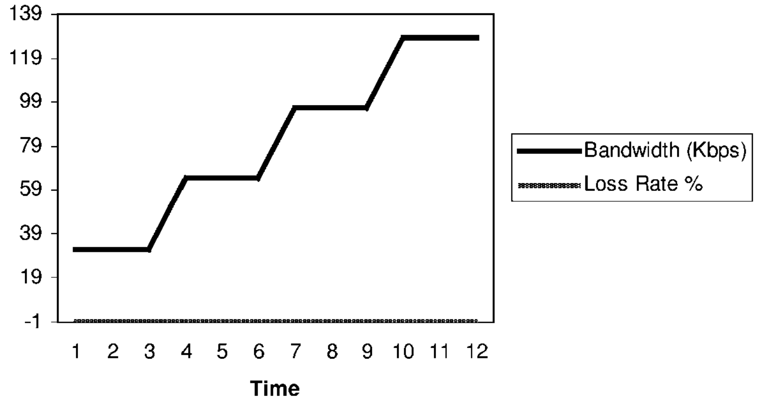

Figure 10. Upward bandwidth adaptation. Initial data rate is 32 Kbps with one subscribed group. Final data rate is 128 Kbps with four subscribed groups.

group was dropped. At 2 groups (64 Kbps) the loss rate dropped below 5% and the adjustment process stopped.

Figure 10 illustrates an increase in allocated bandwidth which triggers an addition of groups. Initially the receiver is subscribed to 1 multicast group representing a bandwidth of 32 Kbps. Allocated bandwidth is set to 150 Kbps and the minimum loss rate is 20%. Measured loss remains at 0% throughout the test, therefore, groups are added incrementally. At 4 groups, the receiver is still not seeing any packet loss, and transmitted audio data bandwidth is 128 Kbps.

These results show that bandwidth adaptation can be used to match offered load to allocated bandwidth. It is reasonable to expect that, over time, many calls will exhibit similar behavior. The current adaptation mechanism does not learn from experience, either from events which take place over the course of the current call or from previous calls. It is here that call preparation control methods may be used most effectively.

5. Integration

Again, using the audioconferencing example from the previous section, we can identify several scenarios where call preparation control would be useful:

- Specifying the initial number of multicast groups to which to subscribe.
- Specifying the number of consecutive report intervals which should trigger adaptation.
- Specifying the levels of loss which are significant, both for indicating congestion in the network and the absence of congestion.

As most teleconferences will consist of many components including audio, video, and shared application control, it will also be necessary to balance the bandwidth needs of each individual tool. In this way, video streams may be constrained to black and white images in favor of high quality audio, or lower priority streams may be shut off entirely in favor of higher priority streams.

5.1. Filtering and identifying constraints of the RTP test log

Various work has been done in the area of Quality of Service adaptation. We will describe several of the mechanisms that have been proposed to facilitate adaptation, and the filters which are used to drive the adaptation decisions.

In the Consenting Equal Division (CED) [15] policy, either the end systems or the network may initiate adaptation. The end systems specify a *range* of acceptable values for each QoS parameter, including bandwidth. They also specify the maximum step size in which each parameter may be changed when adaptation occurs. When a new connection is requested or an end system with an existing connection requests a higher level of service, the network first determines if the request can be granted using free resources or resources relinquished by end systems currently receiving more than their minimum specified level of service. If so, the network divides the additional resources needed among all the end systems which have agreed to adaptation. It then sends a *consent* packet to each of these systems to inform it of the requested adaptation. Upon receiving an acknowledgement, the network reduces the level of service provided to each of the existing connections and uses the resources to set up the new connection. End systems which wish to increase their level of service again in the future must explicitly request the additional resources. The CED adaptation policy requires strict resource reservation facilities, however, to allow the network to track the current network usage, and does not address the multicast environment.

In [5], RTP is used as a feedback mechanism and a multicast environment is assumed. Adaptation is done at the *source* of the data stream based upon the loss ratio indicated in RTCP receiver reports. Each receiver is placed in one of three states, congested, loaded, or unloaded, based on a smoothed estimate of their loss rate. Newly received reports of loss are weighted against previous reports to prevent oscillations in the adaptation mechanism. The data source then calculates the proportion of receivers in the congested and loaded states. If the proportion of receivers in a congested state is over a specified threshold, a decrease is made in the bandwidth used by the source. If the proportion of receivers in a loaded state is over a separate threshold, then the bandwidth usage is left the same, otherwise it is increased. When a bandwidth decrease is called for, it is done multiplicatively, but bandwidth increases are done additively. Any adaptation is done within the bounds of a specified maximum and minimum bandwidth. This mechanism requires that one level of bandwidth usage fit the needs of *all* receivers, which is clearly not possible in a heterogeneous network including receivers connected via wireless links, modems, T1 lines, etc. A receiver with large amounts of bandwidth available would be made to suffer with a low quality data stream if most of the other receivers have little bandwidth available.

Hoffman and Speer [8] suggest that the source transmit a hierarchically layered data stream. The layers are each multicast as separate streams allowing each receiver to select its own bandwidth usage subscribing or unsubscribing from multicast groups. The authors provide two mechanisms for adaptation. In the first, a resource reservation mechanism is required to allow negotiation of bandwidth usage between the receiver and the network. In the second, each receiver subscribes to *all* of the multicast groups initially and drops groups until connection quality improves to an acceptable level.

In [14], the authors also propose that the source multicast a layered set of data streams. Receivers drop multicast groups when the network gets congested, which is signaled by lost packets. They add groups when the network has spare bandwidth. This spare bandwidth is detected via *join-experiments.* A receiver adds a group when congestion appears to be low and evaluates the results. If congestion occurs, the receiver immediately drops the group again. Each receiver uses an exponential backoff technique to ensure that join-experiments are not done too frequently when they are likely to fail, but are done often enough when they are likely to succeed. Receivers multicast their intent to conduct a join-experiment so that other participants do not misinterpret transitory congestion and drop groups unnecessarily. This adaptation mechanism is very advanced but it does not learn from previous connection adaptation decisions nor is it customized to an individual receiver's behavior.

The Self Organized Transcoding (SOT) method described in [9] relies on intermediate nodes along the path from the source to the destination for bandwidth adaptation. These *transcoders* take the data stream arriving from the source and recode it to use less bandwidth. In a multicast environment, multiple receivers make use of the same transcoder by electing a representative who controls the actions of the transcoder. Both transcoding representatives and the provider of the transcoding service itself are active receivers of the data stream. Receivers which see congestion in the network send a request for transcoding services by multicasting an indication of their loss pattern. Loss patterns consist of a bitmap showing which packets have been received and the highest sequence number received. Receivers which are willing to act as transcoders and which have better loss patterns respond and the transcoder closest to the group requiring those services is chosen. After the group has switched over to the transcoded data stream, the representative provides feedback to the transcoder regarding loss experienced. The transcoder uses a mechanism similar to the TCP congestion control algorithm to adapt to current network conditions. It halves bandwidth usage when congestion is detected and increases bandwidth usage additively under low loss conditions. For efficiency reasons, this mechanism requires that a reasonable percentage of receivers are willing to act as transcoders, and that groups of co-located receivers will have similar bandwidth allocations.

5.2. *A feedback control model*

As shown above in figure 7, RTP receiver reports provide feedback in the form of loss ratios, highest sequence numbers received, and jitter values on a per-stream basis. For our study, these values are logged and made available for post-processing. Let each loss ratio report for four multimedia components, audio, video, shared application, and white board be represented by the vector:

$$\Delta \boldsymbol{P}_i = (\Delta p_{i1}, \Delta p_{i2}, \ldots, \Delta p_{in}), \tag{4}$$

where $i = 1, 2, 3, 4$, Δp_{i1} denotes the loss ratio, Δp_{i2} stands for the highest sequence number received, and Δp_{i2} could be used to specify jitter.

If the observed combination of $\{\Delta p_{ik}\}_1^n$ values is judged acceptable for processing without immediate bandwidth adjustment, then the inequality is set up to be negative. If an expert (e.g., a network manager/operator) evaluates this vector as indicating that bandwidth adaptation is necessary, then a non-negative value is set up. The expert responses consolidated during the knowledge acquisition (training) phase would constitute an integrated system of the form:

$$\begin{aligned} &\sum_{j=1}^{n} (W_{ij} \cdot \Delta p_{ij}) \geqslant 0, \\ &\sum_{j=1}^{n} (W_{ij} \cdot \Delta p_{ij}) < 0. \end{aligned} \tag{5}$$

Solution vector $\boldsymbol{W} = \{W_{ij}\}$ for system (5) is used to identify the filter, as a discriminant linear function for audio, video, shared application, and white board streams:

$$\boldsymbol{W}_i \cdot \Delta \boldsymbol{P}_i \geqslant 0. \tag{6}$$

In many cases, the same training vector $\Delta \boldsymbol{P}_i$ could be evaluated as satisfactory for the video stream (i.e., no need to initiate short-term bandwidth adaptation), but at the same time be evaluated as requiring bandwidth adjustment for the voice stream. This would create conflicting constraints in system (5) and would result in a state of infeasibility. When system (5) becomes infeasible, it is not possible to identify a single discriminant function (6). The solution requires a *set* of QoS discriminant functions.

5.3. Hierarchy of QoS discriminant functions: ANN model

How can we facilitate learning and upgrading of $\boldsymbol{W}_i$ solutions for the set of QoS discriminant functions (6)? We implement the following model of a four-layer Artificial Neural Network (ANN) that provides a hierarchical structure of discriminant functions capable of learning changes in the $\boldsymbol{W}_i$ coefficients.

Input layer
The *input layer* represents the learning vector $\Delta \boldsymbol{P}_i$ in which each input node stands for an aspiration-reservation interval for a single constraint $[RL_k, AL_k] = \Delta p_k$ (e.g., loss ratio interval, jitter interval, etc.).

First hidden layer
The first hidden layer represents the discriminant functions for the revisions $\{\Delta p_k\}$ that experts evaluate as "good" or "bad" for initiating RTP bandwidth adaptation without any contradiction. Each of the nodes in the *first hidden layer* represents *one* linear discriminant function $\boldsymbol{W}_i \cdot \Delta \boldsymbol{P}_i \geqslant 0$ that exactly separates "good" and "bad" revisions of $\{[RL_k, AL_k]\}$ intervals. Weights w_{ij} which are the coefficients of discriminant functions, are subject to changes in the process of training and are determined as feasible solutions for a system of constraints in a training sequence (6).

Second hidden layer
Nodes of the *second hidden layer* match the training cases in which revisions of $\{[RL_k, AL_k]\}$ intervals for the shared constraints are conflicting, e.g., patterns of "good" and "bad" QoS are overlapping. In this case, the set of training constraints is infeasible.

Given that the conflict situation in satisfying QoS constraints is caused by agent requests for the same resources on behalf of the different conferencing tasks (nodes), it seems natural to enable learning of W_i coefficient changes that belong to the second hidden layer in a way similar to multi-participant decision making (see figure 11). According to Marakas [12], different multi-participant decision making dependencies are comprised of three basic models: group, team, and committee.

In the group model, the structure of information flows is a mesh network. The team model represents a more centralized pattern of a single decision-maker with no participant interaction. The primary topology type for conflict resolution is a star. The third basic model is the committee model (see figure 12). It allows collective behavior that is based on different types of majority rules or consensus protocols. A combination of star and ring topologies could be used to support conflict resolution relationships.

A group multi-participant structure may not be the most appropriate prototype for multiple agent conflict resolution as it relies on a mesh topology and does not separate the facilitator (coordinator) from the other members. Unlike it, a team topology naturally allocates a role for the decision-maker (facilitator), but it lacks cooperative relationships among the members, which could be critical in the join constraint satisfying process. From that standpoint, the committee model represents a reasonable compromise

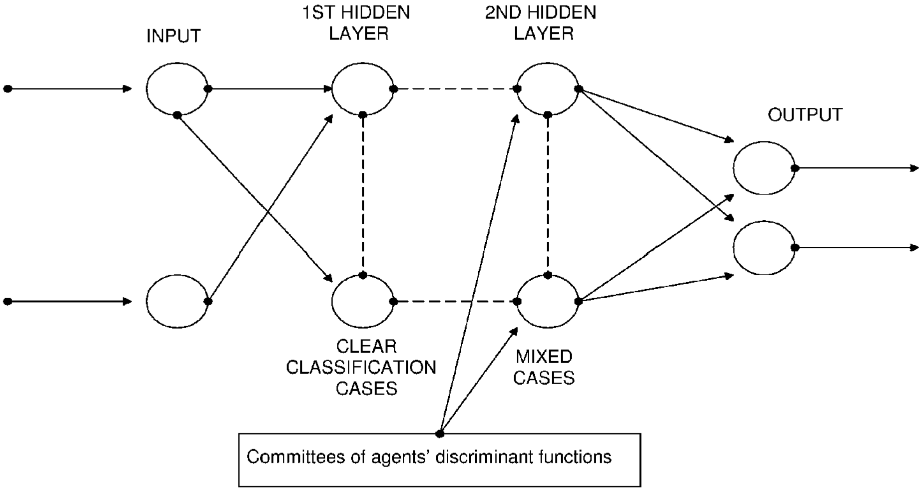

Figure 11. Artificial neural network model of QoS discriminant functions.

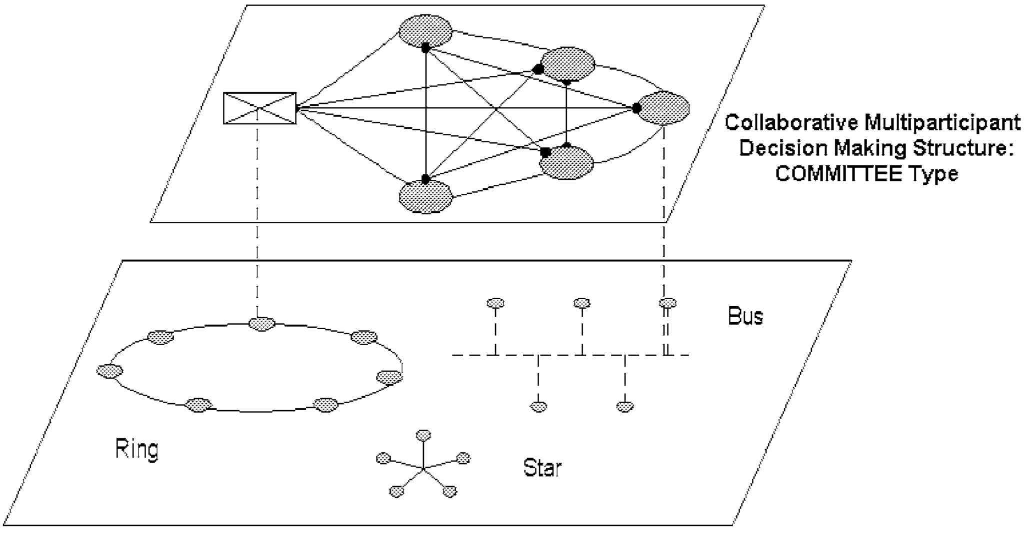

Figure 12. Committee model.

between the group and team multi-participant structures. It allows a facilitator (coordinator) role and compensates for the lack of participants' interaction that is typical for the team structure. Based on the described consideration we adopt a committee model for structuring the QoS constraints conflict resolution process.

Each of the nodes in the second hidden layer represents a committee of discriminant functions. This is a committee of solutions, where the set of weight vectors satisfies more than half of the inconsistent constraints in the system. More precisely, each node of the second hidden layer has a threshold function:

$$F(\underline{w}) = \sum_k \operatorname{sign}(\underline{W}_k \cdot \Delta \underline{P}_k), \tag{7}$$

where $\operatorname{sign}(\cdot) = \{1, 0\}$. If $F(\underline{w}) > (m+1) + r$, where m is the number of members in the committee $\underline{w} = [\underline{w}_1, \ldots, \underline{w}_k, \ldots, \underline{w}_p]$, and r is the ratio of participation (usually one half). When the node fires, the adjacent vectors $\underline{w}_i$ are taken as the coefficient vectors for related empirical constraints.

The selection criteria for the committee of constraints may vary. In the case where weights are equal, the selection criterion is a simple majority rule. The learning process will produce the union of the initial discriminant functions and the set of developed (learned) empirical constraints that represents RTP bandwidth adaptation experience. By capturing and updating such constraints concurrently with connection control sessions, the neural net will represent an adaptive filter for interfacing short term feedback controls with call preparation controls captured into the case memory.

5.4. *Associating filters with application profiles: Call preparation and connection control links in case memory*

For a single function filter the discriminant function (6) is placed into the QoS segment of the case-based memory (see figure 3) stack that contains the associated segment of

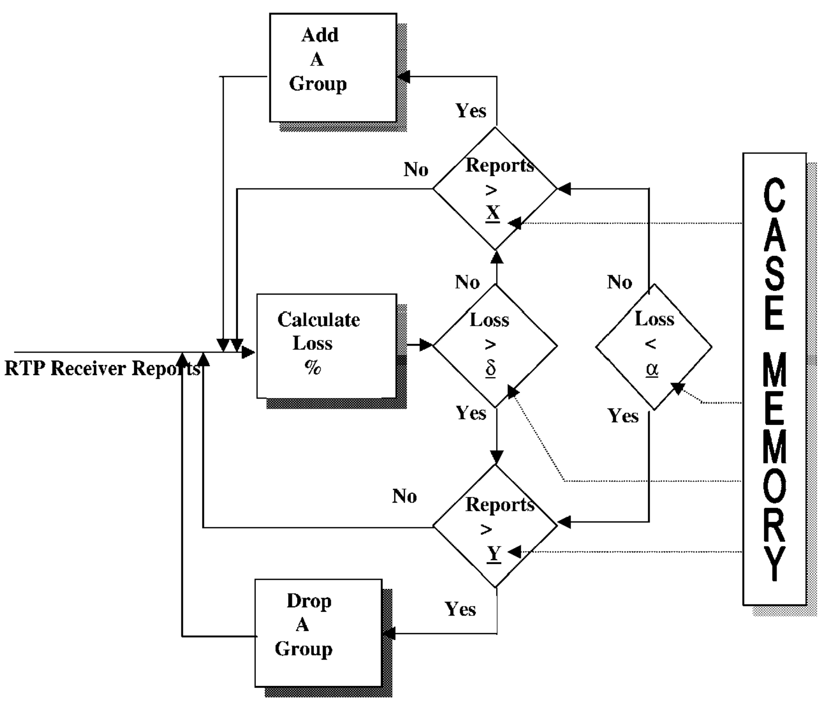

Figure 13. Adaptation control driven by case memory.

application layer feedback controls (see figure 4) and user profiles. Thus the RTP test log becomes associated with the call preparation control via the case-based reasoning feedback control index (1) (see figure 13).

When the connection control process begins, agent-facilitators check the observed values of ΔP by plugging them into the discriminant function (6). If the value of $\boldsymbol{W}_i \cdot \Delta \boldsymbol{P}_i$ is positive, the agent-facilitator transfers control to the RTP bandwidth adaptation tool for providing immediate bandwidth adjustments. When the ANN filter is used, the same integration process takes place. The difference is that, in this case, the QoS filter segment is populated by a set of objects structured into the two hidden layers of the described ANN model. Now it is not a single discriminant function that is used to define whether to initiate the RTP adaptation tool, but rather one or more nodes of the second hidden layer each representing different discriminant function committees. Each application layer profile (see figure 5) in turn is associated through the case memory index with one, or more committee nodes (see figure 14).

The other difference is that when the connection control process starts, an agent-facilitator checks the observed values of ΔP by plugging them into the first layer dis-

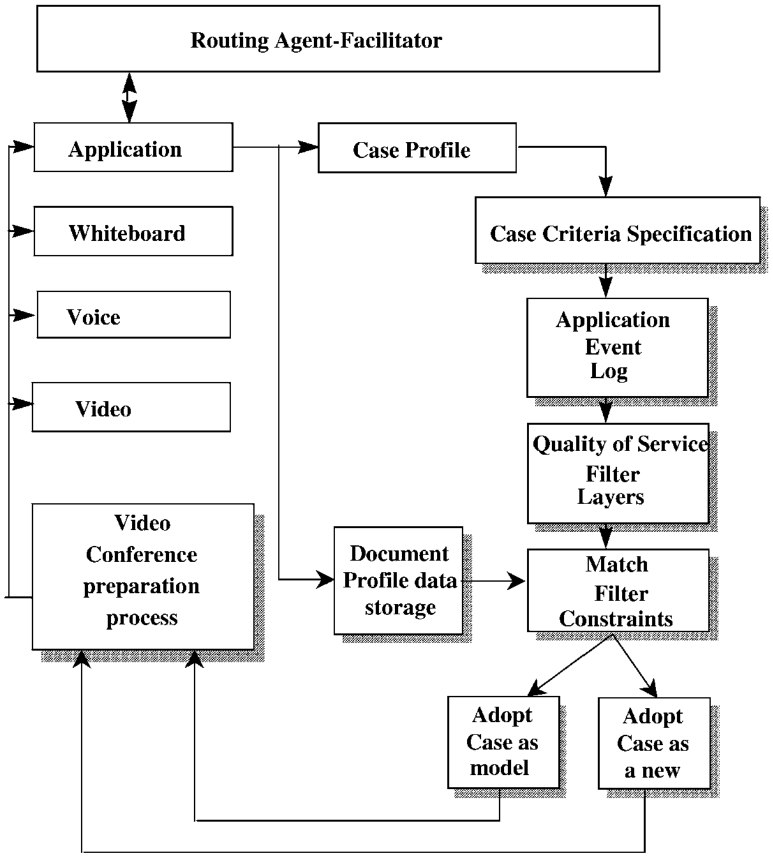

Figure 14. Data model for agent-case memory integration.

criminant functions (6). If, for all nodes, the value of $\boldsymbol{W}_i \cdot \Delta \boldsymbol{P}_i$ is positive, the agent-facilitator transfers control to the RTP bandwidth adaptation tool for providing immediate bandwidth adjustments. If some nodes vote "yes" to bandwidth adjustment, and the others vote "no", then the second layer committee nodes that indicate associations with the current multimedia call profile are checked. If the committee node votes "yes", then RTP bandwidth adaptation is turned "on".

6. Conclusion

The proliferation of wireless IP multipoint teleconferencing applications is being driven by new broadband wireless technologies such as LMDS. Often these applications are run on heterogeneous networks such as wired ATM to satellite connections. In this situation, the characteristics of each type of network have the potential to affect data transmission properties and alter receiver QoS.

In this paper, we have presented a control model which provides response time and bandwidth requirement adaptation of audio, video, and application sharing multipoint IP teleconferences for emerging wireless multimedia communications. Our study is innovative in that it integrates feedback controls between the application and network layers. The proposed model collects feedback for driving multimedia call preparation and call monitoring. Case-based reasoning memory is incorporated in order to create user profiles for real-time connection control. Network agents are employed to capture user and multimedia teleconference application call profiles and transfer them to the case memory. At the network layer, RTP statistics are used to evaluate and manage connection QoS. A receiver-based hierarchical audio encoding scheme was also introduced to provide course-grained network adaptation.

Adaptive capabilities of the proposed agent-memory architecture were tested through practice of functions such as discovery of pertinent collaborators, retrieval of information relevant to the collaboration, and creation of teleconferences among individuals with different user profiles. Our proof-of-concept experiments demonstrated that agents-facilitators may compensate for lack of feedback and provide a means for adaptive management of IP multipoint teleconferencing. Participants of multipoint trials obtained a reduction of transaction time, a reduction in task processing time, an increase in task concurrency, and an increase in complementary knowledge (learning). In conclusion, our model provides the necessary adaptive elements which tie together user preferences, application performance, and network performance and lead to improved QoS for wireless IP multipoint teleconferences

The next step in our research is to explore the adaptation of QoS constraints for multimedia streams and timely events that occur over heterogeneous, multipoint ISDN, ATM, and IP bandwidth-on-demand, satellite-terrestrial communications. Our testbed is the experimental configuration of the National Transparent Optical Network, the Advanced Communication Technology Satellite, and the Internet 2/vBNS networking segments. Work is currently underway to evaluate our model's performance in this environment.

References

[1] S. Aidarous and T. Plevyak, *Telecommunication Network Management into the 21st Century* (IEEE Press, New York, 1995).
[2] J. Bolot and T. Turletti, Experience with control mechanisms for packet video in the Internet, ACM Computer Communication Review (January 1998).
[3] A. Bordetsky, Reasoning on infeasibility in distributed collaborative computing environment, Annals of Mathematics and Artificial Intelligence 17 (1996) 155–176.
[4] A. Bordetsky and E. Bourakov, Agents-facilitators for adaptive management of collaborative environments, in: *Proceedings of the 3rd INFORMS Conference on Information Systems and Technology*, Montreal (26–28 April 1998) pp. 82–96.
[5] I. Busse, B. Deffner and H. Schulzrinne, Dynamic QoS control of multimedia applications based on RTP, in: *International Workshop on High-Speed Networks and Open Distributed Platforms* (1995).
[6] M.R. Genesereth and S. Ketchpel, Software agents, Communications of the ACM 37(7) (1994).
[7] V. Hardman, M. Sasse and I. Kouvelas, Successful multiparty audio communication over the Internet, Communications of ACM 41(5) (May 1998).
[8] D. Hoffman and M. Speer, Hierarchical video distribution over Internet-style networks, in: *Proceedings of the IEEE International Conference on Image Processing*, Luasanne, Switzerland (September 1996).
[9] I. Kouvelas, V. Hardman and J. Crowcroft, Network adaptive continuous-media applications through self organized transcoding, in: *Proceedings of Network and Operating Systems Support for Digital Audio and Video (NOSSDAV '98)*, Cambridge, UK (8–10 July 1998).
[10] L. Lewis, *Managing Computer Networks: A Case-Based Reasoning Approach* (Artech House, Norwood, MA, 1995).
[11] T. Malone and K. Crowston, The interdisciplinary study of coordination, ACM Computing Surveys 6(1) (1994) 87–119.
[12] G. Marakas, *Decision Support Systems in the Twenty-First Century* (Prentice-Hall, New York, 1998).
[13] J. Mayfield, Y. Labrou and T. Finin, Desiderata for agent communication languages, in: *Proceedings of the AAAI Symposium on Information Gathering from Heterogeneous, Distributed Environments, AAAI-95 Spring Symposium*, Stanford University, Stanford, CA (27–29 March 1995).
[14] S. McCanne, V. Jacobson and V. Vetterli, Receiver-driven layered multicast, in: *Proceedings of ACM SIGCOMM*, Stanford, CA (August 1996).
[15] C. Parris, G. Ventre and H. Zhang, Graceful adaptation of guaranteed performance service connections, in: *Proceedings of IEEE GLOBECOM '93*, Houston, TX (November 1993).
[16] H. Schulzrinne, S. Casner, R. Frederick and V. Jacobsen, RTP: a transport protocol for real-time applications, RFC 1889 (January 1996).