Chapter 4

Multiparty Audioconferencing on Wireless Networks

R. Venkatesha Prasad, Vijay S. Rao, H.N. Shankar, and R. Muralishankar

Contents

Internet has been a part and parcel of life for sometime now. With the advent of technology, wireless networking is surging ahead. Multimedia applications are also finding their rightful place in this revolution. This chapter discusses an important multimedia application—audioconferencing. Issues related to media and the wireless networks have been addressed and some existing techniques for conferencing are presented. Learning from their limitations, we formulate the requirements for a better quality conference. We set out with refining the existing Session Initiation Protocol (SIP) architecture by introducing conference servers (CSs). We present a recently proposed metric called "Loudness Number (LN)" that helps in dynamically selecting speakers in a conference. This is aimed at facilitating a smooth transition of speakers while taking turns. Specific to the context of wireless networks, we argue that by adopting the refinements proposed here, some limitations in conferencing are mitigated. More efficient use of the existing bandwidth and reduced computational effort are highlighted, and the chapter concludes with some open issues and possible vistas for investigations.

4.1 Introduction

Concomitant with the Internet maturity and advances in supporting firmware, novel applications are becoming commercially viable at a rapid pace. Multimedia conferencing over Internet Protocol (IP) networks is an

application of growing popularity worldwide. It enables many participants to exchange different types of media streams for collaborative work. The idea that collaborative technology is an activity in search of a need should be laid to rest [1]. Collaborative technology carves out a niche among users who spend most of their time in group endeavors, who use computing instruments to do their work, and whose potential for collaborations is otherwise impaired by geographic separation. The trend among conferencing systems has been to draw on real-world interaction protocols, but not necessarily on ensuring an honest-to-goodness virtual replica of a face-to-face conference (except in Refs 2–7).

The key desirable features of any collaborative technology are (a) less cognitive overload for the participants, (b) simplicity, (c) real-time media, and (d) good speech quality [1]. Collaborative work demands close interactions among participants, feedback or corrections for speakers, and mixing of voice streams—all in real-time.

Hindmarsh et al. [8] show that the participants fall back on audio to resolve difficulties with other aspects of collaborative applications such as negotiating a shared perspective. In fact, Doerry [9] demonstrates very marginal improvements in collaborativeness when video is included. Hence, audio as voice is the most important component of any real-time collaborative application. This chapter focuses on the audio medium of a collaborative work platform, that is, on "audioconferencing."

Miniature computing and communication devices, such as motes, PDAs, and mobiles, are increasingly enabling seamless and ubiquitous services across platforms. Exponentially growing volumes, time-to-market, and time-in-market are inevitable concerns placing consumer electronics at its bleeding edge incessantly. Bandwidth bottlenecks are becoming less stringent in terms of their impact, thanks also to the emerging trends in wireless technologies. Users are demanding progressively rich media content in real-time even in handheld, mobile, low-power, and low-cost devices—all in one breath. Multiparty audioconferencing on wireless networks is one such popular demand. Even this application domain has several dimensions to it.

In particular, this trend is accelerating even as it is maturing in broadband wireless networks and *ad hoc* networks. We observe that 802.11a/b/g has changed the way we connect to the external world. In fact, to enable higher bandwidth in the last mile, 5 GHz band at 60 GHz [10] has been reserved to enable, *inter alia*, rich multimedia. The Institute of Electrical and Electronics Engineers (IEEE) 802.16 Working Group on Broadband Wireless Access Standards is working on the deployment of broadband Wireless Metropolitan Area Networks (WiMAX) [11] more pervasively. Devices with IEEE 802.16a/e are already penetrating the market and the IEEE 802.20 version of the mobile broadband wireless networking is in the pipeline [12]. Consequently, from the consumers' perspective, enhanced multimedia applications are rendered easier by the day. In the process of

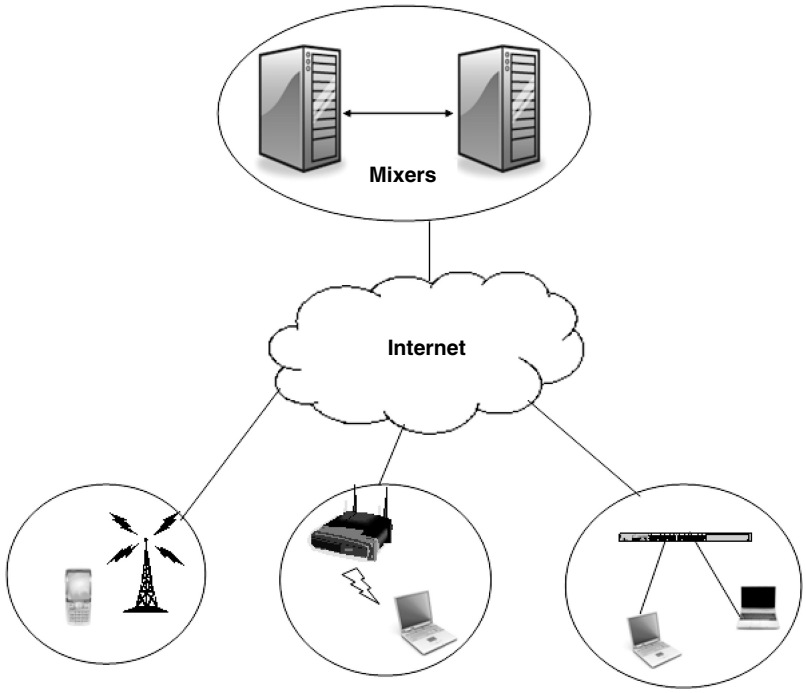

Figure 4.1 Typical scenario of wireless nodes in a session.

launching such applications in the market, however, there are concerns to be addressed. We look at some of them here.

1. Any service provider has to be acutely sensitized to the fact that devices in demand in an aggressively competitive market are functionally heterogeneous (see Figure 4.1). This is a fallout of constraints of computational capabilities, onboard memory, and link capacity having to coexist with mobility. It is strongly desired that such devices have multiple link layers. Then handling multiple linked layers pops up as a problem in its own right. Some devices running interactive multimedia applications, for example, speech conferencing, may not be self-supporting, let alone supporting their peers. This places severe constraints when graduating from devices on wired networks to those on wireless networks.
2. In wired devices characterized by single point of contact, addressing and accessing a device are simple tasks. Such is not the case in the wireless world (except in the case of broadcasting or nonspecific multicasting), wherein mobility across different radios is a key functionality.

3. Wireless devices pose problems as the connectivity that is required to be continuous is affected by many factors. The fact that the dynamics of wireless networks is inherently more random *vis-à-vis* the dynamics of wired networks, not only causes the preceding problems but also goes on to add a further dimension to these problems.
4. Ensuring information security and service reliability in wireless services poses problems that are much more challenging than wired services.
5. Finally, robustness is a tall order in view of the preceding four issues raised.

Design of audio for real-time collaborative activity is constrained by a large number of participants having to share limited bandwidth. Audioconferencing is sensitive to packet loss, latency and delay jitter, and to gaps in the audio caused by the diminished real-time support on host devices as these affect intelligibility.

Section 4.2 outlines the desirable features of an interactive conferencing system and Section 4.3 describes the constraints in designing the conferencing system on wireless networks. Among these constraints, we delve more on mixing architectures and less on session handling. After providing an insight into the existing architectures of mixing, we present an algorithm using the LN and study its effect on the conferencing system. We conclude the chapter with open issues that need further research.

4.2 Desirable Features of an Interactive Audioconferencing System

An interactive audioconferencing system is desired to be characterized by the following features:

i. *Interactivity*. One of the main features in an interactive conference is to allow multiple speakers at any time without any control on "who can speak," "what can be spoken," and "who can hear what." For instance, gaming and other virtual reality services [13] may allow audio from every participant. Supporting most natural audio communication between participants allows spontaneity in the way people interact using the audio medium. A case in point is a face-to-face conference wherein all sounds from participants are heard at all times. The fact that conferencing is as much about human factors as it is about the underlying technology can never be overemphasized. It is crucial to devise collaborative systems with an understanding of how sociological and psychological factors impact group work,

especially because mismatched expectations in group-based systems have resulted in serious groupware failures [14].

ii. *Spatialization*. Mixing of audio streams refers to the weighted addition of several input streams into one output stream. Mixing makes a session interactive and reduces the number of streams. Mixing is crucial to devices that are starved for resources. To customize mixing is to choose the individual weights suitably at different terminals. Spatialization refers to the experience of the listener of the customized mixed stream. Hendrix and Barfield [15] consider the presence and absence of spatialized sound and addition of spatialized and nonspatialized sounds to a stereoscopic display. It is reported that the addition of spatialized sound significantly increases one's sense of presence in the virtual environment.

iii. *Signaling*. Signaling involves the exchange of control packets to establish, maintain, alter, and terminate an audioconference. Signaling must contribute to a reliable, robust, and scalable conferencing system. Signaling is also required to keep the terminals updated of the network status. This in turn helps in choosing across codecs, setting parameters such as sampling time of audio packets, and initiate packet recovery algorithms and time-scaling algorithms.

iv. *Scalability*. Collaborative work demands are rapidly growing across the globe. A scalable conferencing system should adapt to different types of network connections and terminals besides supporting a large number of participants. Expectedly, scalability is much more crucial than ever before.

v. *Availability of multicasting*. In multicasting, a number of processes join to form a group; all processes in the group receive all the data dispatched to that multicast group. This reduces the end-to-end delivery time and network traffic. Multicasting in wireless networks is difficult compared to its wired counterpart. Even majority of the routers in the Internet are not multicast-enabled. A common recipe operating on a dynamically evolving infrastructure must context-sensitively leverage multicasting, if and where available.

vi. *Traffic reduction*. To reduce traffic is to reduce the number of audio streams in the network for a given number of participants. Reducing traffic at the application layer contributes to a more efficient use of network resources. Traffic may be reduced at the network layer by using multicast techniques. A few other techniques are listed as follows:

   a. *Compressing audio streams*. Some compression techniques such as ITU-T G.729, G.726, and G.723 and iLBC coding [16–18] result in low bit rates up to 6 kbps per participant.

   b. *Reducing traffic using voice activity detection (VAD)*. Speech inherently has several pauses. Dropping the silence segments results

in a reduction in the bandwidth requirement [19,20] of up to 50 percent for a single stream.

c. *Reducing the number of streams by mixing necessary streams.* Some of the audio streams may not be relevant in a conference. The conferencing system must be sensitive to the changes in the participants' speech and intelligently select relevant streams to mix.

vii. *Quality of the conference.* The "interactivity" in an audioconference is achieved when the mouth-to-ear delay is below 150 ms (in fact, it can be even up to 600 ms) with permissible packet loss and jitter. The three factors—delay, packet loss, and jitter—influence the quality of the speech and user satisfaction [21]; they should be kept as low as possible.

Watson and Sasse [22] have shown that speech quality depends not only on the percentage loss of the transmitted packets but also on volume settings, microphone settings, and echo. Context sensitive and dynamic customized mixing of audio is one of the solutions to achieve good spatialization through volume and microphone settings. It enables participants to emphasize or deemphasize specific audio streams as desired.

4.3 Constraints of Interactive Audioconferencing in Wireless Networks

Some of the problems of desirable features are listed here and an overview of the existing implementations that aim to mitigate them is presented. While designing a conferencing system in the context of wireless networks, one should keep in mind that

i. A client or terminal* may not have the capability to decode the multiple streams and mix them.
ii. Clients may join or leave the conference because of ephemeral network connectivity, which results in frequent change in the point of contact due to mobility.
iii. The clients experience higher packet loss and, thus, require higher error correction or redundancy.

The implications of these must be handled judiciously.

* We use the terms client, terminal, and participant interchangeably.

The issues in supporting conferencing can be divided into media- and network-related.

A. Media-related issues
   - Quality of the conference is based on interactivity, that is, participants must be able to freely acquire an opportunity to speak to other participants.
   - Mixing and customized mixing of speech for the participants.
   - Signaling.

B. Network-related issues
   - Large-scale distribution of participants over a wide area (scalability)
   - Enabling multicasting when the necessary infrastructure is available
   - Traffic reduction
   - Quality of the conference dependent on packet delay, packet loss, and delay jitter

These issues are explained briefly in the following sections.

4.3.1 Interactivity

"Floor" is a virtual entity that allows a participant to speak. With some form of floor controller [23–26] explicitly telling the participants when to speak and also reserving or waiting for one's turn to speak would evidently result in a "gagging" feeling for participants [2,3,27]. This means that the system's view of "speaking" is not the same as that of the participants. In addition, it even hinders the natural urge to interrupt. This makes a static and explicit floor control strategy [23,24,28] inappropriate in supporting most natural audio communication. Mixing speech from all participants as soon as they occur facilitates communicating the concerns of listeners to the current speaker, which is equivalent to allocate floor to every participant. Without this capability, collaborative platform would be imposing an unnatural curb on the participants. Thus, the requirement of impromptu speech without explicit permission from any controller (deciding who should speak and when) is necessary to mimic a real-life conference. Mixing all streams is a solution. However, it may not be necessary or, for that matter, even desirable, to transmit all speech streams. Floors that are allocated should be relinquished promptly by the participants. Floor control for clients connected through wireless links is difficult to achieve because the point of contact is ephemeral. The floor control as a technique to handle multiple streams is discussed in Section 4.4.

4.3.2 Customized Mixing

Customized mixing is one of the desirable features discussed in Section 4.2. Discussions on mixing *vis-à-vis* Real-Time Transport Protocol (RTP) can be sought in Ref. 29, and some interesting discussion on this with respect to wireless networking is in Ref. 30. Customized mixing at the clients is usually not possible due to the limited capability of wireless devices. Higher bandwidth requirement for transmitting individual streams to enable customized mixing is another constraint here. With the advent of 60 GHz indoor networking, there is a possibility of supporting this in the near future.

4.3.3 Signaling

Wireless and *ad hoc* networks throw many challenges. The centralized entities are to be minimal because the connectivity to a centralized entity is not guaranteed. Further, the dynamic nature of the nodes introduces further constraints. However, for the sake of out-band signaling in wireless broadband networks, we may assume the existence of infrastructure that can support authorization and provide some initial support.

There are two major signaling protocols for IP multimedia conferencing—ITU-T H.323 [31] and Internet Engineering Task Force's (IETF's) SIP [32]. H.323 does not scale for large number of conferences, although the latest version has some improvements. However, the interest in the industry and academia has now been with the SIP and it has established itself as the major signaling standard. Although SIP is very useful and simple, it has not been able to address multiparty conferencing comprehensively. Some variants of SIP, a draft proposal by Mark/Kelley [33], have been useful to a limited extent. Core SIP [32] offers much less for peer-to-peer (P2P) conferencing solutions. Thus, the protocol or any multiparty conferencing solution should workaround these deficiencies. For detailed discussions, see Refs 32–35. We have proposed here just a plausible solution to the signaling problem without going into its in-depth analysis.

4.3.4 Scalability: Large-Scale Distribution of Participants over a Wide Area

Schooler [36] identifies many issues with respect to scalability, *viz.*, (1) session models and its protocols, (2) multicast address management, (3) techniques for bandwidth reduction, and (4) codification of heterogeneity based on voice quality. Handley et al. [37] highlight session scaling based on one or more of the preceding aspects including security and authentication besides network support and reservations.

The key to managing heterogeneity over unmanaged, large-scale networks lies in providing distributed, rather than centralized, solutions.

Centralized solutions are easy to build and maintain [38]. They are woefully inadequate when the participants are geographically far apart because of poor response for time-sensitive operations. They impose heavier traffic at the servers. Distributed solutions that deal with local traffic are suited to large-scale setting of the Internet but are difficult to implement and manage.

4.3.5 Availability of Multicasting

One of the main advantages of packet networks over circuit-switched networks is the ability of the former to support multicasting/broadcasting. With wireless systems, broadcasting in a smaller domain is relatively easy because all the single hop nodes (devices) can listen to the radio frequency (RF) signal. A single multimedia stream can be distributed to a large number of subscribers. The impact of multicasting when the enabling infrastructure—such as wireless broadband connectivity between the multiple participants on networks—is available can never be underestimated. There are no fixed routers on exclusively *ad hoc* wireless networks. Thus, as the end-to-end routes are nonstatic, implementing multicasting is nontrivial.

Deering's [39] monumental work on multicasting offers an efficient multipoint delivery mechanism. A single packet can be sent to an arbitrary number of receivers by replicating packets only at the forks of the network. Transmissions from one to many are accomplished without packet duplication using a common group address. Multicast is based on the formation of groups whereby a number of processes may join a multicast group to receive all the data dispatched to that group. Wireless network must evolve to support IP multicast. This depends on the device capability and implementation. In fact, availability of multicasting in the Internet itself cannot be taken for granted. Thus, if and wherever available, the conferencing architecture must make use of multicasting. However, even if not available, the architecture should support conferences. The solution for operating on a dynamically evolving infrastructure is to context-sensitively leverage multicasting, even as it must not be contingent on its availability. In a wireless domain, many routing protocols do not automatically support multicasting. Multicasting support needs to be explicitly implemented so that one can utilize it. For more details on multicasting in wireless network, see Ref. 40.

4.3.6 Traffic Reduction

Reduction in traffic can be achieved as discussed in Section 4.2. Audio streams consume more bandwidth, which can be reduced by mixing. Depending on the nature of the processor and its speed, wireless devices may implement some or all of the techniques described earlier. One of the challenges is to implement high compression codecs on these wireless

and handheld devices with limited capabilities. We revisit this aspect in our discussions on architecture as well as on mixers.

4.3.7 Quality of the Conference Based on Packet Delay, Packet Loss Percentage, and Delay Jitter

Delay of audio packets is due to transmission and queuing at the routers or intermediate nodes in the network. Variable queue sizes seen by audio packets at intermediate routers introduce delay jitter. In a packet speech system, the end-to-end delay (after nullifying the jitter) is always a critical parameter in a real-time voice system. It should, hence, be kept well below 600 ms [41] in the absence of echoes, if conversation patterns are not to break down. Packets are dropped at congested routers whenever the queue overflows. Transportation error is higher in wireless networks compared to their wired counterparts. The extent of packet loss is a primary factor determining whether a network audio stream will be intelligible to the user, and therefore, of any use at all. Delay and jitter play a secondary role and should also be kept under check.

The present-day networks, wired or wireless, do not support these service models. Real-time traffic has to compete for bandwidth with other non-real-time traffic on the best effort network such as IP. In such networks, there can be no assurance about the quality of service when physical limitations are present; packet loss can be kept under tolerable limits only through adaptation. Adaptation to delay jitter and loss is reported in Ref. 42.

Today, as the conference involves a large number of participants, there is a need to efficiently handle the traffic as well as enable free access to the shared resources for the participants. The desirable features also need to be addressed while efficiently handling the multiple streams. The different techniques are discussed in detail in Section 4.4, hinting at the possible architectures for conferencing.

4.4 Techniques for Handling Multiple Audio Streams

In a conference, speech streams of all participants are often not concurrently permitted. If allowed indiscriminately, quality of play out usually declines. This is because there is an upper bound on the number of distinct streams that a given network can handle. In the context of multiparty audioconferencing on wireless networks, the severity of this bottleneck is pronounced. Independently, human comprehension with multiple streams becomes poor due to sharply declining resolution. Some attempts to address such issues are reported in the sequel.

a. *Floor*. Floor control, or turn-taking mechanism, provides a means to mediate access to shared work items. Greenberg [43] recommends

that systems should "support a broad range of floor control policies" to suit the needs of the participants. Floor control can be important in many situations, such as shared screens allowing only serial interaction, or systems following strict interaction models similar to a teacher monitoring/controlling the work of students. Roseman and Greenberg tackle many of these aspects on GROUPKIT building. For collaborative environments, several types of floor control policies such as explicit release, free floor, round robin, first-in-first-out, preemptive, and central moderator [43] are available. However, these methods are beset with difficulties in supporting impromptu communication. In a setup with floor control, each subsystem must decide the level of support to simultaneity (i.e., number of active participants at any time) and the granularity to enforce access control. In its simplest form, floor control enables floor access to only one participant at any given time and the floor is handed over when a request is incident. In the case of audio, floor control introduces a management framework around the audio session that enforces turn taking, thereby removing any potential simultaneity. Consequently, the ambience of the application becomes suffocating or gagging for the users. This can happen even if there are more floors because the person who wants to speak may not have a floor. Although explicit floor control may be suitable for some applications such as a broadcast by a panel, it is inherently difficult to implement and maintain for a system with many-to-many active participants. When large-scale groups are to be allowed, implementation of these techniques is cumbersome. Making a policy for floor allocation without human intervention is not simple in a large conference where all members can request and be granted the floor.

b. *Push-to-talk*. To talk, the pointer has to be placed in the window and the mouse button clicked (like a walkie-talkie). This mechanism is known as "push-to-talk" [44]. Only after this explicit action, the participant is allowed to speak to others. This is the default approach used by many of the MBone-based tools such as "vat" and remote access Trojan/tools (RAT). These tools enable every participant to hear everybody else in the conference simultaneously. This allows the users to make very careful choices whether to speak or not, avoiding significant amounts of simultaneous speaking in many more restrained contexts. Nonetheless, it is liable to remove subtle nonverbal cues and sounds. Further, conversation and interaction becomes slower and unnatural due to a conscious turn-taking activity that is required to be heard. This reduces the spontaneity and interactivity of the conversations. When many participants speak simultaneously, the network gets flooded suddenly, thereby causing

disruptions due to packet loss. Silence suppression is another form of handling simultaneous speakers.

c. *Audio processing unit (APU).* Yu et al. [45] propose an APU for conferencing. It is a hardware implementation of multipoint control unit (MCU) of H.323. MCU is used for multipoint audioconferencing–videoconferencing systems. This is a centralized unit, which takes a fixed number of audio sources from participants and distributes to each of them a specific mix of selected streams excluding their own to avoid echo. It, therefore, performs total mixing on a dynamic set of input streams. The APU selects four simultaneous speakers on the basis of individual sample energy. Once selected, a participant holds the floor for a certain minimum interval to circumvent frequent changes, thus hindering interactivity. It is implemented using a DSP chip. Its scalability is very limited.

Different techniques allow natural audio communication between participants to different degrees. More specifically, these techniques have different ways of determining as to which speakers are heard in the conference. All the techniques described earlier limit the number of simultaneous audio streams transmitted in the network.

4.5 Mixing Architectures: State-of-the-Art

As mentioned earlier, mixing is an inevitable process in audioconferencing and can reduce the total bandwidth. As a major task, it has attracted different architectures to support audioconferencing. A few important ones are discussed in the following sections.

4.5.1 Centralized Mixing Architecture

In the simplest centralized mixing approach [46], a single mixing server handles all the streams from the participants of the conference (Figure 4.2). Each participant sends his media stream to the mixer. The mixer then adds all the streams and sends it back to the participants after subtracting their own streams to avoid far end echo. If there are some participants who are listen-only, they would be getting the complete mix. When end terminals are not capable of handling many streams, introducing a server for mixing reduces the load on the end terminals as also the number of audio streams in the network.

The advantages. Reduced media streams in the network, at least on the client/participant side only one stream is sent and received. The

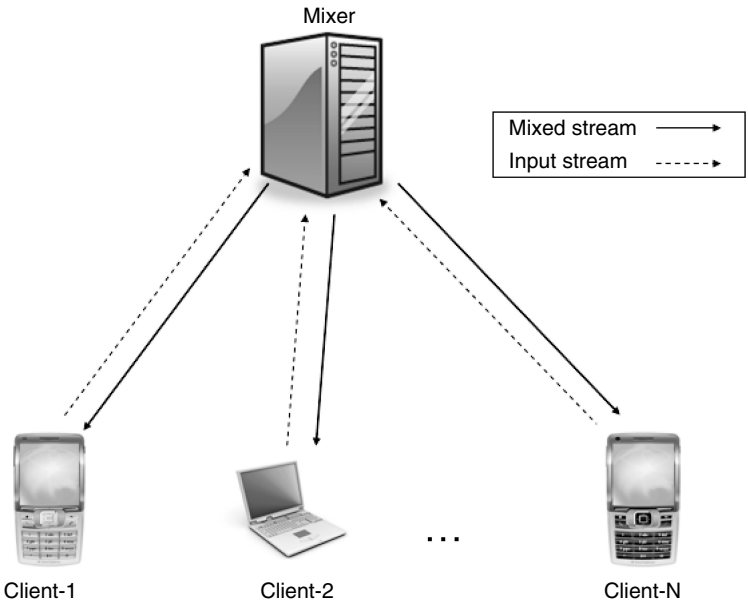

Figure 4.2 Centralized audio mixing.

latency is lesser. Because all the streams meet at a point, synchronization is inherently assured.

The disadvantages. Centralization makes it unsuitable for conferencing in *ad hoc* networks. For *ad hoc* wireless networks it is not good enough because the basic assumption of the availability of a server is not valid. Centralized mixer presents a potential bottleneck in the network. Resources are not used optimally because all the computational and network load is dumped on the server. Network growth depends on the mixer's capabilities. Hence, this architecture is not scalable even with the availability of infrastructure connectivity, although it is implementable in any network.

4.5.2 Endpoint Mixing Architecture

This is a mesh architecture where all the media streams are seen by all the participants. It is P2P and easy to implement [47] (Figure 4.3). Packets are mutually exchanged between all clients. Mixing is done by every client, thus allowing personalized mixing for participants. This can be tailored to the individual's equipment, spatialized according to specific location within a shared virtual space, and under individual control. The number of packets in the network would be enormous when there are too many participants without any control. Even with multicast, it is very demanding on network

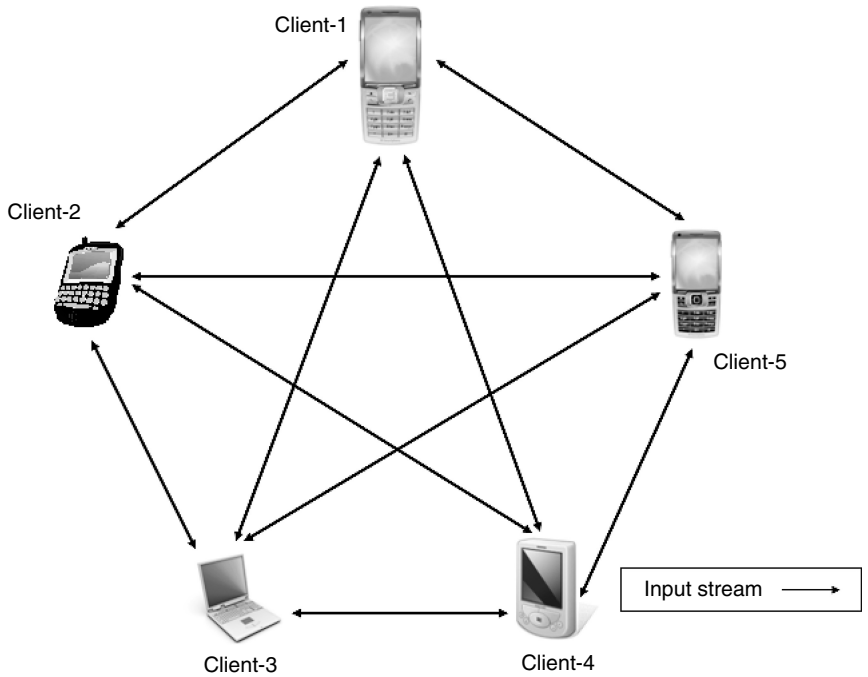

Figure 4.3 Endpoint audio mixing.

resources, particularly bandwidth, apart from computational resources at the end terminals.

The advantages. The architecture is decentralized. Streams are delivered rapidly as there are no intermediate mixers in the end-to-end paths. This also makes synchronization easy. Self-organization is enabled because each participant can mix at its premises and it is faster as there are no intermediate servers.

The disadvantages. Because mixing is done at each client, the duplication of efforts does not lead this architecture to be scalable. Nodes with limited capabilities will not be able to join this sort of conference. Central processing units (CPUs), I/O devices, storage capabilities, codec (dedicated hardware and software), communication protocol, and network interfaces of the participants place limits on the end-system capabilities to process, consume, and generate multimedia data. For example, limited resources in terminals can result in buffer overflow, delay in processing data, and inability to process data. These manifest to the user as unacceptable play out delays, lost audio segments, and poor user interaction. The number of media streams in the network is large,

thus overloading the whole network. All the nodes are stressed to the same extent blindly. In fact, the heterogeneous nature of the wireless devices does not make this architecture useful. This architecture may not work for an *ad hoc* network of end terminals. Thus no assumptions regarding the capability of end terminals are admissible except a few minimal requirements.

4.5.3 Hierarchical Mixing

To support applications involving a large number of simultaneously active audio streams, Rangan et al. [48–50] propose a hierarchical mixing architecture. In such applications, all streams are mixed. They compare a distributed hierarchical mixing architecture with centralized mixing. They exploit the comparative durations required for transporting a packet from an end terminal to a mixer and for mixing two streams at a mixer. Depending on the ratio of these two times, they propose different architectures for media mixing. In this mixing hierarchy, participants constitute leaf nodes and the mixers—nonleaf nodes (Figure 4.4). The mixer at the root of the hierarchy forwards the final mixed packet to each of the leaf nodes. This can also be

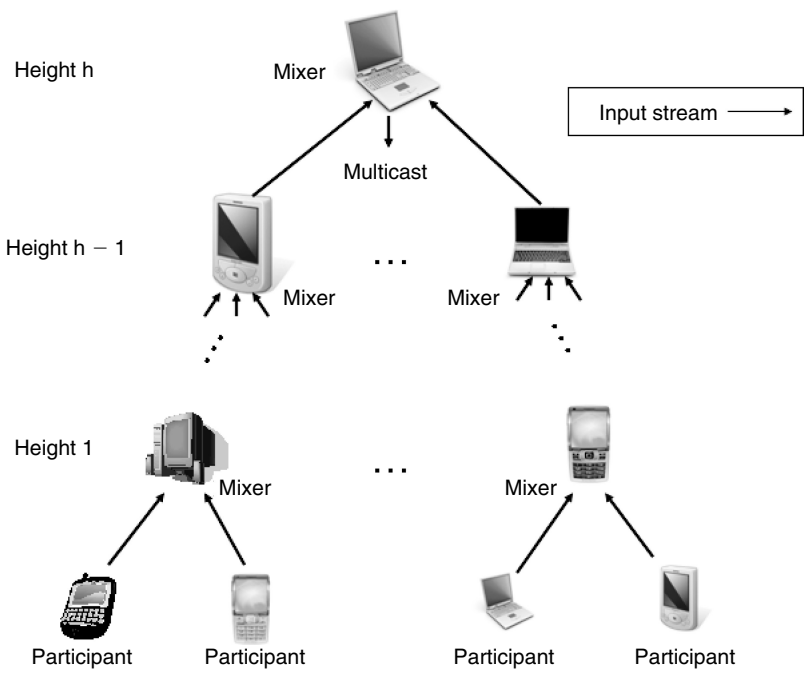

Figure 4.4 Hierarchical audio mixing.

done by multicasting the packet from the root, if supported, to avoid delay during hops. Alternatively, packets can be forwarded through intermediate mixers (application-level multicasting [ALM] in some respect).

The advantages. This tree-based architecture is highly scalable compared to a centralized mixing and completely distributed endpoint mixing. The authors show that it is an order of magnitude more scalable than purely centralized or distributed architectures. The root node mixes lesser number of streams as the intermediate nodes mix streams from their children nodes.

The disadvantages. This approach does not permit the specialization because all streams are mixed at all times as in centralized mixing. The number of hops increases with the height of the tree, albeit slowly. Thus, it is not well suited for interactive and collaborative applications. Mixers are preconfigured and static, and do not adapt to the network conditions. The root node is analogous to a centralized mixer as the final mixing before distribution takes place at the root. If the participants are segregated, then synchronizing their streams, not withstanding the packet loss and jitter, is difficult.

4.5.4 Distributed Partial Mixing

Radenkovic et al. [2,4,27] identify the importance of permitting speech streams from many participants. They consider situations where a large group of people "speak" at the same time. Each participant independently introduces a new audio stream that has to be accommodated by the network and has to be processed by each recipient. Multicasting if used reduces the bandwidth required; however, it puts more pressure on the end terminals in terms of bandwidth and computation. Thus, distributed partial mixing (DPM) reduces traffic on shared links by mixing some streams. An example is shown in Figure 4.5.

Partial mixers (PM) may be dedicated servers or additional roles played by other end terminals. End terminals typically communicate through these PM components, sending their audio to and receiving other participants' audio from a nearby PM. PMs in turn form a completely connected network (e.g., a tree) to achieve end-to-end distribution. PM extends the traditional concept of mixing to render it more dynamic and flexible. Unlike total mixing [2,27], where the entire set of received streams is mixed into a single output stream, partial mixing dynamically chooses only a subset of the available audio streams to mix at a time and forwards it with other unmixed streams. The number of streams mixed in Ref. 29 varies dynamically depending on the number of active participants. Selection of the number of streams for mixing is based on the network conditions. Thus, instead of producing a single output stream in all cases, partial mixing produces

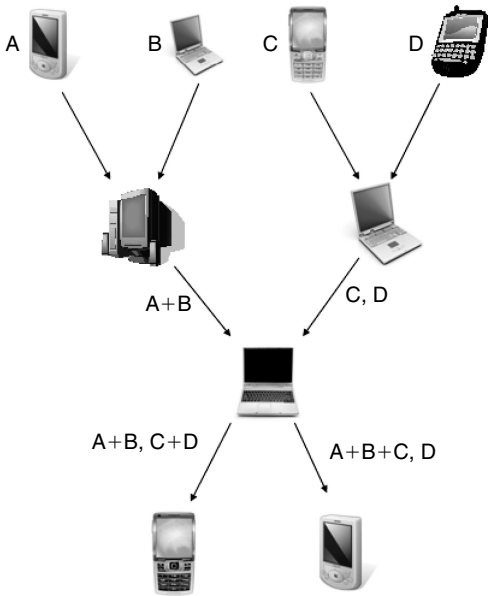

Figure 4.5 Distributed partial mixing.

various number of streams in different situations. Hence, the number of streams in the network would be at least that of hierarchical mixing.

The advantages. It is decentralized and highly scalable. It tries to adapt to the network conditions.

The disadvantages. It introduces many mixers, considerably increasing latency. It leads to fluctuations in the volume of every individual participant causing severe degradation in quality. Customized mixing of streams is not possible when many clients are active. Synchronization is difficult to achieve.

4.5.5 Distributed Mixing System

Khedher et al. [30] propose a modified hierarchical mixing architecture wherein the number of levels is limited to two. Nodes in the first level are mixer nodes (representing cluster heads) and exchange streams among them using multicasting. The other participants in the cluster register to these mixers to avail the facility of mixed streams. Each mixer receives mixed streams from the other mixer nodes in the overlay. It mixes these streams with those received from second-level nodes registered to it. This stream is transmitted to the second-level nodes and to other mixers. In essence, streams from all the nodes are mixed before play out. The idea of

electing mixer nodes dynamically based on the availability of resources in the nodes is also proposed. Authors extend the clustering concept of the wireless *ad hoc* networks to form clusters and elect at least one mixer in a cluster.

The advantages. The architecture is a complete-mesh topology with only two levels of mixing; thus the delay is kept under check. Bandwidth reduction is possible because all the streams are mixed. Synchronization can be achieved by adapting the play out so that streams originating at the same time are played out at the same time. The distributed mixing architecture significantly reduces the number of media streams in the network.

The disadvantages. Users lose spatialism. Limitations of mixing many streams apply here. The limitations of mixing streams are discussed in the following section.

4.5.5.1 Advantages and Disadvantages of Mixing

The pros and cons of mixing architectures are summarized.

4.5.5.1.1 Benefits of Any Mixing-Based Approaches

1. Mixing can drastically reduce network traffic for many simultaneous speakers. This makes it efficient with its support for efficient distribution of audio streams in the network. See Table 4.1, where mixing at an intermediate server can reduce the number of streams drastically.
2. Mixing imposes no constraints on individual speakers unlike gagging as in floor control.
3. The system's view of "speaking" is the same as that of the user; naturally, mixing is more effective for supporting the most natural audio communication.

Table 4.1 Order of Traffic in Different Conference Architectures (with and without Multicasting)

Endpoint	*Centralized*	*Hierarchical*	*Cluster (c) Based [30]*	*Domains (d) with* N_{max}
Unicast Communication				
$M(M-1)$	$M-M$	$2(M-1)$	$c(c-1)$	$N_{max}d(d-1)$
Multicast Communication				
M	$M+1$	M	c	$N_{max}d$

Note: For cluster-based and our proposal of domain-based scenario, we have not considered the streams exchanged within the cluster/domain because we think it is completely localized to a cluster.

4.5.5.1.2 Limitations of Mixing

Mixing of audio streams digitally involves summing up all streams at each sampling instant with some weight for each stream. The weights sum up to unity to avoid signal saturation. Mixing has the following problems [2,27]:

1. The mixed signal has a lower signal-to-noise ratio (SNR) and a lower dynamic range than individual streams.
2. Fidelity is poor as it includes noise from all the separate streams in fewer bits.
3. Mixing may not work well with some ultralow bandwidth codecs due to their compression algorithms.
4. Mixing in stages introduces additional delay as audio streams have to be processed by the mixers.
5. Spatialization and other aspects of control of individual listener over each audio stream are lost.
6. While mixing digitally, the volume level of individual streams is reduced to avoid saturation.
7. Mixing is irreversible and the individual streams cannot be recovered at a later stage.
8. Mixing introduces more components (servers) and increases the complexity in the network with the corresponding hardware and real-time control requirements.

These limitations expose the need for a new scalable architecture either with no mixers or with less intermediate mixers so as to keep a check on the bandwidth usage. Of course, user experience is the most important aspect of the architecture.

4.6 The Proposed Architecture

We propose an architecture drawing lessons from the detailed study done hitherto.

4.6.1 Design Requirements

- *Audio mixing.* In an audioconference, streams from all the clients need not be mixed. Actually, mixing many arbitrary streams [29] from clients degrades the quality of the conference due to the reduction in the volume (spatial aspect of speech). There is a threshold on the number of simultaneous speakers above which increasing the number of speakers becomes counterproductive to conference quality. Fixing the maximum number of simultaneous speakers is dealt in Refs 51 and 52 using ethnomethodology, and is conjectured to be

three. Thus, it is advisable to honor that constraint. Therefore, the proposed architecture should not mix more than three streams.

- *Less mixers*. There must not be many intermediate mixers (similar to CSs as in Ref. 53) in stages as in Ref. 49 because it brings in inordinate delays by increasing the number of hops and is not scalable with interactivity in focus. In Ref. 2, to allow impromptu speech, mixing is not done when the network can afford high bandwidth requirements for sending/receiving all the streams, but it is unnecessary [52].
- *Floor control*. Floor control for an audioconference (even videoconference) with explicit turn-taking instructions to participants renders the conference essentially a one-speaker-at-a-time affair, not a live and free-to-interrupt one. This way, the conference becomes markedly artificial and its quality degrades. Schulzrinne et al. [29] assume that only one participant speaks at a time. In this case, if applications are implemented with some control [24], the service becomes "gagging" for the users. Generally, floor control and, in particular, explicit floor control needs to be avoided.
- *Scalability*. For large conferences [53,54], a centralized conferencing solution cannot scale up. With multicasting, clients will have to parse many streams, and traffic on a client's network increases unnecessarily. Therefore, distributed architecture is a necessity; however, mixing should be done just before the play out.

4.6.2 Description

Two issues must be taken care of when building a Voice-over-IP (VoIP) conferencing system: (i) the front end, consisting of the application program running on the end-users' computers and (ii) the back end that provides other application programs that facilitate conferencing and the conference. The participating users are grouped into several "domains." These domains are local area networks (LANs), typically, corporate or academia networks. This distributed assumption demands distributed controlling and distributed media handling solutions, qualifying it to support large conferences. More explicitly, in each domain, we can identify several relevant logical components of a conferencing facility (Figure 4.6).

- An arbitrary number of end users (clients) take part in at most one audioconference at a time. Every user is in only one domain at a given instant, but can switch domains (nomadism). In our conferencing environment, these clients are regular SIP user agents (SIP UAs), as defined in Ref. 32 so as to gain in interoperability with other existing SIP-compatible systems. These clients are, thus, not aware of the complex setting of the backbone servers enabling the conference.

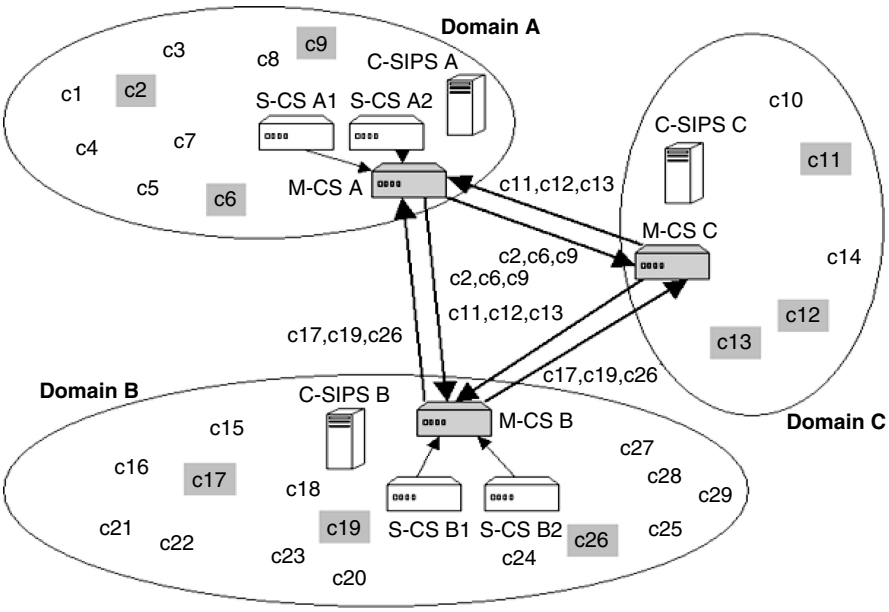

Figure 4.6 Example of a two-level hierarchy of CSs; the shaded clients are the one selected by the M-CS and will be sent to other domains' CSs.

■ One SIP server (SIPS) per domain* is set up to take care of the signaling aspects of the conference (clients joining, leaving, etc.) [55]. In particular, it is considered as a physical implementation encompassing different logical roles, namely, a SIP proxy server, a SIP registrar server, a SIP redirect server, and a SIP B2BUA (back-to-back UA) [32]. This physical implementation enables the handling of incoming/outgoing SIP messages by one or another logical entity according to the needs. SIPS is entrusted with maintaining the total service for the following advantages: (a) it works as a centralized entity that can keep track of the activities of the UAs in a conference; (b) it does all the switching for providing private bank exchanges (PBX) features; (c) it locates the UAs and invites them for a conference; and (d) it does the authentication and billing, if required. SIPSs in different domains communicate with one another using standard SIP messages as described in Ref. 55. If the load on a particular SIPS

* Domains in the context of wireless networks may also mean "clusters." It may have a cluster head, which can act as SIPS or CSs. However, formation of clusters is out of the scope of this chapter [60]. We use the term domain to differentiate between the approach in Ref. 30, which is referred to as cluster based (see also Table 4.1), and ours.

increases beyond its capability, it can create another SIPS in the same cluster/domain to share the load.

- One master CS (M-CS) (simply a CS) for each conference is created by the local SIPS when a conference starts. This server will be used for handling media packets for the clients in its domain. The M-CS can create a hierarchy of CSs inside a domain by adding one or more slave CSs (S-CSs) to accommodate all the active clients and prevent its own flooding at the same time. Its mechanism is described in Section 4.6.3.

The entities described here are exhaustive and conform to the SIP philosophy. Thus, the use of SIP makes this architecture more useful and interoperable with any other SIP clients or servers.

4.6.3 Selection of Streams

Similar to SipConf in Ref. 56, a CS [5] has the function of supporting the conference; it is responsible for handling audio streams using RTP. It can also double to convert audio stream formats for a given client if necessary and can work as translators/mixers of RTP specification behind firewalls [29]. The design of CS is similar to that in H.323 Multipoint Processor (MP) [31]. In brief, the CS receives audio streams from the endpoints and processes them and returns them to the endpoints. An MP that processes audio prepares $N_{\max}$ audio outputs from M input streams after selection, mixing, or both. Audio mixing requires decoding the input audio to linear signals (pulse code modulation [PCM] or analog), performing a linear combination of the signals and reencoding the result in an appropriate audio format. The MP may eliminate or attenuate some of the input signals to reduce noise and unwanted components.

The limitation of H.323 is that it does not address the scalability of a conference. The architecture proposes a cascaded or daisy chain topology [53], which cannot scale up to a large conference. A CS serves many clients in the same conference. Multiple CSs may coexist in a domain when there are several conferences under way. Signaling-related messages of CSs are dealt in Ref. 6.

The working of a CS is illustrated in Figure 4.7. Without loss of generality we select CS-1. For each mixing interval, CS-1 chooses the "best" $N_{\max}$ audio packets out of the M_1 (using a criterion termed "LN," described in Section 4.6.4). It may possibly receive and send packets to these CSs: CS-2 to CS-P. The set of packets sent is denoted by "ToOtherCSs." In the same mixing interval, it also receives the best $N_{\max}$ audio packets (out of possibly M_2) from CS-2, similarly the best $N_{\max}$ (out of possibly M_p) from CS-P. For simplicity, we ignore propagation delay between CSs, which indeed can be taken into account; it is beyond the scope of this chapter. The set of

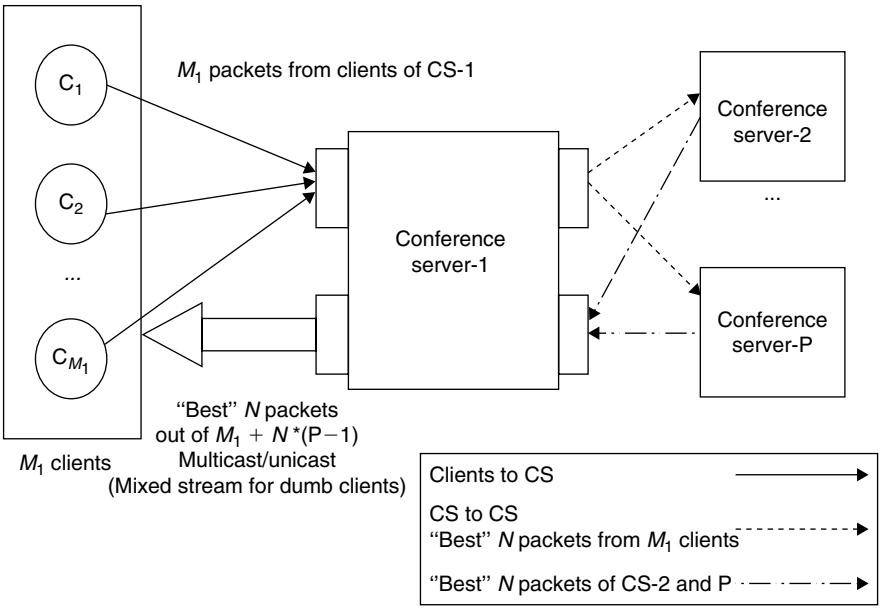

Figure 4.7 Schematic diagram of a CS.

packets received is denoted as "FromOtherCSs." Finally, it selects the best N_{max} packets from the set {ToOtherCSs ∪ FromOtherCSs} and passes these packets to its own group.

It can be seen that the set {ToOtherCSs ∪ FromOtherCSs} is the same at all CSs. This ensures that any client in the conference finally receives the same set of packets for mixing. Hence, all clients obtain a common view of the conference.

Similarly, for each time slot (packet time), a subset, F, of all clients is selected (using the same criterion) from the pool of packets from all other CSs plus the N_{max} clients selected locally. Their packets are mixed and played out at the clients. According to Refs 51 and 52, the cardinality of F, $|F|$, is N_{max} and is fixed at "three." The total number of streams in the network is slightly higher compared to distributed mixer architecture case (see Section 4.5.5). It is only N_{max} times that of a simple cluster-based approach of Ref. 30 (see Table 4.1); however, LN (described later in Section 4.6.4) can be used effectively to reduce the traffic to just N_{max} as explained in Section 4.6.7.

There are cases wherein the processing capacity of an M-CS is exceeded due to an excess of packets—from local and remote domains—to process. In this case, the M-CS will create one or more S-CS (Figure 4.6) and transfer some of its own clients as well as the new clients to it. This implies that a

maximum of two intermediate entities exist for each audio packet instead of two in the conventional setup. As the extra hop happens inside the LAN (LANs usually have a high-speed connectivity), it should not prevent us from using this hierarchy of CSs when there is a need to do so. In this configuration, the algorithm outlined earlier will be slightly modified—the audio packets will go from clients to their dedicated S-CS that will select N_{max} packets to send to the local M-CS, which will then select N_{max} packets from all its S-CSs in the domain before sending them to the remote domains. The incoming packets from other domains will be received by the M-CS, which selects N_{max} of them and sends them directly to the domain clients, bypassing the S-CSs.

4.6.4 Loudness Number

A basic question to be answered by the CSs is, how in a mixing interval it should choose N_{max} packets out of the M it might possibly receive. One solution is to rank the M packets according to their speech energies (this energy should not be confused with the signal strength), and choose the top N_{max}. However, this is usually found to be inadequate because random fluctuations in packet energies can lead to poor audio quality. This indicates the need for a metric different from mere individual energies. The metric should have the following characteristics [7]:

- A speaker (floor occupant) should not be cut off by a spike in the speech energy of another speaker. This implies that a speaker's speech history should be given some weight. This is often referred to as "persistence" or "hangover."
- A participant who wants to interrupt a speaker will have to (i) speak loudly and (ii) keep trying for a little while. In a face-to-face conference, body language often indicates the intent to interrupt. But in a blind conference under discussion, a participant's intention to interrupt can be conveyed effectively through LN.

A floor control mechanism empowered to cut off a speaker forcefully must be ensured to avoid the occupation of floors by one client. Otherwise, in a well-behaved conference, requirements are met by LN [7], which changes smoothly with time so that the selection (addition and deletion) of clients is graceful.

LN (λ) is a function of the amplitude of the current audio stream plus the activity and amplitude over a specific window in the past. It is updated on a packet-by-packet basis. The basic parameter used here is packet amplitude, which is calculated as root mean square (rms) of the energies in audio samples of a packet, and denoted by X_K. Three windows are defined as shown in Figure 4.8.

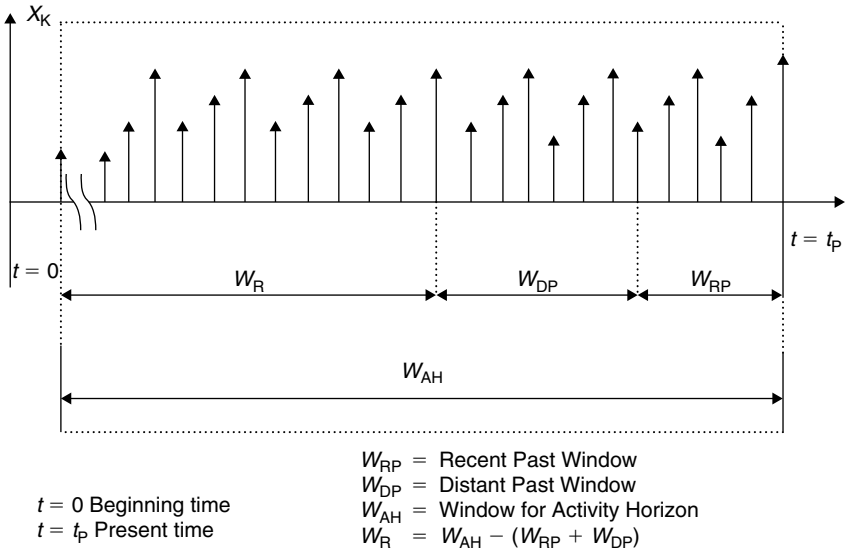

Figure 4.8 Illustration of various windows used in LN computation.

The present amplitude level of the speaker is found by calculating the moving average of packet amplitude (X_K) within a window called Recent Past Window starting from the present instant to some past time. The past activity of the speaker is found by calculating the moving average of the packet amplitude (X_K) within a window called Distant Past Window, which starts at the point where the Recent Past window ends and stretches back in the past for a predefined interval. The activity of the speaker in the past is found with a window called Activity Horizon, which spans the recent past window as well as the distant past window and beyond if necessary. Although the contribution of the activity horizon looks similar to the contribution of the recent past and distant past windows, past activity is computed from activity horizon window differently.

Define the quantities during these three intervals as L_1, L_2, and L_3. L_1 quantifies the Recent Past speech activity, L_2 the Distant Past speech activity, and L_3 gives a number corresponding to the speech activity in the Activity Horizon window quantifying the activity of the speaker in the past few intervals. L_3 yields a quantity that is proportional to the fraction of packets having energies above a predefined threshold (Equation 4.3). The threshold is invariant across clients.

$$L_1 = \frac{1}{W_{RP}} \sum_{K=t_P}^{t_P - W_{RP}+1} X_K \tag{4.1}$$

$$L_2 = \frac{1}{W_{DP}} \sum_{K=t_P-W_{RP}}^{t_P-W_{RP}-W_{DP}+1} X_K \tag{4.2}$$

$$L_3 = \frac{1}{W_{AH}} \sum_{K=t_P}^{t_P-W_{AH}+1} \Theta I_{\{X_K \geq \theta\}} \tag{4.3}$$

where

$$I_{\{X_K \geq \theta\}} = \begin{cases} 1 & \text{if } X_K \geq \theta \\ 0 & \text{otherwise} \end{cases}$$

The threshold θ is a constant. θ is set at 10–20 percent of the amplitude of the voice samples of a packet in our implementation here. LN λ for the present time instant (or the present packet) is calculated as

$$\lambda = \alpha_1 L_1 + \alpha_2 L_2 + \alpha_3 L_3 \tag{4.4}$$

Here, α_1, α_2, and α_3 are chosen such that

$$0 < \alpha_1, \quad \alpha_2 < 1, 0 < \alpha_1 + \alpha_2 < 1, \quad \text{and} \quad \alpha_3 = 1 - (\alpha_1 + \alpha_2) \tag{4.5}$$

Here, α_1 is the weight given to the recent past speech, α_2 the weight given to distant past speech, and α_3 the weight given to speech activity in the activity horizon window considered.

4.6.4.1 Safety, Liveness, and Fairness

The λ parameter has some memory depending on the spread of the windows. After one conferee becomes silent, another can take the floor. Also, as there is more than one channel, interruption is enabled. A loud conferee is more likely to be heard because of elevated λ. This ensures fairness to all participants. After all, even in a face-to-face conference, a more vocal speaker grabs special attention. All these desirable characteristics are embedded into the LN. A comprehensive discussion on the selection of the various parameters and the dynamics of LN are beyond the scope of this chapter.

4.6.5 Selection Algorithm Using the Loudness Number

Following the developments in Sections 4.6.3 and 4.6.4, we present the simple algorithm that runs at each M-CS (Algorithm 1). This algorithm is based on the discussions in Section 4.6.3. The globally unique set F is found using this procedure. The mechanism proposed here is also depicted in Figure 4.6, where a single conference takes place between three domains.

The shaded clients are the ones selected in their local domains; their audio streams will be sent to other CSs.

ALGORITHM 4.1: The Selection Algorithm at CSs
Input: *Streams from all Clients*
Output: *Selected* N_{max} *streams based on LN*
for each Time *Slot* ***do***

1. *Get the packets from all the Clients;*
2. *Find at most* N_{max} *Clients that have maximum* λ *out of* M *streams out of the received streams;*
3. *Store a copy of packets from those* N_{max} *Clients in database DB1;*
4. *Send these* N_{max} *packets to other M-CSs (on Unicast or Multicast, depending on the configuration and availability);*
5. *Similarly, receive packets from all other M-CSs and store them in database DB2;*
6. *Now compare the packets in DB1 and DB2 on the basis of* λ *and select a maximum of* N_{max} *amongst them (to form set* F*) that should be played out at each Client;*
7. *Send the* N_{max} *packets in set* F *to the Clients in its domain;*
8. *Mix these* N_{max} *audio packets in set* F *after linearizing and send it to dumb Clients who can't mix in the domain;*

end

4.6.6 Self-Organization of the Entities

This architecture is an improved version of the one proposed in Ref. 30. Similar to the clients self-organizing in Ref. 30, SIPS could give a list of CSs to the clients initially when a client joins the conference. Clients may also learn by listening to the transmissions about the CSs in the vicinity. Signaling with the use of standard SIP signaling is simpler compared to the media handling. We present a cursory discussion here on some of the maintenance and self-organizational issues mainly from the point of view of wireless networking.

First, although we showed SIPS and CSs as two different entities, we make the amendments now that SIPS and CS can be combined together to bifurcate the signaling and media handling components of the conferencing. In the following discussions, for the sake of simplicity we assume both the servers as one entity and we simply refer to them as servers handling all the conference-support activity.

Signaling is needed to send conference invitation and maintenance; the client that wants to start the conference should also act as a server in an

ad hoc wireless network and send invitations or advertisements periodically to contact potential participants. However, in a broadband network one can set up dedicated servers. Particularly, for complete *ad hoc* networks without infrastructure support we refer to Ref. 30 for setting up of the self-organizing conferencing mixers, which can also take care of the signaling. However, this sort of conference maintenance is limited to small and *ad hoc* structures. For a large conference with clients distributed all over we proceed to allocate some fixed servers in the infrastructure. In fact, the clients themselves may become servers if the resources permit or in the initial stages when the invitations are sent. We propose to use a Service Discovery Protocol (SDP) depending on the scenario (a survey as in Ref. 57) at the clients, although servers advertise the conference services offered. In some scenarios, individual invitations may be sent to all the participants. Once the service information is available, clients can join the servers that are near to them from the list of available servers. For each new client that joins the conference subsequently, they would also get the indication of how much load a particular server is handling which are in the vicinity. This is enabled because the servers talk to one another for exchanging media packets. This helps the clients to decide to whom to connect depending on aspects such as number of hops. Because the servers support limited number of streams, we can also have an S-CS that can take some load off the M-CS. An S-CS can be invoked in the clients as and when required.

The observed improvement in the perceived quality of the conference service is due to the following reasons. (1) Limiting the number of concurrent speakers to a low number such as three. Generally, in a conference if more than two participants speak, intelligibility is lost. Conversational analysis demonstrates that there would be a repair mechanism [51] in such a case. (2) Audio stream between any two clients passes through at most two CSs, thus, reducing the end-to-end delay. (3) As the streams are mixed only at the clients, or at the CS just before sending it to the participants (in case of participant's device having less resources), a customized mix of streams can be achieved. Individual tuning of mixing with different weights guarantees preservation of spatialism. Echo produced when a self-stream is selected can be avoided by reducing the weight. Nonetheless, feedback helps in reassuring a speaker that he or she is heard by all.

4.6.7 Reducing Bandwidth Consumption

The major problem identified in many of the mixing architectures is that when more than one stream is mixed, intelligibility of the speech reduces drastically. Thus, we select only $N_{\max}$ streams to be mixed at the end.

However, to reduce bandwidth consumption, one can use VAD at the clients [19,20]. Thus, many streams that are unnecessarily sent can be stopped because in a conference, by and large, only one participant would be speaking at a time. Although this architecture reduces the bandwidth on the links, it is slightly higher than distributed mixing (Section 4.5.5) as shown in Table 4.1.

We can use the characteristics of LN to avoid exchanging many streams between the CSs. As mentioned earlier in Section 4.6.3, each CS selects the best $N_{\max}$ from its domain and sends it to others. However, at every time slot it finds the best $N_{\max}$ considering the packets from all the CSs. Because it knows the best $N_{\max}$ streams that are selected in the previous time slot, the streams that are selected from its domain, the next time slot should have the LNs equal to or greater than the ones in the previous time slot. Thus, many of the CSs may not forward streams to other CSs. In a steady state, the number of streams exchanged across the CSs will be approximately $N_{\max}$ only. Thus, this architecture with LN saves significant bandwidth. One can note here that time slots are of 20–60 ms and thus, an improper selection due to time delay may last only for one or two time slots. See Ref. 5 for an in-depth analysis.

4.7 Open Issues

There are several open issues with respect to multimedia conferencing on wireless networks. In fact, many of the issues are prevalent in wired networks too. We list here some of the issues needing attention.

1. *Selection of CS (allocation problem)*. We assumed that the CSs can be found by a client, or allocated by a SIPS. We also made an assumption that clients would find the nearest CS based on some criterion, for example, the number of hops. However, none of these aspects were addressed in this chapter. Optimal allocation of clients to servers is a generic facility locator problem. It is found to be an NP-hard problem. However, depending on the particular case, some constraints can be relaxed and an approximate solution found. There exist some heuristic solutions for such problems [58]. There are many avenues for approximation and randomized algorithms tuned to this application. Many of the solutions proposed for this class of problems may not work here because the capacity of the devices to serve others is limited and the mobility-induced constraints are not easy to address.
2. *Mobility*. Distributed and dynamic compositions of wireless networks also pose the problem of ephemeral contacts. Thus, while supporting mobility some self-organization schemes need to be addressed here

to reduce breaks in service. The signaling protocol needs to be reliable and robust to maintain and stabilize the network. The protocol should take care of allocating a nearest CS when a node reconnects to the service. Such a protocol needs to be developed.

3. *Session mobility*. If the clients having multiple link layers move around and change their L_2/L_3 point of contact, handover takes place. Thus, from the point of view of application, connectivity and seamless session mobility have to be implemented in L_2/L_3 layers. For a large multimedia conferencing with distributed servers, these are very crucial for the success of multimedia applications. An improvement is to select the best link layer when there are multiple connection possibilities. Implementation on a multitude of devices calls for the selection of an appropriate link layer. With different radio links, gateway and related tasks will have to be looked into. A notable effort in this direction is IEEE 802.21 [59].
4. *Quality improvement*. There is a large body of research with respect to handling the delay, jitter, and packet loss for multimedia applications on wired links. In a wireless domain, these aspects are not the same. Further, adaptive delay algorithms, forward error correction (FEC), and coding for wireless multimedia are a few important needs.
5. *Enhancing LN*. LN has some limitations at present, such as it does not take into account the prosody of speech. There is a higher scope to improve this metric for a dynamic and auto floor control. Moreover, the effective use of LN yields better reduction in the bandwidth, which needs to be studied in greater detail.
6. *Implementation*. Implementation on devices with a variety of capacities and connectivities, and tuning of the devices for enhanced quality application are other challenges for a successful deployment of conferencing service on wireless devices.

4.8 Conclusions

This chapter described issues, existing solutions, and proposed an architecture for multiparty audioconferencing on wireless networks in detail. Network and audio-related issues were introduced. Mixing, an essential ingredient of multiparty conferencing, and its effects on the overall quality were discussed. Because the state-of-the-art solutions are not satisfactory, a distributed architecture was proposed based on a parameter called LN to select an appropriate subset of speakers. This architecture provides better quality of conferencing with less bandwidth consumption. We have indicated some directions for a better collaborative future.

References

1. Schooler, E. M., S. L. Casner, and J. Postel, Multimedia Conferencing: Has It Come of Age? *Proceedings of 24th Hawaii International Conference on System Sciences*, 3, 707–716, Jan 1991.
2. Radenkovic, M. M. and C. Greenhalgh, Multi-Party Distributed Audio Service with TCP Fairness, *Proceedings of the 10th ACM International Conference on Multimedia (MM'02), Juan-les-Pins* (France), pp. 11–20, Dec 2002.
3. Radenkovic, M. and C. Greenhalgh, Supporting Collaborative Audio in the Internet, *WSES Conference*, Greece, pp. 190–196, Sep 2002.
4. Radenkovic, M. and C. Greenhalgh, Supporting Collaborative Audio in the Internet, *Proceedings of WSEAS ICOMIV 2002* (Skiathos, Greece), pp. 3441–3447, Sep 2002.
5. Venkatesha Prasad, R., R. Hurni, H. S. Jamadagni, and H. N. Shankar, Deployment Issues of a VoIP Conferencing System in a Virtual Conferencing Environment, *ACM symposium on Virtual Reality and Software Techniques* (Osaka, Japan), Oct 2003.
6. Venkatesha Prasad, R., J. Kuri, H. S. Jamadagni, H. Dagale, and R. Ravindranath, Control Protocol for VoIP Audio Conferencing Support, in *International Conference on Advanced Communication Technology* (MuJu, South Korea), pp. 419–424, Feb 2001.
7. Venkatesha Prasad, R., J. Kuri, et al., Automatic Addition and Deletion of Clients in VoIP Conferencing, *6th IEEE Symposium on Computers and Communications* (Hammamet, Tunisia), pp. 386–390, July 2001.
8. Hindmarsh, J., M. Fraser, et al., Fragmented Interaction: Establishing Mutual Orientation in Virtual Environments, in *CSCW 98* (Seattle, WA), pp. 217–226, ACM Press, 1998.
9. Doerry, E., An Empirical Comparison of Copresent and Technologically-mediated Interaction based on Communicative Breakdown, PhD thesis, Graduate School of the University of Oregon, Eugene, OR, 1995.
10. Flament, M. and A. Svensson, Virtual Cellular Networks for 60 GHz wireless Infrastructure, *IEEE International Conference on Communications*, 2, 1223–1227, May 2003.
11. Eklund, C., B. Roger, K. L. Marks, and S. W. Stanwood, IEEE standard 802.16: A technical overview of the wirelessMAN air interface for broadband wireless access, *IEEE Communications Magazine*, 40(6), 98–107, 2002.
12. Bolton, W., Y. Xiao, and M. Guizani, IEEE 802.20: Mobile broadband wireless access, *IEEE Wireless Communications*, 14(1), 84–95, 2007.
13. Greenhalgh, C., S. Benford, and M. Craven, Patterns of network and user activity in an inhabited television event, *Proceedings of the ACM symposium on Virtual Reality Software and Technology* (London, UK), pp. 34–41, 1999.
14. Schooler, E. M., Conferencing and collaborative computing, *Multimedia Systems*, 4(5), 210–225, 1996.
15. Hendrix, C. and W. Barfield, Presence in Virtual Environments as a Function of Visual and Auditory Cues, *Virtual Reality Annual International Symposium (VRAIS'95)* (Research Triangle Park, NC), p. 74, March 1995.
16. Coding of Speech at 8 kbit/s Using Conjugate-structure Algebraic-code-excited Linear-prediction (CS-ACELP), ITU-T Rec. G.729, http://www.itu.int/itudoc/itu-t/rec/g/g729.html, 1996.

17. 40,32,24,16 kbit/s Adaptive Differential Pulse Code Modulation (ADPCM), ITU-T Rec. G.726, http://www.itu.int/itudoc/itu-t/rec/g/g726.html, 1996.
18. Andersen, S., A. Duric, et al., Internet Low Bit Rate Codec (iLBC), RFC 3951, 2004.
19. Hardman, V., M. A. Sasse, and I. Kouvelas, Successful multiparty audio communication over the Internet, *Communications of the ACM*, 41, 74–80, 1998.
20. Prasad, R. V., R. Muralishankar, et al., Voice Activity Detection for VoIP: An Information Theoretic Approach, in *Proceedings of IEEE Globecom* (San Francisco), Dec 2006.
21. Park, K. S. and R. V. Kenyon, Effects of Network Characteristics on Human Performance in a Collaborative Virtual Environment, *Proceedings of the IEEE Virtual Reality Conference* (University College London, London, UK), pp. 104–111, 1999.
22. Watson, A. and A. Sasse, The Good, the Bad, and the Muffled: The Impact of Different Degradations on Internet Speech, *Proceedings of MM 2000* (Los Angeles, CA), 2000.
23. Dommel, H. P. and J. Garcia-Luna-Aceves, Networking Foundations for Collaborative Computing at Internet Scope, *Interactive and Collaborative Computing (ICC 2000)* (Wollongong, Australia), Dec 2000.
24. Dommel, H. and J. J. Garcia-Luna-Aceves, Floor control for multimedia conferencing and collaboration, *Multimedia Systems Journal (ACM/Springer)*, 5(1), 23–28, 1997.
25. Dommel, H. P. and J. Garcia-Luna-Aceves, Network Support for Turn-Taking in Multimedia Collaboration, *IS&T/SPIE Symposium on Electronic Imaging: Multimedia Computing and Networking* (San Jose, CA), Feb 1997.
26. Gonzalez, A. J., A Distributed Audio Conferencing System, MS thesis, Department of Computer Science, Old Dominion University, Norfolk, VA, July 1997.
27. Radenkovic, M., C. Greenhalgh, and S. Benford, Deployment issues in multiparty audio for CVEs, in *Proceedings of ACM VRST 2002* (Hong Kong), pp. 179–185, ACM Press, Nov 2002.
28. Gonzalez, A. J., A Semantic-based Middleware for Multimedia Collaborative Applications, PhD thesis, Computer Science, Old Dominion University, Norfolk, VA, May 2000.
29. Schulzrinne, H., S. Casner, R. Frederick, and V. Jacobson, RTP A Transport Protocol for Real-Time Applications, IETF RFC 3550, July 2003.
30. Khedher, D. B., R. H. Glitho, and R. Dssouli, Media handling aspects of Multimedia Conferencing in Broadband Wireless *ad hoc* Networks, *IEEE Network*, 20(2), 42–49, 2006.
31. ITU-T Rec. H.323, Packet Based Multimedia Communications Systems, http://www.itu.int/itudoc/itu-t/rec/h/h323.html, 1998.
32. Rosenberg, J., H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, SIP: Session Initiation Protocol, RFC 3261, June 2002.
33. Mark/Kelley, Distributed Multipoint Conferences using SIP, IETF Internet Draft draft-mark-sip-dmcs-00.txt, March 2000.
34. Khlifi, H., A. Agarwal, and J. C. Gregoire, A framework to use SIP in *ad-hoc* networks, *IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, 2, 4–7, May 2003.

35. Fu, C., R. Glitho, and R. Dssouli, A Novel Signaling System for Multiparty Sessions in Peer-to-Peer *Ad Hoc* Networks, *IEEE WCNC 2005* (New Orleans, LA), Mar 2005.
36. Schooler, E. M., The impact of scaling on a multimedia connection architecture, *ACM Journal of Multimedia Systems*, 1(1), 2–9, 1993.
37. Handley, M., J. Crowcroft, C. Bormann, and J. Ott, Very large conferences on the Internet: The Internet multimedia conferencing architecture, *Journal of Computer Networks*, 31, 191–204, 1999.
38. Hac, A. and D. chen Lu, Architecture, design, and implementation of a multimedia conference system, *International Journal of Network Management*, 7, 64–83, 1997.
39. Deering, S., Host Extensions for IP Multicasting, IETF RFC 1054, May 1988.
40. Varshney, U., Multicast over Wireless Networks, *Communications of the ACM*, 45(12), 31–37, 2002.
41. Hardman, V. J., M. A. Sasse, A. Watson, and M. Handley, Reliable Audio for Use Over the Internet, in *Proceedings of INET95* (Honolulu, Oahu, Hawaii), Sept 1995.
42. Agnihotri, S., Improving Quality of Speech in VoIP using Time-Sacle Modification, Master's thesis, Indian Institute of Science, Bangalore, India, 2001.
43. Greenberg, S., Personalizable Groupware: Accomodating Individual Roles and Group Differences, in *Proceedings of the European Conference of Computer Supported Cooperative Work (ECSCW '91)* (Amsterdam), pp. 17–32, Kluwer Academic Press, Dordrecht, Sept 1991.
44. Hardman, V. and M. Iken, Enhanced Reality Audio in Interactive Networked Environments, in *Proceedings of the FIVE '96 Framework for Immersive Virtual Environments, the 2nd FIVE International Conference* (London, UK), pp. 55–66, 1996.
45. Yu, K.-Y., J.-H. Park, and J.-H. Lee, Linear PCM Signal Processing for Audio Processing unit in Multipoint Video Conferencing System, in *Proceedings ISCC*, pp. 549–553, Jul 1998.
46. Rangan, P. V., H. M. Vin, and S. Ramanathan, Communication architectures and algorithms for media mixing in multimedia conferences, *IEEE/ACM Transaction on Networks*, 1, 2030, 1993.
47. Yang, S., S. Yu, J. Zhou, and Q. Han, Multipoint communications with speech mixing over IP network, *Computer Communication*, 25, 4655, 2002.
48. Vin, H. A., P. V. Rangan, and S. Ramanathan, Hierarchical Conferencing Architectures for Inter-group Multimedia Collaboration, *ACM Conference proceedings on Organizational Computing Systems* (Atlanta, GA), pp. 43–54, Nov 1991.
49. Ramanathan, S., P. V. Rangan, and H. M. Vin, Designing communication architectures for interorganizational multimedia collaboration, *Journal of Organizational Computing*, 2(3&4), 277–302, 1992.
50. Ramanathan, S., P. V. Rangan, H. M. Vin, and T. Kaeppner, Optimal Communication Architectures for Multimedia Conferencing in Distributed Systems, *ICDCS* (Yokohama, Japan), pp. 46–53, 1992.
51. Venkatesha Prasad, R., H. S. Jamadagni, and H. N. Shankar, Number of Floors for a Voice-Only Conference on Packet Networks—A Conjecture, *IEE Proceedings on Communications, Special Issue on Internet Protocols, Technology and Applications (VoIP)*, 2004.

52. Venkatesha Prasad, R., H. S. Jamadagni, and H. N. Shankar, On the Problem of Specifying Number of Floors in a Voice Only Conference, *IEEE ITRE*, 2003.
53. Koskelainen, P., H. Schulzrinne, and X. Wu, A SIP-based Conference Control Framework, in *Proceedings of the 12th international workshop on Network and operating systems support for digital audio and video (NOSSADAV)* (Miami, FL), pp. 53–61, May 2002.
54. Rosenberg, J. and H. Schulzrinne, Models for Multi Party Conferencing in SIP, Internet Draft, *IETF*, Jul 2002.
55. Venkatesha Prasad, R., R. Hurni, H. S. Jamadagni, A Proposal for Distributed Conferencing on SIP using Conference Servers, *The Proceedings of MMNS 2003* (Belfast, UK), Sep 2003.
56. Thaler, D., M. Handley, and D. Estrin, The Internet Multicast Address Allocation Architecture, RFC 2908, *IETF*, Sep 2000.
57. Zhu, F., M. W. Mukta, L. M. Ni, Service Discovery in Pervasive Computing Environments, *Pervasive Computing, IEEE*, 4(4), 81–90, 2005.
58. Venkatesha Prasad, R., H. S. Jamadagni et al., Heuristic Algorithms for Server Allocation in Distributed VoIP Conferencing, *IEEE Symposium on Computers and Communications* (Cartegena, Spain) 2005.
59. IEEE 802.21, http://www.ieee802.org/21/.
60. Gu, Y., R. Venkatesha Prasad, W. Lu, and I. Niemegeers, *Clustering in Ad Hoc Personal Network Formation, Workshop on Wireless and Mobile Systems (WMS 2007)*, Springer Lecture Notes in Computer Science, vol. 4490, pp. 312–319, International Conference on Computational Science (4), Beijing, 2007.