An architecture of Conference Control Functions

Nadia Kausar, Jon Crowcroft
n.kausar@cs.ucl.ac.uk, jon@cs.ucl.ac.uk
Department of Computer Science
UCL, Gower Street, London WC1E 6BT
Tel: + 44 (0)171 504 4433

ABSTRACT

Conference control is an integral part in many-to-many communications that is used to manage and co-ordinate multiple users in conferences. There are different types of conferences which require different types of control. Some of the features of conference control may be user invoked while others are for internal management of a conference.

In recent years, ITU (International Telecommunication Union) and IETF (Internet Engineering Task Force) have standardised two main models of conferencing, each system providing a set of conference control functionalities that are not easily provided in the other one. This paper analyses the main activities appropriate for different types of conferences and presents an architecture for conference control called GCCP (Generic Conference Control Protocol). GCCP interworks different types of conferencing and provides a set of conference control functions that can be invoked by users directly. As an example of interworking, interoperation of IETF's SIP and ITU's H.323 call control functions have been examined here. This paper shows that a careful analysis of a conferencing architecture can provide a set of control functions essential for any group communication model that can be extensible if needed.

Keywords: SIP, H.323, Conference Control, Interoperability

1.0 Introduction

Conference control is a mechanism to co-ordinate and manage multiple users using different media in different conferences and it forms the basis for the operation of a conferencing system. It includes two types of services: user visible services, and internal management services that are invisible to users[9]. The user visible services are invoked by the human user and causes different actions in a conference. The services that are used to create and terminate a conference, managing and assigning roles, checking permissions, joining and leaving, accounting and billing etc, are classified as the user visible services. Internal management services are not obvious to the users, it includes services like interworking two or more different sets of architecture compliant conferences, maintaining consistency between different conference controllers.

A conferencing session can be an Internet telephone call, multimedia conference or an interactive game. It involves two or more participants. Whatever the nature of the conference is, it requires a control protocol to create, modify and terminate the session. In recent years, two different views of conference control have been proposed by two prominent standard bodies: the International Telecommunication Union (ITU) and the Internet Engineering Task force (IETF). The former has focused on the centralised **tight/formal** control of conferences where each participant knows who all the other participants are at all times and all have complete ordered and reliable floor control (discussed in section 2.2). Whereas, the IETF approach is referred as **loose/informal** type of conferencing[4] where senders may not know who is receiving, and thus, if there is any floor control, it can only be through rough consensus (i.e. statistical), rather than by deterministic algorithm. In addition, there are or can be other types of conference control functions that are different to the standard methods. Each model provides a set of functions that are not easily provided in the other model.

In order to co-ordinate and cooperate the participants and applications in a conference, protocols have been designed and implemented by among others Jorg et al[8], Schooler[12] and Handley[15]. Most of these conference control protocols have emphasised the user visible functionalities and not addressed the interoperation of two or more different architectures or provide a generic basis for communication in conferencing. Interoperation is important because the most popular and most commonly available conferencing systems used for research and commercial purposes currently are based on ITU and IETF standards which are not interoperable. As a consequence of that users have to maintain and support multiple stack technologies and various tools. Therefore, this paper presents a generic basis for communication in conferencing and then proposes an architecture of a conference control system known as GCCP (Generic Conference Control Protocol) that provides a set of user visible functionalities and a set of interoperation services between two or more different architectures. As an example, interoperation of the IETF SIP[3] and ITU H.323[2] call control functions have been examined.

The paper investigates how people communicating using different conference control mechanisms can be seamlessly integrated into a single conference control mechanism. This paper also presents a set of functions and their positions in an architecture essential for any conferencing system.

This paper is organised as follows: section 2 presents a generic basis for communication in conferencing systems, the following two sections look at the two types of conference management services and what has been implemented in GCCP in the reference of the generic model, in section 5 and 6 conference control function mapping between SIP and H.323 are discussed, section 7 and 8 are focused on transport and error control issues and finally the conclusion.

2.0 Generic basis for communication in conferencing: conference protocol stack

This section presents a framework that can be used as a basis of a generic framework for a conferencing system. It allows users to design a complete solution for a conferencing system, as well as it allows one to concentrate on specific details if required. The basis of group communication for conferencing, the conference protocol stack follows a layered architectural pattern which is suitable for designing a system whose dominant characteristics is a mix of low and high level issues. In this pattern high level operations rely on the lower level ones.

This conference protocol stack can be divided into three main sublayers shown in Figure 1. The top layer is divided into two segments: a) conference management and b) groupware applications. The Generic Conference Control Protocol (GCCP) discussed in this paper mainly falls under the category of conference management. Both applications and management on the top layer of the stack require services from the Session and Resource Management. The Resource Management part facilitates guarantees of available resources such as bandwidth and delay and deploys protocols like RSVP[12] (beyond the scope of this paper). The third layer provides communication services provided by the network itself. Components in each layer need to interact with each other.

<table>
<tr><td>Conference Management</td><td>Groupware Application</td></tr>
<tr><td colspan="2">Session and Resource Management</td></tr>
<tr><td colspan="2">Network</td></tr>
</table>

Figure 1: Conference protocol stack

The following sections concentrates on the conference management part of the architecture for conferencing systems. Conference management deals with all functions related to co-ordination and management of a conference. These functions could be visible or invisible to users. These are a set of functions that are user invoked like joining, leaving, creating sessions and others are required for management like media synchronisation, interoperating with other conferences or provide consistency. The rest of the paper deals with the analysis, design, requirements and implementation issues of the conference management layer and GCCP.

3.0 GCCP's Interactions

Conference control has mainly two different types of interactions.

- *Inter-system commination* occurs between peer entities running on different systems: the conference management entities communicate with one another using a conference management protocol over the network. The set of application entities exchange information within application sessions. These application sessions are isolated from one another and they may or may not be controlled by the conference management entity. The peer applications may need to be have compatible attributes, for example, they will need the same codecs.

- *Intra-system communication* is used to coordinate the otherwise unrelated local groupware applications on each teleconferencing systems and integrate them with the conference management entity as well as to provide access to the conference control services. Examples of intra system communication system includes systems like mbus[10] or LBL's message bus.

Figure 2.0 is an example of a system model that shows the interactions between two teleconferencing systems in a point-to-point conference and the various protocols in use. GCCP uses this model to provide inter-system communications.

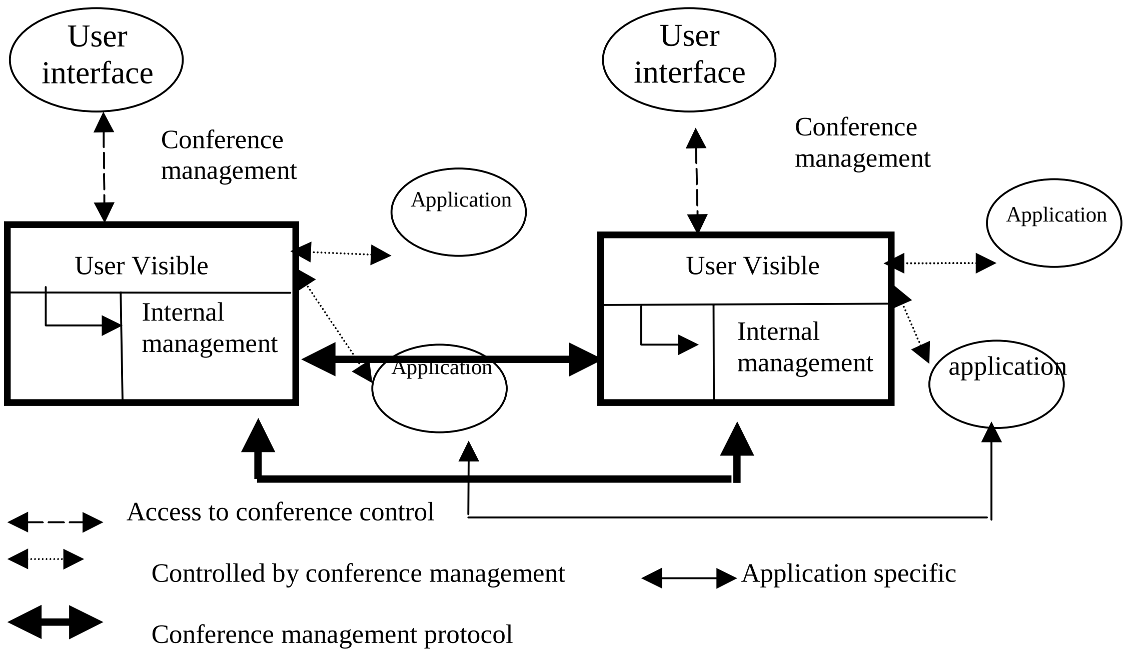

Figure 2: Protocols for conference management services

Users access different types of visible services like inviting another user, joining a conference (discussed in section 4.0) via a user interface. If one user is inviting another user who happens to be in a different conference, the internal management functions of GCCP perform the interoperation. Also GCCP (the conference management protocol in the diagram) supplies connection management in the form of session establishment, maintenance and disconnection; it performs these tasks remotely with peer connection management entities in remote units using a connection control protocol (TCP or UDP over IP as underlying transport protocol).

4.0 GCCP's user visible services

The user visible services of a conference control service are divided into different functional groups. These are: conference configuration, participation management, floor control and security[9].

- Conference configuration: Conference Configuration essentially refers to defining a conference profile. Means for specifying and enforcing conference policies are also provided[1]. Charging and billing to join a conference may also be a part of this service. A conference profile can also define permissible participants, available roles (e.g. chair, speaker) and associated permissions. The role assignment follows: a) assign role initially or b) request role and then grant/deny/share role.

- Participation management: Participation management comprises services for setting up conferences, point-to-point calls, group or individual invitation, and termination.
  Furthermore, functions for charging the membership of a conference are included. Participants may join or leave a conference on their own, or they may be invited or excluded from a conference. Changing from one conference to another is included in this function as well.
- Floor control: Floor control (in CSCW) is a metaphor for "assigning the floor to a speaker", which is applicable for any sharable resource within conferencing and collaboration environments. A floor is an individual temporary access or manipulation permission for a specific shared resource, e.g., a telepointer or voice-channel, allowing for concurrent and conflict-free resource access by several conferees.
- Security functions: Security functions are value added options for conference control and are a part of participation management as well as conference configuration. Authentication is performed when a participant enters the conference and may be repeated arbitrarily during the conference course.

GCCP acts as a server, it does not have a user interface associated with it. The assumption is that the GCCPs reside on machines scattered throughout the network. Among the user visible functions mentioned above, GCCP performs the most essential roles that are common in any conferencing systems. These functions are: join, invite, leave, floor control and basic error controls. The reason for choosing these functions is because in a lifecycle of any canonical conference these functions are essential. The lifecycle of a session is shown in Figure 3.

When a conference is created, the first user/initiator either invites a participant or the session could be advertised. GCCP does not perform the advertising itself because different conferencing applications have their own way of advertising conferences like SDR[16]. The job of GCCP starts when the participants are trying to invite another participant. If the invitee is already logged on, an invitation message appears on their screen. Otherwise, GCCP returns a message indicating they are not contactable.

Conference creation (assignment of a conference ID)

Advertise the conference | invite participant

Participant is chooses to enter conference | Invitee is notified

Reject invitation

Join conference   (if password is needed, provide it)

Media flow (if different media then media gateway is used)

Floor control (send media if the current participant is the floor holder)

End of a conference

Leave

Figure:3.0   Life cycle of a typical conference

If the participants have a conferencing application loaded on their machine (which could be H323 compliant or Mbone[6] based application) the invitation message or the equivalent of telephone “ring” appears on their machine in the format that is specified within that particular architecture. If the initiator/inviter is using an architecture that is different to invitee’s, the GCCP helps to translate that call control functions (see section below for GCCP’s gateway functionalities).

5.0 Internal management

The user-visible conference management services described in the previous section provide functionalities to control the source of a teleconference and reflect the participants’ behaviour. Normally users carry out these functions using a conferencing /groupware application. These groupware applications are considered independent entities rather than merged with a conference control tool. Currently most of the Mbone based conferencing applications – audio, video communication tools as well as signalling can operate stand-alone, i.e. without a conference management entity.

It is the task of the internal management services of conference control to integrate the different types of conference management entities in order to make them appear as coherent teleconferencing system[12]. As discussed in section 2, Intersystem communication and Intra system communication are two ways to solve the synchronisation and integration aspects of conferencing. In order for the Intersystem communication to accomplish the integration and provide a richer set of services, it must perform a) interoperability b) consistency.

Interoperability means that users from two or more different application systems can collaborate. Regardless of the supplier of the application, if there are a number of tools and media available to the users they should be able to exchange information. The level of interoperability can vary. For example, user A from system 1 can only receive and send audio, whereas user B from system 2 can receive audio and video. So there must be a capability exchange mechanism to find out if they can be interoperated such that both users can at least send and receive audio. Therefore, it can be said that there are mainly two types of gateway involved in this situation : a) call control or signalling gateway b) media gateway. A part of GCCP's service is to provide call control gateway.

Consistency provides a way to report a list of different types of application, participants and their status. As the number of participants scale to thousands over the Internet, it becomes very difficult to get an exact list of all the participants and their status. Therefore, it may be possible to get the status of a number of participants at one time from one link which may not be consistent with the number appearing to another link at the same time. There are ways to address the problem [14].

6.0 Main operation of GCCP

GCCP follows a number of steps to accomplish its function as gateway and a conference control provider. The functionality of GCCP can be divided into three categories: a) initialisation and registry update b) client registration and c) session management

Initialisation and registry update

GCCP initialises to process messages. When a client processes a CONNECT[17], (the socket system call to establish a connection with the server), GCCP updates its registry. The registry must keep a track of the following:

- Protocol type - the protocol types of conferences (e.g H.323, Mermaid or SIP)
- Number of participants – the number of participants for which GCCP maintains information and can forward packets
- Participants' IP addresses – the address where the data can be forwarded to
- Participants' port number - GCCP associates different types of applications with different well-known ports. For example, the H323 stack uses port number 1720 where as a SIP initiator will use port no 5060 for delivering control messages like invite a participant, request floor, leave etc.
- Current status (e.g. floor holder) – if a participant/port is sending data/audio that is not current floor holder dot forward the packets to anybody else
- Link status (e.g. broken link, slow link if possible etc.) – if a "Keep Alive" message didn't appear then delete the link

Client registration

A client must register itself with GCCP before it can issue a request to the system or participate in a session. So for example, if a "CONNECT" call came from port no 1720 for example, GCCP knows it will be a H.323 based client. GCCP updates the registry with the port number of the conferee, the type (in this case H.323), the IP address, floor holder status (could be 0 or 1) and the link status (which is 1 if the link is alive). GCCP continues to perform the operations of updating and adding the registry as participants come and go.

Session management

A client creates a session by specifying the initial attributes (passwords, policies for floor etc.). Creation of a new session involves two stages: negotiation of capabilities (like codecs) and allocation of resources. When a client wants to invite another client, GCCP has to make sure that they both have a set of similar capabilities like both are capable of understanding text or ascii values for data. The issues related to resource allocation are beyond the scope of this paper. Session management also involve deploying a floor control policy. When a conference actually starts, GCCP checks the floor holder status. If one of the ports is sending data that is not the current floor holder, GCCP does not forward the packets to all the other participants. Figure 4.0 is a representation of GCCP's main operations like interoperating two different types of clients based on different architectures and checking floor control policy. This figure briefly illustrates the main operations discussed above.

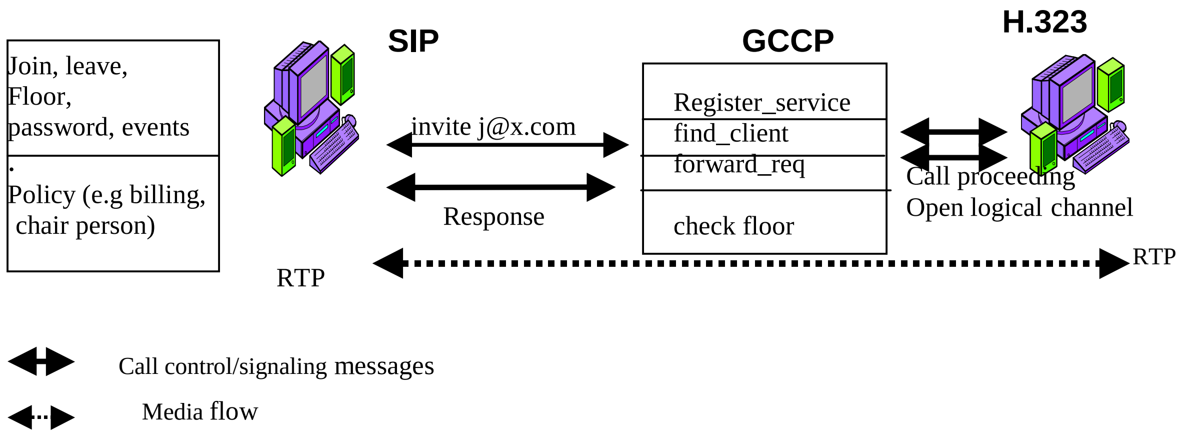

Figure 4.0: Topology of different types architectures interoperating using GCCP

In the following section, GCCP's the call setup and gatewaying SIP and H.323 are discussed.

6.1 GCCP's interoperability services

Out of the two features of internal management mentioned in section 5.0, GCCP mainly provides interoperability. GCCP acts as a gateway in this case that provides conference control functions that are commonly available in both the protocols which allows users of different protocols to interact with each.

When designing a gateway that translates different functions between two completely different architecture, there are certain considerations have to be taken into account. The following sections look at these aspects of a conference control gateway.

6.1.1 Main conference control contrast between H.323 and SIP

To provide interoperability in conference control level between H.323 and SIP, the following issues are the most difficult to resolve.

- Modularity - The conferencing applications based on H323 provide a complete package in one module. In other words, a H323 based conferencing like Intel's Proshare will deliver audio, video and signalling facilities as one compact package.
  Whereas as mentioned in section 2.3 Mbone based audio and video communication tools as well as a signaling protocols are independent applications. They can operate as standalone packages. SIP is an application layer

(signaling) protocol for creating, modifying and terminating sessions with one or more participants. It is used to invite users and invitations used to create sessions carry session descriptions which allow participants to agree on a set of compatible media types. These similar functions are also provided in H323, but it cannot be separated as a different module.

- Supporting protocols - In H323 based systems support for voice is mandatory, while data and video are optional, but if supported, the ability to use a specified common mode of operation is required, so that all terminals supporting that media type can interwork. These modes of operation described in the ITU specifications are not necessarily provided in the Mbone based conferencing specifications. Recommendations in the H.323 series include H.225.0[18] packet and synchronization, H.245[19] control, H.261 and H.263 video codecs, G.711, G.722, G.728, G.729, and G.723 audio codecs, and the T.120 series of multimedia communications. SIP mainly matches some of the functionalities provided by H.225 and H.245 but also performs some other call control functions that are not supported in H.323.

- Advertising – H.323 does not use a standardized protocol to advertise its sessions whereas Mbone based conferencing uses SAP (Session directory Announcement Protocol)[16] to advertise the sessions using IP multicast. H.323 based product NetMeeting uses LDAP to list the participants and their availability. Therefore, a publicly available seminar can be advertised over the Mbone will not be visible by a H323 based application.

- Messaging – H.323 uses a binary representation for its messages, based on ASN.1 and the packet encoding rules (PER). ASN.1 generally requires special code-generators to parse. This makes it harder to debug the messages generated by H.323. SIP encodes its messages as text[14.]

- Multicasting- H.323 supports UDP or multicast for user location, it does not currently provide for group invites. Therefore, although an IP multicast may be running as a transport protocol, H323 cannot take the advantage from the network layer. SIP requests can be sent via multicast.

In conclusion, the main difficulties to design a gateway like GCCP are as follows: a) first of all, identify the functions that are incorporated in H323 which are similar to SIP's main three requests (INVITE, ACK and BYE) . This will allow SIP users to at least join a common session. b) After that, follow the modes of operations that are described in the respective specifications. c) Identify if the user can use multicast capabilities. d) Finally, discard invalid messages and appropriate error messages should be sent different entities involved.

6.1.2 Basic Call set-up between H.323 and SIP

The following example shows interactions that take place when a H.323 client invites a SIP client using GCCP.
Call signalling messages in H.323 may be passed in two ways. The first method is Gatekeeper routed Call signaling (GRC). In this method, call-signaling messages are routed through the gatekeeper between the endpoints. The second method is DiRect Call Signalling (DRC). In this method the call Signalling messages are passed directly.

In figure 5, the calling endpoint Bell sends a set-up message to the well known port of callee (in this case, the GS on behalf of watson is receiving the calls to start with). The gateway then informs the caller that the call is being processed followed by the capabilities of the receiving terminal. It is not necessary that a terminal understands or stores all incoming capabilities; those that are not understood, or can not be used shall be ignored[19]. Once the reliable H.245 control channel has been established, GCCP places the invite message in SIP format to the callee (watson) who is able to process SIP messages. Once the callee answers the call and both parties are prepared to interact, additional channels for audio, video, and data are established on the caller's side (based on the outcome of the capability exchange).

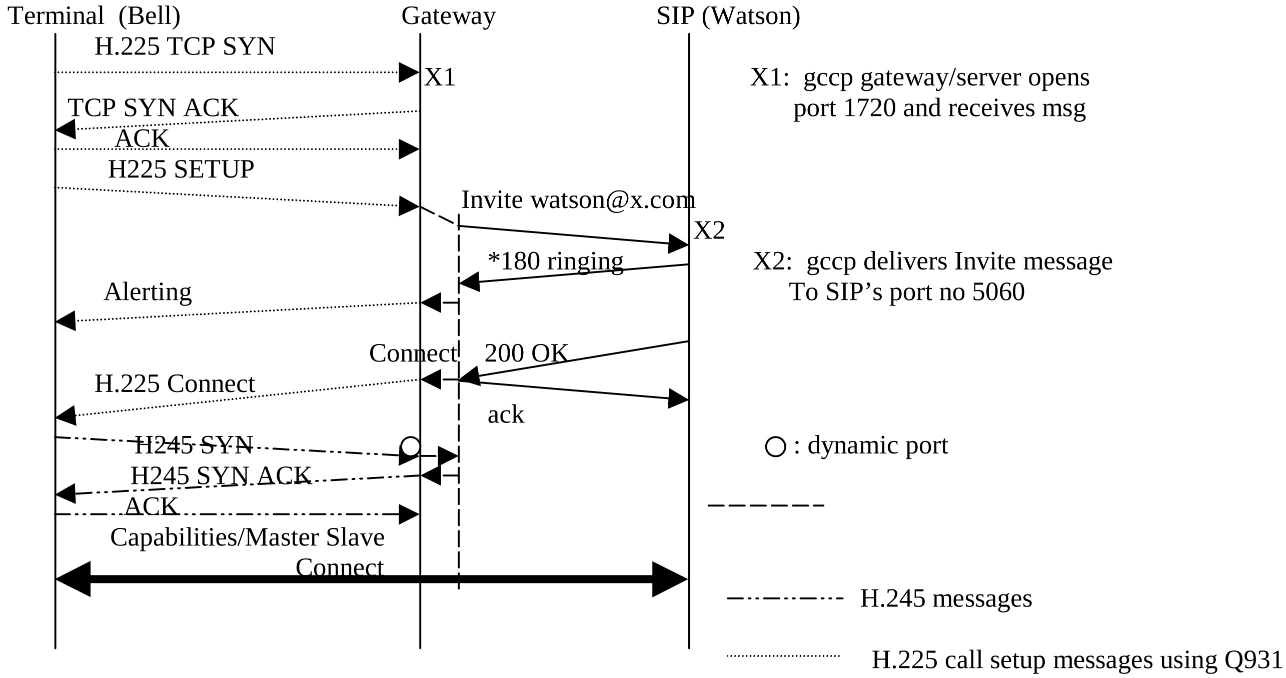

Figure 5: Messages exchanged between H.323 and SIP for "Invite"

* For two-party Internet phone calls, In the example above, Bell calls Watson which is being translated by GCCP. A sample response to the invitation below, the 1st line is the status of client. The Via headers are replaced as the req moves hop by hop towards invitee. Call-ID is unique in this invite.
SIP/2.0 180 Ringing
Via: SIP/2.0/UDP csvax.cs.caltech.edu;branch=837 ;uaddr=128.16.16.16;ttl=16
From: Bell <sip:Bell@cs.ucl.ac.uk>
To: Watson <sip:Watson@x.com> ;tag=9883472 Call-ID: 296331305 Case: 1 INVITE

The response from GCCP to the caller is as follows:
SIP/2.0 200 OK
Via: SIP/2.0/UDP csvax.cs.caltech.edu;branch=837 ;uaddr=128.16.16.16;ttl=16
From:Bell <sip:Bell@cs.ucl.ac.uk>
To: Watson<sip:watson@x.com> ;tag=37462311 Call-ID: 9883472
CSeq: 1 INVITE

For more details on the exact format of messages please refer to SIP and H.323 specifications[*].

6.2 GCCP Message Format

GCCP is a text based protocol. Any request or response sent to the GCCP server or received from the server must contain certain fields in the following format in the following order:

---

[*] A scenario above applies to H.323, but with the following changes:
- Terminal Capability Set is initiated from the gateway
- Terminal capability exchange is tested with just audio encoding

Identifier Value

"ID": *ConfID* – this is the conference ID of a particular conference (randomly generated ascii values) , 6 bytes long

"DestAddr": *Destination Address*- the IP address of where the messages should be delivered to in ascii text format

"SrcAddr": *Source Address* - the IP address of where the message came from in ascii text format

"ConfType": *ConferenceType* – the type of conference stack, normally upto 8 charecters long. Currently defined values are: H.323, SCCP, CCCS, SIP, MBUS

"M": *Message* - It consists one of the following control messages:

GCCP_JOIN
GCCP_LEAVE
GCCP_INVITE
GCCP_LIST -request to see the participants in a conference GCCP_FLOOR_REQ/
GCCP_FLOOR_ACC/GCCP_FLOOR_REJ

"R": *Reserved* – Left blank at the moment (for future purposes)

"V": *Version* – the version of GCCP server running

All the above fields are separated by reserved delimiters.
Token objects by convention have upper case names, e.g., "FLOOR". Token can have zero or more holders(members).
GCCP is responsible for maintaining (upto 25 members in each server) a consistent list of all current participants and the applications that are in use.

7.0 Transport services

In a group communication scenario, any number of senders may be distributing information to the group or to a subset of the group. Each piece of information may be destined for any number of recipients. These messages can be either control messages or data. GCCP needs a way to distinguish the control messages from media data.

Any control messages (for example, GCCP_JOIN, INVITE , BYE etc.) will be sent to the GCCP server on the Unicast channel. For example, when H323 is sending one of its control messages it uses the logical channel (see Figure 6).

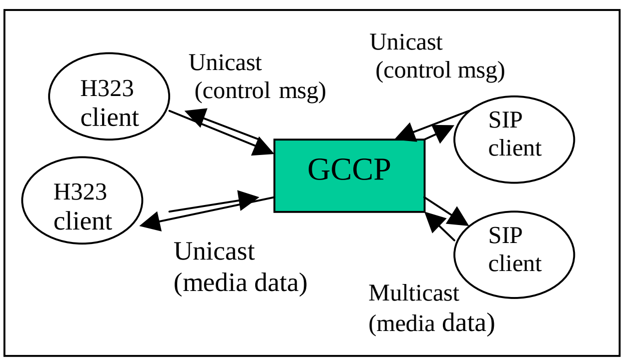

Figure 6: Methods of forwarding and receiving packets in GCCP

If control messages arrive on a multicast port, GCCP ignores those messages as control message. Only data that arrives in multicast channel gets forwarded to all the possible recipients. If the clients are capable of receiving multicast messages, GCCP multicasts it to them otherwise it uses the Unicast channel to forward data. This process allows the clients to be consistent, otherwise, all the control messages like GCCP_FLOOR_REQ messages will be multicast/broadcast to everyone, causing confusion.

The simple diagram shown above does not scale to hundreds of clients particularly, however, it is an example give a concept of Unicast channels used for control messages and multicast channels are used for media data. In order for it to scale, several distributed GCCP servers are required.

8.0 Error control

Different conferencing stacks have different ways to handle failures. For example, normally, in H.323 the underlying reliable protocol of the H.245 Control Channel uses appropriate effort to deliver or receive data on the channel before reporting a protocol failure. Therefore, if a protocol failure is reported on the channel, the H.245 Control Channel, and all associated logical channels shall be closed. This will be done as if the endpoint had issued the H.245 endSessionCommand. With SIP there are mainly two places where error can be generated; either the server or the client. When a client generates a Status code 400 , it means the request contains bad syntax or cannot be fulfilled at this server.

The error handling of GCCP is kept very simple. If GCCP receives any of the above error codes or messages it maps appropriate error messages to underlying protocol (e.g., it can send a 500 status code to SIP or send endSessionCommand for H.323). When a client joins GCCP, it checks for system call failures and errors in user input. Fatal errors include failures in creating the socket systems calls on which to listen for TCP connection requests and failures to listen to multiple sockets due to blocking calls. If these occur, a message is written to standard error and the program exits.

9.0 Conclusion

In this document, an architecture for conference control has been presented called GCCP (Generic Conference Control Protocol). It provides a set of user visible services and it interoperates different types of architectures as a part of internal management features of a conference control. In this paper, the interoperation of SIP and H.323 have been focused on. Although currently there is significant effort is being put into defining how the existing telephone network services will interwork with the Internet, the main objectives of these proposals/projects are to define how voice-based services will work between these two networks. In this research, GCCP is providing a set of services that are not just based around voice-based services but have the capability to provide conference control functions, also better known as "value added services". An architecture like GCCP provides "good" features of both tightly coupled and loosely coupled conferencing which is not provided in one architecture on its own. Some of the major advantages include: a) if H.323 based system wants to do group invites it can do using GCCP's capability to do Multicast invites b) people on the Internet can conference with an H.320 system running on the ISDN, therefore getting the network independence c) SIP users can take advantage of H.323's gatekeepers functionalities like bandwidth management.

In this paper, the IN services like call transfer, call on hold, answering services etc. are not compared or discussed. In future, these services will need to be interworked. The future work will also include experiments on distributing GCCP server across the Internet and getting performance measurements on how many conferees can interact with each other and how they can maintain consistency.

ACKNOWLEDGEMENT

Authors would like to thank Ian Marshall of BT labs, Ipswich for sponsoring this work and Ken Carlberg, Orion Hodson and Steve Wilbur to provide useful comments and suggestions throughout the process.

REFERENCES

1. CuSeeMe Web Page http://cu-seeme.cornell.edu/

2. H.323 ITU Recommendation, “Packet based multimedia communication systems”, 1998
3. Handley98,Schulzrinne, Rosenberg J.“SIP: Session Initiation Protocol”, Internet draft -ietf-mmusic-sip-12.txt

4. Kausar N., Crowcroft J. “General Conference control Protocol”, IEE telecommunication 1998

5. Kouvelas I97., O. Hodson, V. Hardman “Redundancy Control in Real-Time Internet Audio Conferencing, Proceedings of the 1997 International Workshop on Audio-Visual Services Over Packet Networks 15-16 September 1997
6. Mbone FAQ www.mbone.com

7. McCanne, S., and Jacobson, V., vic :A Flexible Framework for Packet Video. ACM Multimedia, November 1995, San Francisco, CA, pp. 511-522.

8. Ott J. 96, Bormann C. “Simple conference control Protocol SCCP”, Internet draft 1996

9. Ott J. “A multipoint data communication infrastructure for standard based teleconferencing system”, PhD thesis 1997, Technical University of Berlin

10. Ott J, Perkins C. june 1999 Internet draft: “Requirements for Local Conference Control “, http://search.ietf.org/internet-drafts/draft-ott-mmusic-mbus-req-00.txt

11. RSVP RFC 2205, Resource Reservation Protocol, Bradan, Zhang, Berson ftp://ftp.isi.edu/in-notes/rfc2205.txt

12. Schooler E.M "Case Study: Multimedia Conference Control in a Packet-switched Teleconferencing System",Journal of Internetworking: Research and Experience, Vol.4, No.2, pp.99-120 (June 1993)

13. Schulzrinne98, Rosenberg J "Timer Reconsideration for Enhanced RTP Scalability", Proceedings of IEEE Infocom 1998,CA, USA

14. Schulzrinne H., Rosenberg, "A Comparison of SIP and H.323 for Internet Telephony", proceedings of the 1998 Network operating System Support for Digital Audio and Video (NOSSDAV '98), July 1998, Cambridge, England.

15. Handley, I. Wakefield, and J. Crowcroft, "CCCP: confrerence control channel protocol-a scalable base for building conference control applications," *ACM Computer Communication Review*, vol. 25, pp. 275-287, Oct. 1995.

16. Handley M., Whelan M., Perkins C. “Session announcement Protocol”, Internet draft http://search.ietf.org/internet-drafts/draft-ietf-mmusic-sap-v2-01.txt

17. Stevens “Unix network Programming”, Prentice Hall, Vol 1

18. H.225 ITU Recommendation “Media stream packetization and synchronization on non-guaranteed quality of service LANs”

19. H.245 ITU Recommendation “Control Protocol for Multimedia Communication”