Network Working Group D. Reed
Request for Comments: 1324 May 1992

A Discussion on Computer Network Conferencing

Status of this Memo

This memo provides information for the Internet community. It does not specify an Internet standard. Distribution of this memo is unlimited.

Abstract

This memo is intended to make more people aware of the present developments in the Computer Conferencing field as well as put forward ideas on what should be done to formalize this work so that there is a common standard for programmers and others who are involved in this field to work with. It is also the intention of this memo to stimulate the computer community and generate some useful discussion about the merits of this field.

Introduction

Computer network conferencing is just now starting to grow and take advantage of the modern technology that is available. Although there are some systems which have been around for some time (BRC - Bitnet Relay Chat and IRC - Internet Relay Chat), there has not been any real move to bring them together under a single protocol. This has led to various protocols and different systems coming to life. As these different systems continue to pop up, it is becoming more obvious that there is need of a standard in this area for developers to follow without the need of worrying about protocol clashes.

In any implementation of a conferencing program, there are likely to be two main components: (1) a client program or interface which users enter commands into (hereafter referred to as a "client") and 2) a server program which acts as a multiplexor for various clients which connect to it. There are other expectations and requirements for both servers and clients which are mentioned in more detail later.

Table of Contents

1.0 NETWORK CONFERENCING TODAY

1.1 Conferencing in general today

Conferences today are an integral part of the business world in many ways. A conference may be held to reassure staff about company problems (boost moral) or may be held by a few directors in an emergency situation where a carefully considered solution is needed. Conferences also form the cornerstone of workshops held where various groups of people, who attend, are to be briefed on new developments. In nearly all of these situations, there will be a group of 2 or more, where each speaks and listens to others. There exist PABXs and

other features of the telephone system which provide for conferencing between people around the globe at a cost effective rate. The only place which really lacks any formal form of conferencing is the internet, although many unofficial conferencing systems already exist, spanning the globe or providing local forums.

1.2 Talk/phone vs. conferencing

To provide instantaneous communication between two users on unix and other multiuser systems, interprocess communication is commonly used either over a network or other local methods. The diversity of unix platforms has introduced as many problems as the presence of various operating systems on the net. Commonly, those on Unix based machines are unable to talk to those on VMS or VM machines. The occasion even arises where two Unix hosts are unable to talk to each other due to different talk protocols.

1.3 Advantages of realtime computer conferencing

By providing a standard for computer conferencing, it should eliminate the problem of who is using what computer. This will mean that someone from a VMS or VM machine can talk with one or more people without having to worry whether their counterpart has an account on a compatible machine for their choice of communication. Electronic mail (email) has already reached this position with most modern mailers on the internet being compliant with RFC822. It is therefore not unreasonable to expect this of realtime conferencing which is to talk as USENet is to email; although of those four (4), only email and news have been covered by RFCs.

USENet is a vast resource and immensely useful for many people around the globe. It does, however suffer from a high noise to signal ratio. It would be unwise to expect much difference in performance from conferencing.

By providing the means for realtime computer conferencing, it opens up a whole new area of usefulness to computers. For both students and staff alike, it opens up new possibilities. In educational institutions where there is a high level of project work with groups of more than 2, it means that students can work from home or other remote places and discuss their project with their fellow students in a manner which would be similar to all students having a conventional meeting or conference. This same situation also applies to staff members. For those who have previously relied on email between fellow researchers in many remote institutions, computer conferencing brings the world together, onto the researchers screen where they can trade ideas and code in real time. Traditionally to achieve these goals, the phone would have been used and a teleconference setup and

it will probably remain so for many years to come with video phones too. However, with phone conferencing, when people talk over each other, the quality of the discussion is degraded.

2.0 Goals for what a protocol should provide

In producing a protocol for conferencing over computer networks, the following problems must be considered:

2.1 State Information problems

The number of users who are a part of the conference may fluctuate continuously by a large amount over any given period of time. The protocol should endevour to make disruptions such as these as smooth as possible but at the same time, keep the realtime feel in the conference. It is not acceptable to buffer a user who quits for any given time but at the same time, if a server has network problems with connecting to another one, it may be wise to find some way around the continual stream of state messages that are passed - or - at least a way to reduce the number.

2.2 Network barriers

Members of a conference may be on physical networks which cannot directly communicate with each other, such as those used from a host on a commercial network talking via a bridge to someone from a network directly connected to a network directly accessible from theirs. So in this case, the users involved have no need to directly use the bridge (as required by unix talk) since the server on the gateway host provides a way for messages to be passed in and out of the unreachable sections. In this case also, there is a minimum security risk to the network which is otherwise unreachable.

2.3 User needs

2.3.1 User privacy

Members of a conference may wish to exchange ideas privately without fear of others eavesdropping or interrupting the current conference. To facilitate this, there should be some support by the protocol to pass messages from one user/client directly to another.

It is also reasonable for a user to want to be able to hide in one way or another from other users, effectively making themself invisible to other users.

2.3.2 Realtime Expectations

Users will expect conferencing to be real time, giving the thereby demanding that the protocol supply a quick, efficient, reliable and accurate delivery of a message. Only when these requirements are met can a conference system hope to be of any use to its users.

2.4 Message Delivery

2.4.1 Deficiencies in using IP only

In routing between conference servers, the problem of routing messages is an important issue. If there was a server for the conference at each domain, this wouldn't be an issue, one could simply do some sort of lookup and find the server for it. This is not the case and unless such a server becomes a standard item for unix machines, it is not reasonable to expect it to ever be so. Thus the need for a layer on top of TCP/IP is needed to deliver messages between the servers for the conference.

2.4.2 Flexibility

The routing protocol used should not be inflexible and should allow for routes to change over time in much the same way as RIP does now. However, there is no need for a special routing protocol such as RIP since this is already part of IP's functionality. Routing information should be updated automatically when the server receives information via that route whether it creates or destroys a route.

2.4.3 Building a flexible transport protocol on top of existing ones

If such a conferencing service is built upon TCP/IP, it is therefore possible to build an abstract routing model which has no relation to the TCP/IP model. However, it is not wise to ignore the presence of either TCP or IP since by integrating them into the protocol, it is easier to use their strengths. If the protocol relies too heavily on TCP/IP features, it will also inherit some of its weaknesses. These maybe taken for granted, but it is worth keeping them in mind when designing a protocol to be both reliable, efficient and useful.

2.5 Network Structure

2.5.1 Size

The potential userbase of a conferencing system using the internet should not be underestimated. It is therefore desirable that the conferencing system should be as distributed as possible, and as little state information kept as possible. If the IRC network is

taken as a guide, with 800 users on 140 servers in some 200 channels, the server was using over 1MB of memory. Due to the nature of conferencing and the server being run as a daemon, this memory was hardly ever swapped out. For this reason, servers should aim to only be authoritive about required users, channels and servers and keep up to date information on these.

There is also no requirement that a global conferencing system be built, although it is an ideal arena to show the strengths of the network. It also goes without saying that it shows up a lot of its weaknesses too.

Any protocol which is developed should operate equally well and efficiently on both a large scale network and on a small scale network.

3.0 Usage

If past usage is any guide, then a network based conferencing system will be largely used by mostly students. This is not as unreasonable as it may sound since students and student accounts easily form the largest body on the internet. To encourage staff or other adults into this field, it might be prudent to reduce the amount of noise and interfearance a bored student (or staff member!) can generate.

Realtime conferencing via computer networks is, however, a very attractive toy to many students. It puts them in touch with the world at no extra charge to them. They are able to construct their own character and mask or hide their real self. This is a field which has already been researched and is an interesting topic to pursue.

4.0 Setting it up

4.1 Installation

The installation and setup of most network utilities/servers is not something that is commonly discussed. It is, however, a point worth considering here after observations made on the setup and installation of systems such as IRC. If the setup is too easy and requires little work, it is not unreasonable to expect students to "install" it in their own accounts to provide themselves and friends with this service. There is little that can really be done about this except to force servers to listen and connect only to a certain priveledged port(s). This need, however, requires root intervention or aid and it is doubtful whether a service such as this should require such steps.

This problem is not often encountered with other network services since they either require large amounts of disk space to be done properly (news) or require the co-operation of other servers before they work in a full serving role (DNS and use of name servers is a good example of this). Of the two, the latter is a good solution if it can be implemented fairly and well.

4.2 Controlling growth

Is it possible to reasonably control the growth and connectivity of a large realtime conferencing network? Should it be compared to other facilities such as USENet which is commonly available and very widespread with no real central control over who gets news?

5.0 Finding the *right* protocol

This section deals with points which are central issues when deciding upon a protocol. There are many points to consider when developing a realtime protocol which is going to provide a service to many users simultaneously.

5.1 Name for protocol

Although names such as IRC and ICB have been used in the past to describe the implementation provided, this document is aimed at stimulating a protocol which is much more general and useful than these. A better name would reflect this. Depending on what network it is implemented on, the Network Conferencing Protocol (NCP) or the Internet Conferencing Protocol (ICP) are two suitable names.

5.2 Responsibilities of conference servers

5.2.1 Message passing

A conferencing server should pass on all messages not destined for itself or its users to the destination as quickly and efficiently as possible. To this end, the server should not be required to do extensive parsing of the incoming message, but rather, look at the header and decide from there whether to send it on in the typical gateway/relay fashion or parse it and pass it to one or more of its users.

5.2.2 Who is on?

Any conference server should be able to supply (on request) a list of attached user(s). The attached user(s) should have the option of being able to say whether they wish to show up in such lists.

5.2.3 Who is who?

All servers should provide *some* method to identify any known user and supply details to the person making the query based on the search key given.

5.2.4 Conference security

Conference servers should not run in such a manner that they deliberately record the private conversation(s) of users which are relying on the server in some way. It might seem that encrypting the message before transmission to other servers in some way would solve this, but this is better left as an option which is implemented in clients and thus leaves it to the users to decide how secure they want their conference to be.

5.2.5 Error reporting

All errors that the server encounters in its running life should one way or another be reported to the operator(s) which are responsible for this. This may include sending messages if an operator is online or logging it to an error file.

5.2.6 Network Friendliness

It is quite easy for any network based application to "abuse" the network it is running on. Also in a relay situation, it is quite possible that a server will become bogged down trying to keep up with just one connection and reduces the performance on an overall scale to all users relying on it. It is therefore recommended that user connections be subject to some sort of monitoring and flood control to stop them dumping large amounts of spurious data and causing the server to slow down.

The server should also aim to maximise the packet size of all packets written out to the network. Not only does this make the packet/bytes statistics look nice, but also increases the efficiency of the server by reducing the time it spends in the system state waiting/doing IO operations such as read/write. The cost here is a fractional decrease in the real-time efficiency of the server.

5.2.7 To ASCII or not to ASCII

Given that most of the widely used Internet protocols such as SMTP, NNTP and FTP are all based on commands which are given via ASCII strings, there seems no reason why a conferencing protocol should be any different. The gains from going to binary are marginal and debugging/testing is not as easy as with ASCII. However, it is not

unreasonable for some part of the protocol to be done in binary.

5.2.8 Queries or messages to a server and replies

For implementation of server queries, it is quite acceptable to use ASCII messages which are made up of words. (Any string of characters which doesn't start with a number). Replies should be some sort of numeric. This is a follow on from from 5.2.7 where all of FTP, NNTP and SMTP work in this manner. By reserving numerics *just* for server replies, there can be no confusion about whether the message is going to or from a server.

5.3 Responsibilities of clients

This section discusses the obligations of clients which are connected to a conference server.

5.3.1 Providing accurate information

Expecting accurate information is foolish, it matters not for most of the internet, but those that we do wish to trace wont give such information. A client is expected to provide accurate and valid information to the server it connects to so that confusion about who is who is not a problem. Optionally, the server may decide to not trust the information from the client and use some authentication scheme that is open to it for such.

5.3.2 Client as servers

If a client is acting as a server and accepting direct connections from other clients, the client should provide information about users as discussed in 5.2.3. It is not necessary that a client be able to handle complex methods of communication such as channels and their advanced forms, but they should at least provide users with being able to send messages to other users.

An example of this type of program might be Xtv where one or more users can connect to another Xtv client program using Xtv clients.

In the case of X windows and perhaps in other areas, one it to ask the destination user to run a program in a similar manner to unix talk.

5.4 How complex should the protocol be?

5.4.1 User identification

When a user signs onto a system that has an implementation of a conferencing protocol, they are usually asked or given some sort of unique key by which they are later able to be referenced by. In a large system, it may be such that any key which has meaning to the user(s) will not be sufficient and that collisions will occur with such. It is therefore suggested that a server generate an identifier for each new user it has. This identifier must not only be unique in space, but also time. It is not reasonable for the user to ever have to be aware of what this identifier is, it should only be known by servers which *need* to know. A similar system to that used by NNTP/SMTP is a fair implementation of such a scheme.

5.4.2 Trees and cycles

Due to the structure of the network being cyclic or forming loops, it is quite natural to want to emulate this within the protocol that is available for users. This has several advantages over trees, mainly the average path between any 2 nodes being shorter. A cyclic structure also poses many problems in getting messages delivered and keeping the connected users and servers up to date. The main problem with using the tree model is that a break in one part of the tree needs to be communicated to all other parts of the tree to keep some sort of realism about it. The problem here is that such communications happen quite often and a lot of bandwidth is needlessly generated. By implementing a protocol which supports a cyclic graph of its connectivity, breakages are less damaging except when it is a leaf or branch that breaks off.

5.5 Protocol summary

It is not expected that any protocol that meets the above demands will be either easy to arrive at or easy to implement. Some of the above requirements may seem to be exotic, unnecessary or not worth the effort. After viewing previous conferencing programs and how they work, many short comings can be seen in taking shortcuts.

6.0 Security Considerations

Security issues are not discussed in this memo.

7.0 Author's Address

Darren Reed
4 Pateman Street
Watsonia, Victoria 3087
Australia

Email: avalon@coombs.anu.edu.au