Internet Broadcast Protocols

The use of a "broadcast" IP (IP) for broadcast-type applications

William W. Plummer
Bolt Beranek and Newman, Inc.
50 Moulton Street
Cambridge, Mass. 02173

7 March 77

A fundamental assumption of Internet protocols (not only TCP)
that the output link is to deliver a minimum size packet, albeit
However, there are many networks that have not made this

Senders should prevent TCP connections from being established

IV. Routing and Network Interoperation

Senders should not initiate TCP connections from being established

The top-level solution is a policy that allows or precludes the

Networks must also support network interfaces that enable

By permitting compound options and packet level

In the absence of a "broadcast" interface, other strategies for

TCP connections may be established at the end hosts

The bandwidth limitation to each multiplexor differs from

The two fundamental changes are: the new " broadcaster"

The two fundamental changes are: the new " broadcaster"
3. Distributor

Building a distributor involves writing a distribution server program which implements a layer of protocol not only above the subnet, but above TCP! This is highly desirable property since it separates the connection maintenance function from the broadcast function.

A distribution server operates as follows: It "listens" for a connection from anyone wishing to initiate a broadcast. Once this connection has become bound, the broadcast source supplies a distribution list (exact form and content will be described later). If all goes well, the distribution server will respond with the name of a specific input connection to itself to be used by the broadcasting process. The distribution server then establishes as many secondary connections as it sees fit -- some of which may go to other distributors in the same or different networks.

The process run by the distributor for each specific connection simply reads a message from the broadcast source over the input connection and sends it out over each of the output connections. As previously mentioned, some of the output connections will be directly to specific receivers, but others will be to secondary distributors. In order to prevent hang-ups, it is required that each output connection address a set of receivers which is more specific than the input distribution list. Thus it would be improper for a distribution server to accept an input list of "Net 3, all hosts" and open two output connections, one to "Net 3, Host 100" and the other to a distributor for the purpose of broadcasting to "Net 3, all hosts". This secondary distribution list is not more specific than the input list and results in "Net 3, Host 100" being covered twice, most likely resulting in a hang-up if only one process is receiving the broadcast.

4. Connection Closing and Errors

Connections out of a distributor may be closed at any time by the receiver. When all of the output connections have closed, the distributor will close its input connection. Retransmission timeouts are to be treated as connection closings if they address a specific receiving process. On the other hand a timeout on an output connection to a secondary distributor might be handled by attempting to reestablish communications via some different distributor, or if resources permit, to make specific output connections to each of the intended receivers.
5. Flow Control

Flow is governed by the window announced by the distributor to the broadcaster. This window is a reflection of the amount of buffer space available in the distribution server itself, which in turn is a function of the number of output connections and the amount of unacknowledged information of each of these. This automatically provides a safeguard against any particular output connection tying up buffer space in the distributor by telling the distributor that it is able to receive a vast amount of data and then being very slow at processing it -- this situation cannot arise because the distributor will not send anymore than it has received, and it will not have received much if its buffer space is committed to waiting for ACK for packets on slow output connections.

6. Resynchronization, etc.

Resynchronization, sending ARQs, letter size management, and segment reassembly are handled inside the distributor, transparently to the broadcaster. Only connection opening and closing, and data acknowledgement involve any communications back to the broadcaster.

7. Multiple Distributors and the Billboard

Any host should be able to run a distribution server and a potential broadcaster could select any one that he can contact. Which one(s) will be dictated by other, outside constraints such as privacy, accounting, etc. On the other hand a potential receiver may want to find out where a good, relatively local source of some type of broadcast information is -- say weather information, but not airline flight information. To handle this need, each distribution list will contain a title. Upon setting up an instance of the distribution function, the title from the distribution list will be copied into a server-wide table along with the name of a local listening connection which is intended to service new output connections.

In addition to the normal listening connection for new distribution lists, the distributor will maintain a listening connection for its billboard function. This will be on a well-known socket. A potential receiver looking for a specific kind of broadcast would do it by polling the distributors that it knows about, establishing a connection to the billboard function on each, and looking at what is listed to see if the desired information is being broadcast at that moment.
3. Distributor

Building a distributor involves writing a distribution server program which implements a layer of protocol not only above the subnet, but above TCP! This a is highly desirable property since it separates the connection maintenance function from the broadcast function.

A distribution server operates as follows: It "listens" for a connection from anyone wishing to initiate a broadcast. Once this connection has become bound, the broadcast source supplies a distribution list (exact form and content will be described later). If all goes well, the distribution server will respond with the name of a specific input connection to itself to be used by the broadcasting process. The distribution server then establishes as may secondary connections as it sees fit -- some of which may go to other distributors in the same or different networks.

The process run by the distributor for each specific connection simply reads a message from the broadcast source over the input connection and sends it out over each of the output connections. As previously mentioned, some of the output connections will be directly to specific receivers, but others will be to secondary distributors. In order to prevent hang-ups, it is required that each output connection address a set of receivers which is more specific than the input distribution list. Thus it would be improper for a distribution server to accept an input list of "Net 3, all hosts" and open two output connections, one to "Net 3, Host 100" and the other to a distributor for the purpose of broadcasting to "Net 3, all hosts". This secondary distribution list is not more specific than the input list and results in "Net 3, Host 100" being covered twice, most likely resulting in a hang-up if only one process is receiving the broadcast.

4. Connection Closing and Errors

Connections out of a distributor may be closed at any time by the receiver. When all of the output connections have closed, the distributor will close its input connection. Retransmission timeouts are to be treated as connection closings if they address a specific receiving process. On the other hand a timeout on an output connection to a secondary distributor might be handled by attempting to reestablish communications via some different distributor, or if resources permit, to make specific output connections to each of the intended receivers.
If so, the distributor (output) socket listed will be used to tap the information. Should the broadcast cease before the data connection is established, it is desirable to have the data connection attempt fail. However, it must also be the case that establishing the data connection does not pick up a newer instance of a distribution list. So, some simple additional protocol is needed to resolve this issue. This most likely will take the form of verifying the data connection attempt on the basis of a random key passed out by the billboard function.

8. Knowing Distributor Addresses

The problem of knowing the identity (Network, Host) where distribution servers are running is common to both potential receivers and potential broadcast senders. This problem is not peculiar to the broadcast mechanism; it is the same one as a user faces when he wants to use a FORTRAN system on some (i.e., any) host. A master service directory must be consulted. This could be a printed list like a telephone book or it could be implemented by a program, but it is not a problem to be solved by the broadcast mechanism.

9. Controlled Unreliability

Although it is felt that TCP is not well suited for the net speech type of service, there are a few considerations which might make this possible. First, the broadcast source can "clear the pipe" at any time by sending a TCP level interrupt (INT) which forces all receivers to flush their current receive buffers. The broadcaster would do this if the rate of delivery had dropped below some predetermined rate. Notice that if the rate had gotten slow due to some host in the tree of distributors and receivers having crashed, then the INT will not be acknowledged, the distributor before the failed host will suffer a retransmission failure on the INT, and will mark that output connection as closed. This has the overall effect of causing a momentary loss of data to many (if not all) receivers, but permanently removes the slow or crashed receiver from the distribution.

Another possibility is that acknowledgments could be faked either by the sources TCP or by distributors. The problems involved are that control bytes can never by covered by fake ACKs, and that window information must be carefully handled.