HOST GROUPS: A MULTICAST EXTENSION FOR DATAGRAM INTERNETWORKS (U) STANFORD UNIV CA DEPT OF COMPUTER SCIENCE D R CHERITON ET AL. JUL 85 STAN-CS-85-1058

UNCLASSIFIED N08039-83-K-0431
Host Groups: A Multicast Extension for Datagram Internetworks

by

David R. Cheriton and Stephen E. Deering

Department of Computer Science
Stanford University
Stanford, CA  94305
Host Groups: A Multicast Extension for Datagram Internetworks

David R. Cheriton and Stephen E. Deering

Computer Systems Laboratory
Department of Computer Science and Electrical Eng.
Stanford University

Defense Advanced Research Projects Agency

July 1985

Unlimited

Unlimited

datagram networks
distributed systems
broadcast
multicast
internetworking

The extensive use of local networks is beginning to drive requirements for internetwork facilities that connect these local networks. In particular, the availability of multicast addressing in many local networks and its use by sophisticated distributed applications motivates providing multicast across internetworks.

In this paper, we propose a model of service for multicast in an internetwork, describe how this service can be used, and describe aspects of its implementation.
including how it would fit into one existing internetwork architecture, namely the US DoD internet Architecture.
Host Groups: A Multicast Extension for Datagram Internetworks

David R. Cheriton
Stephen E. Deering

Computer Systems Laboratory
Stanford University

Abstract

The extensive use of local networks is beginning to drive requirements for internetwork facilities that connect these local networks. In particular, the availability of multicast addressing in many local networks and its use by sophisticated distributed applications motivates providing multicast across internetworks.

In this paper, we propose a model of service for multicast in an internetwork, describe how this service can be used, and describe aspects of its implementation, including how it would fit into one existing internetwork architecture, namely the US DoD Internet Architecture.

1. Introduction

Multicast is the transmission of a datagram packet to a set of zero or more destination hosts in a network or internetwork, with a single address specifying the set of destination hosts. For example, hosts A, B, C and D may be associated with multicast address X. On transmission, a packet with destination address X is delivered with datagram reliability to hosts A, B, C and D.

Multicast has two primary uses, namely distributed binding and multi-destination delivery. It is useful for binding when one or more of a set of hosts contain the desired object but particular host addresses are not known, only a multicast address. For example, in a distributed file system, all the file servers may be associated with one multicast address. To bind a file name to a particular server, a client sends a query packet containing the file name to the file server multicast address, which is delivered to all the file servers. The server that recognizes the file name then responds to the client, allowing subsequent interaction directly with that server host. This also illustrates the use of multicast for logical addressing: the multicast address for a group of hosts can denote function rather than location. One can similarly associate the group of time servers, name servers, computation servers and so on each with their own multicast address.

Multi-destination delivery is useful to several applications, including:

- distributed, replicated databases

- conferencing

- distributed parallel computation, including distributed gaming

Ideally, multicast transmission to a set of hosts is not more complicated or expensive for the sender than transmission to a single host. Similarly, multicast transmission should not be more expensive for the network than traversing the shortest path tree that connects the sending host to the hosts identified by the multicast address.

Multicast transmission to a set of hosts is properly distinguished from broadcast transmission to all hosts on a network or internetwork. Broadcast is not a generally useful facility since there are few reasons for communicating with all hosts. In fact, it is best viewed as an "accident of the technology" for broadcast networks in the same way that self-modifying programs are an accident of the technology for stored program machines; just because the technology provides it does not mean it is efficient or safe to use. A proper multicast facility allows efficient transmission to multiple hosts while avoiding unnecessary loading of the network and receiving hosts that arises with broadcast.

Multicast is now available in standard local networks. For example, the Ethernet provides 2^16 multicast addresses. Sending a packet to an Ethernet multicast address delivers it (with datagram reliability) to the set of hosts listening to that multicast address. A variety of local network applications and systems make use of this facility. For instance, the V distributed system uses network-level multicast for implementing efficient operations on groups of processes spanning multiple machines. Similar use is being made of replicated databases and other distributed applications. Providing multicast in the internetwork environment would allow full use of local network distributed applications to the internetwork, as well as making some existing internetwork applications more robust and portable (by, for example, removing wire-in lists of addresses, such as gateway addresses).

In current internetwork environments, an application logically requiring multicast must send individually addressed packets to each recipient. There are two problems with this approach. Firstly, requiring the sending host to know the specific addresses of all the recipients defeats its use as a binding mechanism. For example, a diskless workstation needs on host to determine the network address of a disk server and it is undesirable to "wire in" specific network addresses. With a multicast facility, the multicast address of the disk server (or name servers that holds the address of the disk server) can be well known, allowing the workstation to transmit its initial queries to this address.

Secondly, transmitting multiple copies of the same packet makes inefficient use of network bandwidth, gateway resources and sender resources. For instance, the same packet may repeatedly traverse the same network links and pass through the same gateways. Furthermore, the network level cannot recognize multi-destination delivery to take advantage of multicast facilities that the underlying network technologies may provide. For example, local area bus, ring, or token networks and even satellite-based wide-area networks can provide efficient multicast delivery directly. Besides using excessive communication resources, the use of multiple transmissions to effect multicast severely limits the amount of parallelism in transmission and...
processing that can be achieved compared to an integrated multicast facility.

In this paper, we describe a model of multicast service we call host groups and discuss aspects of implementing this service in a datagram internetwork environment. We argue that it is feasible to implement this facility in an internetwork as an extension of the existing "unicast" internetwork datagram model and mechanism.

We restrict ourselves to the communication environment of a datagram-based internetwork, like the IP or XNS internetwork architectures. In these architectures, all hosts employ a common internetwork datagram format and a common internetwork addressing convention to identify the sources and destinations of datagrams. On transmission, an internetwork datagram is delivered to its destination address with "best efforts" reliability, via the transmission services of the underlying networks and the relaying services of the gateways. This service host corresponds to OSI layer 3 or the network level in providing host-to-host delivery. Reliable delivery, including error handling and flow control, is handled by higher-level protocols that operate in terms of internetwork datagrams.

Figure 1 illustrates a heterogeneous collection of independent networks interconnected by hosts that serve as store-and-forward gateways typical of datagram internetworks.

![Diagram of a typical internetwork](image)

**Figure 1 A Typical Internetwork**

In Figure 1, a satellite network and a wide area, store-and-forward network connect several local area networks as well as individual hosts. The combination of broadcast and point-to-point technology plus the usual complications of different speeds, delay and maximum transmission unit make an efficient implementation of multicast a challenge.

The next section describes the host group model of multicast service. Section 3 describes the implementation strategy we propose. Section 4 describes how this extension fits into the current US Bolt Internetwork architecture and briefly touches on other internetwork architectures. Section 5 illustrates how this facility can be used by a variety of applications. Section 6 relates this model to other proposals. Finally, we conclude with remarks on the status of our experimental prototype implementation of host groups and our future directions for investigation.

2. The Host Group Model

In an internetwork designed in the host group model, each internetwork address identifies a host group. A host group is a set of zero or more hosts in one internetwork. When an internetwork packet is sent, it is delivered with "best efforts" datagram reliability to all members of the host group identified by the internetwork address in the packet destination field.

The sender need not be a member of the destination group. We refer to such a group as open, in contrast to a closed group where only members are allowed to send to the group. We chose to provide open groups because they are more flexible and more consistent as an extension of conventional unicasts models (even though they are harder to implement).

Dynamic management of group membership provides flexible binding of internetwork addresses to hosts. Hosts may join and leave groups over time. A host may also belong to more than one group at a time. Finally, a host may belong to no groups at times, during which that host is unreachable within the internetwork architecture. In fact, an internetwork host need not have an individual internetwork address at all. Some hosts may only be associated with multi-host group addresses. For instance, there may be no reason to contact an individual time server in the internetwork, so time servers would not require individual addresses. Similarly, a bank of shared processors may be another from the standpoint of clients and only acquire individual internetwork addresses while they are serving individual clients.

Internetwork addresses are dynamically allocated for transient groups, groups that often last only as long as the execution of a single distributed program. A range of host group identifiers is reserved for identifying permanent groups. One use of permanent host groups identifiers is for host groups with standard logical meanings such as "name server group", "boot server group", "internetwork monitor group", etc. Permanently assigned addresses are also used for conventional single-host addresses.

The host group model of internetwork generalizes the binding of internetwork addresses to internetwork hosts by allowing one address to bind to multiple hosts on multiple networks, more than one address to be bound (in part) to one host, and the binding of an address to hosts to be dynamic, i.e. possible to modify under application control. For performance reasons, the conventional case of single-member groups is handled specially as an optimization. A range of internetwork addresses are reserved for designating groups of at most one internetwork host, allowing the delivery mechanism to make appropriate optimizations. Moreover, if an internetwork address is statically bound to a host permanently attached through one network, a network identifier can be embedded as a subfield of its internetwork address in order to simplify gateway routing. As should be apparent, this special case corresponds to the unicast facility provided by several current datagram-based internetwork architectures, including IP and XNS. Thus, the host group model is a compatible extension of these architectures.

The following subsections provide further details of the model.

2.1 Host Group Management

Dynamic binding of internetwork addresses to hosts is managed by the following three operations available to higher-level protocols or applications:

1. **Join** - Adds an internetwork address to a host group.
2. **Leave** - Removes an internetwork address from a host group.
3. **Leave Group** - Removes an internetwork address from all groups.

Each of these operations can be specified with three types of arguments: <internetwork address, host group address, protocol port number>. Different protocols, such as Telnet or TCP, can specify arguments for each operation. For example, a Telnet application might specify arguments for JOIN operations to include the following:

- **JOIN** - 
  - Internetwork address
  - Host group address
  - Protocol port number

In reality, the internetwork address is bound to network interfaces or host access ports, not the host machine per se.

4. **In** this procedure call notation, the arguments for an operation are listed in parentheses after the operation name, and the returned values, if any, are listed after a "->" symbol.
CreateGroup (type)
  --> outcome, group-address, access-key

requests the creation of a new transient host group with the
invoking host as its only member. The type argument specifies
either a general group or a one-member-only group plus whether
the group is restricted or unrestricted. A restricted group restricts
membership based on the access-key. Only hosts presenting a
valid host access-key are allowed to join. All unrestricted host
groups have a null access-key. outcome indicates whether the
request is approved or denied. If it is approved, a new transient
host address is returned in group-address. access-key is the
protection key (or password) associated with the new group.
This should fail only if there are no free transient group
addresses.

JoinGroup (group-address, access-key)
  --> outcome

requests that the invoking host become a member of the
identified host group (permanent or transient). outcome
indicates whether the request is approved or denied. A request
may be denied if the access-key is invalid.

LeaveGroup (group-address)
  --> outcome

requests that the invoking host be dropped from membership in
the identified group (permanent or transient). outcome
indicates whether the request is approved or denied.

There is no operation to destroy a transient host group because
a transient host group is deemed to no longer exist when its
membership goes to zero.

Note that in conventional internetworks allocation and
binding of internetwork addresses is typically performed statically
by internetwork administrators.

2.2 Packet Transmission

Transmission of a packet in the host group model is controlled
by two parameters of scope, one being the destination
internetwork address and the other being the "distance" to the
members in the group. In particular,

Send (dest-address, source-address, data, distance)

transmits the specified data in an internetwork datagram to the
hosts in the host group specified by dest-address that are
within the specified distance. The destination address is thus
similar to conventional networks except that delivery may be to
multiple hosts; the distance parameter requires further discussion.

Distance may be measured in several ways, including number
of network hops, time to deliver and what might be called
administrative distance. Administrative distance refers to the
distance between the administrations of different networks.
For example, in a company the networks of the research group
and an advanced development group might be considered quite
close to each other, networks of the corporate management more
distant, and networks of other companies much more distant.
One may wish to restrict a query to members within one's own
administrative domain because servers outside that domain may
not be trusted. Similarly, error reporting outside of an
administrative domain may not be productive and may in fact be
confusing.

Besides limiting the scope of transmission, the distance
parameter can be used to control the scope of multicast as a
binding mechanism and to implement an expanding scope of
search for a desired service. For instance, to locate a name server
familiar with a given name, one might check with nearby name
servers and expand the distance (by incrementing the distance on
retransmission) to include more distant name servers until the
name is found.

To reach all members of a group, a sender specifies the
maximum value for the distance parameter. This maximum must
exceed the "diameter" of the internetwork.

The distance parameter can be viewed as an extension of the
time-to-live or hop count parameters that are used in several
internetwork architectures to prevent infinite routing cycles. In
those cases, the distance parameter basically ensures that the
delivery mechanism only expends a finite amount of work in
delivery and therefore discards a packet caught in a routing loop.
The distance parameter in the host group model refines this finite
bound into further gradations.

Rather than define specific semantics of the distance
parameter in the model, we see it having a refinement of the
semantics of the time-to-live or hop count parameters specific to
each internetwork architecture. However, in all cases, there is a
need for well-known boundaries values that coincide with
administrative domains. For instance, there is a need for a
distance value that corresponds to "not outside this local
network".

Packet reception is the same as conventional architectures.
That is,

Receive()
  --> dest-address, source-address, data

returns the next internetwork datagram that is, or has been,
received.

2.3 Delivery Requirements

We identify several requirements for the packet delivery
mechanism that are essential to host groups being a useful and
used facility.

Firstly, given the predominance of broadcast local-area
networks and the locality of communication to individual
networks, the delivery mechanism must be able to exploit the
hardware's capability for very efficient multicast within a single
local-area network.

Secondly, the delivery mechanism must scale in sophistication
to efficient delivery across the internetwork as internetworks
acquire high-speed wide-area communication links and high
performance gateways. The latter are being provided by the
introduction of high-speed satellite channels and long-haul fiber
optic links. The latter are made feasible by the falling cost of
memory and processing power plus the increasing importance in
controlling access to relatively unprotected local network
environments. A host group delivery mechanism must be able to
take advantage of these trends as they materialize.

Finally, the delivery mechanism must avoid "systematic
errors" in delivery to members of the host group. That is, a small
number of repeated transmissions must result in delivery to all
group members within the specified distance, unless a member is
disconnected or has failed. We refer to this property as coverage.
In general, most reliable protocols make this basic assumption for
unicast delivery. It is important to guarantee this assumption for
multicast as well or else applications using multicast may fail in
unexpected ways when coverage is not provided. For efficiency,
the multicast delivery mechanism should also avoid regularly
delivering multiple copies of a packet to individual hosts.

Failure notification is not viewed as an essential requirement
given the datagram semantics of delivery. However, a host group
extension of internetwork architectures such as II and XNS
should provide "hint"-level failure notification as the natural extension of their failure notification for unicast.

3. Implementation

In this section, we sketch a design for implementing the host group model in a datagram internetwork. This description of the design is given to further support the feasibility of the host group model as well as point out some of the problems yet to be addressed.

Implementation of host groups involves implementing a binding mechanism (binding internetwork addresses to zero or more hosts) and a packet delivery mechanism (delivering a packet to each host to which its destination address binds). This facility fits most naturally into the gateways of the internetwork and the switching nodes of the constituent point-to-point networks (as opposed to separate machines) because multicast binding and delivery is a natural extension of the unicast binding and delivery (i.e., routing plus store-and-forward). That is, a multicast packet is routed and transmitted to multiple destinations, rather than to a single destination.

A gateway in a host group internetwork is thus viewed as a "communication server", providing multicast delivery and host group management. The multicast delivery service is invoked implicitly by sending packets addressed to host groups, with unicast delivery as a special case. The group management service is invoked explicitly, using a request-response transaction protocol between the client hosts and the server gateways. In addition to the operations for creating transient host groups and adding and deleting host memberships in groups (Section 2.1), the gateway supports operations for administrative allocation of permanent group addresses, including static, single-host group addresses (i.e., unicast addresses).

In the following description, we start with a basic, simple implementation that provides coverage and then refine this mechanism with various optimizations to improve efficiency of delivery and group management.

3.1 Basic Implementation

A host group defines a network group, which is the set of networks containing current members of the host group. When a packet is sent to a host group, a copy is delivered to each network in the corresponding network group. Then, within each network, a copy is delivered to each host belonging to the group.

To support such multicast delivery, every internet gateway maintains the following data structures:
- **routing table**: conventional internetwork routing information, including the distance and direction to the nearest gateway on every network.
- **network membership table**: A set of records, one for every currently existing host group. The network membership record for a group lists the network group, i.e., the networks that contain members of the group.
- **local host membership table**: A set of records, one for each host group that has members on directly attached networks. Each host membership record indicates the local hosts that are members of the associated host group. For networks that support multicast or broadcast, the record may contain only the local network-specific multicast address used by the group plus a count of local members. Otherwise, local group members may be identified by a list of unicast addresses to be used in the software implementation of multicast within the network.

A host invokes the multicast delivery service by sending an internetwork datagram to an immediate neighbor gateway (i.e., a gateway that is directly attached to the same network as the sending host). Upon receiving a datagram from a directly attached network, a gateway looks up the network membership record corresponding to the destination address of the datagram. For each of the networks listed in the membership record, the gateway consults its routing table. If, according to the routing table, a member network is directly attached, the gateway transmits a copy of the datagram on that network, using the network-specific multicast address allocated for the group on that network. For a member network that is not directly attached and is within the distance constraint specified in the datagram, the gateway creates a copy of the datagram with an additional inter-gateway header identifying the destination network. This inter-gateway datagram is forwarded to the nearest gateway on the destination network, using conventional store-and-forward routing techniques. At the gateway, the destination network, the datagram is stripped of its inter-gateway header and transmitted to the group's multicast address on that network. Member networks that are beyond the datagram's distance constraint are ignored.

The network membership records and the network-specific multicast structures are updated in response to group management requests from hosts. A host sends a request to create, join, or leave a group to an immediate neighbor gateway. If the host requests creation of a group, a new network membership record is created by the serving gateway and distributed to all other gateways. If the host is the first on its network to join a group, or if the host is the last on its network to leave a group, the group's network membership record is updated in all gateways. The updates need not be performed atomically at all gateways, due to the datagram delivery semantics: hosts can tolerate misrouted and lost packets caused by temporary gateway inconsistencies, as long as the inconsistencies are resolved within normal host retransmission periods. In this respect, the network membership data is similar to the network reachability data maintained by conventional routing algorithms, and can be handled by similar mechanisms.

In many cases, a host joins a group that already has members on the same network, or leaves a group that has remaining members on the same network. This is then a local matter between the hosts and gateways on a single network: only the local host membership table needs to be updated to include or exclude the host.

This basic implementation strategy meets the delivery requirements stated at the end of Section 2. However, it is far from optimal, in terms of either delivery efficiency or group management overhead. One simple improvement is to recognize the important special case of static, one-member-only groups. This again corresponds to the conventional unicasting provided in (for example) IP and XNS. To this case, the internetwork address for the single-host group encodes within it the network of the one host so there is no need to maintain a separate group membership record for that group. Consequently, the number of group membership records in the gateways is greatly reduced. Also, delivery to these groups degenerates to conventional unicasting techniques such as currently used in IP and XNS implementations. Below, we discuss some further refinements to the basic implementation.

3.2 Multicast Routing Between Networks

Multicast routing among the internetwork gateways is similar to store-and-forward routing in a point-to-point network. The main difference is that the link between the nodes (gateways) can be a mixture of broadcast and unicast-type networks with widely different throughput and delay characteristics. In addition, packets are addressed to networks rather than hosts (at the gateway level).

We use the extended reverse path forwarding algorithm of Dalal and Metcalfe. Although originally designed for broadcast, it is a simple and efficient technique that can serve well for multicast delivery if network membership records in each gateway are augmented with information from neighbouring
gateways. This algorithm uses the source network identifier, rather than a destination network identifier to make routing decisions. Since the source address of a datagram is a general group address, it cannot be used to identify the source network of the datagram: the first gateway must add a header specifying the source network. This approach minimizes redundant transmissions when multiple destination networks are reachable across a common intergateway link, a problem with the basic implementation described earlier.

Note that we eliminated from consideration techniques that fail to deliver along the branches of the shortest delay tree rooted at the source, such as Wall's center-based forwarding because this compromises the meaning of the multicast distance parameter and detracts from multicast performance in general. We also rejected the approach of having a multicast packet carry more than one network identifier in its intergateway header to indicate multiple destination networks because the resulting variable length headers would cause buffering and fragmentation problems in the gateways.

3.3 Multicasting Within Networks

A simple optimisation within a network is to have the sender use the local multicast address of a host group for its initial transmission. This allows the local host group members to receive the transmission immediately along with the gateways (which must now "eavesdrop" on all multicasts transmissions). A gateway only forwards the datagram if the destination host group includes members on the gateway networks. This scheme reduces the cost to reach local group members to one packet transmission from two required in the basic implementation so transmission to local members is basically as efficient as the local multicast support provided by the network.

A similar opportunity for reducing packet traffic arises when a datagram must traverse a network to get from one gateway to another, and the network also holds members of the destination group. Again, use of a network-specific multicast address which includes member hosts plus gateways can achieve the desired effect. However, in this case, hosts must be prepared to accept datagrams that include an intergateway header or, alternatively, every datagram must include a spare field in its header for use by gateways in lieu of an additional intergateway header.

3.4 Distributing Membership Information

A refinement to host group membership maintenance is to store the host group membership record for a group only in those gateways that are directly connected to member networks. Information about other groups is cached in the gateway only while it is required to route to those other groups. When a gateway receives a datagram to be forwarded to a group for which it has no network membership record (which can only happen if the gateway is not directly connected to a member network), it takes the following action. The gateway assumes temporarily that the destination group has members on every network in the internetwork, except those directly attached to the sending gateway, and routes the datagram accordingly. In the intergateway header of the outgoing packet, the gateway sets a bit indicating that it wishes to receive a copy of the network membership record for the destination host group. When such a datagram reaches a gateway on a member network, that gateway sends a copy of the membership record back to the requesting gateway and clears the request bit in the datagram.

Copies of network membership records sent to gateways outside of a group's member networks are cached for use in subsequent transmissions by those gateways. That raises the danger of a stale cache entry leading to systematic delivery failures. To counter that problem, the intergateway header contains a field which is a hash value of the checksum of the membership record used to route the datagram. Gateways on member networks compare the checksum on incoming datagrams with their up-to-date records. If the checksums don't match, an up-to-date copy of the record is returned to the gateway with the bad record.

This caching strategy minimizes intergateway traffic for groups that are only used within one network or within the set of networks which members reside on. It cannot be used in cases where partial replication with caching also reduces the overhead for network traffic to disseminate updates and keep all copies consistent. Finally, it also reduces the space cost for data in large internetworks with large numbers of multiple host groups.

We have not addressed here the problem of maintaining up-to-date, consistent network membership records within the set of gateways connected to members of a group. This can be viewed as a distributed database problem which has been well studied in other contexts. The loose consistency requirements on network membership records suggest that the techniques used in Grapevine might be useful for this application.

4. Integration into the DoD Internet

To show how the host group model can be supported by straightforward extension of an existing internetwork architecture, we outline how it might fit into the US DoD Internet.

The current Internet provides unicast datagram delivery between hosts on a wide variety of networks, both local-area and wide-area, broadcast and point-to-point. An Internet address is a 32-bit value consisting of two subfields: a network number and a host-within-network number. Every Internet gateway maintains a routing table that specifies the distance and direction to every network in the Internet, relative to the gateway. Thus, given a datagram, a gateway can determine from the network number subfield of its destination address, where to send it next on the path towards its destination. When the datagram reaches a gateway into its destination network, that gateway maps the host-within-network number to a local network address for final delivery.

The existing architecture supports our model of static, one-member-only groups. We extend this architecture to support multiple host groups by reserving a single network number to identify all such groups. Each multicast host group is distinguished by a unique value in the host-within-network subfield of its Internet address. The Internet gateways are augmented with the data structures and procedures discussed in Section 3 to support internetwork multicast.

An IP datagram contains a "time to live" field which is decremented by the gateways once a second and on every network hop. If the time to live goes to zero before the datagram reaches its destination, the datagram is discarded. In the host group implementation, this field is used to limit the delivery distance of multicasts.

Other datagram internetwork architectures yield to similar extensions. For example, the Xerox Network Systems architecture is essentially identical to the DoD Internet with regard to address encoding (network, host-within-network) and conventions of routing tables. XNS datagrams contain a hop count field that can be used for multicast scope control.

The proposed ISO internetwork protocol provides the same style of internetwork datagram service as IP or XNS. The draft proposal for ISO internetwork addresses specifies a much more complex distribution of the network membership records which would be inappropriate for the enormous potential size of the ISO
5. Use of Multicast

A number of applications that can use multicast have been cited earlier in the paper, including distributed databases, conferencing, distributed computation and locating internetwork services. Rather than describe these applications in greater detail, we focus on some general issues that were identified in previous work. (This work dealt with the use of local network multicast in a distributed operating system to support the concept of interprocess group communication where process groups are distributed across host groups.)

A key issue is providing reliable communication as required by the application. Firstly, some applications, such as real-time conferencing, do not need reliable delivery, assuming the periodic "directed broadcast" because it is easy to implement within many networks. Secondly, binding applications, such as locating a name server, do not require delivery to all but simply a positive response from at least one host. Retransmission with possibly expanding scope of search until a response is received provides the required semantics.

As an aside, one might argue that the binding use is only really required to locate a name server. While true in theory, it may be simpler for some applications to locate other servers directly using this simple search protocol. Then they do not need to implement the protocol to lookup a name in the name server as well as this simple search protocol to locate the name server in the first place.

For applications requiring reliable delivery, there are basically two approaches. The most common approach is to place the onus for reliable delivery on the sender. Here, the sender knows the membership of a group and retransmits to the group until it has received acknowledgements from each group member. As an optimization, the sender can use unicast to retransmit to particular group members if the number of missing acknowledgements is relatively small compared to the cardinality of the host group.

The second approach places the onus on the receivers to implement reliable delivery, what we call publishing. It is so named because it mimics real world publishing. That is, information is to be sent to a group, the subscribers. A subscriber noticing a missing message or a new message not being received in the expected time interval requests it back from the publisher. A subscriber notices a missing message by a gap in the received messages or a new message not received in the expected time interval.

A family of reliable multicast protocols is specified by Chang and Mazenclah, which combines both techniques built on top of an unreliable broadcast or multicast network. They describe a protocol that guarantees not only that all group members receive all messages, but also that they all receive the messages in the same order, regardless of the number of senders. Furthermore, this strong level of reliability is achieved with only one acknowledgment per message in the normal case, no single point of failure, and survival in the face of multiple host failures and recoveries.

In other papers, Chang describes the use of this protocol to support a distributed, replicated database.

In general, the problem is not implementing reliable delivery for multicast delivery but choosing the right trade-off between cost, performance and reliability as required by the application.

6. Related Work

There is relatively little published work on the use or implementation of internetwork multicast.

Wall's thesis presents several mechanisms for performing efficient broadcast and multicast delivery in point-to-point networks. His results can be applied to providing multicast within point-to-point networks that are constituents of an internetwork, and to the problems of multicast routing to "network groups" of gateways.

Boogg, in his thesis, describes a number of distributed applications that are impossible or very awkward to support without the flexible binding nature of broadcast addressing. Although he recognizes that almost all of his applications would be best served by a multicast mechanism, he advocates the use of "directed broadcast" because it is easy to implement within many kinds of networks and can be extended across an internetwork, without placing any new burden on internetwork gateways. Fortunately, broadcasting has the undesirable side effect of delivering packets to more hosts than necessary, thus incurring overhead on uninvolved parties and possibly creating security problems. Furthermore, directed broadcasting supports simple communication with unknown destinations on directly connected networks only; for destinations on more distant networks, the sender must know their network numbers or perform a search using gateway routing tables.

Recent proposals by Mogul and Aguilai have addressed the issue of multi-destination delivery within the DoD Internet. Mogul proposes an implementation of Boggs's directed broadcast facility. Aguilari suggests allowing an IP datagram to carry additional destination addresses, which are used by the gateways to route the datagram to each recipient. Such a facility would alleviate some of the inefficiencies of sending individual datagrams to a group, but it would not be able to take advantage of local network multicast facilities. More seriously, Aguilari's scheme requires the sender to know the individual IP addresses of all members of the destination group and thus lacks the flexible binding nature of true multicast or broadcast.

Blaustein et al discuss a variety of protocols for reliable multicast delivery based on various internetwork characteristics (e.g., point-to-point or broadcast or both, clusters of fast networks joined by slower networks, degree of multicast support provided by the networks, etc.). As well as making a case for unreliable multicast services at the internetwork level, their work suggests ways of achieving efficient multicast among gateways in a heterogeneous internetwork.

7. Concluding Remarks

We have described a model of multicast communication for datagram-based internetworks. As an extension of existing internetwork architectures, it views unicast communication and timeto-live constraints as special cases of the more general form of communication arising with multicast. We have argued that this model is implementable in current and future internetworks and that it provides a powerful facility for a variety of applications. In some cases, it provides a facility that is required for certain applications to work in the internetwork environment. In other cases, it provides a more efficient, robust and possibly more elegant way of implementing existing internetwork applications.

We are currently implementing a prototype host group facility as an extension of IP. For practical reasons, this prototype implements all group management functions and multicast routing outside of Internet gateways, in special hosts called "multicast agents." The collection of multicast agents in effect provides a second gateway system on top of the existing Internet, for multicast purposes. The major costs of this separation are redundancy of routing tables between gateways and multicast agents and the increased delay and unreliability of extra hops in
the delivery path. Much of the routing information in the multicast agents must be "wired-in" because they do not have access to the gateways' routing tables. However, this rudimentary implementation provides an environment for evaluating the interface to the multicast service and for investigating group management and multicast routing protocols for eventual use in the gateways. It also serves as a testbed for porting multicast-based distributed applications to an internetwork from the V distributed operating system.

For now, we are restricting group membership to local networks that already have a broadcast or multicast capability, such as the Ethernet. We feel that, in the future, any network that is to support hosts other than just gateways must have a multicast addressing mode. Efficient implementation of multicast within point-to-point or virtual circuit networks deserves investigation.

A significant issue raised by the host group model is authentication and access control in internetworks. Gateways must control which hosts can create and join host groups, presumably making their decision based on the identity of the requestor (thus requiring authentication) and permissions (access control lists). This issue does not arise in conventional internetwork architectures because host addresses are administratively assigned with no notion of dynamic assignment and binding as provided by host groups. We believe that access control should be recognized as a proper and necessary function of gateways so as to protect the hosts of local networks from general internetwork activity. Thus, group access control can be subsumed as part of this more general mechanism, although more investigation of the general issue is called for.

On a philosophical point, there has been considerable reluctance to make open use of multicast on local networks because it was network-specific and not provided across internetworks. We were originally of that school. However, we recognized that our "hidden" uses of multicast in the V distributed system were essential unless we resorted to dramatically poorer solutions - wired-in addresses. We also recognized, as described in this paper, that an adequate multicast facility for internetworks was feasible. As a consequence, we now argue that multicast is an important and basic facility to provide in local networks and internetworks. Higher levels of communication, including applications, should feel free to make use of this powerful facility. Networks and internetworks lacking multicast should be regarded as deficient relative to the future (and present) requirements of sophisticated distributed applications and communication systems.

Acknowledgements

The original proposal for host groups was developed as a proposal for extending the US DoD Internet architecture. Work is continuing develop a specific Internet extension in conjunction with the DARPA Internet task force on end-to-end protocols, headed by Bob Braden. This task force has provided valuable input in the development of the host group model.
References


END

FILMED

10-85

DTIC