

# **A protocol for Reliable SSM – Draft**

*By Ramakrishnan Venkitaraman*  
*UT-Dallas (Email: [rxv024000@utdallas.edu](mailto:rxv024000@utdallas.edu))*  
February 2003

## **1. Introduction**

Most traditional applications in the internet such as web browsing and email employ unicasting in which a separate connection is established between the client and the sender. The sender handles the requests of each of the clients separately as each client is interested in a different type of service/data from the sender. In cases like these unicasting made sense and there was no waste of bandwidth.

The steady growth of the internet and the growth of a wide variety of applications that need “many to many” data exchange led to the concept of multicasting. Multicasting effectively handles cases in which there are multiple receivers and all of them need the same data and at the same time. Doing unicast to all the receiver’s results in the waste of available bandwidth in the network and computing power at the host and the intermediate nodes.

SSM is the area of multicasting in which we have a single sender and a set of receivers This is the initial draft of a paper which aims at developing a protocol for Reliable SSM (Source Specific Multicast). Section 2 deals gives some background details about Multicasting. Section 3 deals specifically with Source Specific Multicast. Section 3 addresses the issues and semantics of reliability. Section 5 gives some information about related work in this area. Section 6 has the timeline for this protocol development process and Section 7 is the conclusion. Section 8 lists the references that made this document possible.

## **2. Background**

### ***2.1 Need for Multicasting***

Multicasting enables the sender to transmit only a single stream of data that’s delivered to all the recipients irrespective of the number of recipients for that data. Figure1 represents the case in which a sender is unicasting copies of the same data to all its 90 receivers. In the case of Multicasting the sender is sending only one copy of data due to the multicast capability of the network all the recipients receive the data. As can be seen in for applications like these multicasting is certainly a better option when compared to unicasting.

Note that unicasting may be considered to be a special case of multicasting in which there is only one receiver and sender. Broadcasting can also be viewed as a special case of multicasting in which all the nodes in the network form the set of receivers.



*Registration:* Registration is enabled using the IGMP protocol. When a host wants to be a member of a multicast group it sends a IGMP report (defined in RFC 1112) encapsulated in an IP datagram to the LAN router. From there on this report traverses the network until it reaches the sender or a Rendezvous point (RP). Periodically the multicast enabled router sends and IGMP query report to each of the hosts that has registered with it to be part of a Multicast group. A host that receives this query message responds with a report or reports, each of which correspond to the multicast group that it wants to continue to be the member of. (A host does not need to send a report for a given host group if it has already seen a report for that group on its subnet thus eliminating network congestion locally.) If no IGMP reports are received from a given subnet, then no multicast traffic needs to be sent to that subnet.

*Routing:* IGMP specifies that the routers communicate with each other in order to exchange information with each other about neighboring routers. A single router called the designated router for each physical network holds the responsibility of constructing a spanning tree that connects the members of any given multicast group and in which there is only one route between any two routers. Multicast routers create new branches only when they are needed by copying the multicast datagram's as required. When a branch is no longer needed they prune it from the tree.

There are many algorithms and protocols that have been proposed for IP multicast and they can be broadly classified in to two parts.

- Dense mode protocols
- Sparse mode protocols

*Dense mode protocols:* They initially assume that all the nodes in the network are interested in the multicast and are part of the multicast group. They send out traffic to all the hosts until they are informed otherwise. They are suitable in circumstances in which there is enough network bandwidth and at least one node from each subnet is part of the multicast group. Examples include Distance Vector Multicast Routing Protocol (DVMRP), Multicast Open Shortest Path First (MOSPT) and Protocol Independent Multicast Dense Mode (PIM-DM).

*Sparse Mode Protocols:* They assume that few nodes in the network are interested in being the part of the multicast group and add branches only if there is an explicit request from the host that it wants to be the part of the Multicast group. They are more suited to the WAN than the dense mode protocols. Examples for sparse mode protocols include Core based trees and Protocol Independent Multicast Sparse mode (PIM-SM).

At the transport level for most of the multicast enabled networks we use UDP instead of TCP as the multicast sender will not be able to handle the acknowledgements from the potentially hundreds of hosts which are its clients for each of the packet it sends. This problem is called as “Acknowledgement implosion” at the sender.

The type of Multicasting that we have been describing so far is called the ASM which stands for “Any Source Multicast” (RFC1112). It supports both one to many and many to many form of communication. RFC1112 specifies that any datagram sent to the address G in the range of 224.0.0.0 to 239.255.255.255 is delivered to each of the upper layer protocol modules that has requested for the delivery of packets that comes to G.

### **3. Source Specific Multicast (SSM)**

SSM is formally defined as follows

“A datagram sent with source IP address S and destination IP address G in the SSM range is delivered to each host socket that has specifically requested delivery of datagrams sent by S to G, and only to those sockets.”

Where, using the terminology of [IGMPv3],

"socket" is an implementation-specific parameter used to distinguish among different requesting entities (e.g., programs or processes or communication end-points within a program or process) within the requesting host; the socket parameter of BSD Unix system calls is a specific example.

Any host may send a datagram to any SSM address, and delivery is provided according to the above semantics.

#### **3.1 Need for SSM:**

Many of the multicast applications share a common feature. That is there will be one sender and there will be a set of receivers receiving datagram's from it. In essence they are one to many. Examples of these applications in the internet include those in which there is a server providing some multimedia content (streaming and like) and a set of hosts listening to it amongst others

This led to the concept of SSM, which stands for Source Specific Multicast, which is specifically designed for one to many form of communication. IP addresses in the range of 232/8 (232.0.0.0 to 232.255.255.255) are currently designated as SSM addresses and are to be used by source specific protocols and applications.

*Note:* The SSM destination address 232.0.0.0 is reserved and hosts must not send to this address. The addresses in the range 232.0.0.1 to 232.0.0.255 are available for allocation by IANA (Internet Assigned Numbers Authority). It is also recommended that these addresses be assigned randomly.

SSM delivery semantics are provided for a datagram that is delivered to the SSM address. The semantics state that a datagram with a source IP address of S and a destination IP address of G is delivered to each of the upper layer protocols that has requested for the delivery of packets sent by S to G and only to those sockets. The network service that is

identified by the pair (S,G) is referred to as the “channel”. It’s important to note that SSM in contrast to the ASM semantics SSM provides one to many transfers only but like ASM the set of receivers is generally unknown to the sender.

### **3.2 Advantages of SSM**

Let S be the sender and let G be the multicast group to which the sender is sending the packets.

1. Since channels are identified by the (S, G) pair the same SSM destination address can be used by multiple senders and this also prevents the cross delivery of the traffic.
2. There is no need for the hosts to co-ordinate during the process of choosing the destination addresses as the channel is identified by the (S, G) pair.
3. The router protocols are made simple and the intermediate routers need not generally need to be responsible for shared trees and rendezvous points. The router mechanisms that are required to support SSM are generally considered to be a subset of what’s needed to support ASM. The shortest path tree mechanism of the PIM-SM protocol can be adapted to provide SSM semantics.

As has been described, the key distinguishing feature of the protocol is that the channel is identified and addressed by the combination of a unicast source address and a multicast destination address in the SSM range. A channel that has the address

S, G = (50.1.10.1, 234.1.2.3)

is different from

S, G = (50.1.10.2, 234.1.2.3)

since they have different source addresses though they happen to have the same destination address. In the case of ASM this distinguishing would not have been possible.

### **3.3 SSM Terminologies:**

The following table describes the terminologies that are used in SSM in comparison to the ones used for ASM.

Service Model	ASM	SSM
Network Abstraction	Group	Channel
Identifier	G	S,G
Receiver Operations	Join, leave	Subscribe, UnSubscribe

As can be seen from the table, we use the term Channel to refer to the service associated with the SSM address and a specific service is identified by the S,G pair.

Any host can send to the SSM destination address and similar to ASM in which any host can send to the host group. In the case of ASM the datagram's that are sent by any host S to G are delivered to all the members of the group G whereas in the case of SSM its delivered to hosts that have subscribed to the channel which is identified by (S,G).

The receiver operations that are supported in the case of ASM are denoted by join(G) and leave(G) whereas in the case of SSM the receiver operations are identified by Subscribe(S,G) and UnSubscribe(S,G).

## 4. Issues and Semantics of reliability

So far we have discussed the importance and semantics of multicasting (mainly SSM). Our next goal is to provide reliability on top of SSM. So lets now various issues that are concerned with reliability. Applications requirements for Reliable Multicast (RM) are as broad and as varied as the applications themselves are.

There are a set of issues that are to be taken into consideration when developing a reliable protocol. Some of them are.

1. Does the application need to know whether everyone received data?
2. Does the data reception characteristics for all receivers fall within some range?
3. Scalability requirements of the application.
4. Degree of reliability
5. Total vs. Partially ordered data
6. Timing constraints
7. Is the application part of the public Internet or Internet?
8. Does the network have a feed back path from receiver to the sender?
9. Will the acknowledgements be Multicast or Unicast?
10. Handling newly join's.
11. FEC (Forward Error Correction) based approaches?

Now let's see some details about the issues listed above.

### *1. Does the application need to know whether everyone received data?*

There are 2 levels of feedback that can be sent from the receiver to the sender. One is at the application level and at the packet level.

Application level confirmation gives information to the application about whether the data was received by the receiver or not and information about receivers progress.

Packet level confirmation is useful at the link transport level which helps in it deciding when can it release the buffers which are being used for storing packets whose delivery is yet to be confirmed.

There are applications in which the sender needs to always make sure that all the receivers got the information it's transmitting. Examples of these applications include "paid multicast sessions" in which applications pay to the sender to be part of the channel or group.

If the application needs to ensure that every body got the ADU (Application Data Unit) then getting individual acknowledgements from the receiver is one of the options. We can also use the concept of Acknowledgement aggregation but additional constraints will be imposed if the sender needs to exactly know which of the receivers got the information and who did not?

In essence there are many approaches using which the same functionality can be achieved like ACK based, NACK based amongst others. It should also be noted that for the same application different levels of confirmation can be used, one at the packet level and the other at the ADU level.

Note: NACK based approaches are the ones in which the receiver will send a feedback called the Negative Acknowledgement (NACK) to the sender only when it does not receive a packet that it was expecting from the sender.

## *2. Does the data reception characteristics for all receivers fall within some range?*

For some applications we need to make sure that all the receivers of the channel or the group receive the ADU's at the same time or the reception characteristics (time of receipt) must fall within some range or interval.

Examples include real time stock quotes that are being transmitted to a set of receivers. Such requirements are generally considered hard to satisfy unless we constrain the performance of the system.

## *3. Scalability requirements of the application.*

The scalability requirements of the application largely affect the set of options that are available to provide reliability on top of Multicasting. Solutions that are available to applications that need not scale may not be available to application where scalability is a real issue.

For example, consider the case of a system in which the receivers acknowledge each and every packet sent to by the sender. Such a scheme will give the sender a lot of information about the reception characteristics and may also be used for congestion control. But such a system will not be scalable as the sender will face the problem of acknowledgement implosion as the system scales.

Systems which need to scale use NACKS or the process of ACK aggregation in order to save the sender from the problems that are caused by Acknowledgement implosion. ACK aggregation is the process in which the each ACK that the sender receives corresponds to more than one receiver. This is achieved by having intermediate nodes or routers to collect a set of ACK's and then set an ACK which denotes that a set of receivers have received the packet. In a tree structure a parent node will send an ACK to its parent denoting that all its children have received the packets and have sent it the ACK's.

#### *4. Degree of reliability*

Must the application be Totally Reliable or Semi Reliable? In the former case, if any of the data unit is missing then none of the received portion of the ADU is useful. Examples of these include file transfer applications. In the later case undelivered packets affect the quality of reception but do not make the received ADU's useless. Examples of these applications include multimedia broadcast.

#### *5. Total vs. partially ordered data*

Some applications warrant that data be delivered to the receiver in the same order in which it was sent by the sender. Examples of these include the ones like real time multimedia broadcast.

On the other hand for some applications its sufficient if the data be reliably delivered to the receiver and the order is not important. Re-ordering can be done at the receiver end. Examples of these applications include file transfer applications.

#### *6. Timing constraints*

Some applications have timing constraints that are imposed on the transmission of data to the receivers. That is the data should be delivered to the receiver as fast as possible. If the packets do not arrive on time then they may not be of any use. Examples of these applications include Multimedia transmission having real time constraints or as a result of new data superseding old data.

#### *7. Is the application part of the public Internet or Intranet?*

In principle the internet is same as that of the Internet. In practice however since the intranet is under the same administration, it will allow for solutions to be configured more easily than in the case of the Internet.

In the case of the public Internet it is very unlikely that additional functionality or expense required to support any new protocol would be acceptable.

#### *8. Does the network have a feed back path from receiver to the sender?*

Reliability is normally achieved by transmitting a feedback from the receiver to the sender through a feed back path. In cases where these feedback paths don't exist the set of options for providing reliability is very limited. There are circumstances in which no feed back channel/path may exist. Example is a satellite link. In this case reliability is provided by using FEC in which case we use additional bits in the transmitted data that will aid in the process of reconstructing the lost information from the packets it has got in most of the cases.

#### *9. Will the acknowledgements be Multicast or Unicast?*

In some protocols the feedback sent from the receiver to the sender are also multicast. This scheme is very effective in the case of NACK's as a host can refrain from sending a NACK in case it has already seen a NACK from another host for the same packet. This results in NACK suppression and hence saves bandwidth and computing power.

But in cases in which an arbitrary receiver is not allowed to multicast to a group, only unicast feed back mechanisms will be supported.

#### *10. Handling new joins.*

Another issue to be addressed is how are we going to handle new joins? These refer to the cases in which we have an existing multicast session and a new receiver joins the session say in the middle of the session. Must the packets that have been exchanged so far be retransmitted to the new client? If so, how long to buffer? Where to buffer the packets to be transmitted in the case of new joins?

#### *11. FEC (Forward Error Correction) based approaches?*

Another question to be answered when developing a reliable protocol is “Are we going to use FEC based mechanisms to provide reliability?”

As we had already discussed, in some cases FEC based mechanisms are the only way of providing reliability due to the lack of a feed back path and situations in which other mechanisms are not applicable.

## **5. Related Work**

There are a few working groups that are working in the area of Reliable Multicasting and SSM. The links for these groups can be got at [www.ietf.org](http://www.ietf.org)

For SSM: <http://www.ietf.org/html.charters/ssm-charter.html>

For Reliable Multicasting: <http://www.ietf.org/html.charters/rmt-charter.html>

I have recently posted a question relating to the issues specific to reliable SSM in the SSM working group in an effort to explore more related work and to initiate a discussion in the area of Reliable SSM. It can be found in the link (as on February 20, 2003)

<http://www.ietf.org/mail-archive/working-groups/ssm/current/maillist.html>

## **6. Timeline:**

I plan to decide on the semantics of reliability and to decide on a new protocol specification for Reliable Multicasting by the end of March 2003 and then try to simulate or Implement the same in April 2003.

## **7. Conclusion:**

In this draft we have presented an overview of Multicasting in general and our main emphasis is on the SSM model. We have also presented the notion of reliability and made it clear that the notion of reliability depends on the application for which we are designing the protocol.

## **8. References:**

1. An overview of SSM (Internet Draft) Supratik Bhattacharyya et.al
2. Source Specific Multicast for IP (IETF Draft) by H. Holbrook and B. Cain.
3. Reliable multicast design space for Bulk Transfer by M. Handley et.al
4. Protocol Independent Multicast – Sparse Mode (PIM-SM) Protocol Specification by Bill Fenner et.al