INTERNET DRAFT
<draft-ietf-ipoib-ip-over-infiniband-09.txt>
Expiration Date: July, 2005

H.K. Jerry Chu
Sun Microsystems
V. Kashyap
IBM
January, 2005

Transmission of IP over InfiniBand

Status of this memo

Abstract

This document specifies a method for encapsulating and transmitting IPv4/IPv6 and Address Resolution Protocol (ARP) packets over InfiniBand (IB). It describes the link layer address to be used when resolving the IP addresses in "IP over InfiniBand (IPoIB)" subnets. The document also describes the mapping from IP multicast addresses to InfiniBand multicast addresses. Additionally this document defines the set up and configuration of IPoIB links.

Table of Contents

1.0	Introduction
2.0	IP over UD Mode
3.0	InfiniBand Datalink
4.0	Multicast Mapping
4.1	Broadcast-GID Parameters
5.0	Setting Up an IPoIB Link
6.0	Frame Format
7.0	Maximum Transmission Unit
8.0	IPv6 Stateless Autoconfiguration
8.1	IPv6 Link Local Address
9.0	Address Mapping - Unicast
9.1	Link Information
9.1.1	Link Layer Address/Hardware Address
9.1.2	Auxiliary Link Information
9.2	Address Resolution in IPv4 Subnets
9.3	Address Resolution in IPv6 Subnets
9.4	Cautionary Note on QPN Caching
10.0	Sending and Receiving IP Multicast Packets
11.0	IP Multicast Routing
12.0	New Types of Vulnerability in IB Multicast
13.0	Security Considerations
14.0	IANA Considerations
15.0	Acknowledgments
16.0	References
17.0	Author's Addresses

1.0 Introduction

The InfiniBand specification [IBTA] can be found at www.infinibandta.org. The document [IPoIB_ARCH] provides a short overview of InfiniBand architecture (IBA) along with considerations for specifying IP over InfiniBand networks.

IBA defines multiple modes of transport over which IP may be implemented. The unreliable datagram (UD) transport mode best matches the needs of IP and the need for universality as described in [IPoIB_ARCH].

This document specifies IPoIB over IB's UD mode. The implementation of IP subnets over IB's other transport mechanisms is out of scope of this document.

This document describes the necessary steps required in order to lay out an IP network on top of an IB network. It describes all the elements of an IPoIB link, how to configure its associated attributes, and how to set up basic broadcast and multicast services

for it.

It further describes IP address resolution and the encapsulation of IP and ARP packets in InfiniBand frame.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2.0 IP over UD Mode

The unreliable datagram mode of communication is supported by all IB elements be they IB routers, Host Channel Adapters (HCAs) or Target Channel Adapters (TCAs). In addition to being the only universal transmission method it supports multicasting, partitioning and a 32-bit CRC [IBTA]. Though multicasting support is optional in IB fabrics, IPoIB architecture requires the participating components to support it.

All IPoIB implementations MUST support IP over the UD transport mode of IBA.

3.0 InfiniBand Datalink

An IB subnet is formed by a network of IB nodes interconnected either directly or via IB switches. IB subnets may be connected using IB routers to form a fabric made of multiple IB subnets. Nodes residing in different IB subnets can communicate directly with one another through IB routers at the IB network layer. Multiple IP subnets may be overlaid over this IB network.

An IP subnet is configured over a communication facility or medium over which nodes can communicate at the "link" layer [IPV6]. E.g. an ethernet segment is a link formed by interconnected switches/hubs/bridges. The segment is therefore defined by the physical topology of the network. This is not the case with IPoIB. IPoIB subnets are built over an abstract "link". The link is defined by its members and common characteristics such as the P_Key, link MTU, and the Q_Key.

Any two ports using UD communication mode in an IB fabric can communicate only if they are in the same partition i.e. have the same P_Key and the same Q_Key [IPoIB_ARCH]. The link MTU provides a limit to the size of the payload that may be used. The packet transmission and routing within the IB fabric is also affected by additional parameters such as the traffic class (TClass), hop limit (HopLimit), service level (SL) and the flow label (FlowLabel) [IPoIB_ARCH]. The determination and use of these values for IPoIB

communication is described in the following sections.

4.0 Multicast Mapping

IB identifies multicast groups by the multicast Global Identifiers (MGIDs) which follow the same rules as IPv6 multicast addresses. Hence the MGIDs follow the same rules regarding the transient addresses and scope bits albeit in the context of the IB fabric. The resultant address therefore resembles IPv6 multicast addresses. The documents [IBTA, IPoIB_ARCH] give a detailed description of IB multicast.

The IPoIB multicast mapping is depicted in figure 1. The same mapping function is used for both IPv4 and IPv6 except for the IPoIB signature field.

Unless explicitly stated, all addresses and fields in the protocol headers in this document are stored in the network byte order.

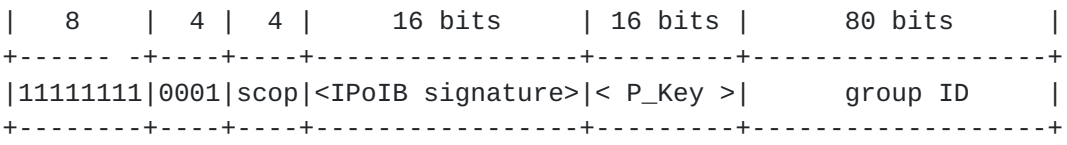

Figure 1

Since an MGID allocated for transporting IP multicast datagrams is considered only a transient link-layer multicast address [IPoIB_ARCH], all IB MGIDs allocated for IPoIB purpose MUST set T-flag to 1 [IBTA].

A special signature is embedded to identify the MGID for IPoIB use only. For IPv4 over IB, the signature MUST be "0x401B". For IPv6 over IB, the signature MUST be "0x601B".

The IP multicast address is used together with a given IPoIB link P_Key to form the MGID of the IB multicast group. For IPv6 the lower 80-bit of the group ID is used directly in the lower 80-bit of the MGID. For IPv4, the group ID is only 28-bit long, and is placed directly in the lower 28 bits of the MGID. The rest of the group ID bits in the MGID are filled with 0.

E.g. on an IPoIB link that is fully contained within a single IB subnet with a P_Key of 0x8000, the MGIDs for the all-router multicast group with group ID 2 [AARCH, IGMP2] are:

FF12:401B:8000::2, for IPv4 in compressed format, and
FF12:601B:8000::2, for IPv6 in compressed format.

A special case exists for the IPv4 limited broadcast address "255.255.255.255" [HOSTS]. The address SHALL be mapped to the "broadcast-GID", which is defined as follows:

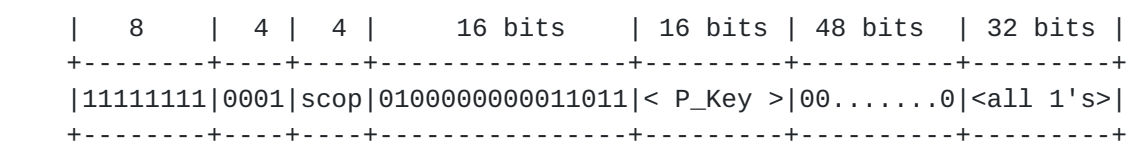

Figure 2

All MGIDs used in the IPoIB subnet MUST use the scop bits used in the broadcast GID.

4.1 Broadcast-GID Parameters

The broadcast-GID is set up with the following attributes:

1. P_Key
   A "Full Membership" P_Key (high-order bit is set to 1) MUST be used so that all members may communicate with one another.

2. Q_Key
   It is RECOMMENDED that a controlled Q_Key be used with the high order bit set. This is to prevent non-privileged software from fabricating and sending out bogus IP datagrams.

3. IB MTU
   The value assigned to the broadcast-GID must not be greater than any physical link MTU spanned by the IPoIB subnet.

The following attributes are required in multicast transmissions and also in unicast transmissions if an IPoIB link covers more than a single subnet.

4. Other parameters
   The selection of TClass, FlowLabel, and HopLimit values is implementation dependent. But it must take into account the topology of IB subnets comprising the IPoIB link in order to allow successful communication between any two nodes in the same IPoIB link.

   An SL also needs to be assigned to the broadcast-GID. This SL is used in all multicast communication in the subnet.

   The broadcast-GID's scope bits need to be set based on whether the IPoIB link is confined within an IB subnet or

the IPoIB link spans multiple IB subnets. A default of local-subnet scope i.e. 0x2 is RECOMMENDED. A node might determine the scope bits to use by interactively searching for a broadcast-GID of ever greater scope by first starting with the local-scope. Or, an implementation might include the scope bits as a configuration parameter.

5.0 Setting Up an IPoIB Link

The broadcast-GID, as defined in the previous section MUST be set up for an IPoIB subnet to be formed. Every IPoIB interface MUST "FullMember" join the IB multicast group defined by the broadcast-GID. This multicast group will henceforth be referred to as the broadcast group. The join operation returns the MTU, the Q_Key and other parameters associated with the broadcast group. The node then associates the parameters received as a result of the join operation with its IPoIB interface. The broadcast group also serves to provide a link-layer broadcast service for protocols like ARP, net-directed, subnet-directed and all-subnets-directed broadcasts in IPv4 over IB networks.

The join operation is successful only if the Subnet Manager (SM) determines that the joining node can support the MTU registered with the broadcast group [IPoIB_ARCH] ensuring support for a common link MTU. The SM also ensures that all the nodes joining the broadcast-GID have paths to one another and can therefore send and receive unicast packets. It further ensures that all the nodes do indeed form a multicast tree that allows packets sent from any member to be replicated to every other member. Thus the IPoIB link is formed by the IPoIB nodes joining the broadcast group. There is no physical demarcation of the IPoIB link other than that determined by the broadcast group membership.

The P_Key is a configuration parameter that must be known before the broadcast-GID can be formed. For a node to join a partition, one of its ports must be assigned the relevant P_Key by the SM [IPoIB_ARCH].

The method creation of the broadcast group and the assignment/choice of its parameters are up to the implementation and/or the administrator of the IPoIB subnet. The broadcast group may be created by the first IPoIB node to be initialized or it can be created administratively before the IPoIB subnet is set up. It is RECOMMENDED that the creation and deletion of the broadcast group is under administrative control.

InfiniBand multicast management which includes the creation, joining and leaving of IB multicast groups by IB nodes is described in [IPoIB_ARCH].

6.0 Frame Format

All IP and ARP datagrams transported over InfiniBand are prefixed by a 4-octet encapsulation header as illustrated below.

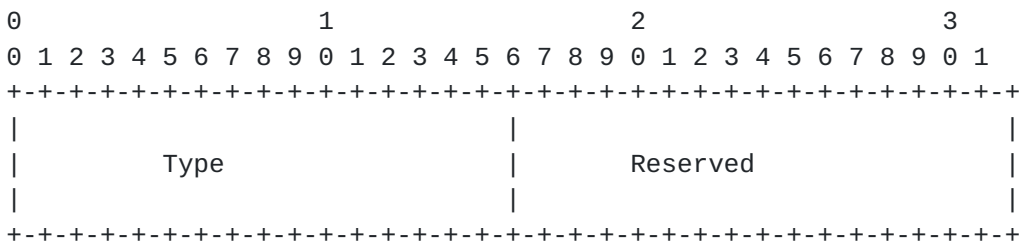

Figure 3

The "Reserved" field MUST be set to zero on send and ignored on receive unless specified differently in a future document.

The "Type" field SHALL indicate the encapsulated protocol as per the following table.

Type	Protocol
0x800	IPv4
0x806	ARP
0x8035	RARP
0x86DD	IPv6

Table 1

These values are taken from the "ETHER TYPE" numbers assigned by Internet Assigned Numbers Authority (IANA). Other network protocols, identified by different values of "ETHER TYPE", may use the encapsulation format defined herein but such use is outside of the

scope of this document.

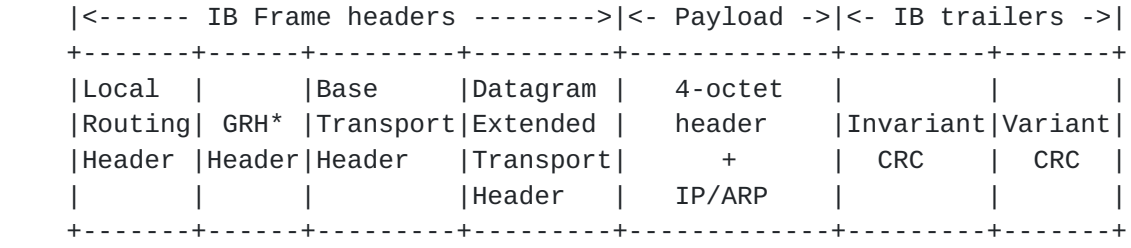

Figure 4

Figure 4 depicts the IB frame encapsulating an IP/ARP datagram. The InfiniBand specification requires the use of Global Routing Header (GRH) [IPoIB_ARCH] when multicasting or when an InfiniBand packet traverses from one IB subnet to another through an IB router. Its use is optional when used for unicast transmission between nodes within an IB subnet. The IPoIB implementation MUST be able to handle packets received with or without the use of GRH.

7.0 Maximum Transmission Unit

IB MTU:
The IB components i.e. IB links, switches, Channel Adapters (CAs), and IB routers, may support maximum payloads of : 256, 512, 1024, 2048 or 4096 octets. The maximum IB payload supported by the IB components in any IB path is the IB MTU for the path.

IPoIB-Link MTU:
The IPoIB-link MTU is the MTU value associated with the broadcast group. The IPoIB-link MTU can be set to any value up to the smallest IB MTU supported by the IB components comprising the IPoIB link.

In order to reduce problems with fragmentation and path-MTU discovery, this document requires that all IPoIB implementations support an MTU of 2044 octets i.e. a 2048 octet IPoIB-link MTU minus the 4 octet encapsulation overhead. Larger and smaller MTUs MAY be supported subject to other existing MTU requirements [IPV6], but the default configuration must support an MTU of 2044 octets.

8.0 IPv6 Stateless Autoconfiguration

IB architecture associates an EUI-64 identifier termed the GUID (Globally Unique Identifier) [IPoIB_ARCH, IBTA] with each port. The Local Identifier (LID) is unique within an IB subnet only.

The interface identifier may be chosen from:

1) The EUI-64 compliant GUID assigned by the manufacturer.

2) If the IPoIB subnet is fully contained within an IB subnet any of the unique 16-bit LIDs of the port associated with the IPoIB interface.

The LID values of a port may change after a reboot/power-cycle of the IB node. Therefore, if a persistent value is desired, it would be prudent to not use the LID to form the interface identifier.

On the other hand, the LID provides an identifier that can be used to create a more anonymous IPv6 address since the LID is not globally unique and is subject to change over time.

It is RECOMMENDED that the link-local address be constructed from the port's EUI-64 identifier as given below.

[AARCH] requires the interface identifier be created in the "Modified EUI-64" format when derived from an EUI-64 identifier. [IBTA] is unclear if the GUID should use IEEE EUI-64 format or the "Modified EUI-64" format. Therefore, when creating an interface identifier from the GUID an implementation MUST do the following:

=> Determine if the GUID is a modified EUI-64 identifier ("u" bit is toggled) as defined by [AARCH]

=> If the GUID is a modified EUI-64 identifier then the "u" bit MUST NOT be toggled when creating the interface identifier

=> If the GUID is an unmodified EUI-64 identifier then the "u" bit MUST be toggled in compliance with [AARCH]

8.1 IPv6 Link Local Address

The IPv6 link local address for an IPoIB interface are formed as described in [AARCH] using the Interface Identifier as described in the previous section.

9.0 Address Mapping - Unicast

Address resolution in IPv4 subnets is accomplished through Address Resolution protocol (ARP) [ARP]. It is accomplished in IPv6 subnets using the Neighbor Discovery protocol [DISC].

9.1 Link Information

An InfiniBand packet over the UD mode includes multiple headers such as the LRH (local route header), GRH (global route header), BTH (base transport header), DETH (datagram extended header) as depicted in figure 4 and specified in the InfiniBand architecture [IBTA]. All these headers comprise the link-layer in an IPoIB link.

The parameters needed in these IBA headers constitute the link-layer information that needs to be determined before an IP packet may be transmitted across the IPoIB link.

The parameters that need to be determined are:

a) LID

The LID is always needed. A packet always includes the LRH that is targeted at the remote node's LID, or an IB router's LID to get to the remote node in another IB subnet.

b) Global Identifier (GID)

The GID is not needed when exchanging information within an IB subnet though it may be included in any packet. It is an absolute necessity when transmitting across the IB subnet since the IB routers use the GID to correctly forward the packets. The source and destination GIDs are fields included in the GRH.

The GID, if formed using the GUID, can be used to unambiguously identify an endpoint.

c) Queue Pair Number (QPN)

Every unicast UD communication is always directed to a particular queue pair (QP) at the peer.

d) Q_Key

A Q_Key is associated with each unreliable datagram QPN. The received packets must contain a Q_Key that matches the QP's Q_Key to be accepted.

e) P_Key

A successful communication between two IB nodes using UD mode can occur only if the two nodes have compatible P_Keys. This is referred to as being in the same partition [IBTA].

f) SL

Every IBA packet contains an SL value. A path in IBA is defined by the three-tuple (source LID, destination LID, SL). The SL in turns is mapped to a virtual lane (VL) at every CA, switch that sends/forwards the packet [IPoIB_ARCH]. Multiple SLs may be used between two endpoints to provide for load-balancing, SLs may be used for providing a QoS infrastructure, or may be used to avoid deadlocks in the IBA fabric.

Another auxiliary piece of information, not included in the IBA headers, is :

g) Path rate

IBA defines multiple link speeds. A higher speed transmitter can swamp switches and the CAs. To avoid such congestion every source transmitting at greater than 1x speeds is required to determine the "path rate" before the data may be transmitted [IBTA].

9.1.1 Link Layer Address/Hardware Address

Though the list of information required for a successful transmittal of an IPoIB packet is large, not all the information need be determined during the IP address resolution process.

The 20-octet IPoIB link-layer address used in the source/target link-layer address option in IPv6 and the "hardware address" in IPv4/ARP has the same format.

The format is as described below:

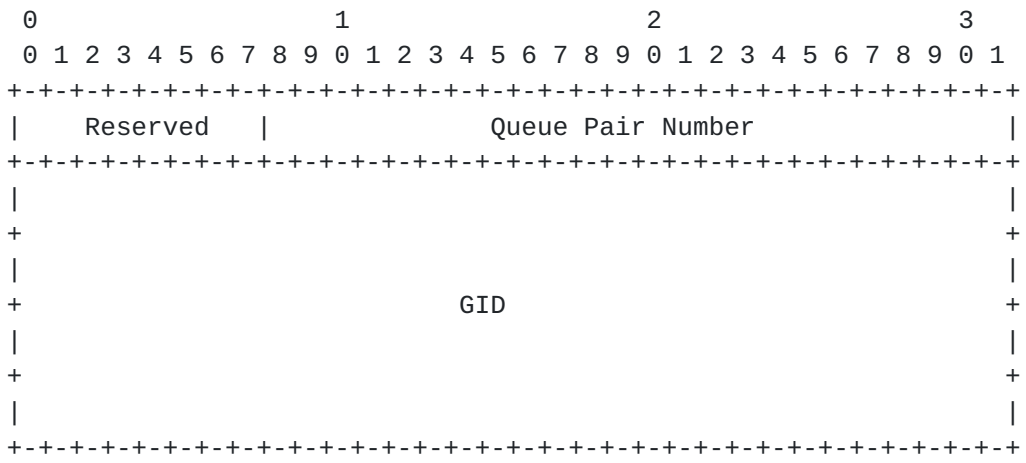

Figure 5

a) Reserved Flags

These 8 bits are reserved for future use. These bits MUST be set to zero on send and ignored on receive unless specified differently in a future document.

b) QPN

Every unicast communication in IB architecture is directed to a specific QP [IBTA]. This QP number is included in the link description. All IP communication to the relevant IPoIB interface MUST be directed to this QPN. In the case of IPv4 subnets the address resolution protocol (ARP) reply packets are also directed to the same QPN.

The choice of the QPN for IP/ARP communication is up to the implementation.

c) GID

This is one of the GIDs of the port associated with the IPoIB interface [IBTA]. IB associates multiple GIDs with a port. It is RECOMMENDED that the GID formed by the combination of the IB subnet prefix and the port's "Port GUID" [IBTA] be included in the link-layer/hardware address.

9.1.2 Auxiliary Link Information

The rest of the parameters are determined as follows:

a) LID

The method of determining the peer's LID is not defined in this document. It is up to the implementation to use any of the IBA approved methods to determine the destination LID. One such method is to use the GID determined during the address resolution, to retrieve the associated LID from the IB routing infrastructure or the Subnet Administrator (SA).

It is the responsibility of the administrator to ensure that the IB subnet(s) have unicast connectivity between the IPoIB nodes. The GID exchanged between two endpoints in a multicast message (ARP/ND) does not guarantee the existence of a unicast path between the two.

There may be multiple LIDs, and hence paths, between the endpoints. The criteria for selection of the LIDs are beyond the scope of this document.

b) Q_Key

The Q_Key received on joining the broadcast group MUST be used for all IPoIB communication over the particular IPoIB link.

c) P_Key

The P_Key to be used in the IP subnet is not discovered but is a configuration parameter.

d) SL

The method of determining the SL is not defined in this document. The SL is determined by any of the IBA approved methods.

e) Path rate

The implementation must leverage IB methods to determine the path rate as required.

9.2 Address Resolution in IPv4 Subnets

The ARP packet header is as defined in [ARP]. The hardware type is set to 32 (decimal) as specified by IANA. The rest of the fields are used as per [ARP].

```
16 bits: hardware type
16 bits: protocol
 8 bits: length of hardware address
 8 bits: length of protocol address
16 bits: ARP operation
```

The remaining fields in the packet hold the sender/target hardware and protocol addresses.

```
[ sender hardware address ]
[ sender protocol address ]
[ target hardware address ]
[ target protocol address ]
```

The hardware address included in the ARP packet will be as specified in section 9.1.1 and depicted in figure 5.

The length of the hardware address used in ARP packet header therefore is 20.

9.3 Address Resolution in IPv6 Subnets

The Source/Target Link-layer address option is used in Router Solicit, Router advertisements, Redirect, Neighbor Solicitation and Neighbor Advertisement messages when such messages are transmitted on InfiniBand networks.

The source/target address option is specified as follows:

Type:

Source Link-layer address	1
Target Link-layer address	2

Length: 3

Link-layer address:

The link-layer address is as specified in section 9.1.1 and depicted in figure 5.

[DISC] specifies the length of source/target option in number of 8-octets as indicated by a length of '3' above. Since the IPoIB link-layer address is only 20-octet long, two octets of zero MUST be prepended to fill the total option length of 24 octets.

9.4 Cautionary Note on QPN Caching

The link-layer address for IPoIB includes the QPN which might not be constant across reboots or even across network interface resets. Cached QPN entries, such as in static ARP entries or in RARP servers will only work if the implementation(s) using these options ensure that the QPN associated with an interface is invariant across reboots/network resets.

It is RECOMMENDED that implementations revalidate ARP caches periodically due to the aforementioned QPN induced volatility of IPoIB link-layer addresses.

10.0 Sending and Receiving IP Multicast Packets

Multicast in InfiniBand differs in a number of ways from multicast in Ethernet. This adds some complexity to an IPoIB implementation when supporting IP multicast over IB.

A) An IB multicast group must be explicitly created through the SA before it can be used.

This implies that in order to send a packet destined for an IP multicast address, the IPoIB implementation must check with the SA on the outbound link first for a "MCMemberRecord" that matches the MGID. If one does exist, the MLID associated with the multicast group is used as the DLID for the packet. Otherwise, it implies no member exists on the local link. If the scope of the IP multicast group is beyond link-local, the packet must be sent to the on-link routers through the use of the all-router multicast group or the broadcast group. This is to allow local routers to forward the packet to multicast listeners on remote networks. The all-router multicast group is preferred over the broadcast group for better efficiency. If the all-router multicast group does not exist, the sender can assume that there are no routers on the local link; hence the packet can be safely dropped.

B) A multicast sender must join the target multicast group before outgoing multicast messages from it can be successfully routed. The "SendOnlyNonMember" join is different from the regular "FullMember" join in two aspects. First, both types of joins enable multicast packets to be routed FROM the local port, but only the "FullMember" join causes multicast packets to be routed TO the port. Second, the sender port of a "SendOnlyNonMember" join will not be counted as a member of the multicast group for purposes of group creation and deletion.

The following code snippet demonstrates the steps in a typical implementation when processing an egress multicast packet.

```
if the egress port is already a "SendOnlyNonMember", or a
"FullMember"
    => send the packet

else if the target multicast group exists
    => do "SendOnlyNonMember" join
    => send the packet

else if scope > link-local AND the all-router multicast group exists
    => send the packet to all routers

else
    => drop the packet
```

Implementations should cache the information about the existence of an IB multicast group, its MLID and other attributes. This is to avoid expensive SA calls on every outgoing multicast packet. Senders MUST subscribe to the multicast group create and delete traps in order to monitor the status of specific IB multicast groups. E.g., multicast packets directed to the all-router multicast

group due to a lack of listener on the local subnet must be forwarded to the right multicast group if the group is created later. This happens when a listener shows up on the local subnet.

A node joining an IP multicast group must first construct a MGID according to the rule described in section 4 above. Once the correct MGID is calculated, the node must call the SA of the outbound link to attempt a "FullMember" join of the IB multicast group corresponding to the MGID. If the IB multicast group doesn't already exist, one must be created first with the IPoIB link MTU. The MGID MUST use the same P_Key, Q_Key, SL, MTU and HopLimit as those used in the broadcast-GID. For the rest of attributes too, the values used in the broadcast-GID SHOULD be used.

The join request will cause the local port to be added to the multicast group. It also enables the SM to program IB switches and routers with the new multicast information to ensure the correct forwarding of multicast packets for the group.

When a node leaves an IP multicast group, it SHOULD make a "FullMember" leave request to the SA. This gives SM an opportunity to update relevant forwarding information, to delete an IB multicast group if the local port is the last FullMember to leave, and free up the MLID allocated for it. The specific algorithm is implementation-dependent, and is out of the scope of this document.

Note that for an IPoIB link that spans more than one IB subnet connected by IB routers, an adequate multicast forwarding support at the IB level is required for multicast packets to reach listeners on a remote IB subnet. The specific mechanism for this is beyond the scope of IPoIB.

11.0 IP Multicast Routing

IP multicast routing requires each interface over which the router is operating to be configured to listen to all link-layer multicast addresses generated by IP. For an Ethernet interface this is often achieved by turning on the promiscuous multicast mode on the interface.

IBA does not provide any hardware support for promiscuous multicast mode. Fortunately a promiscuous multicast mode can be emulated in the software running on a router through the following steps.

A) Obtain a list of all active IB multicast groups from the local SA.

B) Make a "NonMember" join request to the SA for every group that

has a signature in its MGID matching the one for either IPv4 or IPv6.

C) Subscribe to the IB multicast group creation events using a wildcarded MGID so that the router can "NonMember" join all IB multicast groups created subsequently for IPv4 or IPv6.

The "NonMember" join has the same effect as a "FullMember" join except that the former will not be counted as a member of the multicast group for purposes of group creation or deletion. That is, when the last "FullMember" leaves a multicast group, the group can be safely deleted by the SA without concerning any "NonMember" routers.

12.0 New Types of Vulnerability in IB Multicast

Many IB multicast functions are subject to failures due to a number of possible resource constraints. These include the creation of IB multicast groups, the join calls ("SendOnlyNonMember", "FullMember", and "NonMember"), and the attaching of a QP to a multicast group.

In general, the occurrence of these failure conditions is highly implementation dependent, and is believed to be rare. Usually a failed multicast operation at the IB level can be propagated back to the IP level, causing the original operation to fail, and the initiator of the operation to be notified. But some IB multicast functions are not tied to any foreground operation, making their failures hard to detect. E.g., if an IP multicast router attempts to "NonMember" join a newly created multicast group in the local subnet, but the join call fails, packet forwarding for that particular multicast group will likely to fail silently, that is, without the attention of local multicast senders. This type of problems can add more vulnerability to the already unreliable IP multicast operations.

Implementations SHOULD log error messages upon any failure from an IB multicast operation. Network administrators should be aware of this vulnerability, and preserve enough multicast resources at the points where IP multicast will be used heavily. E.g., HCAs with ample multicast resources should be used at any IP multicast router.

13.0 Security Considerations

This document specifies IP transmission over a multicast network. Any network of this kind is vulnerable to a sender claiming another's identity and forging traffic or eavesdropping. It is the responsibility of the higher layers or applications to implement suitable counter-measures if this is a problem.

Successful transmission of IP packets depends on the correct set up of the IPoIB link , creation of the broadcast GID, creation of the QP and its attachment to the broadcast-GID, and the correct determination of various link parameters such as the LID, service level, path rate etc. These operations, many of which involve interactions with the SM/SA, MUST be protected by the underlying operating system. This is to prevent malicious, non- privileged software from hijacking important resources and configurations.

Controlled Q_Keys SHOULD be used in all transmissions. This is to prevent non-privileged software from fabricating IP datagrams.

14.0 IANA Considerations

To support ARP over InfiniBand, a value for the Address Resolution Parameter "Number Hardware Type (hrd)" is required. IANA has assigned the number "32" to indicate InfiniBand [IANA_ARP].

Proposed uses of the reserved bits in the frame format(Figure 3) and link layer address(Figure 5) MUST be published as RFCs. This document requires that the reserved bits be set to zero on send and ignored on receives.

15.0 Acknowledgments

The authors would like to thank Bruce Beukema, David Brean, Dan Cassiday, Aditya Dube, Yaron Haviv, Michael Krause, Thomas Narten, Erik Nordmark, Greg Pfister, Jim Pinkerton, Renato Recio, Kevin Reilly, Kanoj Sarcar, Satya Sharma, Madhu Talluri, and David L. Stevens for their suggestions and many clarifications on the IBA specification.

16.0 References

16.1 Normative

[AARCH] Hinden, R. and S. Deering "IP Version 6 Addressing Architecture", RFC 3513.

[ARP] Plummer D.C., "Ethernet Address Resolution Protocol", RFC 826.

[DISC] Narten, T., Nordmark, E. and W. Simpson, "Neighbor Discovery for IP Version 6 (IPv6)", RFC 2461.

[IANA] Internet Assigned Numbers Authority, www.iana.org

[IANA_ARP] www.iana.org/assignments/arp-parameters

[IBTA] InfiniBand Architecture Specification, www.infinibandta.org/specs

[IPoIB_ARCH] Kashyap V., "IP over InfiniBand (IPoIB) Architecture", draft-ietf-ipoib-architecture-04.txt.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119.

16.2 Informative

[HOSTS] Braden R., "Requirements for Internet Hosts -- Communication Layers", RFC 1122.

[IGMP2] Fenner W., "Internet Group Management Protocol, Version 2", RFC 2236.

[IP6MLD] Deering S., Fenner W., Haberman B., "Multicast Listener Discovery (MLD) for IPv6", RFC 2710.

[IPMULT] Deering S., "Host Extensions for IP Multicasting", RFC 1112.

[IPV6] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460.

17.0 Authors" Addresses

H.K. Jerry Chu

17 Network Circle, UMPK17-201
Menlo Park, CA 94025
USA
Phone: +1 650 786 5146
Email: jerry.chu@sun.com

Vivek Kashyap

15350, SW Koll Parkway
Beaverton, OR 97006
USA
Phone: +1 503 578 3422
Email: vivk@us.ibm.com

Full Copyright Statement

Intellectual Property Statement

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.