IP over InfiniBand: Connected Mode

Status of This Memo

Copyright Notice

Abstract

   This document specifies transmission of IPv4/IPv6 packets and address
   resolution over the connected modes of InfiniBand.

Table of Contents

1. Introduction

   The InfiniBand specification [IB_ARCH] can be found at
   www.infinibandta.org. The document [RFC4392] provides a short
   overview of InfiniBand architecture along with consideration for
   specifying IP over InfiniBand networks.

   The InfiniBand Architecture (IBA) defines multiple modes of
   transports. Of these the unreliable datagram (UD) transport method
   best matches the needs of IP. IP over InfiniBand (IPoIB) over UD is
   described in [RFC4391]. This document describes IP transmission over
   the connected modes of IBA.

   IBA defines two connected modes:

      1. Reliable Connected (RC)
      2. Unreliable Connected (UC)

   As is evident from the nomenclature, the two modes differ mainly in
   providing reliability of data delivery across the connection. This
   document applies equally to both the connected modes. IPoIB over
   these two modes is referred to as IPoIB-CM (connected mode) in this

document. For clarity, IPoIB over the unreliable datagram mode as
described in [RFC4391] is referred to as IPoIB-UD.

IBA requires that all Host Channel Adapters (HCAs) support the
reliable and unreliable connected modes [IB_ARCH]. It is optional
for Target Channel Adapters (TCAs) to support the connected modes.

The connected modes offer link MTUs of up to 2^31 octets in length.
Thus, the use of connected modes can offer significant benefits by
supporting reasonably large MTUs. The datagram modes of InfiniBand
Architecture (IBA) are limited to 4096 octets.

Reliability is also enhanced if the underlying feature of "automatic
path migration" supported by the connected modes is utilized.

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. IPoIB-connected Mode

IPoIB over connected mode is an OPTIONAL extension to IPoIB-UD.
Every IPoIB implementation MUST support [RFC4391] and MAY support the
extensions described in this document.

Therefore, IP encapsulation, default MTU, link-layer address format,
and the IPv6 stateless autoconfiguration mechanism apply to IPoIB-CM
exactly as described in [RFC4391].

2.1. Multicasting

The connected modes of IBA define a non-broadcast, multiple-access
network. The connected modes of IBA do not support multicasting
though every node can communicate with every other node if desired.

This requires that multicasting be emulated in some form by the
network. However, in the case of an InfiniBand network, instead of
an emulation, an unreliable datagram (UD) queue pair (QP) can be used
to support multicasting while the connected mode QP is used for
unicast traffic. Since every IPoIB implementation is required to
support the UD mode, every implementation supporting IPoIB-CM will be
able to utilize the pre-existing IPoIB-UD QP for all
broadcast/multicast communications. Multicast mapping, transmission,
and reception of multicast packets and multicast routing MUST use the
UD QP associated with the IPoIB interface.

2.2. Outline of Address Resolution

Every IPoIB-CM interface MUST have two sets of QPs associated with
it:

    1) One unreliable datagram QP
    2) One or more connected mode QPs

[RFC4391] describes the address resolution method to determine the
link address of the peer. This response is received on the UD QP
associated with the IPoIB interface.

2.3. Outline of Connection Setup

Once the link address of the remote node is known, an IB connection
must be set up between the nodes before any IP communication may
occur.

To make a connection, the sender must know the service-ID to use in
the request to make a connection [IB_ARCH]. It must also supply the
"connection mode" queue pair to the remote node. The peer replies
with its queue pair. Each IB connection is peer to peer and uses one
connected mode QP at each end.

Though the address resolution occurs at an individual IP address
level, the connection between the nodes is at the IB layer.
Therefore, every individual address resolution does not imply a new
connection between the peers.

3. Address Resolution

Address resolution queries are sent out on the "broadcast-GID"
(Broadcast-Group Identifier) over the UD QP associated with the IPoIB
interface [RFC4391]. A unicast reply is received on the UD QP.

3.1. Link-layer Address

IPoIB encapsulation [RFC4391] describes the link-layer address as
follows:

    <1 octet reserved>:QP: GID

This document extends the link-layer address as follows:

    <Flags>:QPN:GID

Flags:

    This is a single-octet field. The bits indicate the connected
    modes supported by the interface.

    Bit 0 specifies the support for the "reliable connected" (RC)
    mode. Bit 1 indicates the support for the "unreliable connected"
    (UC) mode. All other bits in the octet are reserved and MUST be
    set to 0 on transmits and ignored on receives. The format of the
    flags is as follows:

            +--+--+--+--+--+--+--+--+
            |RC|UC| 0| 0| 0| 0| 0| 0|
            +--+--+--+--+--+--+--+--+

    Both the RC and UC MAY be set at the same time if the interface
    supports both the modes. Since the IPoIB-UD mode is always
    supported, there are no flags to indicate IPoIB-UD support.

    If IPoIB-CM is not supported, i.e., if the implementation only
    supports IPoIB-UD, then the implementation MUST ignore the <Flags>
    on reception. It MUST set the <Flags> octet to all zeros on
    transmission as specified in [RFC4391].

QPN:

    The queue-pair number (QPN) on which the unicast address
    resolution replies will be received [RFC4391]. An IPoIB interface
    has only one UD QP associated with it whether or not it supports
    this extension.

    The QPN also serves another purpose. It is used to form the
    Service-ID that is used to set up the IB connection.

On receiving the multicast/broadcast address resolution request, the
receiver replies with its own link address, including the associated
UD QPN and the appropriate flags.

The receiver's reply is unicast back to the sender after the receiver
has, as in the case of IPoIB-UD, resolved the GID to the Local
Identifier (LID), and determined other required parameters [RFC4391].
Once the address resolution is completed, the underlying IB
connection on the supported connection modes can be set up. An
implementation is NOT REQUIRED to set up a connection merely because
the peer indicates the capability. The decision to make such a
connection is left to the implementation.

3.2. IB Connection Setup

   Once the address resolution is complete, the IB connection can be set
   up by either of the peers. To set up a connection, IB Management
   Datagrams (MADs) are directed to the peer's communication manager
   (CM). The connection request always contains a Service-ID for the
   peer to associate the request with the appropriate service. If the
   request is accepted, the peer returns the relevant connected mode QPN
   in the response MAD. The format of the CM connection messages and
   the IB connection setup process is described in [IB_ARCH]. The
   overall handshake is of the form:

              REQ ---->
                   <---- REP [or REJ(reject)]
              RTA ---->
              [or REJ(reject)]

   The CM messages include, among other parameters, the Service-ID,
   Local connection-mode QPN, and the payload size to use over the
   connection.

   Note: The IB connection is set up using the Service-ID as defined in
         Section 3.5 below. The node MUST keep a record of IB
         connections it is participating in. The node MAY attempt
         another connection to the remote peer using the same Service-ID
         as used for an existing IB connection. Similarly, the receiver
         of such a connection MAY drop the request with a suitable error
         indication in the CM response. The decision to accept or
         initiate multiple connections from or to an IPoIB interface is
         left to the implementation.

   The node that initiated the connection is aware of the target node's
   IP address as described above. The node receiving the IB connection
   request, however, cannot determine the initiating node's link
   address. To enable this determination, every CM message exchanged in
   setting up the IB connection MUST include the sender's IPoIB-UD QPN
   in the "private data" [IB_ARCH] field. The IPoIB-UD QPN MUST be
   included in all "REJ" [IB_ARCH] messages too.

3.3. Simultaneous IB Connections

   To ensure that two IB connections are not set up between the peers
   due to REQ crossing, the following rules MUST be followed:

      The receiver forms the remote node's link-layer address using the
      UD QPN received in the "private data" field of the "REQ" message
      and the GID of the sender included in the "REQ" message. The
      link-layer address is used to determine if there is already an

outstanding connection request "REQ" sent by the local interface
to the given received link-layer address. If such an outstanding
request is determined, then the two link-layer addresses (local
and remote) are numerically compared. If the local link-layer
address is numerically smaller, then the connection is accepted,
otherwise rejected. The error code in "REJ" MAD is set to
"Consumer Reject" [IB_ARCH].

Note: The link-layer addresses formed for comparison zero out the
      connection mode flags specified in Section 3.1. The
      comparison is performed from the most significant octet to
      the least significant octet of the link-layer address.

The above holds even if the receiver supports multiple IB
connections from the same peer. This is to ensure that only one
more connection is set up when the "REQ" messages cross.

3.4. IPoIB-CM IB Connection Teardown

IB connections created through IPoIB-CM are considered part of an
IPoIB interface. As such, they SHOULD be torn down when the IPoIB
interfaces they are associated with are torn down.

Furthermore, the IB connection between two peers MAY be torn down by
either peer whenever the address resolution entry expires. An
implementation is free to implement alternative policies for tearing
down of IB connections between peers.

3.5. Service-ID

The InfiniBand specification defines a block of Service-IDs for IETF
use. The InfiniBand specification has left the definition and
management of this block to the IETF [IB_ARCH]. The 64-bit block is
as follows:

```
+--------+--------+--------+--------+-------+--------+--------+------+
|00000001|<-------------------IETF use---------------------------->|
+--------+--------+--------+--------+-------+--------+--------+------+
```

The Service-IDs used by IPoIB will be in the following format:

```
+--------+--------+--------+--------+-------+-------+--------+-------+
|00000001|  Type  |         Reserved         |          QPN          |
+--------+--------+--------+--------+-------+-------+--------+-------+
```

The "Type" field MUST be set to 0.

The "Reserved" field MUST be set to zeros.

The QPN MUST be the UD QP exchanged during address resolution.

4. Frame Format

All IP datagrams transported over InfiniBand are prefixed by a
4-octet encapsulation header as described in [RFC4391].

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 Type          |            Reserved           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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

Type	Protocol
0x800	IPv4
0x86DD	IPv6

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.

5. Maximum Transmission Unit

The IB connection setup might be used for both IPv4 and IPv6 or it
could be used for only one of them while a different connection is
used for the other. The link MTU MUST be able to support the minimum
MTU required by the protocols.

The default MTU of the IPoIB-CM interface is 2044 octets, i.e.,
2048-octet IPoIB-link MTU minus the 4-octet encapsulation header.

However, connected modes of InfiniBand allow message sizes up to 2^31
octets. Therefore, IPoIB-CM can use a much larger MTU for unicast
communication between any two endpoints. The maximum and/or optimal
payload that can be received or sent over an InfiniBand connection is
dependent on the implementation, IB Channel Adapter, and the
resources configured.

An implementation MAY utilize the following mechanism to exchange the
optimal message size across the IB connection.

5.1. Per-Connection MTU

Every IB connection setup message includes a "private data" field
[IB_ARCH]. The "private data" field in the connection setup message
(CM REQ) MUST include the "Receive MTU". This indicates the maximum
packet size the requester can accept. The requester MUST be able to
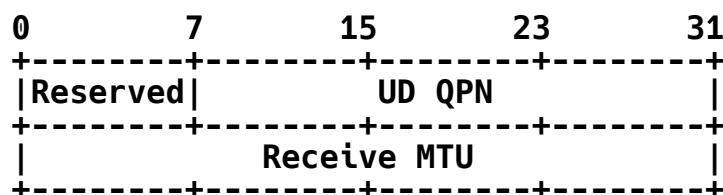accept smaller MTU sizes as well.

It is up to the implementation to utilize this mechanism for setting
the per-IB connection MTU. To calculate the resultant IPoIB MTU over
the connection the smaller of the two IB "Receive MTU" values is used
by both the peers. The IPoIB interface must also account for the 4-
octet encapsulation header and so the IPoIB MTU over the connection
will be further reduced by that amount.

6. Private-Data Format

The "private data" field in every CM message for connection
establishment must include the following values:

    1. UD QPN of the sender
    2. Receive MTU supported by the sender

The format of the "private data" field MUST be as follows:

```
        0        7       15       23       31
        +--------+--------+--------+--------+
        |Reserved|        UD QPN            |
        +--------+--------+--------+--------+
        |             Receive MTU           |
        +--------+--------+--------+--------+
```

The Reserved value MUST be set to zero on transmit and ignored on
receive.

7. IPoIB-CM Considerations

   Every IPoIB interface supports IPoIB-UD. It may additionally support
   one or both of the IPoIB-CM modes. Therefore, there can be multiple
   methods of communicating between any two peers. This implies that an
   interface MAY transmit/receive a packet over any of the RC, UC, or UD
   modes depending on the modes supported between it and the peer. It
   further follows that every IPoIB implementation compliant with this
   document MUST accept all IP unicast transmissions over any of the
   IPoIB modes it supports. Multicast and broadcast packets by their
   nature will always be transmitted and received over the IPoIB-UD QP.
   Additionally, all address resolution responses (ARP or Neighbor
   Discovery) MUST always be encapsulated in a UD mode packet.

7.1. A Cautionary Note on IPoIB-RC

   The RC mode of InfiniBand guarantees in-order delivery of packets.
   Every message transmitted over the RC connection is broken into
   physical MTU-sized packets by the RC connection. If any packet is
   lost, it is retransmitted until the complete message is exchanged.
   Therefore, there is a possibility of an upper transport layer
   experiencing a timeout, while the RC layer is still in the process of
   transferring the complete message. TCP will view the timeout as an
   indicator of congestion and enter slow-start thereby affecting
   throughput drastically [RFC2581]. Other upper-layer protocols might
   insert retransmissions into the fabric, adding to the already
   existing congestion.

   The applicability of Infiniband reliability is on a fabric with short
   latencies (not wide area). Therefore, the RC timer values should be
   short compared with the starting minimum time values used by the
   upper end-to-end transports. In addition, because the RC mode does
   not have measurement-based reliable transmission, its use over
   fabrics with long latency or very dynamic latency may be a concern
   for congestion-aware traffic traversing those fabrics.

7.2. IPoIB-CM Per-Destination MTU

   As described above, interfaces on the same subnet may support
   different link MTUs based on the negotiated value or due to the link
   type (UD or connected mode). Therefore, an implementation might
   choose to define a large IP MTU, which is reduced based on the MTU to
   the destination. The relevant MTU may be stored in a suitable per-
   destination object, such as a route cache or a neighbor cache. The
   per-destination MTU is known to the IPoIB-CM interface as described
   in Section 5.

Implementations might choose not to support differing MTU values and
always support an MTU equal to the IPoIB-UD MTU determined from the
broadcast GID.

8. Security Considerations

An impostor may return a false set of flags to an IPOIB interface.
This may cause unnecessary attempts and some delay/disruption in
IPoIB communication. The same is the case if wrong/spurious QPN
values are provided during address resolution broadcast/multicast.

9. IANA Considerations

Future uses of the reserved bits and octets in the link-layer address
(Section 3.1), Service-ID (Section 3.5), and "Private-Data Format"
(Section 6) MUST be published as RFCs. This document requires that
the reserved bits be set to zero on sends.

10. Acknowledgements

The author thanks the IPoIB Working Group for the various comments
and suggestions. A special thanks to Bernie King-Smith and Dror
Goldenberg for the detailed review and suggestions.

11. Normative References

   [IB_ARCH]   InfiniBand Architecture Specification, version 1.2
                  www.infinibandta.org

   [RFC4392]   Kashyap, V., "IP over InfiniBand (IPoIB) Architecture",
                  RFC 4392, April 2006.

   [RFC4391]   Chu, J. and V. Kashyap, "Transmission of IP over
                  InfiniBand (IPoIB)", RFC 4391, April 2006.

   [RFC2119]   Bradner, S., "Key words for use in RFCs to Indicate
                  Requirement Levels", BCP 14, RFC 2119, March 1997.

12. Informative References

   [RFC2581]   Allman, M., Paxson, V., and W. Stevens, "TCP Congestion
                  Control ", RFC 2581, April 1999.

Author's Address

    Vivek Kashyap
    15350, SW Koll Parkway
    Beaverton
    OR 97006

    Phone: +1 503 578 3422
    EMail: vivk@us.ibm.com

Full Copyright Statement

Intellectual Property

Acknowledgement