Some Experience with Linux IPoIB Implementations

Hal Rosenstock

IETF IPoIB WG

August 2004

Agenda

- Interoperation Experience
- Issues

Interoperation Experience

- **January 2004 IBTA Plugfest**
  - 2 IPoIB implementations showed
  - More extensive testing planned for upcoming August Plugfest
    - **IPv4 only**
      - ping, ftp (TCP)
      - UDP, IP multicast (new)
      - No DHCP yet (next Plugfest)

Interoperation Experience

- **OpenIB**
  - Several vendor implementations currently
    - **IPv4 with DHCP support**
    - **Some with IPv4 multicast support**
  - Heading toward single implementation for Linux (2.6) kernel
  - InfiniBand open source (including IPoIB) targeted for 2.6 Linux kernel inclusion (kernel.org)
    - **IPoIB is a deliverable in the first phase**
  - Issue summary is primarily a result of openib discussions
    - **Views represented are not necessarily my own or those of the presenter but attempt to fairly represent openib discussions ☺**
  - http://www.openib.org/

Issues

- ARP Table
- Join
- Broadcast Group Setup
- Interface “RUNNING”
- SM Restart
- Performance

ARP Table

- **Problem: Current hardware address size in (Linux) ARP table is insufficient for IB "hardware" address**
  - Alternatives
    - **Rebuild kernel with larger hardware address size**
      - Not acceptable in some deployments
    - **Other implementation techniques to use table as is**
- **Solution: Expand (Linux) ARP table hardware address size**
  - Increase MAX_ADDR_LEN (include/linux/netdevice.h) from 8 to a minimum of 20 octets
    - **Some other minor changes associated with this (core/dev.c)**
  - This was in Linux 2.4 kernels
    - **Fixed in Linux 2.6 kernels**
      - MAX_ADDR_LEN is 32 octets
      - Also, core/dev.c for this

Join

- **Implementations are currently only successfully joining when the group has been pre-administered at the SM**
  - Due to additional components required for group creation
- **Does “join” mean “create” if group does not currently exist ?**
  - Either not all implementers have interpreted it this way or just not implemented this way
  - Does I-D language need any tightening up ?
    - **"An IB multicast group must be explicitly created through the SA before it can be used.“**
    - **"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."**

Broadcast Group Setup

- **Defined by P_Key, Q_Key, SL, MTU, and network layer components (TClass, FlowLabel, HopLimit)**
- **Administrative Model**
  - Who is responsible for broadcast domain ?
    - **Network admin for SM/SA**
    - **Individual end node administrators**
  - There are differing views on where this responsibility lies

Broadcast Group Setup

- **Affects Join issue**
  - If group is pre-administered at SM/SA, end nodes only need to join rather than create (which also joins)
  - If group is not pre-administered, more configuration information is needed at end nodes to create group as there is a “race” for which end node creates group
    - **Q_Key “additional” configuration item**
      - I-D states “It is RECOMMENDED that a controlled Q_Key be used with the high order bit set”
        - » Many controlled Q_Keys
        - » Recommendation not requirement
      - Algorithm to work around not needing to configure this at every end node in the IPoIB subnet

Interface "RUNNING"

- **Issue:**
  - IPoIB transmit packets are dropped because one or more multicast groups could not be joined (created). All unaffected transmits (non IP multicast and and IP multicast to groups which are established) should work rather than be dropped.
  - Some "split" or degraded connectivity is possible depending on which multicast "overlays" work (like IPoATM)

Interface "RUNNING"

- **When is the IPoIB interface "RUNNING" ?**
  - Does it require the successful join of the broadcast group ?
    - **Does it make sense to set the interface RUNNING without this ?**
      - This should be the minimum
      - Note that interface states are OS specific and this may not apply to all Oss
      - I-D states "the IPoIB link is formed by the IPoIB nodes joining the broadcast group"
        - » Is this a sufficient statement for an IPoIB node failing to join the broadcast group ?
        - » Perhaps it should state "From the IPoIB node perspective, the node is not part of the IPoIB link until (at least) the broadcast group is successfully joined" as well

Interface "RUNNING"

- **When is the IPoIB interface "RUNNING" ? (cont'd)**
  - Any other multicast groups required for this ?
    - **All systems on subnet (224.0.0.1) as well ?**
    - **Any others ?**
    - **Not in IPoIB I-Ds**
      - Should some statement be added ?

SM Restart

- **In case of SM restart where SMs do not synchronize subscriptions (e.g. multicast), IBA is missing a way to tell a client to resubscribe**
  - It should be noted that SM synchronization is a requirement but there was no compliance for this
  - MgtWG is currently discussing the addition of a mechanism for this for possible inclusion in IBA 1.2

SM Restart

- **In window between SM takeover/failover (or single SM restart (which is not part of IBA architecture but appears to be a deployment scenario)), IPoIB subnet for existing nodes may not function (when subnet local)**
  - Only if node has talked with other node (and cached information); otherwise SA interaction is currently needed
    - **GID to LID translation**
    - **Additional components (SL, Path MTU, network layer (Tclass, FlowLabel, HopLimit)) from PathRecord**
    - **This is the boot up scenario**
      - IPoIB nodes have to access the SA to obtain pathrecord information to fill the pathrecord cache and send unicast ARP messages

SM Restart

- **No Active SM Window (cont'd)**
  - Intent is to continue operation for all IPoIB nodes currently on the subnet (in the absence of any changes) when in the window when no SM/SA exists
  - Should enhancements for this be made (for subnet local) ?
    - **Would this require ARP snooping/periodic promiscuous ARPing ?**
  - What enhancements would be needed ?
    - **Add LID to ARPs when subnet local**
      - Other PathRecord components
        - » SL at a minimum
        - » NL components not issue as subnet local only

Performance

- **Performance to date is not equivalent to “dumb” ethernet**
  - Bandwidth
  - CPU Utilization
  - Quantify per byte and per packet overheads and where CPU time is spent
    - **Where is overhead in implementation ?**
      - HCA hardware, driver, access layer, IPoIB

Performance

- **Bandwidth is currently limited by use of UD (and MTU)**
  - Max UD MTU is 4K
    - **In practice, currently 2K**
  - Increasing the IPoIB link MTU from 1500 to 2000 bytes yields ~20% gain in bandwidth
  - Connected mode addresses larger MTU which would help with bandwidth utilization issues
- **Checksum offload**
  - Appears to be insufficient market demand for this
  - Other solutions available

Thank You

Voltaire