Status of FCoE Work Items

IBM
Scott Carlson, Roger Hathorn,
Renato Recio

T11/07-443v0 Aug 8, 2007

Outline

- Stuff that impacts the FCoE Header
  - Length field vs no length field
  - Preventing OX_ID reuse
  - Timestamp option
  - Reserved bits
- Stuff that doesn’t impact the FCoE Header
  - ACL Mechanism
  - N_Port ID to MAC Address Translation
  - FCoE Port Virtualization
  - Virtual FCoE Switch Protocol

FCoE Header – Length Field

- One standard FCoE frame format is required
- Two key values associated with the length field debate
  - No length field enables cut-through or speed matching (vs store-and-forward) gateways
    - Only relevant for specific windows in time when FC and Ethernet link speeds are close.
  - Packet efficiency of using a length field.
    - T11/07-421v0 (Snively) shows packet efficiency is sufficiently equivalent to not represent a strong argument in support of a length field vs no length field (see Recio, Krause, Snively e-mails).
- No length field enables flexibility in FCoE switch design
  - Enable cut-through designs at minor overhead
- We need to select between the “no length” proposals:
  - Broadcom’s Pad Start field or
  - Brocade’s 64 byte minimum packet size

Preventing OX_ID reuse

- Baseline Approach - Black (T11.3 e-mail 7/26 9:05am CDT, subject “OX_ID Re-use - baseline approach”)
  - FC fabric architectures have an 8-hop limit, based on:
    - Near-universal 10 sec R_A_TOV,
    - 2 second E_D_TOV and
    - 0.5 second max switch hold time
  - For FCoE, divide FCoE and Ethernet-only switches into 3 classes:
    - Good → configurable to <= 0.5 sec max hold time
    - Not so Bad → configurable to <= 1 sec max hold time
    - Bad → Max hold time is > 1 second.
  - Apply the following rules for FCoE fabrics
    - Do not use Ugly switches (> 1 sec max hold time)
    - The 8 hop limit applies to Good switches (<= 0.5 sec max hold time)
    - Each Not so Bad switch (<= 1 second max hold time) counts as 2 hops against the 8 hop limit.
  - Above does constrain diameter of fabrics with FCoE, but constraint seems tolerable.

Timestamps in FCoE Header

- Seems to be agreement that we want to enable **optional** use of timestamps by FCoE switches within an FCoE fabric.
  - Can allow operation in less robust or controlled environments (more hops, “ugly” switches, etc)
  - May provide other problem analysis benefits (see Hufferd 7/30 1:41pm CDT e-mail, subject “RE: [T11.3] OX_ID Re-use - baseline approach”)
- Areas where there seems to be agreement
  (more interesting areas in next slide :{)
  - General consensus that packet efficiency is sufficiently equivalent with or without Timestamp fields (Snively, Krause)
  - To enable an architecturally cleaner constant header size,
    FCoE Header always carries two fields, which have the following semantics:
    - Timestamp Valid - If set, the timestamp field is valid.
    - Timestamp - Timestamp mark (see Brocade proposal for semantics).
  - The Timestamp Valid and Timestamp field are defined as:
    - "Reserved" for N_Ports and
    - Optional for F_Ports and E_Ports.
  - FCoE adapter semantics
    - On egress Timestamp Valid and Timestamp fields are set to zero
    - On ingress Timestamp Valid and Timestamp fields are ignored.
  - FCoE switch semantics
    - An FCoE switch can either use or not use the two timestamp fields.
  - To fully exploit the use of Timestamps in an FCoE fabric requires:
    - FCoE adapters to attach through an FCoE switch that supports Timestamps and
    - All FCoE switches in the topology to support Timestamps

Timestamps in FCoE Header - Endpoints

- All agree to **not** making the Timestamp fields required for N_Ports.
- The area where folks do not seem to have reached agreement is whether the timestamp is Reserved or Optional for N_Ports.
  - Option 1: Reserved for N_Ports means Timestamps **must not** be used by an FCoE adapter.
    - An adapter that uses the Timestamp fields would ***not be in compliance*** with the FCoE Release 1.0 specification.
  - Option A: Optional for N_Ports means Timestamps **may** be used by an FCoE adapter.
    - An adapter that uses the Timestamp fields would ***be in compliance*** with the FCoE Release 1.0 specification.
- We must pick between one of these two options!

FCoE Header – Reserved Bits

- General consensus on the need for reserved bits (Recio, Krause, Snively, Pelissier → see "[T11.3] Reserved bits" e-mail thread).
- Specific option for reserved bits not agreed to yet.
- Two basic options
  1. FCoE Header supports versioning - That is, FCoE Header contains some designation of the version supported (see Pelissier and Thaler 7/23 e-mails):
     - A sub-type field - used to indicate the version supported (3 bits appears to be consensus)
     - Reserved bits - which may be used to enable functions supported by a future sub-type. *Though one could argue a 3-bit sub-type field is sufficient, because new versions could simply have a larger header.*
  2. FCoE Header doesn't support versioning - Instead a log-in protocol (FLOGI/PLOGI?) would be modified to support FCoE N_Port log-in parameters, possibly including FCoE version (see Krause 7/23 e-mail):
     - Reserved bits - number of bits determined after all other fields defined.
- Option 2 may require addition of FCoE versioning to log-in

Summary of FCoE Header Work Items

- We need to select between the “no length” proposals:
  - Broadcom’s Pad Start field or
  - Brocade’s 64 byte minimum packet size
- Preventing OX_ID reuse
  - R_A_TOV-based approach must be specified.
- Enable optional switch based timestamps through the following required header fields:
  - Timestamp Valid - Optionally set by FCoE Switches.
    If set, the timestamp field is valid.
  - Timestamp - Timestamp mark.
- Reserved bits - If we do not want to tamper with log-in, then support for FCoE Header versioning appears to be favored.
  - Based on e-mail threads, a 3-bit sub-type field provides sufficient growth of future versions.

ACL Mechanism

- Consensus in Colorado that we need to support direct FCoE endpoint to FCoE endpoint communications.
  - Also consensus that a standard ACL mechanism is needed to provide zone integrity for direct communications.
- Three approaches discussed in Colorado:
  1. Endpoint based ACL approach (esp. hard zoning) - see slide 10 of 07-408v1.
  2. Switch based ACL approach (esp. hard zoning) - see slide 11 of 07-408v1.
  3. Both
- Insufficient reflector discussion on this one to determine which of above approaches is preferred.
  - What additional work is needed for us to decide?

N_Port ID to MAC translation

- Consensus in Colorado on the need to standardize one of the three N_Port ID to MAC translation alternatives:
  1. FCoE changes or adds FC nameserver responses (proposal was provided in 07-408v1, page 22)
  2. FCoE has new FCID to MAC translation service (proposal was provided in 07-408v1, see page 23)
  3. Direct mapping by dropping FCID into FC MAC template (included in Cisco's 07-303v0 proposal.
- Insufficient reflector discussion on this one to determine which of above approaches is preferred.
  - What additional work is needed for us to decide?

FCoE N_Port Virtualization

- Consensus in Minneapolis on the need to support N_Port Virtualization.
- However, we did not reach consensus in Colorado on ***which*** of the possible physical port virtualization models we will want to enable.
- The models discussed were:
  1. Single MAC shared by:
     - A single FC stack (i.e. N_Port ID) and
     - A single IP stack (i.e. IP address).
  2. Single MAC shared by:
     - Multiple FC stacks (i.e. multiple N_Port IDs) and
     - Multiple IP stacks (i.e. IP addresses).
  3. Two MACs, where:
     - All FC stacks (N_Port IDs) share a MAC.
     - All IP stacks (IP addresses) share a MAC.
  4. Multiple MACs, where:
     - Each FC stack (N_Port ID) gets a unique MAC.
     - Each IP stack (IP address) gets a unique MAC.
  5. A mix of the above?
  6. All of the above?
- Insufficient reflector discussion on this one to determine which of above approaches is preferred.
  - What additional work is needed for us to decide?

Virtual FCoE Switch Protocol (VFSP?)

- Consensus that a proposal analogous to VRRP is needed for FCoE.
  - Proposal created by T11 folks and taken through IETF RFC process when nearing completion.
- VFSP problem statement - cover the case where an FCoE Switch (FS?) fails, such that all traffic going through the failed switch is routed to a back-up FS.
  - Necessary for FCoE HA environments.
- Basic approach
  1. Creation of a single Virtual FCoE Switch (FS) from multiple physical FSs.
  2. Creation of a well known Virtual FCoE Switch Protocol (VFSP) MAC address set that is used to access the virtual FSs (e.g. nn-nn-nn-nn-nn-{VFSID}, where the last byte is used to to carry the Virtual FCoE Switch's Identifier.
  3. The FC domain space is shared across all physical FSs that share the same Virtual FS's Identifier.
     - Allows a packet with an N_Port ID to be routed to any physical FS that shares the same Virtual FS's Identifier.
  4. All physical FCoE gateways that share the same Virtual FS's Identifier communicate with each other to select which is master and which are back-ups.
  5. When configuring an interface, the VFSP FSs broadcast a gratuitous ARP request containing the VFSP MAC addresses for each N_Port ID on that interface.
     - An FCoE multicast group IP address is used in the ARP.
     - This can be used by switches to learn the number of VFGP gateways that have the same MAC address and thereby route traffic to all of them.
     - Proprietary options may also be used to configure the switch routing tables (e.g. command line interfaces to configure switches), but a standard option (5 in this list) must be provided.
  6. An VFSP gateway only routes packets if it is the master.