T11 FC-SW-6

NPZD Re-ordering

Roger Hathorn

04/08/12 – 12-141v0
10/04/12 – 12-141v1
12/04/12 – 12-141v2
01/04/13 – 13-051v0

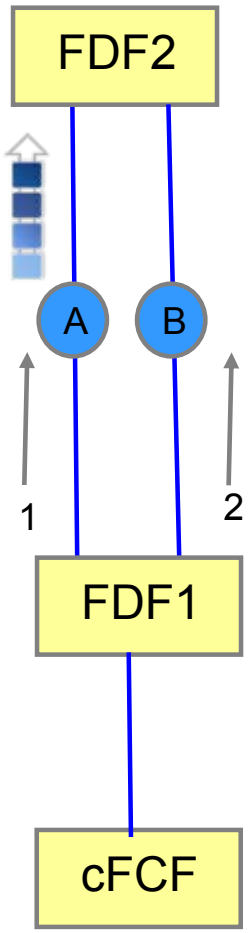
History

- **12-141v0 presented in April, 2012, 12-141v1 presented in October, 2012**
- **12-141v2 presented in January, 2013 and revised as 13-051v0**
- **Although other approaches to solving this problem have been presented, no complete alternate solutions have been proposed**
- **Since original proposal, NPZD was changed in October to use a global sequence number. This version goes back to a sequence number per FDF.**
- **Updates since v1 in October**
  - Add Sequence number per FCDF
    - Implementation flexibility to minimize retries.
  - General NPZD Retry Rule
  - Improvements in text.
  - Functional changes in green

Re-ordering of NPZD Requests

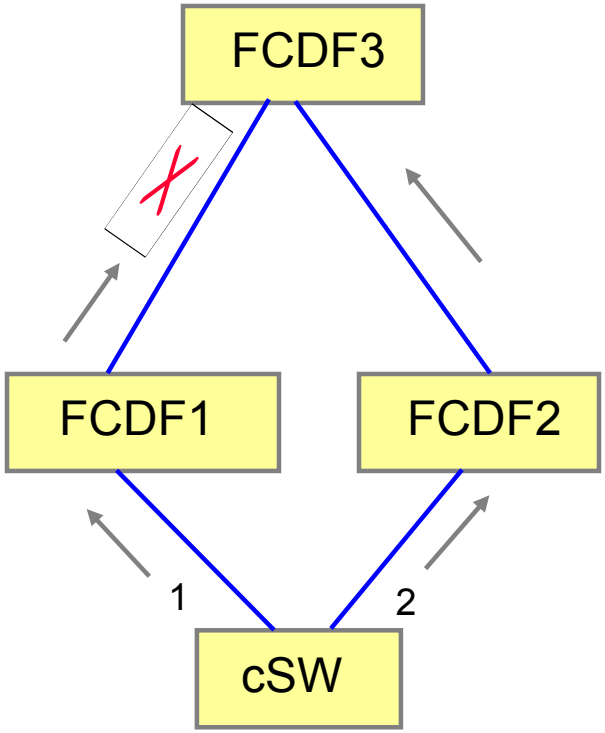
- **Multiple paths can exist from cFCF to FDFs**
- **Since each NPZD is a different exchange, each can take a different path if there are multiple equal cost paths to FDF.**
- **If the network uses spanning tree, topology changes can result in frames taking different paths**
- **NPZD recovery retransmission results in re-ordering of requests.**
- **Hence, NPZD frames can be received by FDFs out of order!**
- **When re-ordering happens, NPZD peering entries contain N_Port IDs that are unknown to the recipient in terms of allocation.**

Re-ordering ...



- cFCF sends NPZD frames 1 and 2 in that order to FDF2.
- FDF1 switches/routes the frame to FDF2.
- Frame 1 is sent thru link A and Frame 2 is sent thru link B.
- Link A is being flow controlled. Hence, frame 1 is queued.
- Hence, it is possible that FDF2 receives frame 1 later than frame 2.
- It is possible that NPZD frame is switched based on MAC address by FDF1 hardware.

Re-ordering ...

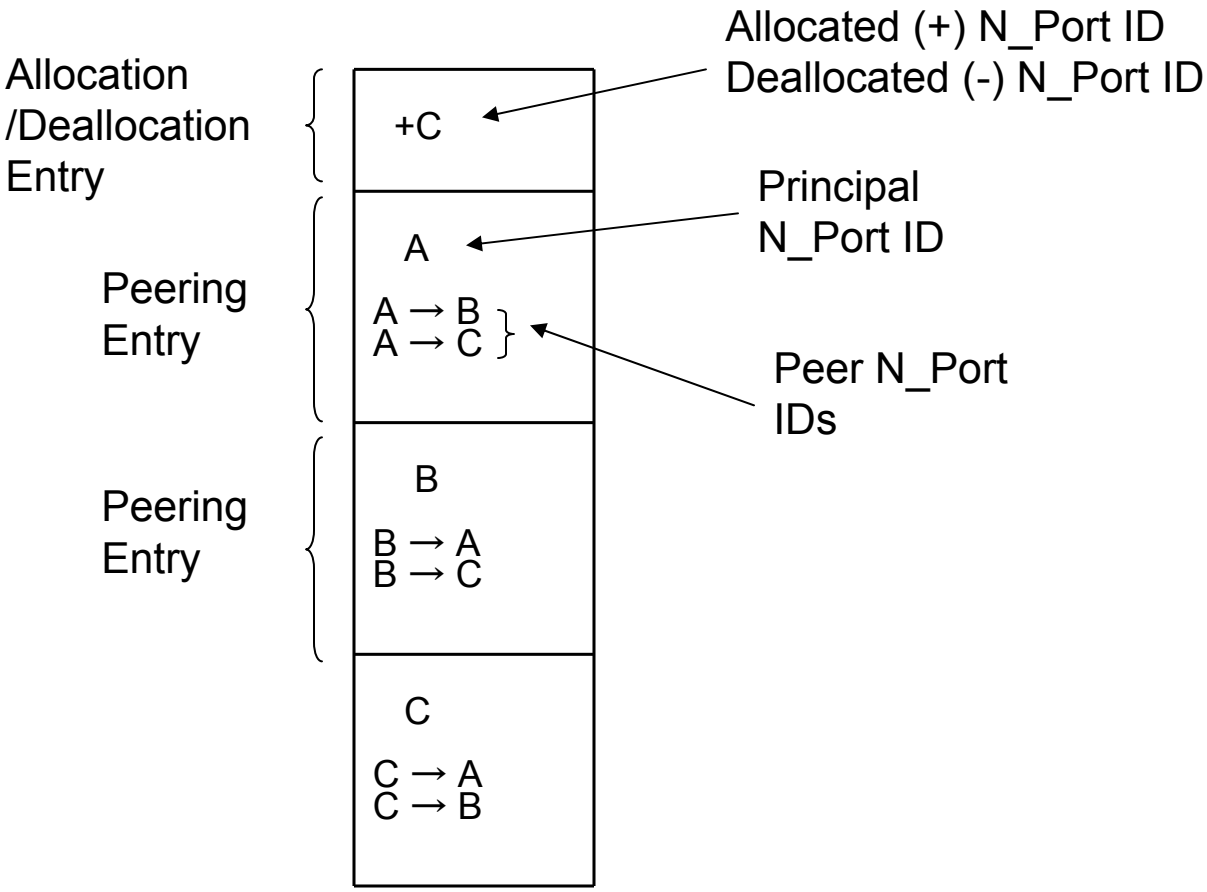


- cFCF sends NPZD frames 1 and 2 in that order to FCDF3.
- Frame 1 is sent thru FCDF1 and Frame 2 is sent thru FCDF2.
- Link between FCDF1 and FCDF3 is being flow controlled. Hence, when FCDF1 forwards frame 1 to FCDF3, it is queued.
- Hence, it is possible that FCDF3 receives frame 1 later than frame 2.
- Controlling Switch software may not always have control over which link the frame is being forwarded by hardware.

Frequency of Out of Order NPZD?

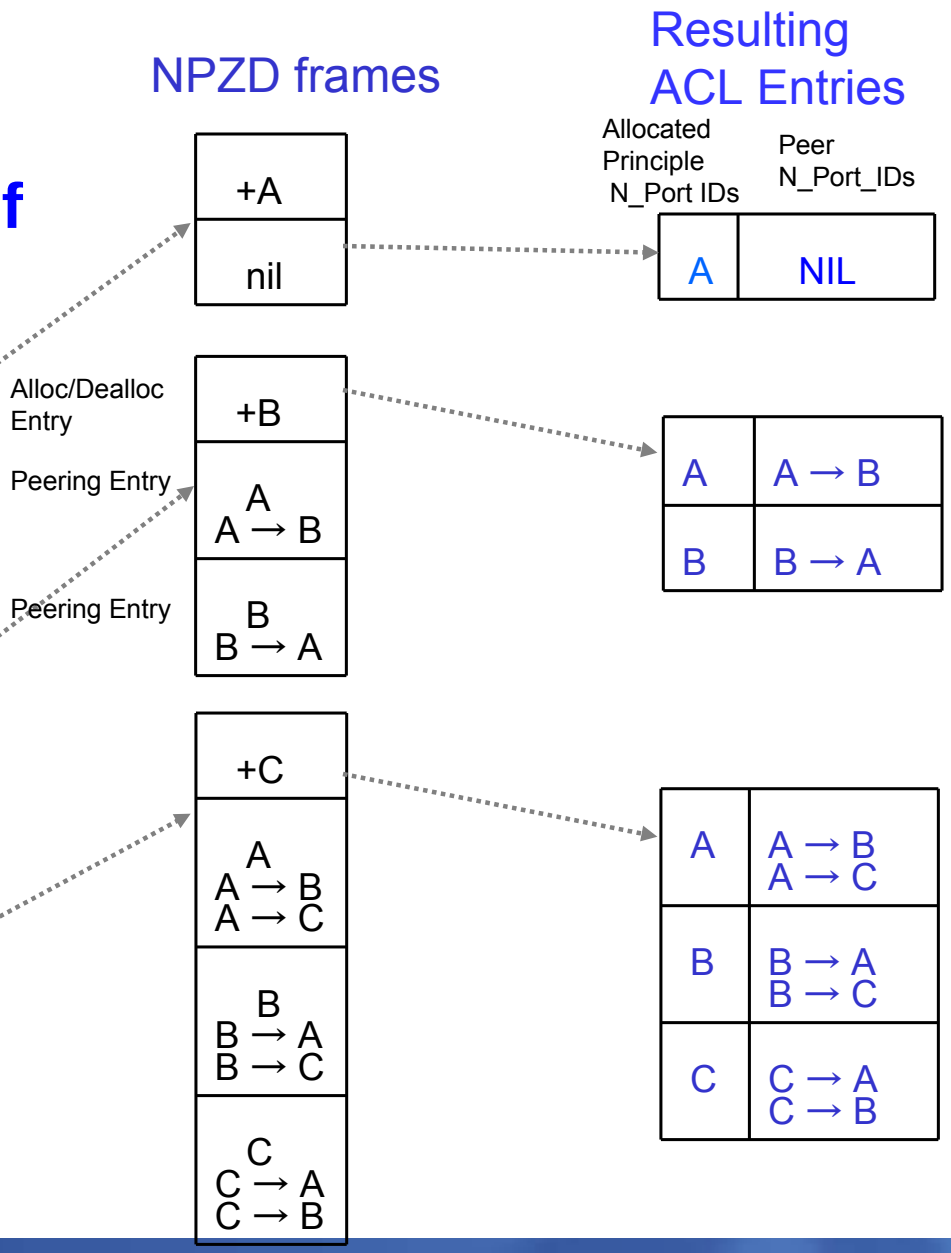
- **This is not I/O**
  - There are not 100's of thousands of these flowing per second.
  - Login happens infrequently and for a short period of time
- **Link Aggregation**
  - Methods can be used to steer control traffic (NPZDs) down specific paths when outstanding NPZDs exist
  - Separation of control traffic with I/O traffic can be achieved with separate priorities.
- **However, when it does happen, we have to deal with it.**

NPZD Format Used in These Slides



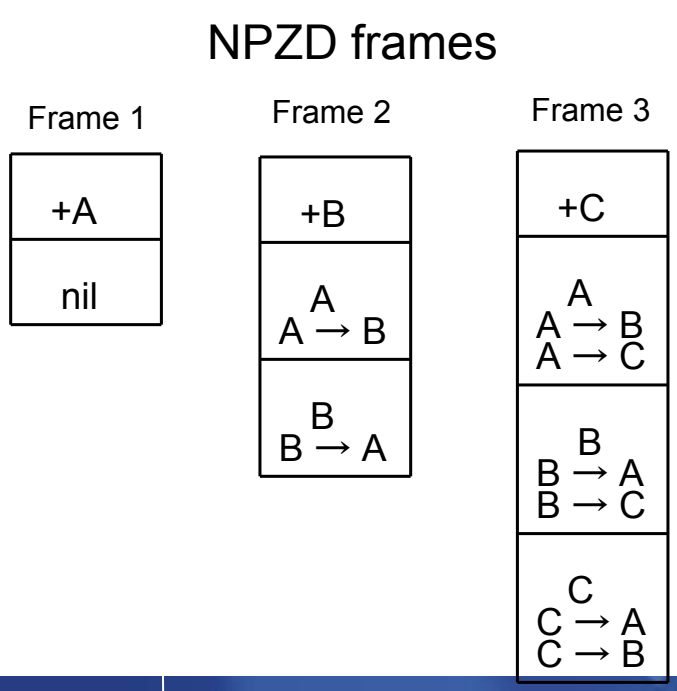
Case 1 – Good Case

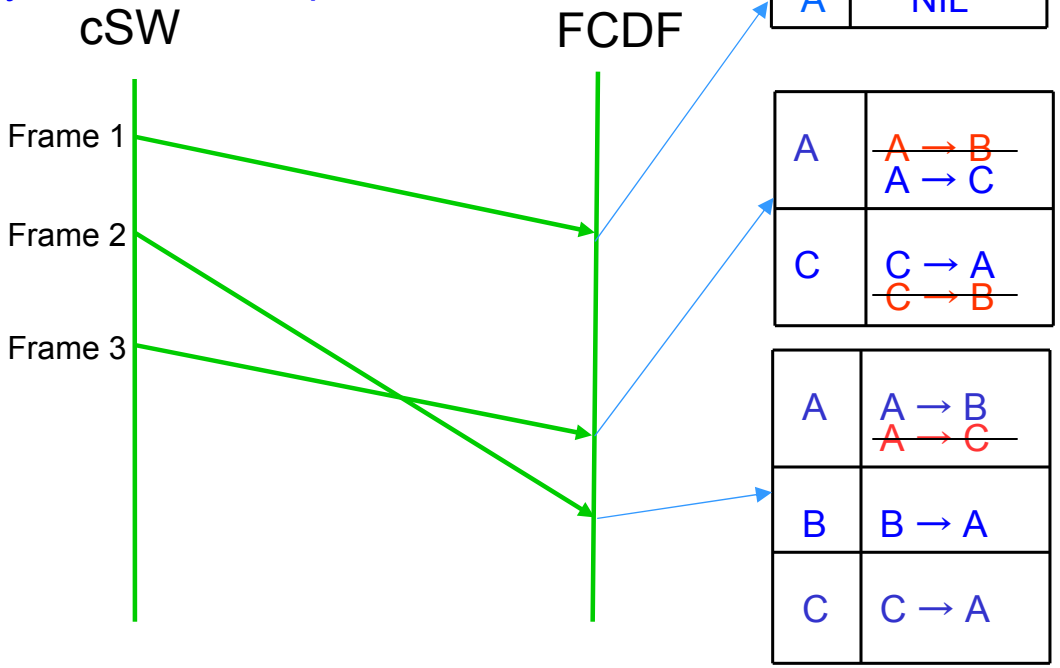
- **Devices A, B and C part of the same zone and log in through same FDF**
- **State 1**
  - Device A logged in
- **State 2**
  - Device B logged in
- **State 3**
  - Device C logged in



Case 1 Problem … (frame 2 and 3 received out-of-order)

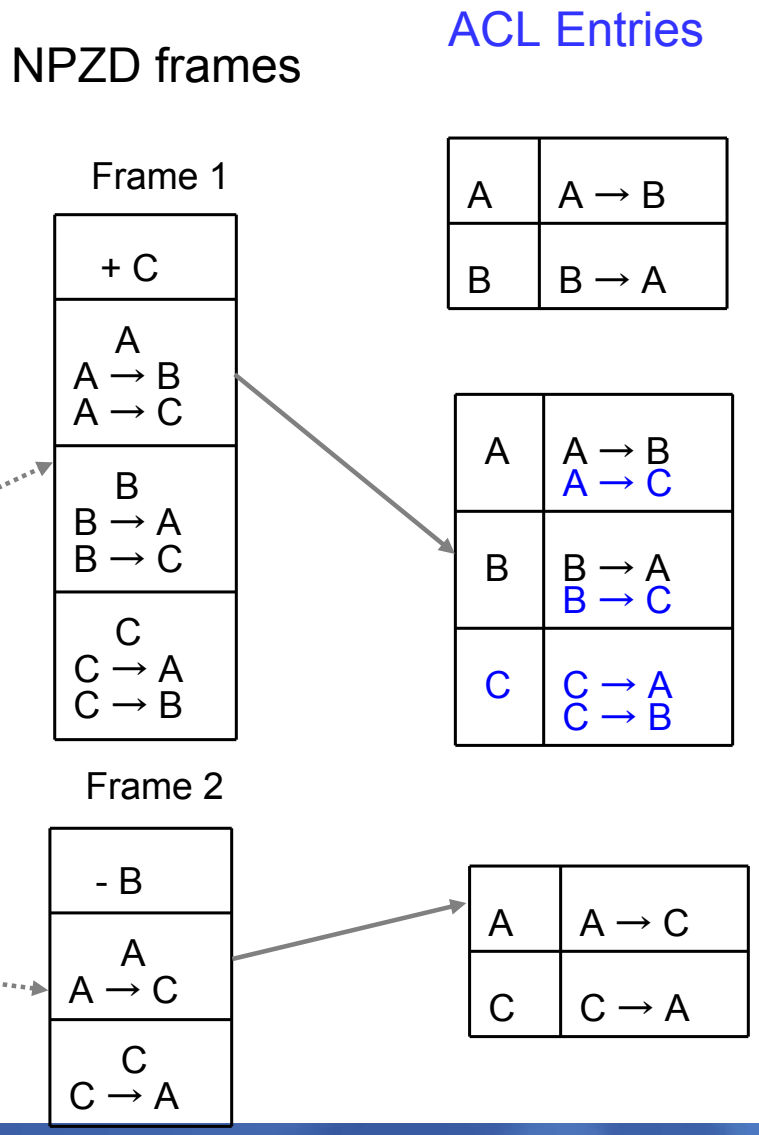
- When Frame 3 is received, FDF does not know about device B information as allocation information about device B has not yet reached FDF. Peering entries associated with device B cannot be used.
- Peering Entries contain a full list of peers with which the Principle N_Port ID is allowed to communicate. When Frame 2 reaches FDF, it may remove A; A → C ACL entry !
- Entry B;B→C and C; C→B are not programmed at all. Principle N_Port_ID C may be removed entirely, since it is not present in Frame 2





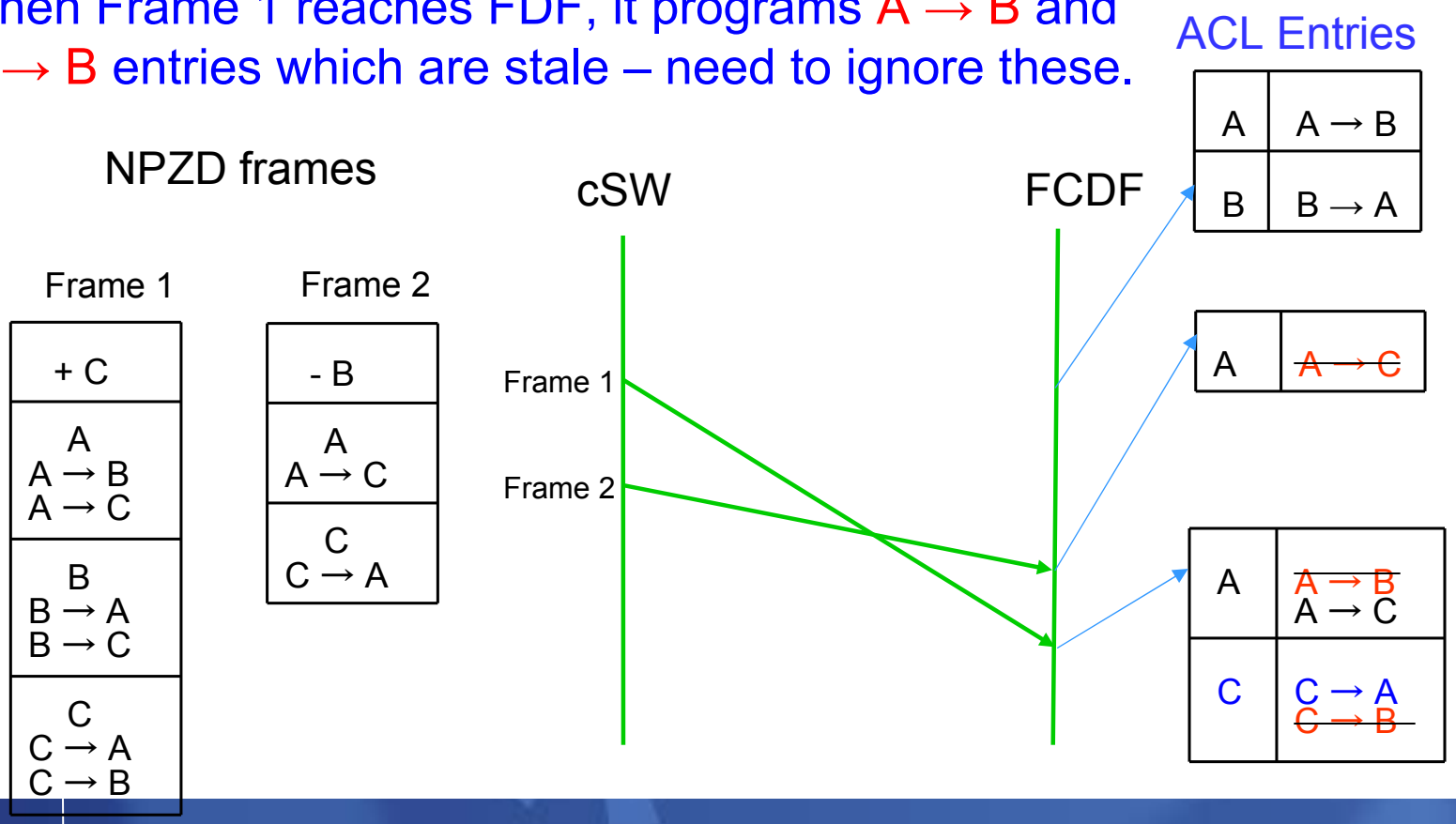
Case 2 – Good Case

- **Devices A, B and C part of the same zone, logged in through same FDF**
- **State 1**
  - Device A, B already logged in
- **State 2**
  - Device C logged in
- **State 3**
  - Device B logged out



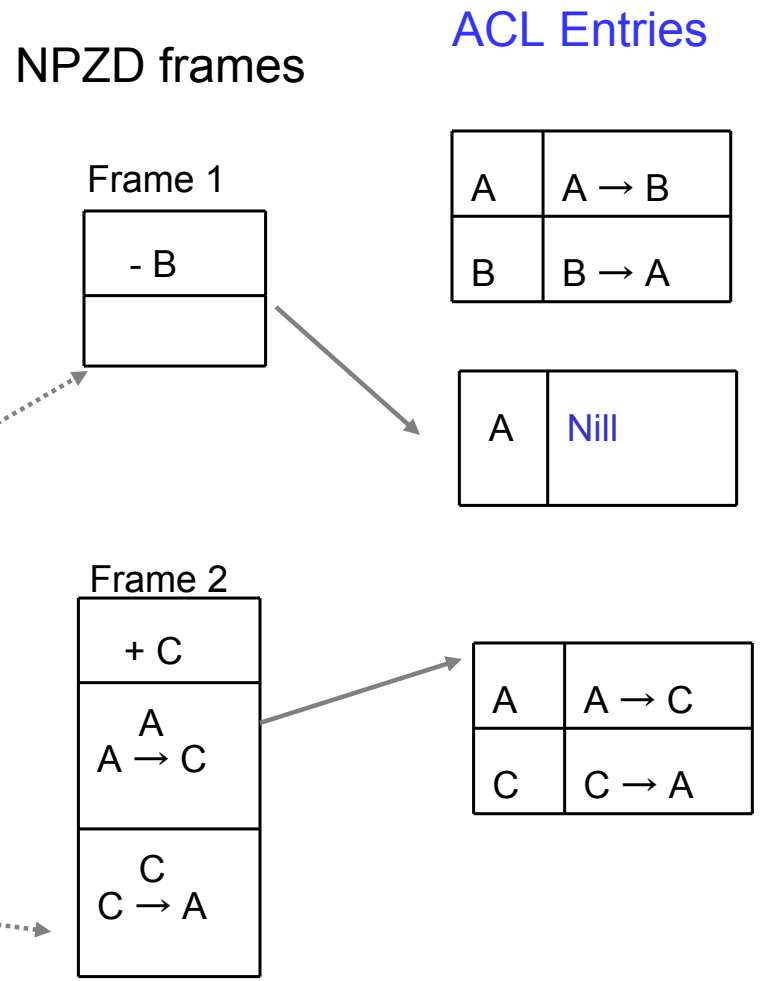
Case 2 Problem … (frame 1 and 2 received out-of-order)

- When Frame 2 is received first, FDF does not know about Device C, hence it should not program A → C entry or C → A entry
- When Frame 1 reaches FDF, it programs A → B and C → B entries which are stale – need to ignore these.



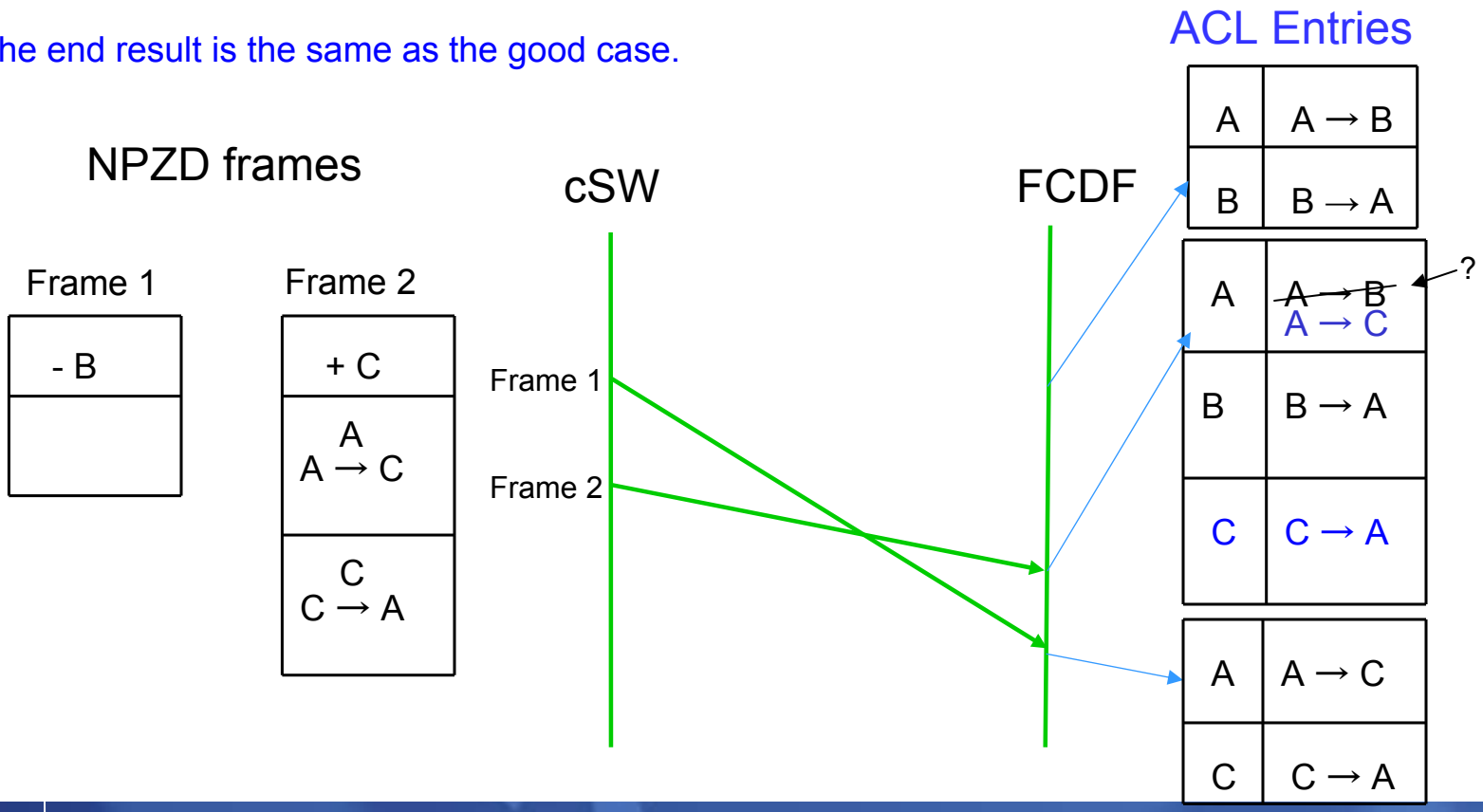
Case 2.5 – Good Case (same events as case 2, in opposite order)

- **Devices A, B and C part of the same zone, connected to same FDF**
- **State 1**
  - Device A, B already logged in
- **State 2**
  - Device B logged out
- **State 3**
  - Device C logged in



Case 2.5 Problem? (frame 1 and 2 received out-of-order)

- A peering entry contains a *complete* list of peer N_port IDs
- When A;A→C is received in Frame 2, it should **fully replace** the existing A;A→B entry
- If not, the A;A→B entry will be removed when Frame 1 (-B) arrives
- The end result is the same as the good case.



Solution – Sequence numbers in NPZD Requests

- Use the Sequence Number descriptor in NPZD messages to manage order
- cSW maintains a sequence number per FCDF in the membership list.
- cSW increments the sequence number by 1 for an FCDF upon each sent NPZD request to that FCDF, and includes the incremented sequence number in the NPZD.

Solution (continued) – NPZD Sequence Checking

- FCDF expects frames with Sequence Number higher than last "processed" Sequence Number (**Processed means accepted or rejected or otherwise discarded)**
- If NPZD request has a lower Sequence Number than the last one processed, it is out-of-sequence
- If NPZD frame is out-of-sequence, FCDF shall
  - Discard the frame and
  - Send a VA_RJT with OUT_OF_ORDER reason code

Solution (continued) – NPZD Retry

- When a controlling switch receives VA_RJT (OUT_OF_ORDER), NPZD is retried.
- When an NPZD is retried for ANY reason, the current state of logged in devices and zoning is used to re-build NPZD content and the sequence number is incremented by 1 from the last request that was initiated.

Solution (continued) – NPZD Processing

- NPZD requests that are processed may contain Peering Entries or peer N_Port IDs within Peering Entries for VN_Ports which are unknown to the FCDF (i.e. an allocation entry has not yet been processed).
- To avoid inconsistent states in forwarding tables,
  - If NPZD request contains a Peering Entry with an “unknown” Principal N_Port_ID, the FCDF shall ignore that entire Peering Entry (see case 2, peering entry with principle C)
  - If NPZD frame contains an unknown Peer N_Port_ID within a peering entry for a known principle N_Port ID, FCDF shall ignore that Peer N_Port_ID within the peering entry (see case 2)

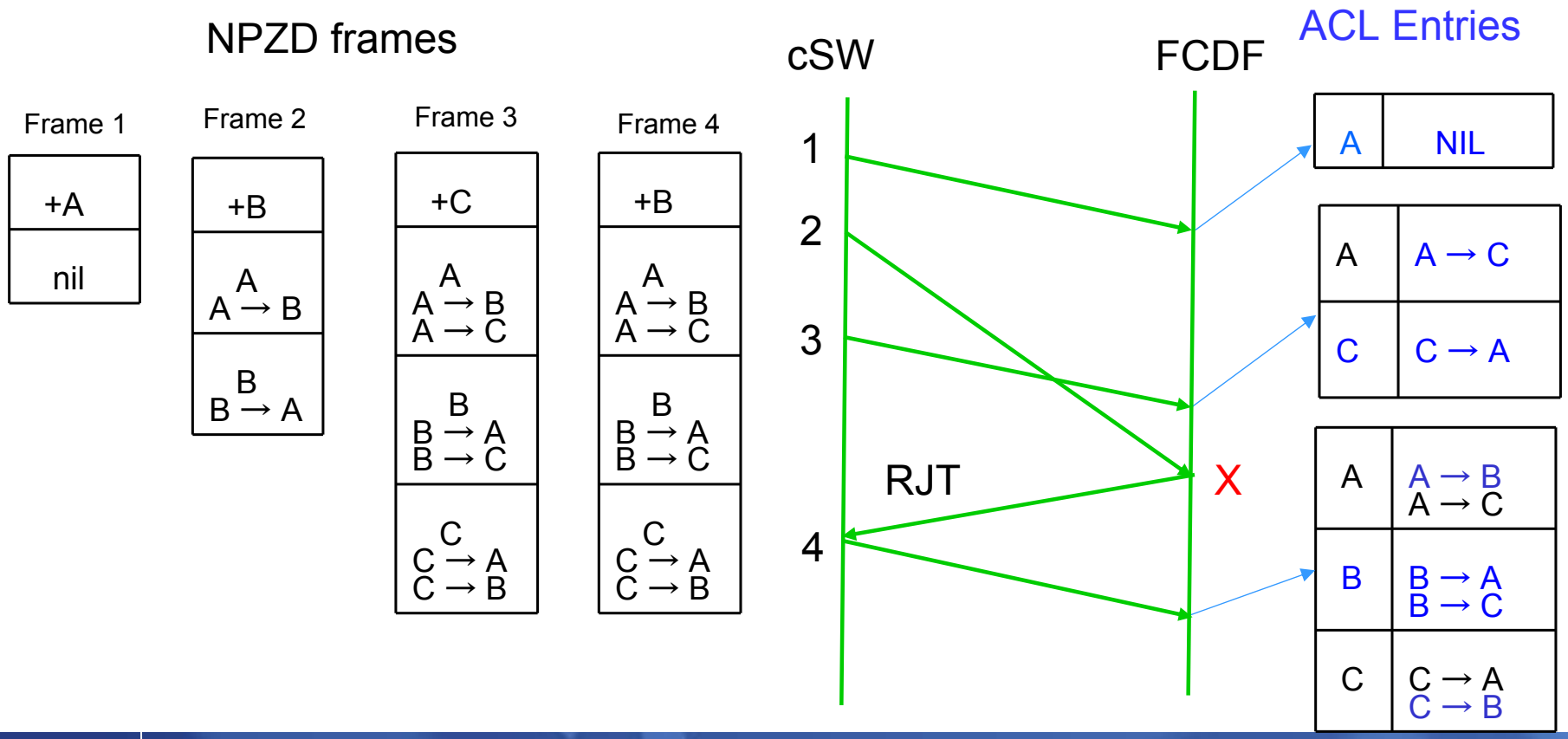
Minimizing Rejects - implementation flexibility

NOT PART OF THIS PROPOSAL

- FCDF may detect that a request has a sequence number which is more than one higher than a previously processed sequence number
  - Before processing such a request FCDF may wait a small period of time (e.g. 10ms) for requests with missing sequence numbers to arrive
  - If all requests with missing sequence numbers arrive before said wait time, requests are processed in sequence number order.
  - If said time expires, request that have been received are processed in sequence number order.

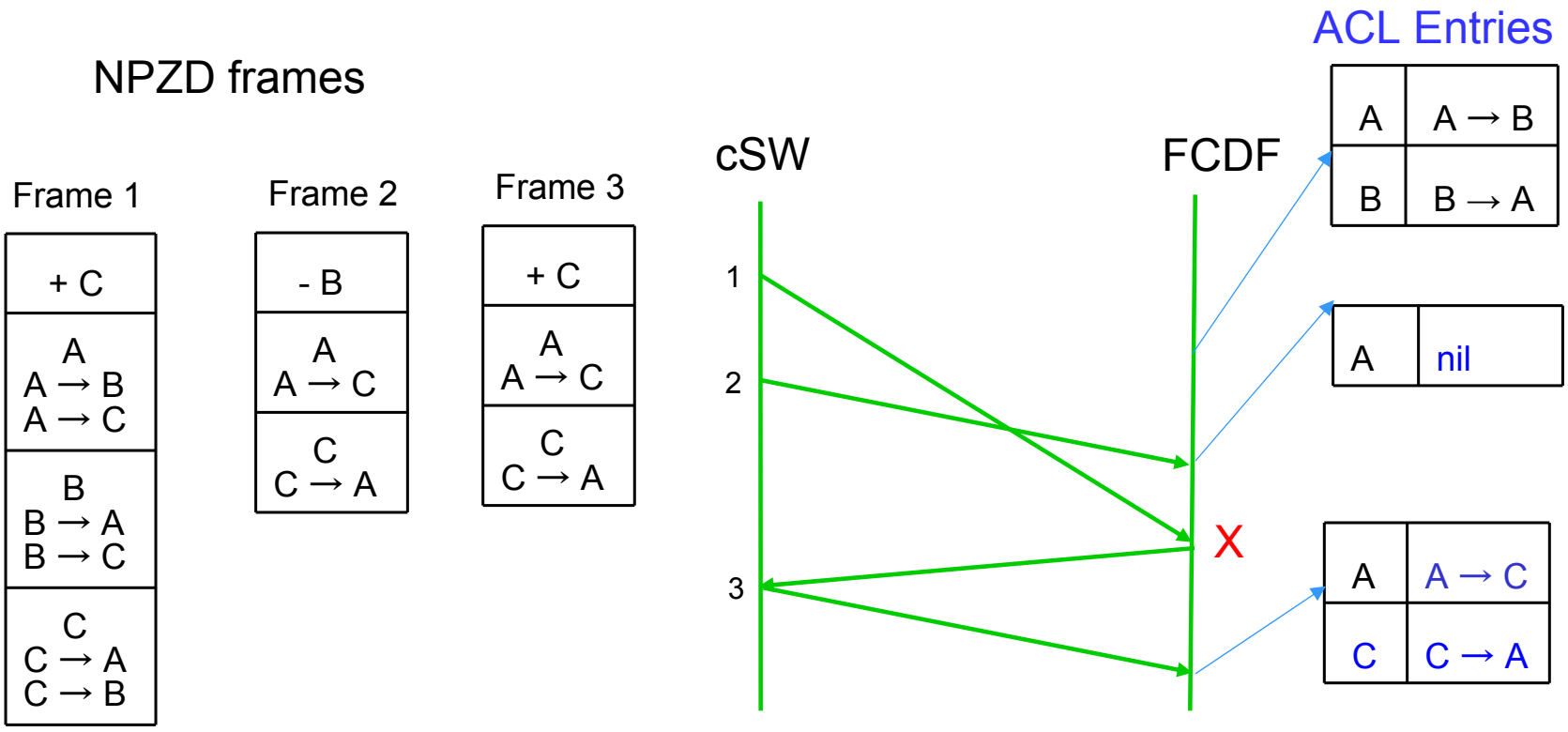
Case 1 … (frame 2 and 3 received out-of-order)

- When FDF receives frame 3, it is accepted as it is higher than the previous seq #.
- Frame 2 is rejected by FCDF. cSW recomputes and retransmits NPZD as frame 4 (Seq # 4).



Case 2 … (frame 1 and 2 received out-of-order)

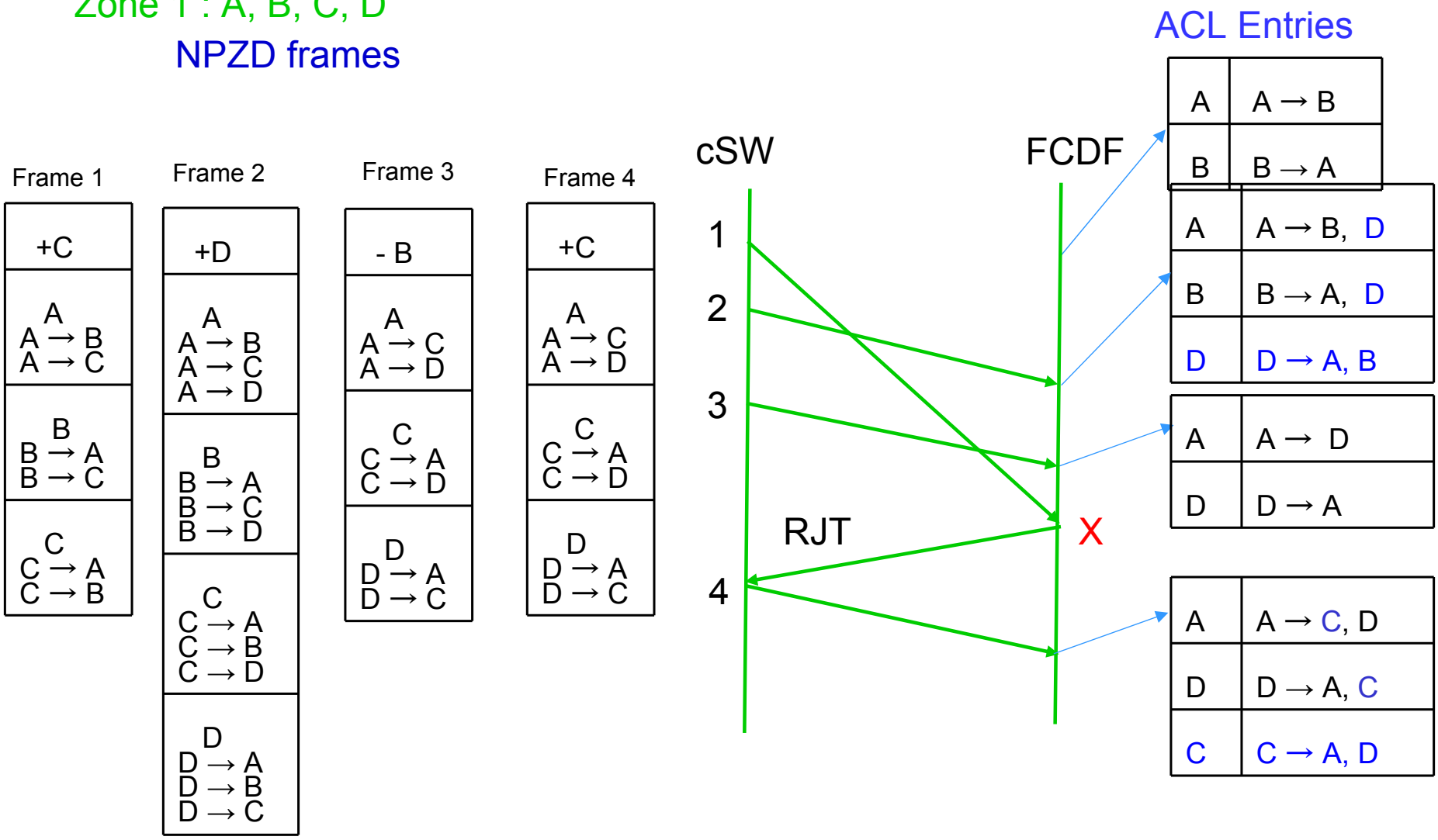
- A and B are logged in previously
- When frame 2 is received, peering entry with Principle N_Port_ID C is ignored because C is unknown. Peering Entry for Principle A contains a peer for C which is also ignored.
- Frame 1 is rejected by FDF. cFCF recomputes and retransmits NPZD as frame 3 (Seq # 3).



Case 3 – two frames later

Zone 1 : A, B, C, D

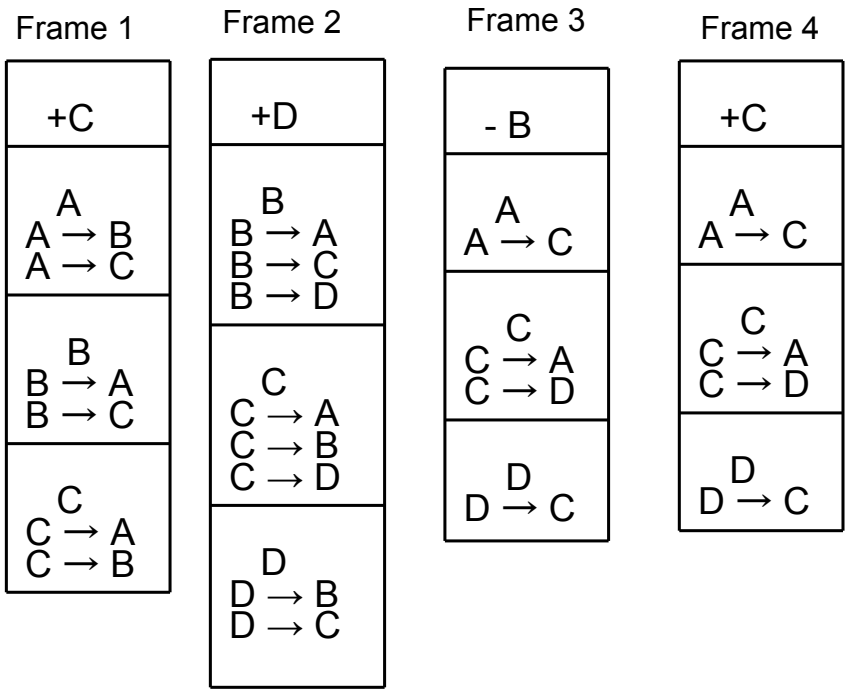
NPZD frames

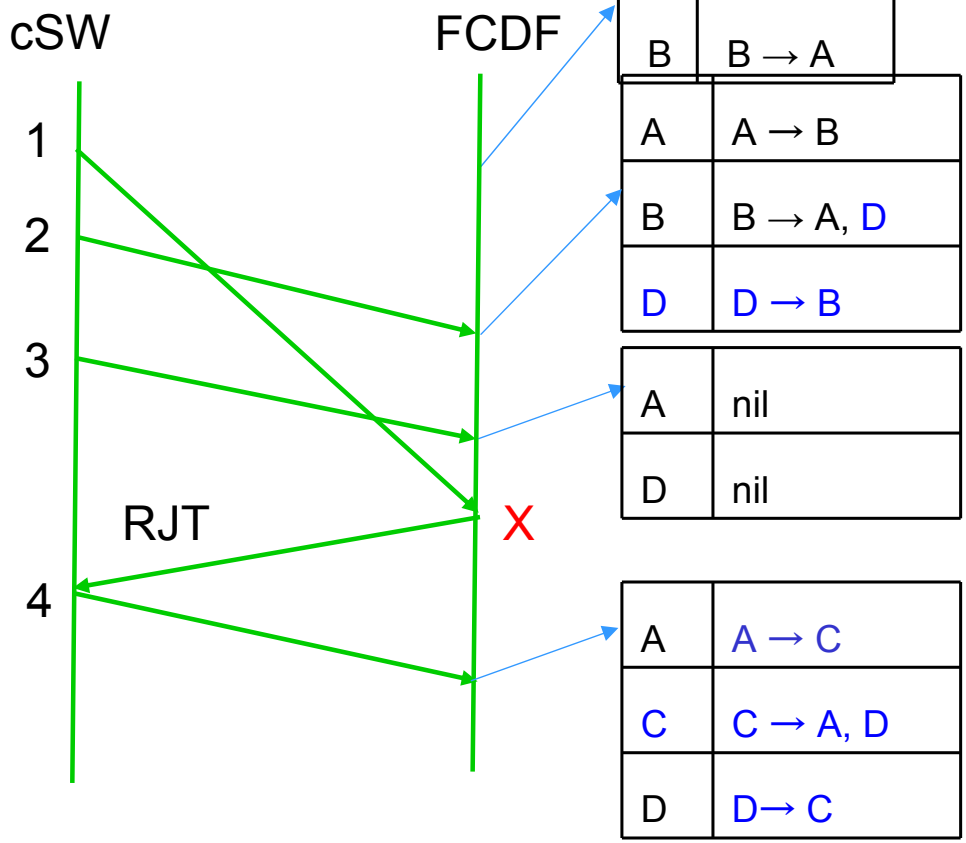


Case 4 (A and D are not in the same zone)

Zone 1 : A, B, C
Zone 2 : B, C, D

NPZD frames





Proposal

- **Add a SW_RJT Reason code explanation to FC-SW-6 for**
  Out of Order (e.g. '5D'h)
- **Add in the appropriate place (probably not where the Sequence Number Descriptor is defined):**

  A controlling switch maintains a sequence number for each FCDF in the FCDF set. The sequence number is incremented by one and included in a sequence number descriptor each time an NPZD is sent.

Proposal

- **In 12-035v2, section 1.2.3 N_Port_ID Handling, add:**

  Upon receipt of an NPZD request, an FCDF compares the sequence number in the received sequence number descriptor to that of the last processed NPZD request, or 00000000_00000001 if no NPZD has previously been processed. If the received sequence number is lower, the NPZD request shall be discarded and a VA_RJT shall be sent with Reason Code of Logical Error and Reason Code Explanation of Out of Order Sequence. If the received sequence number is higher, then the NPZD is processed.

  An FCDF considers an N_Port ID to be allocated when it has successfully received the N_Port ID in an Allocation Entry of the current or previous NPZD. If an NPZD request contains a peering entry with a Principle N_Port ID that has not been allocated, that entire peering entry shall be ignored. If an NPZD request contains a peering entry with a Principle N_Port ID that is currently allocated, but that peering entry contains Peer N_Port_ID that has not been allocated that Peer N_Port ID shall be ignored.

  Whenever an NPZD request is retried for any reason (e.g., timeout) the Zoning ACLs for the affected N_Port_IDs shall be recomputed and a new NPZD request including a new sequence number and the newly computed peering entries shall be sent.

  If a Primary Controlling Switch receives a VA_RJT with a Reason Code of Logical Error and Reason Code Explanation of OUT_OF_ORDER in response to an NPZD request, the Primary Controlling Switch shall retry the NPZD request.

Other things to think about

- **How should NPRD and NPZD be serialized with NPZD?**
  - Current thought is
    - an NPRD request shall not be sent when there are outstanding NPZD or AZAD requests to an FCDF.
    - An NPZD request shall not be sent when there are outstanding NPRD or AZAD requests to an FCDF
    - An AZAD request shll not be sent when there are outstanding NPRD or NPZD requests.
- **What happens when a controlling switch sends a peering entry to an FCDF with a Peering Entry containing a Principle N_Port_ID that is not allocated to (logged in through) that FDF? (But, it is allocated to a different FDF)**
  - Ignore it?
  - Flag an error?
  - Reject it?

Thank you