802.1AX -- Link Aggregation:

Editor's Report: November 2017

Version 1

Stephen Haddock

November 9, 2017

Preparing 802.1AX-Rev-d0.2

- AX-Rev-d0.1 went to Task Group ballot in June and comment resolution in July
  - Major changes were in Clause 6 (LACP) and specifically relating to Conversation Sensitive Collection and Distribution (CSCD).
  - Clause 9 (DRNI and DRCP) was mostly untouched.
- AX-Rev-d0.2:
  - Incorporates comment resolutions from first Task Group ballot.
  - Major changes in Clause 9 (DRNI and DRCP)
    - The rest of this presentation summarizes those changes
  - No changes yet to MIB or PICS
    - Do this for Working Group ballot
  - Plan to have second Task Group ballot shortly after November meeting (?)

Distributed Relay

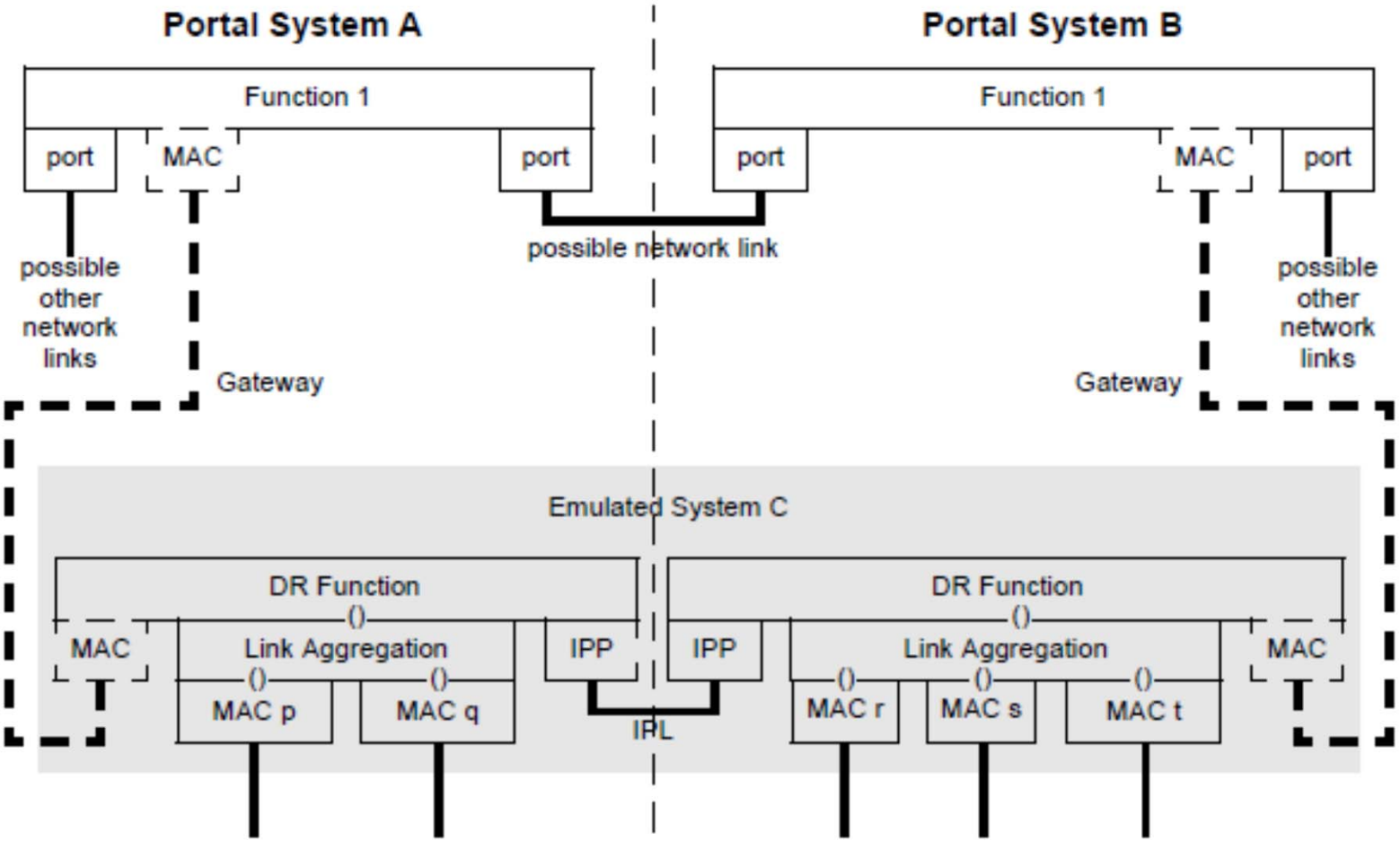
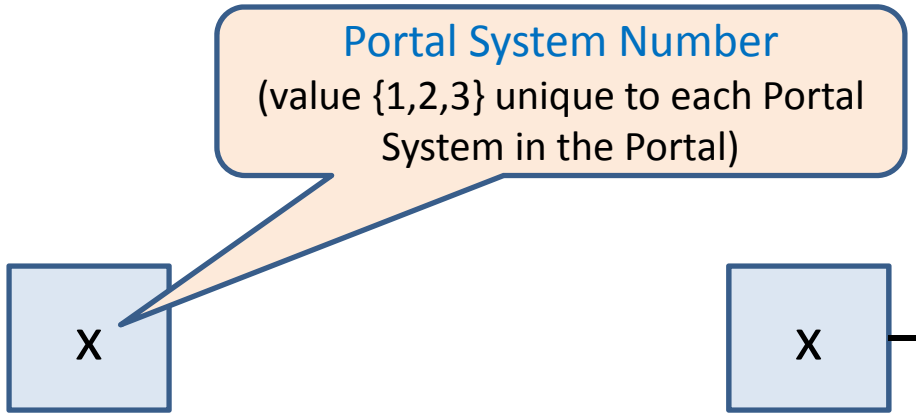


Figure 9-3—Distributed Relay: as seen by Systems A and B

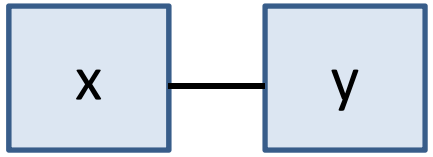
Distributed Relay Control Protocol

- DRCP operates between peer Distributed Relay Functions in Portal Systems connected by an Intra-Portal Link (IPL).
- The objectives of the protocol are:
  1. Exchange Portal System configuration information to validate the Portal topology formed by activating the Intra-Portal Link(s).
  2. When a valid Portal topology can be formed, coordinate the transition from multiple Stand-Alone Portal Systems to a single Portal (Emulated System).
  3. Exchange Portal System state information so that each Portal System develops its own view of the state of the entire Portal, and understands its Neighbors' view of the state of the entire Portal.
  4. Using the configuration and state information , coordinate how data frames are to be forwarded between the Gateway Port, Aggregator Port, and Intra-Portal Port(s) at each Distributed Relay Function.

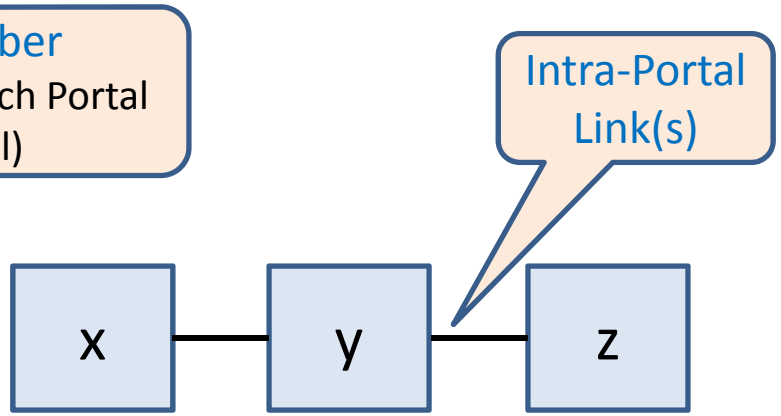
Valid Portal Topologies



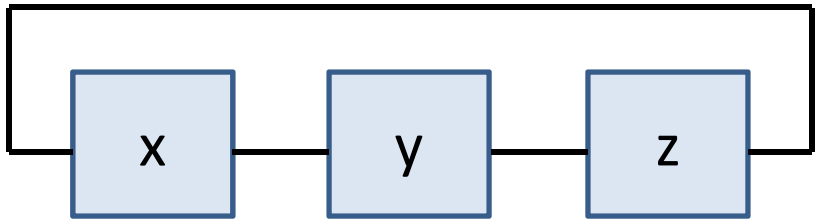
1. Single Portal System



2. Two Portal Systems

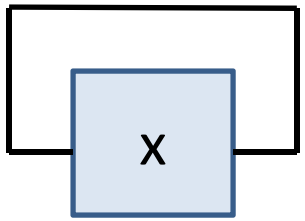


3. Chain of three Portal Systems

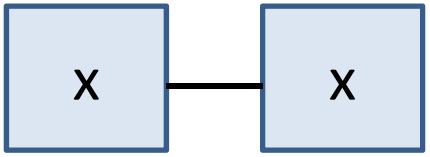


4. Ring of three Portal Systems

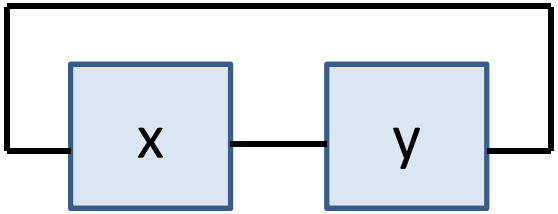
Portal Topology Errors



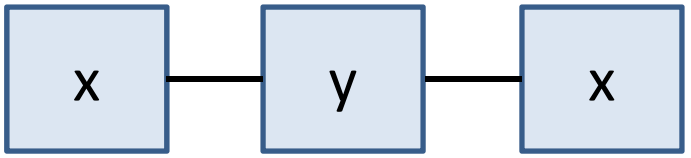
1. Loopback Portal System



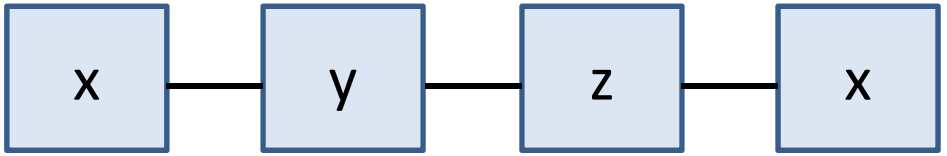
2. Duplicate Portal System Number



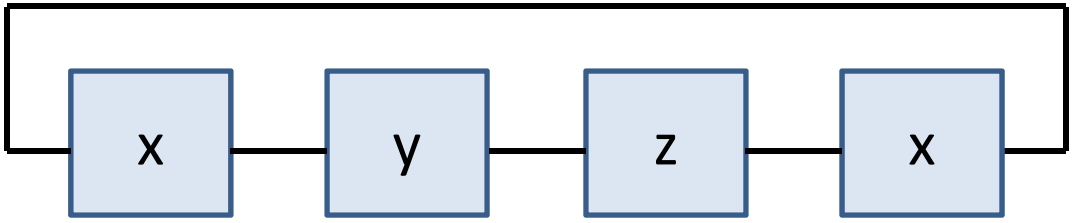
3. Ring of two Portal Systems



4. Duplicate Portal System Number



5. Chain of four or more Portal Systems



6. Ring of four or more Portal Systems

Portal Topology Errors (Changes from 802.1AX-2014)

- AX-2014 will not activate an IPP if the Neighbor Portal System Number (PSN) does not match a preconfigured value in the Home Portal System.
  AX-Rev learns the Neighbor PSN, and will not activate an IPP if the Neighbor PSN is zero, is the same as the Home PSN, or is the same as the Neighbor PSN at the other IPP on the Home Portal System. Simplifies configuration, eliminates a cabling error condition, and eliminates variables:
  - DRF_Home_Conf_Neighbor_Portal_System_Number
  - DRF_Neighbor_Conf_Portal_System_Number
  - Differ_Conf_Portal_System_Number
- AX-Rev changes the definition of Drni_Portal_System_Addr to be unique to each Portal System (rather than common for all Portal Systems in the Portal), and verifies that the Neighbor Portal System Address at one IPP does not match the Home Portal System Address or the Neighbor Portal System Address at the other IPP.
- AX-Rev transmits in DRCPDUs at one IPP the Neighbor PS Address learned at the other IPP. The receiving Portal System can then verify that this value matches the Neighbor PS Address learned at the other IPP of the receiving system, which differentiates a ring of three from a chain or ring of more than three Portal Systems.

Handling Portal Topology Errors (Changes from 802.1AX-2014)

- 802.1AX-2014 only detected errors in the configuration of immediate Neighbors at an IPP, and prevented that IPP from being incorporated into a Portal. The result is that the Portal Systems continued to operate as Stand-Alone systems or, if the other IPP was active, as disjoint Portals. 802.1AX-Rev will prevent all IPPs from being incorporated into a Portal when a topology error is detected, resulting in all the involved Portal Systems operating as Stand-Alone systems.
- If the topology error involves multiple Portal Systems with the same PSN, then (with both AX-2014 and AX-Rev) the Link Aggregation Partner will think they are a single system and will aggregate the links.
  - Should this be prevented by prohibiting any data forwarding through any Portal System when a topology error is detected?
  - Problem with not forwarding any data when a topology error is detected is the observed behavior of having a subset of the topology that appears to be working (even if incorrectly), but suddenly stopping when the final IPL is connected. Provides an incentive to simply leave the IPL unconnected.

Aggregator variables

- AX-2014 specifies a number of DRF variables that are simply copies of Aggregator variables.
  - Maintaining separate copies requires specifying how/when one variable tracks the other, and what to do if they differ.
  - These could be interpreted as “aliases” rather than copies, but the Editor’s opinion is that using aliases results in confusion.
  - AX-Rev eliminates these variables and directly references the equivalent Aggregator variables.
- AX-2014 has several Admin_xxx variables to initialize operational values that can be initialized to fixed default values.
  - AX-Rev eliminates these administrative variables.

Aggregator variables eliminated

Per-DR Function variables eliminated:

- Drni_Aggregator_Priority
- Drni_Aggregator_ID
- DRF_Home_Admin_Aggregator_Key
- DRF_Home_Oper_Aggregator_Key
- DRF_Home_Port_Algorithm
- DRF_Home_Conversation_PortList_Digest
- DRF_Neighbor_Admin_Gateway_Algorithm
- DRF_Neighbor_Admin_Conversation_GatewayList_Digest
- DRF_Neighbor_Admin_Port_Algorithm
- DRF_Neighbor_Admin_Conversation_PortList_Digest

Detecting Neighbor has stale data

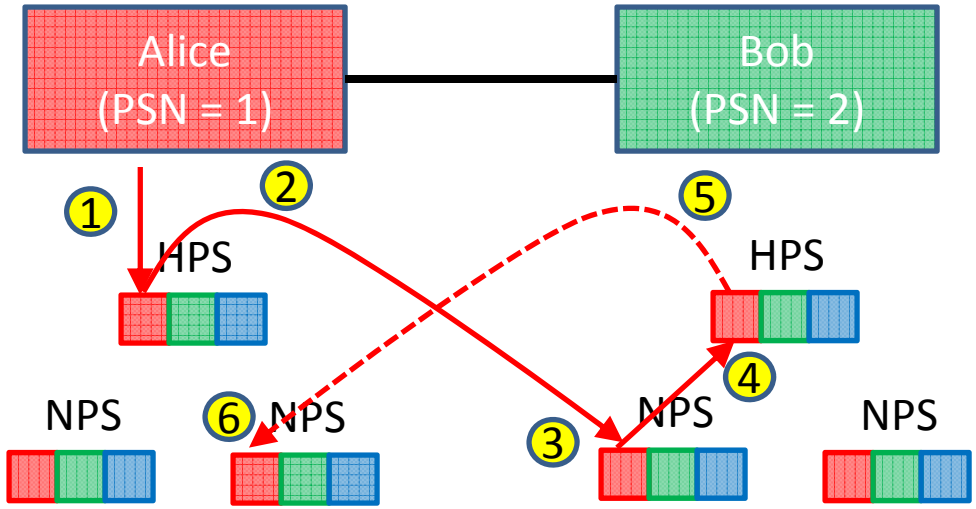
- AX-2014 stores several Neighbor Portal System parameter received in a DRCPDU, and sets flags if these do not match the corresponding Home Portal System parameters, but does not echo the Neighbor values or send the match flags in DRCPDUs. One or the other (echo or flags) need to be included in the DRCPDUs in order for the Portal System originating the parameters to know when the Neighbor has stale data.
- AX-Rev computes the match flags based on the received parameters, and sends these flags in transmitted DRCPDUs, but does not store the Neighbor parameter values. Allows detection of stale data while eliminating variables:
  - DRF_Neighbor_Aggregator_Priority
  - DRF_Neighbor_Aggregator_ID
  - DRF_Neighbor_Gateway_Algorithm
  - DRF_Neighbor_Port_Algorithm
  - DRF_Neighbor_Conversation_GatewayList_Digest
  - DRF_Neighbor_Conversation_PortList_Digest

Portal State Variable

		Portal System Number		
Field	Type	1	2	3
Gateway Operational	Boolean			
Gateway Sequence Number	32-bit unsigned integer			
Gateway Vector	Boolean Vector (4096)			
Active Aggregation Links	List of Link Numbers			

- The Portal State variables contain Gateway and Port (Aggregation Link) state information for all Portal Systems.
  - **Drni_Portal_System_State[]** contains the Home Portal System's view of the Portal State.
  - **Drni_Neighbor_State[]** at each IPP contains the Neighbor Portal System's view of the Portal State.

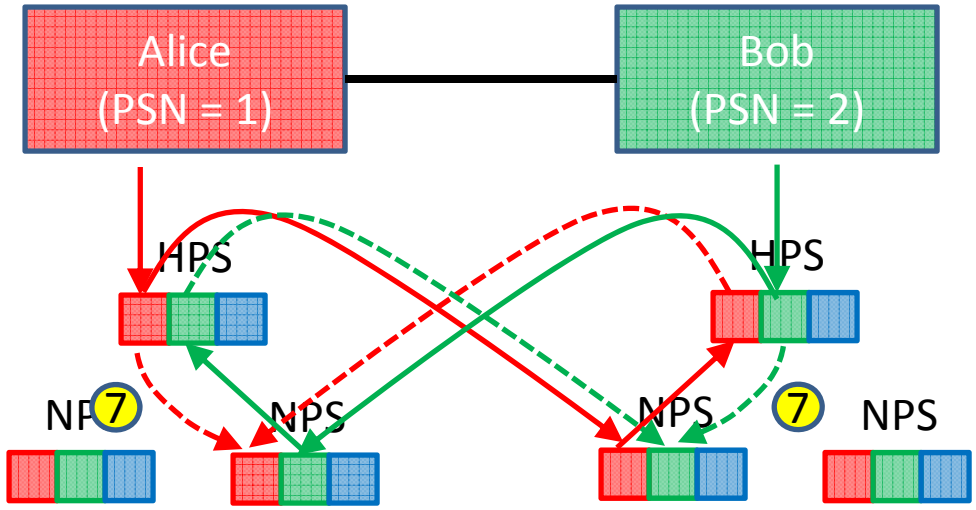
Communicating Portal State (Two Portal Systems)



Home Portal State (HPS) = Drni_Portal_System_State[]
Neighbor Portal State (NPS) = Drni_Neighbor_State[]

1. The home state for Alice (Portal System Number (PSN) = 1) gets stored in Alice's Drni_Portal_System_State[] variable.
2. Alice transmits the contents of Drni_Portal_System_State[] in a DRCPDU.
3. Bob stores the received information in that IPP's Drni_Neighbor_State[].
4. Bob copies Alice's state to Bob's Drni_Portal_System_State[].
5. Bob transmits the contents of Drni_Portal_System_State[] in a DRCPDU.
6. Alice stores the received information in that IPP's Drni_Neighbor_State[].

Communicating Portal State (Two Portal Systems)

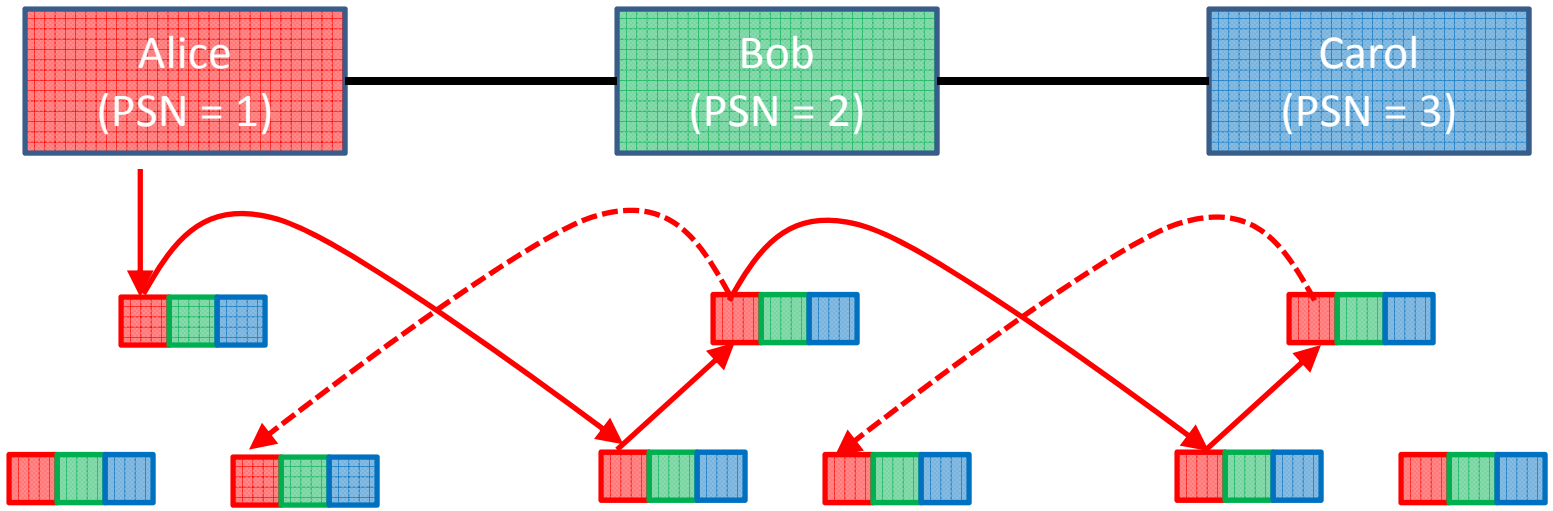


The flow of the home state information for Bob (PSN = 2) is the reciprocal of the flow of the home state information for Alice.

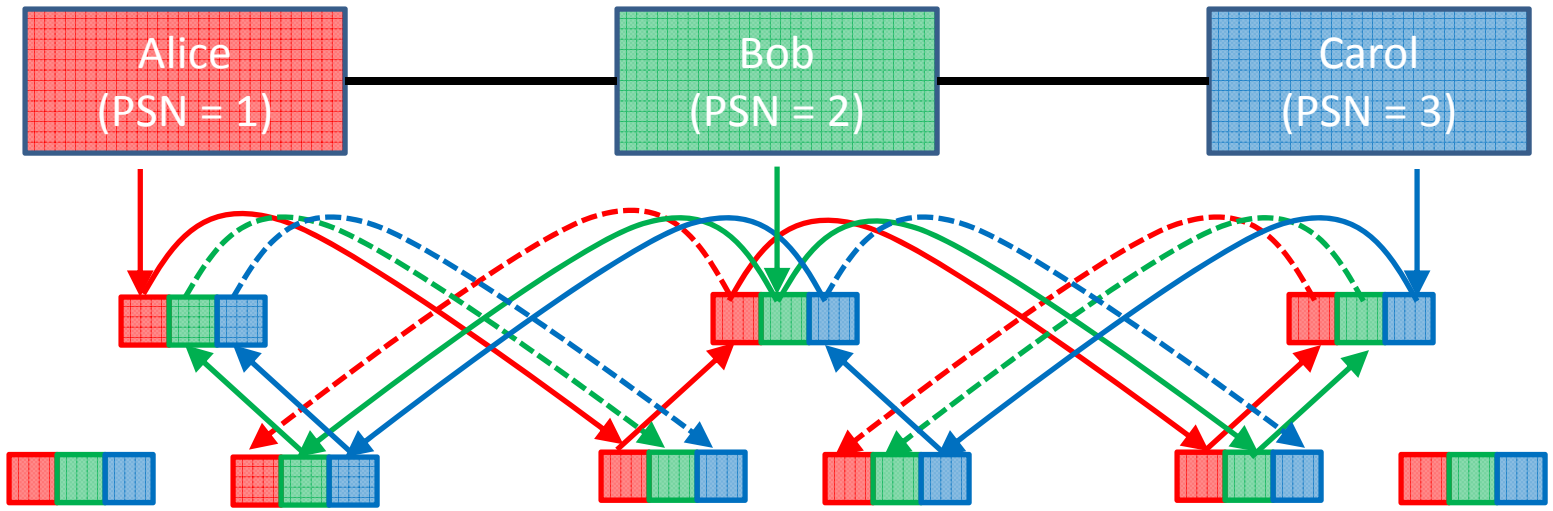
- Every DRCPDU includes the contents of the Drni_Portal_System_State[] variable for all Portal Systems in the Portal, except ….
- Data being "echoed" back to the original source of the data (e.g. data for PSN = 1 being sent in a DRCPDU from Bob to Alice) does not include the Gateway Vector. This case is represented by the dashed lines in the diagram.

7. When a Portal System receives "echoed" data from its Neighbor, it fills in the Gateway Vector corresponding to the received Gateway Sequence (not shown in subsequent diagrams).

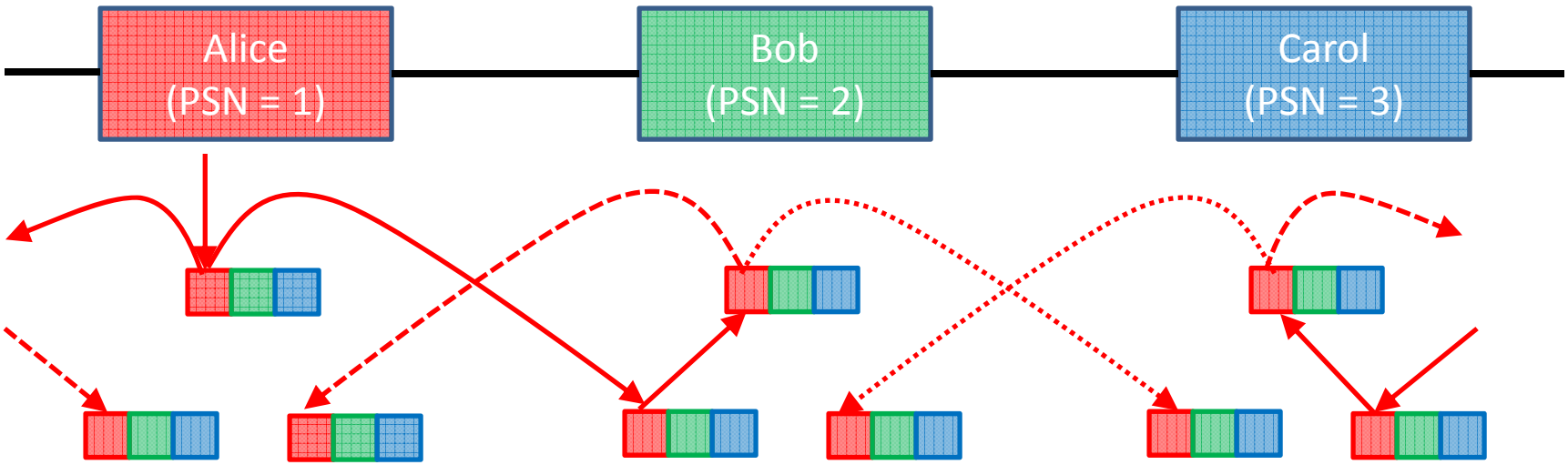
Communicating Portal State (Three Portal System Chain)



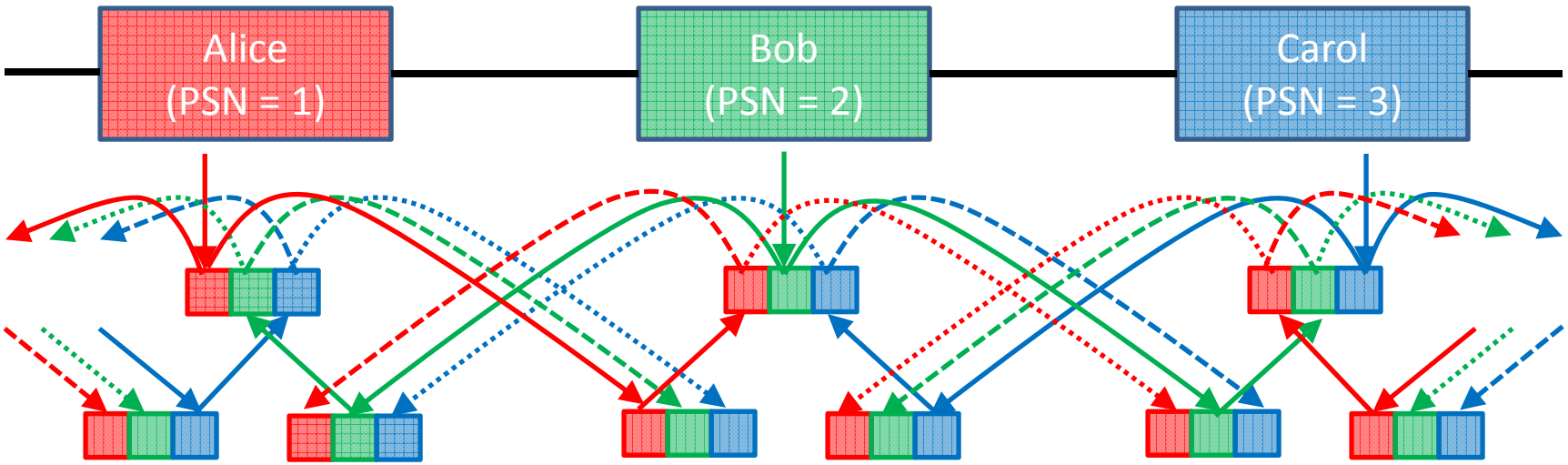
Communicating Portal State (Three Portal System Chain)



Communicating Portal State (Three Portal System Ring)



Communicating Portal State (Three Portal System Ring)



Communicating Portal State (Changes from 802.1AX-2014)

- In 802.1AX-2014, the process of storing data in the Portal State variables involved moving the data through many intermediate variables (with very similar names).
- In 802.1AX-Rev-d0.2, data from the received DRCPDU is stored directly into Drni_Neighbor_State[], resulting in the elimination of many per-IPP variables (see next slide).
- In 802.1AX-Rev-d0.2, the DRCP Receive state machine is only responsible for storing data received in a DRCPDU to the Drni_Neighbor_State[]. Copying the state regarding the immediate Neighbor to the Drni_Portal_System_State[], filling in the Gateway Vector for echoed data, and maintaining the history of Gateway Sequence Numbers and Gateway Vectors, is handled by the updatePortalState() function of the Portal System state machine.

Portal State variables eliminated in 802.1AX-Rev-d0.2

Per-IPP variables:

- DRF_Rcv_Neighbor_Gateway_Conversation_Mask
- DRF_Rcv_Neighbor_Gateway_Sequence
- DRF_Neighbor_Gateway_Conversation_Mask
- DRF_Neighbor_Gateway_Sequence
- DRF_Neighbor_State
- DRF_Rcv_Other_Gateway_Conversation_Mask
- DRF_Rcv_Other_Gateway_Sequence
- DRF_Other_Neighbor_Gateway_Conversation_Mask
- DRF_Other_Neighbor_Gateway_Sequence
- DRF_Other_Neighbor_State
- DRF_Rcv_Home_Gateway_Conversation_Mask
- DRF_Rcv_Home_Gateway_Sequence
- Ipp_Portal_System_State[]

Per-DR Function variables

- DRF_Home_Gateway_Conversation_Mask
- DRF_Home_Gateway_Sequence
- DRF_Home_State

Additional Aggregator Port State

- To verify that all Portal Systems are connected to the same LACP Partner, it is necessary to verify that the Aggregator attached to each DR Function has the same:
  - Partner_System_Priority
  - Partner_System
  - Partner_Oper_Aggregator_Key
- AX-2014 transmitted the last of these in DRCPDUs, but not the first two. AX-Rev corrects this.

Handling different LACP Partners

- AX-2014 specifies that if the Portal Systems have different LACP Partners, only the Aggregation Ports with the lowest Partner Key value among the Portal Systems are allowed to attach to the Aggregator. This is problematic:
  - Implies the LACP Partner information for each Portal System is transmitted in DRCPDUs before the Aggregator Ports on the Portal Systems have selected an Aggregator.
  - Choosing the lowest Partner Key value does not correspond to the recommended default selection logic.
- AX-Rev currently allows Aggregation Ports on a Portal System to select an Aggregator without knowledge of the state of other Portal Systems. If the Portal Systems have different LACP Partners, DRCP selects one of the Partners and only the Aggregation Links to that Partner are used for forwarding data by the Distributed Relay.
  - This is not an ideal solution.
  - Hopefully a better solution can be found. TBD.

Port Algorithms

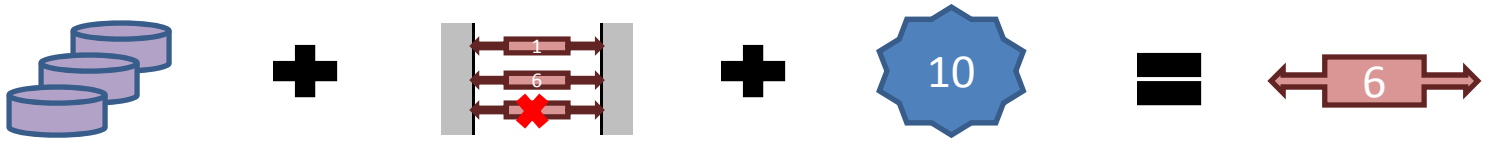
- The Aggregator associated with each DR Function uses Conversation Sensitive Distribution to distribute frames among the Aggregation Ports on that Aggregator.
- The DR Function in each Portal System uses the same Conversation Sensitive Distribution mechanism to forward frames to one of the Aggregators.
  - The managed objects controlling the distribution mechanisms need to be exchanged in DRCPDUs to verify that the Portal Systems have consistent configuration:
    - Actor_Port_Algorithm
    - Actor_Conversation_Service_Mapping_Digest
    - Actor_Conversation_LinkList_Digest
  - AX-2014 did not include Actor_Conversation_Service_Mapping_Digest. AX-Rev corrects this.

Conversation Sensitive Distribution Overview

1: For each egress frame, associate the frame with a Conversation ID:

Managed Objects | Frame Contents | Conversation ID

2: Select the LAG Link:



(more) Managed Objects | Current operational state of LAG links | Conversation ID | Selected Link

3: Transmit the frame on the selected link:

Managed Objects for Step 1

- Actor_Port_Algorithm
  - Per-Aggregator configuration (read/write) variable
    - a.k.a. aAggPortAlgorithm in clause 12
    - a.k.a. dot3adAggPortAlgorithm in MIB
  - Specifies which fields in the frame are used, and the mechanism to derive a 12-bit Conversation ID from those fields.
  - Some port algorithms do this in two steps:
    a) Derive a "Service ID" (up to 32-bit value) from fields in the frame.
    b) Use the Admin_Conversation_Service_ID_Map to map the Service ID to the Conversation ID.

Managed Objects for Step 2

- Admin_Conversation_Link_Map
  - Per-Aggregator configuration (read/write) variable
    - Basically a table with 4096 rows (one per Conversation ID).
    - Each row has a list of link numbers. The first link number in the list that identifies a currently active link in the LAG will be used as the selected link for that Conversation ID.
  - A MD-5 digest of the table is one of the values conveyed in LACPv2 PDUs so that the actor and partner systems can determine if they are using the same table.
  - The link number together with the Aggregator identifier uniquely identify the Aggregation Port through which a frame is transmitted or expected to be received.

Port Algorithms (cont.)

- AX-2014 specifies that all Portal Systems have to use the same Port Algorithm in order to pass any data.
  - This is harsh. Especially when the difference is a due to changing the Actor_Conversation_Service_ID_Map, data should continue to flow for Conversations that have matching configuration.
  - AX-Rev specifies that when the Port Algorithms or Service ID Map Digest do not match, each DR Function will forward all “down” frames to the Aggregator attached to that DR Function.
- AX-2014 does not specify any support for Conversation Sensitive Collection.
  - AX-Rev exchanges the Aggregator’s Discard_Wrong_Conversation flag in DRCPDUs. If these flags are TRUE for all Portal Systems, “up” frames will be discarded if they are received from a different Aggregator than would be selected for a “down” frame with the same Port Conversation ID.

Port_Algorithms (cont.)

- AX-2014 specifies a set of Port_Conversation_Vector_TLVs to be exchanged in DRCPDUs when Portal Systems have Differ_Port_Digest TRUE in order to detect forwarding conflicts.
  - This is not strictly necessary. When the digests differ, neither the LACP Actor nor Partner operate with Discard_Wrong_Conversation TRUE, so it doesn't matter which Aggregator (or Aggregator Port) is selected.
  - AX-Rev could eliminate this TLV and have each DR Function forward all "down" frames according to it's own distribution algorithm. Any "down" frames received at an IPP that the distribution algorithm would forward to that IPP will be forwarded to the Aggregator instead. (TBD)
- When Portal Systems have different port algorithms (and possibly when they have different port digests), the distribution algorithm advertised to the LACP partner should be "Unspecified".
  - AX-Rev will specify a mechanism for this (TBD).

Gateway Algorithms

- AX-2014 specifies that all Portal Systems have to use the same Gateway Algorithm in order to pass any data.
  - This is harsh. Especially when the difference is a due to changing the Gateway_Service_ID_Map, data should continue to flow for Conversations that have matching configuration.
  - AX-Rev specifies that when the Gateway Algorithms or Service ID Map Digest do not match, the Gateway in a single Portal System will be selected for all Gateway Conversation IDs.
- AX-2014 specifies a set of Gateway_Conversation_Vector_TLVs to be exchanged in DRCPDUs when Portal Systems have Differ_Gateway_Digest TRUE in order to detect forwarding conflicts.
  - AX-Rev reduces this to a single Gateway_Conversation_Vector_TLV containing a Boolean vector (indexed by Gateway Conversation ID).
  - In AX-2014, Differ_Gateway_Digest modified several functions and intermediate variables in determining the final forwarding rules for the DR Function. In AX-Rev the effect of Differ_Gateway_Digest is confined to the final step in determining the forwarding rules.

Port and Gateway Algorithms

- AX-2014 calculates Differ_Gateway_Digest and Differ_Port_Digest flags, but does not distribute these in DRCPDUs.
- AX-Rev does distribute Differ_Gateway_Algorithm, Differ_Gateway_Digest, Differ_Port_Algorithm, Differ_Port_Digest flags.
  - When a Portal System receives a flag from the Neighbor that doesn't match its own, it means the Neighbor has stale data from this Portal System. In this case NTT is set to send a DRCPDU with fresh data.
  - Allows Portal System to defer actions in response to a change in one of these flags until both Portal Systems have matching flag values.

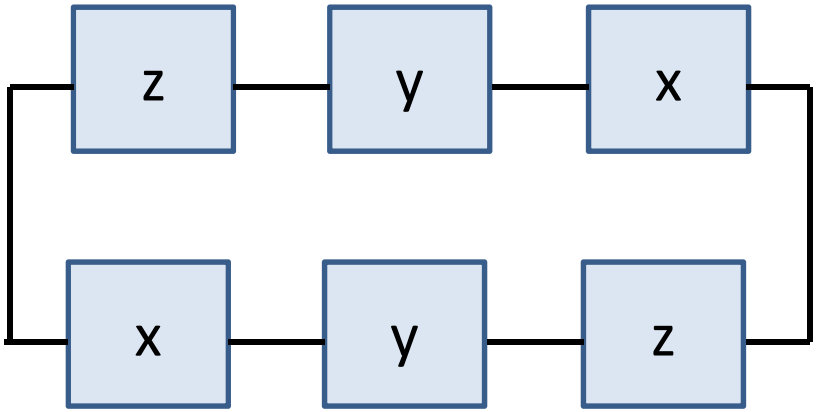
Backup Slides

Misc

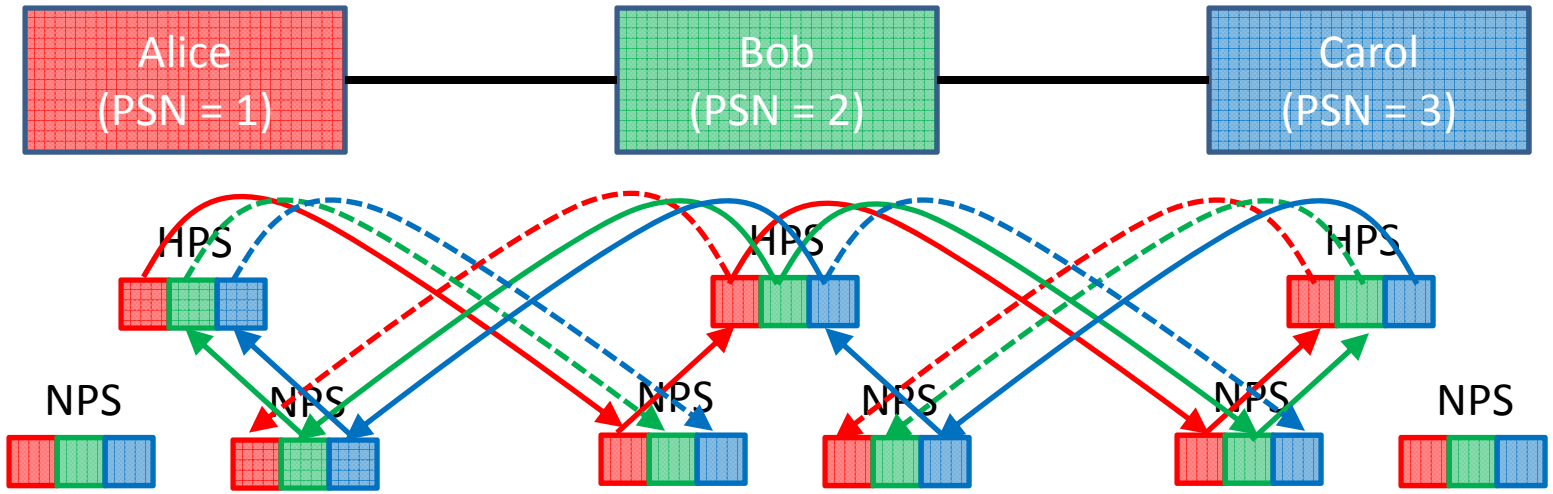
- Variable name changes
- Strikeout/underscore
- Haven’t touched Network/IPL sharing
- Plan to update clause 7, but not touching MIB yet
  - Do that for Working Group Ballot
- All comments welcome, but primarily interested in whether Clause 9 is readable/comprehensible.
- Editing diagrams

Portal Topology Errors

(Requires System Identifier of Portal Systems to detect)



7. Ring of multiples of three Portal Systems


Alice
(PSN = 1)
Bob
(PSN = 2)
Carol
(PSN = 3)
HPS
HPS
HPS
NPS
NPS
NPS
NPS
NPS
NPS