802.1AX -- Link Aggregation:

Conversation Sensitive Collection and Distribution (CSCD)

Version 2

Stephen Haddock

May 15, 2017

Distribution/Collection Background

- The primary value proposition of Link Aggregation:
  1. **Resiliency**:
     - Upon failure of one link in the Link Aggregation Group (LAG), any traffic streams using that link are quickly moved to other links in the LAG with no or minimal disruption to higher layers.
  2. **Load-Sharing**:
     - Take advantage of the bandwidth available on all links of the LAG.
- Accordingly, the primary data plane functions are:
  1. **Distribution**:
     - Accept frames for transmission from the higher layer and distribute them among the active links in the LAG.
  2. **Collection**:
     - Receive frames from the links in the LAG and collect them into a single traffic stream for delivery to the higher layer.

Traditional Collection

- Any frame received on an active LAG link is passed from the Aggregation Port to the Aggregator to which it is attached, and from there to the higher layer.
  - Collector will maintain the relative order of any frames received on the same link, but makes no guarantee to the relative order of frames received on different links.
- Advantages:
  1. No modification of the frames is required (for example, no sequence numbers are added).
  2. The receiving system does not need to know anything about how the transmitting system distributes frames.
- Disadvantages:
  1. It was necessary to specify the Marker Protocol to maintain the relative order of a sequence of frames when traffic is redistributed (i.e. the LAG link used for that sequence of frames is changed) in response to changes in the operational status of the links in the LAG.

Traditional Distribution

- The method used by the distributor to determine on which LAG link a given frame will be transmitted is not specified in the standard.
  - The only requirement is that any sequence of frames that must have their relative order maintained must be transmitted on the same link.
    - Such a sequence of frames is defined to be a "conversation".
- All details of the distribution algorithm are left to the LAG system implementer.
  - It is up to the implementer to determine what frames must have their relative order maintained (i.e. what frames are part of the same conversation).
  - It is up to the implementer to decide what conversations get transmitted on each link.
  - Typically this is done by calculating a hash of fields from the frame header (e.g. some combination of MAC-SA, MAC-DA, VLAN-ID, Ethertype, IP-SA, IP-DA, IP-Protocol, TCP-Source-Port, TCP-Destination-Port).
  - Typically the hash algorithm is not known to the network administrator.
  - Typically there are no managed objects allowing administrative control over the distribution algorithm.

Traditional Distribution (cont.)

- Advantages
  1. Link Aggregation systems do not need to use the same distribution algorithm, or to know anything about the partner's distribution algorithm, to interoperate.
  2. Simplifies adding Link Aggregation as a new feature on existing systems.
- Disadvantages
  1. The resulting load-sharing between the links in the LAG may be very imbalanced.
  2. It is difficult or impossible for a network administrator to predict or control which frames are transmitted on which links.
     - This complicates traffic management and monitoring.

Conversation Sensitive Collection and Distribution (CSCD)

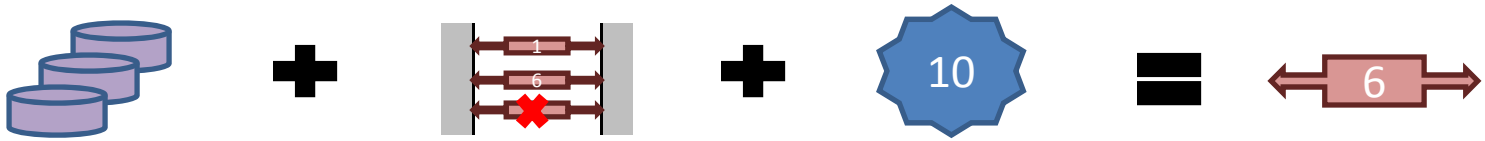
- The primary value proposition of Link Aggregation Control Protocol version 2 (LACPv2) is:
  1. The addition of managed objects and processes that allow administrative control over the distribution algorithm in use:
     a) What frames comprise a conversation, and
     b) How conversations are mapped to active links in the LAG.
  2. TLVs in the LACPDUs that allow a system to convey to it's partner system what distribution algorithm is in use.
     - Allows receiving system to only accept frames received on the expected link.
- Advantages
  - Provides an alternative to the Marker Protocol.
  - Simplifies traffic management (provisioning and policing).
  - Simplifies service monitoring (connectivity and performance).
  - Enables some topology options for DRNI.

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:

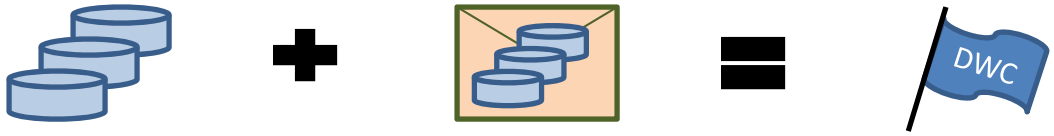


(more) Managed Objects | Current operational state of LAG links | Conversation ID | Selected Link

3: Transmit the frame on the selected link:

Conversation Sensitive Collection Overview

0: LACP determines whether to collect only from the expected LAG link:

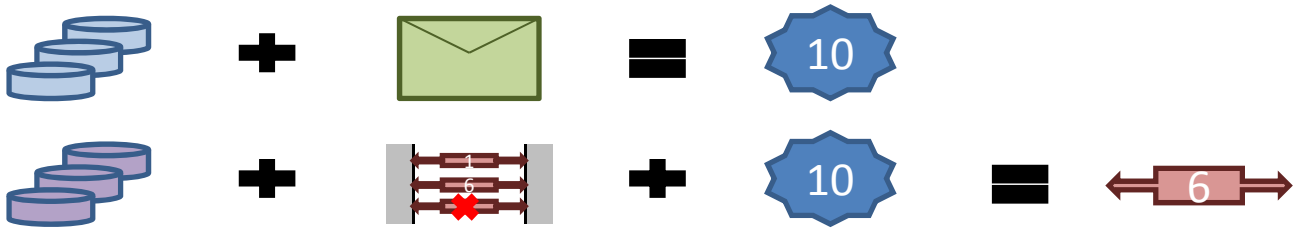


Managed Objects

Partner Managed Objects from received LACPDU

"Discard Wrong Conversation" Flag

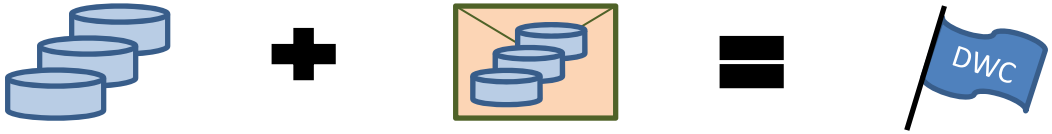
1 and 2: For each ingress frame, determine the expected LAG link:



(same process as in Conversation Sensitive Distribution)

3: Collect or discard the frame based on the DWC flag and whether the received link is the same as the expected link.

Managed Objects for DWC



- Admin_Discard_Wrong_Conversation
  - Per-Aggregator configuration (read/write) variable
    - a.k.a. aAggAdminDiscardWrongConversation in clause 12
    - Values: Force_False, Force_True, Auto
  - The Discard_Wrong_Conversation (DWC) flag will be:
    - False if
      - Conversation Sensitive Collection is not supported, or
      - Admin_Discard_Wrong_Conversation is Force_False, or
      - Admin_Discard_Wrong_Conversation is Auto and
        - » Partner Port Algorithm and MD5 Digest values match the Actor values and
        - » Neither Partner nor Actor Port Algorithm is “Unspecified”
    - True otherwise.

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.

Actor_Port_Algorithm values

- Algorithms identified by 32-bit OUI-based values
  - Most significant 3 bytes contain the OUI or CID of the organization responsible for the algorithm specification.
  - Least significant byte allows up to 256 algorithms to be specified by that organization.
- 802.1AX contains a table of values for algorithms specified in the standard
- Actor_Port_Algorithm is one of the values conveyed in LACPv2 PDUs so that the actor and partner systems can determine if they are using the same algorithm.
  - If the algorithm uses the aAggAdminServiceConversationMap[] table, a MD5 digest of that table is also conveyed in LACPv2 PDUs.

802.1AX standard port algorithms

Value	Frame field(s)	Mechanism
00-80-C2-00	Unspecified	Whatever the system uses for a default distribution algorithm.
00-80-C2-01	C-VID	Conversation ID = Service ID = C-VID
00-80-C2-02	S-VID	Conversation ID = Service ID = S-VID
00-80-C2-03	I-SID	Service ID = I-SID Conversation ID from service mapping table
00-80-C2-04	TE-SID	Service ID = TE-SID* Conversation ID from service mapping table
00-80-C2-05	ECMP Flow Hash	Service ID = ECMP Flow Hash Conversation ID from service mapping table

*This appears to be incompletely specified in 802.1AX-2014.

Example OUI-based Port Algorithm

- Dead Networking Co. has a favorite hash algorithm to use a TCP five-tuple for frame distribution on a LAG.
- To take advantage of Conversation Sensitive Collection and Distribution when connected to another Dead Networking Co. device, they use their OUI to assign a Port Algorithm value for this hash (FF-DE-AD-01).
- To take advantage of CSCD when connected to other devices they choose to publish their hash algorithm and the corresponding Port Algorithm value.
- The wildly enthusiastic response to their exceptional algorithm renews interest in their products and they re-launch the company as Thriving Networks Inc.

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.

Example
Admin_Conversation_Service_Map

Conversation ID	Link Selection Priority List
1	1,4,3,2,0
2	3,4,2,1,0
33	1,4,2,3,0
40	2,4,0

- Conversation ID 1:
  - Goes on Link 1 if it is up, otherwise to Link 4 if it is up, otherwise to Link 3 if it is up, otherwise to Link 2 if it is up, otherwise discarded.
- This is a 3+1 resiliency configuration
  - If all Links are up then traffic goes on Links 1, 2, and 3. Link 4 is standby.
  - If Link 1, 2, or 3 goes down, those conversations move to Link 4.
- Conversation ID 678:
  - Not in the map, so gets discarded (Link Selection Priority List is implicitly set to 0).

Observations

- Admin_Conversation_Service_Map provides very fine grain control over Frame Distribution
  - Which was the objective!
  - Great for “traffic engineered” interfaces
- But it has some disadvantages
  - Data structure has to be configured by management to get any traffic flowing
  - Have to know Link Numbers of the links that might be in the LAG
- Could also have “pre-fabricated” map
  - Simplify configuration burden and possibly enable “plug-and-play”

Example “pre-fabricated” maps

Active/Standby

Conversation ID	Link Selection Priority List
0	1, 2, 3, … 65535, 0
1	1, 2, 3, … 65535, 0
…	…
4095	1, 2, 3, … 65535, 0

- Active/Standby for any number of links with any possible Link Number:
- All traffic goes to lowest Link Number; all other links standby
- Does not require any foreknowledge of what links might be in LAG

Example “pre-fabricated” maps

Even/Odd Load Sharing

Conversation ID	Link Selection Priority List
0	1, 2, 3, … 65535, 0
1	65535, 65534, 65533, … 1, 0
…	…
4094	1, 2, 3, … 65535, 0
4095	65535, 65534, 65533, … 1, 0

- Load share between 2 links picked from any number of links with any possible Link Number
- All “even” traffic goes to lowest Link Number; all “odd” traffic goes to highest Link Number
- Does not require any foreknowledge of what links might be in LAG

Example “pre-fabricated” maps

Eight Link Load Sharing

Conversation ID	Link Selection Priority List
0	1, 4, 7, 6, 2, 3, 8, 5
1	2, 3, 8, 5, 1, 4, 7, 6
2	3, 6, 1, 8, 4, 5, 2, 7
3	4, 5, 2, 7, 3, 6, 1, 8
4	5, 8, 3, 2, 6, 7, 4, 1
5	6, 7, 4, 1, 5, 8, 3, 2
6	7, 2, 5, 4, 8, 1, 6, 3
7	8, 1, 6, 3, 7, 2, 5, 4

- Spread Conversation IDs approximately evenly over up to 8 links in LAG.
- Can replicate table across remaining Conversation IDs.
- Can algorithmically extend map so it will pick up to 8 links out of any number of links with any possible Link Number.

Question

- Is the concept of the “pre-fabricated” maps worth adding to the standard?
  - Implementer can use these maps now:
    - just by filling in the table appropriately, but it takes a lot of configuration.
    - Could add a short-cut management operation that fills the table in one go, but then if want to read this back without reading back the full table we have effectively created a new managed object.
    - This could be proprietary, but if so then what is the effect of a write to the table when we have set this new managed object to say we are using a “pre-fab” map?
  - Only advantage of describing as an option in the standard is to increase the likelihood that both Actor and Partner can do it.

Backup Slides