Thinking on conversation-sensitive frame collection

Ting Ao

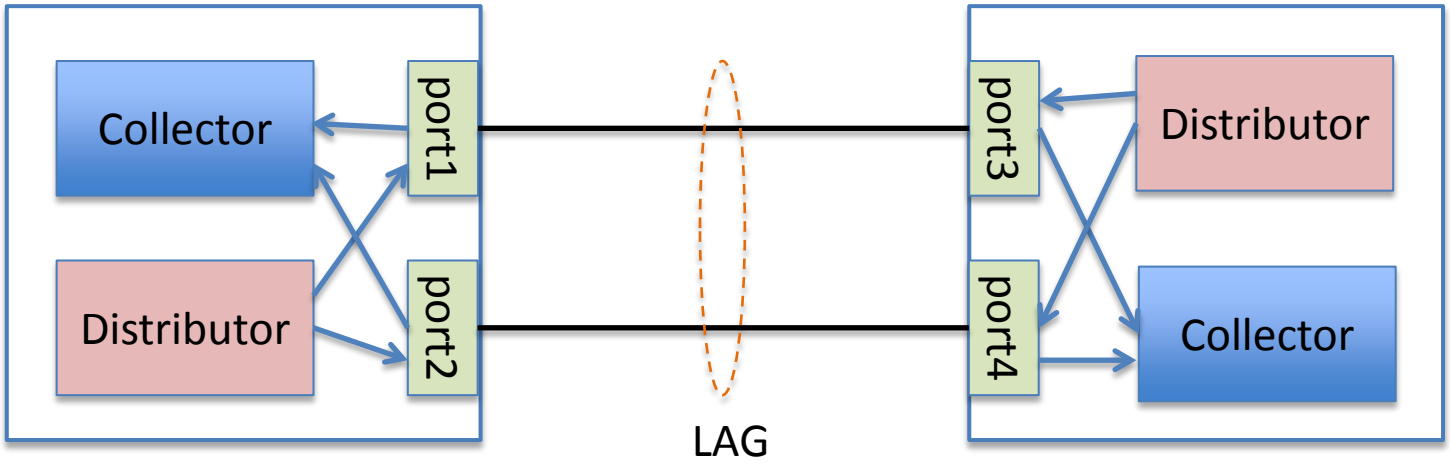
ao.ting@zte.com.cn

Content

- Requirements on conversation-sensitive collection
- Three ways for conversation-sensitive collection
- Pros & Cons of the three ways
- Something needs to be determined

Requirements

- In Clause 5.6 of Draft0.4
  - Collector is in the same **Conversation-to-Aggregation Port mapping(CPM)** with peer distributor.
  - Collector is in the same **flow-to-Conversation mapping(FCM)** with peer distributor.
- In Clause 7.2.1 of Draft0.4
  - If FCM is **per service distribution**, the two ends of an Aggregation Group use the same physical link in both directions for a given service.
  - For protection switch, when an aggregation link shutdown, conservation can be switched to the same active aggregation link for both ends , and vice versa.



Three ways for conversation-sensitive collection

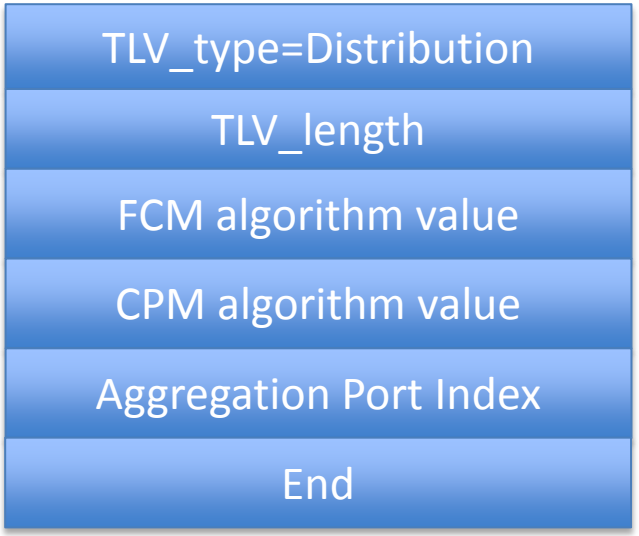
- In my opinion, there are three ways to meet the requirements of conversation-sensitive collection :
  - Algorithm TLV
  - Short CPM TLV
  - Long CPM TLV

Algorithm TLV(1)

- Define a set of algorithms for CPM, every CPM algorithm has its different value.
- Define a set of algorithms for FCM, every FCM algorithm has its different value.
- Number every aggregation link in the LAG.
- Once CPM and FCM have been determined on the two ends of the LAG, the collector can predict on which aggregation link the peer distributor would distribute the frames.

Algorithm TLV(2)

- TLV in LACPDU includes:
  - FCM algorithm
  - CPM algorithm
  - Aggregation port index
- FCM algorithm:
  - DA-MAC+SA-MAC
  - Per-service
    - VLAN
    - I-SID
  - ……
- CPM algorithm:
  - Hash，simplest: Entropy
  - Others?
  - ……
- Need to add some variables for management
  - Flow2Conversation_mapping
  - Converstaion2Aggport_mapping
  - AggrportIndex

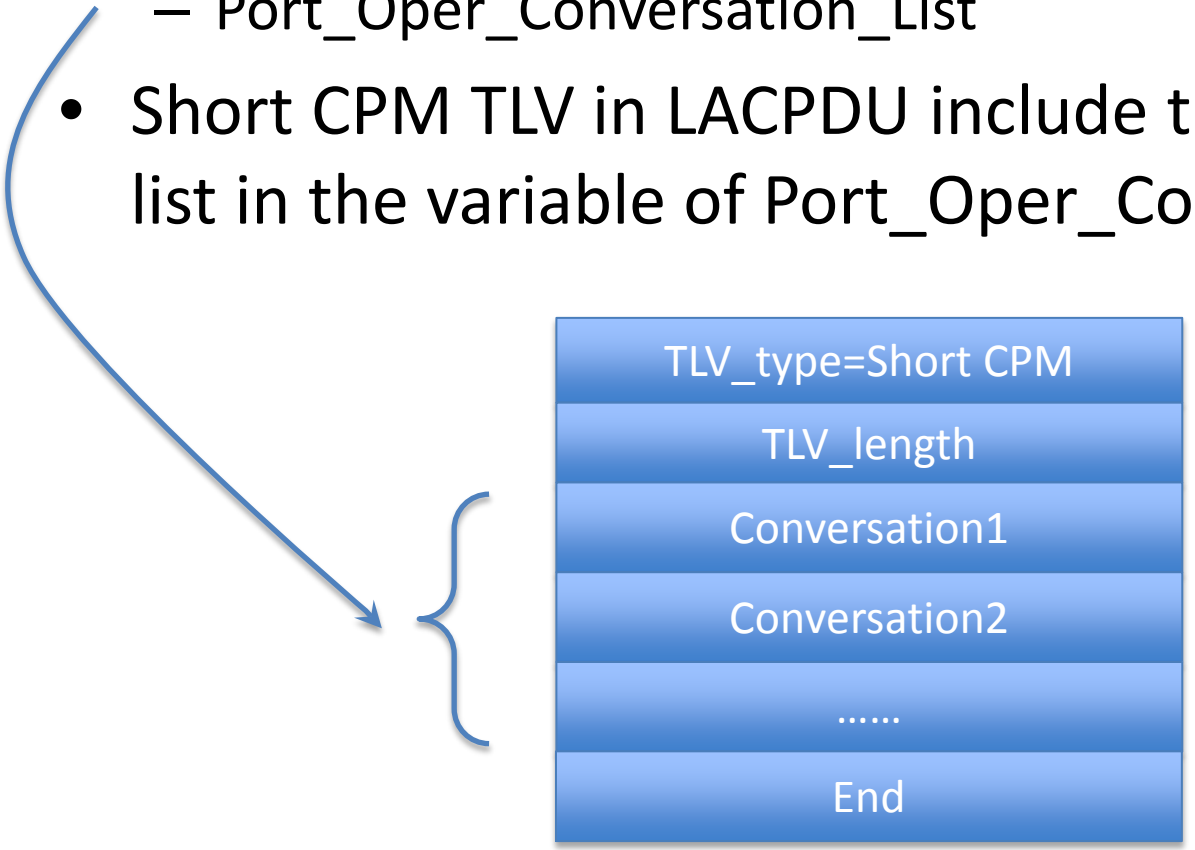


Short CPM TLV(1)

- FCM can be guaranteed by configuration or by coordination like FCM algorithm on two ends of the LAG.
- Short CPM TLV in LACPDU only include the conversation that is being transmitted on the aggregation port.
- Once an aggregation link fail, then:
  - Two ends adjust their CPM separately.
  - Short CPM TLV in LACPDU of an active aggregation port include a new set of conversations according to the adjustment.
  - Switchover the conversations impacted by the failure.
- Each end of the LAG should signal the change to the peer before they switch the impacted conversations to another active aggregation link.

Short CPM TLV(2)

- We already have two variables in the draft:
  - Aggregator_Conversation_Admin_Port[]
  - Port_Oper_Conversation_List
- Short CPM TLV in LACPDU include the conversation list in the variable of Port_Oper_Conversation_List.

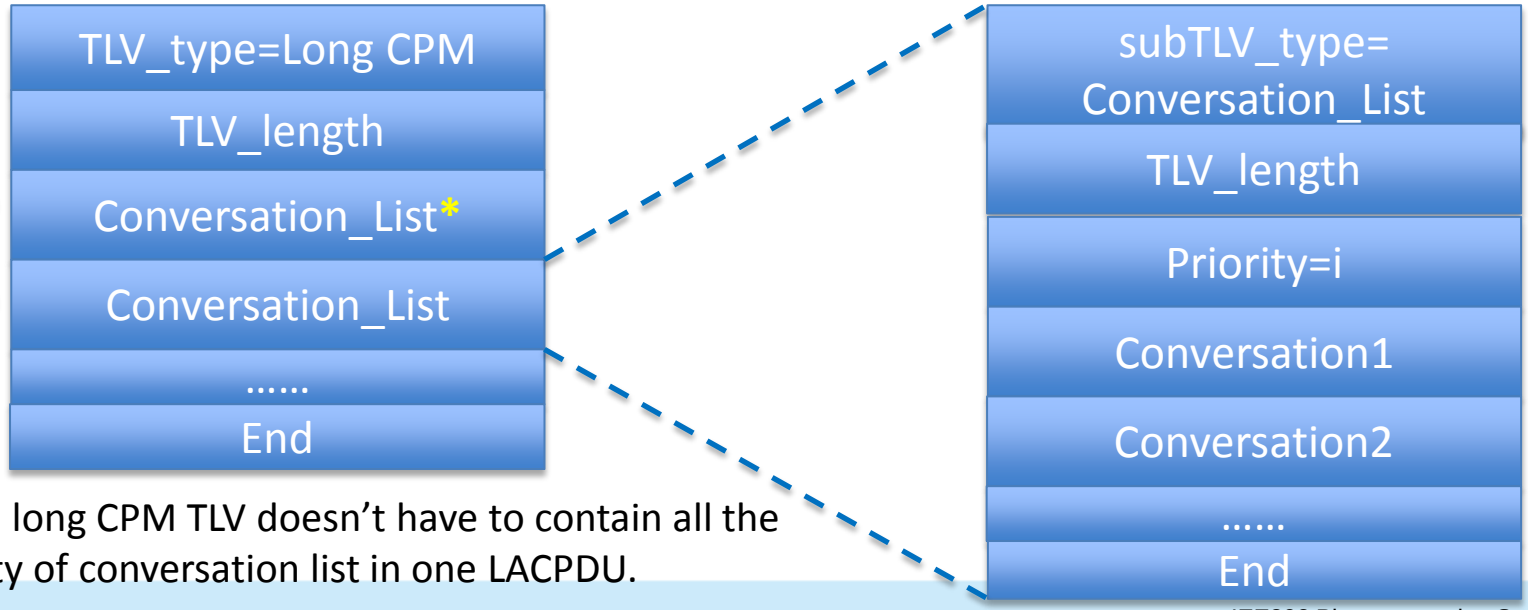


Long CPM TLV(1)

- FCM can be guaranteed by configuration or by coordination like FCM algorithm on two ends of the LAG.
- Long CPM TLV in LACPDU not only include the conversations that are being transmitted on the aggregation port, but also the conversations that are potentially transmitted on the aggregation port.
- Once an aggregation link fail, then:
  - The two ends of the DRNI switch separately according to the information in Long CPM TLV gotten in advance.
- Only the change on configuration would trigger the changed Long CPM TLV to transmit.

Long CPM TLV(2)

- From the variable Aggregator_Conversation_Admin_Port[] , we can get a set of conversation list on a aggregation port in priority:
  - Port_Oper_Conversation_List_P1(highest priority=Port_Oper_Conversation_List)
  - Port_Oper_Conversation_List_P2
  - ……
  - Port_Oper_Conversation_List_Pn (n_{max}=Number of the LAG member)
- Conversation_List subTLV include the conversation list which are in Port_Oper_Conversation_List_Pi.



*: The long CPM TLV doesn't have to contain all the priority of conversation list in one LACPDU.

Pros & Cons

	Algorithm signal	Current mapping	Whole mapping
Pros	1. Flexible, few information need to communicate 2. Switch quickly	1. Easy to standardize 2. A few information need to communicate	1. Easy to standardize 2. Switch quickly
Cons	Most hard to standardize.	Switch slowly, packet loss heavy.	Lots of information need to communicate

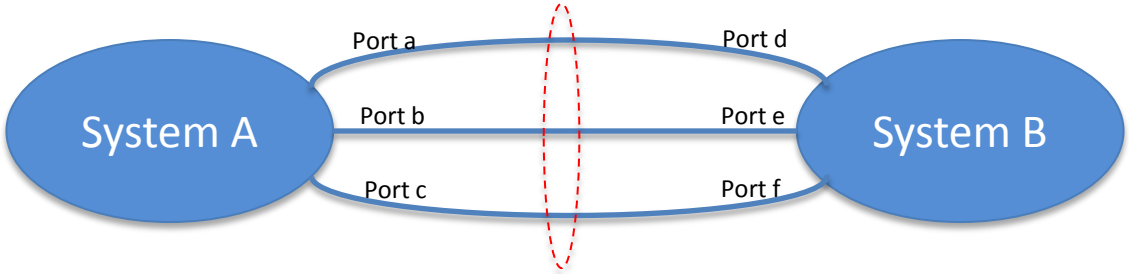
An issue need to be decided…

- Bidirectional congruity can be easy to achieve by the three ways, but need some rules to get it:
  - Configuration to make sure the two ends are same.
  - One end accept CPM FCM from the peer end
  - Coordination between the two ends： system priority or master/slave
  - Any other？

Thanks!

Backup：an example on short/long CPM TLV

- Here is a LAG with conversation-sensitive collection



Conversation	Port List
101-200	a、b、c
201-300	b、c、a
301-400	c、a、b

Conversation	Port List
101-200	e、f、d
201-300	f、d、e
301-400	d、e、f

Take port a as an example:
Port_Oper_Conversation_List={101-200}
Port_Oper_Conversation_List_P1={101-200}
Port_Oper_Conversation_List_P2={301-400}
Port_Oper_Conversation_List_P3={201-300}

Take port d as an example:
Port_Oper_Conversation_List={301-400}
Port_Oper_Conversation_List_P1={301-400}
Port_Oper_Conversation_List_P2={201-300}
Port_Oper_Conversation_List_P3={101-200}