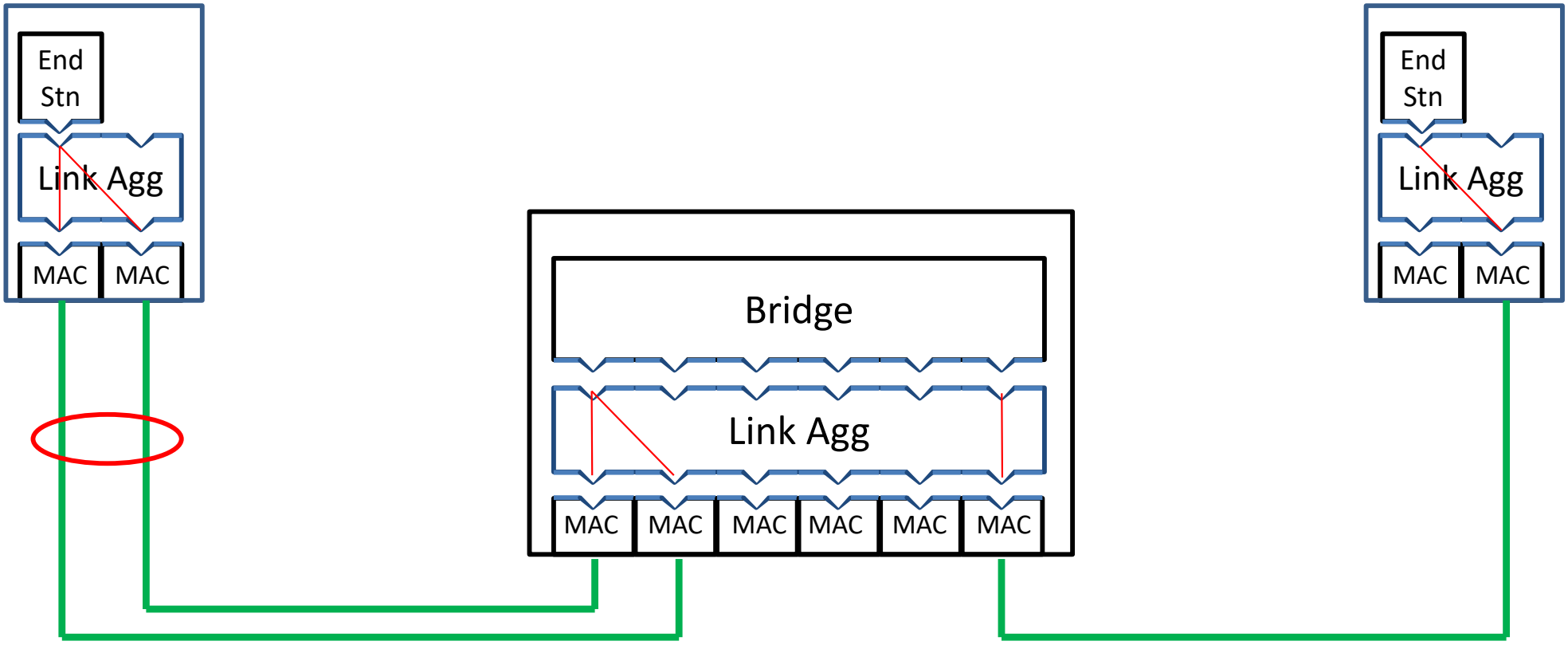
Link Aggregation Modeling Questions

Version 1

Stephen Haddock

March 8, 2024

LinkAgg Sublayer in Systems

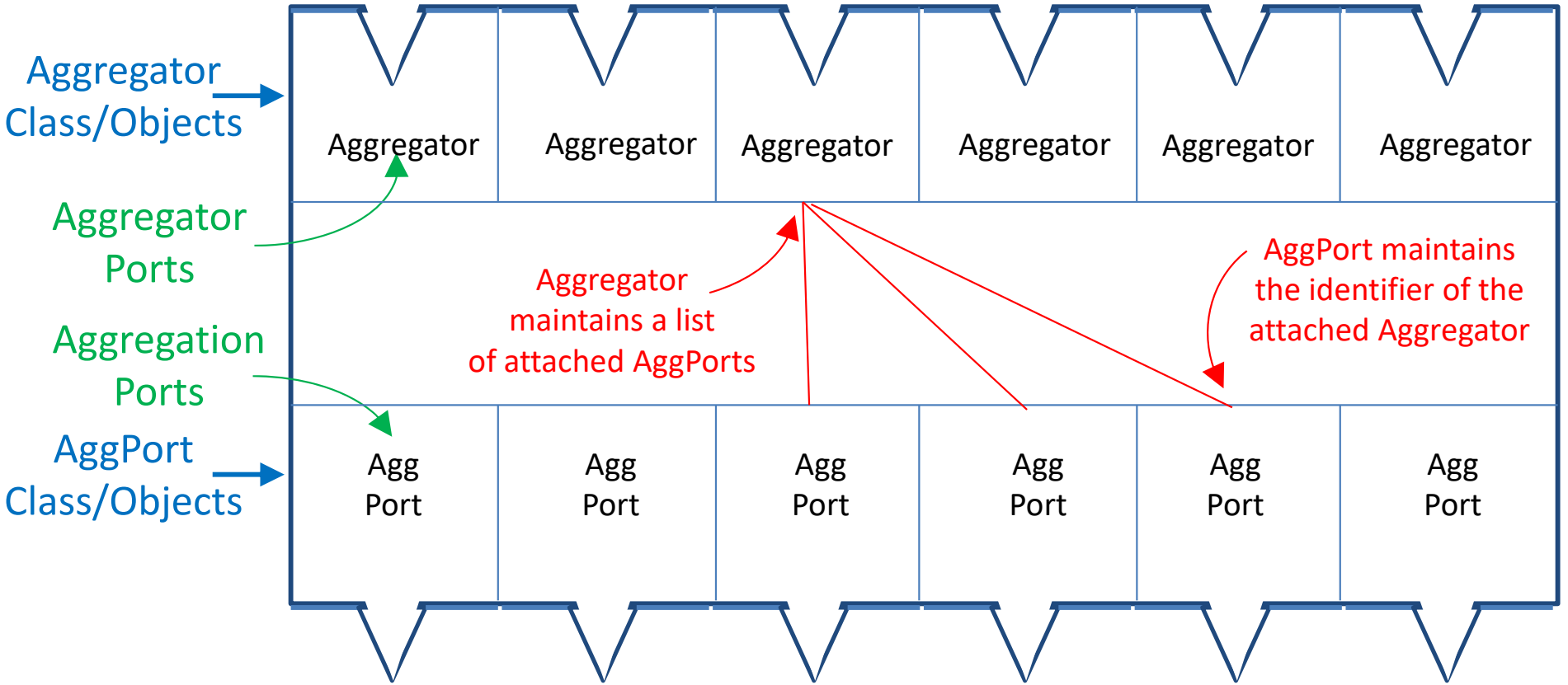


- Link Aggregation Sublayer sits between Bridge Ports and MACs in a bridge; between MAC Client(s) and MACs in an end station.

Link Aggregation Service Sublayer

The Link Aggregation Sublayer is a collection of Aggregators and Aggregation Ports.

- Typically one Aggregation Port per MAC.
- At least one Aggregator per BridgePort/MACSAP. (Can always model with one Aggregator per Aggregation Port, but some Aggregators may be unused.)
- Aggregation Ports are attached to Aggregators dynamically by the Link Aggregation Control Protocol (LACP).



Interface Stack Modeling Assumptions

1. An Aggregation Port sits in the interface stack with the MAC (and shares the same ifindex).
2. An Aggregator is a separate interface (with a separate ifindex) and sits in the interface stack with a BridgePort (or MAC Client).
3. Attachments between Aggregation Ports and Aggregators are indicated using the “higher-layer-if” and “lower-layer-if” pointers.

In YANG terms (?)

- MAC interfaces are an IETF interface assigned a MAC iftype (e.g. “ianaif:ethernetCsmacd”) and augmented with a MAC specific YANG module
  - These interfaces are then augmented with a 802.1AXdz Aggregation Port YANG module.
- Aggregator interfaces are an IETF interface assigned the iftype “ianaif:ieee8023adLag” and augmented with a 802.1AXdz Aggregator YANG module.
  - I think this is the intended use of this iftype. Is there any way to verify? Should we define a new type for 802.1AX?
  - These interfaces can be augmented with a BridgePort (or a MAC Client) YANG module.
    - Or should BridgePorts be separate IETF interfaces and use the “higher-layer-if” and “lower-layer-if” nodes to connect Aggregators to Bridge Ports?
    - This question has particular significance when add DRNI to the picture.

Experimental LAG YANG module

- In 2016 Marc Holness created an experimental YANG module based on 802.1AX-2008.
  - yang/experimental/ieee/802.1/ieee802-dot1ax.yang at main · YangModels/yang · GitHub
  - Does not include CSCD or DRNI
    - Jeff Hartley has volunteered to modify this model to 802.1AX-2020.
  - The experimental model mostly follows the above assumptions, but …
    1. It does not document the use of the “higher-layer-if” and “lower-layer-if” nodes. Is there a place in the YANG module where this would be appropriate? Or do we do it in text in the standard?
    2. The use of the “when” statement for augmentation confuses me.

Aggregation Port snippet

```
 //-------------------------------------------------
//  Aggregation Port Nodes
//-------------------------------------------------
augment "/if:interfaces/if:interface"
{    when "if:type = 'ianaif:ethernetCsmacd' or
                 if:type = 'ianaif:bridge'"{
      description
        "Applies to Ethernet interfaces or Bridge Ports.";
      }
    description
      "Augment interface model with Aggregation port
      configuration nodes.";
    reference
      "IEEE 802.1AX-2008, Clause 6.3.2";
    container aggregation-port {
      description
        "Contains Aggregation Port configuration related nodes,
        which provides the basic management controls necessary
        to allow an instance of an Aggregation Port to be managed,
        for the purposes of Link Aggregation.";
```

This makes sense to me

This does not make sense to me

Aggregator Snippet

```
//
//  Aggregator Nodes
//
augment "/if:interfaces/if:interface“
{   when "if:type = 'ianaif:ieee8023adLag' or
              if:type = 'ianaif:ethernetCsmacd' or
              if:type = 'ianaif:bridge'" {
        description
          "Applies to Ethernet interfaces or Bridge Ports.";
        }
    description      "Augment Interface with Aggregator configuration attributes.";
    container aggregator {
        description
          "Contains the Aggregator configuration information which
          provides the management controls necessary to allow an
          instance of an Aggregator to be managed.";
```

This makes sense to me

This does not make sense to me

This make sense only if not derived from a MAC

Snippet from 802.1Q bridge module

```
augment "/if:interfaces/if:interface" {
  when
    "derived-from-or-self(if:type,'ianaif:bridge') or "+
    "derived-from-or-self(if:type,'ianaif:ethernetCsmacd') or "+
    "derived-from-or-self(if:type,'ianaif:ieee8023adLag') or "+
    "derived-from-or-self(if:type,'ianaif:ilan')" {
  description
    "Applies when a Bridge interface.";
  }
    description
      "Augment the interface model with the Bridge Port";
    container bridge-port {
      description
        "Bridge Port is an extension of the IETF Interfaces model
        (RFC7223).";
```

This make sense only if not derived from a MAC

These should be mutually exclusive: Do not augment a MAC if the LinkAgg sublayer is present.

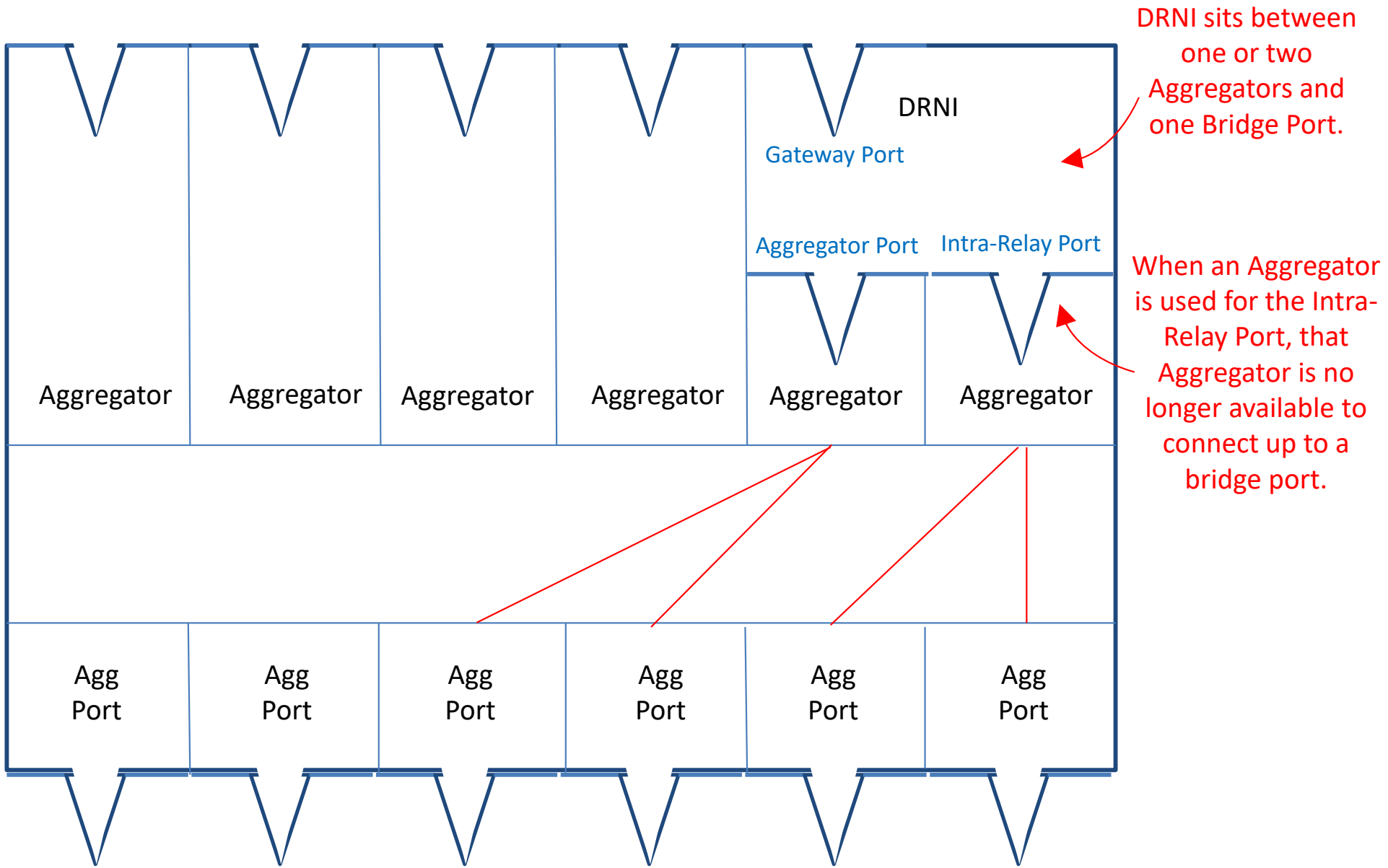
CSCD

- Conversation Sensitive Collection and Distribution (CSCD)
  - Optional feature in 802.1AX-2020 that allows actor and partner to agree on a distribution algorithm and keep bidirectional conversations on the same link.
  - Adds managed objects to both the Aggregation Port and the Aggregator.
  - Presumably need a CSCD module to augment the Aggregation Port module, and a CSCD module to augment the Aggregator module.

DRNI

- DRNI is an optional function within the Link Aggregation Sublayer that sits above two ISSs and below one ISS.
  - At least one of the lower ISSs (the DRNI Aggregator Port) always sits above an Aggregator.
    - In the YANG model, the DRNI can augment this Aggregator.
      - Note: Jeff Hartley tried this, and his comment was "[I] went with a simplified DRNI top-level structure for now. I was thoroughly unsatisfied with the duplicative baggage that was inherited when I tried to reuse an instance of Aggregator and augment DRNI inside of that, but we absolutely CAN go back to that method if you have a preference."
  - The other lower ISS (the Intra-Relay Port) may sit above a second Aggregator.
    - In this case the Aggregator supporting the Intra-Relay Port cannot also support a Bridge Port.
    - This is easy to model in YANG when the "upper-layer-if" and "lower-layer-if" pointers are used to connect the Aggregators to whatever sits above.
    - But what happens if the Aggregator that gets taken over by DRNI for the IRP was augmented by a Bridge Port?
      - Can we avoid augmenting this Aggregator with a Bridge Port by modifying the "when" statement in the Bridge Port module?

LAG Sublayer with single DRNI



IRP modeling questions

- If IRP Aggregator is/was augmented with a Bridge Port module …
  - Seems like there would be a lot of data accessible through that ifIndex that would make sense either for the IRP or the Bridge Port, but not both.
  - E.g. oper-status: Want this always false for the Bridge Port, but true for the IRP (when one or more Aggregation Ports are attached to the IRP Aggregator).

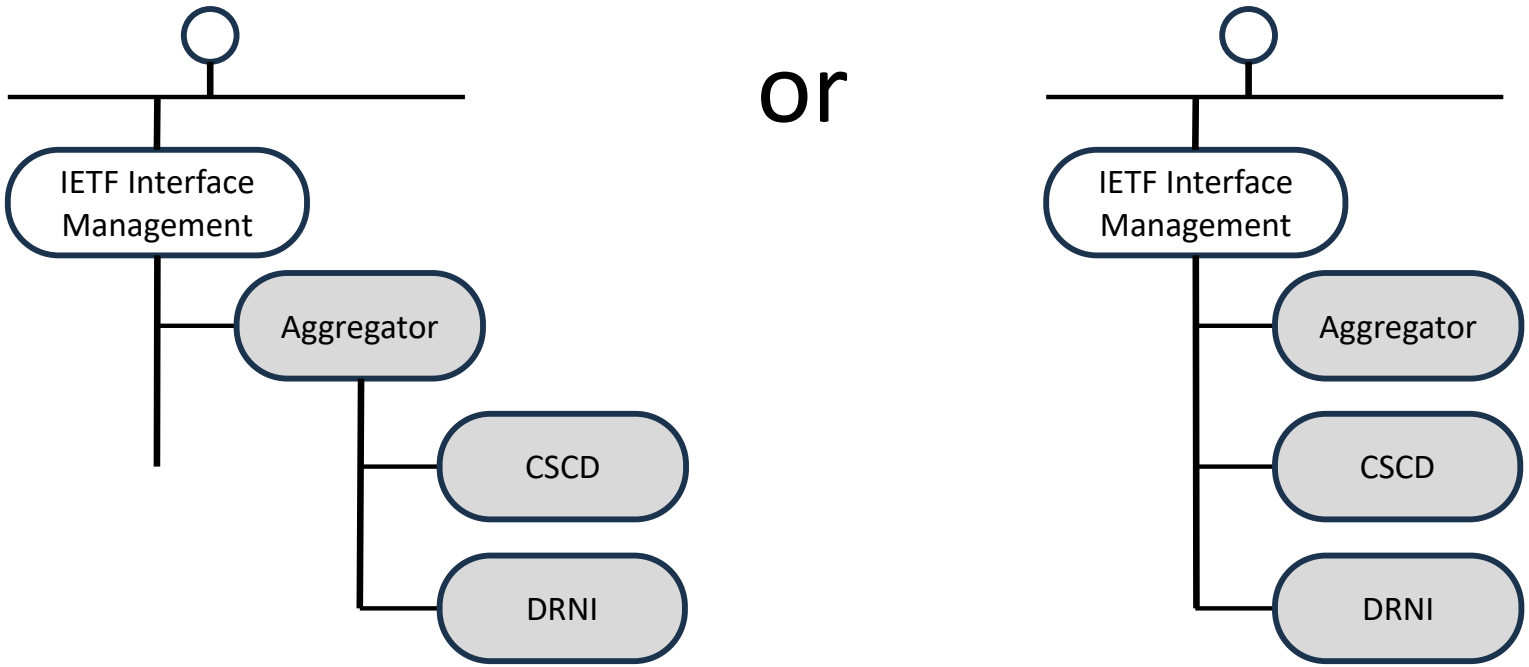
- If we block the augmentation of the IRP Aggregator with a Bridge Port module ...
  - Would have to change the ifType from “ieee8023Lag” to something like “ieee8021AXDrniIRP” when the Aggregator attached to the DRNI Aggregator Port is augmented with a DRNI module and the ifIndex of the Aggregator to be used for the IRP is configured into the DRNI module(?)
  - In order to modify the “when” statement to replace like “when … (derived-from-or-self(if:type,'ianaif:ieee8023adLag’) or …” with something like “when … ((derived-from-or-self(if:type,'ianaif:ieee8023adLag’) and not(derived-from-or-self(if:type,'ianaif:ieee802AXDrniIrp’))) or …”
  - Would this work if device was up and running before DRNI was configured? Can an interface suddenly not be augmented by a Bridge Port module any more?

- This shows my interface-stack-built-from-bottom-up-bias
  - Do these questions get easier if we look at it as a Bridge Port being augmented with an Aggregator, rather than the other way around?

- Do any similar questions come up with modeling devices with multiple bridge components?
- Do these questions get easier or harder if we model DRNI as a top-level system instead of an augmentation?

YANG Root Hierarchy Diagram

- Since CSCD and DRNI augmentations only make sense if the interface is also augmented with an Aggregator module, do I show that hierarchy in the diagram?



Backup Slides

DRNI with dedicated IRC Link

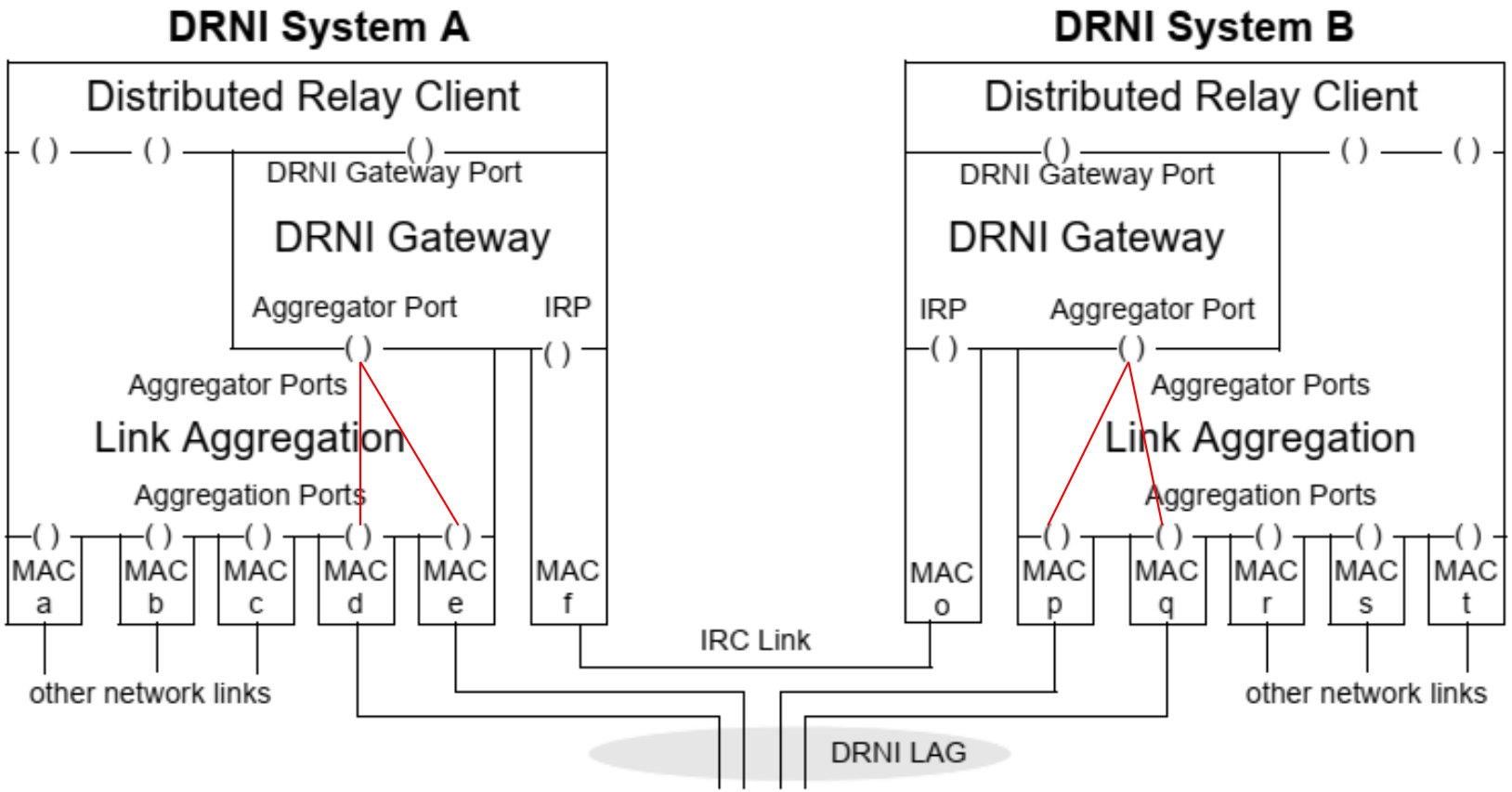


Figure 9-3—DRNI with dedicated IRC Link

DRNI with IRC LAG

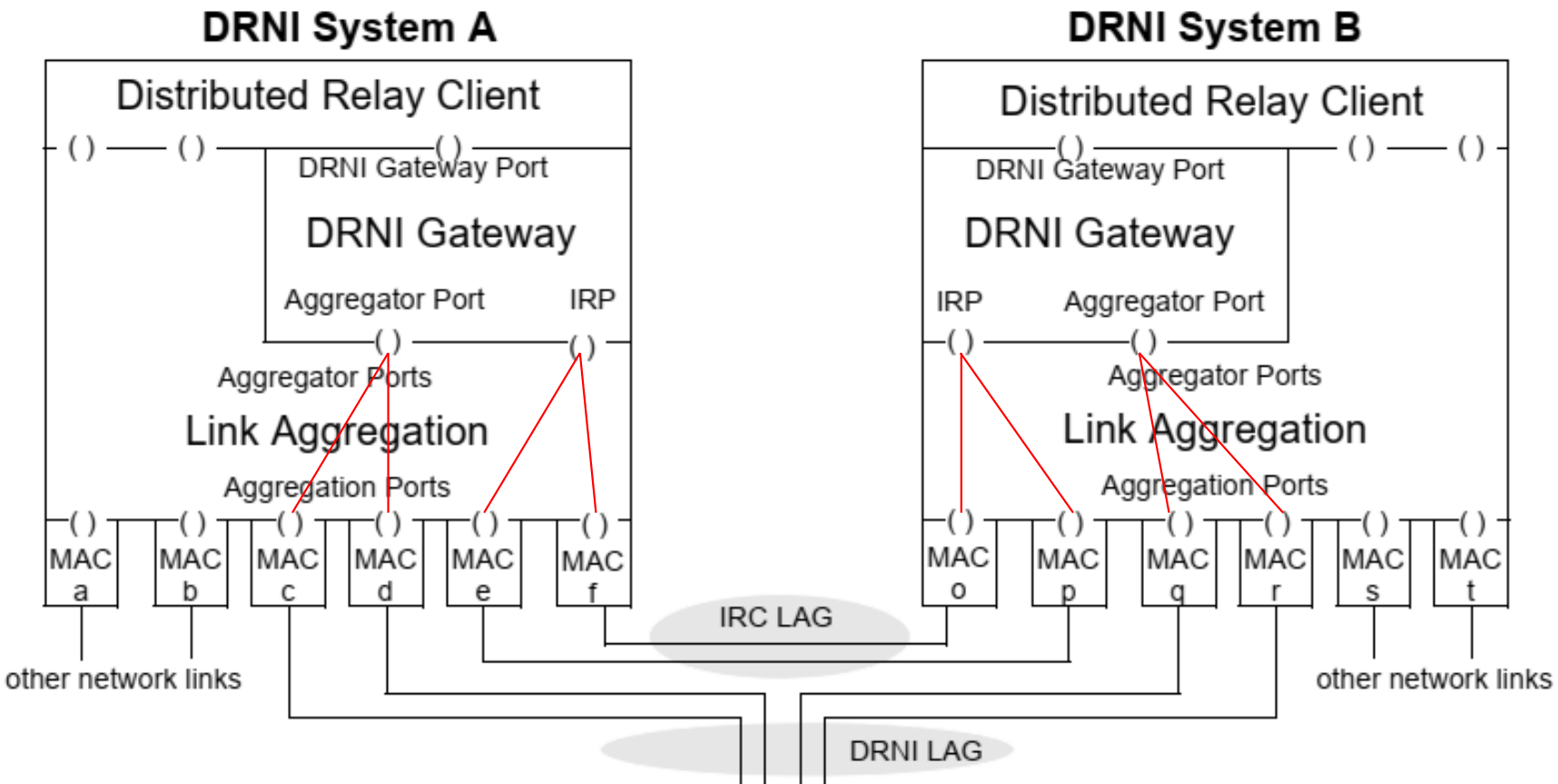


Figure 9-4—DRNI with dedicated IRC LAG

Acid test: 802.1AXbk (LAG of LAGs)