Merging the “shim” and “relay” Architectural Models

Stephen Haddock

January 11, 2006

Where we are:

- Two architectures have been proposed for 802.1ah
  - "Shim" model: proposed by Paul Bottorff and is in the current draft (p802.1ah/D1.52)
  - "Relay" model: proposed by Steve Haddock in

    http://www.ieee802.org/1/files/public/docs2005/ah-haddock-architectural-model-1105.pdf
- The models have several similarities …
  - Both are dual relay models with new functions on the port stack of the ports that interconnect the relays.
  - Both models have identical data forwarding behavior when both relays are in the same piece of equipment.
- … and one fundamental difference
  - Where the relationship between customer addresses and backbone addresses is learned and maintained.

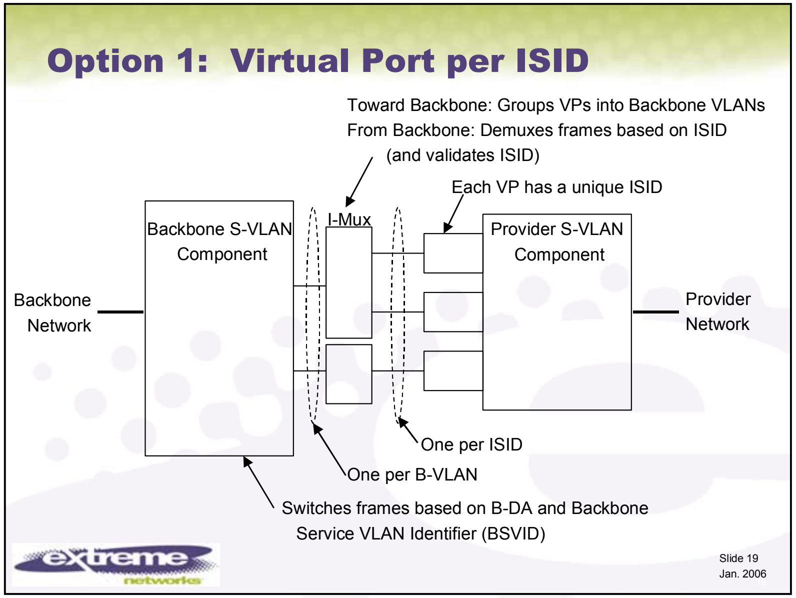
Simplified view of the debate

- Steve's objection to the "shim" model:
  - Having to create a address learning/resolution function in a shim layer of the B-component
- Paul's objection to the "relay" model:
  - Does not provide an I-tagged interface to allow:
    - Separation of B-component and I-component into different devices at the backbone edge
    - Interconnection of B-components at a backbone NNI
- So we need to find a model that:
  - uses the I-component relay for the customer-to-backbone address learning/resolution, and
  - provides an I-tagged interface

Moving to a merged model

- To get to the merged model from the “shim” model:
  - Take the customer-to-address learning/resolution function from the B-component side of the I-B connection and integrate it into the I-component relay and I-shim as proposed in the “relay” model.
- To get to the merged model from the “relay” model:
  - First, use the terminology established for the shim model in the current draft.
  - Second, discard both of the two options proposed for interconnecting the I-component and B-component in favor of a third option that creates a single logical connection without a B-tag. This creates an I-tagged interface.

Where the relay model went wrong



Option 1: Virtual Port per ISID

Toward Backbone: Groups VPs into Backbone VLANs
From Backbone: Demuxes frames based on ISID
(and validates ISID)

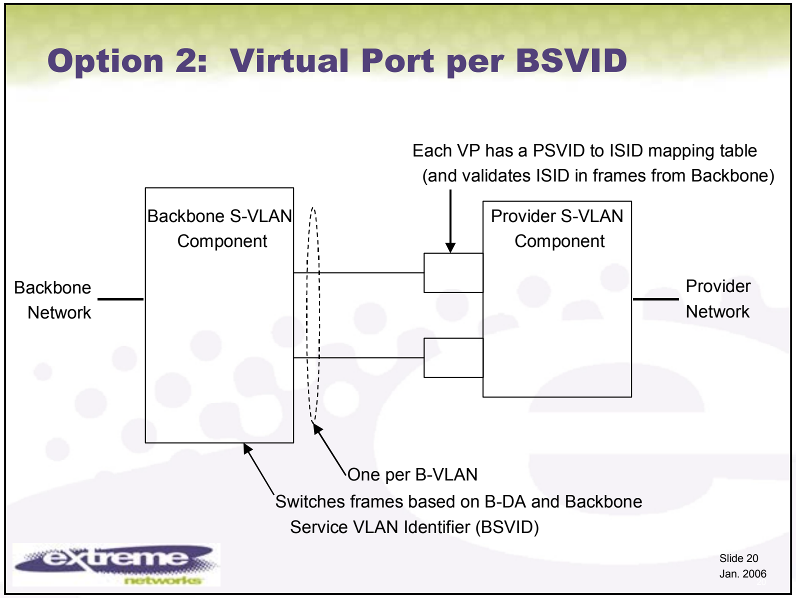
Each VP has a unique ISID

One per ISID

One per B-VLAN

Switches frames based on B-DA and Backbone
Service VLAN Identifier (BSVID)

Slide 19
Jan. 2006



Option 2: Virtual Port per BSVID

Each VP has a PSVID to ISID mapping table
(and validates ISID in frames from Backbone)

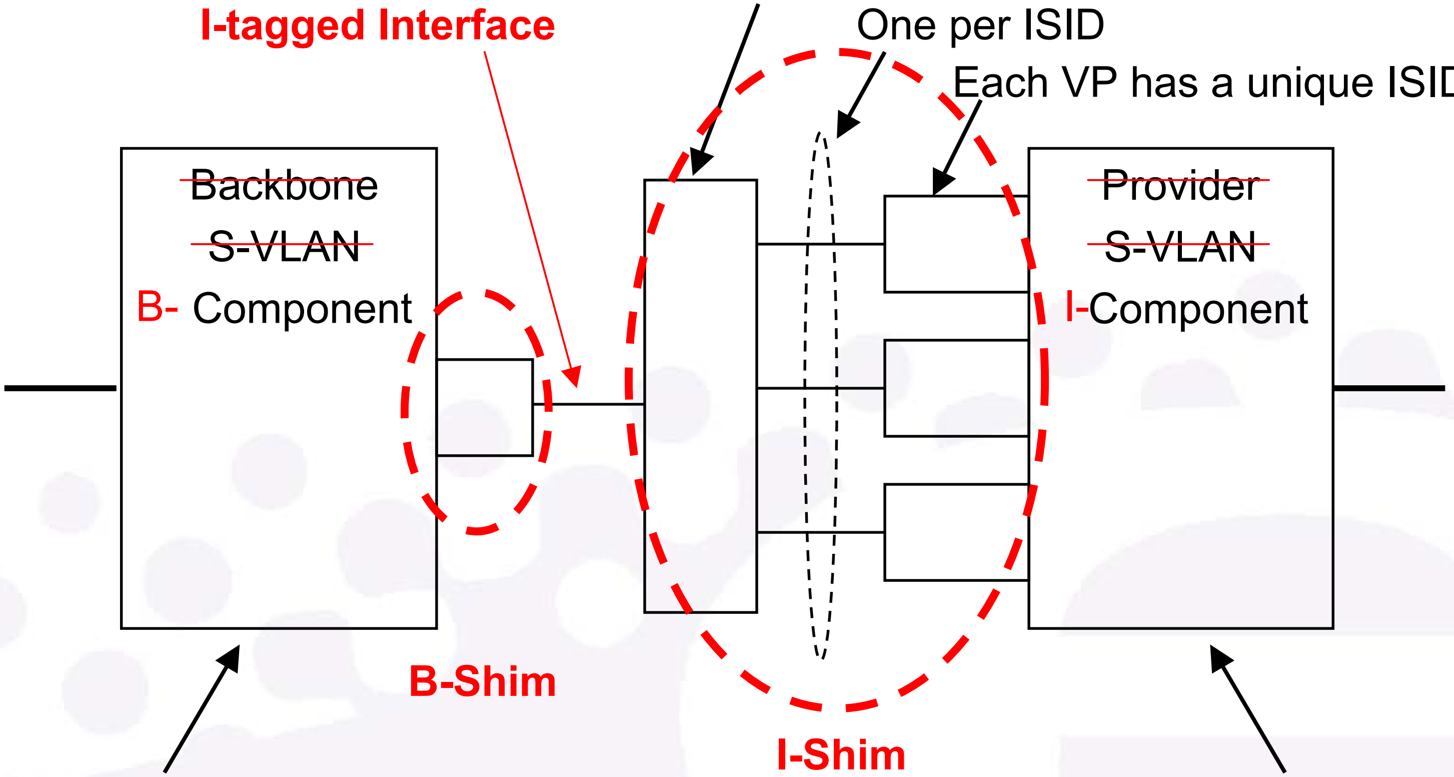
One per B-VLAN

Switches frames based on B-DA and Backbone
Service VLAN Identifier (BSVID)

Slide 20
Jan. 2006

- Both options present a logical connection per B-VLAN to the B-component. Although this an easy way to get the B-VID assigned (leverages normal bridge component functionality), it doesn’t make sense to have B-VIDs at this point in the network:
  - If split the B-component and I-component into backbone provider equipment and backbone customer equipment with a demarcation point between, don’t need or want B-VIDs on this link.
  - If connect a B-component to a B-component at an NNI, don’t need or want B-VIDs on this link.

Option 3: Virtual Port per ISID



Toward Backbone: Muxes frames to single link

From Backbone: Demuxes frames based on ISID

I-tagged Interface

One per ISID

Each VP has a unique ISID

~~Backbone~~

~~S-VLAN~~

B- Component

~~Provider~~

~~S-VLAN~~

I-Component

Backbone Network

Provider Network

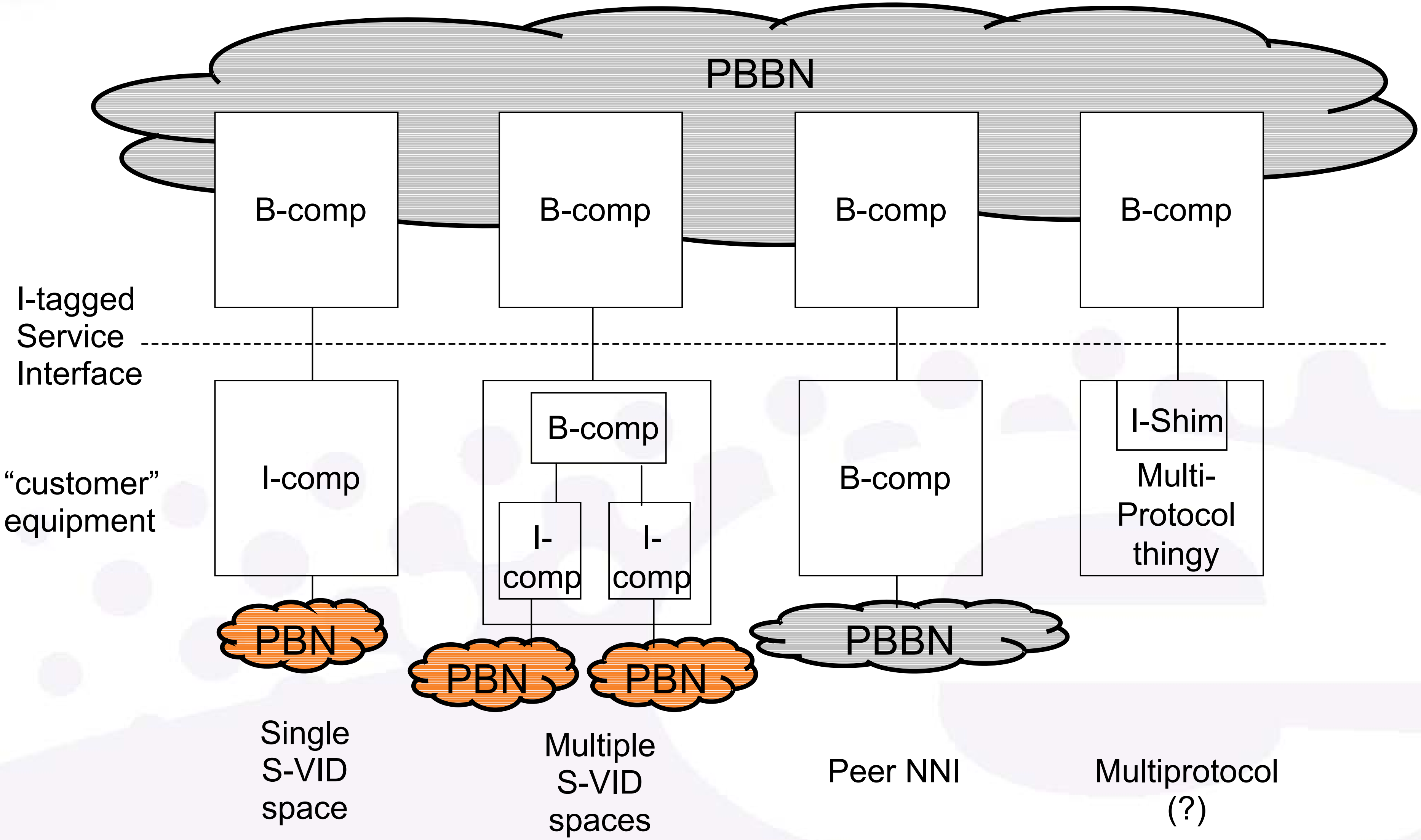
B-Shim

I-Shim

Switches frames based on B-DA and Backbone VLAN Identifier (B-VID)

Switches frames based on C-DA and Provider VLAN Identifier (S-VID)

I-tagged service interface examples



Impact of Merged Model on I-tagged I/F

- Advantages :
  - No new data base to learn and store C-MAC + I-SID to B-MAC relationships
  - Nothing on PBBN side of I-tagged interface knows or cares about customer addresses
    - No customer addresses learned at Peer NNI
    - Whether multiprotocol interface (if we decide it is within .1ah scope) deals with customer addresses at all is determined by the multiprotocol customer equipment.
  - Nothing on PBBN side of I-tagged interface participates in customer spanning tree protocols
  - S-tagged interface naturally supports bundling
- Disadvantages :
  - B-MAC addresses cross I-tagged interface
    - If a backbone provider wants to conceal it's addresses, need a MAC address translation capability in B-shim.

B-shim functions to support I-tagged I/F

- I-SID filter
  - Prevent sending/receiving frames across the I-tagged interface for I-SIDs that do not belong to that customer (analogous to S-VID filtering in 802.1ad S-tagged interface).
- I-SID translation
  - Allow I-SIDs to be locally significant at the interface (same reasons for having S-VID translation in 802.1ad).
- Mapping service instances to backbone tunnels
  - I-SID to B-VID mapping
  - I-SID to multicast B-DA mapping
    - If broadcast B-DA then translate to multicast so create mcast tunnels
    - Alternatively configure I-shim in customer equipment to use the multicast B-DA instead of broadcast but then B-shim may need to verify that the I-SID and mcast B-DA are a legal combination
- B-DA translation
    - Translate unicast B-DA if backbone provider wants to conceal true addresses