802.1ah Architecture Diagrams

Stephen Haddock

November 16, 2005

extreme networks

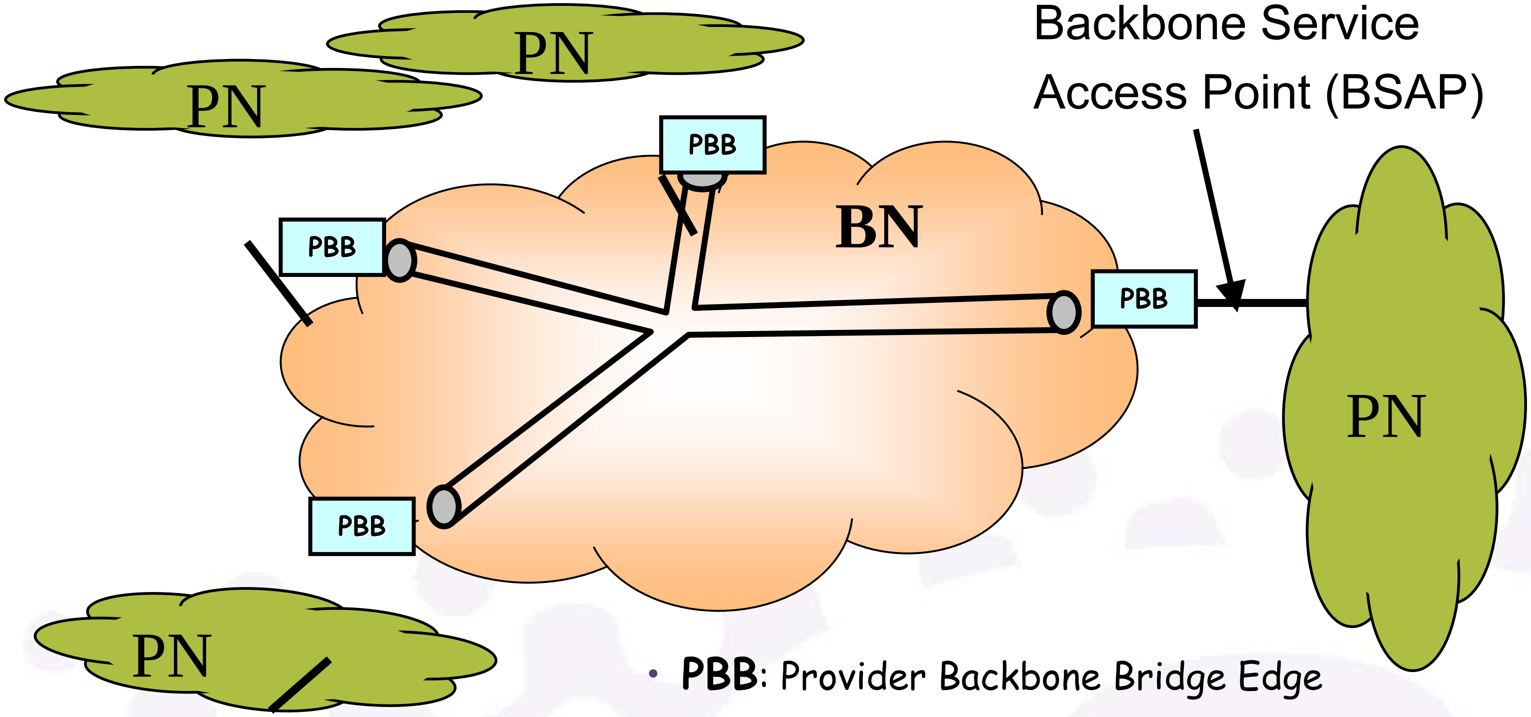
Fundamental Concepts of Provider Backbone Bridging

1. On Backbone Network, customer MAC addresses are “hidden” behind a new MAC header containing MAC addresses of the ingress and egress Backbone Service Access Point (BSAP)
2. On Backbone Network, the identification of a service (ISID) is separated from the identification of a broadcast domain on the backbone network (B-VID)
   - In Provider Bridging, both of these functions were combined in a single field – the SVID.
   - Separating them allows a large number of services to be identified (ISID can be 20 or 24 or 28 bits) without the control plane and data plane scaling issues associated with increasing the number of VLANs.

Part 1:

Backbone Service Access Points and creating Backbone MAC headers

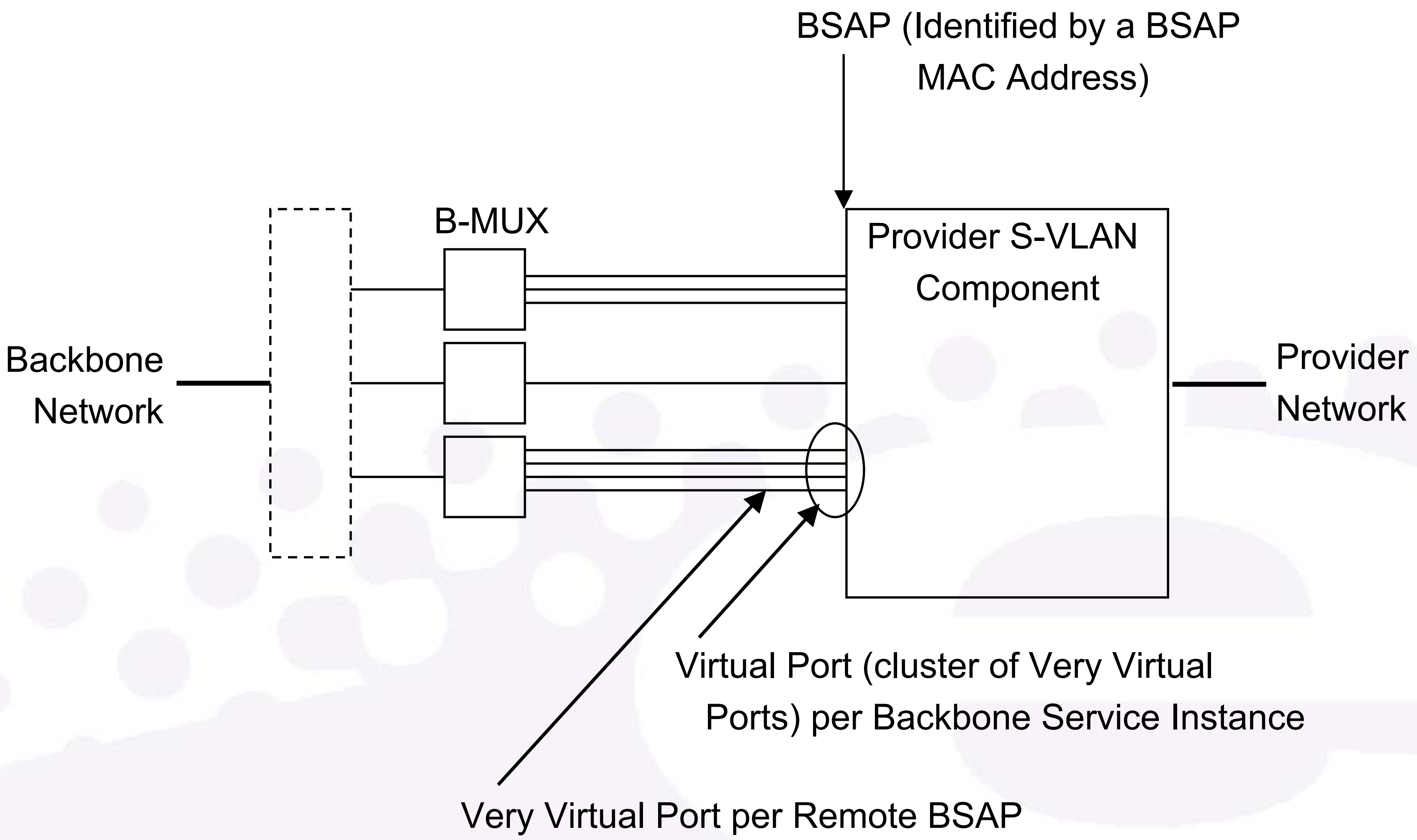
Simplified Problem Statement



- **PBB**: Provider Backbone Bridge Edge

- Backbone Service Instance interconnects two (P2P case) or more (Multipoint case) Backbone Service Access Points (BSAP) across the Backbone Network
- Need a mechanism at the BSAP where a frame enters the Backbone Network to associate the Customer destination MAC address with a remote BSAP where the frame exits the Backbone Network

Rudimentary Edge Bridge Model


BSAP (Identified by a BSAP MAC Address)
B-MUX
Provider S-VLAN Component
Backbone Network
Provider Network
Virtual Port (cluster of Very Virtual Ports) per Backbone Service Instance
Very Virtual Port per Remote BSAP

Very Virtual Ports

- Very Virtual Port (VVP) provides a structure for MAC-in-MAC encapsulation
  - Each VVP associated with a unique combination of local BSAP address, remote BSAP address, and Backbone Service Instance
    - In packets, Local and Remote BSAP addresses are B-SA and B-DA
  - For packets going toward Backbone, Provider S-VLAN component **forwards** packets to VVP based on Customer MAC Destination Address (C-DA) and Provider Network S-VLAN Identifier (PSVID)
  - For packets coming from Backbone, Provider S-VLAN component **learns** the association of Customer MAC Source Addresses (C-SA) and PSVID with a VVP
- B-Mux structure steers packets from the Backbone to a VVP on the basis of B-SA (remote BSAP address)

VVPs and Broadcast/Unknown Flooding

- Option 1: Replicate packet for each VVP in the VP
  - This is the natural behavior of the Provider S-VLAN Component
  - Inefficient because replicating packets at the ingress to the backbone network rather than only at points where the service instance branches within the backbone network.
- Optimization: Add a VVP specifically for Flooding
  - Modify S-VLAN component to send all packets with Broadcast or Unknown C-DA to the Flood-VVP
  - Flood VVP uses broadcast address for B-DA
  - B-Mux never steers packets from backbone to Flood-VVP
- Further Optimization: Add VVPs for specific multicast
  - Allows building multicast trees with mcast B-DA within backbone network

Pros and Cons of VVP model

- Good:
  - Provider S-VLAN component sees VVP as a “port” with respect to learning, because provides association between a Customer address and remote BSAP address
- OK:
  - Creating a VVP for Broadcast and Unknown flooding avoids unnecessary packet replication at ingress to backbone network, but requires a modification to the Provider S-VLAN component behavior that is unique for VVP clusters.
- Bad:
  - When a broadcast or unknown is received from a VVP, we do not want to flood it back out other VVPs in the same cluster. Requires another change to Provider S-VLAN component.
  - Control Protocols: without delving into details here, it is unlikely we want control protocols (e.g. RSTP, GMRP) to see each VVP as a “port”.

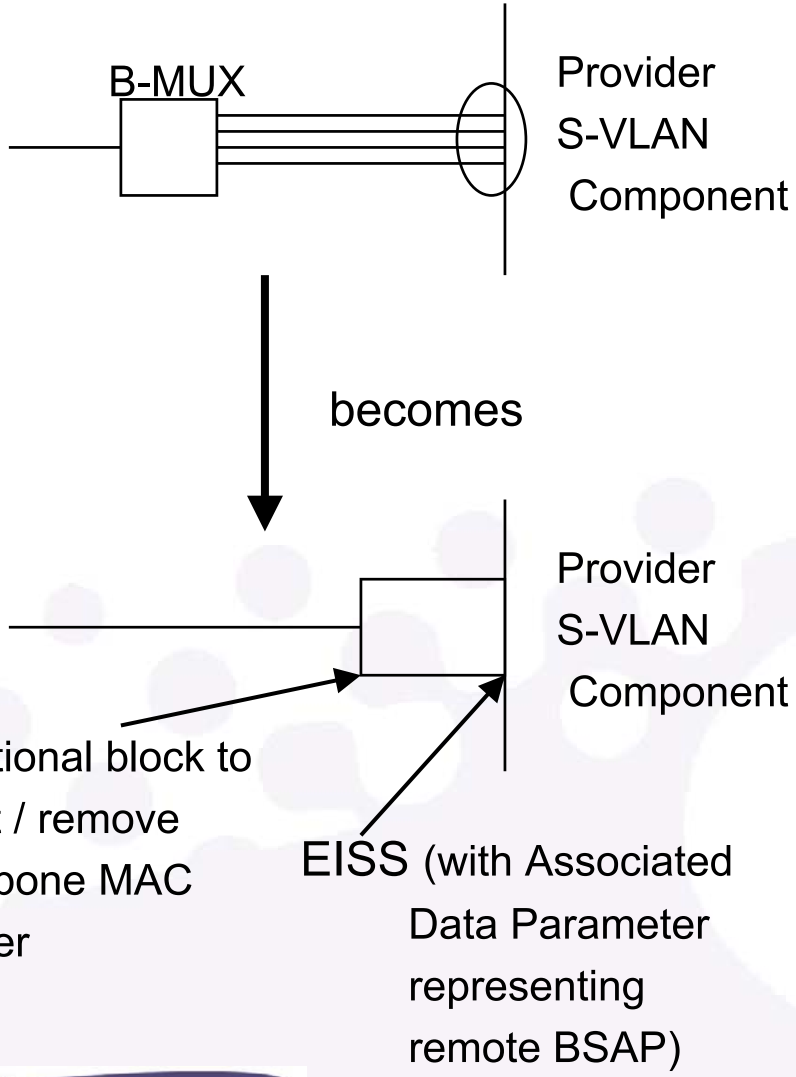
A Different Perspective

- Summary of Pros and Cons:
  - We want VVPs to look like a port for the purposes of learning, but we want the cluster of VVPs to look like a single port for all other bridge functions.
- Turn the VVP model on it’s head
  - Instead of considering the unicast VVPs the base case and the broadcast/unknown VVP an optimization, consider the broadcast/unknown VVP the base case and the unicast VVPs and optimization.
  - Note that you could run the backbone with just a broadcast/unknown VVP (if you were willing to send all unicast frames to all BSAPs in the backbone service instance)
  - Create a model where the VP (cluster of VVPs) is considered the “port” by all Provider S-VLAN component functions but allows VVPs to be an optimization that applies only for learning.

Associated Data in the FIB

- Add an optional field to FIB entries named something very generic like "Associated Data".
- Add an Associated Data parameter to the EISS.
  - When the learning functions creates a FIB entry, it stores the value from the received Associated Data parameter.
  - Static FIB entries (created by management) may have an Associated Data value.
  - When a frame is forwarded based on a FIB entry, it puts the Associated Data value in the EISS parameter.
  - If there is no FIB entry found for a packet, or if the Associated Data value is null, the parameter is null (results in bcast B-DA)
- For MAC-in-MAC support of the EISS, the Associated Data parameter is the remote BSAP address (or has a 1-to-1 mapping to the remote BSAP address).

New Model



B-MUX

Provider S-VLAN Component

becomes

Provider S-VLAN Component

Functional block to insert / remove Backbone MAC header

EISS (with Associated Data Parameter representing remote BSAP)

- Cluster of VVPs becomes a single VP at the EISS
  - Looks like a single port for all Provider S-VLAN component functions
  - Resolves all “OK” and “Bad” aspects of VVP model
- VVP concept achieved with the Associated Data parameter
  - Provides association of Customer address and PSVID to remote BSAP address
  - Preserves “Good” aspects of VVP model

Specification vs Implementation

- Specification of Associated Data as a FIB field and EISS parameter maximizes utilization of functionality already specified:
  - Leverage FIB that already contains associations with customer addresses (rather than inventing a new table in a shim),
  - Maintenance of the FIB already specified (including learning, "forgetting", dynamic and static entries, etc.)
- Implementation options for MAC-in-MAC:
  - Associated Data is a 48 bit remote BSAP address
  - Associated Data is a smaller field that maps to remote BSAP address
  - Associated Data is not implemented in FIB and broadcast B-DA always used
  - Associated Data is not implemented in FIB, and a customer address to remote BSAP address table is created at a virtual port level

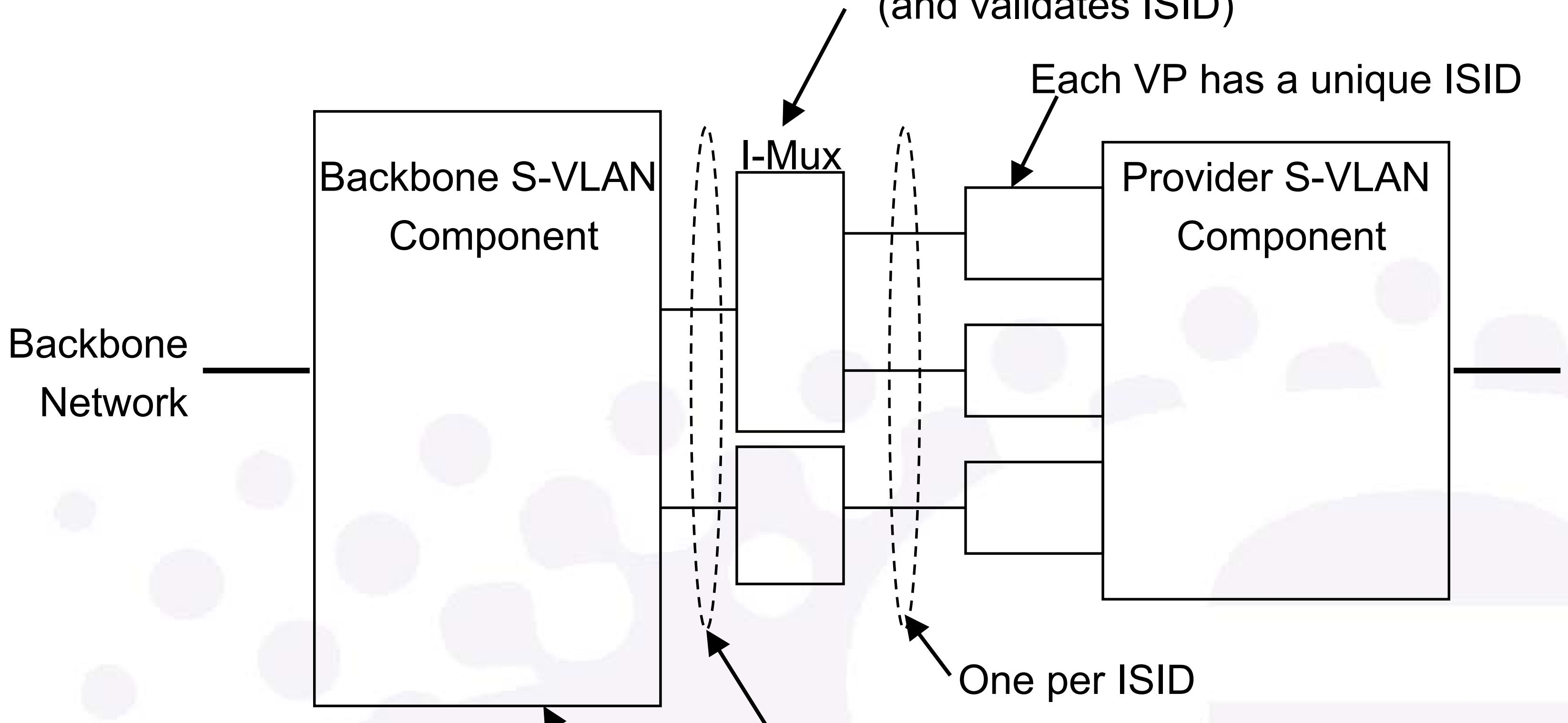
Part 2:

Backbone Service Instances

Backbone Service Instances

- 802.1ad had one identifier for both service identification and identification of VLANs in the Provider Network (S-VLANs)
- These identifiers are separated in 802.1ah
  - One identifier for the service (ISID) can be large (20-28 bits or conceivably more)
  - One identifier for the VLAN on the Backbone Network (BSVID)
    - This is an S-VID carried in a 802.1ad S-tag. Calling it a "BSVID" is an indication of where in the network topology it appears, not an indication of new functionality.
- What is the implication of this separation on the model?

Option 1: Virtual Port per ISID



Toward Backbone: Groups VPs into Backbone VLANs

From Backbone: Demuxes frames based on ISID (and validates ISID)

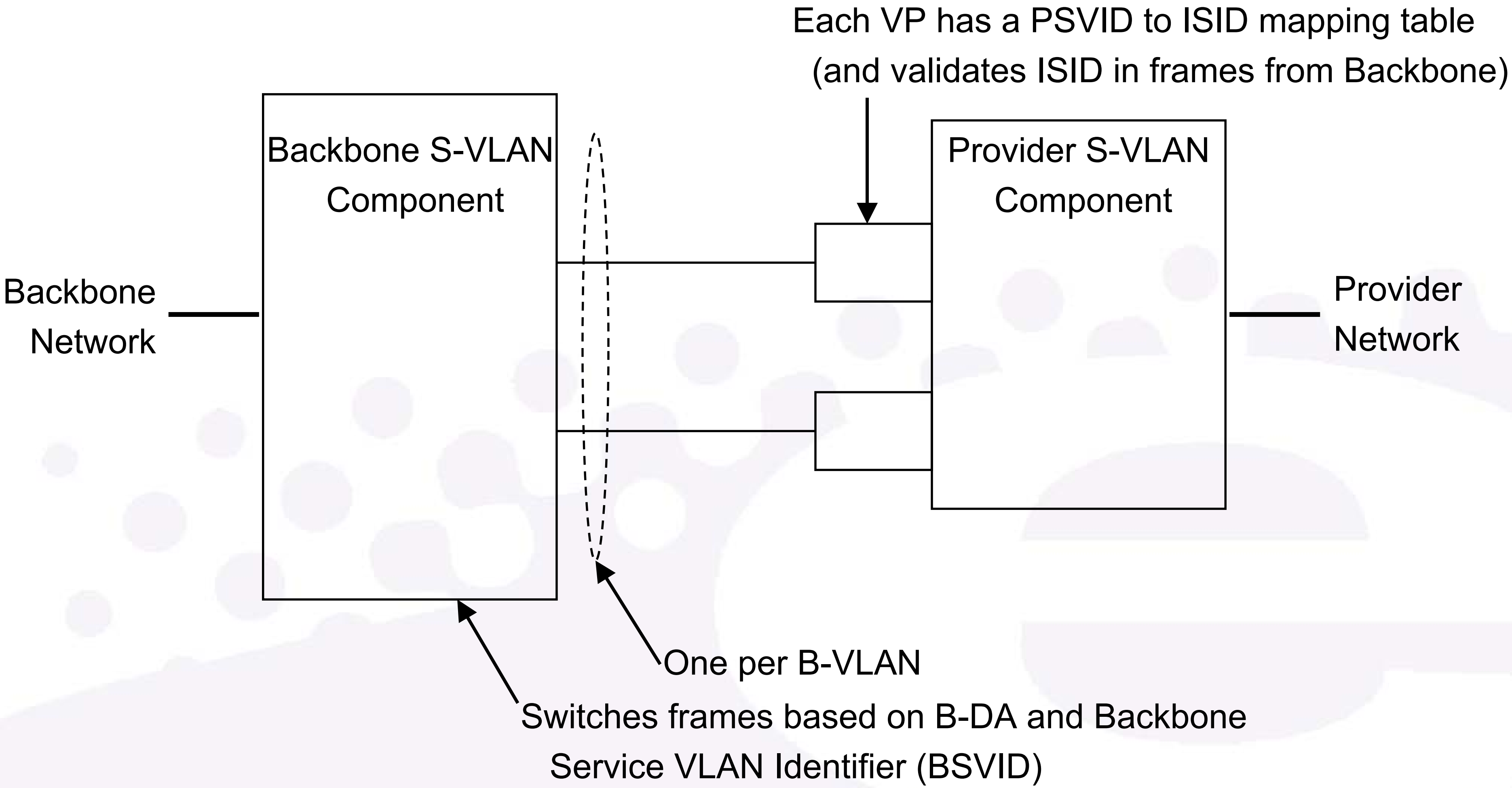
Each VP has a unique ISID

One per ISID

One per B-VLAN

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

Option 2: Virtual Port per BSVID



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

Backbone S-VLAN Component

Provider S-VLAN Component

Backbone Network

Provider Network

One per B-VLAN

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

Considerations in deciding which option is better

- Are the options simply different representations of the same functionality, or are there subtle (yet critical) differences in functionality inherent in the representation?
- Is one option better at leveraging functionality already specified versus requiring similar functionality to be re-invented?
- Does one option require more or less changes to existing (already specified) functions, interfaces, or control protocols?
- Is one option better at highlighting (versus hiding) critical aspects of system behavior?

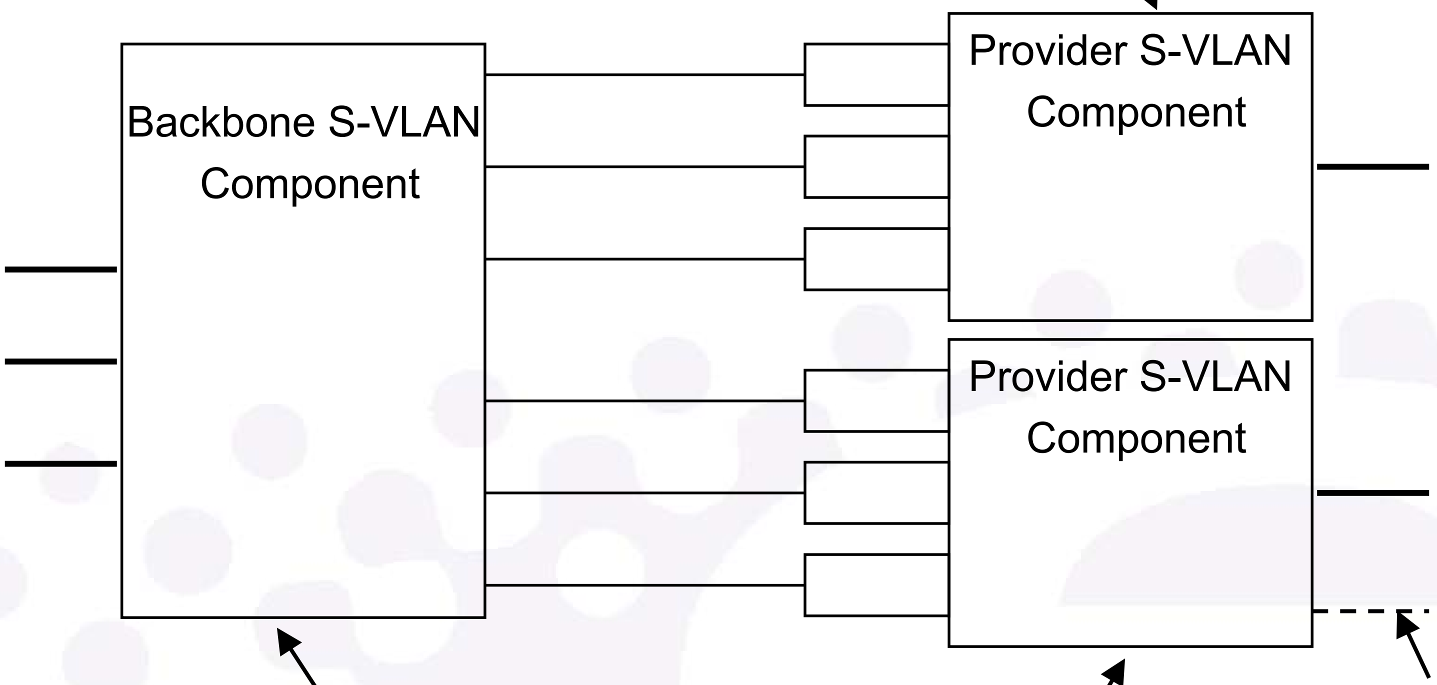
Hypothetical functional distinction

- As an example of when there might be a functional difference between options 1 and 2, hypothesize that we allowed one PSVID to map to more than one ISID (some customer address on that Provider S-VLAN went to one ISID, while other customer addresses on the same Provider S-VLAN went to a different ISID). Then:
  - Broadcast packets would need to be replicated to both ISIDs
  - A frame received on one ISID may need to be switched by the Provider S-VLAN component to the other ISID

  Both of these would be reasons to have a virtual port per ISID (if both ISIDs appeared on the same virtual port we would have to change the behavior of the S-VLAN component to enable the above functionality).
- Since we don't allow mapping one PSVID to more than one ISID, the models in option 1 and 2 are functionally identical with respect to this example.

Extending Model to multiple ports

Each Provider S-VLAN Component has unique BSAP MAC address and independent PSVID space.



Provider S-VLAN Component

Provider Network

Backbone S-VLAN Component

Backbone Network

Provider S-VLAN Component

Provider Network

Allow multiple Ports?

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

Switches frames based on C-DA and Provider Service VLAN Identifier (PSVID)

A couple of closing comments

- At this point I cannot identify any functional difference between the option 1 and 2 models, but it would be good to have others look at this.
- A significant characteristic of the final model is that all of the switching components (Backbone S-VLAN component and all Provider S-VLAN components) operate on 48bit addresses with 12bit VLAN tags. No switching is performed on the ISID, and PSVID to ISID mapping tables have a maximum size of 4K regardless of how large we make ISID. This is the benefit of separating the service identification from the VLAN identification. The ISID can be arbitrarily large because there is no place in the model where it has negative scaling impact.