SAS Grab Bag

March 14-17, 2005
Atlanta, Georgia

Mike Takefman

Agenda

- This presentation covers a variety of topics with the goal of driving a technical decision
  - Hosts, VLANs and duplicate MACs
  - SAS bypass on a frame by frame basis
  - SAS service interfaces
  - Frame formats and their effect on learning design
  - SAS indication method
  - Multicast scoping method

VLANs and Hosts: Theory

- 802.1Q is not explicit on the method of making a host VLAN aware:
  - “Well, what Q says is pretty sketchy, but I think there's (almost) enough information to work it out.”
    – Tony Jeffree
  - “That would be virgin territory. We've not defined a stack for an end station. I see why you'd ask, though. Your safest bet is to use the EISS, and decode the Q-tag yourself ”
    – Norm Finn

VLANs and Hosts: Practice

- How does a host participate then?
  - "The host that participates in 802.1Q usually follows one of two models:

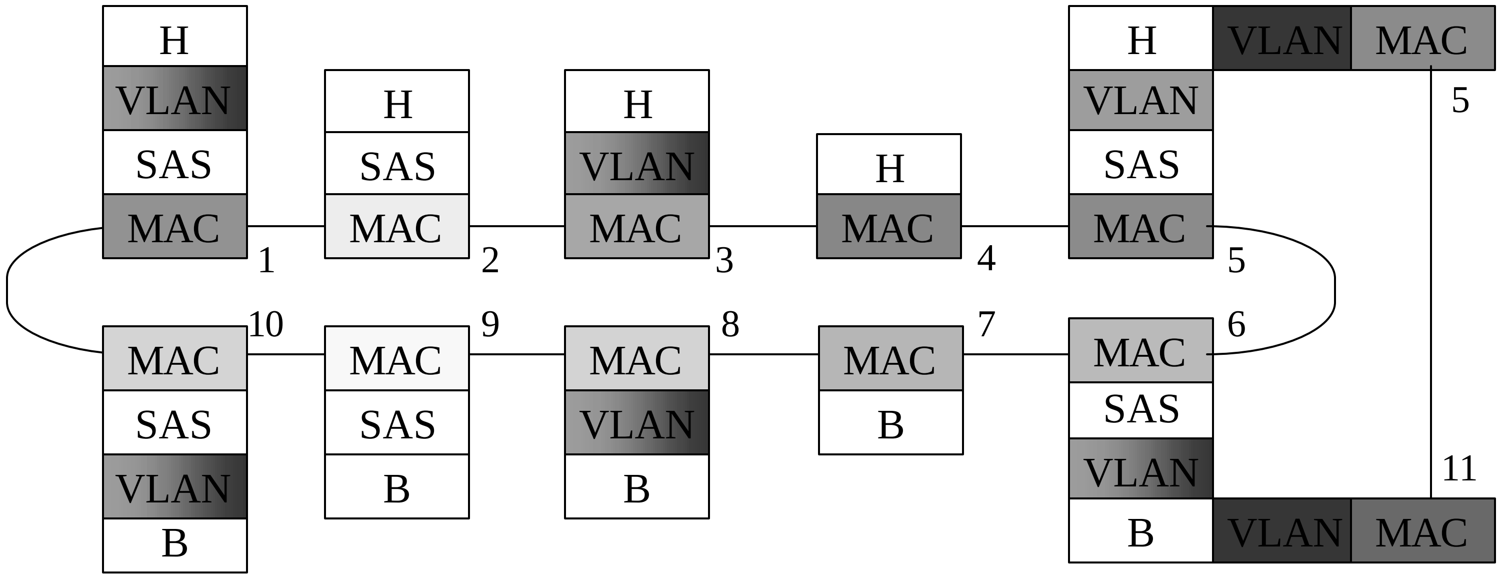
    1. Add a .1p tag (priority, but VID = 0) on transmission. Ignore Q tags entirely on reception. (Strip the tag, ignoring its contents, then deal with what's left.)

    2. Establish n virtual Ethernet ports inside the box, one for each VLAN the box participates in. Use the tag to identify which virtual port on input. always tag on output.

    I don't know of any other model that works. " – Norm Finn

Canonical Network Diagram

- Layer stackup, note IVL versus SVL not shown



VLANs, Hosts and .17

- 802.17-2004 does not handle the scenario of MAC-5
  - Not a likely scenario and certainly could be defined as illegal for .17-2004
- 802.17b could certainly handle such a scenario
  - SAS must force flooding of ring local DAs (when transmitted by SAS)
  - I.E. a host acts like a bridge when SAS is on

SAS Bypass

- Consider a L2/L3 box
  - Services split over both L2 and L3
  - SAS bypass allows L3<->L3 interactions on ring
    - no pollution of SDB
    - consistency with 802.17-2004
- How to signal on the ring?
  - easy with frame bit or special multicast address
  - harder with topology method unless SAS always uses extended frame format

SAS Bypass

- Have an optional parameter: sas_enable
- Use the 4 address form and signal SAS when DA = special multicast
- Use the 2 address form, but client encapsulates frame in SDU so it appears correct on the wire

SAS Service Interfaces

- Bridges access the MAC via
  - [E]M_UNITDATA.request/indicate
  - .17 maps [E]M_UNITDATA to MA_DATA
- Augment MA_DATA.request with a parameter sas_enable (default false)
  - sas_enable is true when SAS is active and the [E]M_UNITDATA.request has been called
  - sas_enable must be false for clients using the MA_DATA.request interface
- Allows hosts to do frame by frame SAS by sending frames that should be SAS'd to the bridge like interface

SAS Rx Learning / Frame Formats

- SAS must learn certain frame fields:
  - ignore how SAS operation is signaled on the ring
  - using 802.17-2004 frame formats

Tx Station	Key	Value
SAS Bridge	SA inner, {VID || FID}	SA outer
SAS Host sourcing	SA outer, {VID || FID}	SA outer
SAS Host tunneling	SA inner, {VID || FID}	SA outer

SAS Tx Frame Formats

Tx Station	Unknown Destination	Known Destination
SAS Bridge	Extended/Flood	Extended/Direct
SAS Host local direct to local SAS Host (no VLAN funnies)	-	Basic/Direct
SAS Host via non-local route to local SAS Host	-	Extended/Direct
SAS Host routed frame to off ring node via SAS Bridge	Basic/Flood	Extended/Direct
SAS Host tunneled frame to off ring node via SAS Bridge	Extended/Flood	Extended/Direct

Consistent Frame Format

- It is possible to allow SAS to use both basic and extended frame depending on source and destination nodes
  - its not necessarily a good idea
- A single frame format would be easier to implement and debug
  - no special rules for hardware to create entries for the SDB
  - no dynamic changes in frame format during transmit operation
- Simplicity does have a cost in terms of BW
  - Savings only on the flooded frame
  - Per frame SAS removes the cost to hosts that can differentiate between SAS and non SAS frames

Straw Poll

- I agree that SAS should be available on a frame by frame basis.
  - Voters: Y/N/A
  - All: Y/N/A
- I agree that the extended frame format should be used by all frames sent via the SAS sublayer.
  - Voters: Y/N/A
  - All: Y/N/A

SAS in Bridge / MAC interaction

- Two migration paths for SAS into the bridge
  - – SAS is defined in the MAC, but current bridges implement SAS as part of their “logic”
    - this is an implementation issue, and one we just have to make sure we don’t preclude by doing something dumb
  - – The Grand Unification of Bridges occurs and SAS migrates into the bridge proper
    - we don’t yet know how much influence we wield in terms of shaping this

SAS in Bridge / MAC interaction

- Current Bridge MAC interface does not include RPR optional parameters or 4 address fields
  - the bridge can encapsulate the original frame as the SDU, (almost but not exactly the same as backbone bridging) then the existing interface works with existing MAC service definition
    - use of the multicast DA forces flooding around the ring and indicates SAS to the other bridge client
    - use of a ring local DA forces directed transmission
    - protection is assumed
    - ringlet selection is controlled by MAC
    - not clear where multicast scoping is controlled, but it would be nice to have it in the bridge in some way

SAS Indication Method

- Topo DB and Explicit bit methods are harder to migrate into the grand unification theory
  - Topo DB has to be part of the datapath
  - Explicit bit not part of the service interface
- Topo DB check requires a CAM in ASIC implementations or many more cycles in uCode
- Topo DB does not support per frame SAS support
- Explicit bit removes 1/3 of the free bits in the header
  - As there are other methods available, why waste the resource?

SAS Indication Method

- Given that a single consistent extended frame format is desirable
- SAS Multi-cast address is:
  - a low cost solution
  - allows migration of SAS into the .1 sphere eventually

Multicast Scoping

- Functionally speaking:
  - the SAS DB is a fine place for multi-cast scoping to be added to a new ASIC system and this could be in the MAC or the Bridge
  - either the Topo DB or SAS db is a fine place for multi-cast scoping to be added to a uCode system
- Architecturally speaking:
  - the multicast scoping DB should be checked during the InitialTTL function call (part of the MAC transmit state machine)
  - the actual DB that is used could easily be the SAS DB or some other DB
    - The advantage of making it the SAS DB is compliance to our PAR
  - this does not provide an obvious path for this functionality to move into the bridge layer due to the limits of the MA_UNITDATA interface
    - Nor the ability for the MAC to snoop the bridge DB, therefore the bridge would have to program the MAC DB

Recommendations

- SAS always uses extended frame format
- SAS is invoked via an MA_UNITDATA.request with the MAC configured for SAS
  - Allows non-SAS traffic to use MA_DATA.request
- SAS does support the aliased MAC/VLAN topology
  - SAS determines flooding requirement (if its not in the SAS DB flood it)
- SAS DB holds the multicast scoping information and is accessed as part of InitialTTL() to do actual scoping