ALANTEC™

Intelligent Switching Hubs

Forwarding Methods for VLAN-Tagged Frames in a Bridged LAN

John Wakerly, CTO
2115 O'Nel Drive
San Jose, CA 95131
1-800-ALANTEC
wakerly@alantec.com

General Assumptions, Issues, and Caveats

Pros and cons of VLAN tagging

- I dunno, but here's an approach to evaluate for possible merits and drawbacks of tagging

Tagging approach

- Based on prepended tags with calculated tag check

Applicability

- Explained here in terms of Ethernet application
- Can be extended to FDDI and Token Ring, including source routing
- CRCs will probably have to be recomputed for Ethernet <--> FDDI/TR

More Assumptions, Issues, and Caveats

Spanning Tree

- Works with single, global Spanning Tree
- Works better (I think) with Spanning Tree per VLAN
- No changes to existing Spanning-Tree Algorithm

Existing 802.1 bridges

- "VLAN unaware"
- Fully compatible without change

Topology restrictions

- None discovered (but there probably are some)
- VLAN-aware core plus VLAN-unaware switches at edge, or vice versa
- Mixture of both, loops created at any level

Further Assumptions, Issues, and Caveats

MTU violation

- Not addressed, but fragmentation is possible
- #!!@&@%%&&!!?%## (censored discussion here)

Interoperability of VLAN styles

- Definition of VLAN membership is left to the edge switches

Management strategy

- Left open, but...
- Real-time dissemination of management information (VLAN identifiers, addresses, etc.) is not required (a property of most tagging schemes)

More Assumptions, Issues, and Caveats

Algorithms

- Extends existing learning-bridge algorithm defined by 802.1
- Uses (multiple instances of) standard 802.1 Spanning-Tree Algorithm

Details

- Left open as much as possible, for committee discussion
  - VLAN identifiers, tag-format details
  - Special-address allocation
  - Security vs. connectivity

So, What Is It? Layer 2.1, sort of...

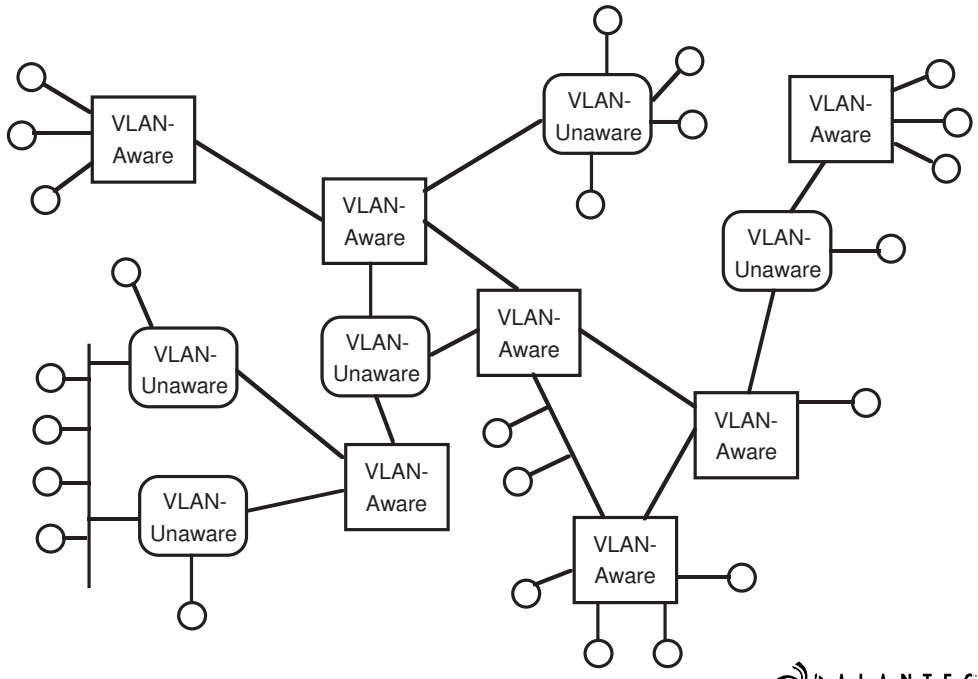
A layering of multiple, virtual 802.1 bridged LANs on top of a single, global 802.1 bridged LAN

- The global bridged LAN has an extent defined by the global spanning tree
- Multiple VLAN-specific spanning trees overlay the global bridged LAN
  - Each VLAN may cover only a subset of the global bridged LAN
- When carrying VLAN traffic, VLAN-unaware portions of the global bridged LAN simply provide "transport"
- In the VLAN -aware portion of the global bridged LAN, VLAN and non-VLAN frames are easily discriminated due to tagging

Definitions

- VLAN-aware switches (new stuff)
- VLAN-unaware switches (current bridges)
  - Handle tagged frames, but don't know it!
- Ingress switch — the one that adds the tag
- Egress switch — the one that removes the tag
- Segment and port type established administratively:
  - Interswitch (trunk) segments and ports
  - Access segments and ports
  - Hybrid segments and ports

Topology Example



The VLAN Tag

VLAN Tag (VTAG)

- Designed so a tagged frame starts out looking like an ordinary frame

Type field — New Ether-type = "VLAN"

- or could be 802.2/SNAP, or whatever
- (Type = VLAN) ==> tagged
- All others ==> untagged

VLAN ID (VID) — administratively allocated per VLAN

Flags — defined and used as needed

More Assumptions / Definitions

If multiple spanning trees, a VLAN-aware switch must have one MAC address (or equivalent) per VLAN it supports

- ... since frames in different VLANs may arrive on different paths
- VLAN-specific MAC address = VMAC
- VMACs appear only in VTAGs
- Equivalent: "Logical VMAC" = unique h/w MAC + VID

VLAN multicast addresses (or equivalent) = VCAST

- One per VLAN in the bridged LAN
- VCASTs appear only in VTAGs
- "Logical VCAST" = BCAST or unique MCAST + VID

Still More Assumptions / Definitions

A VLAN-aware switch does not have to handle all VLANs defined in the bridged LAN

- If VID is not recognized, frame is handled like an untagged frame

The question of allocated vs. logical VMACs and VCASTs is a good one for committee discussion

- Lots of OUIDs vs. few
- Consistency of VIDs with VMACs/VCASTs
- Ease of switch implementation
- # of MAC addresses to be learned by VLAN-unaware switches in the backbone
- Usefulness of filtering by VLAN-unaware switches

What's in a VLAN Tag?

Most frequent case (communication established)

- Dest addr = VMAC of egress switch
- Src addr = VMAC of ingress switch

Broadcast / multicast / unknown frames

- Dest addr = VCAST for designated VLAN
- Src addr = VMAC of ingress switch

VLAN-aware learning bridge operation

- Ingress switches learn MAC addresses of directly-connected (local) stations
- Ingress switches learn VMAC address of egress switch for each remote destination
- Intermediate switches learn only VMACs

Bridging-Table Requirements

For each MAC address, the following additional information is needed besides port #:

- Regular-MAC vs. VMAC flag
- If regular-MAC, local vs. remote flag
- A list of VLANs to which the MAC address belongs
  - Subject to administrative definition / restriction
- If remote, the VMAC address of the egress switch for each VLAN
  - Simplication — allow only one egress switch per MAC

How to use these fields is discussed in what follows

Forwarding Algorithms

General rules

- "Type=VLAN" ==> tagged; all other frames are untagged
- Frames never forwarded back onto arrival port (even after tag stripping or insertion)
- Frame forwarding is always governed by spanning tree for the "determined" or "specified" VLAN
  - Or the global spanning tree if no VLAN is "determined" or "specified"
  - Storms could occur if different VLAN-aware switches determine VLAN membership inconsistently

Forwarding Algorithms

Untagged unicast frames

- (Membership) Determine VLAN membership
- (Learning) Check source address, add to bridging table as "local"
- (Forwarding) Check destination address
  - If present and local, forward untagged frame
  - If present and remote, forward tagged frame:
    - VTAG Dest Addr = dest VMAC (from bridging table)
    - VTAG Src Addr = this switch's VMAC
  - If not present, flood:
    - Access and hybrid ports — untagged frame
    - Interswitch and hybrid — tagged frame:
      - VTAG Dest Addr = VCAST
      - VTAG Src Addr = this switch's VMAC

Forwarding Algorithms

Tagged unicast frames, if the VTAG's dest VMAC is assigned to this switch:

- (Membership/Qualification) Check tag consistency
- (Learning) Check untagged source address, add to bridging table as "remote", along with VMAC in VTAG Src Addr field
- (Forwarding) Check untagged destination address
  - If broadcast/multicast, then error
  - If present and local, forward untagged frame
  - If present and remote, then error
  - If not present, then (optionally) flood:
    - Access and hybrid ports — untagged frame

Forwarding Algorithms

Tagged unicast frames, if the VTAG's dest VMAC is not assigned to this switch:

- (Membership/Qualification) Check tag consistency
  - If VLAN unknown, forward like an untagged unicast of unknown membership (along the global spanning tree)
- (Learning) Check VTAG Src Addr, add to bridging table as "VMAC"
- (Forwarding) Check VTAG Dst Addr
  - If present and flagged as "VMAC", forward tagged frame
  - If present and not flagged as "VMAC", error
  - If not present, then flood all interswitch and hybrid ports
    - and (optionally) flood all access and hybrid ports with the untagged frame

Forwarding Algorithms

Untagged bmcast frames

- (Membership) Determine VLAN membership
- (Learning) Check source address, add to bridging table as "local"
- (Forwarding) Flood:
  - Access and hybrid ports — untagged frame
  - Interswitch and hybrid ports — tagged frame:
    - VTAG Dest Addr = VCAST
    - VTAG Src Addr = this switch's VMAC

Forwarding Algorithms

Tagged bmcast frames (VTAG Dst Addr = VCAST)

- (Membership/Qualification) Check tag consistency
  - If VLAN unknown, forward like an untagged bmcast of unknown membership (along the global spanning tree)
- (VMAC Learning) Check VTAG Src Addr, add to bridging table as "VMAC"
- (Remote Learning) If this switch has access/ hybrid ports for specified VLAN, check untagged source address, add to bridging table flagged as "remote", along with VMAC from VTAG Src Addr
- (Forwarding) Flood:
  - Access and hybrid ports — untagged frame
  - Interswitch and hybrid ports — tagged frame

Spanning-Tree Operation

Each VLAN-aware switch participates in global STA

Each VLAN-aware switch also participates in a separate STA for each VLAN it handles:

- VLAN-specific STA frames (BPDUs) are tagged
  - Dst Addr = VCAST
  - Src Addr = VMAC of originating switch
  - Possibly use a flag in VTAG to simplify decoding
- VLAN-specific STA works transparently across VLAN-unaware portions of the bridged LAN, following the global spanning tree

Other Considerations

Tags can be pre-computed by each ingress switch for each egress/VLAN combination

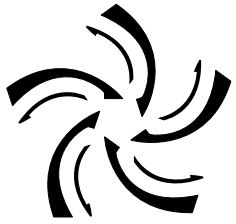
- With scatter-gather DMA, tags can be sitting in memory buffers, ready to go...
- Bridging table need only store pointers to tags, rather than actual VMAC addresses

Full write-up — 85% drafted (part of tagging write-up) — will be posted within a week

This is just a "first cut"

- It all needs to be discussed, checked, refined, and accepted or rejected by the talented people here....

Any more questions?



ALANTEC™

Intelligent Switching Hubs