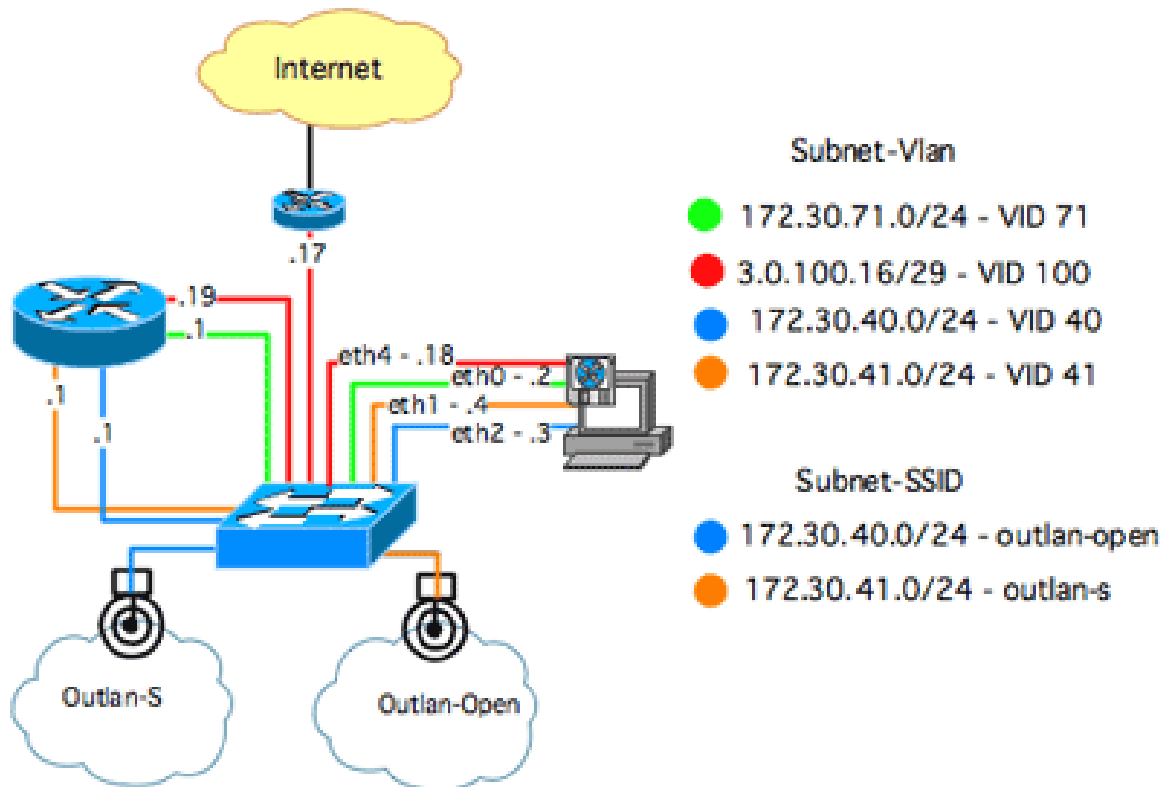


## Building 802.11Q VLANs

Michael J. Martin

Network segmentation can be a great way to increase security, but it can be a real pain to add all of the physical interfaces to support it. Implementing VLAN interfaces on Fast Ethernet or Gigabit Ethernet server interfaces is an alternative.

First, let's talk about VLANs and VLAN interfaces. Below is a variation on the WLAN topology we have been working with in this article series. Here the configuration supports two WLAN SSIDs (outlan-s 172.30.40.0/24 and outlan-open 172.30.41.0/24), along with interfaces connected to the private LAN (172.30.71.0/24) and the Internet (3.0.100.16/29). This topology is illustrated here:

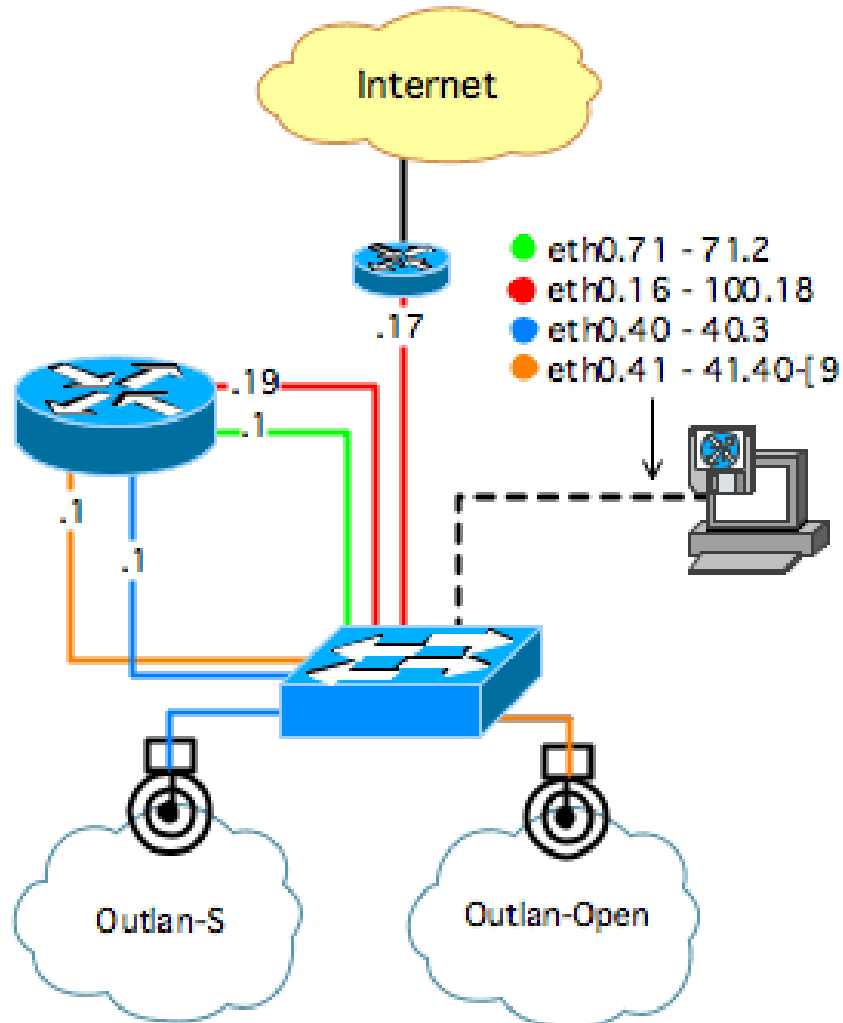


To support this topology, the server needs to support four physical interfaces. Each server interface is connected to a switch port assigned to a different VLAN. VLANs provide the capability to partition Ethernet switch ports into logical Layer 3 broadcast domains. Nodes connected to ports assigned to the same VLAN communicate as if they were all attached to the same physical wire. Because each switch port operates as a single Layer 2 broadcast domain, Layer 2/3 broadcast and multicast traffic is visible to all of the hosts in the VLAN, but unicast traffic is only visible between the sender and receiver (or IP segment gateway, making VLANs far more secure than broadcast-based hubs).

Here's what it looks like if the same interface support used by Ethernet switches to interconnect VLAN-capable switches were used on the Linux server in the same topology:

## Building 802.11Q VLANs

Michael J. Martin



With a single VLAN interface, all four networks can be supported, reducing both complexity and cost.

The concept of VLANs was developed by a number of Ethernet vendors (Intel, Cabletron, Grand Junction and Kalpana) in the early 1990s to extend the functionality of Ethernet switching hardware. When Ethernet switches were first introduced, they were very expensive in comparison to Ethernet hubs. This expense was due in large part to the custom ASICs (Application Specific Integrated Circuit) that enabled the switching hardware to switch packets between switch ports at "wire rate."

Early Ethernet switches supported port densities of eight to 12 ports, making them better suited to operate as high-speed multi-port bridges. Introducing VLANs made it possible to better utilize these limited port densities by having the Ethernet switch support switching across multiple Layer 3 broadcast domains. As Ethernet switch costs lowered and port density and adoption increased, the issue of extending VLANs across switches and between different vendor implementations gave rise to the development of the IEEE 802.1Q standard for virtual bridged local area networks.

The IEEE standards board approved the IEEE 802.1Q standard in December of 1998. The standard was updated in 2003, incorporating IEEE standards 802.1u (restricted VLAN address registration), 802.1v (protocol groups and VLAN identifiers set per port) and 802.1s (support for multiple spanning tree domains). There are three VLAN types:

## Building 802.11Q VLANs

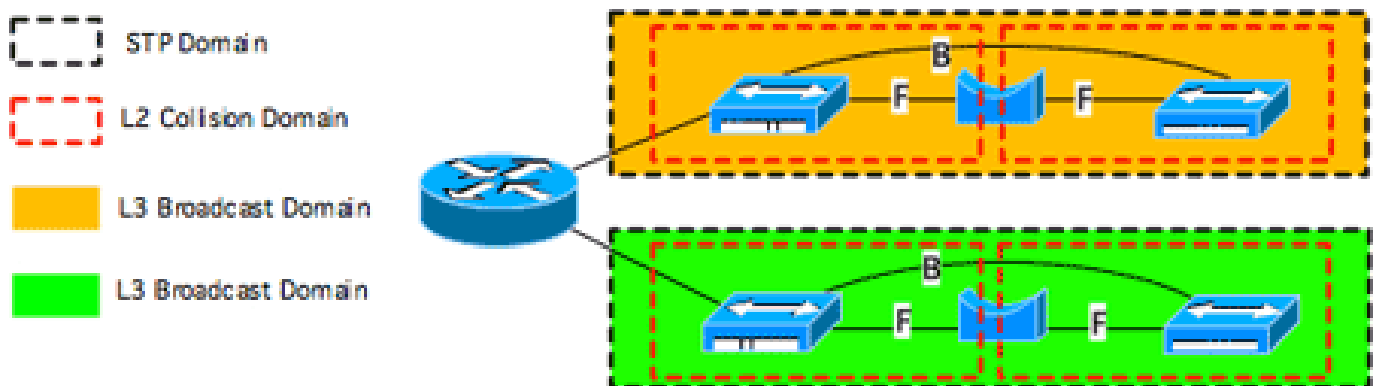
Michael J. Martin

- Port-based VLANs: The most common implementation, common VLAN identifiers (VIDs) are created on each switch. Ports on switches are statically assigned to VLANs.
- MAC-based VLANs: Common VIDs are created on each switch. MAC address access control lists (ACLs) are mapped to VLANs. The switch auto-configures the ports to the correct VLAN when the host port becomes active.
- Protocol-based VLANs: Common VIDs are created on each switch. Layer 3 protocol (i.e., IP, IPX, AppleTalk) ACLs are mapped to VLANs. The switch auto-configures the ports to the correct VLAN when the host port becomes active.

The major challenge of operating a virtual Bridged network is the interaction between the switches that make up the LAN. This interaction is driven by two factors: the maintenance of a loop-free topology between the interconnected switches and the delivery of packets belonging to different VLANs between switches.

The original 802.1Q standard defined the 802.1d spanning tree protocol (STP) for bridge topology management. 802.1d is the original IEEE standard that defines a link-management protocol algorithm allowing adjacent bridge devices connected to common Layer 2 segments to discover a loop-free link topology. Under 802.1d, only one forwarding link can exist between each switch in the LAN, although multiple physical links may exist. When the LAN first comes online, and each subsequent time a switch joins the LAN, the STP algorithm prunes all of the available network links and creates an inverse tree with a "root bridge" at the top and other bridges below. The result of STP is a single "spanning tree" domain consisting of interconnected Layer 2 collision domain segments (or switches) without loops, allowing all of the connected end-nodes to exchange frames. The STP process constantly monitors the state of all of the active links; in the event that one fails, the STP is executed again and a new root bridge is elected.

The 802.1d STP standard was approved in 1993, five years before the 802.1Q standard was ratified. The result of this had some unforeseen consequences. When the original 802.1d standard was written, bridges were used to "extend" a single Layer 3 network segment by connecting two or more Layer 2 broadcast domains. This worked well because Layer 3 segments could be internetworked using routers. So, if you had two Layer 3 segments, each consisting of Layer 2 bridges and hubs interconnected with a router, each Layer 3 segment would operate as its own STP domain, as you can see in this simple illustration:

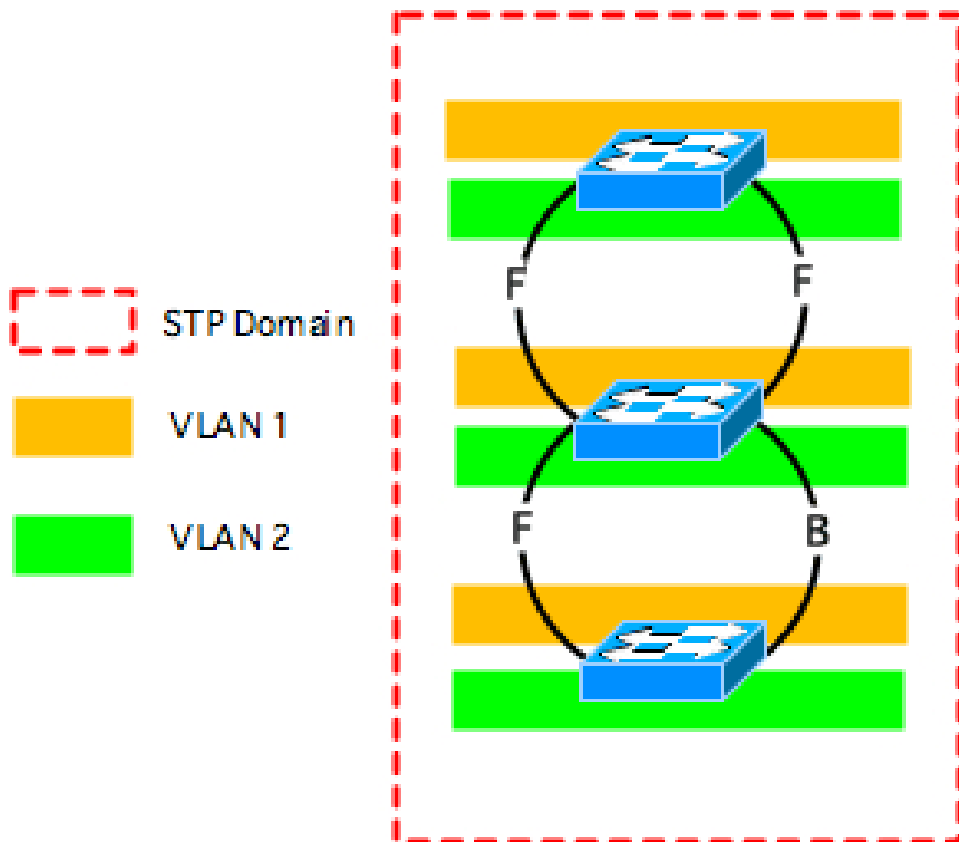


Each time STP recalculates, due to the addition of a switch or link failure, the bridges/switches stop forwarding until the network has stabilized. So in the event that a forwarding link failed and the backup link needed to come online, only the Layer 3 segment with the failure would be affected by the STP

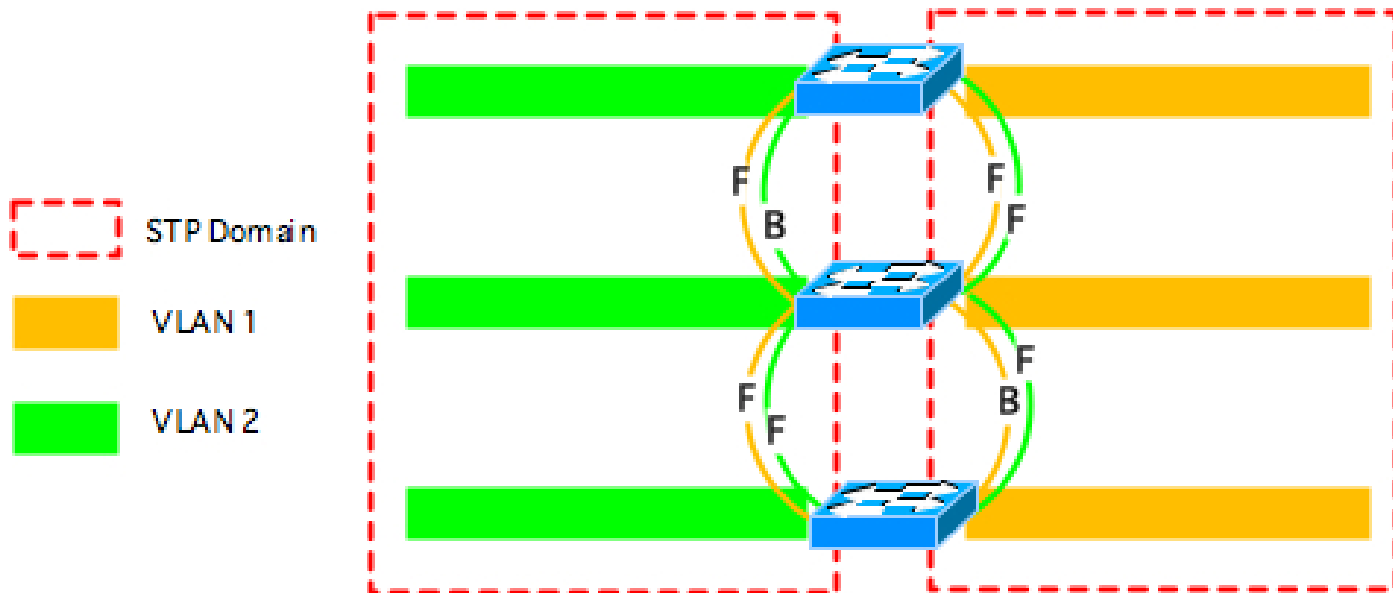
## Building 802.11Q VLANs

Michael J. Martin

recalculation. When VLANs were introduced, it became possible to have multiple Layer 3 broadcast domains existing within the same switch group, as illustrated in this figure:



The choice of supporting only a single STP domain began to show its broader impact. Because now when a link failed, all of the VLANs were affected, so a single Layer 2 link failure or switch addition could take down a whole network for up to 30 seconds. The 2003 update to the 802.1q standard added support for 802.1s and multiple spanning tree domains, which gave administrators the ability to configure different STP paths for each VLAN.

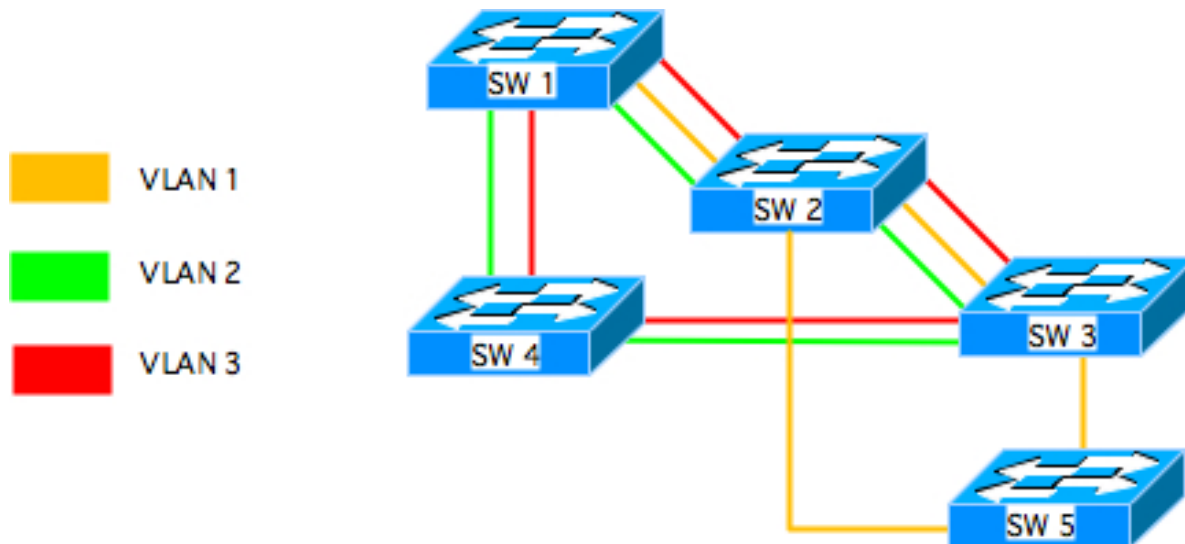


# Building 802.11Q VLANs

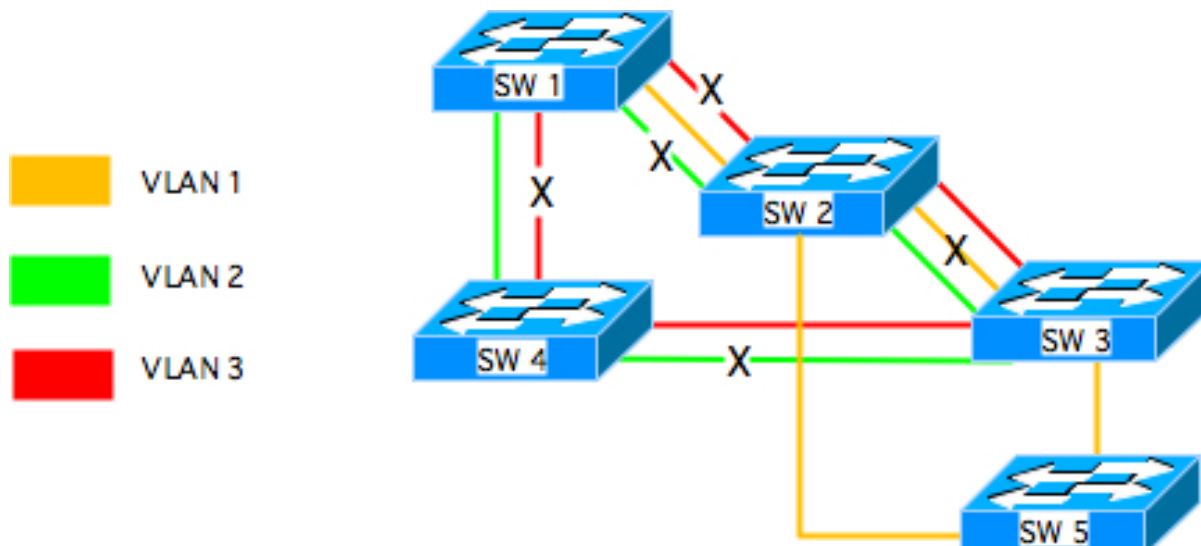
Michael J. Martin

The 802.1Q standard defines two types of switch ports: untagged and tagged. Untagged ports provide access to a single VLAN using one of the three configuration methods above. Tagged ports provide access to multiple VLANs. Tagged ports are primarily used to interconnect switches to provide multi-network transport over a single physical link. The same holds true for operating systems such as Linux that have driver support for 802.1Q interfaces.

Although tagged port interfaces are the preferred method for interconnecting switches that support VLANs, it is also possible to utilize multiple cross-connect links to interconnect switches. Here is a topology example using multiple cross-connections:



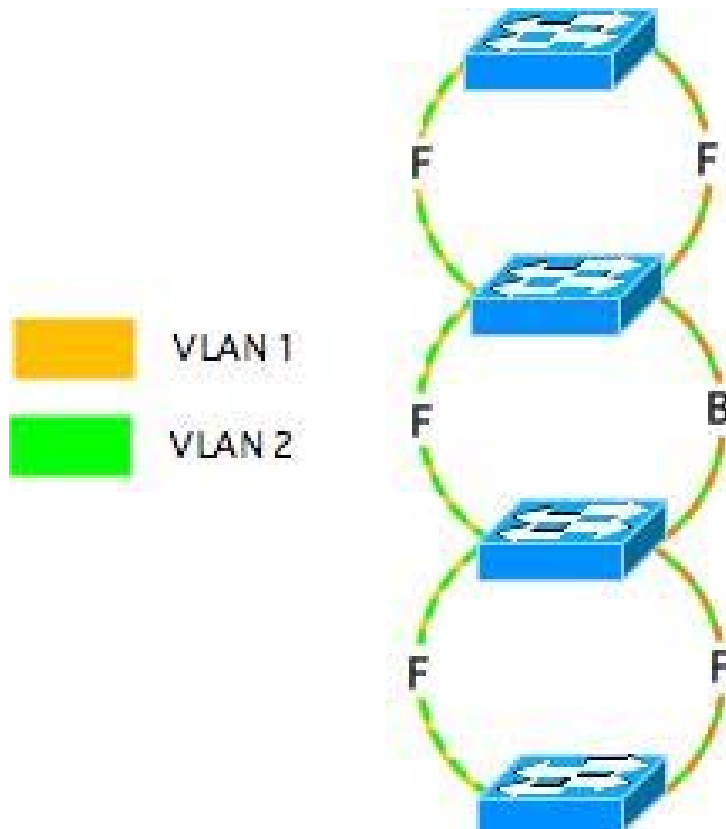
In this example, SW 1 through 3 support all of the VLANs, while SW 4 and 5 only support some. An interconnection topology such as this is utilized when all of the available switching interfaces are of the same bandwidth. Utilizing the discrete links minimizes the impact of oversubscription of the cross-connect links, giving each VLAN (in this case) a 100 Mbps path instead of aggregating all of the VLANs over a single 100 Mbps link. There is a small caveat to using this topology, which is the requirement for Cisco per-VLAN STP or 802.1s multiple STP support. If only 802.1d STP is utilized, only one crossover connection will be active between each switch in the LAN.



## Building 802.11Q VLANs

Michael J. Martin

This could result in only one VLAN being available across the switch core, or worse, none of the VLANs being available across the switch core, as illustrated above. The preferred option for interconnecting VLAN-capable switches is to use tagged port cross-connects:



When interconnecting switches made by different hardware vendors, keep in mind that the behavior of tagged ports varies depending on the vendor implementation. There are two schools of thought on this: all or nothing. The "all" school (Cisco) adds the tagged port to each VLAN group. The "nothing" school (Nortel, Extreme) requires a tagged port to be assigned manually to each VLAN group for which the port needs to transport traffic.

It is support for tagged interfaces that makes a device 802.1Q-capable. Switch and host interfaces configured as tagged interfaces transmit (with the exception of the "native VLAN" interface) tagged Ethernet frames. Tagged Ethernet frames contain four more bytes than standard Ethernet frames. These additional bytes support two additional frame fields: tag protocol ID (TPID) and tag control information (TCI). The IEEE 802.3ac standard defines these two additional frame extensions to accommodate VLANs. Here is the format of the 802.3ac frame:

7	1	6	6	2	2	2	42-1496	4
Preamble	SFD	DA	SA	TPID	TCI	Length	Data	FSC

Preamble is a standard pattern of 56 bits with alternating ones and zeros (1010101) that provides notification to hosts connected to the wire to synchronize themselves to receive an incoming frame. The preamble is discarded once a host receives the frame.

## Building 802.11Q VLANs

Michael J. Martin

Start of frame delimiter (SFD) is an alternating pattern of ones and zeros. The first six bits are 101010, and the last two bits are 11 (10101011). The SFD is used to break the sync alert to the hosts and informs the host that the next bit is the start of the frame's destination address. The SFD is also discarded once the frame is received. Both the preamble and SFD are not included in the frame size calculation utilized by the host for runt and giant frame error detection.

Destination address (DA) is a 6 byte address. The first three bytes of the address are the organizational unique identifier (OUI), which indicates the manufacture of the Ethernet card. The last three bytes are a unique number. Combined, they provide a unique host identifier address. The destination address indicates the station on the wire that receives the frame.

Source address (SA) is a 6 byte address. The first three bytes of the address are the OUI, and the last three bytes are a unique number. Combined, they provide a unique host identifier address. The source address indicates which station on the wire transmitted the frame.

TPID is always set to a value of 0x8100. It indicates the frame carries an IEEE 802.1Q or 802.1p tag. On a normal 802.3 frame, this bit segment is the beginning of the Type/Length field.

TCI is a 16 bit field containing three values. First is the user priority, which is a 3 bit field used by the IEEE 802.1p traffic prioritization standard. Next is the canonical format indicator (CFI), which is 1 bit and set to zero or one. The CFI is used to maintain compatibility between Ethernet and Token Ring. The field value indicates how the packet should be handled. A value of zero (default) means the packet should be forwarded as is. A value of one means it should forward the frame untagged. The last part is the VID, which is 12 bits in length, supporting 4096 possible VLANs. VLAN 0 and 4095 are reserved, leaving 4094 usable VIDs.

Length is a 16-bit field that contains a value no less than 46 or greater than 1500, indicating the size of the user data field in bytes.

Data is a field containing a minimum of 48 bytes and up to a maximum of 1500 bytes of user data. If the user data is less than 48 bytes, padding is added to make the field meet the minimum size.

Frame sequence check (FSC) is a 4 byte cyclic redundancy check (CRC) value computed by the source host when the frame is assembled. The CRC is computed from the destination field to the data field. The preamble, SFD and FSC values are not used as part of the calculation.

That's probably more than enough on the 802.1Q standard. Next month, we will cover supporting 802.1Q interfaces on Linux and Cisco IOS.