IEEE 802.1Q VLAN

Introduction

IEEE 802.1Q is a protocol for carrying VLAN (Virtual Local Area Network) traffic on an Ethernet. A VLAN is a type of local area network that does not have its own dedicated physical infrastructure, but instead uses another LAN to carry its traffic. The traffic is encapsulated so that a number of logically separate VLANs can be carried by the same physical LAN.

You should consider using VLANs whenever there is a need for traffic to be segregated at the link layer. For example, on Internet Protocol networks it is considered good practice to use a separate VLAN for each IP subnet. Reasons for doing this include:

- preventing a machine assigned to one subnet from joining a different one by changing its IP address: and
- avoiding the need for hosts to process broadcast traffic originating from other subnets.

Tagging

802.1Q VLAN frames are distinguished from ordinary Ethernet frames by the insertion of a 4-byte VLAN tag into the Ethernet header. It is placed between the source MAC and the EtherType fields:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18 ...
Destination address						Source address						VLAN tag				EtherType		Payload
												0x8100		TCI				

The first two bytes of the tag contain the TPID (tag protocol identifier), which is defined to be equal to 0x8100. Since it is located where the Ether Type would appear in an ordinary Ethernet frame, tagged frames appear to have an Ether Type of 0x8100.

The remaining two bytes contain the TCI (tag control information), of which 12 bits correspond to the VID (VLAN identifier, described below) and 4 bits contain metadata used for quality-of-service management.

VLAN numbering

Each 802.1Q VLAN is identified by a 12-bit integer called a VID (VLAN Identifier) in the range 1 to 4094 inclusive. The values 0 and 4095 are reserved and should not be used.

The first VLAN, with a VID of 1, is the default VLAN to which ports are presumed to belong if they have not been otherwise configured. It is considered good practice to move traffic off the default VLAN where practicable (see below).

The remaining values have no special status and can be used freely, but be aware that many network devices place a limit on the number of VLANs that can be configured so it will not necessarily be feasible to make use of all 4094 possible VIDs.

Trunk and access ports

There are two ways in which a machine can be connected to a switch carrying 802.1Q VLAN traffic:

- via an access port, where VLAN support is handled by the switch (so the machine sees ordinary, untagged Ethernet frames); or

- via a trunk port, where VLAN support is handled by the attached machine (which sees 802.1Q-tagged Ethernet frames).

It is also possible to operate a switch port in a hybrid mode, where it acts as an access port for one VLAN and a trunk port for others (so the attached Ethernet segment carries a mixture of tagged and untagged frames). This is not recommended due to the potential for VLAN hopping (described below).

Port configuration

The 802.1Q standard does not itself make any formal distinction between trunk and access ports. Instead, it allows the manner in which each VLAN is handled to be configured separately. The way this is typically presented is to allow each port to be in one of three states for a given VLAN:

State	Ingress	Egress
tag	allowed	allowed, will be tagged
untag	allowed	allowed, will not be tagged
non-member	prohibited	prohibited

In addition to this, each port has a PVID (Port VLAN ID) which specifies which VLAN any untagged frames should be assigned to. It may also be possible to specify which frame types are acceptable for ingress (tagged, untagged or both).

This method of configuration provides a lot of flexibility but be aware that just because a configuration is possible does not mean that it is useful or safe. For all but the most unusual purposes you should configure each port so that it is either an access port or a trunk port:

- For an access port there should be exactly one untagged VLAN (the one to be made accessible) and no tagged VLANs. The PVID should be set to match the untagged VLAN. If there is a choice, then the port should admit only untagged frames.
- For a trunk port there may be any number of tagged VLANs, but no untagged VLANs. Ideally the PVID should be set to a VLAN that does not carry any legitimate traffic, but this is not essential. If there is a choice, then the port should admit only VLAN-tagged frames.

Effect on the MTU

The MTU (maximum transmission unit) of a network interface is the size of the largest block of data that can be sent as a single unit. The standard Ethernet MTU is 1500 bytes at the network layer or 1518 bytes at the link layer, the difference being due to the 14-byte header and 4-byte frame check sequence that enclose the payload of an Ethernet frame.

On a VLAN trunk the need for each frame to be tagged adds a further 4 bytes of link-layer framing. This can be accommodated either by increasing the link-layer MTU or by reducing the network-layer MTU:

- To use the standard network-layer MTU of 1500 bytes, the equipment must support a link-layer MTU of at least 1522 bytes.
- If the link-layer MTU were limited to the standard value of 1518 bytes then the network-layer MTU would need to be reduced to 1496 bytes to compensate.

Devices with explicit VLAN support are supposed to accommodate a link-layer MTU of at least 1522 bytes, but if you are using generic hardware then it may be necessary to accept a lower value. All devices on a given IP subnet must use the same network-layer MTU, so if you intend to deviate from the standard value of 1500 bytes then you will need to configure all affected machines

Similar considerations apply when using jumbo frames. The link layer MTU is then much larger, but so is the potential payload, so allowance must still be made.

VLAN Stacking

Because an 802.1Q VLAN can carry arbitrary Ethernet traffic, it is in principle feasible to nest one VLAN within another. Possible reasons for doing this include:

- carrying more than 4094 separate VLANs on one physical bearer,
- simplifying the configuration of backbone switches, or
- allowing customer and service-provider VLANs to be administered independently of each other.

A basic 802.1Q-compatible switch cannot be used to add further tags to an already tagged frame, but there is an amendment to the standard called IEEE 802.1ad (also known as QinQ) which adds this capability. Note that VLAN stacking exacerbates the effect on the MTU, as each extra level adds a further 4 bytes of link-layer framing.

Linux support

Linux has the ability to use an Ethernet interface as an 802.1Q trunk port, allowing it to concurrently send and receive traffic on multiple VLANs. This is provided by the 8021q kernel module, which can be configured non-persistently using the ip link command:

```
ip link add link eth0 name eth0.2 type vlan id 2
```

or the older vconfig command:

```
vconfig add eth0 2
```

In both of these examples a virtual network device called eth0.2 would be created, bound to VLAN ID 2 of the physical interface eth0. This means that:

- Inbound 802.1Q-encapsulated frames arriving on eth0 with a VLAN ID of 2 are detagged, then re-presented to the network stack as inbound frames arriving on eth0.2.
- Outbound frames sent to eth0.2 are tagged with a VLAN ID of 2, then passed to eth0 for transmission.

VLAN devices can be configured and used in much the same way as physical network interfaces. For example:

- they can be given an IP address using ifconfig or ip add.
- they can be bridged; and
- they can be monitored using tools such as tcp dump or snort.

Linux makes the assumption that a virtual interface should be set to the same network-layer MTU as the physical interface to which it is attached. It does this whether or not the physical interface is capable of handling the resulting link-layer MTU. If it cannot be handled, or if you want to run with a smaller MTU for some other reason, then you will need to set it explicitly.

When using vconfig it is necessary for the virtual device name to follow one of four pre-defined naming schemes, which in the example above would result in a device name of eth0.2, eth0.0002, vlan2 or vlan0002. The ip link command is more flexible in that it allows any valid device name to be used.

Most GNU/Linux distributions provide a mechanism for persistently creating and configuring virtual interfaces; however, the method varies. See the microHOWTOs:

- *Configure an Ethernet interface as a VLAN trunk*
- *Configure an Ethernet interface as a VLAN trunk (Debian)*
- *Configure an Ethernet interface as a VLAN trunk (Red Hat)*

for detailed instructions.

Security considerations

VLAN Hopping

For most purposes, 802.1Q VLANs can be expected to provide a degree of isolation that is almost as good as would be provided by separate physical networks. This isolation is not complete because there will usuaully be competition for shared network bandwidth, but if the infrastructure has been securely configured then no traffic should be able to enter a VLAN unless it has been deliberately routed or bridged there by one of the connected hosts.

Care is needed to achieve this state of affairs, because some configurations can be exploited to inject unauthorised traffic into a VLAN. This practice is known as ‘VLAN hopping’ and is usually accomplished by:

- double tagging (described below), or
- somehow persuading a switch to reconfigure an access port as a trunk port.

Double tagging

One method than can sometimes be used to hop between VLANs is to construct a frame with two tags. If this traverses a VLAN-aware switch that has been poorly configured, then it may be possible to forward the frame onto a VLAN trunk with its outer tag removed but the inner tag intact and exposed.

Two conditions must be satisfied for this attack to be feasible:

- The egress port on the switch must operate in the hybrid mode described above, where traffic belonging to one of the possible VLANs is forwarded in untagged form.
- The ingress port must allow access to that VLAN by means of a tagged frame.

The inner tag should match the VLAN you want to hop to. The outer tag should match the VLAN that is untagged on the egress port.

An effective way to defend against this technique is to ensure that the conditions described above do not arise. Specifically, you should ensure that every active port is either:

- a trunk port which tags all outbound frames for all VLANs, or
- an access port which does not tag any frames.

As an additional protection, some switches may allow you to prohibit the ingress of tagged frames on ports that are supposed to be access ports (but be aware that whether frames are tagged or untagged on egress normally has no bearing on which frame types are acceptable for ingress).

Avoiding use of the default VLAN

It is considered good practice to move traffic off the default VLAN where possible, in order to minimize the extent to which an unconfigured switch port would give access to the network.

This does not mean that a network which uses the default VLAN is necessarily insecure and vacating it may not be feasible since some devices have functions that are hardwired to a VID of 1. However, if you have the opportunity to use a different VLAN then that is usually preferable.

Further reading

- Virtual Bridged Local Area Networks, IEEE Std 802.1Q-2005, IEEE Computer Society, May 2006
- 802.1Q VLAN implementation for Linux
- Virtual LAN Security Best Practices, Application Note, Cisco Systems
- VLAN Hopping, Hakipedia