

Supported by: **Viper 108** **Viper 408**

Switch operation

Introduction

A switch has to forward and receive packets from one LAN or device to another. The switch could forward all packets, but if this were the case it would have similar behavior to a hub.

It would be more intelligent if the switch only forwarded packets, which need to travel from one LAN or device to another. To do this, the switch must learn which devices or LANs are connected to each port. In simplistic terms; it needs to learn the destination and source ports of each and every packet received on each individual Switch port. Once learnt, any identically addressed packet will be automatically be forwarded.

Error Detection

The switch stores every incoming packet and scans this for errors, usually by checking the frame CRC (cyclic redundancy check sum). If any errors are found or detected the packet is discarded. In addition each frame is checked for size. Undersized packets (less than 64 Bytes) and oversized packets (more than 1518 bytes (*)) are also discarded. Once these basic checks have been carried out the switch can then start learning packet source and destination information.

(*) When implementing Ethernet MAC tagging maximum Ethernet packet length is increased to 1522 bytes.

Flooding

The switch needs to make a decision regarding which port(s) the packet is to be forwarded to. This decision is based upon the MAC tables that are maintained and updated automatically by the Switch. The process is known as Layer 2 Switching.

When first powered on the MAC tables within the Switch are empty. When a packet is received on a port the Switch does not know where the destination MAC address is located. The Switch learns the address by 'flooding' the packet out to all ports. Eventually, the destination node responds, the address is located and the Switch remembers the destination port. In simplistic terms; when a Switch receives a packet on a port it stores the source MAC address in the MAC table that corresponds to that Port. The flooding technique is always used with Broadcast and Multicast packets. If the switch is equipped with multicast management then multicast packets will not be flooded.

MAC Table and Packet Memory

The MAC table can hold up to 2 K entries with a MAC aging interval of five minutes. MAC aging means that a MAC address learned on a given port will be removed from the MAC table if no packets with this MAC address as the source MAC address are received on the port for approximately two minutes.

The total packet memory is 1Mbit. This means that 82 (maximum packet length – 1522 bytes) to 1953 (minimum packet length – 64 bytes) packets. The packet memory is used to handle short high load/overload situations. Exceeding the packet memory means that the switch engine will drop packets. Packet re-transmission is then required and must be handled by the end nodes (e.g. TCP).

A MAC table of 2 K entries and a packet memory of 1 Mbit is adequate for normal networks.

Full Wire Speed

The Switch supports full wire speed. This equates to 100Mbit/s full duplex on every port. 100Mbit/s in each direction on all ports equals 200Mbit/s per port.

MDI/MDI-X

There are two types of copper Ethernet ports available; MDI (Medium Dependant Interface) and MDI-X (Medium Dependant Interface Crossover). The MDI port types are associated with copper interfaces available on NICs (Network Interface Cards), PLCs, VSDs and DCSs etc. The latter type of interface (MDI-X) is found on Hubs or Switches.

In addition there are two types of Ethernet cable available. These are referred to as a 'straight through cable' or 'crossed cable'.

Auto MDI/MDI-X

Some switches have auto MDI/MDI-X this eliminates the need to source two types of patch cable (crossed and straight through). An auto MDI/MDI-X function detects the connected interface and if needed change the switch interface ether to MDI or MDI-X, therefore only one type of patch cable is needed.

Electrical Isolation

The copper (TX) ports incorporate high electrical isolation between the signal lines and the internal electronics. In addition, the switch can also withstand over 500 Amps through the shield for short periods of time (20-30mS) without effecting the operation and communication of the Switch. However; this is not advisable. Each TX port is isolated to chassis and other ports. Isolation is rated 1500Vrms (1 minute).

Auto-Negotiation

Auto-Negotiation is a protocol that controls the speed and duplex of a copper cable when a connection is established between two Ethernet devices. Auto-Negotiation detects the various modes that exist in the device on the other end of the cable and highlights its own abilities to automatically configure itself. Therefore, it will automatically operate at the highest performance in relation to speed and duplex. This allows simple and automatic connection of devices that support a variety of modes from a variety of manufacturers. The auto-negotiation protocol only functions on copper ports.

Supported by: **Viper 108** **Viper 408**

SNMP

SNMP software

Software used to communicate with the agent is called Network Management Solution (NMS). The exchange of data with the agents is similar to communication between a master and slaves, i.e. communication with the underlying devices takes place through polling. The manager can request information from or perform an action on the agent, this responds to the enquiries or actions requested. Another option is for the agent to set a "trap" i.e. an event controlled function that is activated by a predetermined condition. When this occurs the agent sends data back to the manager.

Devices or Ethernet Switches that support SNMP are usually referred to as Managed Switches.

- ⌘ There are currently three different versions of SNMP available; SNMPv1, SNMPv2 and SNMPv3. A SNMP enabled node incorporates a SNMP agent that is responsible for the following:
- ⌘ Collecting and maintaining information about the local environment and network.
- ⌘ Providing that information to a SNMP Master, either responding to a request or in an unsolicited fashion, or, when an event the managed device has been configured to monitor occurs.
- ⌘ Responding to manager commands to alter the local configuration or operating parameters.

SNMP, SNMPv2 and SNMPv3

There are three versions of SNMP. The original version of SNMPv1 has a multi security mechanism, which is a password. In version 1 you can not identify the sender of a message with all certainty. This makes SNMP open, which allows the reconfiguration of devices in the network. As a consequence of this many equipment manufacturers have chosen not to implement all the functions in the standard. These deficiencies were identified from the offset and a significantly improved version, SNMPv2, was planned.

This uses an encryption algorithm for authentication of transfers between the SNMP servers and agents. SNMPv2 can also encrypt the transfer. SNMPv2, which was intended as the follow-up was never accepted as a standard.

A contributing factor was the inability to reach agreement about how security should be implemented. However, SNMPv2 is an important link in the development of the next version, SNMPv3. The SNMPv3 work group was formed in March 1997 with the task to examine the submitted security and administration proposals and from this find a common solution to the problem. The focus of the work was, as far as possible, to complete the submitted proposals and not put forward any new ideas. The proposal for SNMPv3 was finished in 1998. This was based on version 2 as well as a security and administration concept that centered on different modules which could be switched depending on the level of security to be attained. SNMPv3, the current standard, provides many more opportunities to make network devices secure, yet introduction is slow. Most installed devices still follow SNMPv1.

MIB

Each agent in the network has a set of MIBs (Management Information Base), a MIB is an object that can be called by a manager. Information can either be standard information such as port status or port state, or company specific MIBs (private) for example the temperature inside the device. MIBs are structured tables made up of the different objects that can be called. The structure can be compared to a tree with a root and underlying directories. On the lowest level are directories for the standard MIB and for private MIBs.

SNMP implementation in Viper

In Viper we have support for SNMP v2c the following MIBs

The Viper MIB's are divided into groups allowing the SNMP manager to poll the SNMP agents for information. The following MIB groups are implemented and can be found on Westermo home page under [Downloads/Software/FIRMWARE](#)

- MIB-2 System Group, RFC1213-MIB, OID: 1.3.6.1.2.1.1. Contains generic configuration information such as system description (switch type, software version), location, hostname, etc on the switch CPU.
- MIB-2 Interface Group, RFC1213-MIB, OID: 1.3.6.1.2.1.2. Contains generic information on the entities at the interface layer. This means port speed, switch MAC address, and packets statistics (number of packets sent and received, number of unicast and multicast, packet sizes, over- and undersized packets, CRC errors, collisions, etc) per port on the switch.
- MIB-2 Internet Protocol Group (IP), RFC1213-MIB, OID: 1.3.6.1.2.1.4. Contains information used to keep track of the IP layer on the switch CPU.
- MIB-2 Internet Control Message Protocol Group (ICMP), RFC1213-MIB, OID: 1.3.6.1.2.1.5. Contains 26 counters, counting how many times this message type was generated by the local IP entity and how many times this message type was received by the local IP entity. It also counts the total number of ICMP messages received, sent, received in error; or not sent due to error on the switch CPU.
- MIB-2 Transmission Control Protocol Group (TCP) , RFC1213-MIB, OID: 1.3.6.1.2.1.6. Contains information used to keep track of the application entities using TCP on the switch CPU.
- MIB-2 User Datagram Protocol Group (UDP), RFC1213-MIB, OID: 1.3.6.1.2.1.7. Contains information used to keep track of the application entities using UDP on the switch CPU.
- MIB-2 SNMP Group, RFC1213-MIB, OID: 1.3.6.1.2.1.11. Contains information used to keep track of SNMP application entities. It provides statistical information about the SNMP protocol entity and tracks the amount of management traffic that the switch CPU responds to.
- BRIDGE-MIB dot1dBridge dot1dStp Group, RFC1493, OID: 1.3.6.1.2.1.17.2. This MIB holds Spanning Tree Protocol information on per port basis.
- ifMIB ifMIBObjects ifxTable Group, RFC2863, OID: 1.3.6.1.2.1.31.1.1. Contains network load on per port basis and represents an extension to the MIB2-Interface group.
- ifMIB ifRcvAddressTable Group, RFC2863, OID: 1.3.6.1.2.1.31.1.4. Contains network MAC table on the switch CPU.

Private MIB Information

The Viper private MIB contains the following:

- General
- FRNT 0 status
- Status information
- Multicast configuration
- Primary and secondary power source status
- SNMP host addresses

SNMP Traps

One feature of SNMP is that the SNMP agent can send SNMP traps to one or more SNMP Hosts. SNMP traps means system alarms such as a port link up/down or a port enabled for port alarms. When a trap is detected from an SNMP agent (e.g link loss in Viper) the agent will send the trap to the SNMP manager.

The address to the manager is defined by the trap host address. In Viper two host addresses is available.

Supported by: **Viper 408**

FRNT

Introduction

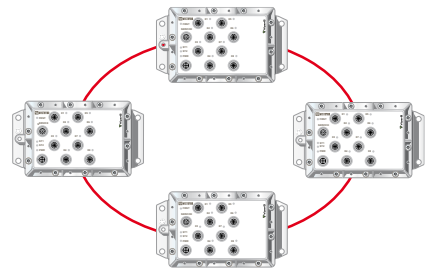
The Westermo industrial managed switch series are available with redundant ring technology. This eliminates network failure caused by copper failures on the trunk ports (ring ports). The speed of ring recovery is an essential part of designing your network. The FRNT (Fast Re-configuration of Network Topology) version 0 protocol can recover from a failure in only 20ms if such a failure does occur. When used in conjunction with redundant power supplies a very reliable system can be designed.

Standard Ethernet networks would collapse and fail if normal office based Ethernet Switches were formed into a complete ring. This failure is commonly referred to as a 'broadcast storm' as Ethernet Packets have multiple routes on a network to communicate to devices. Usually, an incorrect type of packet broadcasts (or floods) over a network and causes hosts to respond all at once, typically with wrong responses. This starts the process over and over again; hence your network crashes.

FRNT version 0 principles

The FRNT0 is similar to the IEEE Spanning Tree Protocol (STP) except for the following: Each switch in a ring topology has knowledge of the network topology.

I.e. not only to its neighboring switches as is the case for STP.



Event based principle

A FRNT topology change event packet will be sent directly to the focal point switch in case of a topology change (e.g. a link loss or a link establishment), while a STP implementation will only send STP control packets one network hop. The focal point switch will,

based on the received topology change event packet from the topology change detecting switches, generate a topology change command. This packet is sent to each member switch in the ring. The time it takes from the occurrence of a topology change until the corresponding topology change event packet is received on the focal point is typical a fraction of a ms or a few ms at the most, even though the number of the switches in the ring is high.

The FRNT0 concept contains in principle no limitation for the maximum number of FRNT0 enabled switches that can be installed in a single ring. The maximum number has been set to 200 switches.

20ms re-configuration time

A large number of switches in a ring have only a small impact on the re-configuration time of the network topology. The switch latency in the no load scenario is 17 microseconds (μs), while a conservative estimate in case of 50 % load is 70 microseconds (μs). This gives 3ms round trip propagation delay in a no load scenario and 14ms round trip propagation delay in the (conservative) high load scenario.

Full immunity vs. any type of network load

The loss of FRNT packets due to a network overload situation is not an issue for the FRNT0 control protocol. Thus, any unicast-, multicast- or broadcast network load can be generated on the network without any FRNT0 packet loss. An overload situation in this context means that the interface to the switch CPU is a network bottleneck. I.e. important control packets must compete vs. other packet to the CPU. Broadcast load is for all practical purposes the most critical network load in this context. The FRNT0 protocol is, however, protected against such an overload scenario due to the following properties:

- ⌘ Broadcast bandwidth limitation
- ⌘ FRNT packets defined as packet with highest possible QoS level.

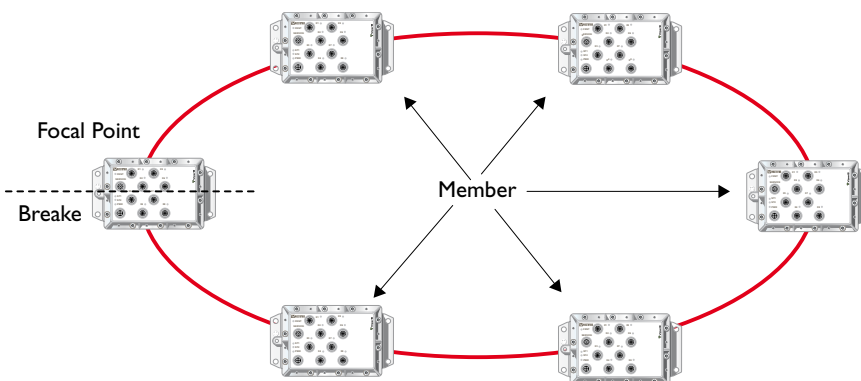
Similar proprietary network redundant protocols from other vendors are in most cases based on polling instead of event controlled handling of a topology change. This will introduce a slower establishment of a new topology. The FRNT0 protocol is also based on polling as a supplementary function to the event based part of the protocol. This function has only relevance in case of a multiple point of failure (single point of failure is handled by the of event controlled handling of a topology change).

Link qualification based on data link layer protocol, LHP

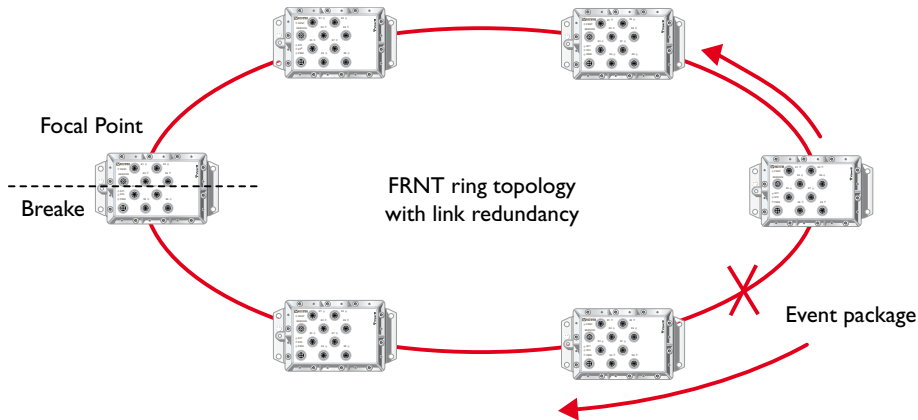
A major problem with most network redundancy protocols is the probability for having a network loop where a potential storm can be generated only in one direction (not both directions), and where this is not detected by the root (master) switch. This problem can only be handled if the switches support a link layer protocol that is used in order to qualify a link. The FRNT0 protocol has support for such a protocol. This Westermo protocol is referred to as the Link Health Protocol (LHP). The LHP makes sure that packets can be both sent and received on a trunk port before the link is properly qualified.

Fault in the ring

In a FRNT0 ring the focal point is the master of the ring, and there can only be one switch configured as focal point, all other is configured as member switches. When everything is working the focal point act as a virtual brake in the ring. The virtual brake prevents a loop in the ring.



When a fault occurs, the switches closest to the fault indicate that something has happened, the first level of detection is link detection, the second level is, lost health packages (according to LHP, see above). Normally there is idle traffic between all switches, if there is a brake in the idle traffic the switches will indicate this as a fault and send an error package to the focal point. The focal point then will re-configure the topology of the network and remove the virtual brake since here is a "real" brake in the system. Because of the reconfiguration the topology have now been changed from ring to bus. This is displayed in the tool, the exclamation mark also indicates where the fault is.



What happens if the focal point fails

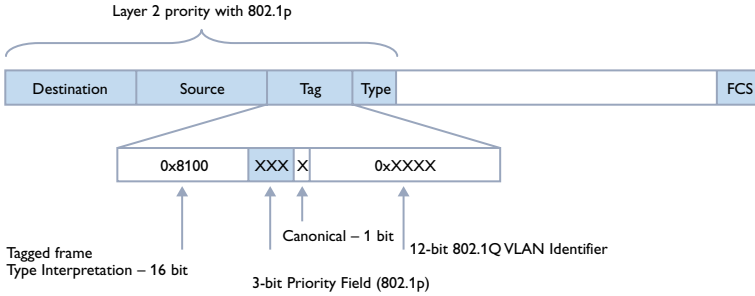
Since the focal point is the master of the ring this switch has the important function to control the complete application, but what happens if there is a fault in the focal point?

Since the switches closest to the focal point will not receive any idle traffic, they will try to send information to the focal point. Since the focal point will not respond they will become "edge switches" and take over all responsibilities.

Supported by: **Viper 408**

VLAN

A physical Ethernet network can be divided into several overlapping Virtual LANs (VLAN) without having IEEE802.1Q tagging support on the Ethernet end nodes.



All Ethernet trunk ports (FRNT or white ports) are member of all of the seven legal VLANs. A trunk port means a switch port connected to another switch; where a network redundancy protocol is running (e.g. FRNT). This means that the VLAN tables on each switch are dynamically updated during a network topology change. The VLAN implementation in Viper is meant for both Ethernet end nodes that support tagging and for those that do not.

VLAN Configuration

Name	Port Nr								Vlan Id	Pri
	1	2	3	4	5	6	7	8		
WHITE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1	7
RED	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	2	0
BLUE	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	3	3
GREEN	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	4	5
YELLOW	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	5	7
BROWN	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	6	0
PINK	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	7	0
Default	white	blue	red	blue	blue	green	white	white		
Remove Tag	X	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		

Apply

Restore Default Settings

Disable VLAN

Supported by: **Viper 408**

VLAN ID and colour code

In order to make the configuration easier we have implemented a colour code for each port. The colours of the ports define the logical belonging to a VLAN, then, the VLAN Id defines the actual VLAN address. The default drop down list chooses the colour for each port. You can select colour code for each port, if needed two ports can also have the same colour; in the picture below port 2, 4 and 5 belong all to the blue VLAN, then the blue VLAN id is 3.

White ports

If a port is defined as a white VLAN it becomes a trunk port, port 1 is always defined as white (see note 1 below) also the configured FRNT ports will become white.

Name	Port Nr								Vlan Id	Pri
	1	2	3	4	5	6	7	8		
WHITE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1	7
RED	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	2	0
BLUE	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	3	3
GREEN	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	4	5
YELLOW	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	5	7
BROWN	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	6	0
PINK	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	7	0
Default	white	blue ▾	red ▾	blue ▾	blue ▾	green ▾	white ▾	white ▾		
Remove Tag	X	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		

Note 1: Port 1 has the white VLAN id as the default and this cannot be changed. This port is configured in order to connect a node that is used for network management (SNMP or IP configuration). This type of node must always use the white VLAN (port 1) in order to communicate with the switch CPUs. The switch CPUs can always be accessed via port 1 with untagged packets.

This means that red, blue, green, yellow, brown and pink packets never will be sent to the switch CPUs.

This is important in order to avoid that the port between the switch fabric and the CPU becomes a bottleneck, where important packets might be lost (e.g. FRNT control packets). Example: a non-white broadcast load close to full wire speed is not a problem for correct switch CPU operation!

End nodes that do not support tagged package

An Ethernet end node that is not able to send tagged packets can, however, participate in one VLAN since the switch can add the tagged information when packages is sent out on the trunk port of the switch.

A VLAN id for a given (defined by the VLAN colour) default port will be associated to each untagged packet. This VLAN id will be added to packet as an IEEE802.1Q tag. This tag can be removed at the output port(s) if the port(s) is configured for tag removal.

Seven different VLANs are available:

- White, VLAN id	= 1, priority	= 7 [high]
- Red, default VLAN id	= 2, default priority	= 0 [low]
- Blue, default VLAN id	= 3, default priority	= 0 [low]
- Green, default VLAN id	= 4, default priority	= 0 [low]
- Yellow, default VLAN id	= 5, default priority	= 0 [low]
- Brown, default VLAN id	= 6, default priority	= 0 [low]
- Pink, default VLAN id	= 7, default priority	= 0 [low]

The tag is not removed on packets sent on a trunk port, and each trunk port is member of all seven VLANs. This means that the user does not need to set any VLAN parameters on the trunk ports, and that any network topology change will be handled automatically.

The layer 2 priority of a given VLAN can also be set. I.e.:

- Priority 0, 1: QoS level 1
- Priority 2, 3: QoS level 2
- Priority 4, 5: QoS level 3
- Priority 6, 7: QoS level 4

This priority will be added to the tag. See for the MAC header with tag.

The legal VLAN id range is [1 .. 4094]. A few VLAN ids in this range are reserved for other use. These id:s can be set in the Web tool but not in the IP configuration tool.

The network should only be based on switches enabled for VLAN or not. A mix of switches with and without VLAN support will not provide the user with the capability of tag removal on all parts of the network.

The figure below shows an example of a VLAN setup with three VLANs (red, blue and green VLAN) in a network with ring topology.

Supported by: **Viper 408**

IGMP

Multicast filtering

Several applications are based on multicast communication. Data is only sent once even though the data is meant for more than one receiver. However, the multicast packets will be sent on every drop link in the network unless the Ethernet switches support multicast filtering. The Viper 408 support IP multicast filtering based on IGMP snooping. This means that IP multicast "join" and "leave" requests will be trapped by the switches, and the multicast filters will be set based on which drop links where these requests are received.

Router less operation

IGMP "Join" and "leave" request are forwarded to the IGMP servers (routers) present in the network. This is one of the main properties of IGMP. The Viper IGMP snooping implementation does not depend on the availability of an IGMP Querier (IGMP server) in the network. This is important from robustness or a performance point of view. Thus, the multicast filtering feature will work even though the network connection to a standalone IGMP Querier is lost or not, and the drop link to such standalone IGMP Querier will not become a bandwidth bottleneck in the network, because the switch can also act as an IGMP Querier. The IGMP Querier operation of the switch is controlled by the "Auto mode" and "Querier" parameters. The following combinations of these two parameters are possible:

- "Auto mode" enabled + Querier enabled: the switch is able to act as an IGMP Querier (IGMP server) and the IGMP Querier in the network is selected automatically. The switch (with Querier support) in the network with the lowest IP address will be chosen as the network Querier (i.e. IGMP focal point). Only one Querier will exist in the network if all IGMP enabled switches and routers in the network have this configuration. This is the default IGMP settings.
- "Auto mode" enabled + Querier disabled: IGMP packages will be forwarded only
- "Auto mode" disabled + Querier enabled: the switch will always act as an IGMP Querier. Each switch/router will act as IGMP Querier if this configuration is used on each switch/router in the network.

A switch with "Auto mode" enabled, which is not acting as the IGMP Querier, will forward IGMP Queries received from the IGMP Querier on all ports except the port where the IGMP Queries are received. The port where IGMP Queries are received is referred to as the "Router port". This port is part of every active multicast filter. The use of "Router port" is not relevant in case "Auto mode" is disabled since the switch in this mode always is acting as a Querier (IGMP focal point). A switch in this mode will not forward IGMP queries received. IGMP Measurement reports for each active multicast filter on the switch will be sent back for each IGMP Query received. This is valid for both "Auto mode" being enabled and disabled.

The interval between two IGMP query packets can also be set in the IP configuration tool. Four intervals are possible: [12, 30, 70, 150] seconds.

The IGMP snooping implementation will also forward IGMP information (join, leave, measurements reports) on the switch trunk ports. A trunk ports is automatically detected in case a network redundancy protocol such as if FRNT is running, but the user may also configure manually ports as trunk ports. Manually trunk port configuration might be relevant in case no network redundancy protocol is running on a port connected to another IGMP snooping enabled switch. This feature is required in case the multicast producers (i.e. Ethernet end nodes sending IP multicast packets) make no IGMP join or IGMP measurements reports according to IGMP v2. IP multicast producers are not required to make an IGMP join during start up or answer with IGMP measurement reports on received IGMP query packets (ref. RFC 2236).

Stop filter option

A stop filter will be set if a multicast packet is received prior to a "join" to an IP multicast group where the received multicast address belong if the "Multicast stop filter" option is enabled. This means that IP multicasting based on IGMP is required in order to get multicast through the network. Multicast filters will be properly set only for IP multicast packets if this option is disabled. That means that multicast packets not based on IP will be forwarded in the same way as broadcast packets. This is acceptable if the non IP based multicast network load is reasonable low.

FRNT integration

The IP multicast filter implementation is integrated with the Fast Re-configuration of Network Topology (FRNT) protocol. This means that the multicast filters will be updated as fast the FRNT implementation handles a topology change, i.e. approx. 20 ms.

Description of IGMP querier

The querier in the network will issue the membership queriers in the network to scan for unit that wants to subscribe on multicast groups.

Description of fast re-connects

This option is only relevant when FRNT is enabled and with a increased load in the network the reconnection times will be decrease to less than 100 ms.

Definition of trunk port in IGMP

The membership reports will be forwarded on those ports and all multicast traffic will be transmitted on the trunk ports if fast re-connect is enabled.

Recommended querier intervals

When the querier interval expires the querier will send out a membership querier to check if the multicast streams should be stopped or redirected to other port. With low interval changes in the multicast topology will be detected faster but will the IGMP management load will be higher. Longer interval is the opposite, longer detection of multicast topology changes and less management load.

Enable IGMP Automode <input type="checkbox"/> Enable IGMP Querier <input type="checkbox"/>	Enable IGMP Automode <input checked="" type="checkbox"/> Enable IGMP Querier <input type="checkbox"/>	Enable IGMP Automode <input type="checkbox"/> Enable IGMP Querier <input checked="" type="checkbox"/>	Enable IGMP Automode <input checked="" type="checkbox"/> Enable IGMP Querier <input checked="" type="checkbox"/>
Invalid option	Proxy	Always Querier	Auto-mode

Description

Always querier: This unit will always act as a querier

Proxy: The unit will never act as querier but will forward membership queries and reports

Auto-mode: The switch with the lowest IP address will be selected as querier.

Supported by: **Viper 408**

MAC address filtering

This function should be used with care otherwise it can block all traffic through the switch and it will not be possible to access it anymore, a factory reset of the switch will be needed.

With the Mac filter function it is possible to allow certain mac addresses to pass through the switch. This function can be found in the menu under Configuration- > Mac filter.

The mac addresses can be added in 3 types of format as described below:

1. A single mac address in the format like 00:07:7c:12:34:56.
2. An address range can be added by using a star, *, as a wild card like 00:07:7c:12:34:**.

This will allow addresses between 00:07:7c:12:34:00 to 00:07:7c:12:34:ff.

3. Instead of adding many addresses one by one it is possible to add a string with addresses on the format under the point 1 and 2 and separate each address with a semicolon like
00:07:7c:00:00:00;00:07:7c:00:01:00;00:07:7c:00:0*:**;

Remember to add the mac address of the computer where the web browser is used. To be able to upgrade the switches remote all switches mac addresses has to be added to the mac filter tables of each switch, an easy way to do this is by adding a range of allowed mac addresses like 00:07:07:**:****, this wide rage will allow all mac addresses from Westermo.

Note: The switch has to be restarted before the Mac filter function will be enabled.

Supported by: **Viper 108** **Viper 408**

QoS

Principles of Deterministic Ethernet

Westermo switches can operate in full duplex mode. This ensures that an Ethernet controller will never see any collisions occurring when operated in such a manner. The core section of the Network; the redundant ring topology always runs full duplex and at 100Mbit/s (FE Viper); this cannot be altered.

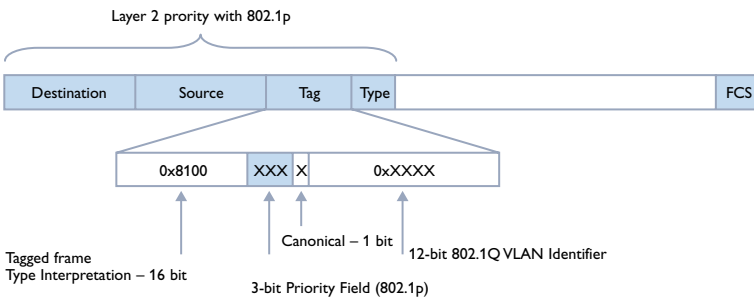
In addition a very fast switching core is provided to ensure that the switch can handle full wire speed on each port. Finally, a large buffer is available to store packets destined for a busy port. However, it is very unlikely that the buffers are used during normal network operation.

It should be noted that if buffers are used in such a network then it is not viable to state that a network is Deterministic. In practice, the only time such buffers maybe used is when 10 and 100 Mbit/s ports are combined. When various speeds are in use, a feature called Head of Line Blocking Prevention is automatically implemented to ensure critical data is received at the destination node.

The switch contains four priority queues. A packet that is identified as a high priority packet is put in the high priority queue. The switch alternates between the two queues by using strict priority. I.e. packets from the low priority queue are only sent if the high priority queue is empty. A packet is identified as a high priority packet based on priority tagging according to IEEE 802.1p (layer 2 priority) or IP Type of Service (ToS -layer 3 priority).

Layer 2 priority

The IEEE 802.1p and IEEE802.1Q standards specify an extra field for the Ethernet MAC header. This field is called Tag Control Info (TCI) field, and is inserted between the source MAC address and the MAC Type/Length field of an Ethernet packet, see figure below.



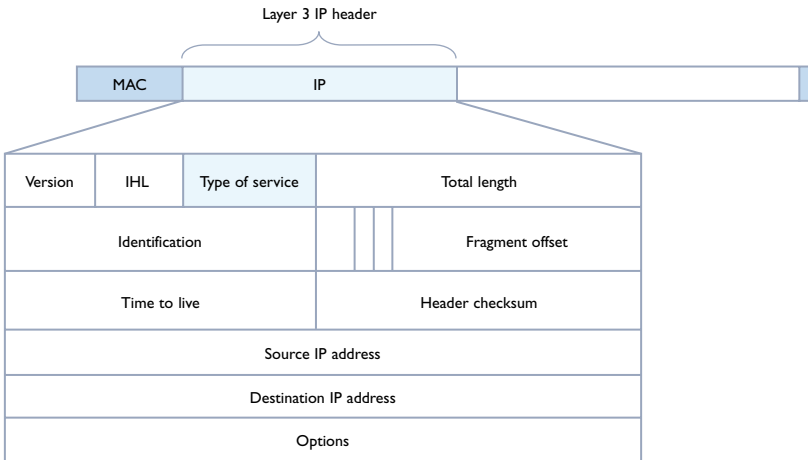
Layer 3 priority

Each IPv4 header contains a ToS field, see figure below. The switch is configured to put IP packets with the following ToS values in the following priority queues: nm

- 0x04 (IPTOS_RELIABILITY): QoS level 2
- 0x08 (IPTOS_THROUGHPUT) : QoS level 2
- 0x10 (IPTOS_LOWDELAY): QoS level 2
- 0xF1: QoS level 1 (lowest priority level)
- 0xF2: QoS level 2
- 0xF3: QoS level 3
- 0xF4: QoS level 4 (highest priority level)

High priority setting of the IP ToS field of real time critical packets must be set in the IP protocol of the sending station. This can be done on TCP/UDP socket level by a `setsockopt()` command both on the client and server socket side in most Operating Systems (OS). E.g.:

```
tos = 0xF4;
setsockopt( ..., IP_TOS, &tos,...)
```



Flow control

By default the switch is disabled for flow control (IEEE 803.3x), since flow control is not a good real time property.

Head of Line Blocking Prevention

The switch supports head of line blocking prevention for low priority packets only. This means that low priority packets received on any port will not be forwarded to ports that are congested. This will reduce the amount of packets in the output buffer. This function is particularly useful when high amounts of multicast, unknown unicast and broadcast traffic are available in large networks where 10Mbit/s and 100Mbit/s ports are used. High priority packets will always be forwarded.

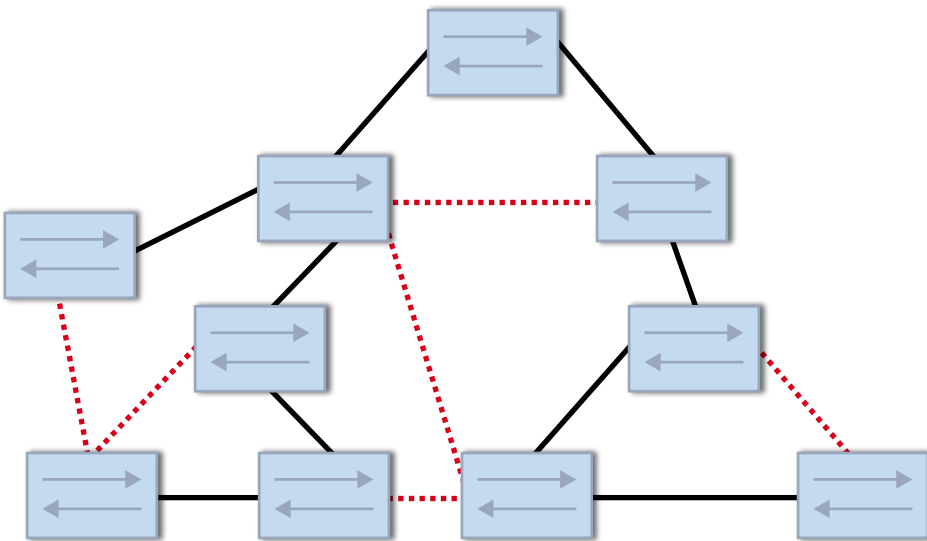
Supported by: **Viper 408**

STP/RSTP

Spanning tree protocol (STP)

The Viper and switch supports the Spanning Tree Protocol according to IEEE802.1D as an alternative to FRNT. This is a redundancy alternative within already installed applications using STP/RSTP.

STP is a Layer 2 link management protocol that provides path redundancy while preventing loops in the network. For a Layer 2 Ethernet network to function properly, only one active path can exist between any two stations. Multiple active paths among end stations cause loops in the network. If a loop exists in the network, end stations might receive duplicate messages with broadcast storm and an unstable network as result.



The STP uses a spanning-tree algorithm to select one switch of a redundantly connected network as the root (focal point) of the spanning tree. The algorithm calculates the best loop-free path through a switched Layer 2 network by assigning a role to each port based on the role of the port in the active topology:

- ⌘ Root – a forwarding port elected for the spanning-tree topology
- ⌘ Designated – a forwarding port elected for every switched LAN segment
- ⌘ Alternate – a blocked port providing an alternate path to the root port in the spanning tree

———— Best loop-free path in the application above

..... Redundant path in the application above

Switches that have ports with these assigned roles are called root or designated switches. Spanning-tree forces redundant data paths into a standby (blocked) state. If a network segment in the spanning-tree fails and a redundant path exists, the spanning-tree algorithm recalculates the spanning-tree topology and activates the standby path. Switches send and receive spanning-tree frames, called bridge protocol data units (BPDU:s), at regular intervals.

The switches do not forward these frames, but use them to construct a loop-free path. BPDU:s contain information about the sending switch and its ports, including switch and MAC addresses, switch priority, port priority, and path cost. Spanning tree uses this information to elect the root switch and root port for the switched network and the root port and designated port for each switched segment. When two ports on a switch are part of a loop, the spanning-tree port priority and path cost settings determine which port is put in the forwarding state and which is put in the blocking state. The spanning-tree port priority value represents the location of a port in the network topology and how well it is located to pass traffic. The path cost value represents the media speed. The stable, active spanning-tree topology of a switched network is determined by these elements:

- ⌘ The unique bridge ID (switch priority and MAC address) associated with each VLAN on each switch
- ⌘ The spanning-tree path cost to the root switch
- ⌘ The port identifier (port priority and MAC address) associated with each port

When the switches in a network are powered up, each switch functions as if is the root switch. Each switch sends a configuration BPDU through all of its ports. The BPDU:s communicate and compute the spanning-tree topology. Each configuration BPDU contains this information:

- ⌘ The unique bridge ID of the switch that the sending switch identifies as the root switch
- ⌘ The spanning-tree path cost to the root
- ⌘ The bridge ID of the sending switch
- ⌘ Message age
- ⌘ The identifier of the sending port
- ⌘ Values for the hello, forward-delay, and max-age protocol timers

When a switch receives a configuration BPDU that contains superior information (lower bridge ID, lower path cost, and so forth), it stores the information for that port. If this BPDU is received on the root port of the switch, the switch also forwards it with an updated message to all attached LANs for which it is the designated switch. If a switch receives a configuration BPDU that contains inferior information to that currently stored for that port, it discards the BPDU. If the switch is a designated switch for the LAN from which the inferior BPDU was received, it sends that LAN a BPDU containing the up-to-date information stored for that port. In this way, inferior information is discarded, and superior information is propagated on the network.

A BPDU exchange results in these actions:

- ⌘ One switch in the network is elected as the root switch (the logical centre of the spanning-tree topology in a switched network).
- ⌘ For each VLAN, the switch with the highest switch priority (the lowest numerical priority value) is elected as the root switch. If all switches are configured with the default priority (0x8000), the switch with the lowest MAC address in the VLAN becomes the root switch.
- ⌘ A root port is selected for each switch (except the root switch). This port provides the best path (lowest cost) when the switch forwards packets to the root switch.
- ⌘ The shortest distance to the root switch is calculated for each switch based on the path cost.
- ⌘ A designated switch for each LAN segment is selected. The designated switch incurs the lowest path cost when forwarding packets from that LAN to the root switch. The port through which the designated switch is attached to the LAN is called the designated port.
- ⌘ Ports included in the spanning-tree instance are selected. Root ports and designated ports are put in the forwarding state.
- ⌘ All paths that are not needed to reach the root switch from anywhere in the switched network are placed in the spanning-tree blocking mode.

The user can easily set the root of the network by configuring one of the switches in the network as the STP focal point (see Installation manual). This will result in a lower priority value for this switch than for the other switches in the network. Propagation delays can occur when protocol information passes through a switched LAN. As a result, topology changes can take place at different times and at different places in a switched network. When a port transitions directly from no participation in the spanning-tree topology to the forwarding state, it can create temporary data loops. Ports must wait for new topology information to propagate through the switched LAN before starting to forward frames. They must allow the frame lifetime to expire for forwarded frames that have used the old topology.

Each port on a switch using spanning tree exists in one of these states:

- ⌘ Blocking – the port does not participate in frame forwarding.
- ⌘ Listening – the first transitional state after the blocking state when the spanning tree determines that the port should participate in frame forwarding.
- ⌘ Learning – the port prepares to participate in frame forwarding.
- ⌘ Forwarding – the port forwards frames.
- ⌘ Disabled – the port is not participating in spanning tree because of a shutdown port, no link on the port, or no spanning-tree instance running on the port.

A port moves through these states:

- ⌘ From initialization to blocking
- ⌘ From blocking to listening or to disabled
- ⌘ From listening to learning or to disabled
- ⌘ From learning to forwarding or to disabled
- ⌘ From forwarding to disabled

The typical time it takes to enter forwarding state from blocking state or vice versa (i.e. the network re-configuration time) is approx. 40 seconds.

Supported by: **Viper 408**

RSTP implementation in Viper

Description of edge port:

For RSTP there are two kind of ports that can be configured, edge ports or not. An edge port is a port that is running RSTP and configured as an access port, i.e. only connected to an end node. An edge port will open the port immediately since there is no risk for loops an end node.

The behavior of a “non-edge” RSTP port is: This port is set in to blocking mode then wait for a RSTP packet. If it has not received a RSTP packet after 30 seconds it will open up the port. If it received a RSTP packet it will make a decision to open the port or not, depending on configuration.

Timing parameters

Hello time	2 seconds
Forward delay	15 seconds
Number of nodes	20
Root priority	Priority between 0 and 61440 (where 0 is the highest priority)

Static and dynamic trunking

Static and dynamic trunking, is something that is used when RSTP is enabled together with VLAN, otherwise it has no effect. When RSTP is enabled together with VLAN only on one RSTP process is running and not one for each VLAN. This can cause a problem in the network if the VLAN settings not are configured correctly so a redundant link in the network is not configured for all VLANs.

When dynamic trunking is used all redundant links in the network will be dynamically configured so all VLAN are allowed to be transmitted on that link so no traffic is stopped.

Static trunking is chosen if dynamic trunking is not activated and then the VLAN settings will not change dynamically during runtime. If a redundant link in the network is configured for only some VLAN that link could stop some traffic in the network if the configuration is not with care.