RCOMP - Redes de Computadores (Computer Networks)

2025/2026

Theoretical-practical lesson 03

- Virtual LAN (VLAN).
- VLAN Trunking Protocol (VTP).
- Multiple VLAN Registration Protocol (MVRP).
- Spanning Tree Protocol (STP).
- Link Aggregation – LACP.
- Ethernet flow control.

Virtual LAN

A **Virtual Local Area Network (VLAN)** is a part of a physical layer two infrastructure intended to act as an independent layer two network. Above all, a layer two frame sent through one VLAN will never reach another VLAN, or parts of the infrastructure not assigned to any VLAN, thus one VLAN is one single and **distinct broadcast domain** and behaves like **a separate LAN**.

Under all points of view, a VLAN must be equivalent to a physically separated LAN

There are several approaches for creating a VLAN, they all require the use of VLAN capable active layer two devices, usually switches.

A switch receives frames and retransmits them, by manipulating which frames are retransmitted from which ports to which ports, VLANs can be established.

As an example, take a switch with sixteen ports numbered from 1 to 16, we can instruct the switch to:

- for any received frame on ports 1 to 4, only retransmit them on ports 1 to 4.
- for any received frame on other ports, never retransmit them on ports 1 to 4.

With this settings, ports 1 to 4 will become a VLAN.

Port based VLAN

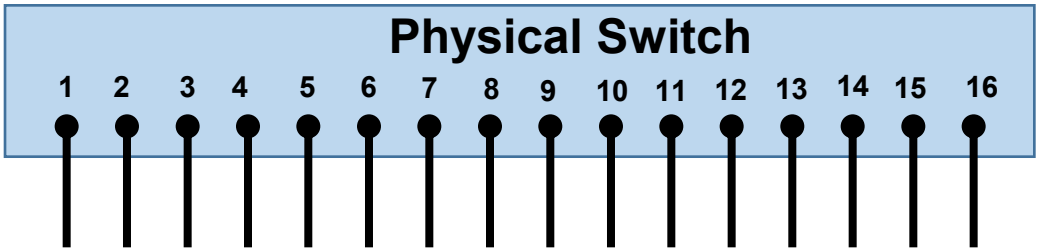
Defining VLANs as a subset of the switch ports may be seen as creating several virtual switches from a single physical switch. Picking again the previous sixteen ports switch example, we could define several VLANs, for instance:

By defining VLANs:

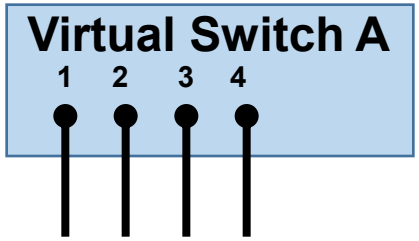
A – Ports 1,2,3,4

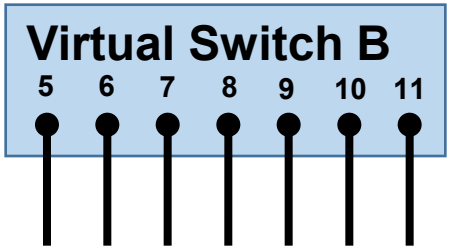
B – Ports 5,6,7,8,9,10,11

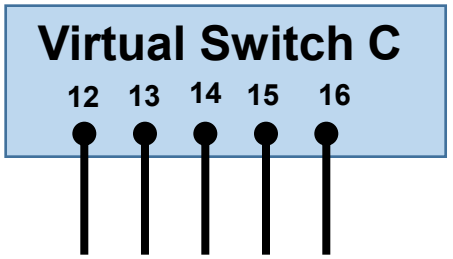
C – Ports 12,13,14,15,16



The equivalent to having three distinct switches (virtual switches) is achieved:







Although virtual, each of those switches will operate independently from others, no traffic will ever be transferred between them.

VLAN frame tagging

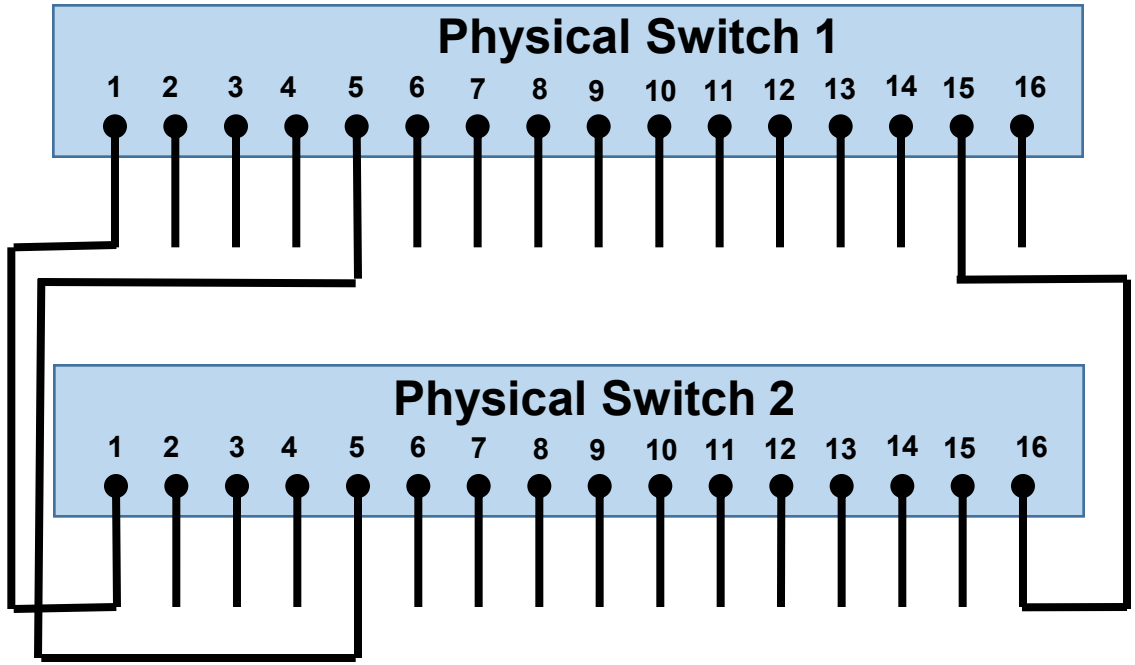
Older switches may support nothing else beyond what has been described, and we could live with that. But let us imagine we have two switches with the same defined VLANs as before and we want each VLAN on one switch to be connected to the same VLAN on the other switch, two ports, and one cable are required for each VLAN:

Defined VLANs on both switches:

A – Ports 1,2,3,4

B – Ports 5,6,7,8,9,10,11

C – Ports 12,13,14,15,16



Current switches allow a workaround to save a lot of hardware (ports and cables), it's called **VLAN frame tagging**. Such switches are capable of assigning the same port to several VLANs at the same time.

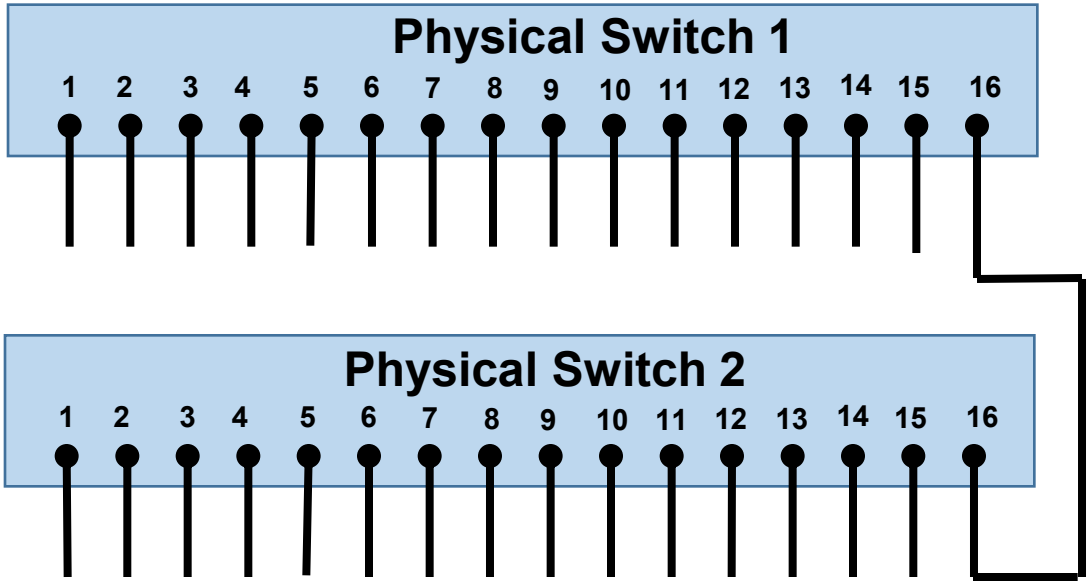
IEEE 802.1q

By assigning more than one VLAN to a switch port, that port can be used to send and receive frames belonging to different VLANs, however, they can´t be mixed. To accomplish that, frames are labeled (tagged) before being sent, when a frame is received, the label it carries is checked to see to which VLAN it belong to.

IEEE 802.1q establishes how VLAN tags can be inserted in frames before sending and removed after being received. The tag contains a VLANID, it's a 12-bits number that identifies the VLAN. The same VLAN must have the same VLANID on all switches so that frames are kept within the same VLAN.

Connecting the three VLANs on the previous configuration of two switches, now requires just one cable and two ports.

To achieve that all the three VLANs are assigned to port 16 of both switches and 802.1q VLAN frame tagging is used.



VLANID and the native VLAN

One important advantage of VLAN frame tagging is that it can be scaled up to any number of VLANs without additional hardware.

When a switch port is associated to several VLANs, tagging is required, yet one (and only one) VLAN is used without tagging (**untagged**), this is called the **native VLAN**. Of course, the absence of a tag works itself as a tag, thus the native VLAN must be the same on both ends of a cable connection.

VLANID are unique 12-bit numbers, 0 and 4095 are reserved and should never be used, all other VLANIDs ranging from 1 to 4094 can be used. On many devices VLANID=1 is reserved for dialogs between devices, it's called the **default VLAN**, usually, by default, it's also the native VLAN.

Depending on manufacturers, different terminologies may be used, of course, CISCO is a reference. On CISCO devices, associating a port to more than one VLAN is called **trunk-mode**, while associating a port to a single VLAN is called **access-mode**. Also, on CISCO devices, only VLANIDs from 1 to 1001 should be used, upper values are reserved.

Bear in mind that any device that is not VLAN capable will ignore tagged frames, the only VLAN they will be able to see is the native VLAN (untagged).

The VLAN database and VLAN names

On most VLAN capable devices, prior to any assignment of VLANs to ports, a VLAN database has to be created. Although VLANs are identified by the VLANID number, for human convenience they can also be named, an up to 20 characters unique name is recommended. VLAN names do not interfere with frame handling, only the VLANID really matters, names just make things easier for human administrators.

The VLAN database is a simple table defining VLAN names and for each the corresponding unique VLANID number. This allows the administrator to refer to VLANs by name instead of the VLANID number.

After creating the VLAN database, defined VLANs can be assigned to ports. On each port, one VLAN will be assigned in untagged mode (native VLAN), additional VLANs must be assigned in tagged mode.

In a layer two network infrastructure, all devices should share the same VLAN database, manually setting and keeping the VLAN database updated on several devices is a heavy and error exposed task. Several specific protocols were developed to automate VLAN configurations of devices spread throughout a layer two infrastructure.

VTP – VLAN Trunking Protocol

VTP is a very simple **CISCO proprietary protocol** aiming at spreading and keeping updated the VLAN database on all devices.

All devices that are supposed to share the same database must be belong to the same **VTP domain** (a name from 1 to 32 characters long).

In a VTP domain, at least one device must be configured in **VTP Server Mode**. The devices in VTP Server Mode manage the VLAN database, it's on those devices the administrator will manually define and change the VLAN database whenever required.

Devices configured in **VTP Client Mode** at the domain will acquire the VLAN database established by the servers but can't change it.

Devices can also be configured in **VTP Transparent Mode**, this means they will not acquire the VLAN database, however, they will retransmit the received VTP information to other devices.

VTP only operates through trunk-mode ports (multiple VLANs), this means devices must be interconnected by trunk-mode ports.

VTP – VLAN Trunking Protocol

By default, CISCO devices are in VTP Server Mode, with no VTP Domain defined.

When an unconfigured Cisco device (no VTP domain defined) is connected from an unconfigured port to a VTP enabled trunk-mode port of another Cisco device, then it will automatically set the connected port to trunk-mode and will also set the VTP domain to the one announced by the connected device.

For security's sake, VTP also supports setting an 8 to 64 characters long access password, this is optional, if set, them the same password must be manually defined on all devices belonging to the VTP Domain. VTP messages are exchanged through the **default VLAN** (usually VLANID=1), users' traffic shouldn´t have access to this VLAN anyway. Nevertheless, setting a VTP password creates an additional obstacle for an attacker trying to use VTP messages to interfere with devices' VLAN database.

Multiple VLAN Registration Protocol (MVRP)

MVRP (IEEE 802.1ak) is more ambitious than VTP, it allows the dynamic VLAN creation and dynamic port assignment to VLANs on any device.

MVRP does the same as VTP, it spreads to all devices VLAN information, however, it doesn't use the client-server model as with VTP. Instead, any device may create a new VLAN, MVRP will then ensure the new VLAN information will be acquired by all devices.

When an MVRP enabled device is connected to a network device port that also has MVRP enabled, it will acquire information about all existing active VLANs and may then connect to any of them. This is possible because MVRP can change the VLANs that are assigned to a device port.

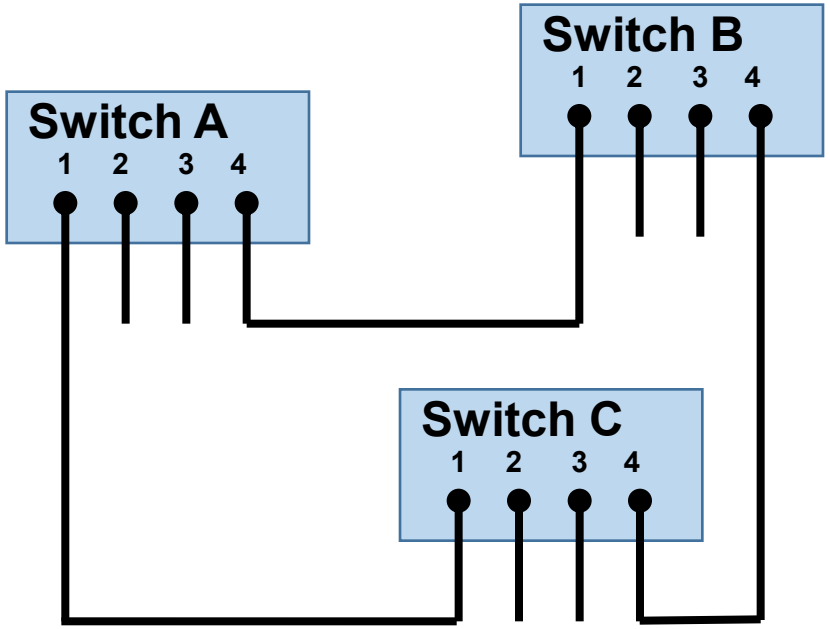
As with VTP, MVRP requires ports to be configured in trunk-mode.

Providing MVRP enabled ports to end users raises some security issues, in these cases, 802.1x standard should be used to authenticate and authorize user access to the VLAN, this is enforced prior to the MVRP negotiation. Depending on authentication, authorization may be granted to access only some specific VLANs.

Spanning Tree Protocol (STP)

Common layer two technologies like Ethernet are not prepared for redundant layouts. A redundant layout means there's more than one path to go from one point in the network to another point. Switches, however, don't select between alternative paths, they use all the paths at the same time; by doing so, frames are put into an infinite loop whenever there's a redundant layout.

The image represents a redundant layer two layout, if a frame is sent to this network, it will start an infinite loop being always retransmitted by each of the three switches. As more frames are sent more frames are added to the loop and **ultimately the network will be unusable due to excessive traffic**.



Spanning Tree Protocol (STP)

The Ethernet layer two intermediate nodes (switches) can´t take full advantage of redundant layouts, that would be **load balancing,** by distributing traffic throughout alternative paths. Nevertheless, **failover** is supported by the Spanning Tree Protocol (STP).

Failover means the existing alternatives are not all used at the same time, instead only one is used and the others are disabled or in a stand-by state, however, if the active alternative fails, then one of the others is activated.

The main goal of STP is removing loops by temporarily disabling some ports in some switches. STP is fully automated, and no configuration is usually required beyond enabling it (on switches that support STP it’s enabled by default).

To operate, STP uses special frames called Bridge Protocol Data Units (BPDU), the first step for STP is electing a root bridge (one switch). The elected will be the one with the lowers value for the MAC address combined with configurable bridge priority number.

Once the root device is elected, a tree starting from the elected switch is built reaching every other STP capable device within the layer 2 network.

Spanning Tree Protocol (STP)

Each branch on this tree represents a connection between two devices, each connection will have an associated cost, this cost can be configured on each switch port.

Based on the built tree, paths are defined to reach any other switch starting from the root, each path cost will be calculated as the sum of the costs of used links in the path. If more than one path exists to reach a device (a potential loop), a switch port will be disabled to block the alternative path with the highest cost. A disabled port will be in **Blocking mode**, this means it doesn't send frames, neither receive data frames, however BPDU will be received.

STP operates continuously and keeps rebuilding the tree, calculating the best paths and disabling other alternative paths, this means that if the currently active path fails, it will be removed from the tree, and the second-best path will be activated instead.

One issue on STP is the high **convergence time** (up to a minute), the convergence time is the time elapsed between a physical change on the network and the time when that is reflected by STP to make the network work again. So, the convergence time should be kept as low as possible.

Rapid Spanning Tree Protocol (RSTP)

RSTP is a simpler, more recent and more efficient version of STP. As the name states, is faster, the default convergence time is only 6 seconds. The convergence time depends on how frequently **Hello** messages are sent, this is configurable, usually, defaults to 2 seconds, convergence is granted after three **Hello** messages, thus resulting in 3x2 seconds convergence time.

RSTP is backward compatible with STP, when an RSTP device detects an STP BPDU in a port, that port will run in STP compatibility mode, this means it's ok to mix in the same layer two infrastructure STP and RSTP devices.

Comparing with STP, the number of possible states of a port has been reduced from five in STP to only three in RSTP:

Discarding – the port is blocked, yet BPDU frames are received

Learning – the port receives frames and updates the MAC table, however, received frames are not retransmitted.

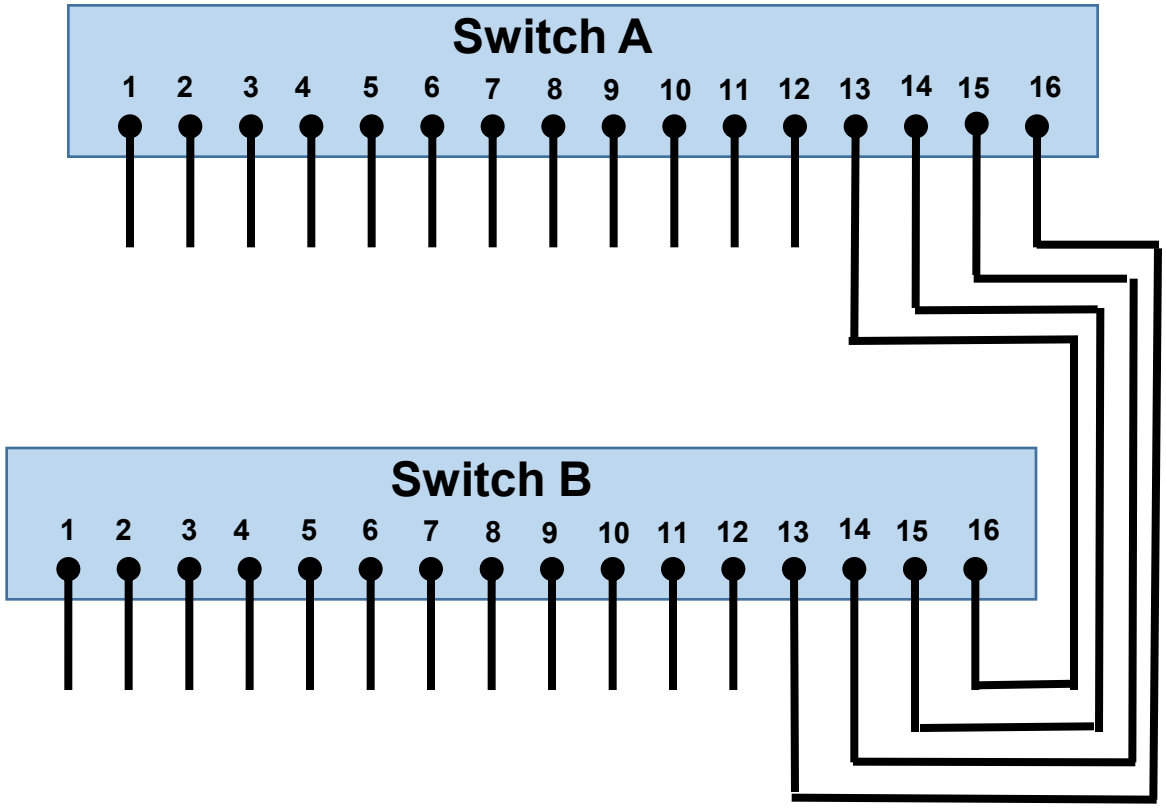
Forwarding – port in normal operation, frames are retransmitted.

Link Aggregation (port trunking)

Link aggregation is a technique in which a set of ports is configured to work together as a single port:

The main goal in increasing network traffic throughput by multiplying each port capacity by the number of aggregated ports.

Another advantage is the link becomes redundant, this depends on the protocol used to create the link aggregation, if one cable fails, it will be disabled, and the remaining cables will continue to be used.



Link Aggregation

Link aggregation does not mean a single frame can be transmitted faster, this is because each frame's transmission uses a single port/cable, yet the number of frames that can be transmitted per second will be multiplied by the number of ports in the link aggregation.

Link aggregation is an important tool for a more precise dimensioning of links capacity. Current Ethernet technologies have fixed transmission rates, for instance 1 Gbps, 2.5 Gbps, 5 Gbps, 10 Gbps, 25 Gbps, and 40 Gbps, by using link aggregation intermediate values can be attained.

This is also why, when designing a structured cabling system, it's so important that each backbone connection is made of several parallel cables. Even if at first, they are not used to create link aggregations, they may be in the future.

A link aggregation may be created manually, or automatically by using a specific protocol. The goal of a link aggregation protocol is detecting multiple physical direct links between two devices and configure them as a link aggregation.

Setting up a link aggregation between devices from the same vendor is usually simple, however, link aggregation between devices from different vendors may in some cases be trickier.

Manual Link Aggregation

Link aggregation is not specific for switches, end nodes like servers, workstations or routers, can also use it.

Since a long time, vendors are providing solutions for link aggregation, most of them require a manual setup on both devices at the ends of the link.

Vendor documentation must be followed carefully, typically, some restrictions apply may on some devices, for instance: multiples of four ports must be used, or only adjacent ports of the device may be used to create the link aggregation.

In addition to manual configuration, some vendors also provide protocols for automatically setting up the link aggregation, this is the case of CISCO proprietary **EtherChannel**. It may be manually configured providing load balance and cable fail detection, however, it may also be dynamically managed by CISCO Port Aggregation Protocol (PAgP).

Protocols for automated link aggregation management are able to detect all active direct cable links between two devices and use them to settle the link aggregation.

Link Aggregation Control Protocol (LACP)

LACP is defined by 802.3ad, however, it will be subject to further revision. This is nevertheless the most globally accepted standard for automated link aggregation.

Each LACP link aggregation can have up to eight ports, load balancing and cable failure is supported. Each LACP port must be configured in one of two ways:

> **ACTIVE** – when the link comes up, the port is used to send LACPDUs to the counterpart, inviting it to use LACP and make part of the aggregation.
>
> **PASSIVE** – activates LACP only if an invite is received, no invites are sent.

To enable LACP over a cable, an active port must be connected to an either passive or active port. After initial negotiation, LACP creates a link aggregation using up to 8 LACP enabled ports directly connected between the two devices.

Current CISCO devices also support LACP, in addition to EtherChannel.

On **Windows Servers** LACP link aggregations may be created by putting together several network interfaces into what is called a **NIC Team**. The same happens on **Linux Servers**, several network interfaces, can be used to create a single **Bonded NIC**. The same concept with different names.

Ethernet flow control

Flow control aims at avoiding a node congestion by receiving more data than it can handle. This can happen on an end-node if it's too slow on processing all data it's receiving but also on intermediate nodes like layer 2 switches.

On a switch congestion will happen when frames incoming through several ports, possibly at high data rates, and are all meant to be retransmitted through a single port, possibly at a lower data rate.

Both end-nodes and intermediate nodes have buffers to accommodate peaks of traffic, but they are not infinite. If the traffic peak situation persists the device gets congested and frames will be lost, simply because there is no where to store them anymore.

Usually flow control is implemented together with error control, in that case the receiver controls the flow by sending or not ACK frames, the sender is required to wait for a receiver ACK, this can be implemented in stop & wait mode or sliding window protocol mode.

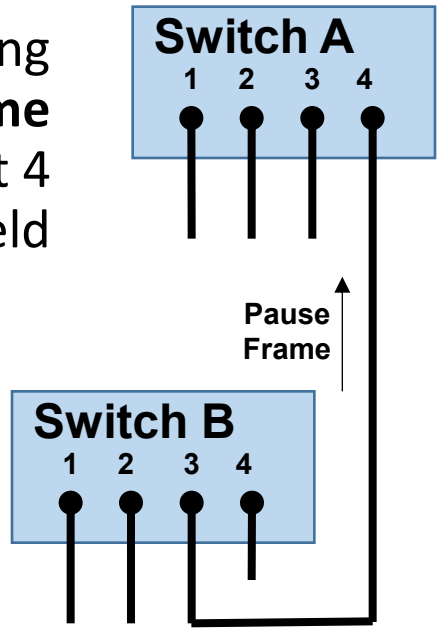
Yet, another way to control the flow can be used, the appropriate expression to describe it is **congestion notification**.

Ethernet flow control

With congestion notification, when a node is overwhelmed by incoming traffic it sends a message to the counterpart requesting it to stop sending. Ethernet flow control is based on this principle.

Ethernet flow control is implemented between ports of directly connected Ethernet switches.

In the image, if Switch B is overwhelmed with traffic arriving from Switch A through port 3, it may send a **Pause Frame** through port 3. Switch A will then stop sending through port 4 during some time. The **Pause Frame** contains a field specifying for how long should the pause be.



Ethernet flow control is supported by most manufacturers, however, is not often used. Because the flow control is implemented in a local link base, it lacks the knowledge about the real destination of frames, thus it will affect all traffic, including traffic that is not causing any congestion.

Ethernet flow control

In a switch, congestion happens when frames are to be sent, namely when there are more frames to be send than the required output port can handle.

Sending a **Pause Frame**, however, will stop all incoming traffic including frames that were not meant to the troubled output port and could be retransmitted without any problem.

Also, sending a **Pause Frame**, transfers the problem to the device receiving it. If a device is requested to stop sending, then it will also be in trouble to store all frames it was previously sending.

We must also question if layer two flow control is that important, the most frequent heavy traffic transactions are made using TCP (Transmission Control Protocol) at layer four. TCP is a reliable protocol with error control (recovery) and flow control based on the sliding window protocol; it also implements congestion detection mechanisms based on packets lost. This means when a switch reaches the congestion point and starts losing frames, that will be detected by TCP and will automatically reduce or stop the data flow.

Congestion detection by TCP is far more effective because it handles communications between end-nodes, thus it does not affect other nodes traffic.