

Ethernet Switches

In this lesson, we'll gain a brief introduction to Ethernet switches.

We'll cover the following ^

- Introduction
 - MAC Address Tables
 - Retaining Plug & Play with Switches
 - MAC address learning algorithm
 - Pseudocode
 - Timestamp
 - Unicast, Broadcast, & Multicast Frames
- Handling Failures
 - Spanning Tree Protocol
- Quick Quiz!

Introduction

Increasing the physical layer bandwidth as in Fast Ethernet was only one of the solutions to improve the performance of Ethernet LANs.

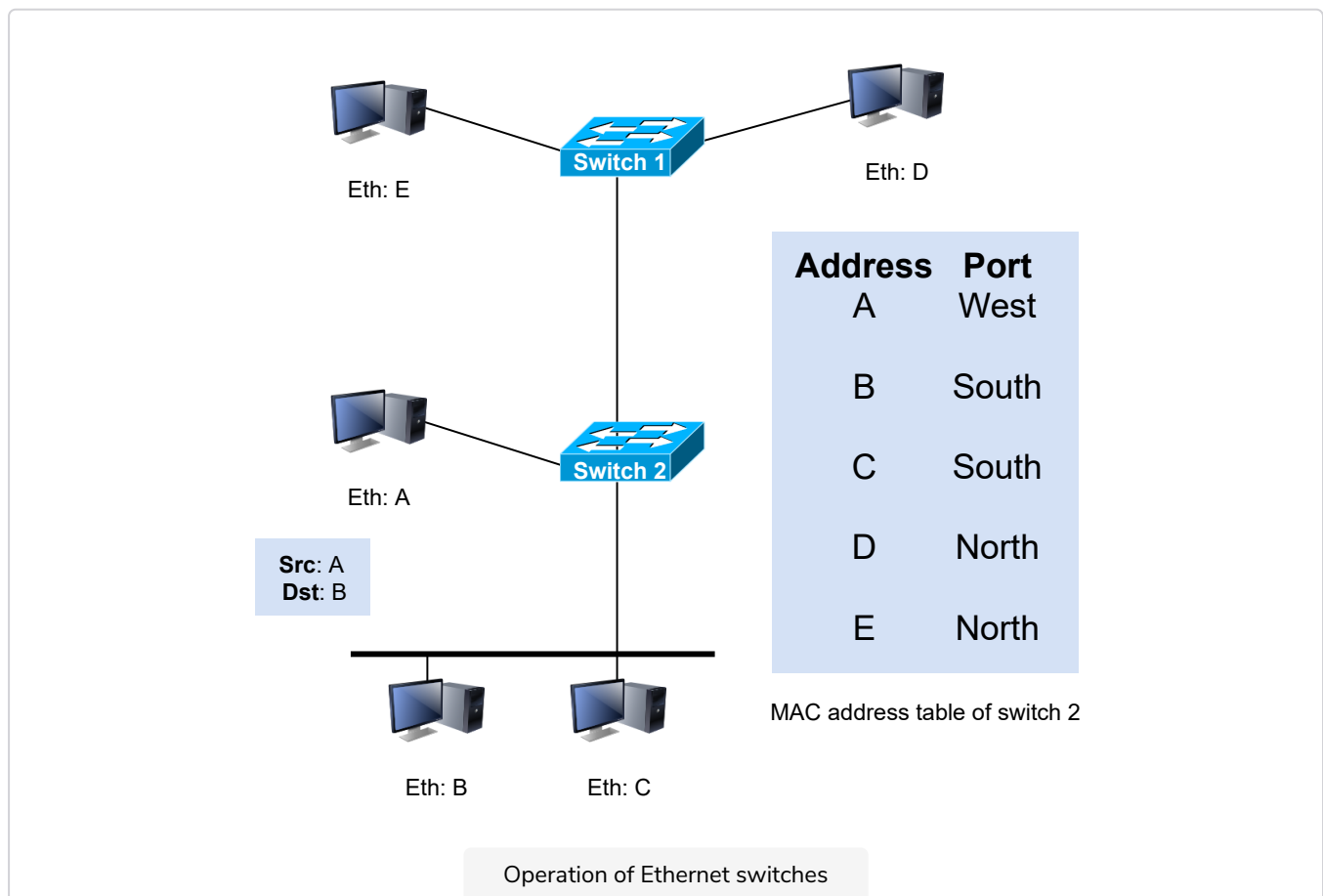
A second solution was to replace the hubs with more intelligent devices. As Ethernet hubs operate in the physical layer, they can only regenerate the electrical signal to extend the geographical reach of the network. From a performance perspective, it would be more interesting to have **devices that operate in the data link layer** and can analyze the destination address of each frame and forward the frames selectively on the link that leads to the destination. This would allow two hosts to communicate on one pair of interfaces while other pairs of interfaces can be simultaneously used for other communication, thereby improving communication efficiency. Such devices are usually called **Ethernet switches**. An **Ethernet switch is a relay that operates in the data link layer**.

MAC Address Tables

An Ethernet switch understands the format of the Ethernet frames and can *selectively* forward frames over each interface. For this, each Ethernet switch maintains a **MAC address table**. This table contains, for each MAC address known by the switch, the identifier of the switch's port over which a frame sent towards this address must be forwarded to reach its destination.

This is illustrated below with the MAC address table of **switch 2**.

- When the switch receives a frame destined to address B, it forwards the frame on its South port.
- If it receives a frame destined to address D, it forwards it only on its North port.



Retaining Plug & Play with Switches

One of the selling points of Ethernet networks is that thanks to the utilization of 48 bit MAC addresses, an Ethernet LAN is plug and play at the data link layer. When two hosts are attached to the same Ethernet segment or hub, they can immediately exchange Ethernet frames without requiring any configuration.

MAC address learning algorithm

It is important to retain this plug and play capability for Ethernet switches as well. This implies that **Ethernet switches must be able to build their MAC address table automatically without requiring any manual configuration.** This automatic configuration is performed by the **MAC address learning algorithm** that runs on each Ethernet switch.

1. This algorithm extracts the source address of the received frames and remembers the port over which a frame from each source Ethernet address has been received.
2. This information is inserted into the MAC address table that the switch uses to forward frames.
3. This allows the switch to automatically learn the ports that it can use to reach each destination address, provided that this host has previously sent at least one frame. This is not a problem since most upper-layer protocols use acknowledgments at some layer and thus even an Ethernet printer sends Ethernet frames as well.

Pseudocode

The pseudocode below details how an Ethernet switch forwards Ethernet frames. It first updates its MAC address table with the source address of the frame.

Timestamp

The MAC address table used by some switches also contains a timestamp that is updated each time a frame is received from each known source address. This timestamp is used to remove from the MAC address table entries that have not been active during the last n minutes. This limits the growth of the MAC address table, but also allows hosts to move from one port to another.

```
# Arrival of frame F on port P
# Table : MAC address table dictionary : addr->port
# Ports : list of all ports on the switch
src=F.SourceAddress
dst=F.DestinationAddress
Table[src]=P # src heard on port P
if isUnicast(dst):
    if dst in Table:
        ForwardFrame(F,Table[dst])
    else:
```



```

1133:         for o in Ports:
1134:             if o!= P: ForwardFrame(F,o)
else:
    # broadcast destination
    for o in Ports:
        if o!=P: ForwardFrame(F,o)

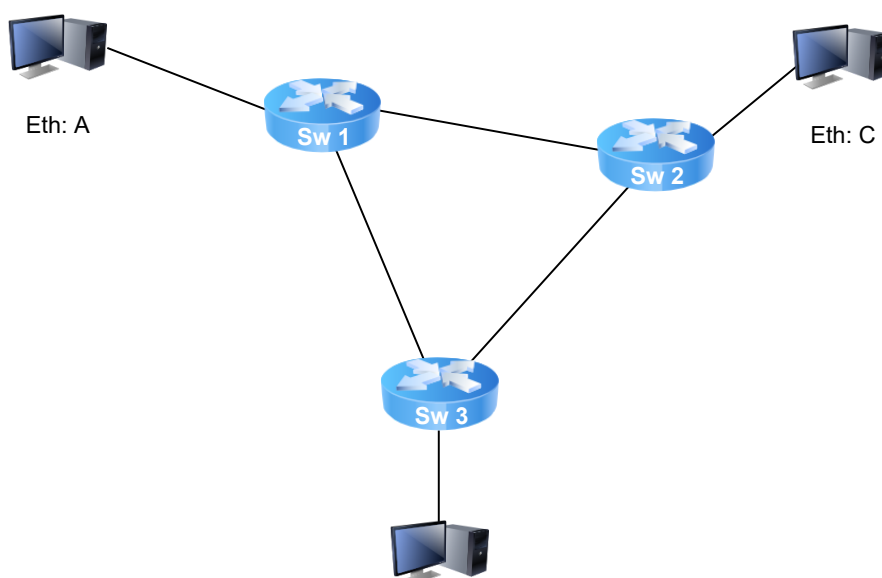
```

Unicast, Broadcast, & Multicast Frames

The switch uses its MAC address table to forward the received unicast frame. If there is an entry for the frame's destination address in the MAC address table, the frame is forwarded selectively on the port listed in this entry. Otherwise, the switch does not know how to reach the destination address and it must forward the frame on all its ports except the port from which the frame has been received. This is called **flooding** and it ensures that the frame will reach its destination, at the expense of some unnecessary transmissions. These unnecessary transmissions will only last until the destination has sent its first frame. Multicast and Broadcast frames are also forwarded in a similar way.

Handling Failures

The MAC address learning algorithm combined with the forwarding algorithm work well in a tree-shaped network such as the one shown above. However, to deal with link and switch failures, network administrators often add redundant links to ensure that their network remains connected even after a failure. Let us consider what happens in the Ethernet network shown in the figure below.



When all switches boot, their MAC address table is empty. Assume that host A sends a frame towards host C.

- Upon reception of this frame, switch 1 updates its MAC address table to remember that address A is reachable via its West port.
- As there is no entry for address C in switch1's MAC address table, the frame is forwarded to both switch 2 and switch 3.
- When switch 2 receives the frame, it updates its MAC address table for address A and forwards the frame to host C as well as to switch 3. switch 3 has thus received two copies of the same frame.
- As switch 3 does not know how to reach the destination address, it forwards the frame received from switch1 to switch 2 and the frame received from switch2 to switch1 and so on...
- The single frame sent by host A will be continuously duplicated by the switches until their MAC address table contains an entry for address C.
- Quickly, all the available link bandwidth will be exhausted to forward all the copies of this frame.
- As Ethernet does not contain any TTL or Hop Limit, this loop will never stop.

The MAC address learning algorithm allows switches to be plug-and-play. Unfortunately, the loops that arise when the network topology is not a tree are a severe problem. Forcing the switches to only be used in tree-shaped networks as hubs would be a severe limitation.

Spanning Tree Protocol

To solve this problem, the inventors of Ethernet switches have developed the **Spanning Tree Protocol**. The spanning tree protocol enables switches to exchange control messages by means of which a root switch is first elected in the topology. Then, all switches designate one of their ports that reaches the root switch with the minimum number of hops to be part of the spanning tree

root switch with the minimum number of hops to be part of the spanning tree. All other ports that lead to the root switch, and hence create a loop are disabled as far as frame forwarding is concerned. Eventually, the frames are only forwarded on ports that are part of the spanning tree. The ports that are not part of the spanning tree continue to send and receive control frames. This helps to recover from switch or link failures.

Quick Quiz!

1

Consider the first network given in this lesson. Suppose host D joins it later than the rest and immediately sends a frame to host C. Will host A then be able to send it a frame right after?

☐ A) Yes

☐ B) No

COMPLETED 0%

1 of 2



In the next lesson, we'll study virtual LANs!