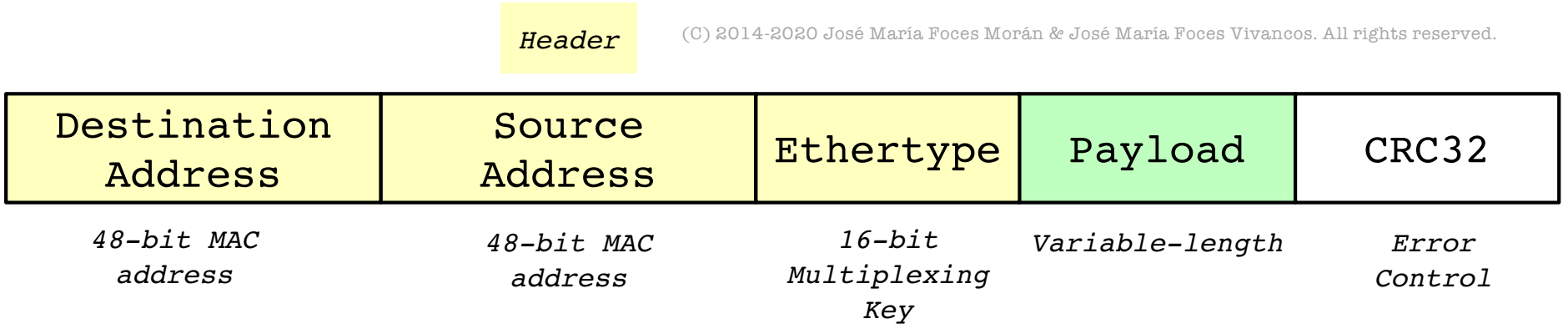
LAN SWITCHING

V 2.1 30th/April/2020

Ethernet Frame

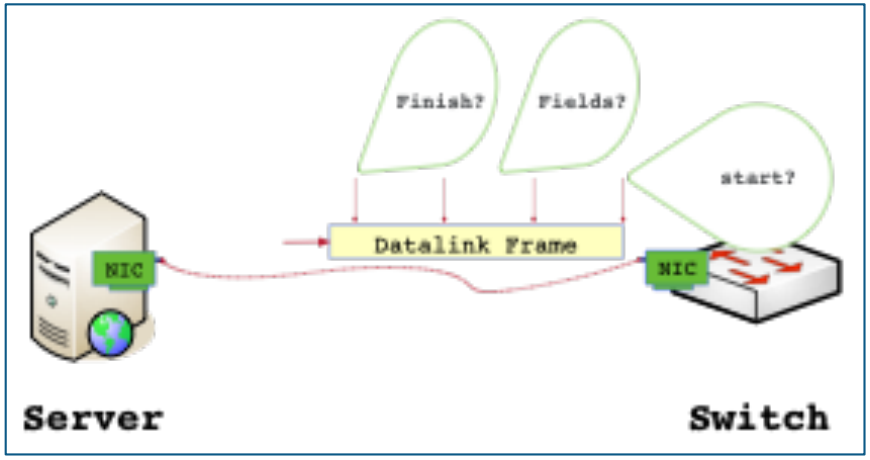


Ethernet frame

Ethernet Frame

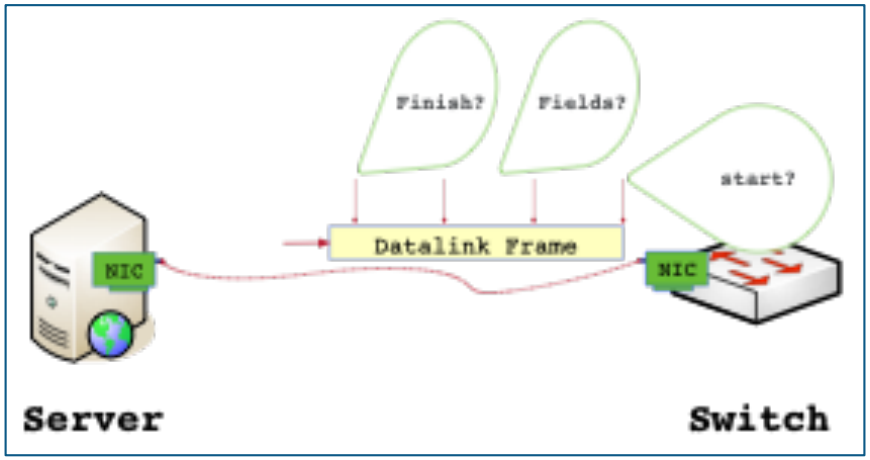
Destination Address	Source Address	Ethertype	Payload	CRC32
48-bit MAC address	*48-bit MAC address*	*16-bit Multiplexing Key*	*Variable-length*	*Error Control*

- MAC Addresses
  - Unicast: Represents a single station
  - Broadcast: All stations in LAN
    All 1's address: 0xffffffffffff
  - Multicast: A subset of stations



NIC: Network Interface Card

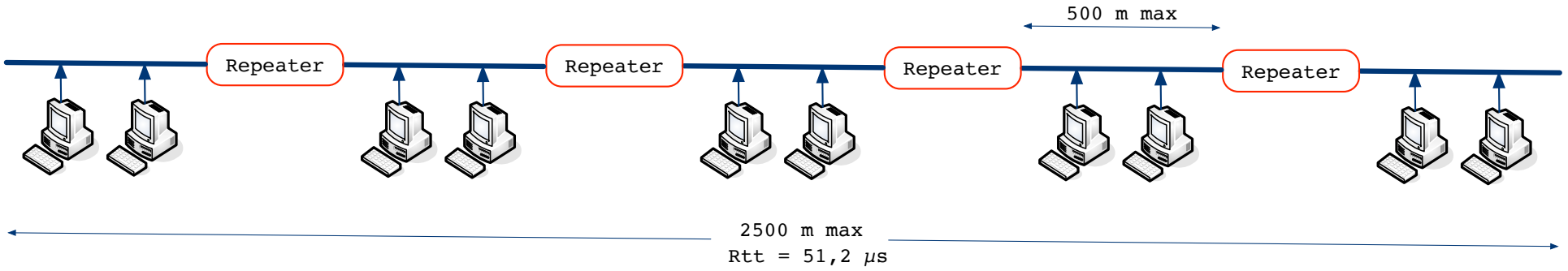
- Each NIC:
  - NIC has a MAC address of its own
  - Sends frames
  - Receives frames
- Frames accepted by a NIC
  - If promiscuous mode is **SET**
    - All frames
  - If promiscuous mode is **NOT** SET
    - Only frames which Dest MAC is == NIC's MAC
  - All frames sent to broadcast



Limitations of Ethernet

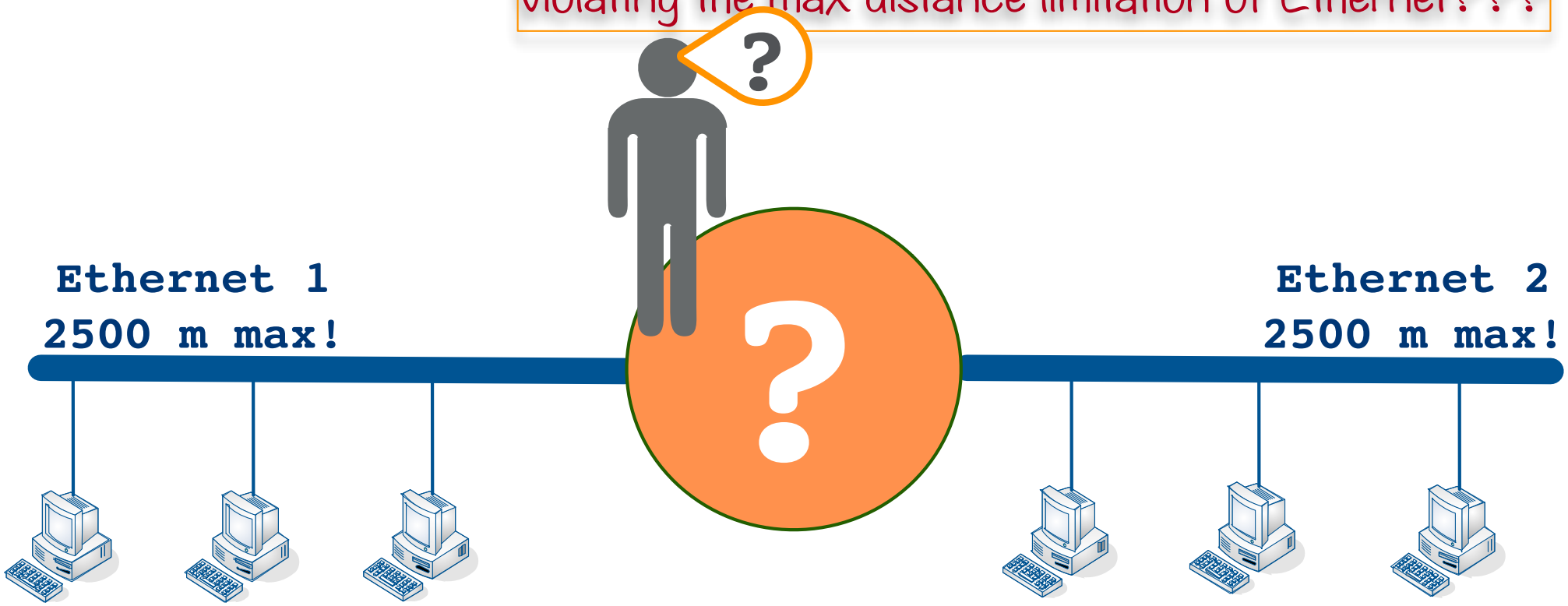
Multiple Ethernet segments can be joined together by using *repeaters*.

- A *repeater* is a device that *regenerates digital signals*.
  - No more than four repeaters
  - 2500m max
  - Limited total number of stations (Computers)
- Broadcast media: A sent frame is received by all the stations, necessarily
- Half duplex: Only one frame can be being transmitted at any one time
- HUB = Multiport *repeater*

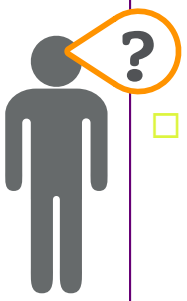


Communicating two max Ethernets

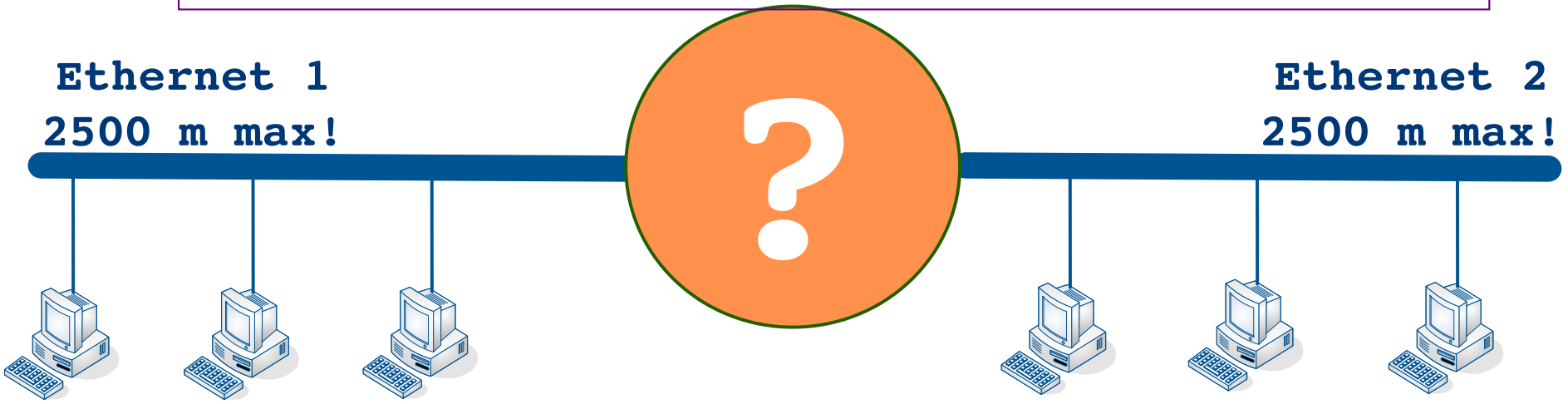
What network device can communicate E1 with E2
without
violating the max distance limitation of Ethernet???



Connecting two max Ethernets



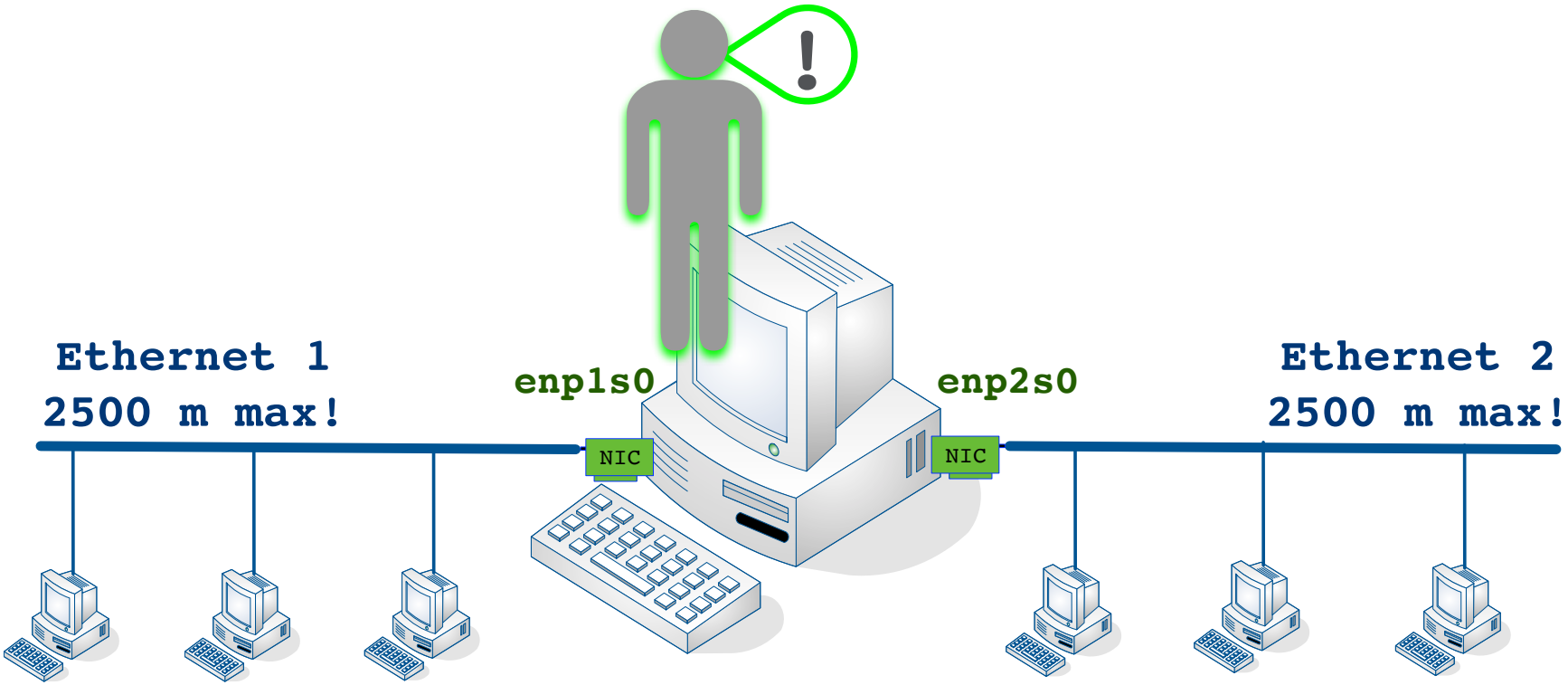
- **A) Repeater in between them?**
  - It might exceed the physical limitation of the Ethernet
- **B) Hubs? Hub simply regenerate electrical signals**
  - Same limitations as repeaters.
  - Hubs are Physical-layer devices
- **C) Bridge?** Networking equipment that forwards **frames** between segments
  - Bridges/switches are Datalink-layer devices



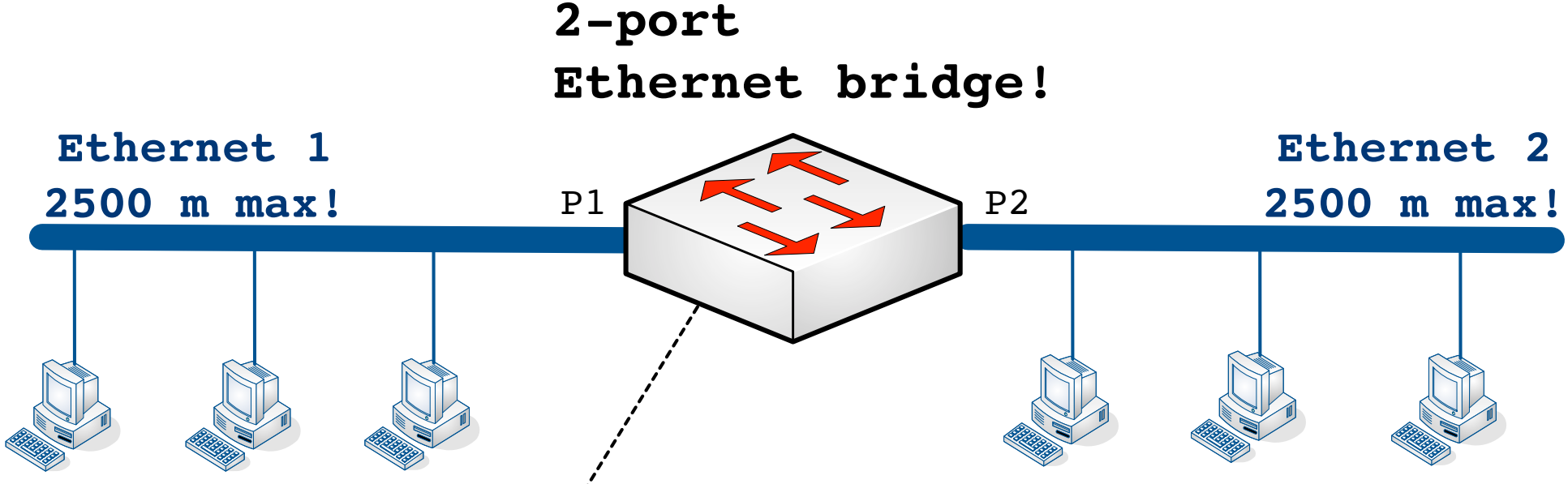
Connecting two max Ethernets

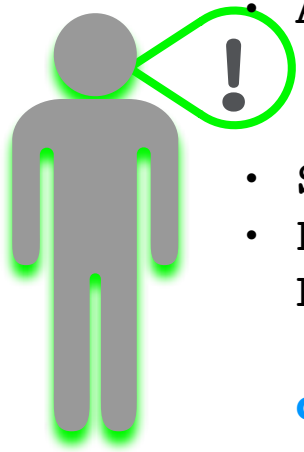
Bridge service (Software) communicates
Ethernets as though they were one!

```
doCopy(receivedFrame, destinationPort);
```



Connecting two Ethernets: Bridge





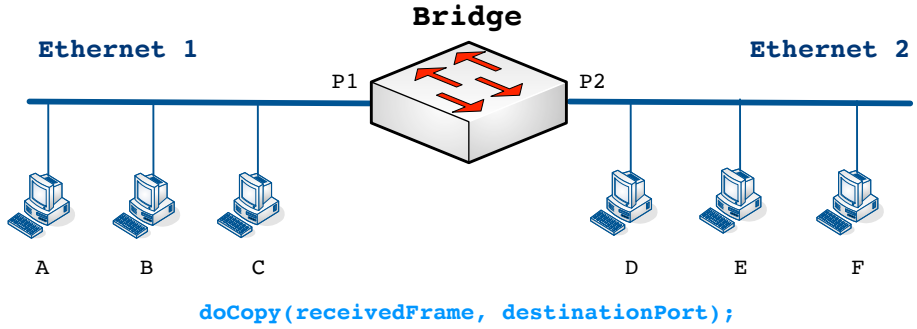
- Electrically separates Ethernets
- At least, 2 ports:
  - NIC on port 1
  - NIC on port 2
- Store/Forward device = S/F
- Bridge service (Software) communicates Ethernets as though they were one!

```
doCopy(receivedFrame, destinationPort);
```

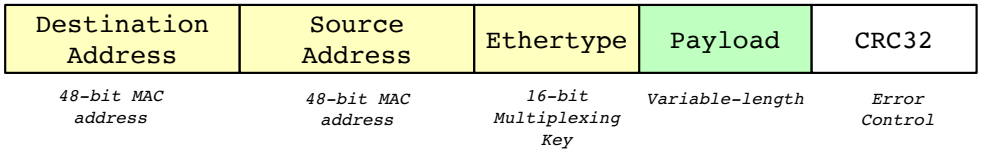
Bridges

- The *no-frills* bridge (simplest, oldest, not used today)
  - Each frame received on a port is forwarded to all its other ports
  - Not used today

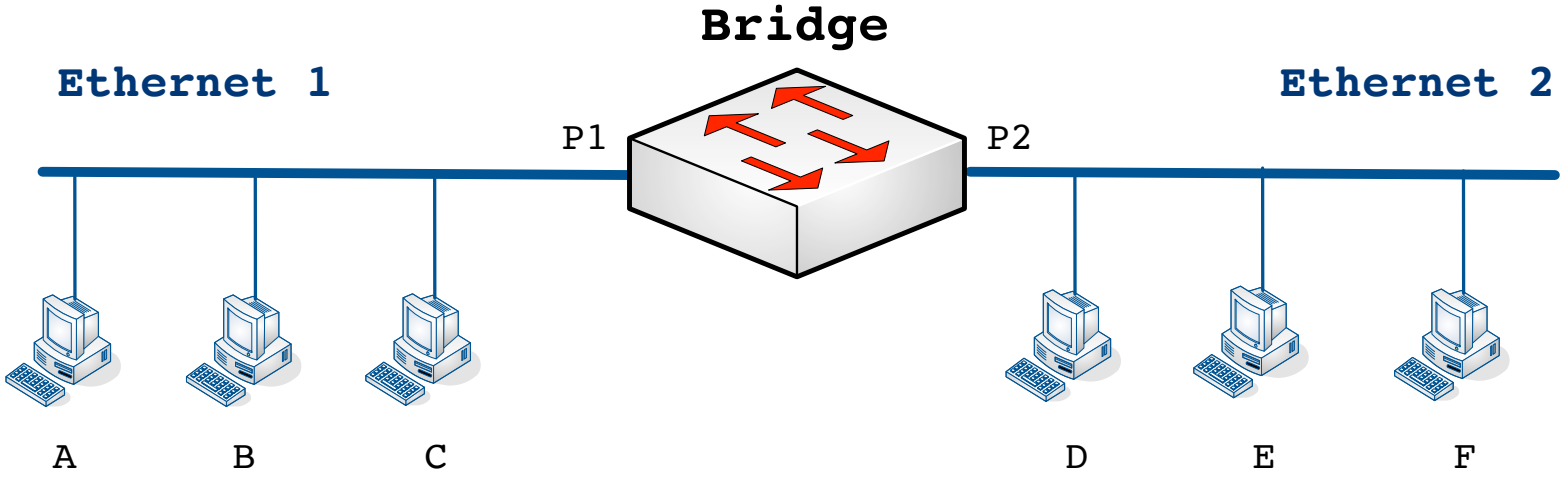
- Learning Bridge
  - Learn MAC addresses as nodes send traffic
  - Have a Station cache or Forwarding Table
    - It contains a MAC – Port table
    - Station sends a frame onto the network for the first time
    - Switch records its source MAC and the port number it was received onto



Bridge algorithm



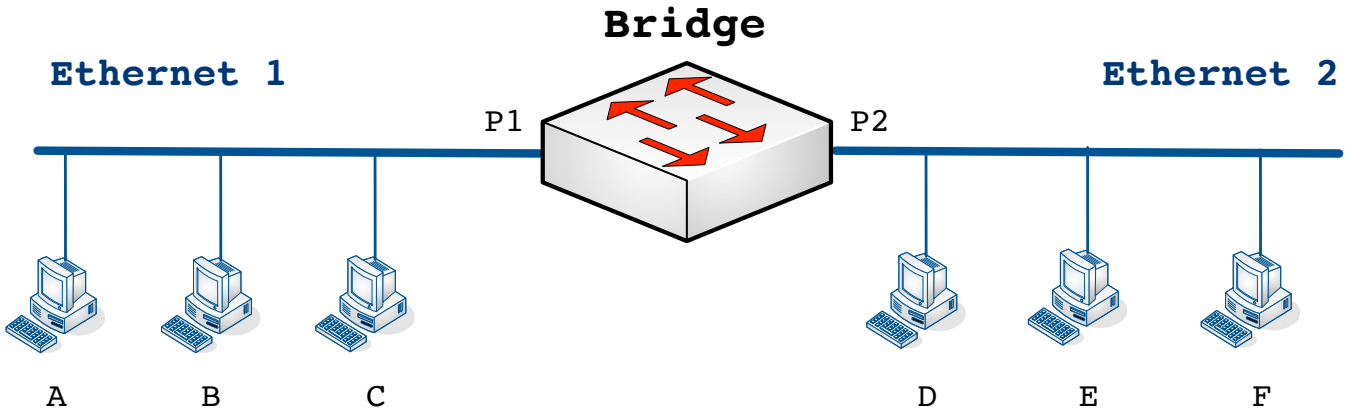
1. Receive a frame on a port and store it into the incoming frame buffer
2. Consult forwarding table
   1. Record the source MAC address into de forwarding table
   2. If destination MAC belongs to the another port, send it onto that port when possible
   3. If destination MAC belongs to receiving port, do nothing
   4. If destination MAC has not been recorded into the forwarding table yet, flood the frame (Send it onto all ports except the one it was received onto)




```
doCopy(receivedFrame, destinationPort);
```

Learning and forwarding examples

- Learning on which **port** each host resides?
  - *Download a table into the bridge ☺ NO!* (Too much maintenance)
  - Record new source MAC A into the Forwarding Table when host A sends its first frame

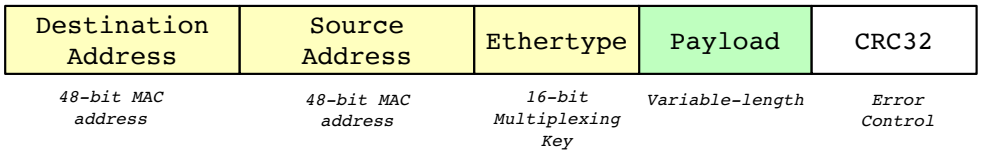



```
doCopy(receivedFrame, destinationPort);
```

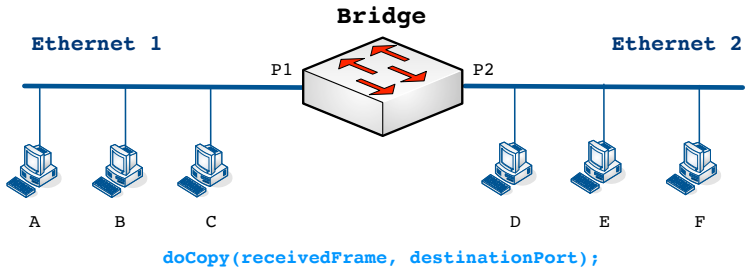
Forwarding table

Host MAC	Port
A	P1
B	P1
C	P1
D	P2
E	P2
F	P2

Learning and forwarding examples



- Can the bridge *learn* this information by itself?
  - Yes: this is the *learning bridge*



- Here's how:
  - A bridge inspects the source MAC address in every Ethernet frame it receives
  - Record that information into the forwarding table (FT)
  - When a bridge first boots, this table is **empty**
  - Entries are added over time as hosts inject frames into their ports
    - A timeout is associated with each entry (aging)
    - The bridge discards the entry after a specified period of time
    - It server to protect against the situation in which a host is moved from one network to another
  - If the bridge receives a frame that is addressed to a host not currently in the table
    - Send the frame onto all *other* ports (Not on the one it was received on): flooding

Forwarding Table

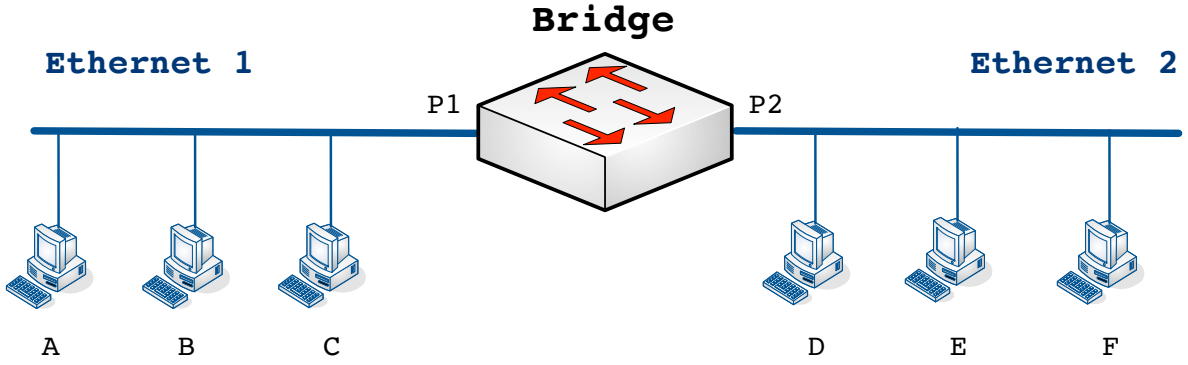
Host MAC	Port
A	P1
B	P1
C	P1
D	P2
E	P2
F	P2

Extended LAN domains

- Can A collide with B? Yes, it can since A and B are connected to the *same* Ethernet segment
- Can A collide with X? No, since A and X belong to different Ethernet segments

- There exist TWO segments or collision domains
  - A, B, C and bridge port P1
  - D, E, F and bridge port P2

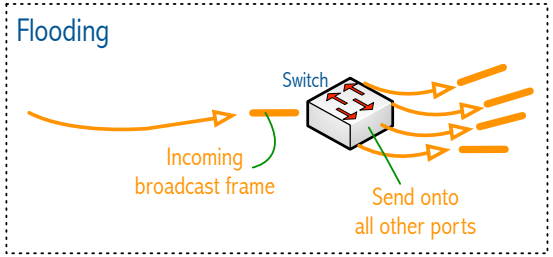
- HOWEVER, there is only one Extended LAN (Network)
  - When a broadcast frame is sent, it is received by all network hosts, we say that it contains a single BROADCAST DOMAIN



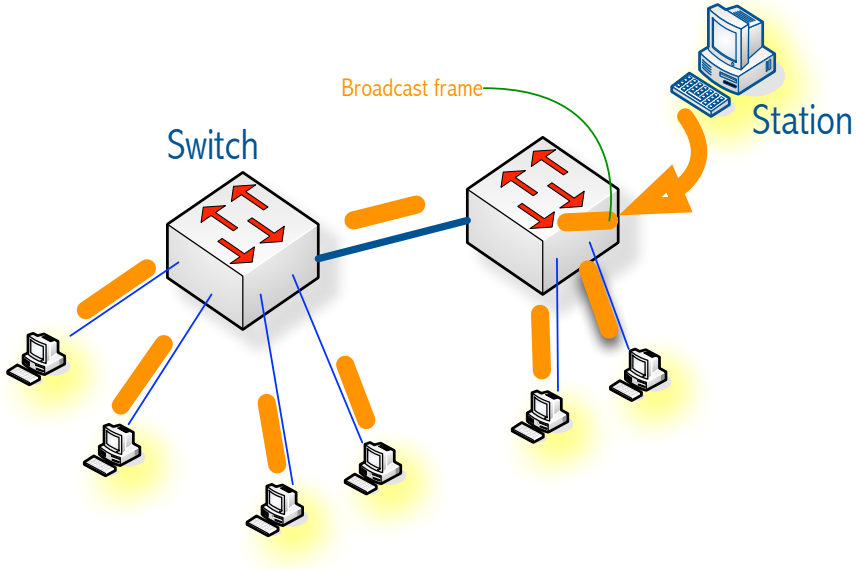

```
doCopy(receivedFrame, destinationPort);
```

Ethernet vs. Switched Ethernet

- Shared Ethernet:
  - Inherently BROADCAST
  - Every frame is delivered to all hosts, inevitably
  - Half-duplex
  - Only one flow active at a time
    - Bus topology and Star topology (hub)

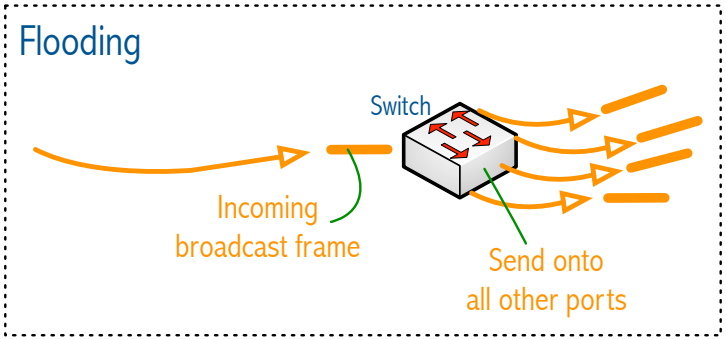


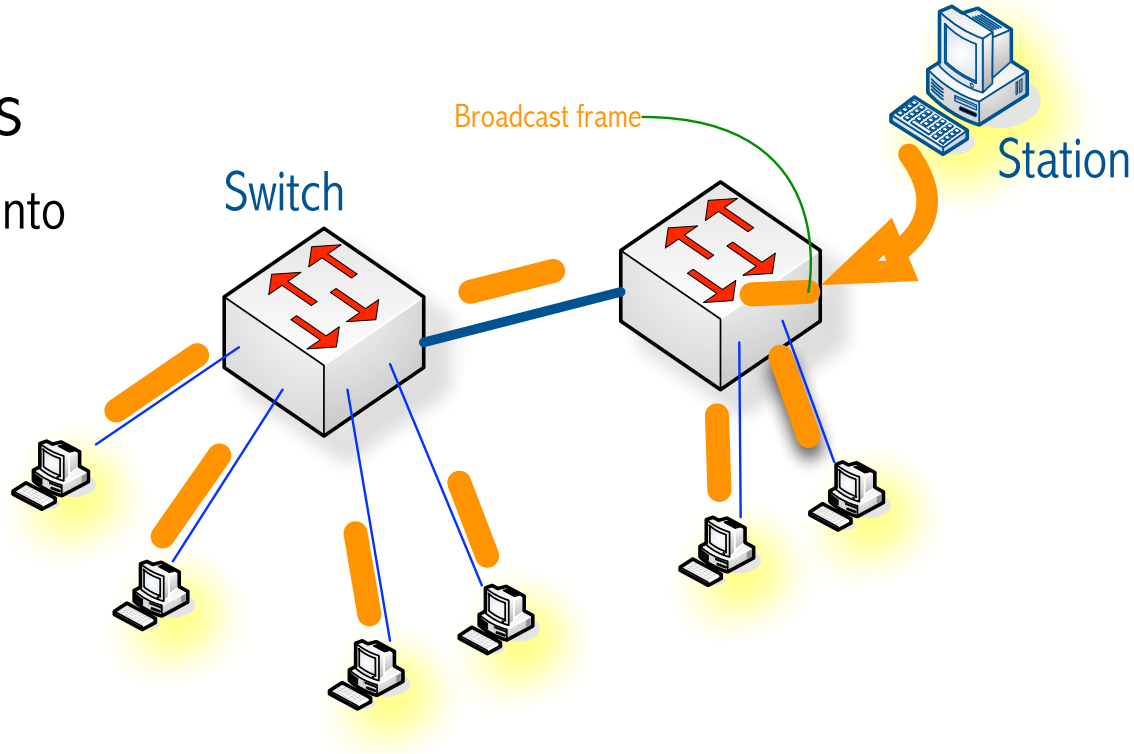
- Switched Ethernet:
  - An Extended LAN based on the interconnection of LAN segments by using bridges and switches
  - BROADCAST is possible but not inherent to the technology, how?
  - Full-duplex
  - Several simultaneous communication paths (Flows) active
    - Star topology, only



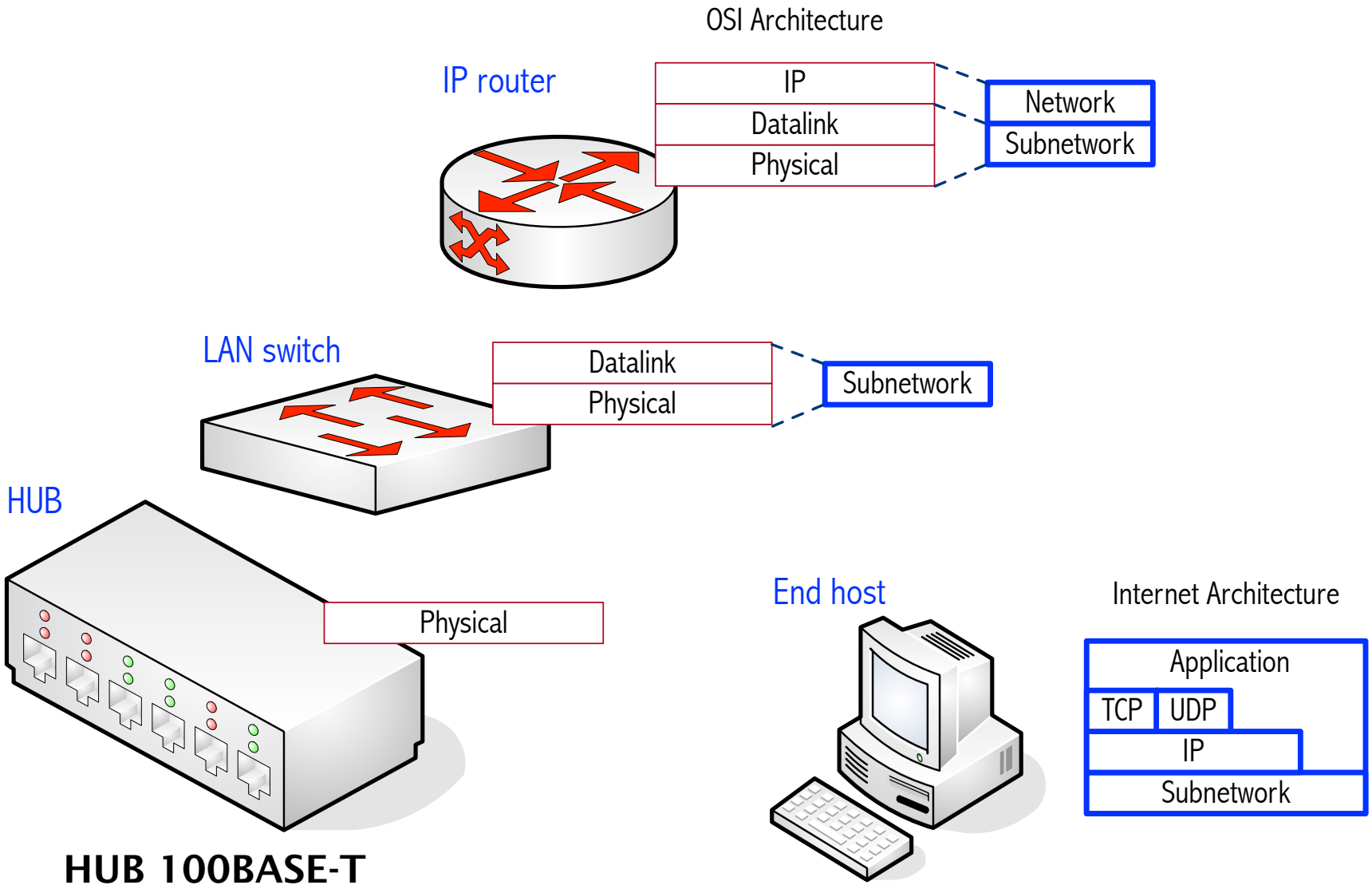
Switches do support broadcast

- As usual, a frame can be addressed to the **broadcast** address
- The switch will forward a broadcast frame to all ports
  - Except the port it was received onto
  - Known as **Flooding**





Devices and Network Architecture



The end