Network

Fundamentals of Ethernet LAN

So Ethernet?

Ethernet is everywhere

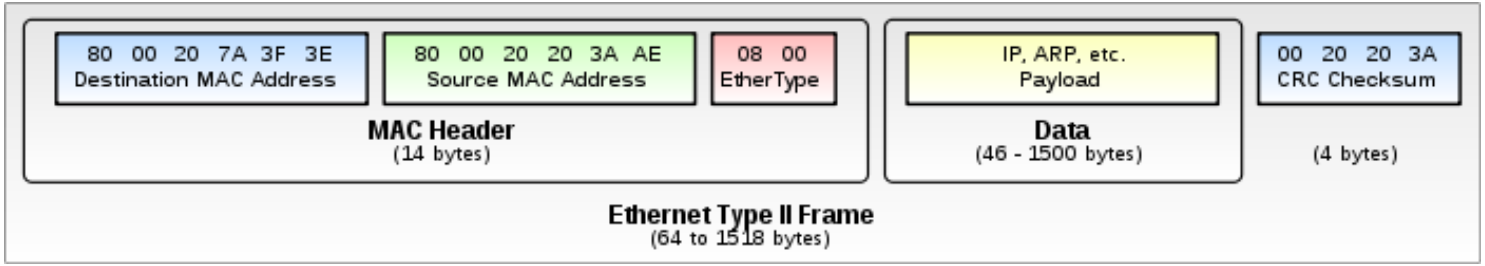
Packet data transmission = small blocks (Ethernet)

Addressing in every packet(frame)
MTU(1500kbytes) in Ethernet

Addressing in Ethernet and IP

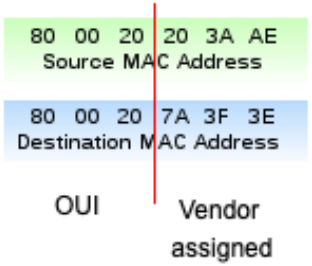
Ethernet		IP		DATA
dmac:00:0c:ce:13:b9:a0	smac:00:22:5f:98:91	sip:192.168.0.89	dip:192.168.0.1	

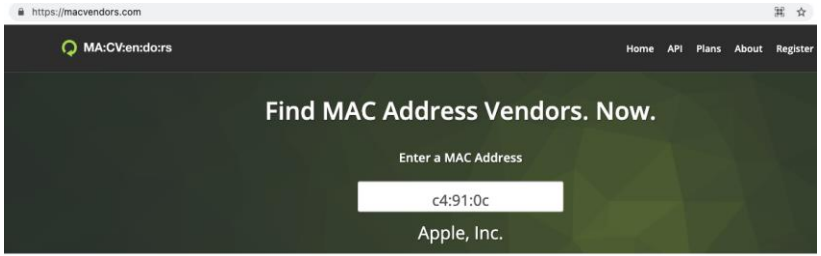
Ethernet frame



Next slide is about Ethertype
Payload = Data

Lets find a mac-address






```
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        options=400<CHANNEL_IO>
        ether c4:91:0c:b2:e4:31
        inet 192.168.1.65 netmask 0xffffff00 broadcast 192.168.1.255
        inet6 fe80::46f:541f:b95c:fb03%en0 prefixlen 64 secured scopeid 0xc
        nd6 options=201<PERFORMNUD,DAD>
        media: autoselect
        status: active
```

Ethernet frame

https://en.wikipedia.org/wiki/EtherType

Values [edit]

EtherType values for some notable protocols[8]

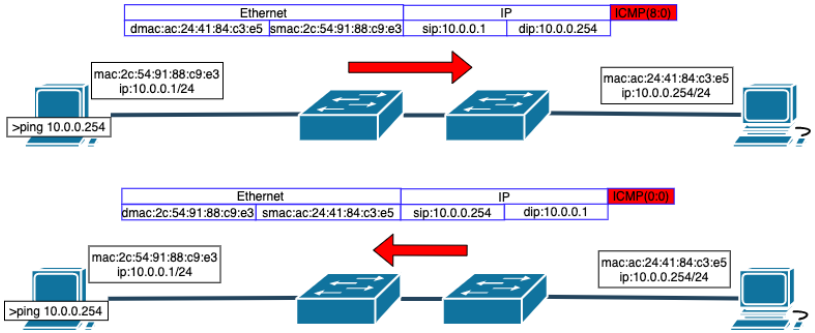
EtherType (hexadecimal)	Protocol
0x0800	Internet Protocol version 4 (IPv4)
0x0806	Address Resolution Protocol (ARP)
0x0842	Wake-on-LAN[9]
0x22F0	Audio Video Transport Protocol (AVTP)
0x22F3	IETF TRILL Protocol
0x22EA	Stream Reservation Protocol
0x6002	DEC MOP RC
0x6003	DECnet Phase IV, DNA Routing
0x6004	DEC LAT
0x8035	Reverse Address Resolution Protocol (RARP)
0x809B	AppleTalk (Ethertalk)
0x80F3	AppleTalk Address Resolution Protocol (AARP)
0x8100	VLAN-tagged frame (IEEE 802.1Q) and Shortest Path Bridging IEEE 802.1aq with NNI compatibility[10]
0x8102	Simple Loop Prevention Protocol (SLPP)
0x8103	Virtual Link Aggregation Control Protocol (VLACP)
0x8137	IPX
0x8204	QNX Qnet
0x86DD	Internet Protocol Version 6 (IPv6)
0x8808	Ethernet flow control
0x8809	Ethernet Slow Protocols[11] such as the Link Aggregation Control Protocol (LACP)
0x8819	CobraNet
0x8847	MPLS unicast
0x8848	MPLS multicast
0x8863	PPPoE Discovery Stage
0x8864	PPPoE Session Stage
0x887B	HomePlug 1.0 MME

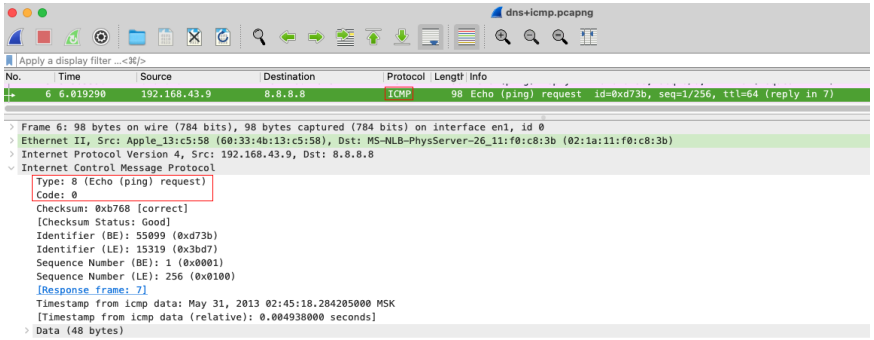
https://en.wikipedia.org/wiki/EtherType

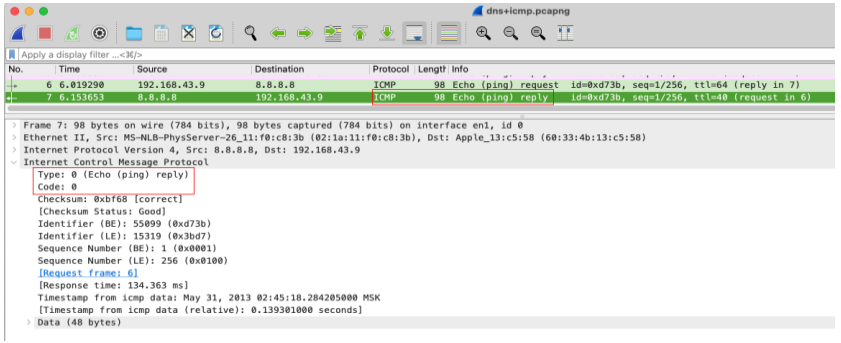
0x8819	CobraNet
0x8847	MPLS unicast
0x8848	MPLS multicast
0x8863	PPPoE Discovery Stage
0x8864	PPPoE Session Stage
0x887B	HomePlug 1.0 MME
0x888E	EAP over LAN (IEEE 802.1X)
0x8892	PROFINET Protocol
0x889A	HyperSCSI (SCSI over Ethernet)
0x88A2	ATA over Ethernet
0x88A4	EtherCAT Protocol
0x88A8	Service VLAN tag identifier (S-Tag) on Q-in-Q tunnel.
0x88AB	Ethernet Powerlink[citation needed]
0x88B8	GOOSE (Generic Object Oriented Substation event)
0x88B9	GSE (Generic Substation Events) Management Services
0x88BA	SV (Sampled Value Transmission)
0x88BF	MikroTik RoMON (unofficial)
0x88CC	Link Layer Discovery Protocol (LLDP)
0x88CD	SERCOS III
0x88E3	Media Redundancy Protocol (IEC62439-2)
0x88E5	IEEE 802.1AE MAC security (MACsec)
0x88E7	Provider Backbone Bridges (PBB) (IEEE 802.1ah)
0x88F7	Precision Time Protocol (PTP) over IEEE 802.3 Ethernet
0x88F8	NC-SI
0x88FB	Parallel Redundancy Protocol (PRP)
0x8902	IEEE 802.1ag Connectivity Fault Management (CFM) Protocol / ITU-T Recommendation Y.1731 (OAM)
0x8906	Fibre Channel over Ethernet (FCoE)
0x8914	FCoE Initialization Protocol
0x8915	RDMA over Converged Ethernet (RoCE)
0x891D	TTEthernet Protocol Control Frame (TTE)
0x893a	1905.1 IEEE Protocol
0x892F	High-availability Seamless Redundancy (HSR)
0x9000	Ethernet Configuration Testing Protocol[12]
0xF1C1	Redundancy Tag (IEEE 802.1CB Frame Replication and Elimination for Reliability)

Ping adventure. Network is working

Devices are working for a long time







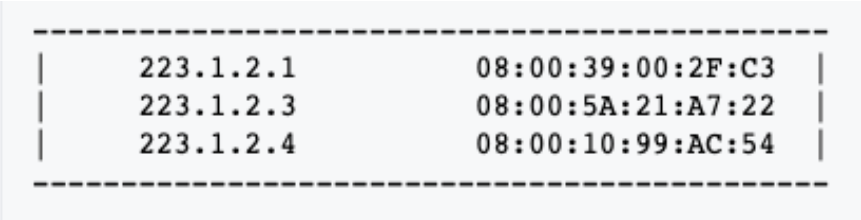
1)We can’t send ICMP without IP header and Ethernet header
2)But how do we get Ethernet mac-addresses?

ARP

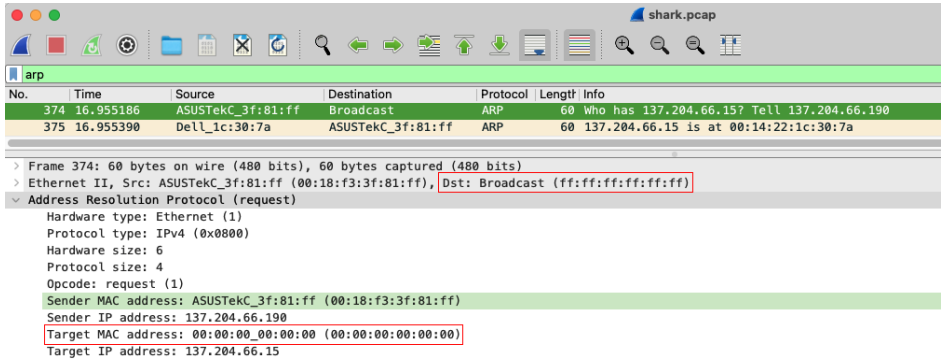
ARP its a protocol for getting DMAC from DIP

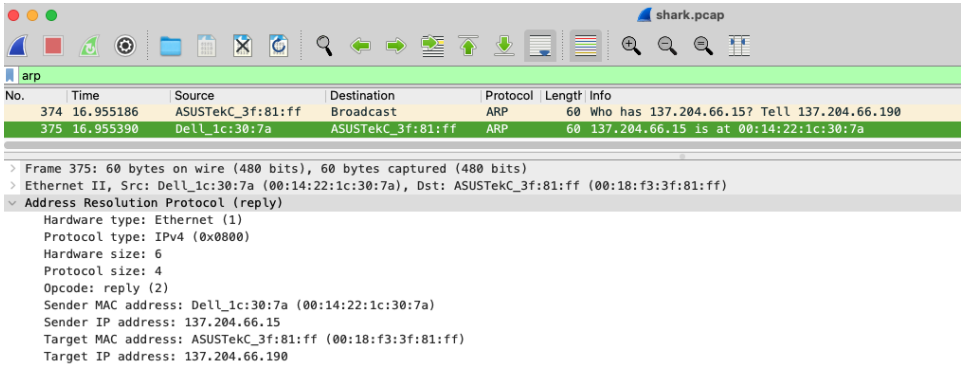
Arp -a (windows, linux command)

Every router, PC/server, or any what has got an IP address should fill ethernet headers to send info via NIC. So its should have ARP table
ARP entries could be static, dynamic, dynamic entries have timers (default 4hours)



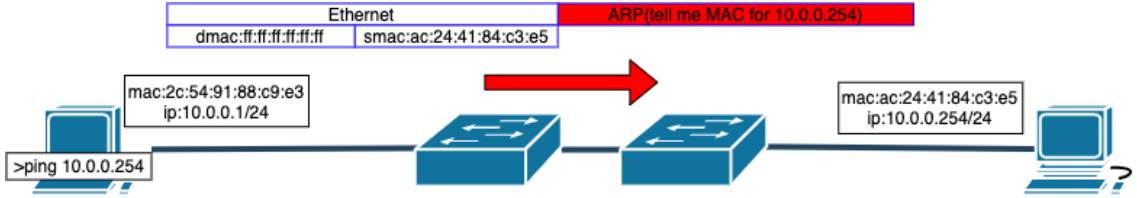
ARP wireshark

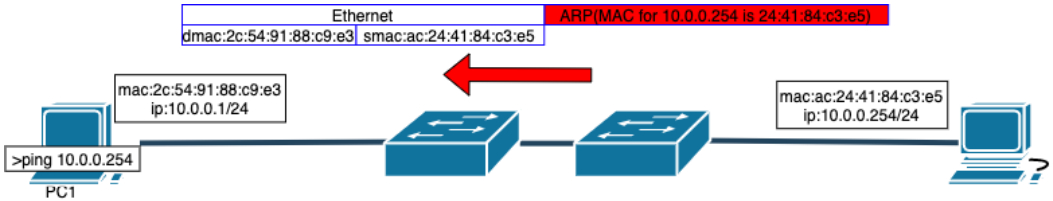




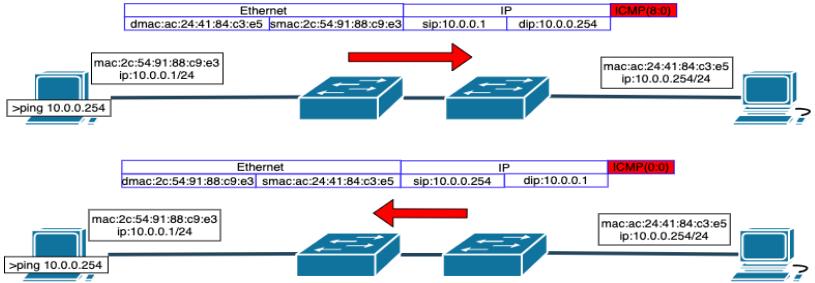
ARPs don’t have an IP header
They can’t be routed.
They can’t be via a Router.

Ping adventure 2. PC were rebooted





PC1:
Arp -a
10.0.0.0.254 24:41:84:c3:e5



From ‘ping adventure 1’ slide

Ethernet could be different

Speed/Standard
10mb
100mb
1gb
10gb
40gb
100gb

100meters?

Copper/Fiber

Half-duplex vs Full-duplex

Full-duplex faster then half-duplex
In Full-duplex we don’t have collisions (in Half-duplex we have collusions)

Collisions in half duplex

Hi, how are you doing? (same time) Hi, nice to see you!

What did you say? Sorry, what?

Half-duplex vs Full-duplex

Full-duplex

Hi, how are you doing? (same time) Hi, nice to see you!

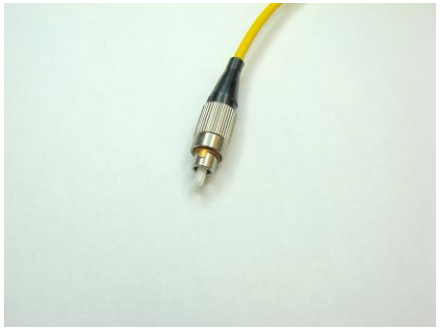
Hey, hey, I’m doing well. I’m okay.

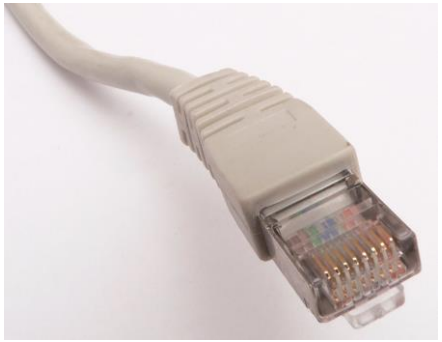
Mouth

Ear

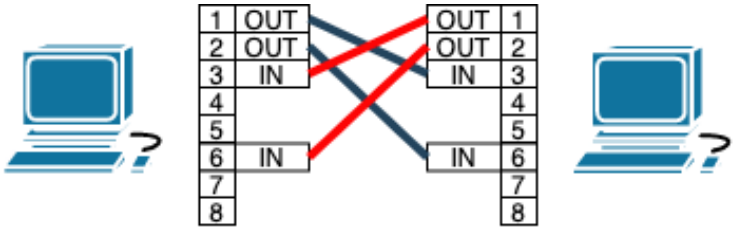
Cables

Crossover vs Straight-through





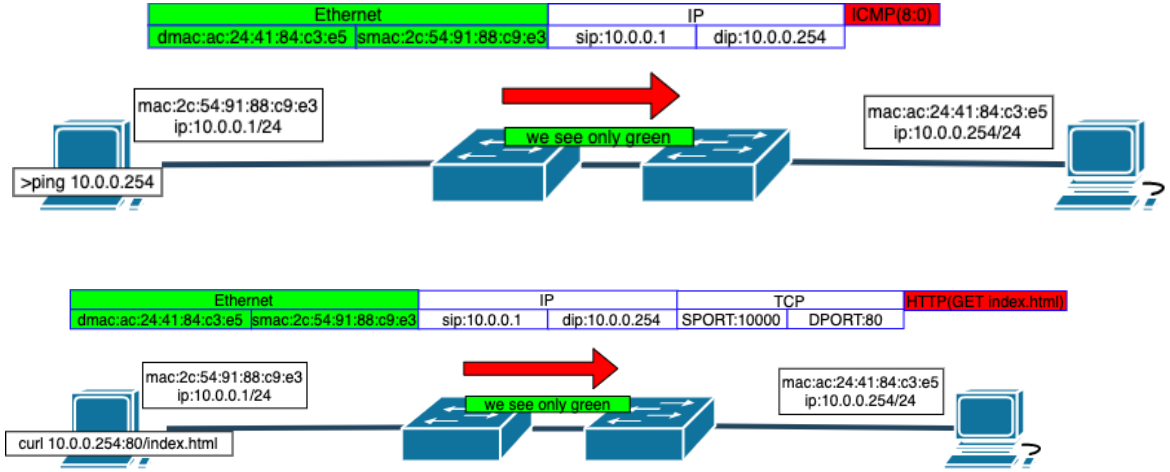
Cables



Crossover (for pc-pc, with hub)

Straight-through (with switches, routers)

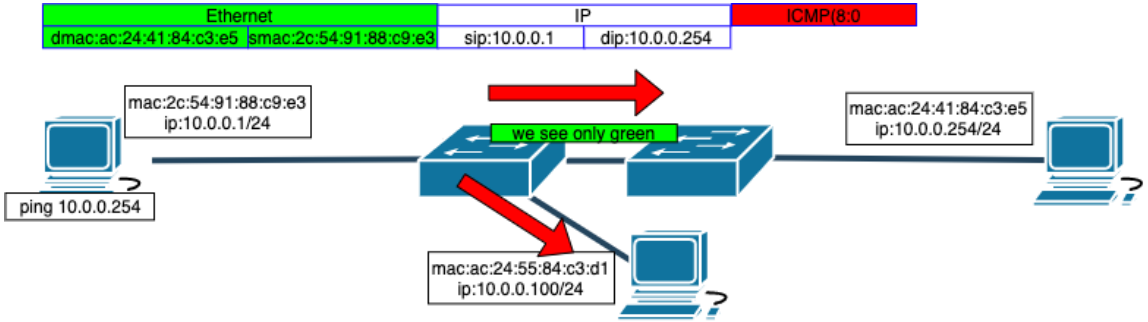
Point of view



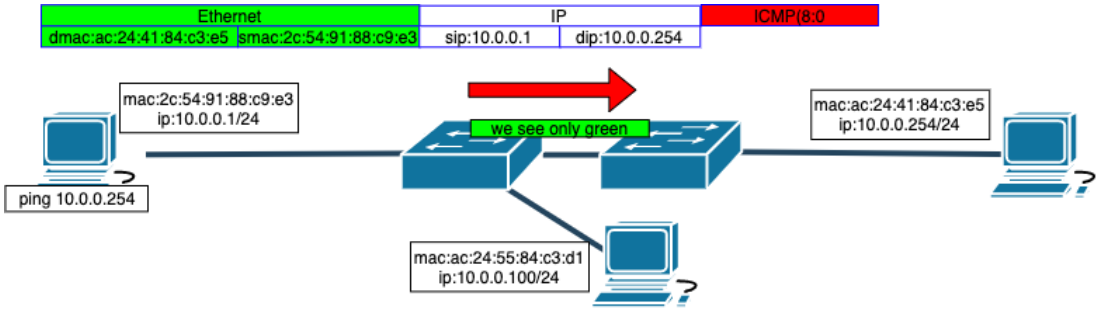
Switches are not smart?
Well, switches, hubs, bridges see only ethernet headers(mac addressing, ethertype)
They are only forwarding Ethernet, they see ONLY ethernet.

Hub, Switches, Bridges

Hubs send to every interface except that interface, a packed has come

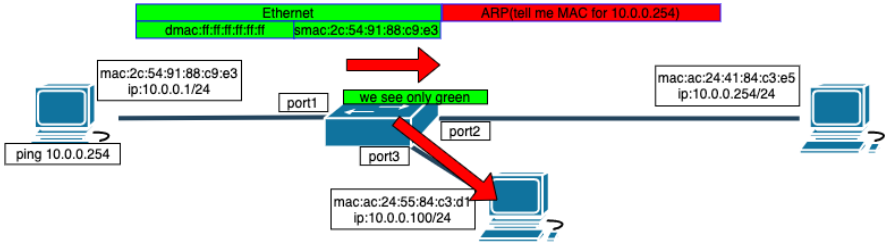


Switches send to interface except that interface. But how do they know? CAM/MAC table

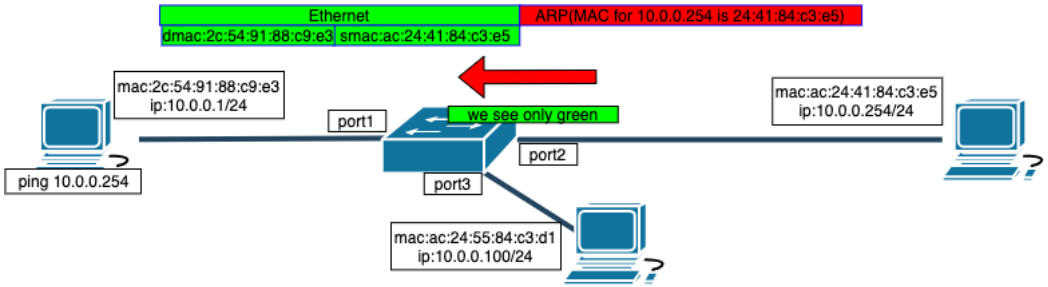


MAC table

ARP



Switch MAC table:
2c:54:91:88:c9:e3 on port1 (new entry)



Switch MAC table:
2c:54:91:88:c9:e3 on port1
ac:24:41:84:c3:e5 on port2 (new entry)

Switches learn SMAC and add them to the MAC table.
Why? To be the switches, not hubs :-)

MAC table

MAC table:
no entries

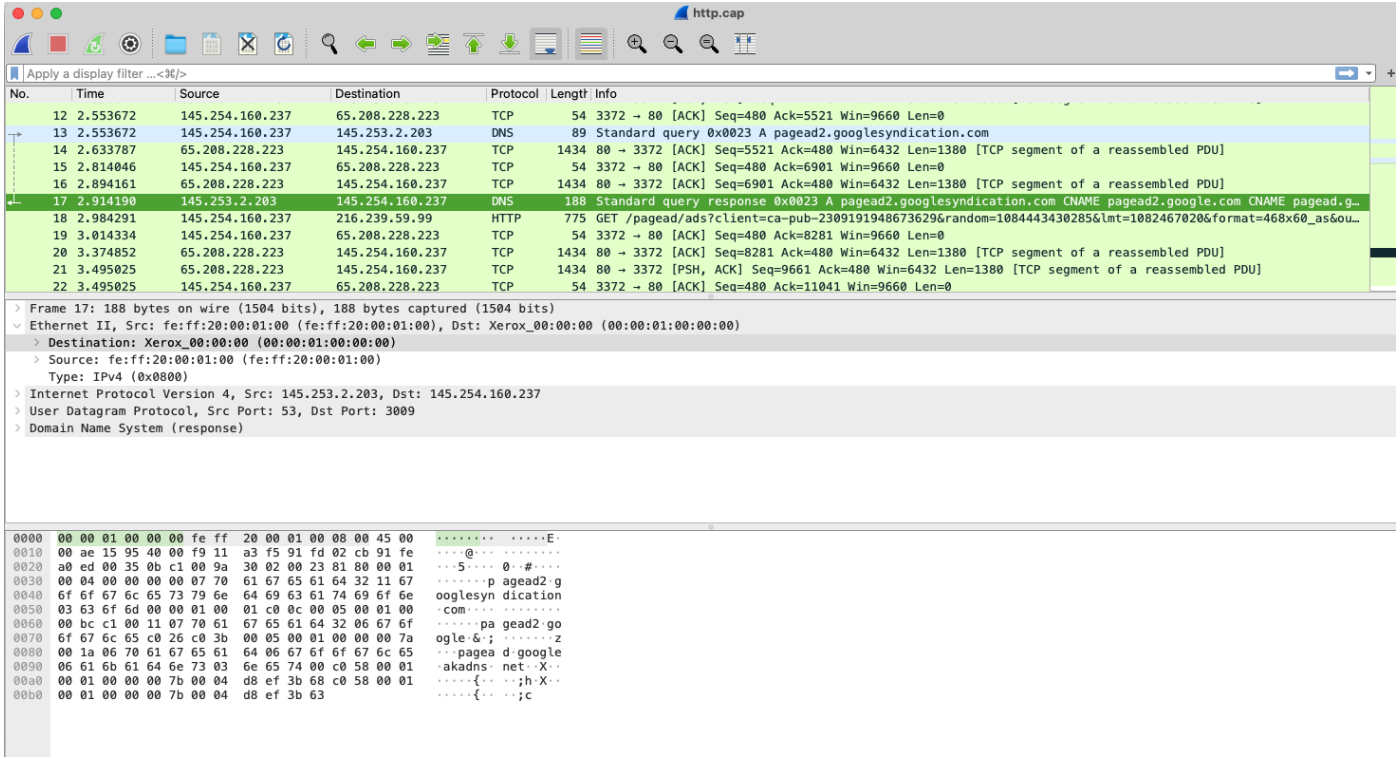
MAC table:
no entries

MAC table:
aa:aa:aa:aa:aa:aa port 2

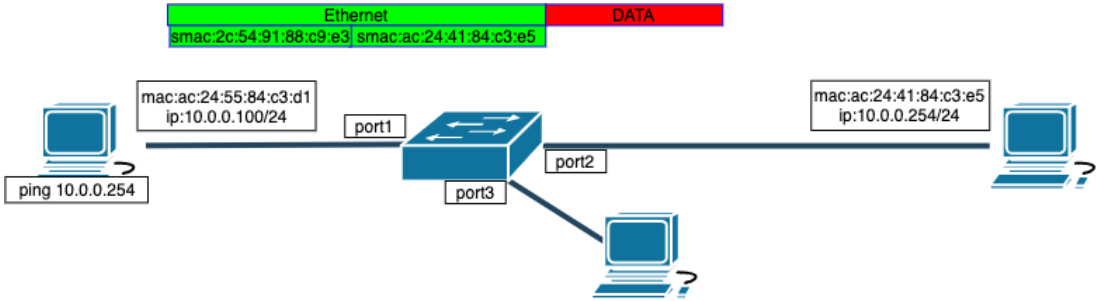
MAC table:
cd:cd:cd:cd:cd:cd port 2
bb:bb:bb:bb:bb:bb port 2
ee:ee:ee:ee:ee:ee port 2
ab:ab:ab:ab:ab:ab port 1
aa:aa:aa:aa:aa:aa port 1

Hub or Switch?
If thing has got MAC table – its a Switch
If thing doesn't have MAC table – its a Hub

Wireshark Ethernet

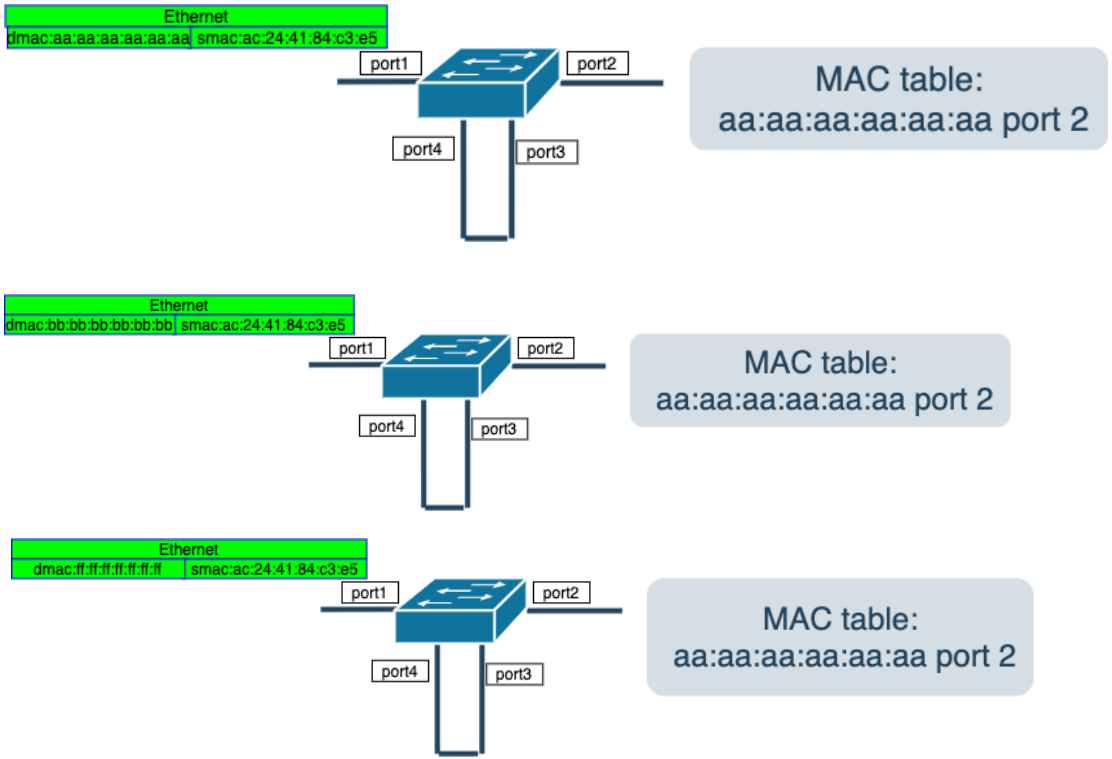


Lets do it again?



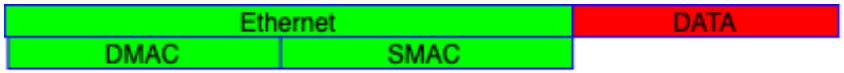
1) We have Ethernet NIC
Any ICMP/HTTP/HTTPS/TCP/UDP will be encapsulated to IP and to Ethernet.
2)PC should fill the ETHERNET headers. PC should know DMAC(uses ARP)
3)Switch uses MAC table to send ETHERNET frames
4)PC has got 1 table: ARP table (matching ip to DMAC). ARP table is showing WHERE
5)Switch has got 1 table: MAC table (matching SMAC to interface). MAC table is showing WHERE and WHERE FROM

Ethernet Loop?



FF:FF:FF:FF:FF:FF for port possible?
NO!

Ethernet addresses types



BROADCAST – where DMAC is FF:FF:FF:FF:FF
MULTICAST – where DMAC is 01:00:5e:xx:xx:x
UNICAST – where DMAC is something else (aa:aa:aa:aa:aa:aa, 1b:2c:3c:11:db:fe etc)
UNKNOWN UNICAST – switch doesn't got THAT unicast DMAC – so send to all ports (except the received port) – like BROADCAST

How to get Ethernet address

```
[stepan@sun /]$ ifconfig
enp0s13f0u1: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether 94:05:bb:14:45:7e  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

```
[stepan@sun /]$ ifconfig
enp0s13f0u1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.78  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::71b7:eae0:bd4e:d798  prefixlen 64  scopeid 0x20<link>
        ether 94:05:bb:14:45:7e  txqueuelen 1000  (Ethernet)
        RX packets 5345  bytes 7064496 (6.7 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 3946  bytes 342580 (334.5 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

THANK YOU