Network Supervision et Troubleshooting

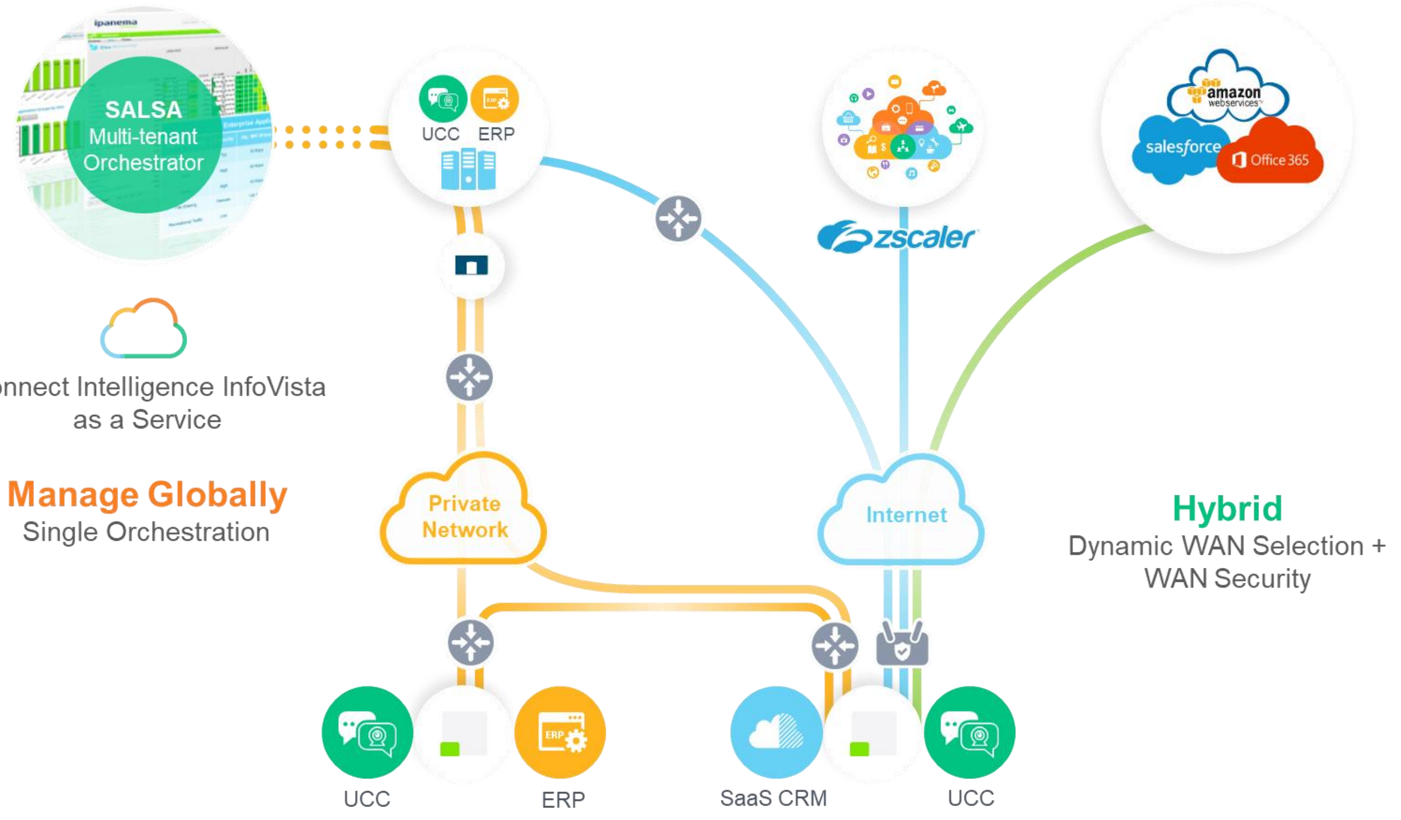
Chapter 1

Networking basics

B. Daheb

About Ipanema SDWAN

- Solution Overview
  - Entreprises
    - Ipanema



ESME Sudria

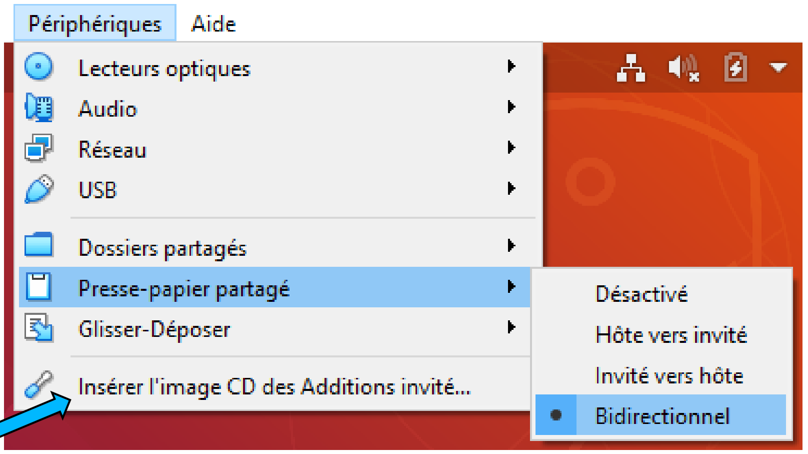
Why this course ?

- "Everything fails all the time"
  - Werner Vogels, Amazon CTO
- How to configure TCP/IP network
- Understand network failures
  - And the impact on applications
- Learn techniques to resolve networks issues

Practical exercises

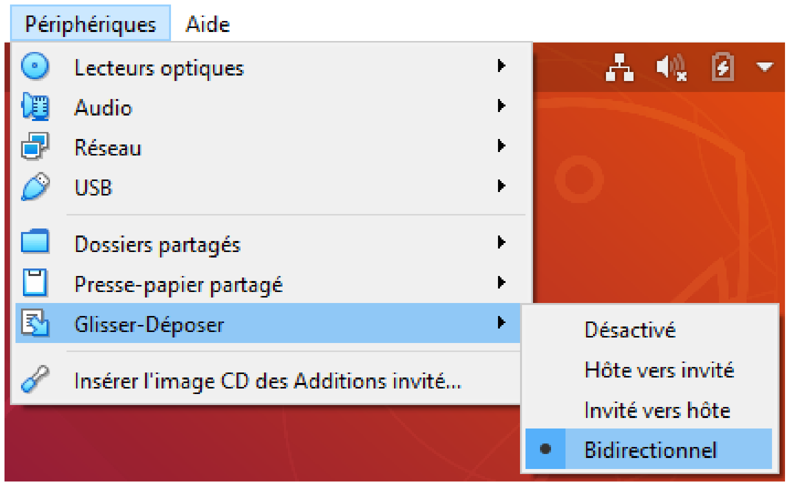
- Linux Ubuntu
  - Install VirtualBox
  - Create a virtual machine with Ubuntu-disk001.vmdk
- Don't forget
  - Install AddOns on Linux
  - Update Ubuntu from terminal
    - sudo apt-get update
    - sudo apt-get upgrade

- Transfer Windows-Ubuntu



Périphériques | Aide

- Lecteurs optiques
- Audio
- Réseau
- USB
- Dossiers partagés
- Presse-papier partagé
  - Désactivé
  - Hôte vers invité
  - Invité vers hôte
  - ● Bidirectionnel
- Glisser-Déposer
- Insérer l'image CD des Additions invité...



Périphériques | Aide

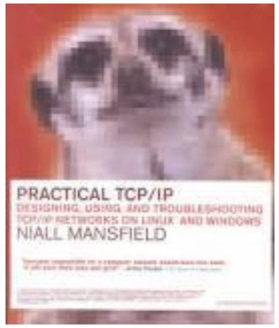
- Lecteurs optiques
- Audio
- Réseau
- USB
- Dossiers partagés
- Presse-papier partagé
- Glisser-Déposer
  - Désactivé
  - Hôte vers invité
  - Invité vers hôte
  - ● Bidirectionnel
- Insérer l'image CD des Additions invité...

Plan

- Network troubleshooting tools
- End-user and system applications
- Connecting to the Internet and security

References

- Network troubleshooting tools, Joseph D. Sloan, O'Reilly Media
- Practical TCP/IP : designing, using, and troubleshooting TCP/IP networks on Linux and Windows, Niall Mansfield, Addison Wesley


PRACTICAL TCP/IP
DESIGNING, USING, AND TROUBLESHOOTING TCP/IP NETWORKS ON LINUX AND WINDOWS
NIALL MANSFIELD

Plan

- Network troubleshooting tools
- End-user and system applications
- Connecting to the Internet and security

Networking basics

- 1 A quick introduction to TCP/IP
- 2 The tcpdump packet sniffer
- 3 How packets move on the local wire
- 4 Basic routing
- 5 IP addressing and netmasks
- 6 Routing in practice
- 7 The DNS Troubleshooting

Networking basics

What is TCP/IP?

- TCP/IP is a set of protocols developed to allow cooperating computers to share resources across a network
- TCP stands for “Transmission Control Protocol”

- IP stands for “Internet Protocol”

- They are Transport layer and Network layer protocols respectively of the protocol suite
- The most well known network that adopted TCP/IP is Internet –
  - the biggest WAN in the world

TCP/IP Model

- Because TCP/IP was developed earlier than the OSI 7-layer mode,
- It does not have 7 layers but only 4 layers

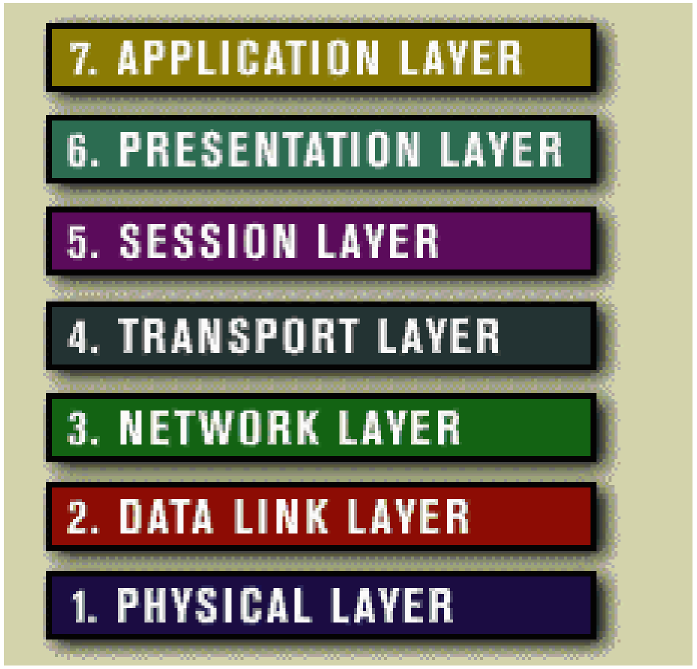
TCP/IP Protocol Suite **OSI 7-layer**

FTP, SMTP, Telnet,

TCP

IP

Ethernet, ATM



TCP/IP Model

- Application layer protocols define the rules when implementing specific network applications

- Rely on the underlying layers to provide accurate and efficient data delivery

- Typical protocols:
    - FTP – File Transfer Protocol
        - For file transfer
    - Telnet – Remote terminal protocol
        - For remote login on any other computer on the network
    - SMTP – Simple Mail Transfer Protocol
        - For mail transfer
    - HTTP – Hypertext Transfer Protocol
        - For Web browsing

TCP/IP Model

- TCP/IP is built on “connectionless” technology
    - each datagram finds its own way to its destination

- Transport Layer protocols define the rules of
    - Dividing a chunk of data into segments
    - Reassemble segments into the original chunk

- Typical protocols:
    - TCP – Transmission Control Protocol
        - Provide the functions such as reordering and data resend
    - UDP – User Datagram Service
        - Use when the message to be sent fit exactly into a datagram
        - Use also when a more simplified data format is required

TCP/IP Model

- Network layer protocols define the rules
  - of how to find the routes for a packet to the destination

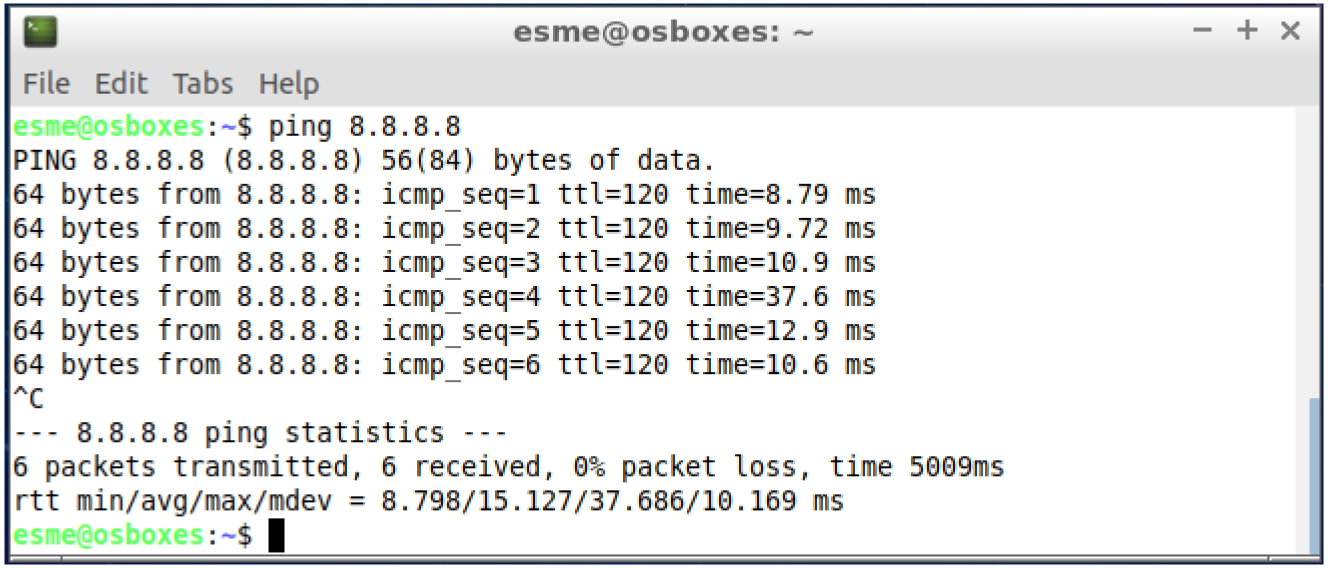
- It only gives best effort delivery
  - Packets can be delayed, corrupted, lost, duplicated, out-of-order

- Typical protocols:
  - IP – Internet Protocol
    - Provide packet delivery
  - ARP – Address Resolution Protocol
    - Define the procedures of network address / MAC address translation
  - ICMP – Internet Control Message Protocol
    - Define the procedures of error message transfer

Ping and traceroute

- Ping
  - measure the time for a packet to travel to a remote host and back
  - The server sends back an acknowledgment when the packet arrives
- Traceroute
  - list the router hops between the client host and a remote host
  - The IP address and domain name (if there is one) of each router is returned to the client

Ping results

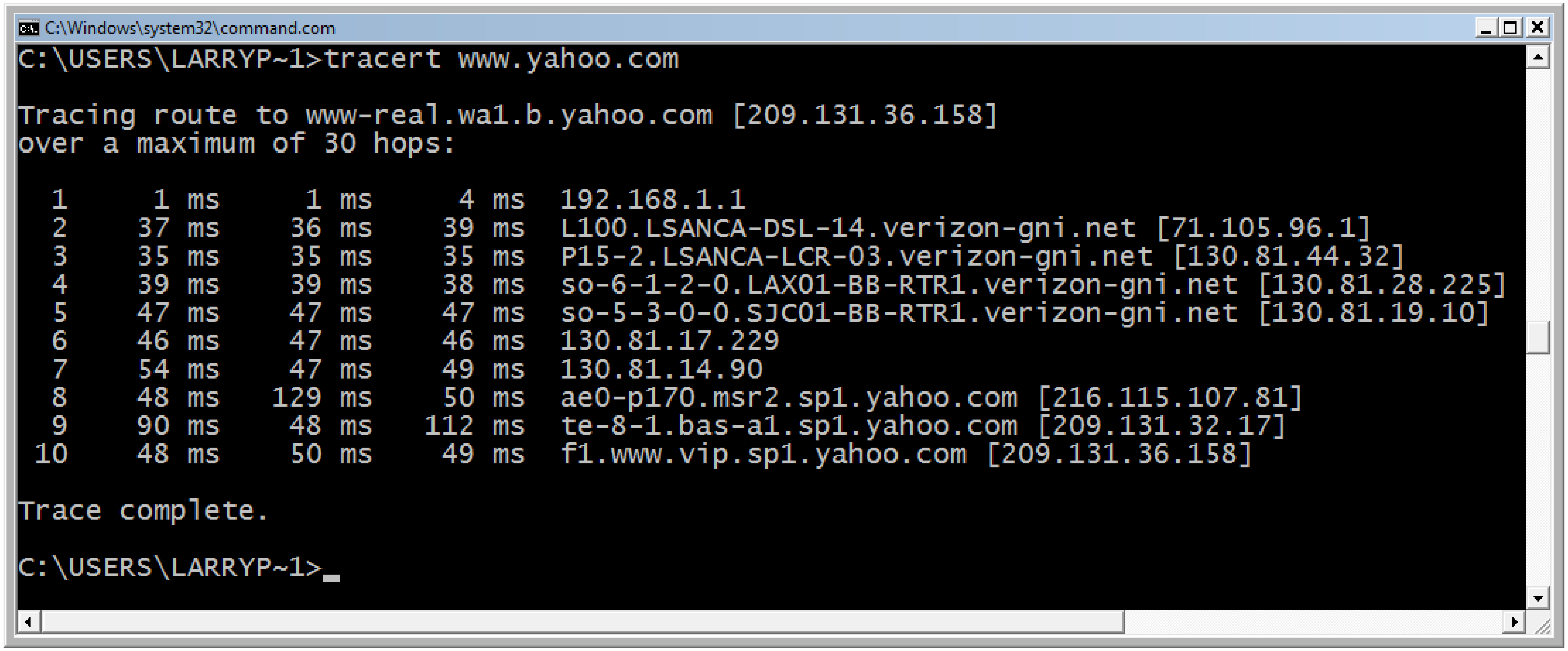


1
2
3
4

```
esme@osboxes:~$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=120 time=8.79 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=120 time=9.72 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=120 time=10.9 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=120 time=37.6 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=120 time=12.9 ms
64 bytes from 8.8.8.8: icmp_seq=6 ttl=120 time=10.6 ms
^C
--- 8.8.8.8 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5009ms
rtt min/avg/max/mdev = 8.798/15.127/37.686/10.169 ms
esme@osboxes:~$
```

1. The command to ping four times
2. The replies took between 8 and 37 milliseconds
3. No packets were lost
4. The average ping time was 15 milliseconds

Traceroute results



```
C:\USERS\LARRYP~1>tracert www.yahoo.com

Tracing route to www-real.wa1.b.yahoo.com [209.131.36.158]
over a maximum of 30 hops:

  1     1 ms     1 ms     4 ms  192.168.1.1
  2    37 ms    36 ms    39 ms  L100.LSANCA-DSL-14.verizon-gni.net [71.105.96.1]
  3    35 ms    35 ms    35 ms  P15-2.LSANCA-LCR-03.verizon-gni.net [130.81.44.32]
  4    39 ms    39 ms    38 ms  so-6-1-2-0.LAX01-BB-RTR1.verizon-gni.net [130.81.28.225]
  5    47 ms    47 ms    47 ms  so-5-3-0-0.SJC01-BB-RTR1.verizon-gni.net [130.81.19.10]
  6    46 ms    47 ms    46 ms  130.81.17.229
  7    54 ms    47 ms    49 ms  130.81.14.90
  8    48 ms   129 ms    50 ms  ae0-p170.msr2.sp1.yahoo.com [216.115.107.81]
  9    90 ms    48 ms   112 ms  te-8-1.bas-a1.sp1.yahoo.com [209.131.32.17]
 10    48 ms    50 ms    49 ms  f1.www.vip.sp1.yahoo.com [209.131.36.158]

Trace complete.

C:\USERS\LARRYP~1>_
```

Hop 1: my home LAN router
Hops 2-7: Verizon (my ISP) network
Hops 8-10: the Yahoo LAN

Networking basics

Main Tools

- tcpdump / windump
  - Great, simple capture tool
  - Standard format
- Tcptrace
  - Capture file analysis: information on each connection
    - Elapsed time, bytes sent and received, round trip times, etc.
    - graphs for further analysis.
- Wireshark
  - Great GUI capture tool

TCPDump / Windump

- Low level packet sniffer.

–Good

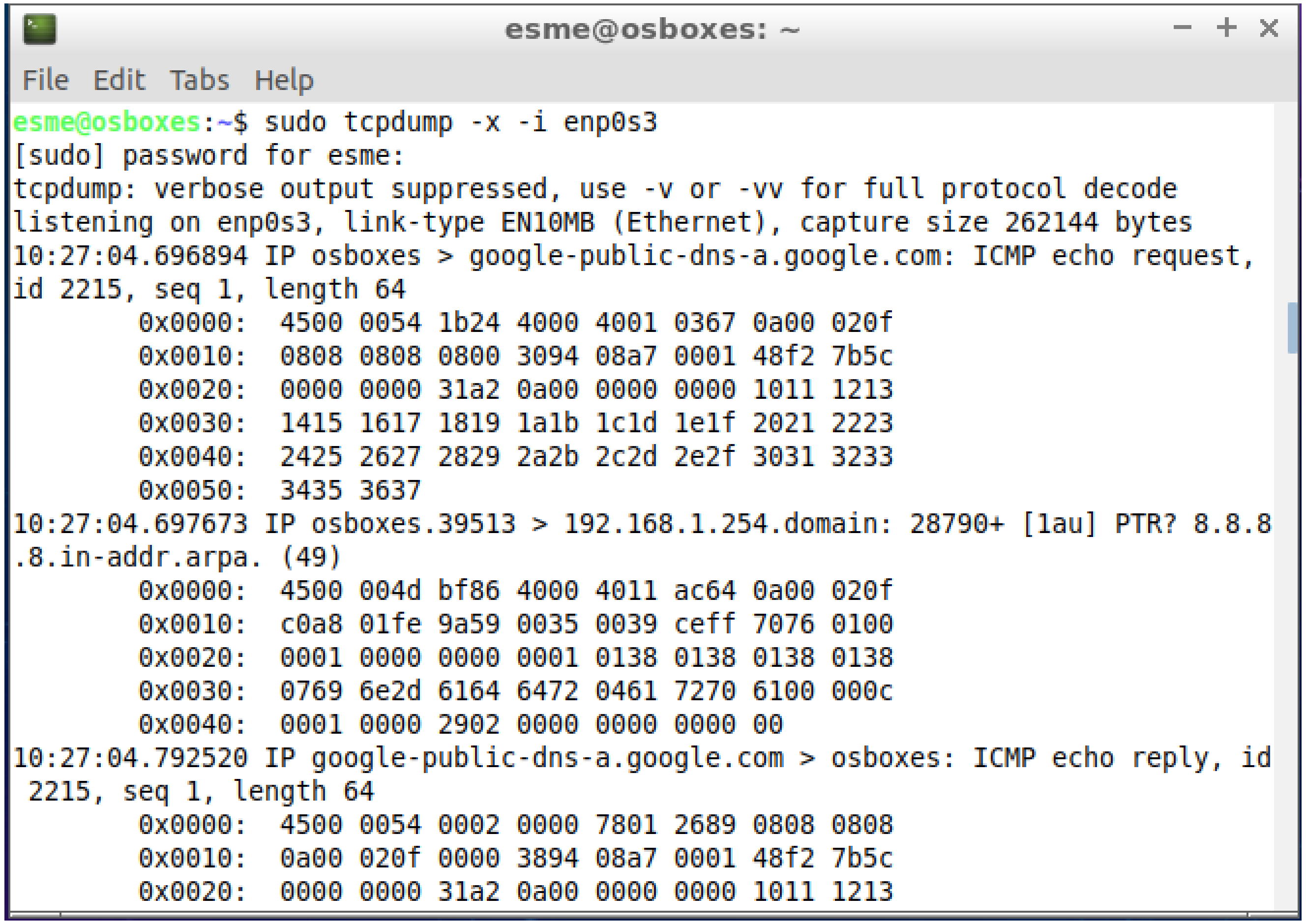
- –see a new type of attack
- –try to diagnose a networking problem
- –Universally available and cheap

–Bad

- –have to look at all packets
- –and interpret them
- –does not scale well
- –state / connections are hidden

Running TCPDump

- tcpdump –x
  - looks at packets in hex format

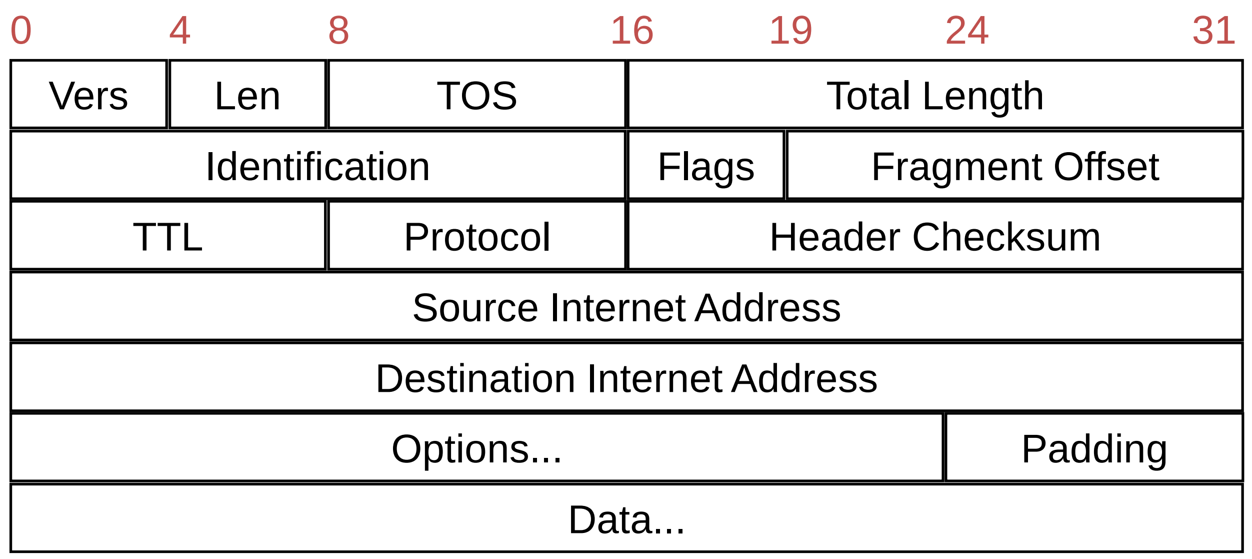


```
esme@osboxes:~$ sudo tcpdump -x -i enp0s3
[sudo] password for esme:
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp0s3, link-type EN10MB (Ethernet), capture size 262144 bytes
10:27:04.696894 IP osboxes > google-public-dns-a.google.com: ICMP echo request,
id 2215, seq 1, length 64
        0x0000:  4500 0054 1b24 4000 4001 0367 0a00 020f
        0x0010:  0808 0808 0800 3094 08a7 0001 48f2 7b5c
        0x0020:  0000 0000 31a2 0a00 0000 0000 1011 1213
        0x0030:  1415 1617 1819 1a1b 1c1d 1e1f 2021 2223
        0x0040:  2425 2627 2829 2a2b 2c2d 2e2f 3031 3233
        0x0050:  3435 3637
10:27:04.697673 IP osboxes.39513 > 192.168.1.254.domain: 28790+ [1au] PTR? 8.8.8
.8.in-addr.arpa. (49)
        0x0000:  4500 004d bf86 4000 4011 ac64 0a00 020f
        0x0010:  c0a8 01fe 9a59 0035 0039 ceff 7076 0100
        0x0020:  0001 0000 0000 0001 0138 0138 0138 0138
        0x0030:  0769 6e2d 6164 6472 0461 7270 6100 000c
        0x0040:  0001 0000 2902 0000 0000 0000 00
10:27:04.792520 IP google-public-dns-a.google.com > osboxes: ICMP echo reply, id
 2215, seq 1, length 64
        0x0000:  4500 0054 0002 0000 7801 2689 0808 0808
        0x0010:  0a00 020f 0000 3894 08a7 0001 48f2 7b5c
        0x0020:  0000 0000 31a2 0a00 0000 0000 1011 1213
```

Running TCPDump

- Interpret packets in which format?
- Use the TCP/IP and tcpdump reference:
- Internet Protocol: RFC791
  - https://fr.wikipedia.org/wiki/Internet_Protocol
- TCP: RFC793
  - https://fr.wikipedia.org/wiki/Transmission_Control_Protocol

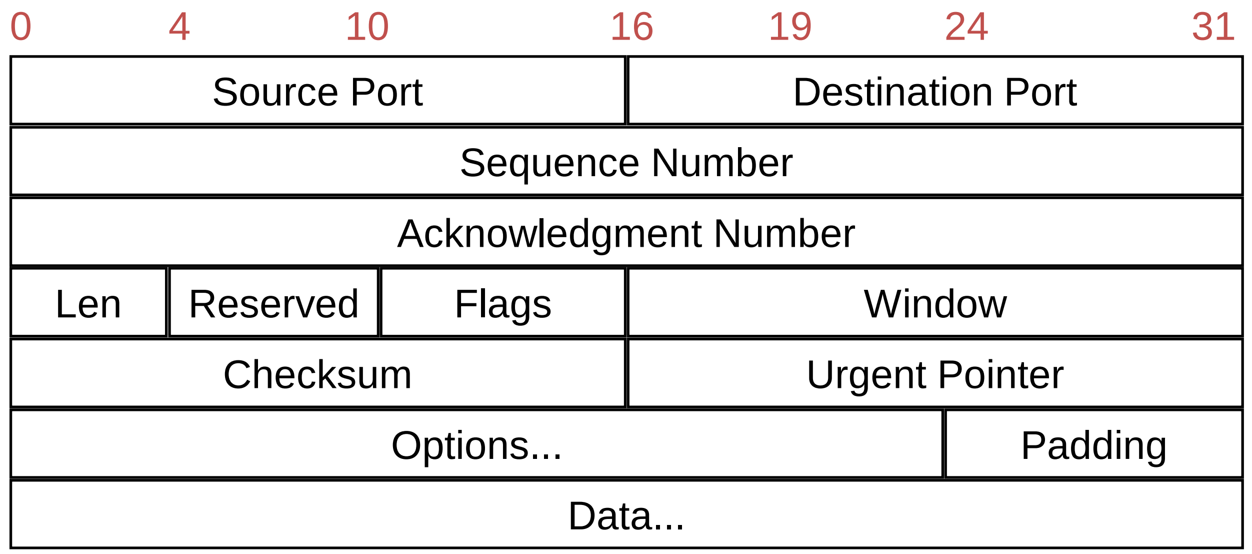
IP Datagram



Field	**Purpose**
Vers	IP version number
Len	Length of IP header (4 octet units)
TOS	Type of Service
T. Length	Length of entire datagram (octets)
Ident.	IP datagram ID (for frag/reassembly)
Flags	Don’t/More fragments
Frag Off	Fragment Offset

Field	**Purpose**
TTL	Time To Live - Max # of hops
Protocol	Higher level protocol (1=ICMP, 6=TCP, 17=UDP)
Checksum	Checksum for the IP header
Source IA	Originator’s Internet Address
Dest. IA	Final Destination Internet Address
Options	Source route, time stamp, etc.
Data...	Higher level protocol data

TCP Segment



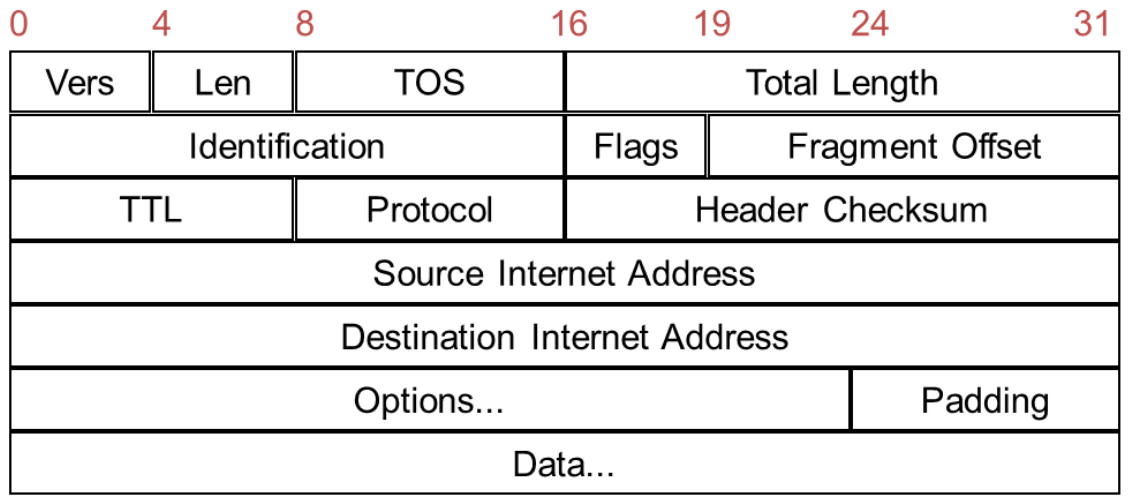
Field	**Purpose**
Source Port	Identifies originating application
Destination Port	Identifies destination application
Sequence Number	Sequence number of first octet in the segment
Acknowledgment #	Sequence number of the next expected octet (if ACK flag set)
Len	Length of TCP header in 4 octet units
Flags	TCP flags: SYN, FIN, RST, PSH, ACK, URG
Window	Number of octets from ACK that sender will accept
Checksum	Checksum of IP pseudo-header + TCP header + data
Urgent Pointer	Pointer to end of “urgent data”
Options	Special TCP options such as MSS and Window Scale

You just need to know port numbers, seq and ack are added

Running tcpdump

IP Header

ICMP Header



`tcpdump –x`

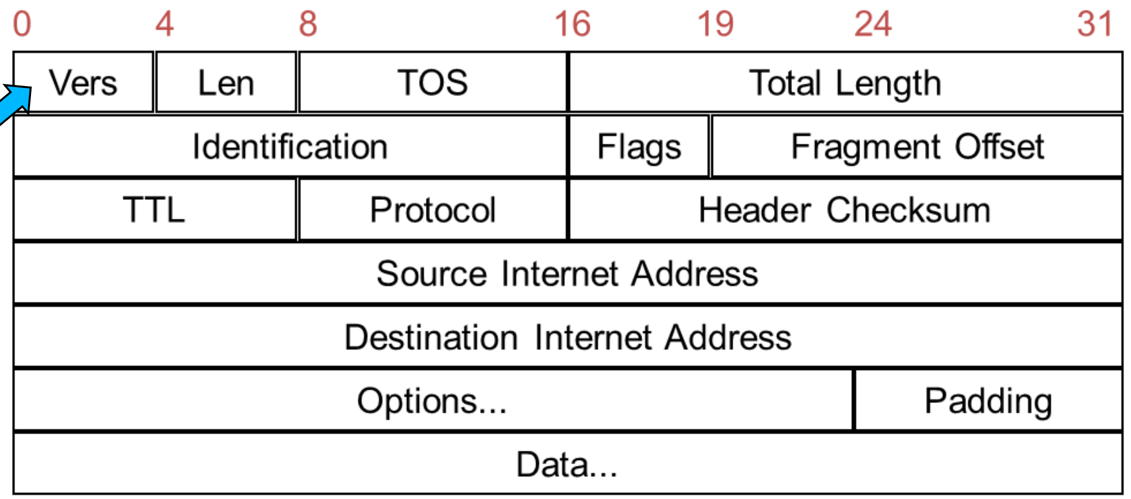
```
20:20:55.778140 IP dhcp-19-211.engr.scu.edu > Bobadilla.scu.edu: icmp 108: echo request
seq 4864
        4500 0080 0231 0000 8001 0d0f 81d2 13d3
        81d2 13c6 0800 d5ee 0200 1300 6162 6364
        6566 6768 696a 6b6c 6d6e 6f70 7172 7374
        7576 7761 6263 6465 6667 6869 6a6b 6c6d
        6e6f 7071 7273 7475 7677 6162 6364 6566
        6768
```

tcpdump

Use TCP/IP reference to identify fields

IP Version 4

Header Length (Nr * 4B)



20:20:55.778140 IP dhcp-19-211.engr.scu.edu > Bobadilla.scu.edu: icmp 108: echo request seq 4864

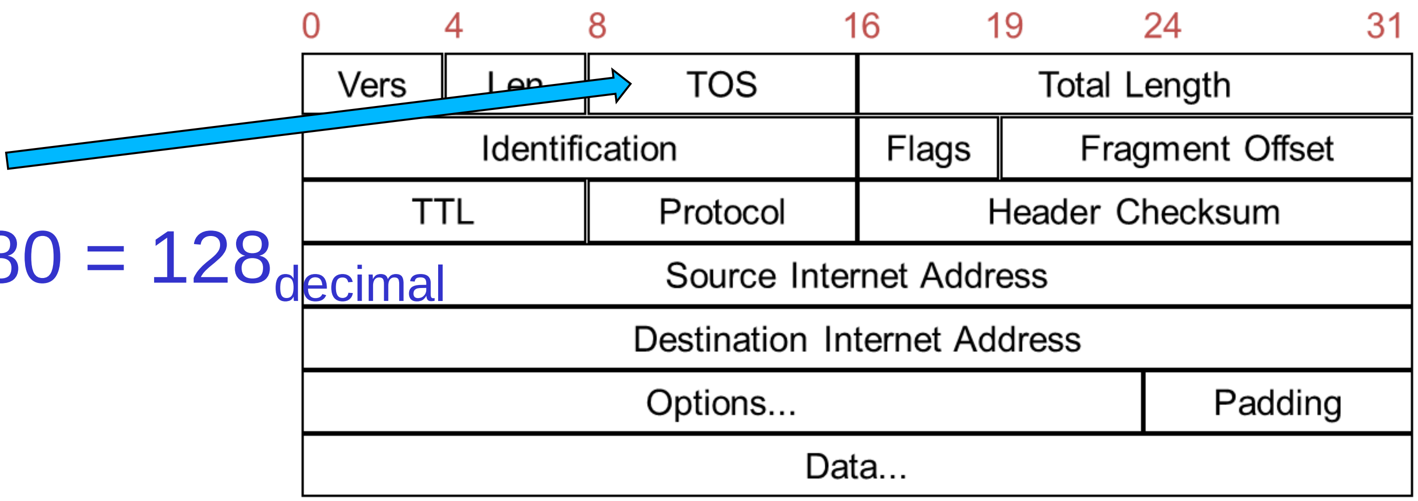
```
4500 0080 0231 0000 8001 0d0f 81d2 13d3
81d2 13c6 0800 d5ee 0200 1300 6162 6364
6566 6768 696a 6b6c 6d6e 6f70 7172 7374
7576 7761 6263 6465 6667 6869 6a6b 6c6d
6e6f 7071 7273 7475 7677 6162 6364 6566
6768
```

tcpdump

20B header

Type of Service

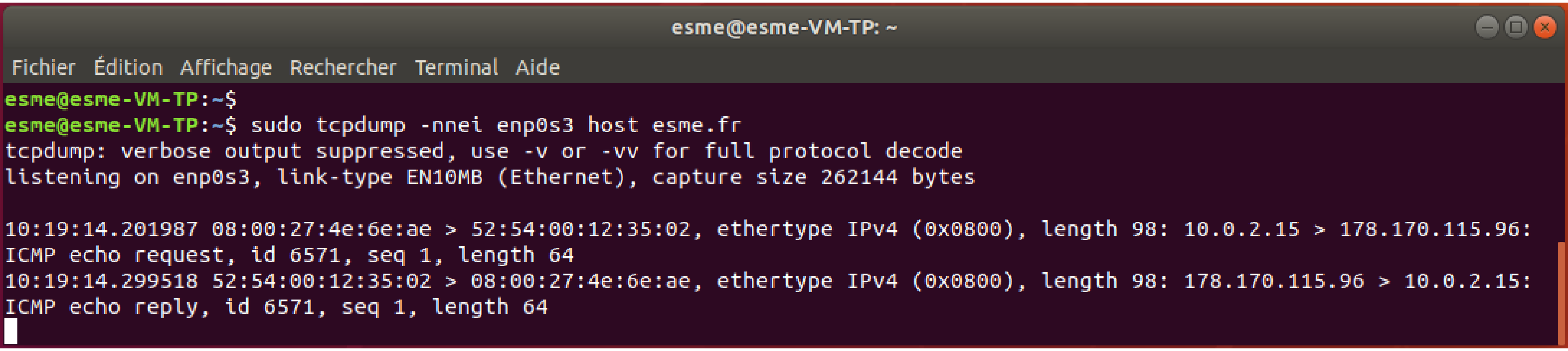
Total Length: 0x80 = 128_{decimal}



```
20:20:55.778140 IP dhcp-19-211.engr.scu.edu > Bobadilla.scu.edu: icmp 108: echo request
seq 4864
                4500 0080 0231 0000 8001 0d0f 81d2 13d3
                81d2 13c6 0800 d5ee 0200 1300 6162 6364
                6566 6768 696a 6b6c 6d6e 6f70 7172 7374
                7576 7761 6263 6465 6667 6869 6a6b 6c6d
                6e6f 7071 7273 7475 7677 6162 6364 6566
                6768
```

tcpdump

- tcpdump –e host bobadilla
  - –Displays data link data filtered by host named bobadilla.
- Shows Source MAC
- Destination MAC
- Protocol



```
esme@esme-VM-TP:~$
esme@esme-VM-TP:~$ sudo tcpdump -nnei enp0s3 host esme.fr
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp0s3, link-type EN10MB (Ethernet), capture size 262144 bytes

10:19:14.201987 08:00:27:4e:6e:ae > 52:54:00:12:35:02, ethertype IPv4 (0x0800), length 98: 10.0.2.15 > 178.170.115.96:
ICMP echo request, id 6571, seq 1, length 64
10:19:14.299518 52:54:00:12:35:02 > 08:00:27:4e:6e:ae, ethertype IPv4 (0x0800), length 98: 178.170.115.96 > 10.0.2.15:
ICMP echo reply, id 6571, seq 1, length 64
```

Tcpdump Fragmentation Total Length

- Total Length: Number of Bytes in Packet

```
20:42:07.217979 IP Bobadilla.scu.edu.137 >
239.255.255.250.137: udp 50
4500 004e 892b 0000 0111 aae1 81d2 13c6
efff fffa 0089 0089 003a adb9 8ce2 0000
0001 0000 0000 0000 2043 4b41 4141 4141
4141 4141 4141 4141 4141 4141 4141 4141
4141 4141 4141 4141 4100 0021 0001
```

Tcpdump
Fragmentation Offset Header

- Length 0x33c = 828 (-20B for header)
- Offset: 1ce8 → 0001 1100 1110 1000 = 7400
  - Leading 000 are flags.
- Multiply by 8: Offset = 59200 bytes

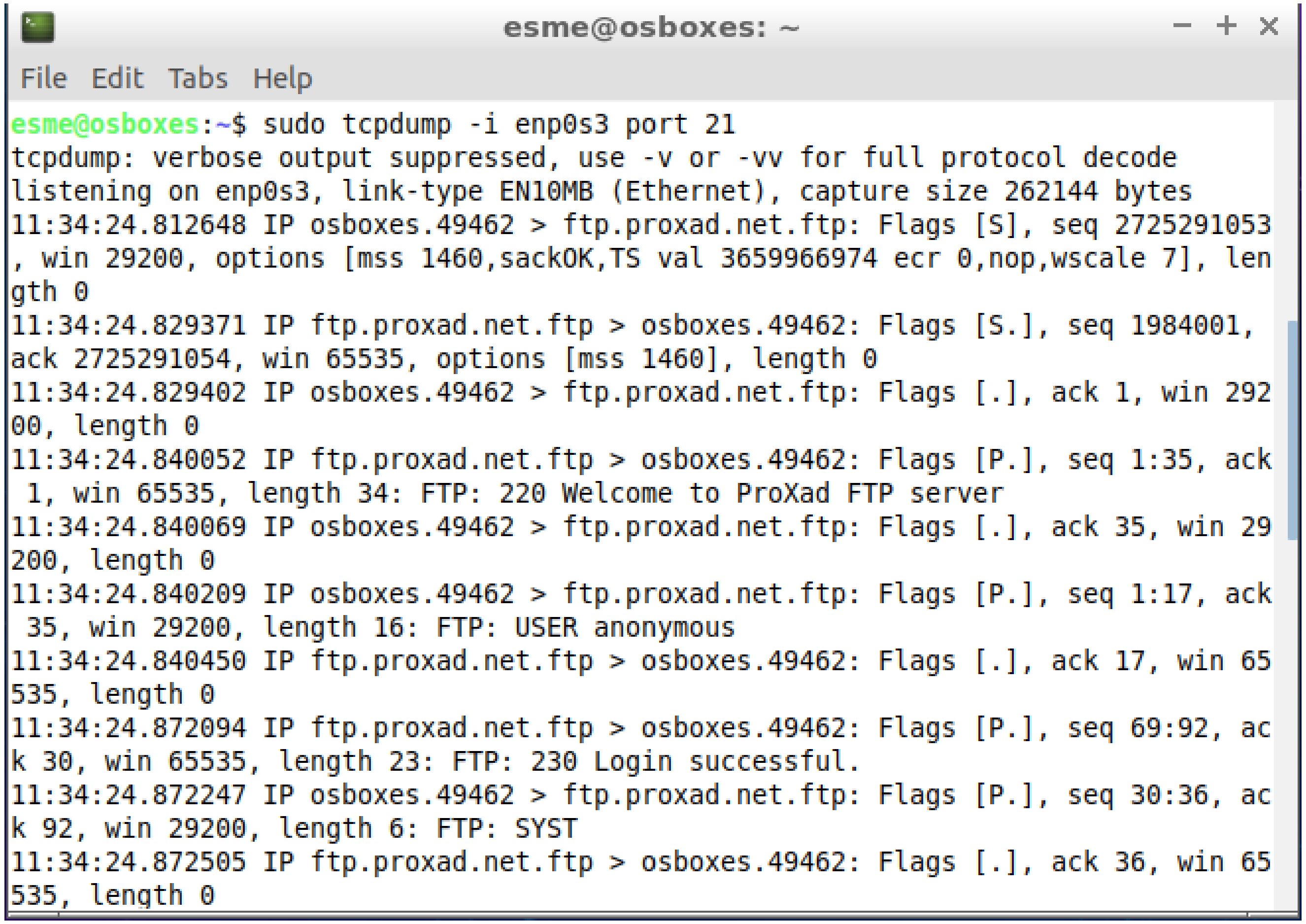
```
20:53:26.443325 IP Bobadilla.scu.edu > dhcp-19-
211.engr.scu.edu: icmp (frag 35188:808@59200)
4500 033c 8974 1ce8 8001 6627 81d2 13c6
81d2 13d3 6e6f 7071 7273 7475 7677 6162
6364 6566 6768 696a 6b6c 6d6e 6f70 7172
7374 7576 7761 6263 6465 6667 6869 6a6b
6c6d 6e6f 7071 7273 7475 7677 6162 6364
6566
```

TCPDump Filters

- Capture only packets that are useful
  - Specify in the filter what items are interesting
  - Filters use common fields such as *host* or *port*
  - Filters also for individual bytes and bits in the datagram

TCPDump Filters

- Format 1: macro and value
- "tcpdump port 21"
  - Only displays packets going to or from port 21.

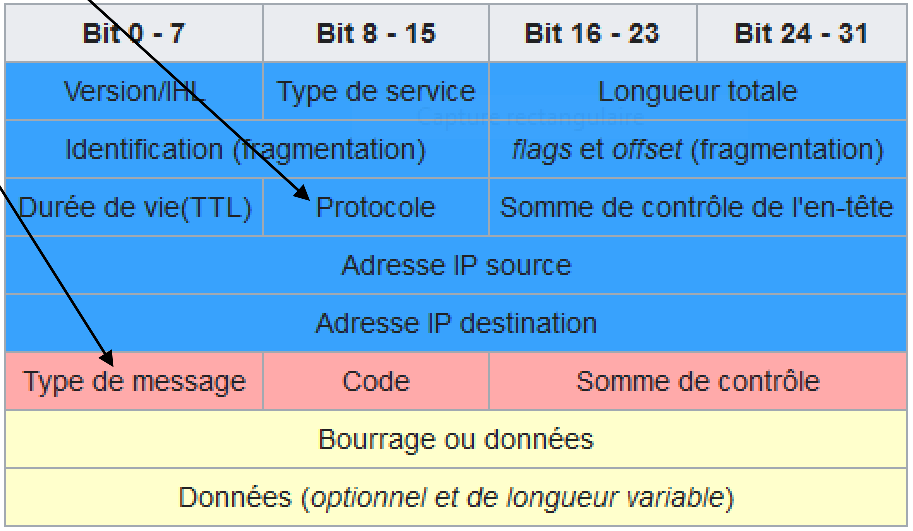

esme@osboxes: ~

File Edit Tabs Help

```
esme@osboxes:~$ sudo tcpdump -i enp0s3 port 21
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp0s3, link-type EN10MB (Ethernet), capture size 262144 bytes
11:34:24.812648 IP osboxes.49462 > ftp.proxad.net.ftp: Flags [S], seq 2725291053
, win 29200, options [mss 1460,sackOK,TS val 3659966974 ecr 0,nop,wscale 7], len
gth 0
11:34:24.829371 IP ftp.proxad.net.ftp > osboxes.49462: Flags [S.], seq 1984001,
ack 2725291054, win 65535, options [mss 1460], length 0
11:34:24.829402 IP osboxes.49462 > ftp.proxad.net.ftp: Flags [.], ack 1, win 292
00, length 0
11:34:24.840052 IP ftp.proxad.net.ftp > osboxes.49462: Flags [P.], seq 1:35, ack
 1, win 65535, length 34: FTP: 220 Welcome to ProXad FTP server
11:34:24.840069 IP osboxes.49462 > ftp.proxad.net.ftp: Flags [.], ack 35, win 29
200, length 0
11:34:24.840209 IP osboxes.49462 > ftp.proxad.net.ftp: Flags [P.], seq 1:17, ack
 35, win 29200, length 16: FTP: USER anonymous
11:34:24.840450 IP ftp.proxad.net.ftp > osboxes.49462: Flags [.], ack 17, win 65
535, length 0
11:34:24.872094 IP ftp.proxad.net.ftp > osboxes.49462: Flags [P.], seq 69:92, ac
k 30, win 65535, length 23: FTP: 230 Login successful.
11:34:24.872247 IP osboxes.49462 > ftp.proxad.net.ftp: Flags [P.], seq 30:36, ac
k 92, win 29200, length 6: FTP: SYST
11:34:24.872505 IP ftp.proxad.net.ftp > osboxes.49462: Flags [.], ack 36, win 65
535, length 0
```

TCPDump Filters

- Format 2:
  - <protocol header> [offset:length] <relation> <value>
- "ip[9] = 1"
  - –Selects any record with the IP protocol of 1 (ICMP).
- "icmp[0] = 8"
  - –Selects any record that is an ICMP echo requests.
    - » That's why you should learn protocol headers format

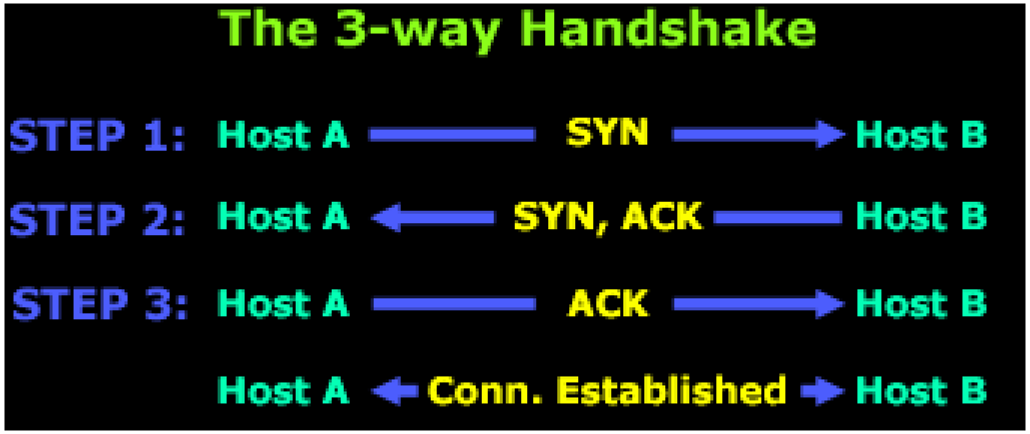


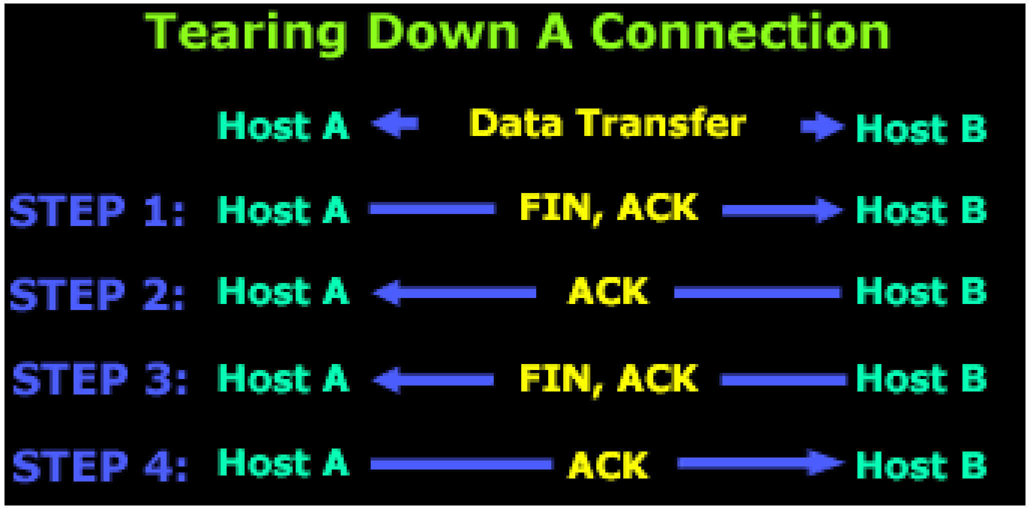
Bit 0 - 7	Bit 8 - 15	Bit 16 - 23	Bit 24 - 31
Version/IHL	Type de service	Longueur totale	
Identification (fragmentation)		*flags* et *offset* (fragmentation)	
Durée de vie(TTL)	Protocole	Somme de contrôle de l'en-tête	
Adresse IP source			
Adresse IP destination			
Type de message	Code	Somme de contrôle	
Bourrage ou données			
Données (*optionnel et de longueur variable*)			

TCPDump Filters

- Reference single bits through bit masking
  - An example is TCP flag bits

- Byte 13 in a TCP header has the 8 flag fields.
  - CWR,ECE,URG,ACK,PSH,RST,SYN,FIN


The 3-way Handshake
STEP 1: Host A SYN Host B
STEP 2: Host A SYN, ACK Host B
STEP 3: Host A ACK Host B
Host A Conn. Established Host B


Tearing Down A Connection
Host A Data Transfer Host B
STEP 1: Host A FIN, ACK Host B
STEP 2: Host A ACK Host B
STEP 3: Host A FIN, ACK Host B
STEP 4: Host A ACK Host B

TCPDump Filters

- Assume we want to mask out the PSH field
- Translate the mask into binary
    - 0x08
- Set filter to
    - tcp[13] & 0x08 != 0
- Your turn:
    - Filter for packets that have the Syn or the Ack flag set

cwr	ece	urg	ack	psh	rst	syn	fin
0	0	0	0	1	0	0	0

TCPDump Filters

- Your turn:
  - Filter for packets that have the Syn or the Ack flag set.
  - tcp[13] & 0x12 != 0
- Try:
  - sudo tcpdump -i eth1 'tcp[13]&0x12!=0'
  - wget http://www.esme.fr/

cwr	ece	urg	ack	psh	rst	syn	fin
0	0	0	1	0	0	1	0

TCPDump Filters

- We can of course use exact values for filtering.

- tcp[13] = 0x20 looks only for tcp-packets that have the urg flag set.

cwr	ece	urg	ack	psh	rst	syn	fin
0	0	1	0	0	0	0	0

TCPDump Filters

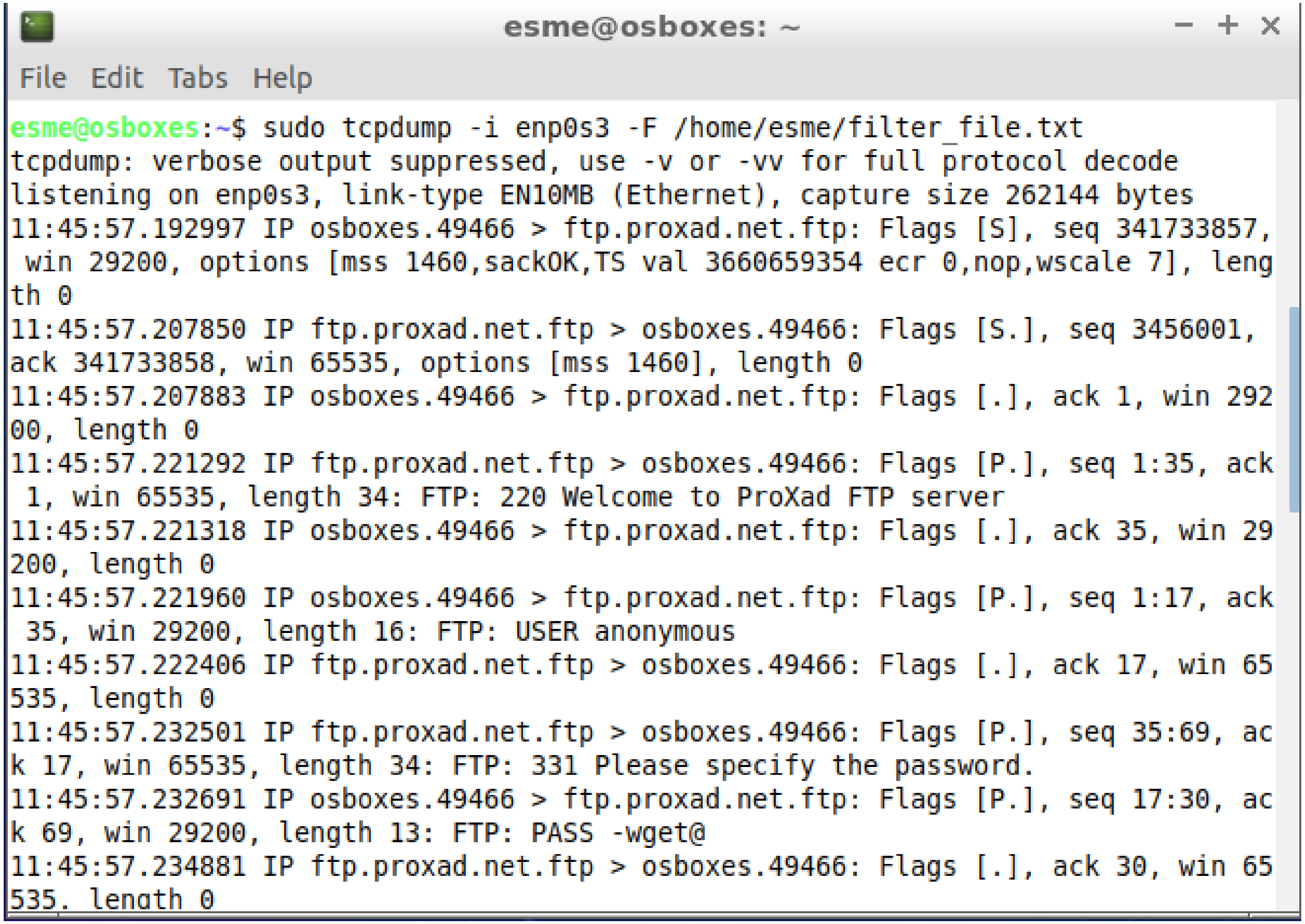
- Can combine filters
  - with the *and*, *or*, *not* operators

- Example
  - (tcp and tcp[13]&0x0f != 0 and not port 25) or port 20

- Filter can be written in file,
  - specified with the –F flag

TCPDump Filters

- Use –F filename to specify a file containing the filter.



```
esme@osboxes:~$ sudo tcpdump -i enp0s3 -F /home/esme/filter_file.txt
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp0s3, link-type EN10MB (Ethernet), capture size 262144 bytes
11:45:57.192997 IP osboxes.49466 > ftp.proxad.net.ftp: Flags [S], seq 341733857,
 win 29200, options [mss 1460,sackOK,TS val 3660659354 ecr 0,nop,wscale 7], leng
th 0
11:45:57.207850 IP ftp.proxad.net.ftp > osboxes.49466: Flags [S.], seq 3456001,
ack 341733858, win 65535, options [mss 1460], length 0
11:45:57.207883 IP osboxes.49466 > ftp.proxad.net.ftp: Flags [.], ack 1, win 292
00, length 0
11:45:57.221292 IP ftp.proxad.net.ftp > osboxes.49466: Flags [P.], seq 1:35, ack
 1, win 65535, length 34: FTP: 220 Welcome to ProXad FTP server
11:45:57.221318 IP osboxes.49466 > ftp.proxad.net.ftp: Flags [.], ack 35, win 29
200, length 0
11:45:57.221960 IP osboxes.49466 > ftp.proxad.net.ftp: Flags [P.], seq 1:17, ack
 35, win 29200, length 16: FTP: USER anonymous
11:45:57.222406 IP ftp.proxad.net.ftp > osboxes.49466: Flags [.], ack 17, win 65
535, length 0
11:45:57.232501 IP ftp.proxad.net.ftp > osboxes.49466: Flags [P.], seq 35:69, ac
k 17, win 65535, length 34: FTP: 331 Please specify the password.
11:45:57.232691 IP osboxes.49466 > ftp.proxad.net.ftp: Flags [P.], seq 17:30, ac
k 69, win 29200, length 13: FTP: PASS -wget@
11:45:57.234881 IP ftp.proxad.net.ftp > osboxes.49466: Flags [.], ack 30, win 65
535, length 0
```

TCPDump other options

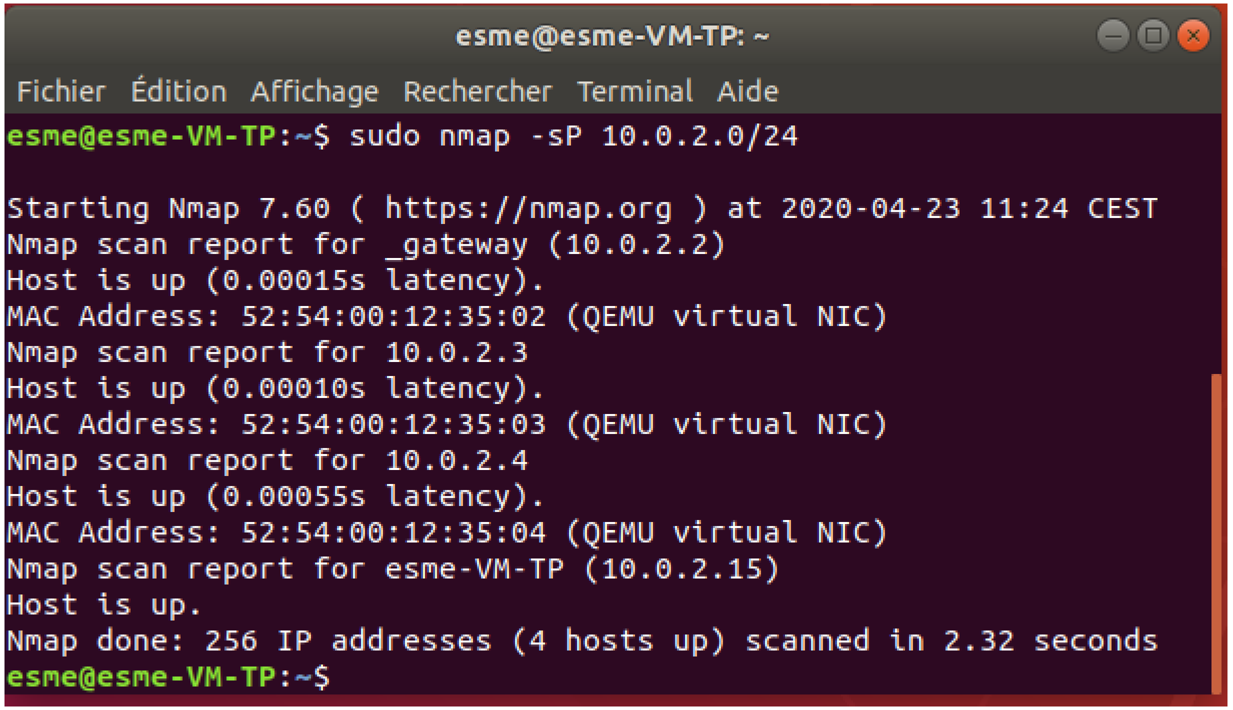
- -n Don't convert addresses (i.e., host addresses, port numbers, etc.) to names.
- –w capture into a file.
- –c limit the number of packets captured.
- –v, -vv, -vvv for verbosity.
- –x for hexadecimal values of packet contents.
- –tttt to display time / day stamps.
- –r to specify capture file.

NMap

- Available in Windows and Linux
- Scans host with many different connections
- Uses responses to determine OS

–Target Acquisition

– Network mapping

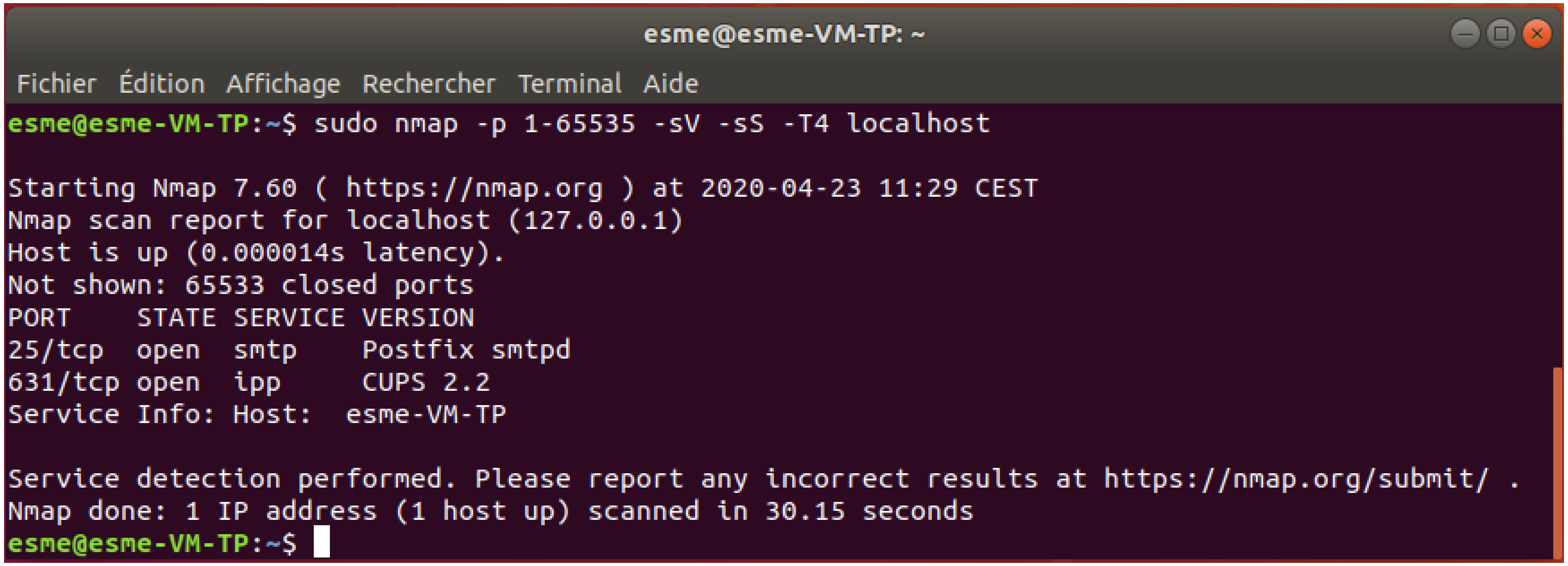


```
esme@esme-VM-TP:~$ sudo nmap -sP 10.0.2.0/24

Starting Nmap 7.60 ( https://nmap.org ) at 2020-04-23 11:24 CEST
Nmap scan report for _gateway (10.0.2.2)
Host is up (0.00015s latency).
MAC Address: 52:54:00:12:35:02 (QEMU virtual NIC)
Nmap scan report for 10.0.2.3
Host is up (0.00010s latency).
MAC Address: 52:54:00:12:35:03 (QEMU virtual NIC)
Nmap scan report for 10.0.2.4
Host is up (0.00055s latency).
MAC Address: 52:54:00:12:35:04 (QEMU virtual NIC)
Nmap scan report for esme-VM-TP (10.0.2.15)
Host is up.
Nmap done: 256 IP addresses (4 hosts up) scanned in 2.32 seconds
esme@esme-VM-TP:~$
```

TCPDump Filter against NMap

- Standard behavior for Nmap port scan
  - The Flags [S] and [R] sent to random destination ports
  - When the SYN finds a closed port a RESET is sent



```
esme@esme-VM-TP:~$ sudo nmap -p 1-65535 -sV -sS -T4 localhost

Starting Nmap 7.60 ( https://nmap.org ) at 2020-04-23 11:29 CEST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000014s latency).
Not shown: 65533 closed ports
PORT    STATE SERVICE VERSION
25/tcp  open  smtp    Postfix smtpd
631/tcp open  ipp     CUPS 2.2
Service Info: Host:  esme-VM-TP

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 30.15 seconds
esme@esme-VM-TP:~$
```

tcptrace

- Uses a network traffic capture file as input
- Found two tcp connections.
- (a2b), (c2d) is a labelling scheme for ports.
- (complete) shows that the connection was gracefully shut down.
- Numbers are the number of packets sent and received.

```
> tcptrace tigris.dmp
1 arg remaining, starting with 'tigris.dmp'
Ostermann's tcptrace -- version 6.4.5 -- Fri Jun 13, 2003

87 packets seen, 87 TCP packets traced elapsed wallclock time: 0:00:00.037900, 2295
pkts/sec analyzed
trace file elapsed time: 0:00:12.180796
TCP connection info:
1: pride.cs.ohiou.edu:54735 - elephus.cs.ohiou.edu:ssh (a2b) 30> 30< (complete)
2: pride.cs.ohiou.edu:54736 - a17-112-152-32.apple.com:http (c2d) 12> 15< (complete)
```

tcptrace

- -l gives detailed statistics.
- -lW estimates the congestion window in addition.
- -o can filter out connections:

–tcptrace –o3,5,7

  - Filters out all but the third, fifth, and seventh connection.

- Allows quick and accurate view of tcp connections.
- With –u also analyzes udp traffic.

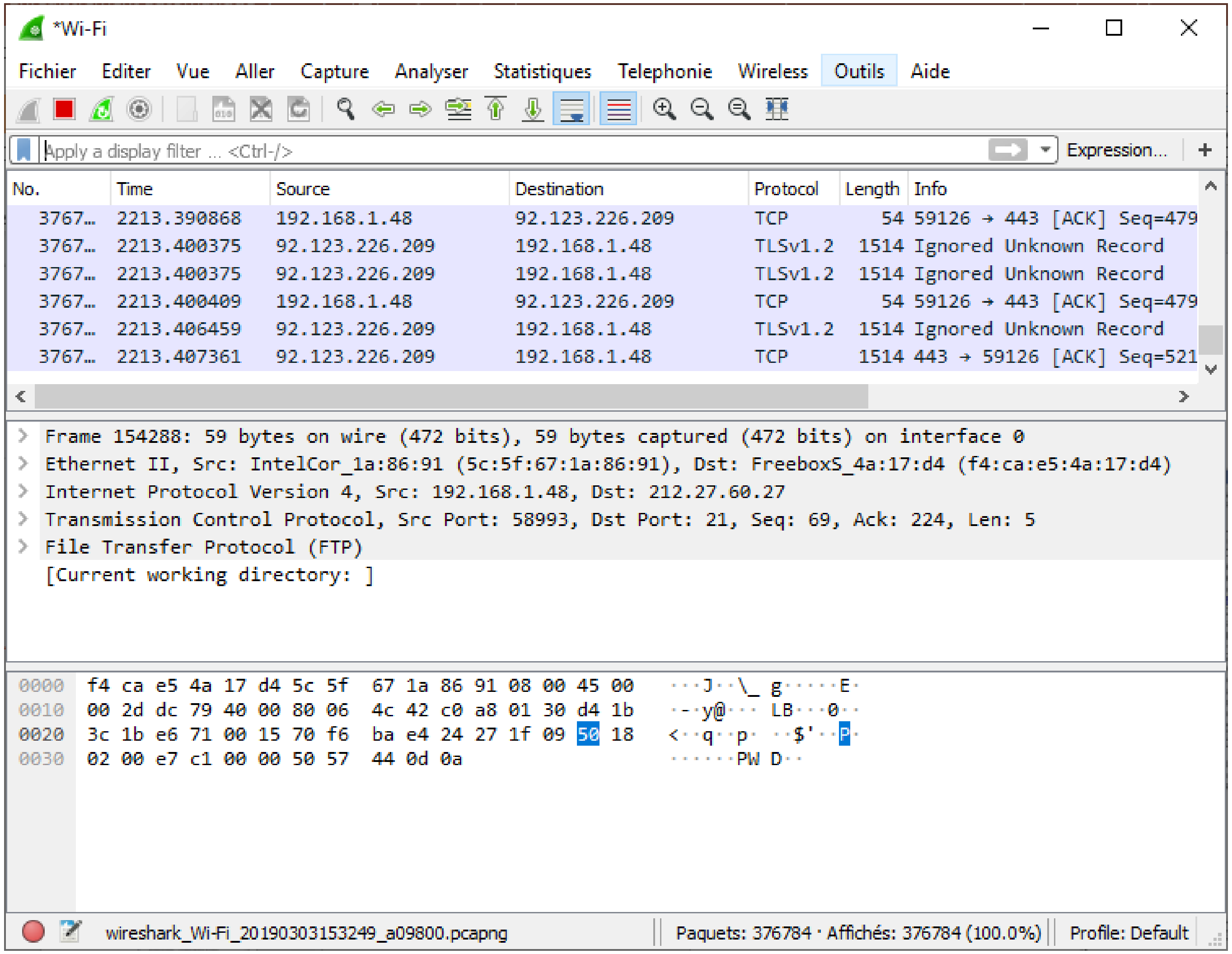
tcpflow

- Captures data transmitted as a TCP connection
  - A flow
- Reconstructs the actual data stream.
  - Reconstruct email, http sessions, …
- tcpflow stores data in files with names of the form
  - 192.168.101.102.02345-010.011.012.013.45103

```
esme@osboxes:~$ cat 010.000.002.004.58196-212.027.060.027.00080
GET / HTTP/1.1
User-Agent: Wget/1.19.4 (linux-gnu)
Accept: */*
Accept-Encoding: identity
Host: ftp.free.fr
Connection: Keep-Alive
```

Wireshark

- GUI tool that can do a lot of neat things
  - –Reconstruct TCP sessions
  - –Handles IP fragmentation
  - –…


*Wi-Fi
Fichier Editer Vue Aller Capture Analyser Statistiques Telephonie Wireless Outils Aide
Apply a display filter … <Ctrl-/>
Expression…

```
No.     Time         Source           Destination      Protocol  Length Info
3767…   2213.390868  192.168.1.48     92.123.226.209   TCP         54 59126 → 443 [ACK] Seq=479
3767…   2213.400375  92.123.226.209   192.168.1.48     TLSv1.2   1514 Ignored Unknown Record
3767…   2213.400375  92.123.226.209   192.168.1.48     TLSv1.2   1514 Ignored Unknown Record
3767…   2213.400409  192.168.1.48     92.123.226.209   TCP         54 59126 → 443 [ACK] Seq=479
3767…   2213.406459  92.123.226.209   192.168.1.48     TLSv1.2   1514 Ignored Unknown Record
3767…   2213.407361  92.123.226.209   192.168.1.48     TCP       1514 443 → 59126 [ACK] Seq=521
```

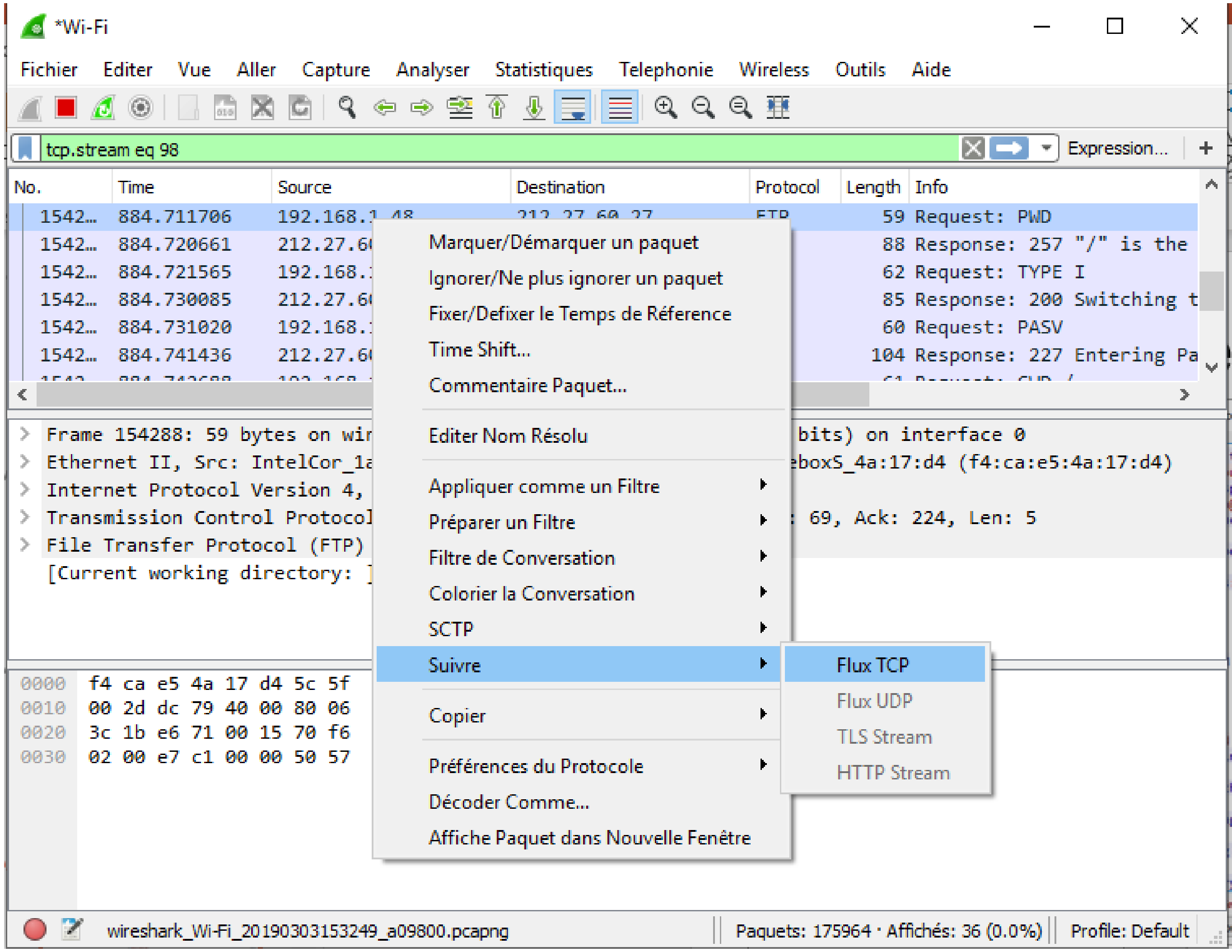
```
> Frame 154288: 59 bytes on wire (472 bits), 59 bytes captured (472 bits) on interface 0
> Ethernet II, Src: IntelCor_1a:86:91 (5c:5f:67:1a:86:91), Dst: FreeboxS_4a:17:d4 (f4:ca:e5:4a:17:d4)
> Internet Protocol Version 4, Src: 192.168.1.48, Dst: 212.27.60.27
> Transmission Control Protocol, Src Port: 58993, Dst Port: 21, Seq: 69, Ack: 224, Len: 5
> File Transfer Protocol (FTP)
  [Current working directory: ]
```

```
0000  f4 ca e5 4a 17 d4 5c 5f  67 1a 86 91 08 00 45 00   ···J··\_ g·····E·
0010  00 2d dc 79 40 00 80 06  4c 42 c0 a8 01 30 d4 1b   ·-·y@··· LB···0··
0020  3c 1b e6 71 00 15 70 f6  ba e4 24 27 1f 09 50 18   <··q··p· ··$'··P·
0030  02 00 e7 c1 00 00 50 57  44 0d 0a                  ······PW D··
```

wireshark_Wi-Fi_20190303153249_a09800.pcapng | Paquets: 376784 · Affichés: 376784 (100.0%) | Profile: Default

Wireshark

- To follow a TCP stream
- highlight packet
- Select: Analyze → Follow → TCP Stream



*Wi-Fi

Fichier Editer Vue Aller Capture Analyser Statistiques Telephonie Wireless Outils Aide

tcp.stream eq 98

```
No.       Time         Source          Destination     Protocol  Length  Info
1542…     884.711706   192.168.1.48    212.27.60.27    FTP           59  Request: PWD
1542…     884.720661   212.27.6                                      88  Response: 257 "/" is the
1542…     884.721565   192.168.1                                     62  Request: TYPE I
1542…     884.730085   212.27.6                                      85  Response: 200 Switching t
1542…     884.731020   192.168.1                                     60  Request: PASV
1542…     884.741436   212.27.6                                     104  Response: 227 Entering Pa
```

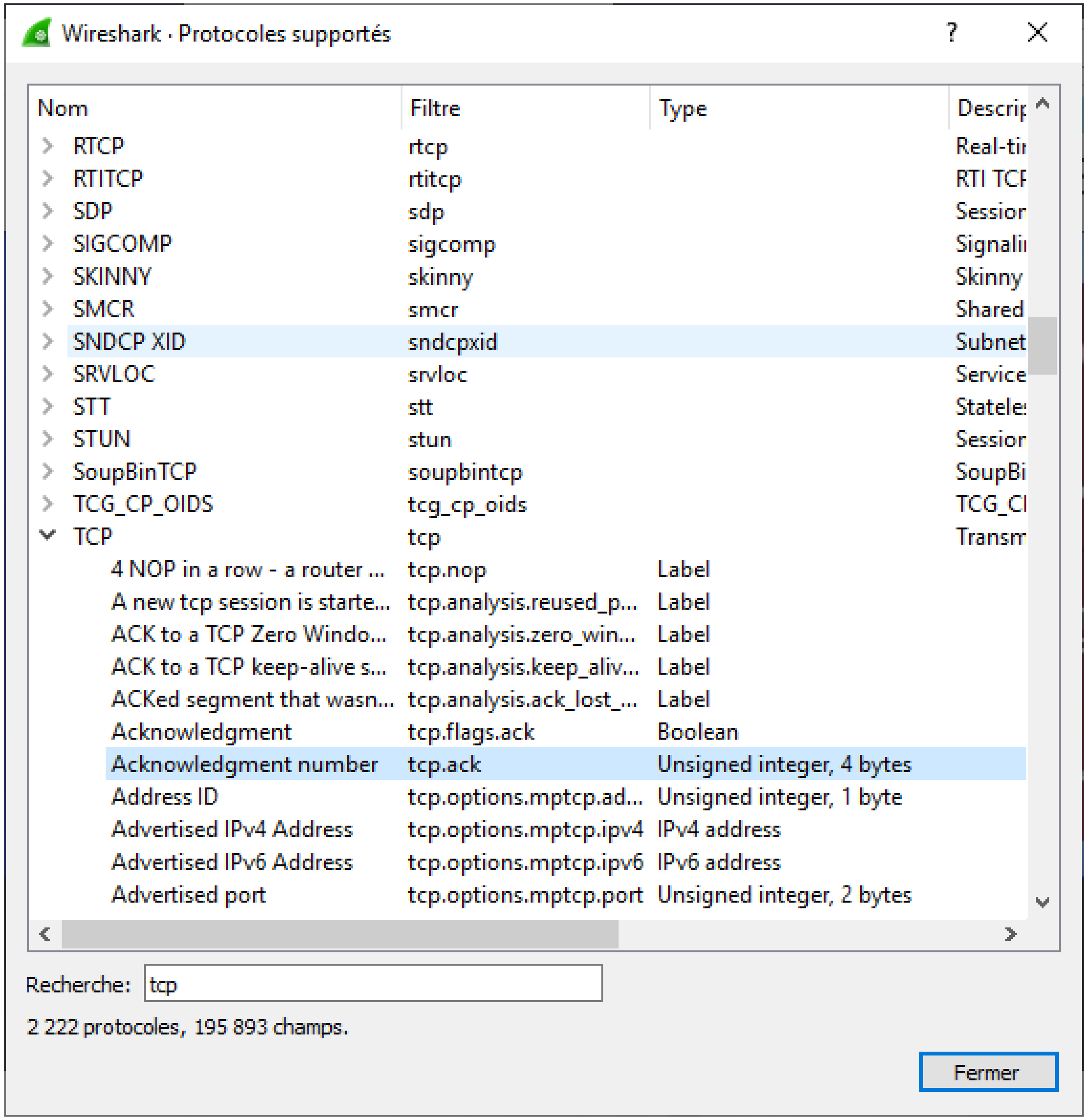
```
> Frame 154288: 59 bytes on wir                       bits) on interface 0
> Ethernet II, Src: IntelCor_1a                       eboxS_4a:17:d4 (f4:ca:e5:4a:17:d4)
> Internet Protocol Version 4,
> Transmission Control Protocol                       : 69, Ack: 224, Len: 5
> File Transfer Protocol (FTP)
  [Current working directory: ]
```

```
0000  f4 ca e5 4a 17 d4 5c 5f
0010  00 2d dc 79 40 00 80 06
0020  3c 1b e6 71 00 15 70 f6
0030  02 00 e7 c1 00 00 50 57
```

wireshark_Wi-Fi_20190303153249_a09800.pcapng | Paquets: 175964 · Affichés: 36 (0.0%) | Profile: Default

Wireshark

- Filters
  - Only packets that fit the filter are captured.
  - Available filter fields are under "View" → "Internals" → "Supported Protocols"



Wireshark · Protocoles supportés

Nom	Filtre	Type	Descrip
RTCP	rtcp		Real-ti
RTITCP	rtitcp		RTI TCF
SDP	sdp		Session
SIGCOMP	sigcomp		Signali
SKINNY	skinny		Skinny
SMCR	smcr		Shared
SNDCP XID	sndcpxid		Subnet
SRVLOC	srvloc		Service
STT	stt		Statele
STUN	stun		Session
SoupBinTCP	soupbintcp		SoupBi
TCG_CP_OIDS	tcg_cp_oids		TCG_CI
TCP	tcp		Transm
4 NOP in a row - a router ...	tcp.nop	Label	
A new tcp session is starte...	tcp.analysis.reused_p...	Label	
ACK to a TCP Zero Windo...	tcp.analysis.zero_win...	Label	
ACK to a TCP keep-alive s...	tcp.analysis.keep_aliv...	Label	
ACKed segment that wasn...	tcp.analysis.ack_lost_...	Label	
Acknowledgment	tcp.flags.ack	Boolean	
Acknowledgment number	tcp.ack	Unsigned integer, 4 bytes	
Address ID	tcp.options.mptcp.ad...	Unsigned integer, 1 byte	
Advertised IPv4 Address	tcp.options.mptcp.ipv4	IPv4 address	
Advertised IPv6 Address	tcp.options.mptcp.ipv6	IPv6 address	
Advertised port	tcp.options.mptcp.port	Unsigned integer, 2 bytes	

Recherche: tcp

2 222 protocoles, 195 893 champs.

Fermer

Wireshark

- Filters
  - Comparisons are specified with 2 letter abbreviations or C-like syntax
    - ip.addr==10.0.0.5
    - ip.addr!=10.0.0.5
    - frame.pkt_len ge 0x100 and tcp
    - tr.dst[0:3] == 0.6.29 xor tr.src[0:3] == 0.6.29

Wireshark

- Filters

–Expressions can be combined with English or C-like syntax

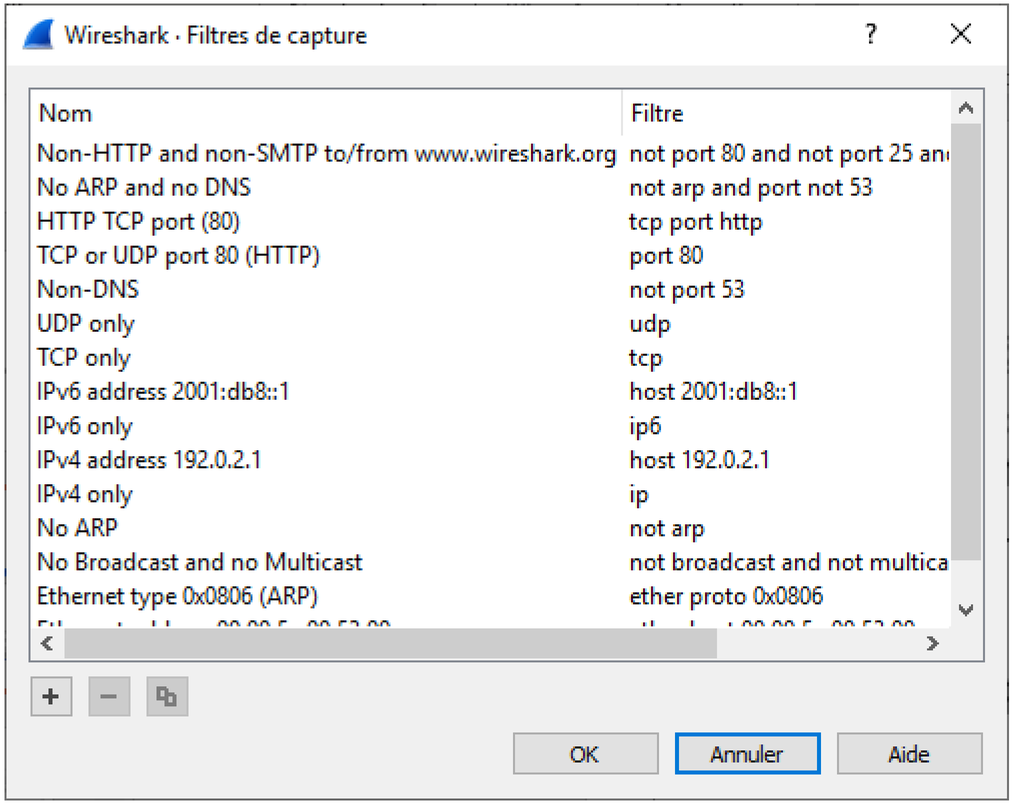
  - ip.addr eq 10.0.0.5 and tcp.flags.fin
  - tcp.flags.syn || tcp.flags.ack

–Allows selection of subsequences.

  - After a label, place a pair of brackets containing a comma separated list of range specifiers:
    - eth.src[0:3] == 00:00:83
    - eth.src[1-2] == 00:83
    - eth.src[0:3,1-2,:4,4:,2] == 00:00:83:00:83:00:00:83:00:20:83

Wireshark

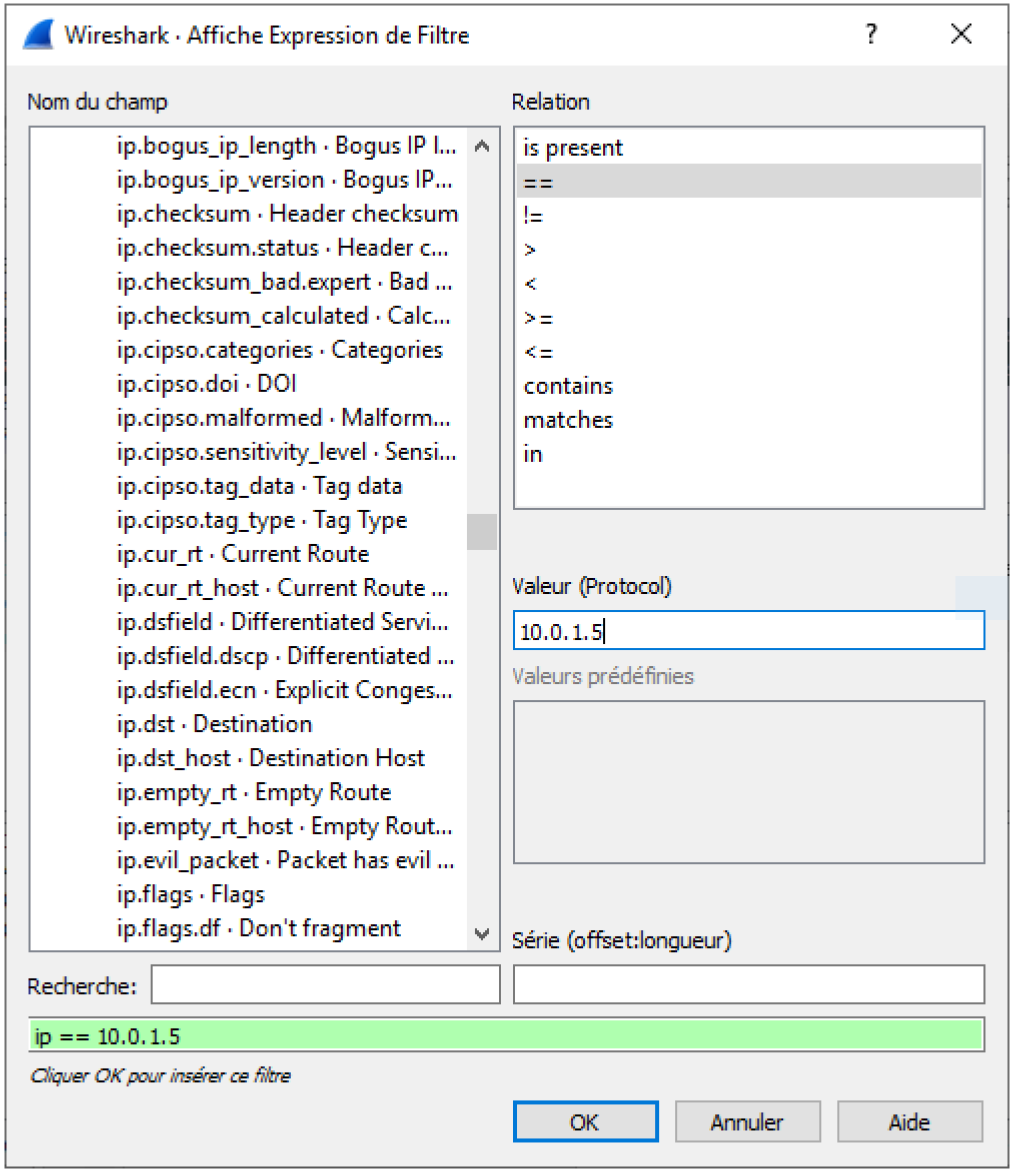
- Defined Filters
  - Use filters predefined in Wireshark
  - Capture → Capture Filters
  - Add/remove new filters


Wireshark · Filtres de capture

Nom	Filtre
Non-HTTP and non-SMTP to/from www.wireshark.org	not port 80 and not port 25 an
No ARP and no DNS	not arp and port not 53
HTTP TCP port (80)	tcp port http
TCP or UDP port 80 (HTTP)	port 80
Non-DNS	not port 53
UDP only	udp
TCP only	tcp
IPv6 address 2001:db8::1	host 2001:db8::1
IPv6 only	ip6
IPv4 address 192.0.2.1	host 192.0.2.1
IPv4 only	ip
No ARP	not arp
No Broadcast and no Multicast	not broadcast and not multica
Ethernet type 0x0806 (ARP)	ether proto 0x0806

Wireshark

- Filter Expression
  - Build filters with Filter Expression dialog box.
  - Analyze -> Display Filter-> Expression…



Nom du champ:

```
ip.bogus_ip_length · Bogus IP l...
ip.bogus_ip_version · Bogus IP...
ip.checksum · Header checksum
ip.checksum.status · Header c...
ip.checksum_bad.expert · Bad ...
ip.checksum_calculated · Calc...
ip.cipso.categories · Categories
ip.cipso.doi · DOI
ip.cipso.malformed · Malform...
ip.cipso.sensitivity_level · Sensi...
ip.cipso.tag_data · Tag data
ip.cipso.tag_type · Tag Type
ip.cur_rt · Current Route
ip.cur_rt_host · Current Route ...
ip.dsfield · Differentiated Servi...
ip.dsfield.dscp · Differentiated ...
ip.dsfield.ecn · Explicit Conges...
ip.dst · Destination
ip.dst_host · Destination Host
ip.empty_rt · Empty Route
ip.empty_rt_host · Empty Rout...
ip.evil_packet · Packet has evil ...
ip.flags · Flags
ip.flags.df · Don't fragment
```

Relation:

```
is present
==
!=
>
<
>=
<=
contains
matches
in
```

Valeur (Protocol): 10.0.1.5

```
ip == 10.0.1.5
```

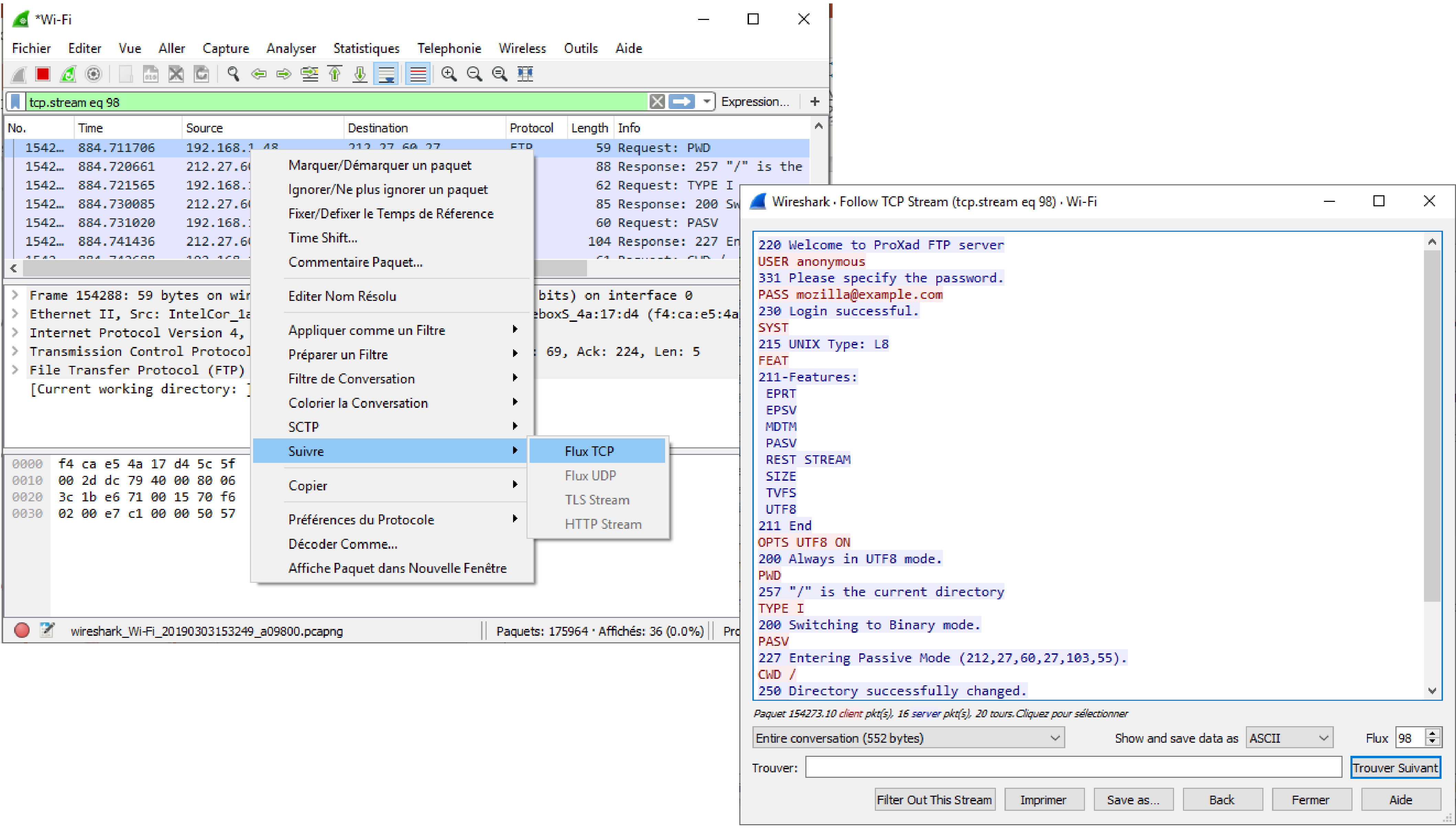
Cliquer OK pour insérer ce filtre

Wireshark

Exercise

1. Start wireshark capture
2. Open http://ftp.free.fr/ on a browser
3. Identify the TCP connection with the right filter
4. View content of TCP connection

Wireshark



Filtre : tcp.stream eq 98

```
No.      Time        Source        Destination   Protocol  Length  Info
1542…    884.711706  192.168.1.48  212.27.60.27  FTP           59  Request: PWD
1542…    884.720661  212.27.6                                  88  Response: 257 "/" is the
1542…    884.721565  192.168.1                                 62  Request: TYPE I
1542…    884.730085  212.27.6                                  85  Response: 200 Sw
1542…    884.731020  192.168.1                                 60  Request: PASV
1542…    884.741436  212.27.6                                 104  Response: 227 En
```

```
> Frame 154288: 59 bytes on wir                 bits) on interface 0
> Ethernet II, Src: IntelCor_1a                 eboxS_4a:17:d4 (f4:ca:e5:4a
> Internet Protocol Version 4,
> Transmission Control Protocol                 : 69, Ack: 224, Len: 5
> File Transfer Protocol (FTP)
  [Current working directory: ]
```

```
0000  f4 ca e5 4a 17 d4 5c 5f
0010  00 2d dc 79 40 00 80 06
0020  3c 1b e6 71 00 15 70 f6
0030  02 00 e7 c1 00 00 50 57
```

wireshark_Wi-Fi_20190303153249_a09800.pcapng | Paquets: 175964 · Affichés: 36 (0.0%)

```
220 Welcome to ProXad FTP server
USER anonymous
331 Please specify the password.
PASS mozilla@example.com
230 Login successful.
SYST
215 UNIX Type: L8
FEAT
211-Features:
 EPRT
 EPSV
 MDTM
 PASV
 REST STREAM
 SIZE
 TVFS
 UTF8
211 End
OPTS UTF8 ON
200 Always in UTF8 mode.
PWD
257 "/" is the current directory
TYPE I
200 Switching to Binary mode.
PASV
227 Entering Passive Mode (212,27,60,27,103,55).
CWD /
250 Directory successfully changed.
```

Paquet 154273. 10 client pkt(s), 16 server pkt(s), 20 tours. Cliquez pour sélectionner

Entire conversation (552 bytes) — Show and save data as: ASCII — Flux: 98

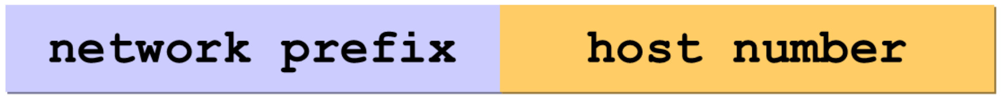
Networking basics

How packets move ?

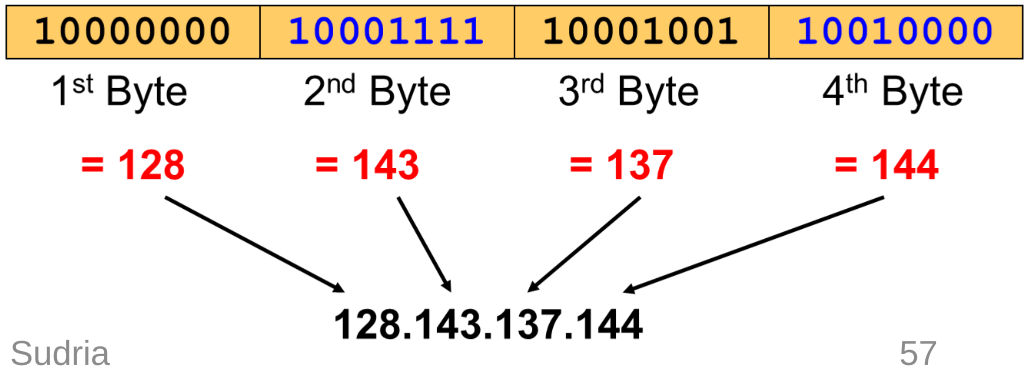
- Plan
  - –Structure of an IP address
  - –Classful IP addresses
  - –Limitations and problems with classful IP addresses
  - –Subnetting
  - –CIDR
  - –IP Version 6 addresses

What is an IP Address?

- A unique global address for a network interface
  - –Dynamically assigned → DHCP
  - –Private networks → NAT
  - –32 bit long identifier (IPv4)
  - –encodes a network prefix
  - –and a host number



- How long the network prefix is?
  - <1993: class-based addressing
  - After, the network prefix is indicated by a netmask
- dotted decimal notation



Example

- Example: www.esme.fr

91.212	**26.84**

- Network address is: `91.212.0.0(or 91.212)`
- Host number is: `26.84`
- Netmask is: `255.255.0.0` (or `ffff0000`)
- Prefix or CIDR notation: `91.212.26.84/16`
    - » Network prefix is 16 bits long

Special IP Addresses

- Reserved special addresses: Loopback interfaces
  - –127.0.0.1-127.255.255.255 reserved loopback interfaces
- IP address of a network
  - –Host number is set to all zeros, e.g., 128.143.**0.0**
- Broadcast address
  - –Host number is all ones, e.g., 128.143.**255.255**
- Private addresses
  - – should get dropped
  - – 10.0.0.0-10.255.255.255; 172.16.0.0-172.31.255.255; 192.168.0.0-192.168.255.255
- Convention (but not a reserved address)
  - Default gateway has host number ‘1’, e.g. 192.0.1.1

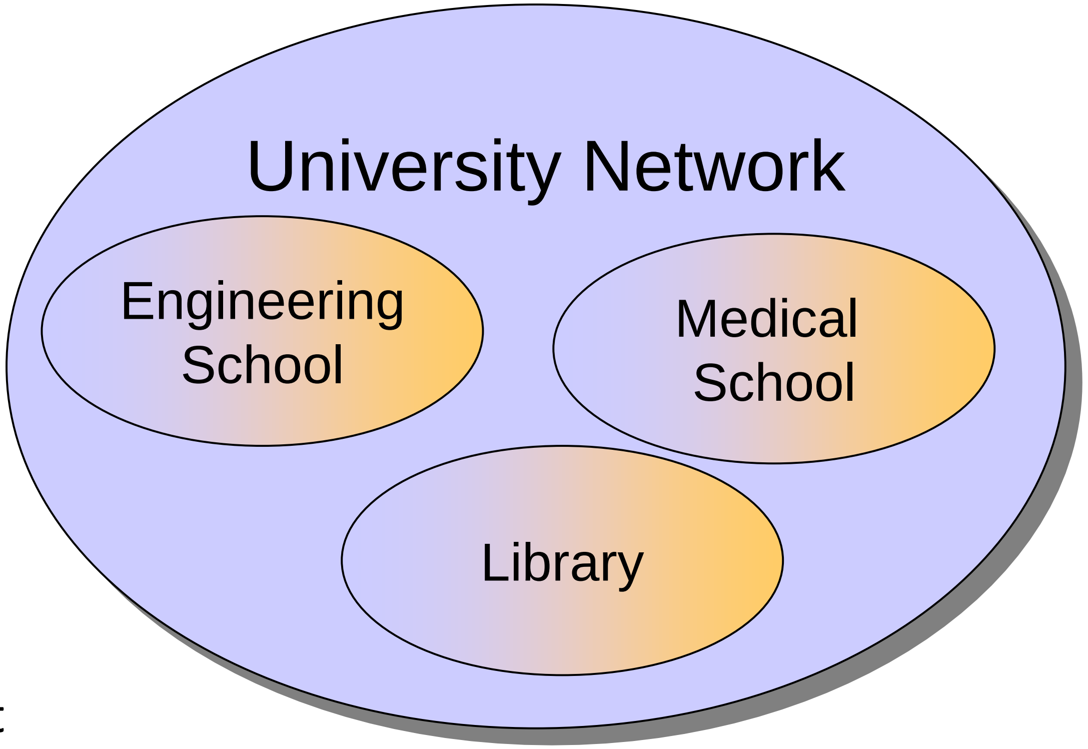
Subnetting

Problem: Organizations have multiple networks which are independently managed

Solution 1: Allocate a separate network address for each network

- Difficult to manage
- From the outside of the organization, each network must be addressable.

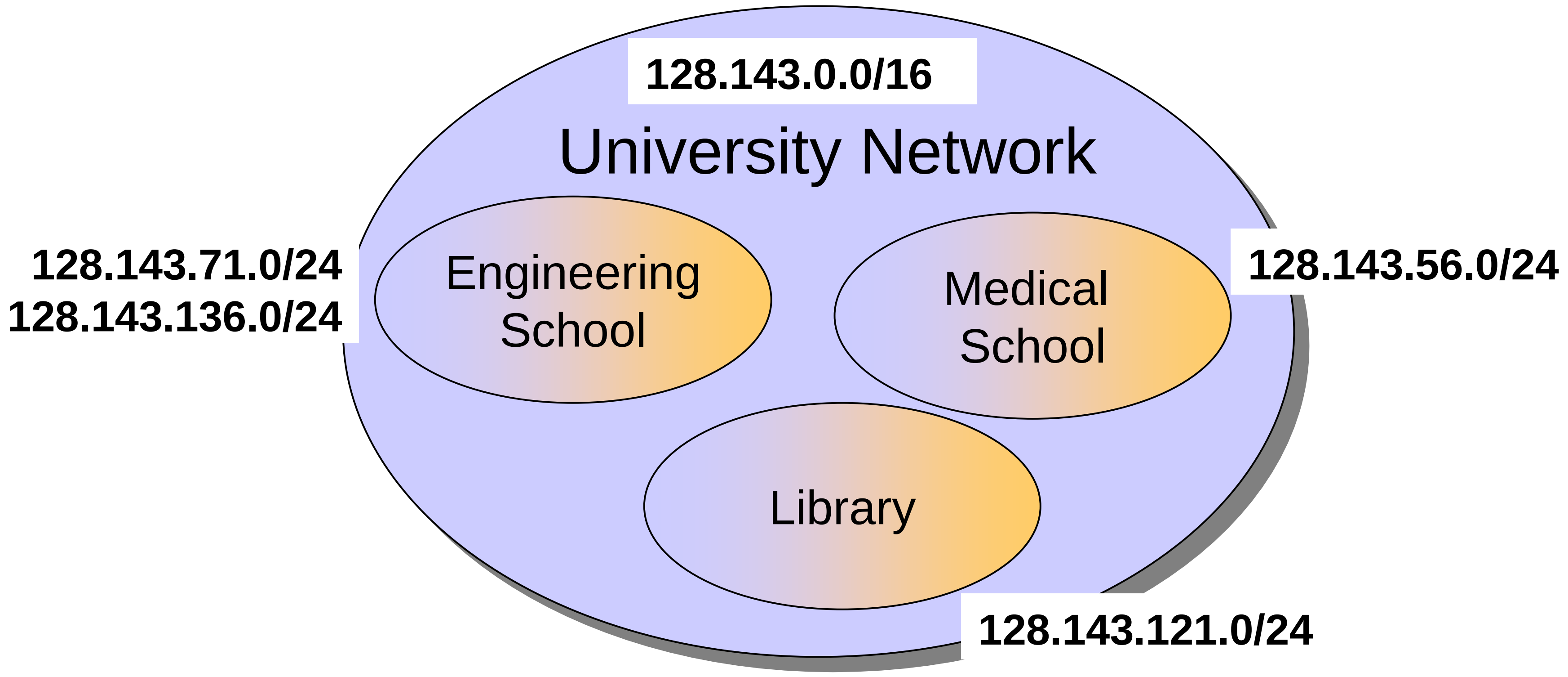
Solution 2: Add another level of hierarchy to the IP addressing structure



→ **Subnetting**

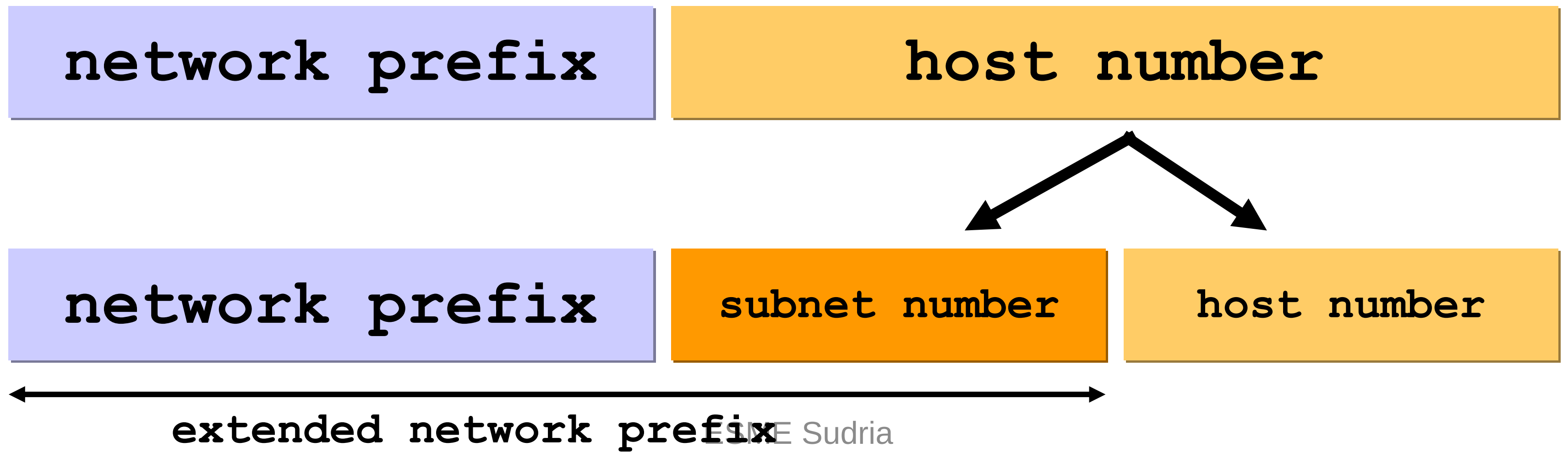
Address assignment with subnetting

- Each part of the organization is allocated a range of IP addresses (subnets or subnetworks)
- Addresses in each subnet can be administered locally



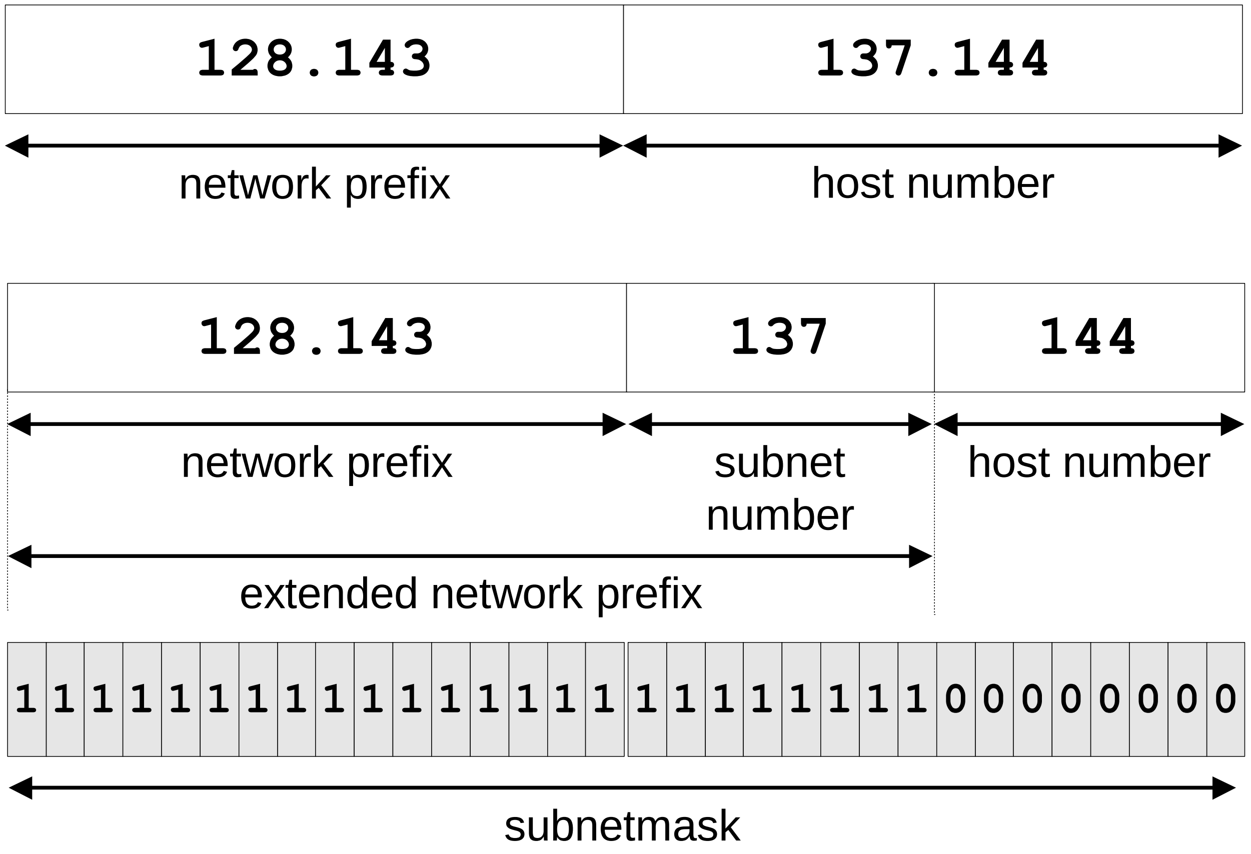
Basic Idea of Subnetting

- Split the host number portion of an IP address into a subnet number and a (smaller) host number.
- Result is a 3-layer hierarchy
- Then:
  - Subnets can be freely assigned within the organization
  - Internally, subnets are treated as separate networks
  - Subnet structure is not visible outside the organization



Subnetmask

- Routers and hosts use an extended network prefix (subnetmask) to identify the start of the host numbers

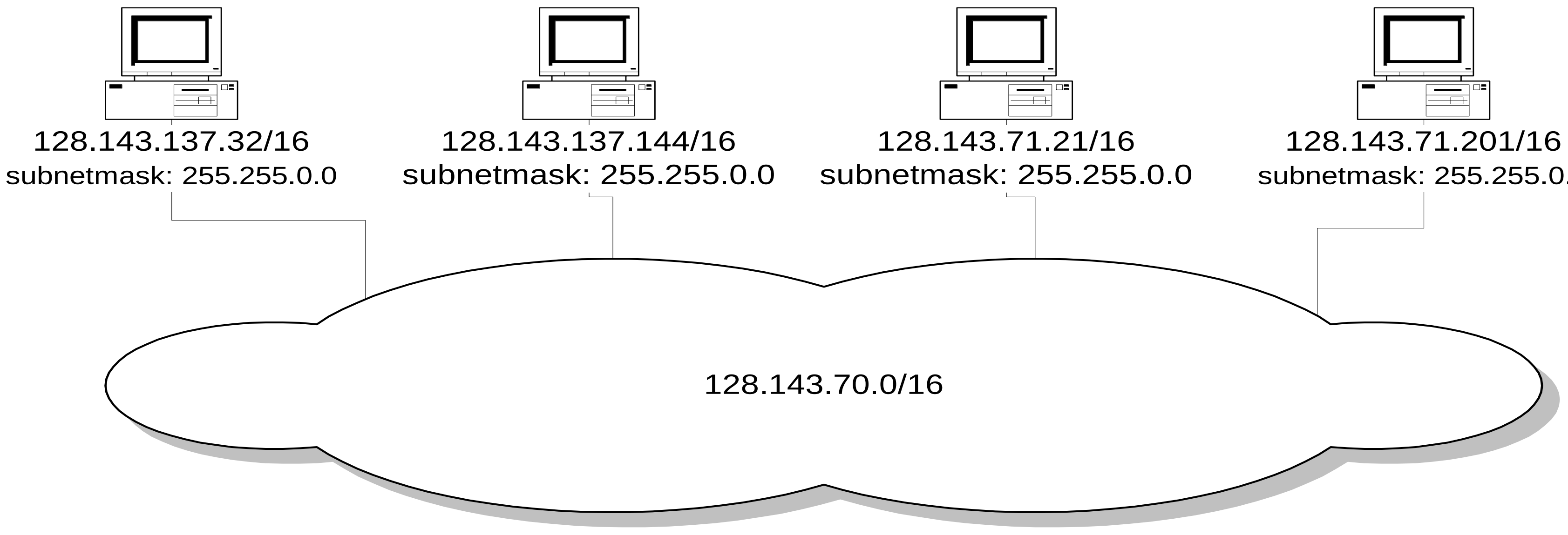


Advantages of Subnetting

- 3-layer hierarchy: Network, Subnet, Host
- Reduces router complexity
  - Since external routers do not know about subnetting
  - Same length of the subnet mask for all subnetworks ?
- Example
  - 128.143.0.0/16 is the IP address of the network
  - 128.143.137.0/24 is the IP address of the subnet
  - 128.143.137.144 is the IP address of the host
  - 255.255.255.0 (or ffffff00) is the subnetmask of the host
- Consistency of subnetmasks is responsibility of administrator

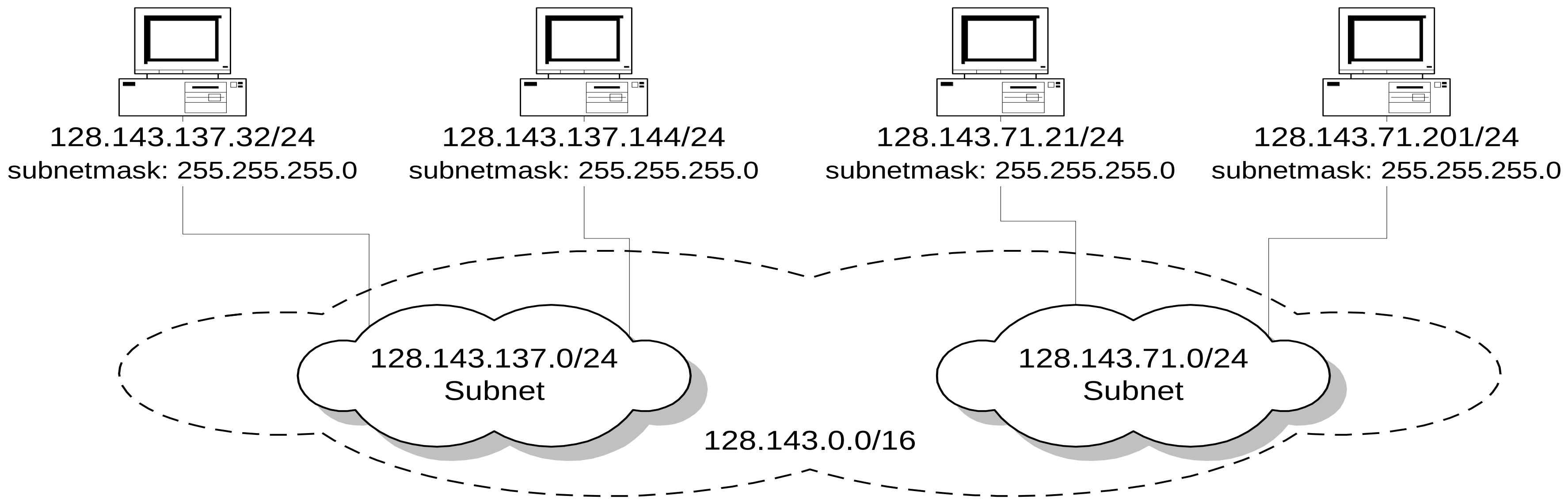
No Subnetting

- All hosts think that the other hosts are on the same network



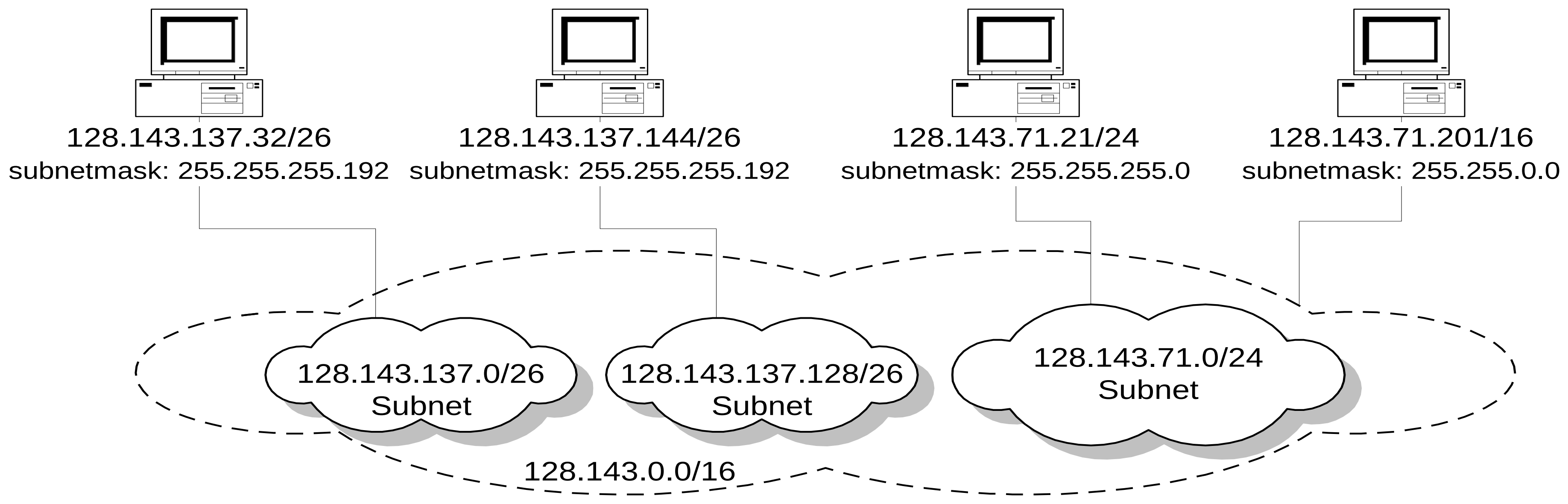
With Subnetting

- Hosts with same extended network prefix belong to the same network

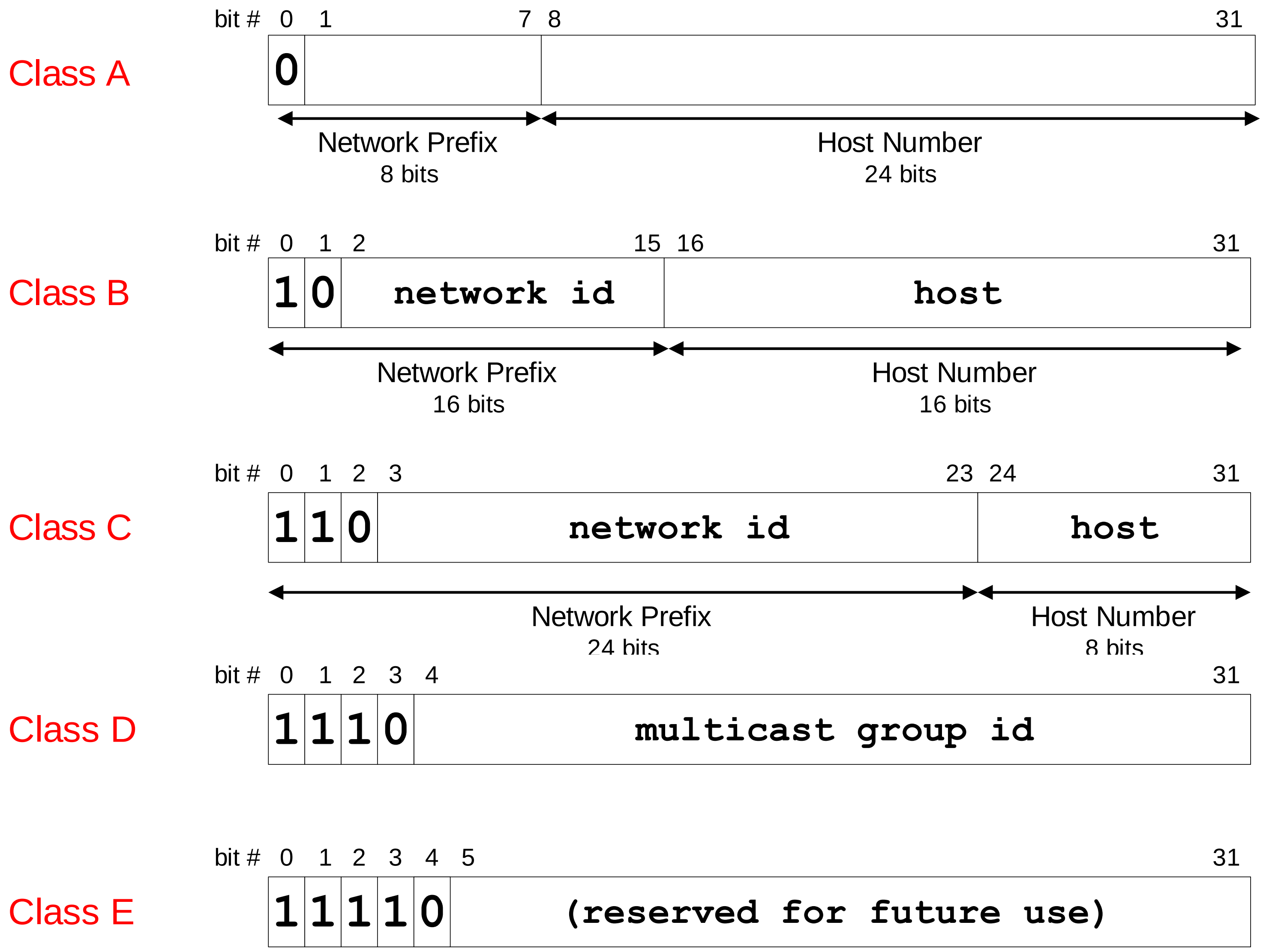


With Subnetting

- Different subnetmasks lead to different views of the size of the scope of the network



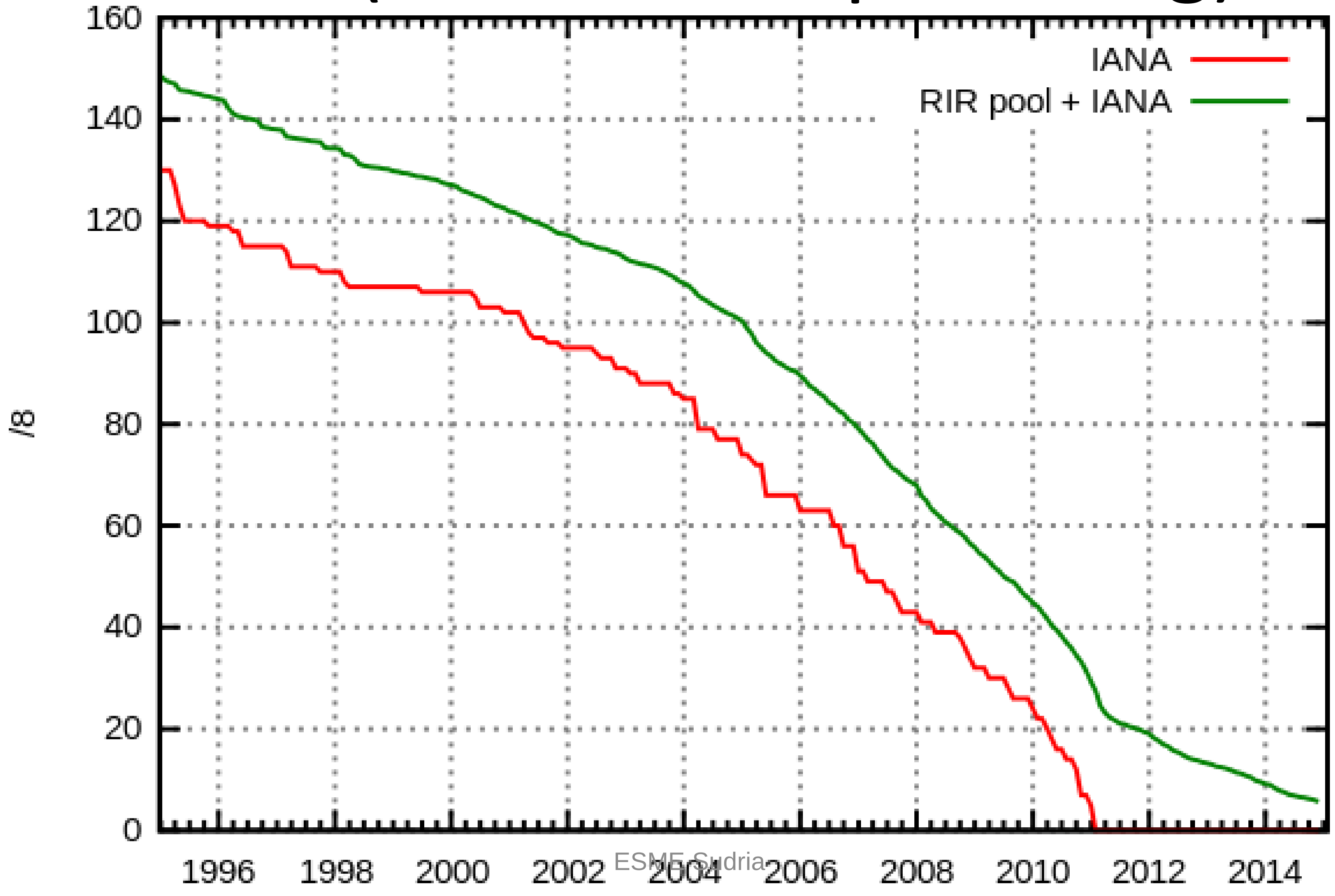
Classful IP Adresses (Until 1993)



Problems with Classful IP Addresses

- Flat address space
  - Routing tables on the backbone : an entry for each network address.
  - Class C networks were widely used
  - The size of the routing tables outgrows the capacity of routers
- Too few network addresses for large networks
  - Class A and Class B addresses were gone
- Limited flexibility for network addresses
  - Class A and B addresses are overkill (>64,000 addresses)
  - Class C address is insufficient (requires 40 Class C addresses)

Exhaustion of IPv4 addresses since 1995 (source: wikipedia.org)



CIDR - Classless Interdomain Routing

- New interpretation of the IP address space
- Restructure IP address assignments to increase efficiency
- Permits route aggregation to minimize route table entries
- Abandons the notion of classes

- Key Concept: length of the network prefix is arbitrary

- Consequence: size of the network prefix must be provided with an IP address

CIDR Notation

- CIDR notation of an IP address:
  - 192.0.2.0/18
    - "18" is the prefix length.
    - first 18 bits are the network prefix of the address
    - 14 bits are available for specific host addresses
- CIDR notation can replace the use of subnetmasks
  - –IP address 128.143.137.144 and subnetmask 255.255.255.0 becomes 128.143.137.144/24
- Allows to drop trailing zeros of network addresses
  - 192.0.2.0/18 can be written as 192.0.2/18

CIDR address blocks

- CIDR notation can nicely express blocks of addresses
- Blocks are used when allocating IP addresses for a company and for routing tables (route aggregation)

CIDR Block Prefix	# of Host Addresses
/27	32
/26	64
/25	128
/24	256
/23	512
/22	1,024
/21	2,048

CIDR Block Prefix	# of Host Addresses
/20	4,096
/19	8,192
/18	16,384
/17	32,768
/16	65,536
/15	131,072
/14	262,144
/13	524,288

CIDR and Address assignments

- Backbone ISPs obtain large block of IP addresses space
  - and then reallocate portions of their address blocks to their customers.
- Example:
  - An ISP owns the address block 206.0.64.0/18
  - Suppose a client requires 800 host addresses
- With classful addresses:
  - Assign a class B address (and waste ~64,700 addresses)
  - 4 individual Class Cs (4 new routes into the Internet routing tables)
- With CIDR:
  - 206.0.68.0/22 and allocated a block of 1,024 (2^{10}) IP addresses

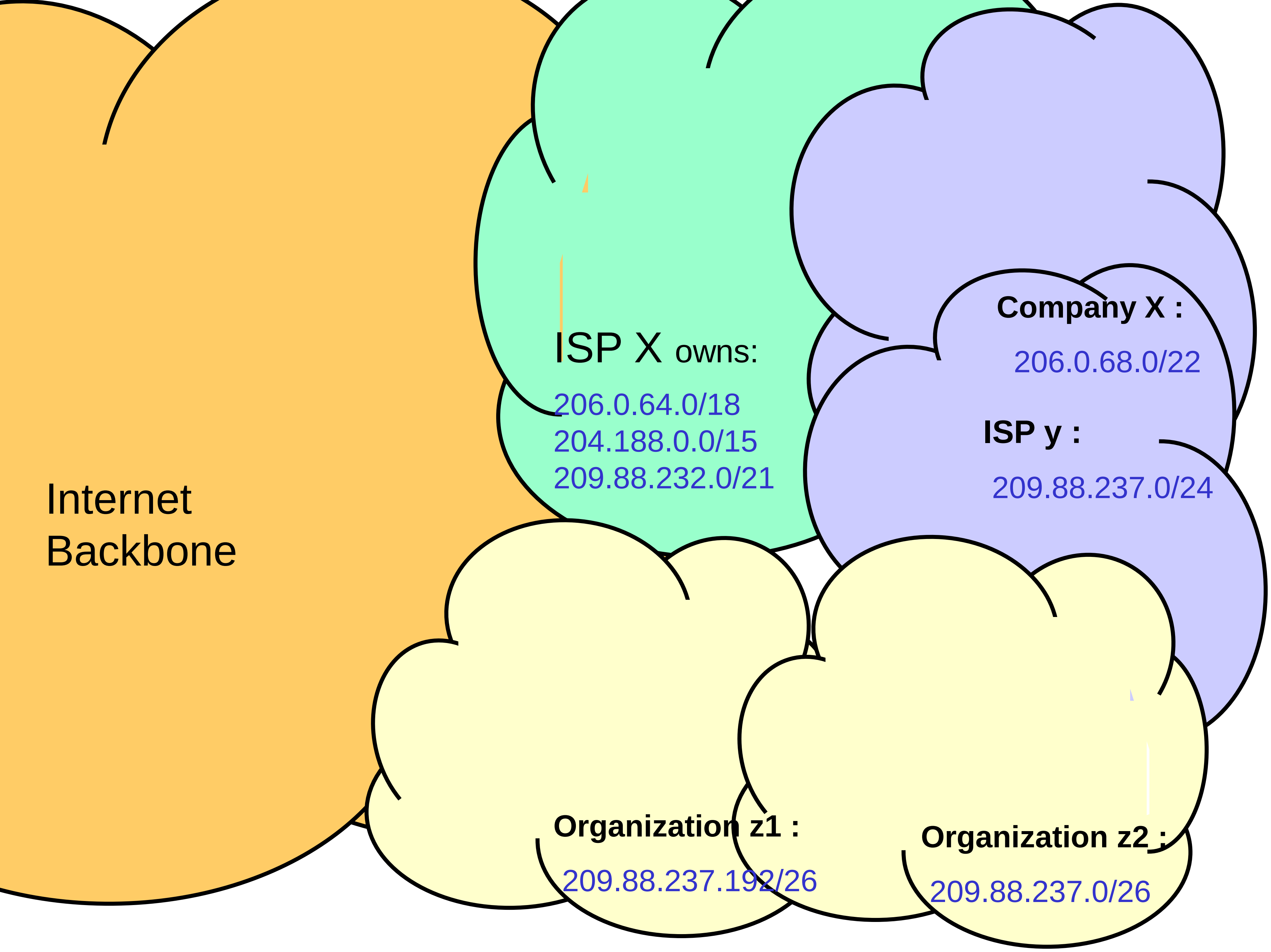
CIDR and Routing

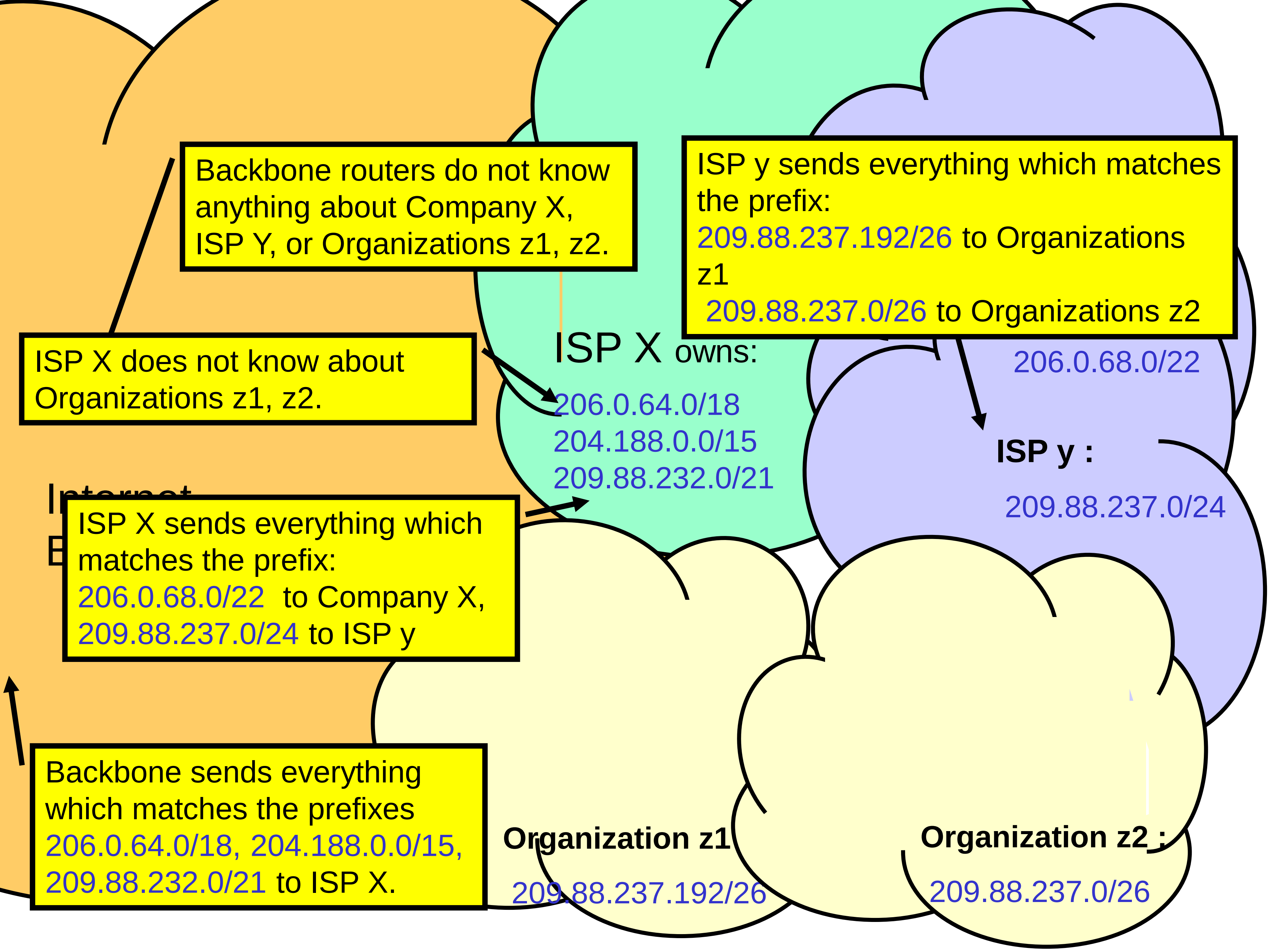
- Aggregation of routing table entries:
  - 128.143.0.0/16 and 128.144.0.0/16 are represented as 128.128.0.0/11
- convert all the network addresses to binary
- find all the identical digits, starting from the left.
- the digits are all the same up to the *N*th position: so your new mask is /*N*
- Find the network address by ANDing the address and the mask

CIDR and Routing

- Longest prefix match
  - Routing table lookup finds the routing entry that matches the longest prefix
- Route aggregation can be exploited
  - when IP address blocks are assigned in a hierarchical fashion
- What is the outgoing interface for 128.143.137.0/24 ?

ID	Prefix	Interface
1	128.0.0.0/4	interface #5
2	128.128.0.0/9	interface #2
3	128.143.128.0/17	interface #1


Internet Backbone
ISP X owns:
206.0.64.0/18
204.188.0.0/15
209.88.232.0/21
Company X :
206.0.68.0/22
ISP y :
209.88.237.0/24
Organization z1 :
209.88.237.192/26
Organization z2 :
209.88.237.0/26



Backbone routers do not know anything about Company X, ISP Y, or Organizations z1, z2.

ISP X does not know about Organizations z1, z2.

ISP y sends everything which matches the prefix:
209.88.237.192/26 to Organizations z1
209.88.237.0/26 to Organizations z2

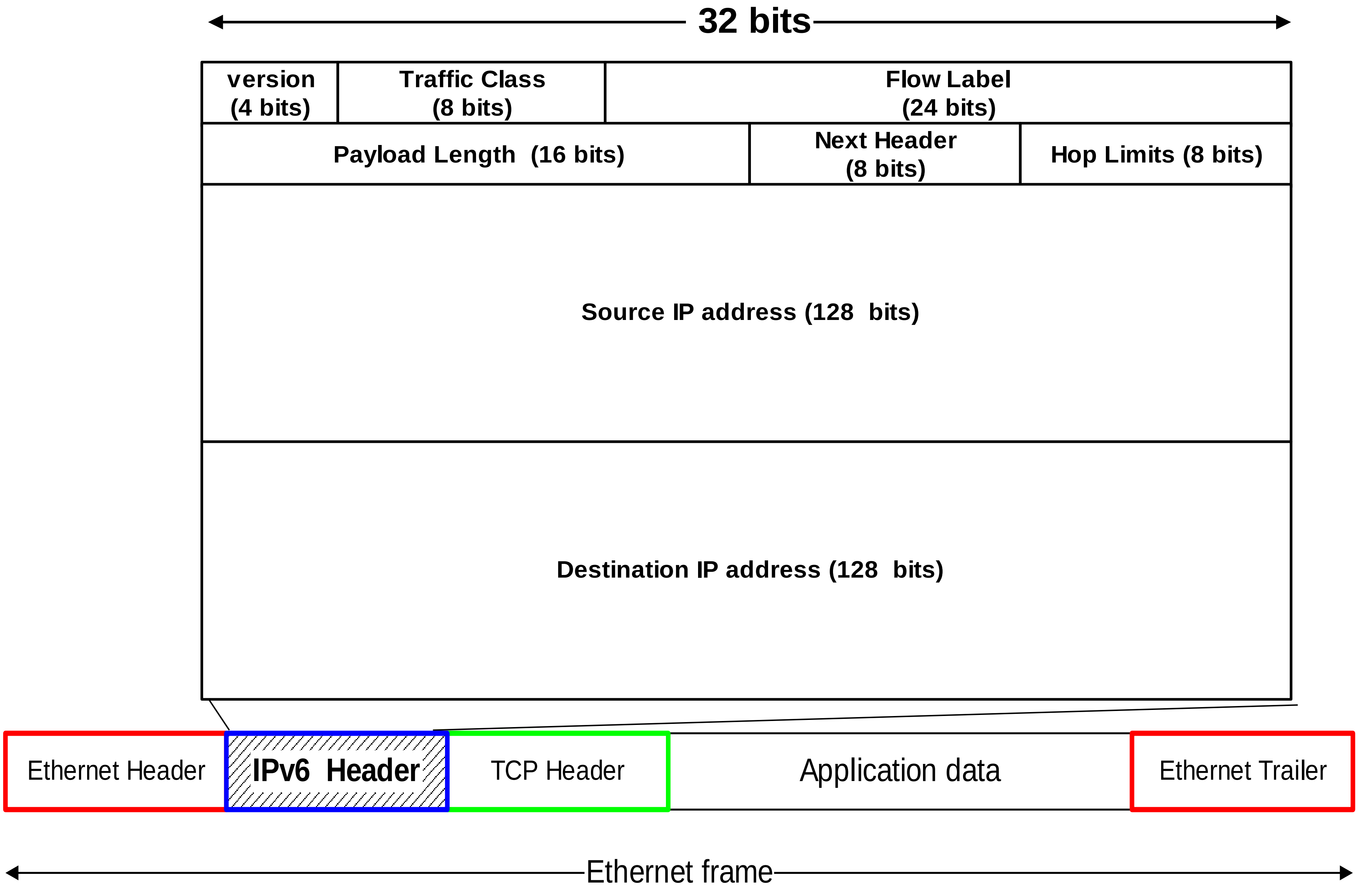
ISP X sends everything which matches the prefix:
206.0.68.0/22 to Company X,
209.88.237.0/24 to ISP y

Backbone sends everything which matches the prefixes 206.0.64.0/18, 204.188.0.0/15, 209.88.232.0/21 to ISP X.

IPv6 - IP Version 6

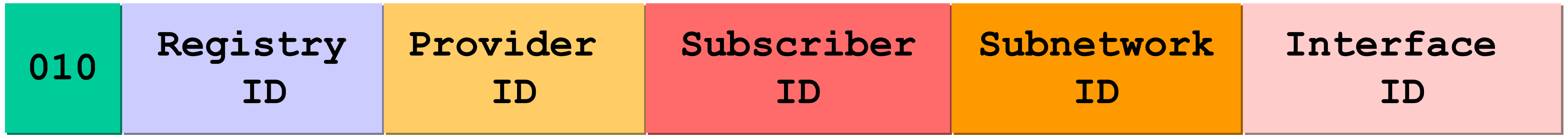
- IP Version 6
  - Makes improvements to IPv4 (no revolutionary changes)
- Main feature of IPv6
  - significant increase of the IP address to 128 bits (16 bytes)
  - 10^{24} addresses per square inch on the surface of the Earth.
- Maximum number of addresses
  - IPv4 has $2^{32} \approx$ 4 billion addresses
  - IPv6 has $2^{128} = (2^{32})^4 \approx$ 4 billion x 4 billion x 4 billion x 4 billion
- Notation of IPv6 addresses
  - 16-bit integers: CEDF:BP76:3245:4464:FACE:2E50:3025:DF12
  - Abbreviations: CEDF:BP76:0000:0000:009E:0000:3025:DF12 -> CEDF:BP76:0:0:9E :0:3025:DF12

IPv6 Header


32 bits
version (4 bits)
Traffic Class (8 bits)
Flow Label (24 bits)
Payload Length (16 bits)
Next Header (8 bits)
Hop Limits (8 bits)
Source IP address (128 bits)
Destination IP address (128 bits)
Ethernet Header
IPv6 Header
TCP Header
Application data
Ethernet Trailer
Ethernet frame

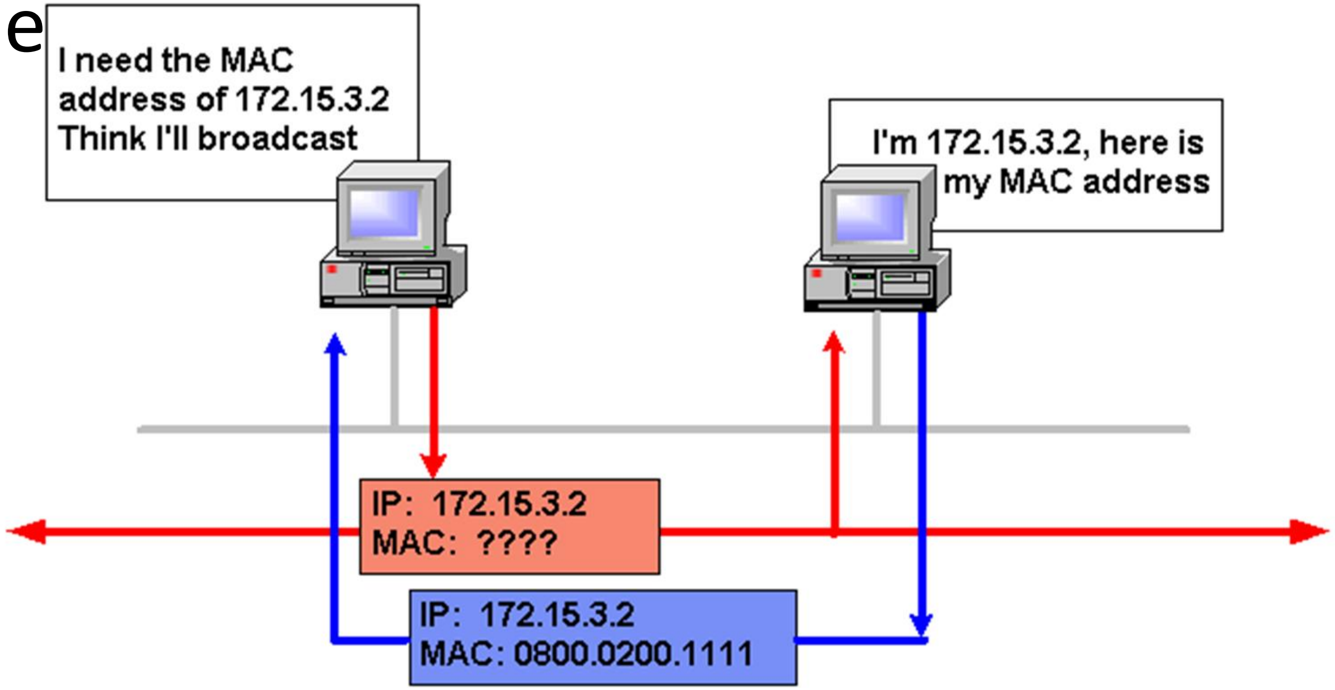
IPv6 Provider-Based Addresses

- The first IPv6 addresses will be allocated to a provider-based plan
- Type: Set to "010" for provider-based addresses
- Registry: identifies the agency that registered the address
- Fields have a variable length (recommeded length in "()")
- Provider: Id of Internet access provider (16 bits)
- Subscriber: Id of the organization at provider (24 bits)
- Subnetwork: Id of subnet within organization (32 bits)
- Interface: identifies an interface at a node (48 bits)



ARP

- Address Resolution Protocol
- Map layer 3 addresses and layer 2 addresses
- MAC address 48 bit unique in the LAN.
- Hardware addresses are not computable
- Used static table


I need the MAC address of 172.15.3.2 Think I'll broadcast
I'm 172.15.3.2, here is my MAC address
IP: 172.15.3.2
MAC: ????
IP: 172.15.3.2
MAC: 0800.0200.1111

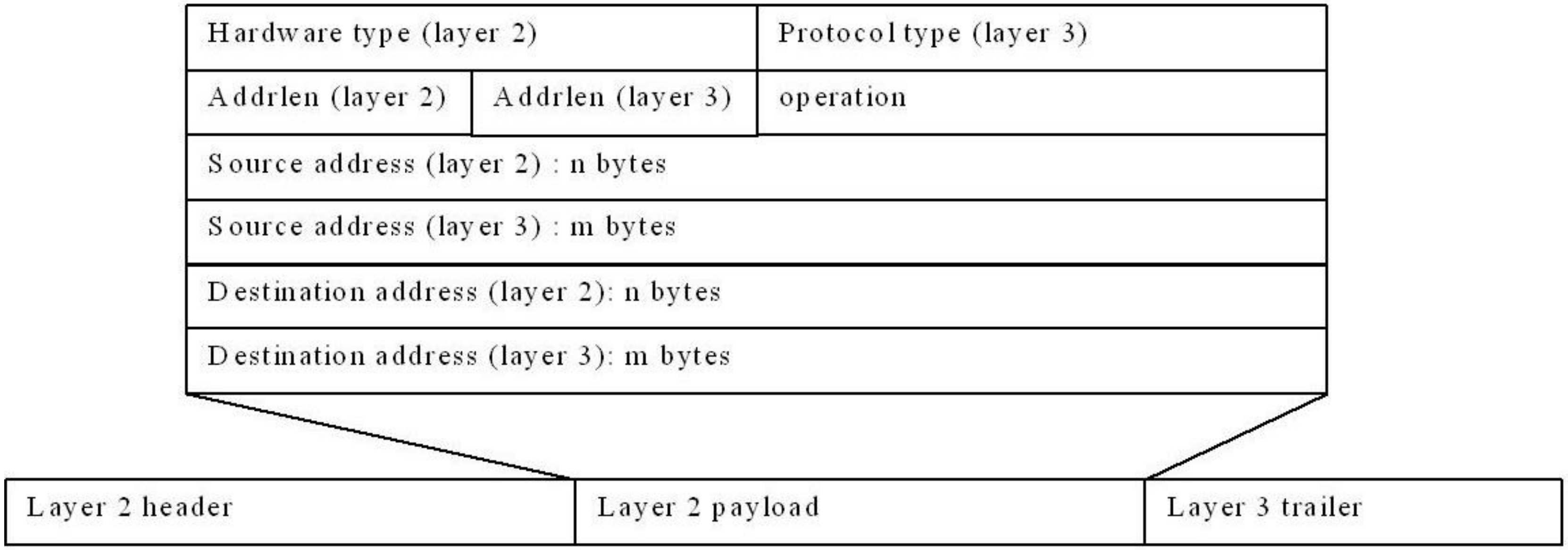
Arp Cache

- Sending ARP request and wait for response is inefficient
- More bandwidth / time
- ARP cache maintained at each node
- Size limit = 512 entries
- States of a neighbour entry in ARP Cache
- Cache space may be limited
- Hosts move or change IP addresses
- Drop entries after 20 minutes

Arp Command

- To display table
  - ip neigh show
- To enter manually (Static Entry)
  - sudo ip neigh add 10.0.2.10 lladdr 52:54:00:12:35:10 dev enp0s3
- To delete entry
  - sudo ip neigh delete 10.0.2.10 dev enp0s3

Structure of Arp Protocol



Hardware type (layer 2)
Protocol type (layer 3)
Addrlen (layer 2)
Addrlen (layer 3)
operation
Source address (layer 2) : n bytes
Source address (layer 3) : m bytes
Destination address (layer 2): n bytes
Destination address (layer 3): m bytes
Layer 2 header
Layer 2 payload
Layer 3 trailer

ARP Request and Reply PDU

ARP request to FF:FF:FF:FF:FF:FF

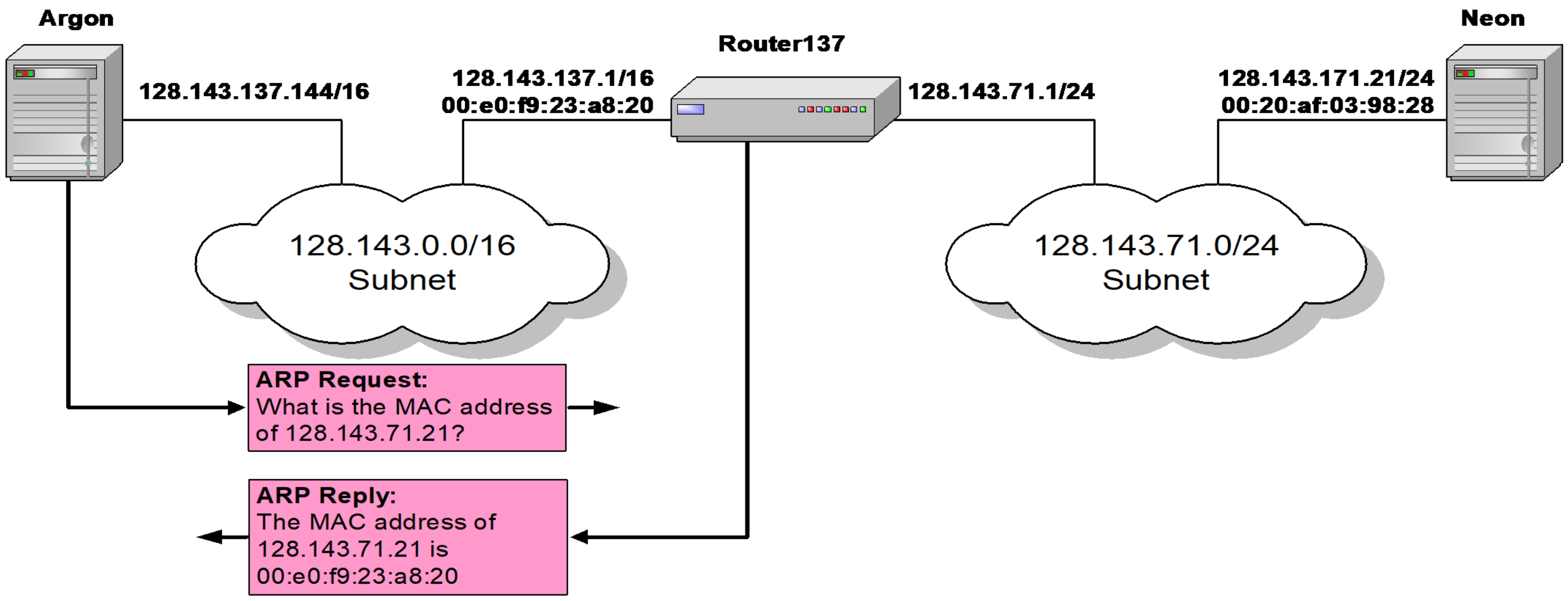
0	15	31
0x00 01 (*Ethernet*)		0x80 00 (*Internet Protocol*)
6	4	0x00 01 (*ARP request*)
49 72 16 08		
64 14		129 25
10 72		00 00
00 00 00 00		
129 25 10 11		

ARP reply to 49:72:16:08:64:14

0	15	31
0x00 01 (*Ethernet*)		0x80 00 (*Internet Protocol*)
6	4	0x00 02 (*ARP reply*)
49 72 16 08		
64 14		129 25
10 72		49 78
21 21 23 90		
129 25 10 11		

Proxy ARP

- Responds to ARP Request from one of its connected networks for a host on another



Things to know about ARP

- What happens if an ARP Request is made for a non-existing host?
  - Several ARP requests are made with increasing time intervals between requests.
  - Eventually, ARP gives up.
- A host periodically sends ARP Requests for all addresses listed in the ARP cache
  - This refreshes the ARP cache content, but also introduces traffic
- Gratuitous ARP Requests
  - A host sends an ARP request for its own IP address
  - Useful for detecting if an IP address has already been assigned

Vulnerabilities of ARP

- No authentication: Requests and Replies can be forged
- ARP is stateless: Replies sent without an ARP Request
- A node receiving an ARP packet MUST update its local ARP cache
- ARP Poisoning: a forged ARP Request or Reply used to update the ARP cache of a remote system
    - This can be used to redirect IP traffic to other hosts

How to diagnose a problem ?

- ping the remote IP address
  - If ping prints replies, the connection is OK
- “ip -a address” (Linux) or “ipconfig /all” (Windows)
  - check TCP/IP settings
- Ping another host on the network
  - If it does not work than your host might have a problem
- Check the “link” indicator light for each PC on the UTP hub
  - If the light is off, you have a cable problem to fix
- "ping 127.0.0.1" (loopback address)
  - If ping doesn’t print replies, the TCP/IP installation on this PC is broken

How to diagnose a problem ?

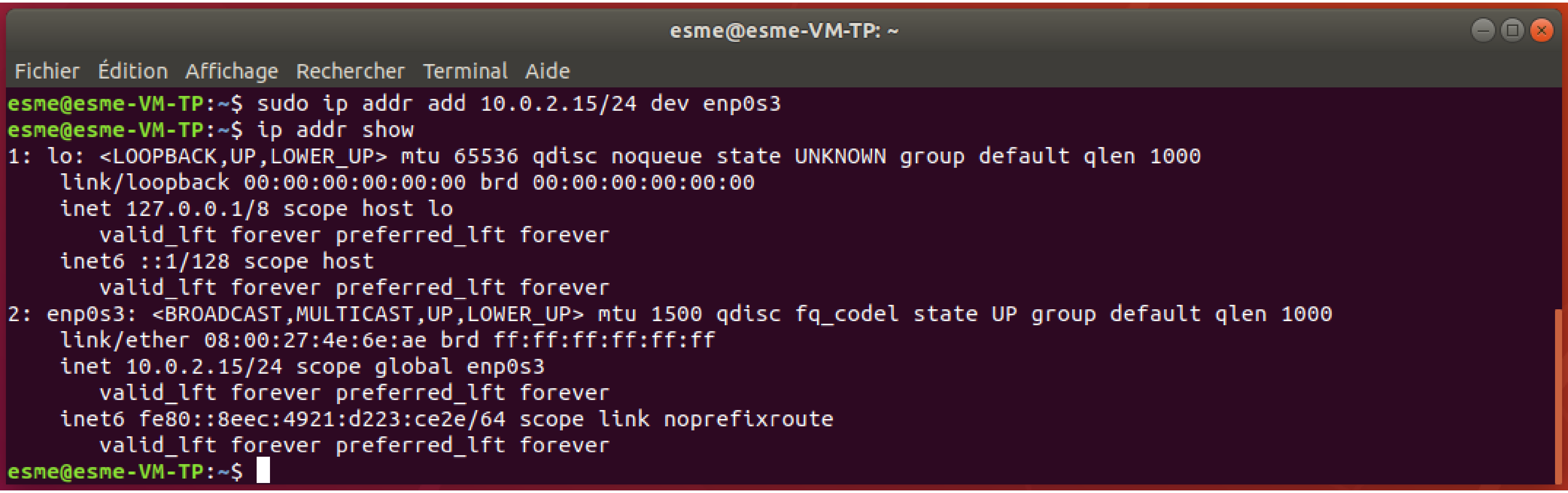
- ping local (non-loopback) IP address
  - If ping doesn't print replies,
    - either your TCP/IP is broken,
    - or you need to reboot,
    - or this machine's IP number is different from what you think.
  - tcpdump shows packets on the wire => wrong address
    - because packets from a machine to itself never appear on the wire
- ping remote address
  - If tcpdump doesn't show an ARP "who-has 10.9.8.2" on the wire, then:
    - run "ip -a neigh show" (Linux) or "arp -a" (Windows) and see whether this machine has an ARP cache entry for remote address
    - If it has, delete it using "sudo ip neigh delete 10.9.8.2 dev enp0s3"

How to diagnose a problem ?

- ping 10.9.8.2 again
  - If tcpdump doesn’t show an ARP “who-has 10.9.8.2”
    - then this is not on the same network as 10.9.8.2
    - Check the machine’s IP address and netmask
  - If tcpdump showed an ARP “who-has” for a different IP address
    - 10.9.8.2 is on a different network and this machine is a router
    - clear the default gateway setting for this machine

Interface Configuration

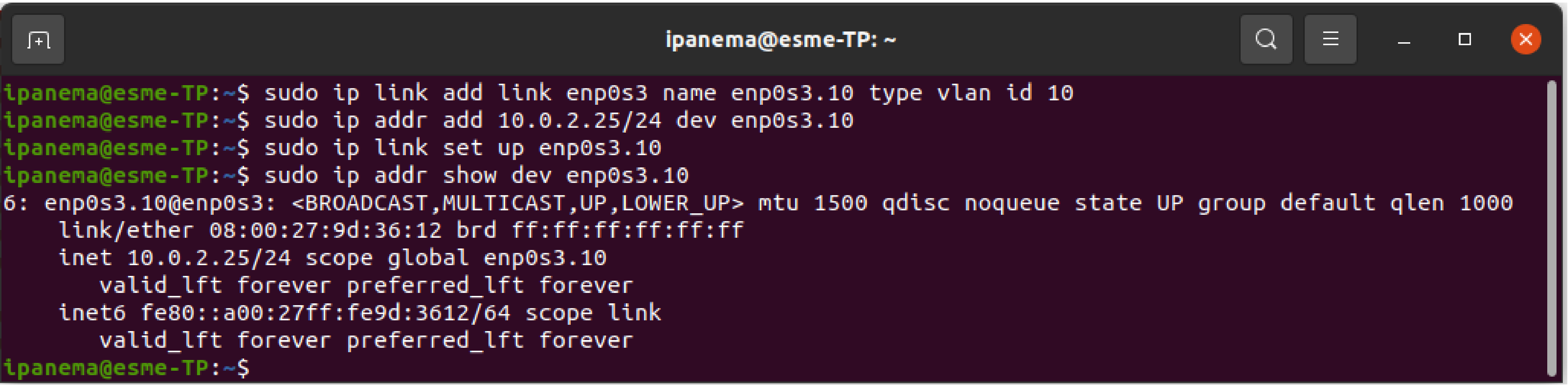
- Add an ip address
- Show configured addresses



```
esme@esme-VM-TP:~$ sudo ip addr add 10.0.2.15/24 dev enp0s3
esme@esme-VM-TP:~$ ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:4e:6e:ae brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 scope global enp0s3
       valid_lft forever preferred_lft forever
    inet6 fe80::8eec:4921:d223:ce2e/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
esme@esme-VM-TP:~$
```

VLAN Interface Configuration

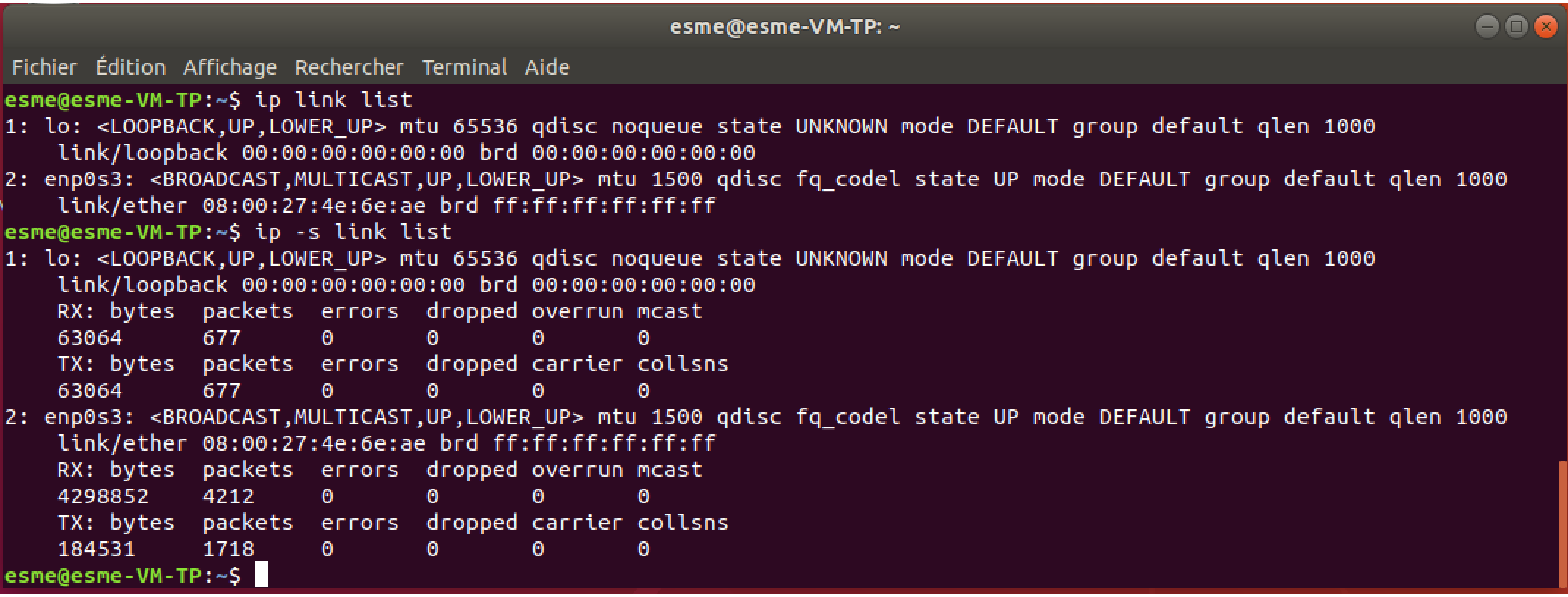
- VLANs is used to divide a network
  - Separate hosts that shouldn't be able to access each other
- A VLAN is assigned a specific id [1..4094]
- Requires 802.1q support
  - Load 8021q module



```
ipanema@esme-TP:~$ sudo ip link add link enp0s3 name enp0s3.10 type vlan id 10
ipanema@esme-TP:~$ sudo ip addr add 10.0.2.25/24 dev enp0s3.10
ipanema@esme-TP:~$ sudo ip link set up enp0s3.10
ipanema@esme-TP:~$ sudo ip addr show dev enp0s3.10
6: enp0s3.10@enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether 08:00:27:9d:36:12 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.25/24 scope global enp0s3.10
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe9d:3612/64 scope link
       valid_lft forever preferred_lft forever
ipanema@esme-TP:~$
```

Interface Configuration

- List interfaces
- Show intefaces' statistics



```
esme@esme-VM-TP:~$ ip link list
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP mode DEFAULT group default qlen 1000
    link/ether 08:00:27:4e:6e:ae brd ff:ff:ff:ff:ff:ff
esme@esme-VM-TP:~$ ip -s link list
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    RX: bytes  packets  errors  dropped overrun mcast
    63064      677      0       0       0       0
    TX: bytes  packets  errors  dropped carrier collsns
    63064      677      0       0       0       0
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP mode DEFAULT group default qlen 1000
    link/ether 08:00:27:4e:6e:ae brd ff:ff:ff:ff:ff:ff
    RX: bytes  packets  errors  dropped overrun mcast
    4298852    4212     0       0       0       0
    TX: bytes  packets  errors  dropped carrier collsns
    184531     1718     0       0       0       0
esme@esme-VM-TP:~$
```

Interface Configuration

- To bring the interface eth1 down or up via ifconfig command

```
$ ip link set eth1 up
$ ip link set eth1 down

$ ifconfig eth1 down
$ ifconfig eth1 up
```

Permanent Configuration

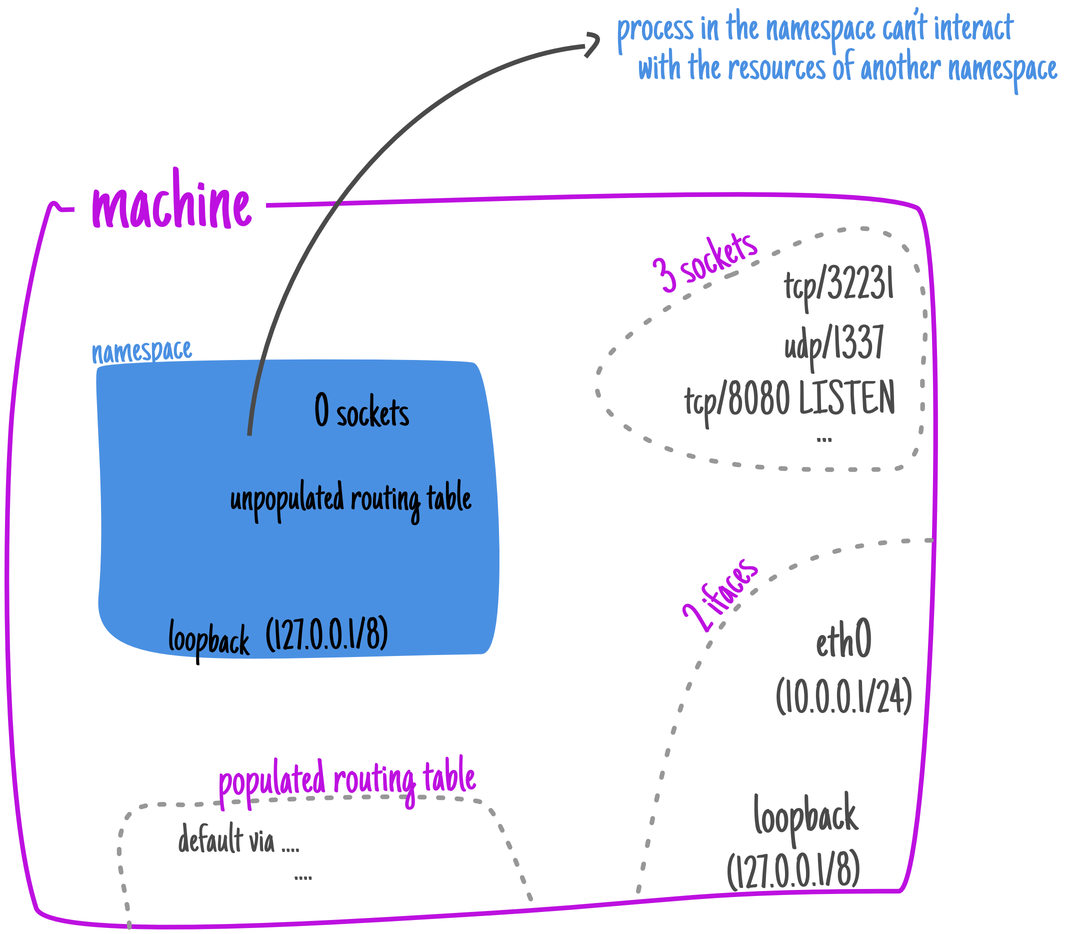
- Add the interface in /etc/network/interfaces

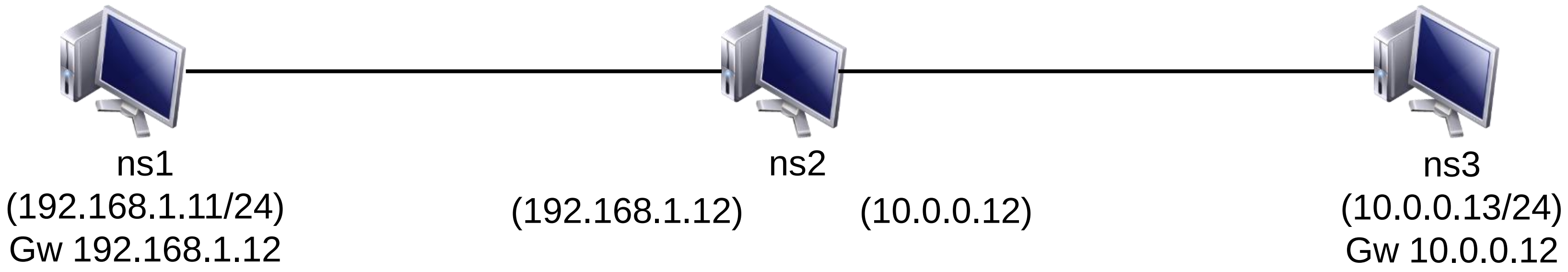
```
auto eth0
iface eth0 inet static
    address 192.168.10.2
    netmask 255.255.255.0
    gateway 192.168.10.1
```

- Restart networking service
  - sudo /etc/init.d/networking restart

Network namespaces example

- Different and separate instances of network interfaces and routing tables
- Script netns.sh
  - Sudo ip netns commands





Lab–software configuration: Linux

- What happens when ?

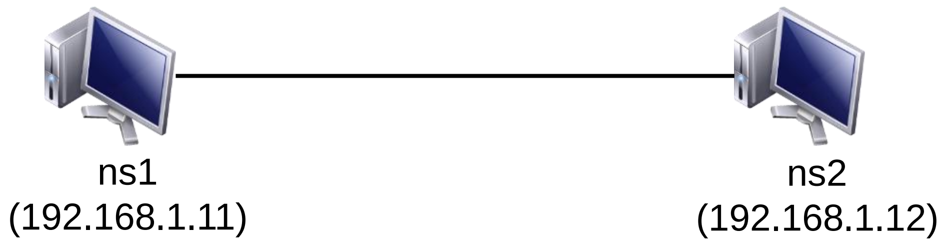
  1) ns1 have the same IP address as ns2 (192.168.1.12)

  2) Change ns2's IP address to 192.168.1.11

  3) Change ns1's IP address to 192.168.3.11

  4) Change ns2's IP address to 192.168.3.12

  5) Change ns1's netmask to 255.255.0.0, and its address to 192.168.66.12



Networking basics

Basic Routing

- Moving packets from one network to another network using routers
- A routing protocol is used by routers to
  - dynamically find all the networks in the internetwork
  - Ex: RIP, OSPF
- Routers only care about the best path to each network
  - Destination Network address is used route packets
  - Hardware address is used on the LAN

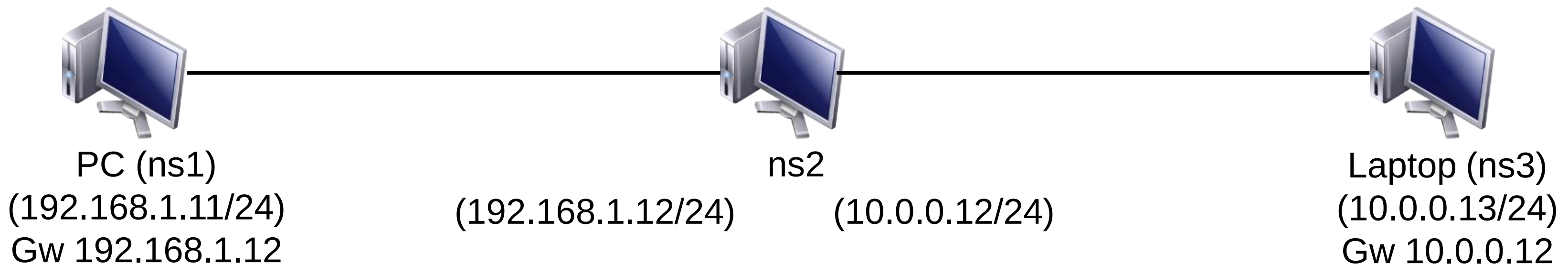
What should router knows ?

- Destination address
- Neighbor routers from which it can learn about remote networks
- Possible routes to all remote networks
- The best route to each remote network
- Learn about remote networks
  - From neighbor routers or from an administrator
  - If a network is directly connected, then the router already knows how to get to it

Static and Dynamic Routing

- Static routing
  - someone hand-type all network locations into the routing table
- Dynamic routing
  - a protocol on one router communicates with the same protocol running on neighbor routers
- The routers then update all the networks they know about
  - place this information into the routing table
- If a change occurs in the network
  - routing protocols automatically inform all routers about the event

IP Routing Process



- PC want to ping the laptop
- IP determines whether the destination address is local or remote
- a remote request, the packet needs to be sent to the default gateway

The router received the packet

- Every device within the LAN receives these bits and builds the frame
- Check the frame check sequence (FCS)
- Check the hardware address
- If it matches
    - the Ether-Type field is the protocol used at the network layer
- The packet is pulled from the frame
- the packet is then handed to IP

Router routes the packet

- IP receives the packet and checks the IP destination address
- It is not router IP, check routing table
- The routing table must have an entry for the destination network
- Otherwise discarded, “network unreachable message”

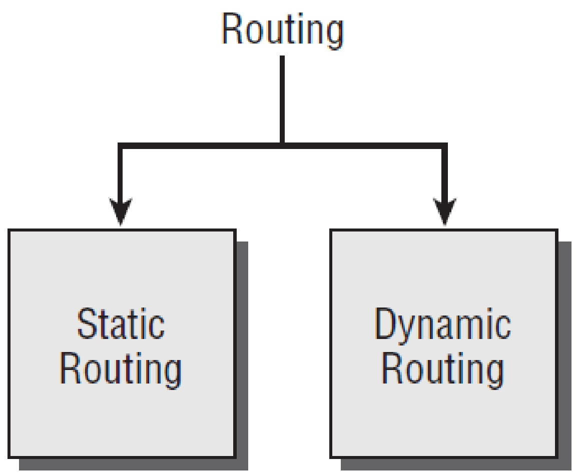
At the destination

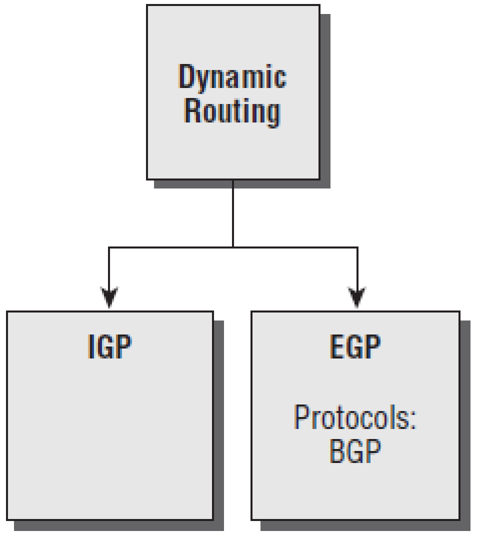
- The destination host receives the frame, and immediately runs a CRC
- If the result matches what's in the FCS field
  - the hardware-destination address is then checked
- If the host finds a match
  - the Ether-Type field is then checked, give to IP

Static and Dynamic Routing

- Packet not listed in the routing table?
  - Packet is just discarded

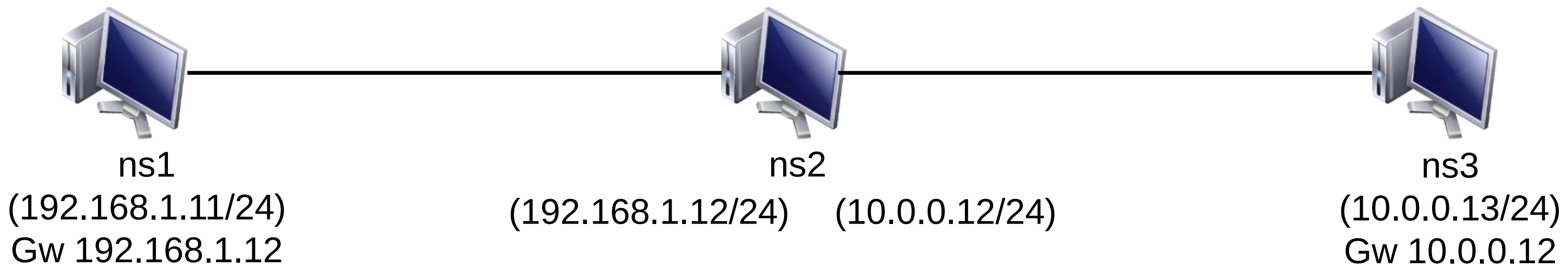
- IGP routes within an AS
- EGP routes between more than one AS





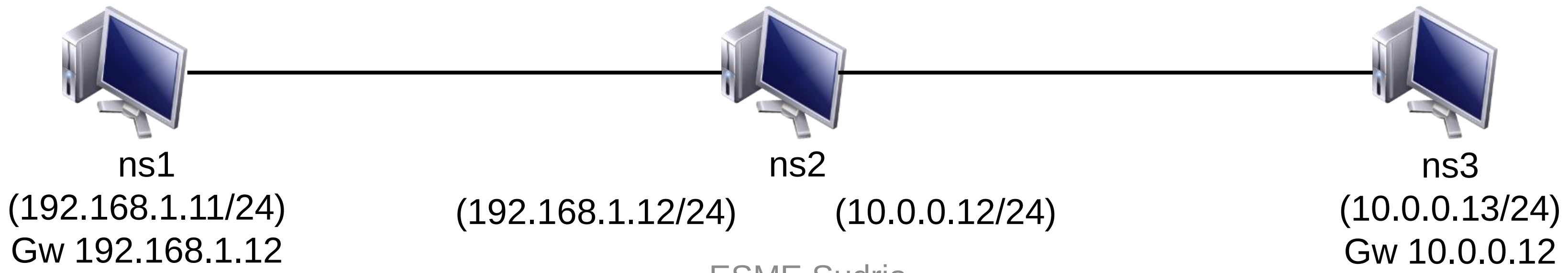
Lab : Setting your default gateway

- To display the current default gateway setting:
  - $ ip route
- To set the default gateway:
  - $ ip route add default via 192.168.1.12
- Try now a wrong gateway address, what happens ?
  - $ ip route add default via 1.2.3.4



Lab : building a simple internetwork

- Configuring multiple network cards
  - sudo ip addr add 192.168.1.12/24 dev vethrc
- To configure Linux to act as a router
  - sudo sysctl -w net.ipv4.ip_forward=1
- Test your network :
  - ip netns exec ns1 ping -c 4 192.168.1.12
- Trace packets across the router
  - sudo tcpdump -n -t -e -l



Networking basics

Network Devices in Linux

- Linux design is based on the concept of devices
  - Devices are accessed via the /dev directory
  - A driver is software that communicates with the device
- Networking devices are handled by the kernel
- Examples
  - Eth0: first ethernet interface
  - Ppp0: point to point protocol interface

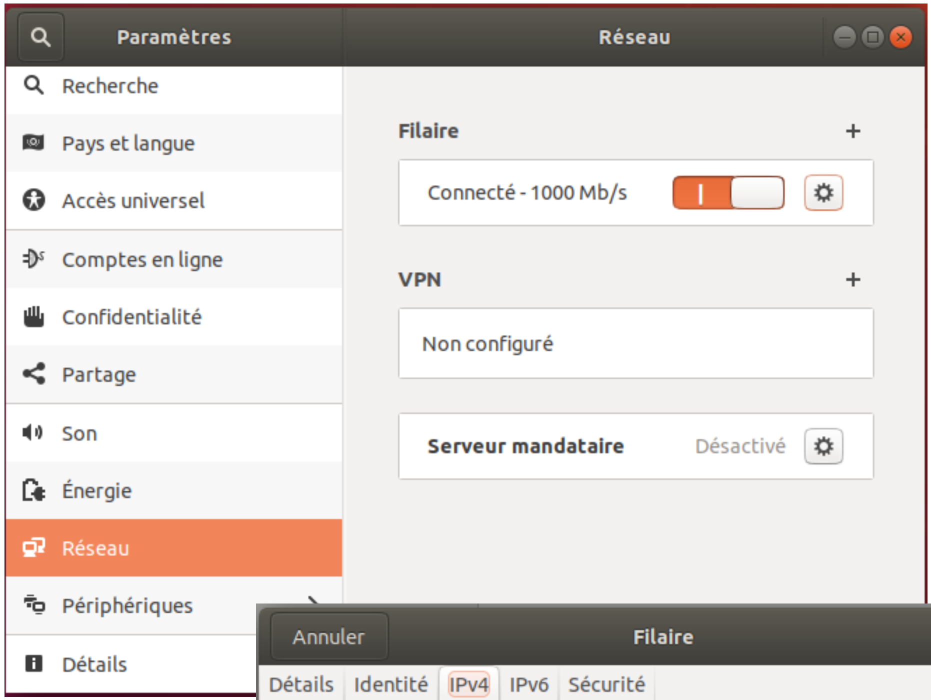
Preparing to Configure Networking

- To create a networking device
  - Add the appropriate module to the Linux kernel
  - The module creates the appropriate device name
  - Networking device kernel modules are in /lib/modules
  - Use **modprobe** command to load the networking device
  - Use **lsmod** command to list the modules loaded
- Exercise
  - Use lsmod and modinfo to find the network module?
    - lsmod
    - modinfo rt2500 -F description

System Networking Scripts

- Scripts and configuration files to manage networking
  - Scripts follow the model used for most system services on UNIX-based computers
  - Scripts are found in the /etc/sysconfig/network-scripts (Centos)
  - Configuration files are found in
    - Centos : /etc/sysconfig/networking subdirectory
    - Ubuntu: /etc/network/interfaces

Configuring with Graphical Tools



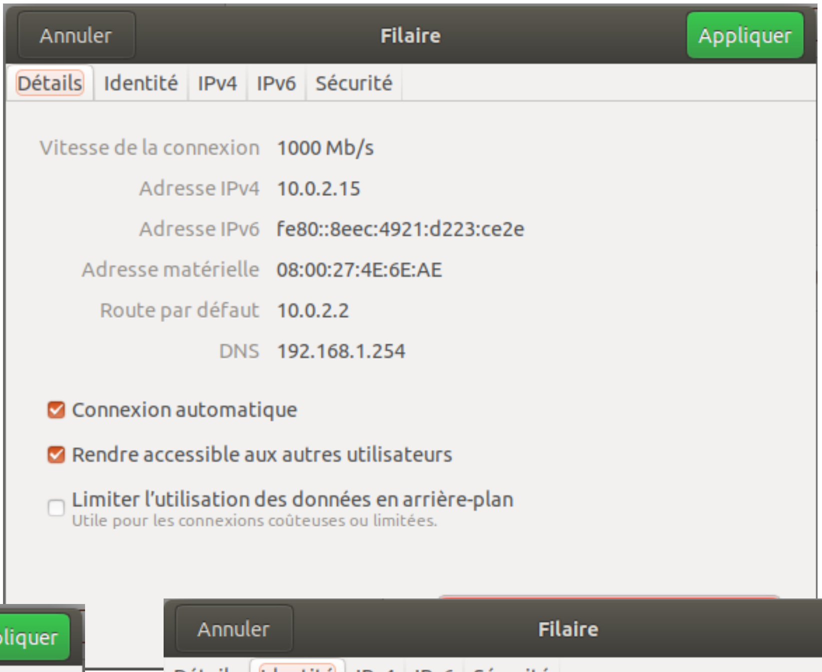
Filaire

Connecté - 1000 Mb/s

VPN

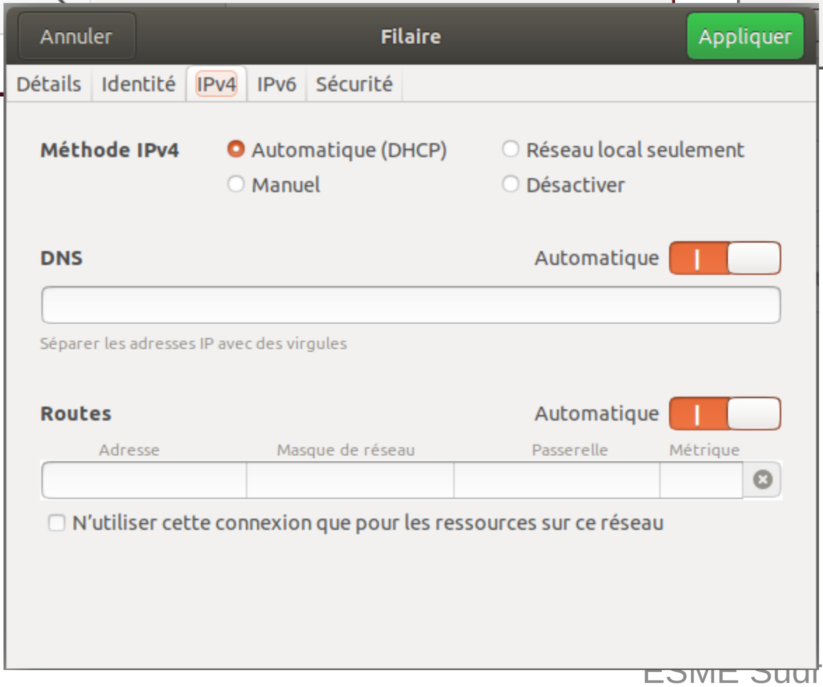
Non configuré

Serveur mandataire Désactivé



Vitesse de la connexion	1000 Mb/s
Adresse IPv4	10.0.2.15
Adresse IPv6	fe80::8eec:4921:d223:ce2e
Adresse matérielle	08:00:27:4E:6E:AE
Route par défaut	10.0.2.2
DNS	192.168.1.254

- [✓] Connexion automatique
- [✓] Rendre accessible aux autres utilisateurs
- [ ] Limiter l'utilisation des données en arrière-plan
  Utile pour les connexions coûteuses ou limitées.



Méthode IPv4: [● Automatique (DHCP)] ○ Réseau local seulement ○ Manuel ○ Désactiver

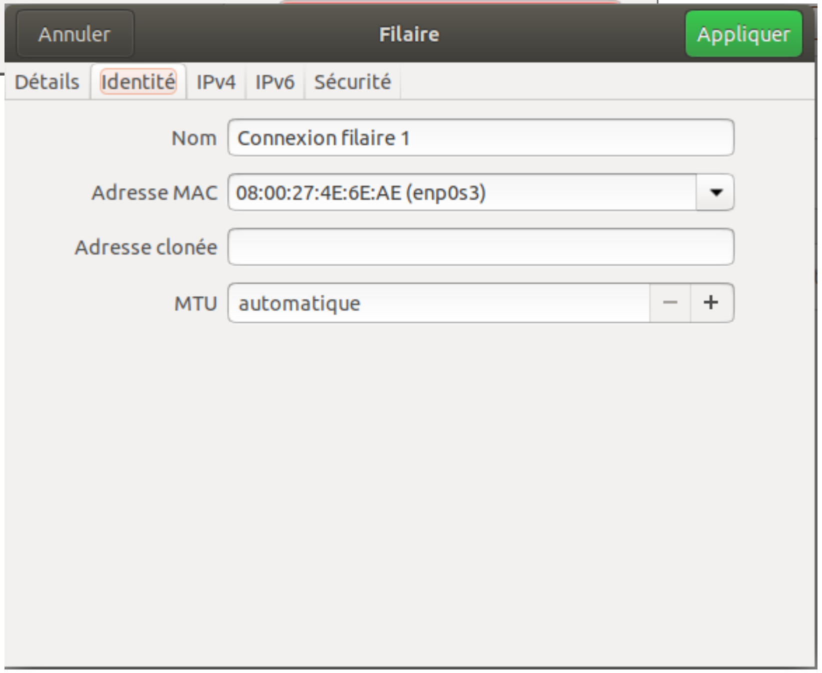
DNS — Automatique: activé

Séparer les adresses IP avec des virgules

Routes — Automatique: activé

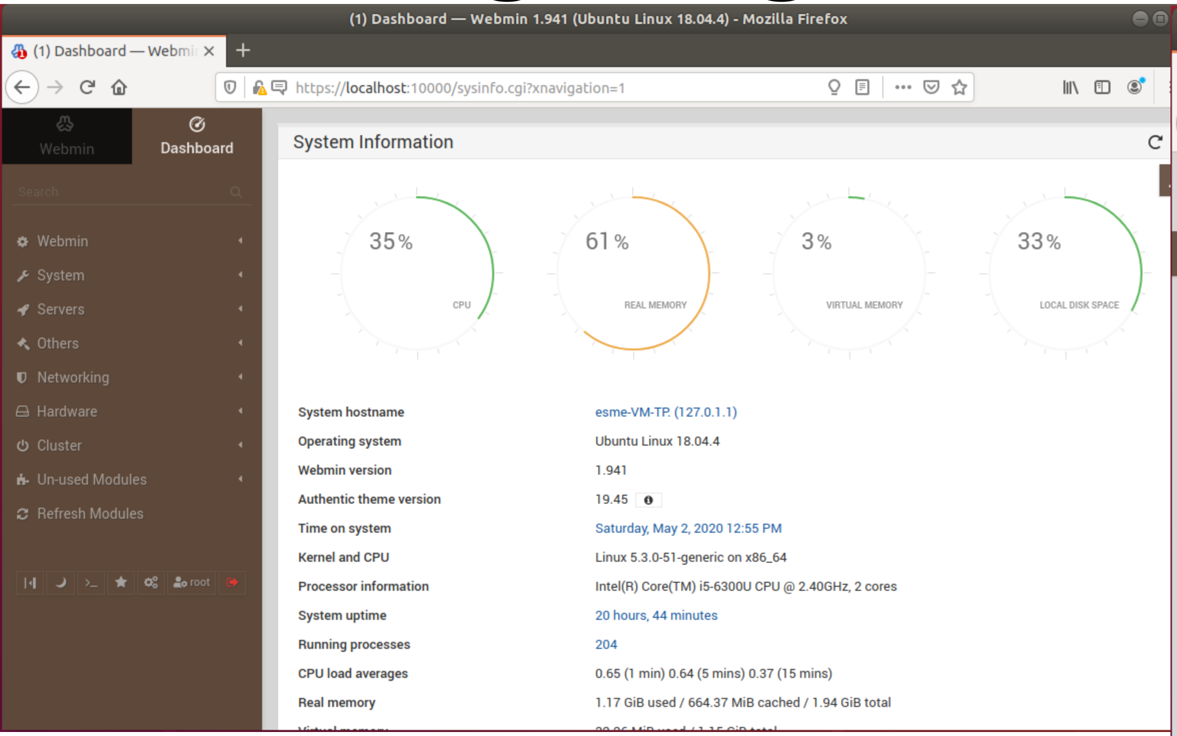
Adresse	Masque de réseau	Passerelle	Métrique

- [ ] N'utiliser cette connexion que pour les ressources sur ce réseau



Nom	Connexion filaire 1
Adresse MAC	08:00:27:4E:6E:AE (enp0s3)
Adresse clonée	
MTU	automatique

Configuring with Graphical Tools

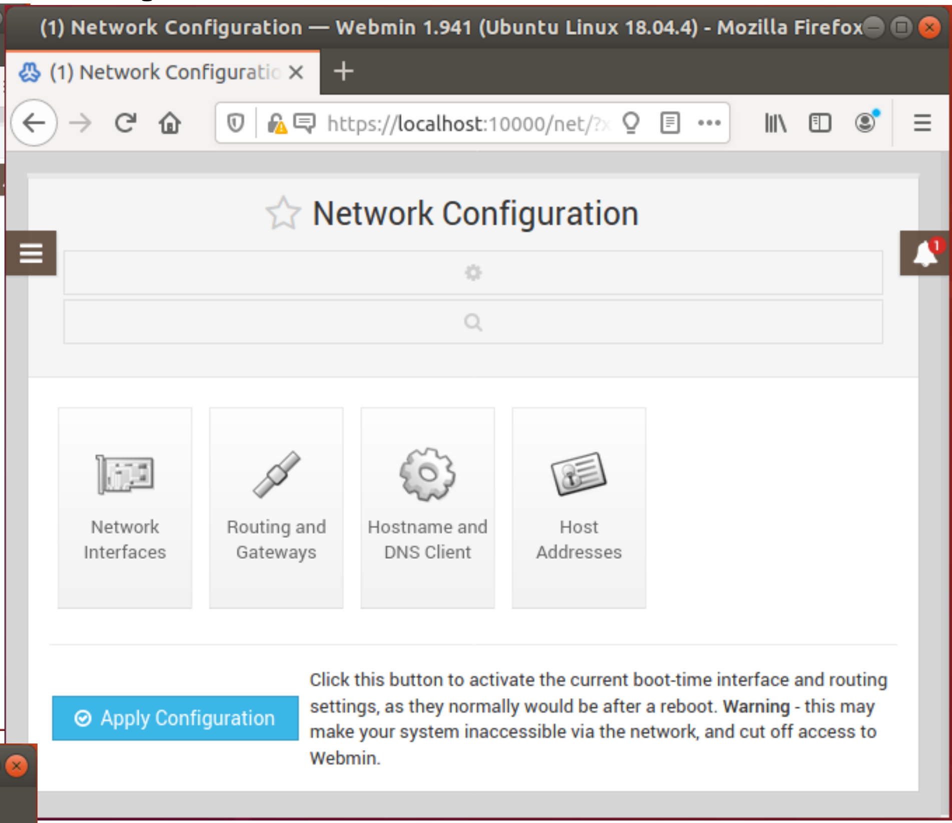

(1) Dashboard — Webmin 1.941 (Ubuntu Linux 18.04.4) - Mozilla Firefox

https://localhost:10000/sysinfo.cgi?xnavigation=1

System Information

35 % CPU · 61 % REAL MEMORY · 3 % VIRTUAL MEMORY · 33 % LOCAL DISK SPACE

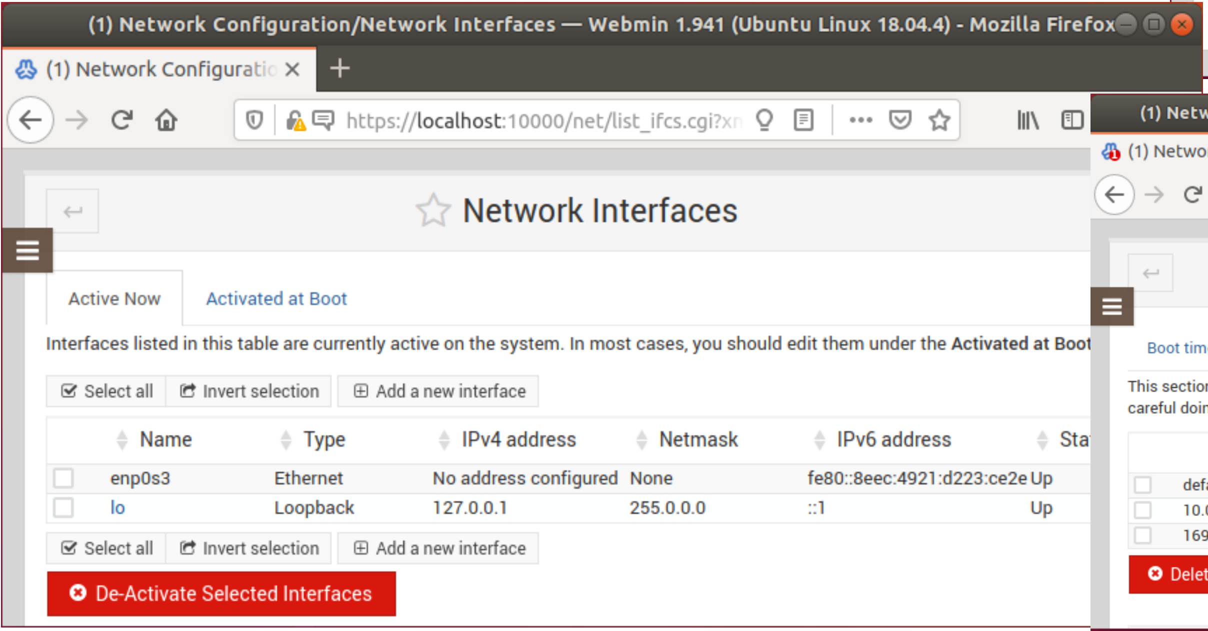
System hostname	esme-VM-TP (127.0.1.1)
Operating system	Ubuntu Linux 18.04.4
Webmin version	1.941
Authentic theme version	19.45
Time on system	Saturday, May 2, 2020 12:55 PM
Kernel and CPU	Linux 5.3.0-51-generic on x86_64
Processor information	Intel(R) Core(TM) i5-6300U CPU @ 2.40GHz, 2 cores
System uptime	20 hours, 44 minutes
Running processes	204
CPU load averages	0.65 (1 min) 0.64 (5 mins) 0.37 (15 mins)
Real memory	1.17 GiB used / 664.37 MiB cached / 1.94 GiB total


(1) Network Configuration — Webmin 1.941 (Ubuntu Linux 18.04.4) - Mozilla Firefox

https://localhost:10000/net/?x

Network Configuration

Click this button to activate the current boot-time interface and routing settings, as they normally would be after a reboot. **Warning** - this may make your system inaccessible via the network, and cut off access to Webmin.

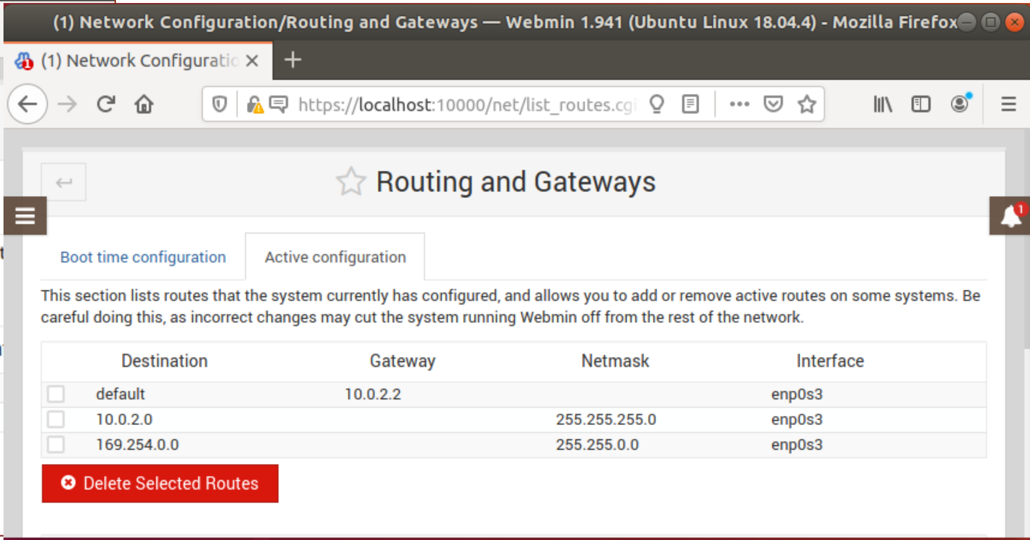

(1) Network Configuration/Network Interfaces — Webmin 1.941 (Ubuntu Linux 18.04.4) - Mozilla Firefox

https://localhost:10000/net/list_ifcs.cgi?xn

Network Interfaces

Interfaces listed in this table are currently active on the system. In most cases, you should edit them under the **Activated at Boot**

Name	Type	IPv4 address	Netmask	IPv6 address	Sta
enp0s3	Ethernet	No address configured	None	fe80::8eec:4921:d223:ce2e	Up
lo	Loopback	127.0.0.1	255.0.0.0	::1	Up


(1) Network Configuration/Routing and Gateways — Webmin 1.941 (Ubuntu Linux 18.04.4) - Mozilla Firefox

https://localhost:10000/net/list_routes.cgi

Routing and Gateways

This section lists routes that the system currently has configured, and allows you to add or remove active routes on some systems. Be careful doing this, as incorrect changes may cut the system running Webmin off from the rest of the network.

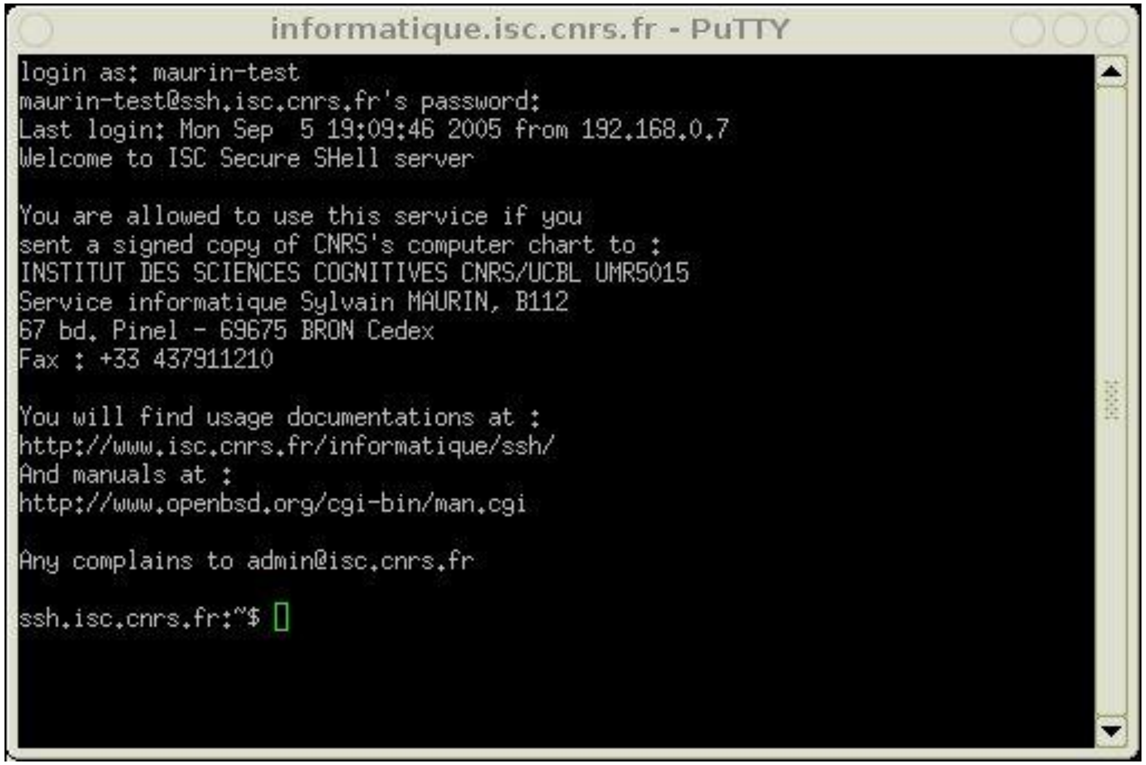
Destination	Gateway	Netmask	Interface
default	10.0.2.2		enp0s3
10.0.2.0		255.255.255.0	enp0s3
169.254.0.0		255.255.0.0	enp0s3

Using Basic Networking Utilities

- The Telnet Remote Login Utility

–As if present at the computer itself

- Telnet is not secure, use ssh instead



```
login as: maurin-test
maurin-test@ssh.isc.cnrs.fr's password:
Last login: Mon Sep  5 19:09:46 2005 from 192.168.0.7
Welcome to ISC Secure SHell server

You are allowed to use this service if you
sent a signed copy of CNRS's computer chart to :
INSTITUT DES SCIENCES COGNITIVES CNRS/UCBL UMR5015
Service informatique Sylvain MAURIN, B112
67 bd. Pinel - 69675 BRON Cedex
Fax : +33 437911210

You will find usage documentations at :
http://www.isc.cnrs.fr/informatique/ssh/
And manuals at :
http://www.openbsd.org/cgi-bin/man.cgi

Any complains to admin@isc.cnrs.fr

ssh.isc.cnrs.fr:~$
```

Troubleshooting Network Connections

- Networking doesn’t appear to function at all
  - Use the ‘ip link’ command to see whether networking is up and running
    - if not, try the network script /etc/init.d
  - Remember to check things that seem too obvious :
    - Is the cable plugged into the Ethernet card?
    - Is an Ethernet card installed?
    - Is the cable plugged into the wall or the hub?
- The network script doesn’t appear to work
  - Use the lsmod command to see whether any network modules are installed in the kernel
  - If not, use modprobe –a or a graphical utility to install the right kernel module for your networking card.

Troubleshooting Network Connections

- Can’t ping any other systems on the local network segment
  - Check the cables;
  - Check with ‘ip addr’ whether a valid IP address has been assigned;
  - Check the routing tables to see that a route is listed for the local network.
- Can’t ping any system on another segment within the organization
  - Check the cables at the hub or server room;
  - Check with ‘ip addr’ to see if a valid IP address is assigned
    - and how that address compares with those of the other network segment (is there a conflict?)
  - Does the routing table include a route for the other network that refers to an intermediate router if necessary?

Troubleshooting Network Connections

- Traffic to another segment seems any slow
  - Check the routing table to see whether the most direct route is defined;
  - check whether another system has the same IP address assigned;
  - review the output of the ‘ip -s addr’ command to see whether a large number of collisions are occurring
    - if so the network may be overloaded ;

Troubleshooting Network Connections

- – check ‘ip –s addr’ output to see whether a large number of errors are occurring
  - if so the NIC may be defective;
- – review whether fragmentation problems between the two segments could be slowing down traffic;
- – use traceroute to see at what point along the way the transmission slows significantly.

Troubleshooting – netmasks and addresses

- Run tcpdump on the wire
  - So you can see exactly what's happening
- Remember that if you see Alice ARP for Bob
  - She thinks he's connected directly to her LAN
- Draw a diagram of your network
  - Write in the values of the IP addresses, netmasks, routes, and default gateways
  - Check them off one by one against the real values

Troubleshooting – netmasks and addresses

- Symptom 1: ARPs for addresses that are not on this LAN instead of using the default gateway
  - Cause: sub-net mask is too broad – the range of directly connected IP addresses is too big
  - Try it on ns1: IP address = 192.168.1.11, netmask 128.0.0.0, no default gateway
    - Half of all the IP addresses in the world are connected directly to you!

Troubleshooting – netmasks and addresses

- Symptom 2: on the LAN, a machine ARPs for another on the same LAN but gets no ARP reply
  - Cause: (trivial) you typed the wrong IP address,
    - the machine doesn't exist
  - Cause: my subnet mask isn't the same as yours
    - you can receive my packets but can't send to me
  - Try it:
    - ns1: IP 192.168.1.11 / 255.255.255.0, no default gateway
    - ns2: IP 192.168.1.12 / 255.255.255.254, no default gateway
    - ping ns1 from ns2 and vice versa

Troubleshooting – netmasks and addresses

- Symptom 3: on the LAN, a machine can communicate with some machines but not others, whereas all the others can communicate
  - Cause: a wrong netmask on the machine
  - Try it:
    - ns1 and ns2 as above
    - ns4: IP 192.168.1.6/255.255.255.0, no default gateway
    - ns5: IP 192.168.1.233/255.255.255.0, no default gateway

Troubleshooting – netmasks and addresses

- Symptom 4: over the Internet, you are trying to communicate with a remote machine. Your packets are coming in, but you are not getting any replies
  - Cause: (one of many possibles) the remote machine has no default gateway set
    - Machines that are accessed only from the LAN don’t need default gateways

Troubleshooting – netmasks and addresses

- Treat the packet’s path as a series of individual hops
- Draw a diagram of the network
- Verify each hop:
  - treat each hop as a single machine-to-machine link
  - remember the three fundamental networking parameters – IP address, netmask, and default gateway
  - check this machine’s routing tables to see whether it has been configured to use any other router
  - annotate your diagram as you check each hop

Troubleshooting – netmasks and addresses

- At the source machine, at every router
  - see which next-hop router is being used
    - by looking at the destination MAC address in the Ethernet frame
  - At each point, if the packet is being forwarded to somewhere you didn’t expect
    - check out the routing tables on this machine

Troubleshooting – netmasks and addresses

- Symptom: you can see (ping) the near-side interface of a router, and the far-side interface, but nothing beyond it
  - Cause: the link onwards from that router is dead
  - Cause: you haven't configured the box to act as a router
    - You can see the far-side interface because it doesn't involve packet forwarding
    - even though that IP address belongs to the far-side network

Troubleshooting – netmasks and addresses

- Try it: draw the network diagram of this configuration :
  - ns1: IP 192.168.1.11/255.255.255.0, default gateway = 10.1.1.254
  - ns3: IP 10.0.0.13/24, default gateway =192.168.168.254
  - ns2: IP address-1= 10.0.0.12/24 IP address-2= 192.168.1.12/24 No default gateway. Not configured as a router.
  - From ns1, ping all other addresses are they working ?

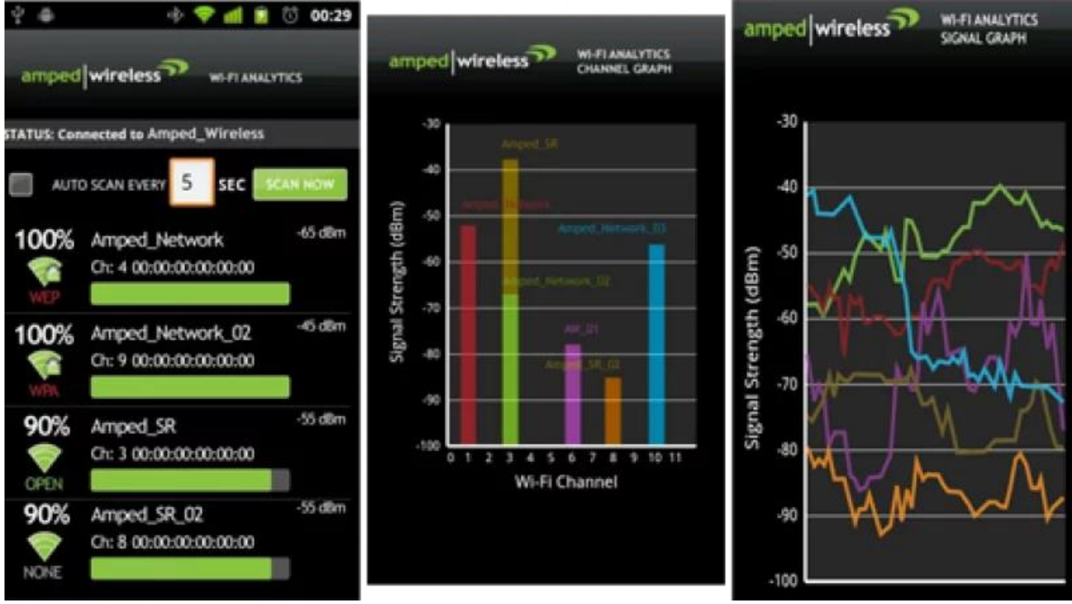
Troubleshooting – netmasks and addresses

- Symptom: you get the error message “no route to host” when you run telnet or ftp, etc. on Linux
  - Cause: (trivial) the IP address you’re connecting to doesn’t exist.
    - The error message is very misleading: it has nothing to do with routing
    - and really means your machine can’t ARP the destination
  - Try it:
    - Ftp 10.1.1.188, where 10.1.1.188 is an IP address that doesn’t exist on your LAN. On Linux this gives:
    - ftp: connect: No route to host
    - and running telnet 10.1.1.188, from Linux gives :
    - telnet: Unable to connect to remote host: No route to host

Wifi Troubleshooting

- Wifi interference usually use 2.4 gigahertz
  - though they can affect 5 gigahertz networks also.
- 2.4GHs is especially susceptible
  - it only has 3 non-overlapping 20 megahertz channels
- 802.11n have 2 channels to work with
- to send information, an interface will listen to ensure the channel is quiet
- On a collision, it will wait a random amount of time

Wifi Troubleshooting


00:29
amped|wireless WI-FI ANALYTICS
STATUS: Connected to Amped_Wireless
AUTO SCAN EVERY 5 SEC SCAN NOW

100% WEP	Amped_Network Ch: 4 00:00:00:00:00:00	-65 dBm
100% WPA	Amped_Network_02 Ch: 9 00:00:00:00:00:00	-45 dBm
90% OPEN	Amped_SR Ch: 3 00:00:00:00:00:00	-55 dBm
90% NONE	Amped_SR_02 Ch: 8 00:00:00:00:00:00	-55 dBm

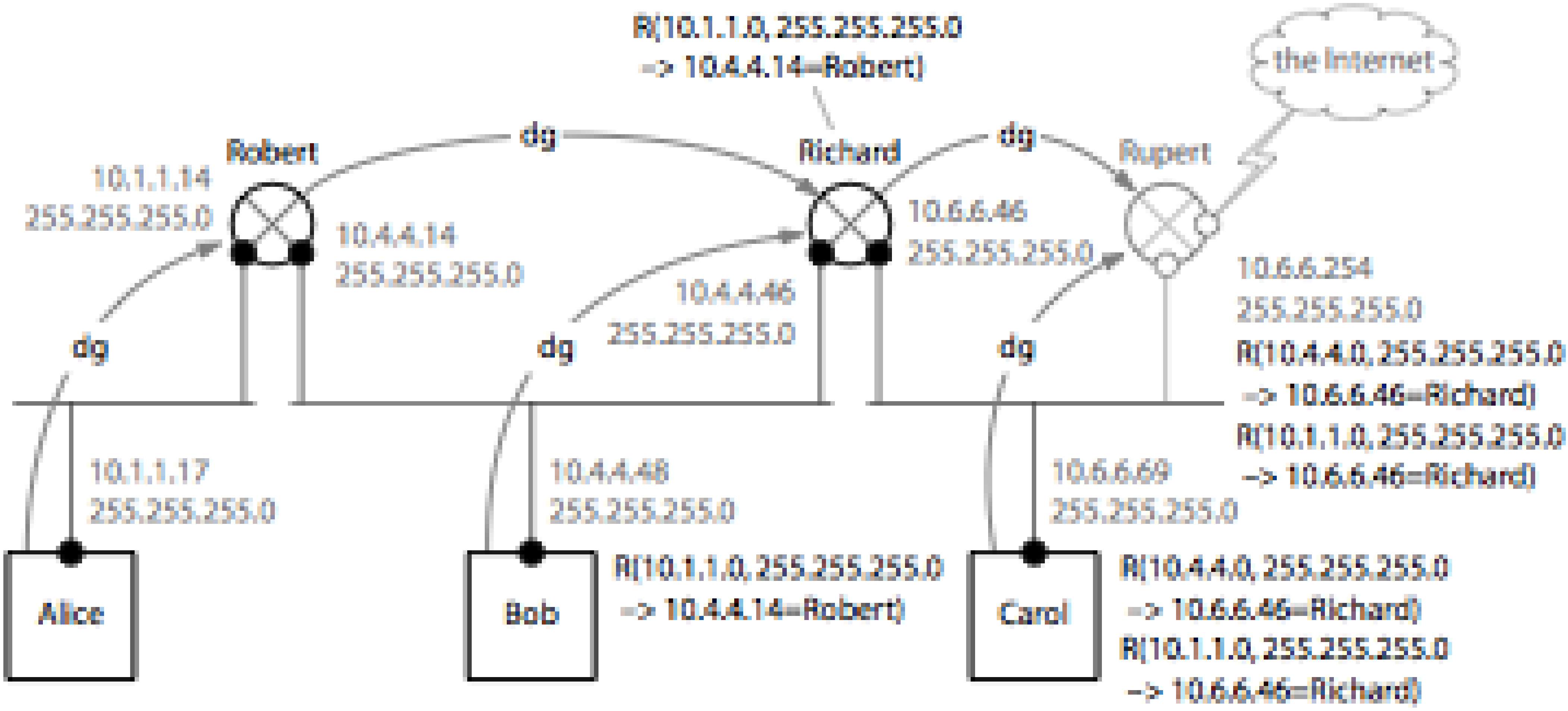
- Use a WiFi Analyzer to detect interferences
  - Display SSIDs, the channel they are using, and signal strength
- Switch the channel to the least utilized frequency
- A lot of devices use 2.4GHz : mics, baby monitors, etc.
  - If 2.4GHz is saturated, then switch to the 5GHz frequency
- Stay vigilant as some new source of interference may appear at any time

Wifi Troubleshooting: Linux

- Use nmcli to check if your wireless is connected
- Troubleshooting Steps
  1. Check that your wireless adapter is enabled
     - lshw -C network
  2. Check if drivers are available for your wireless adapter
     - Use lsmod and modprobe to load the driver
  3. Check your connection to the Internet
     - sudo iwconfig <ath0> essid <essid> ap <xx:xx:xx:xx:xx:> key <XXX> mode <> commit
  4. Check if your wireless device is disabled by hardware switch
     - enable your wireless device during the boot process

Networking basics

Lab–network w multiple routers



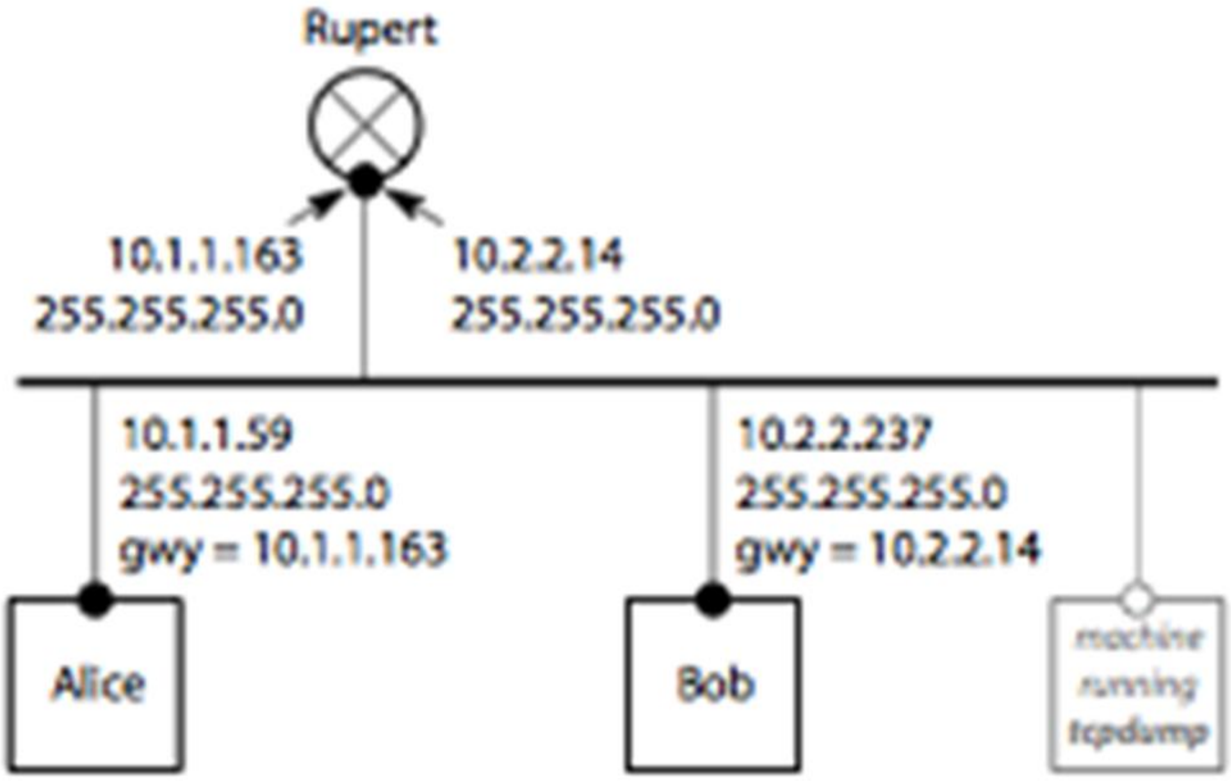
Lab–network w multiple routers

- Fill in the table with all routes from each host to each network

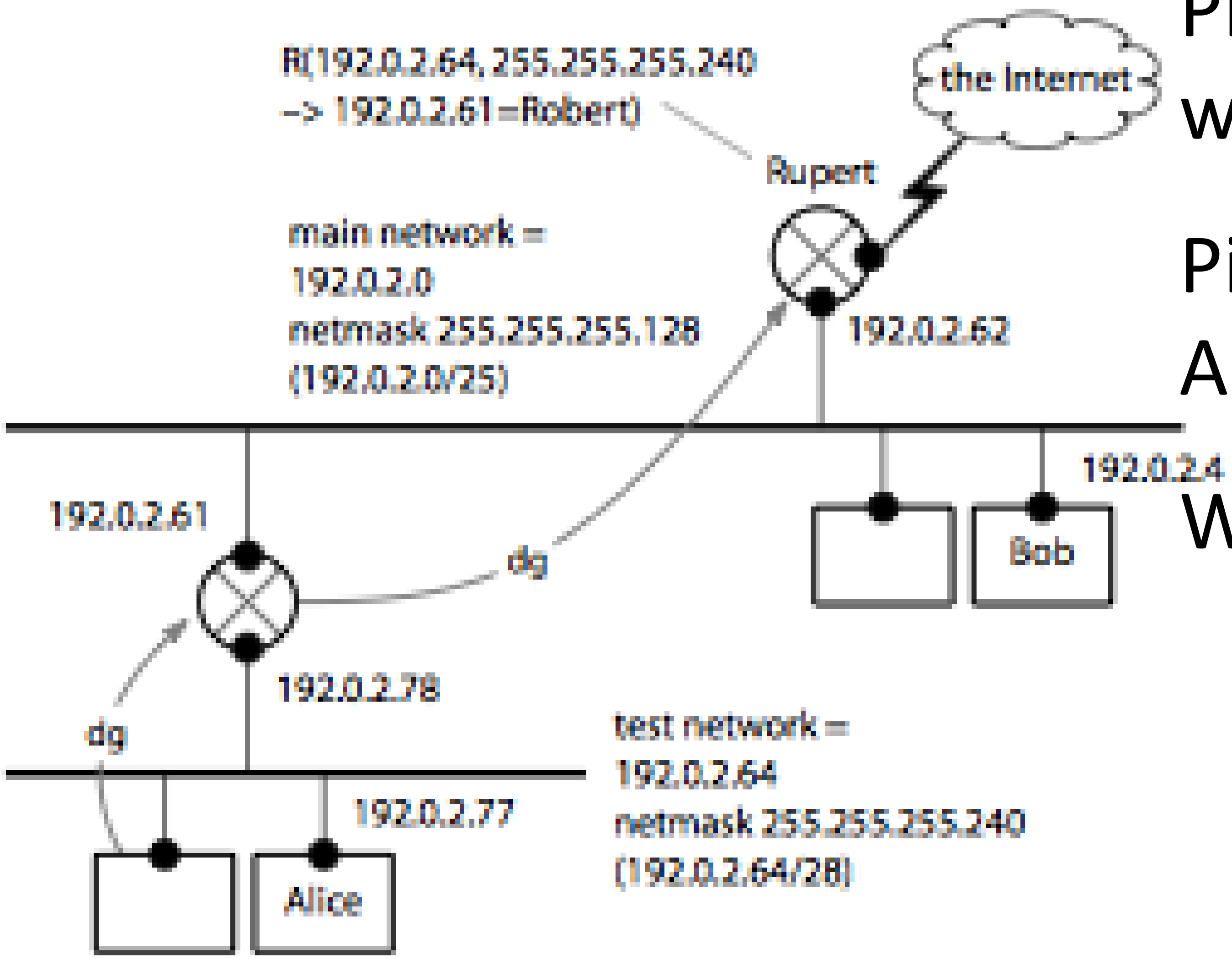
	10.1.1.0	10.4.4.0	10.6.6.0	Internet
Alice				
Bob				
Carol				

Lab–multiple networks on the same local wire

- The names of extra interfaces on a network card are created by appending ":1," ":2," etc.
- ifconfig eth0:1 10.2.2.14 netmask 255.255.255.0
- Ip addr
- ping 10.2.2.



Lab – creating simple sub-nets – the wrong way



Ping Bob from Alice, what happens ?

Ping Bob from Alice ?

What is wrong ?

Lab–creating simple sub-nets–the right way

- Instead of allocating our whole 128-address range to the main network, we must allocate a smaller range
- We could have the following arrangement:
  - main network = .0–.7, sub-net-1 = .64–.127, with all the other little ranges .8–.15, .16–.31, 32–.63 still available for use by other sub-nets later
- You can have a tiny main network and many bigger sub-nets

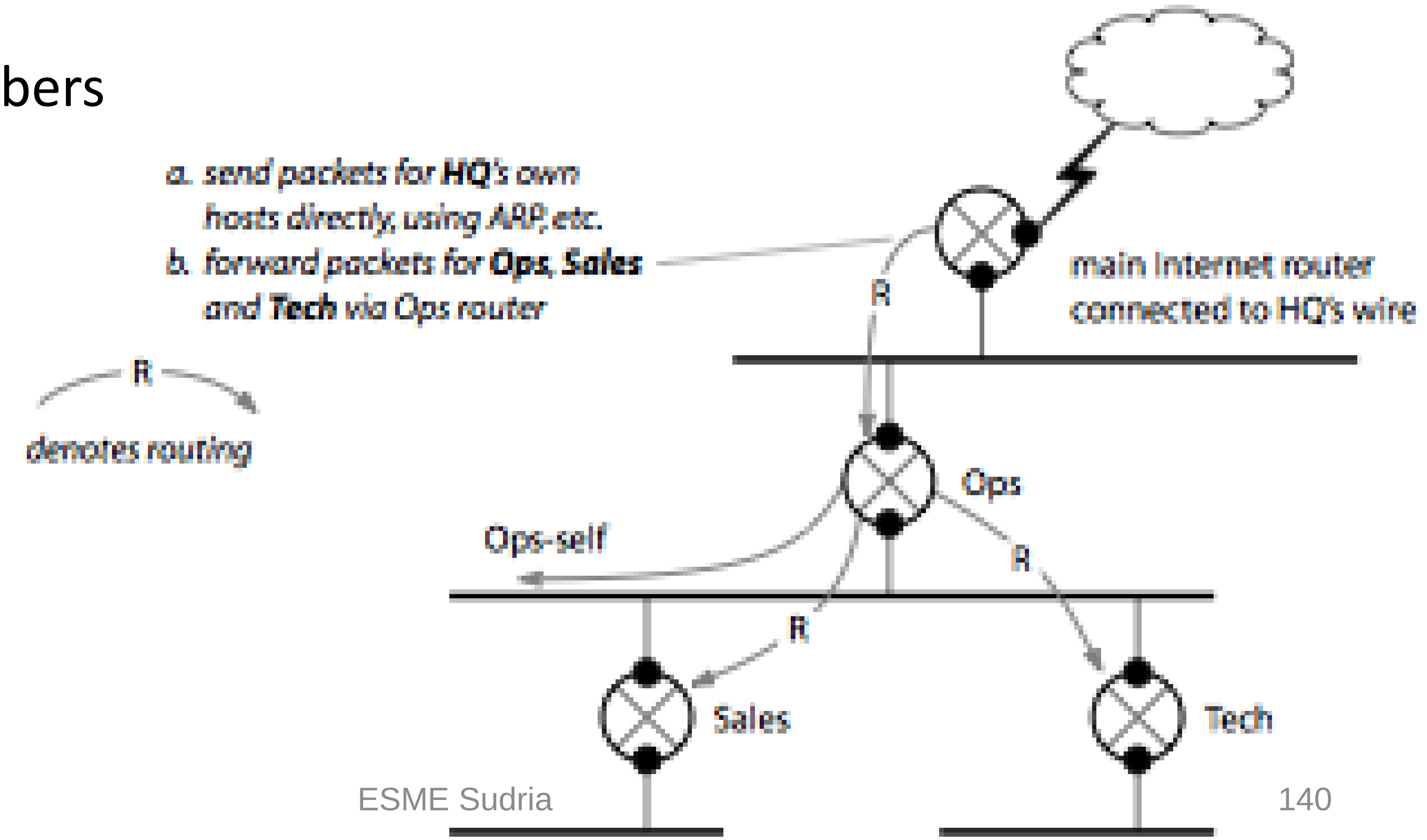
Lab – complex sub-nets

Range allocated by our ISP : 256 addresses = 192.0.2.0 netmask 255.255.255.0.

Calculate ranges

Assign IP numbers

Network	No. hosts
HQ-self	10
HR	8
Ops-self	6
Sales	50
Tech	15
Fin	14


a. send packets for HQ's own hosts directly, using ARP, etc.
b. forward packets for Ops, Sales and Tech via Ops router
main Internet router connected to HQ's wire
R
R
denotes routing
Ops
Ops-self
R
R
Sales
Tech

1 Networking basics

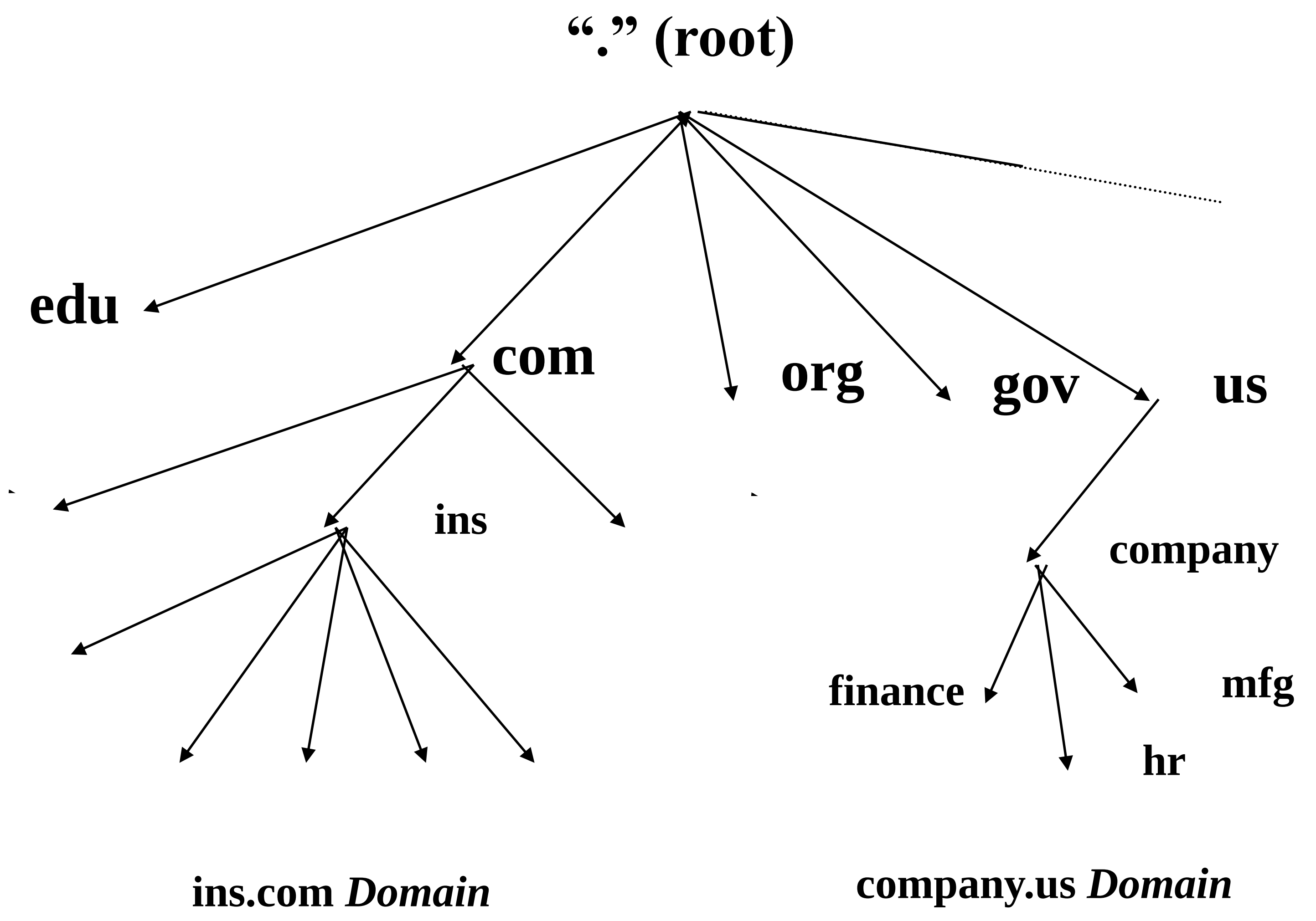
The Basics

- DNS is simple:
  - Ask it a question, it gives you an answer;
  - the trick: how the answer is obtained
- It was designed as a rudimentary directory
- The most common questions and answers
  - –Please translate a name to an IP Address
  - –Please translate an IP address to a name
  - –Please give me the IP Address of a gateway so I can deliver e-mail

The Basics

- Client-Server architecture
  - Client is called a resolver
  - DNS server is pre-configured manually or via DCHP
- The DNS server looks up the query data
  - In its configuration files
  - In it’s cache from prior queries
  - From other DNS servers
- The implementation of the DNS Server is called BIND (Berkeley Internet Name Domain) Server

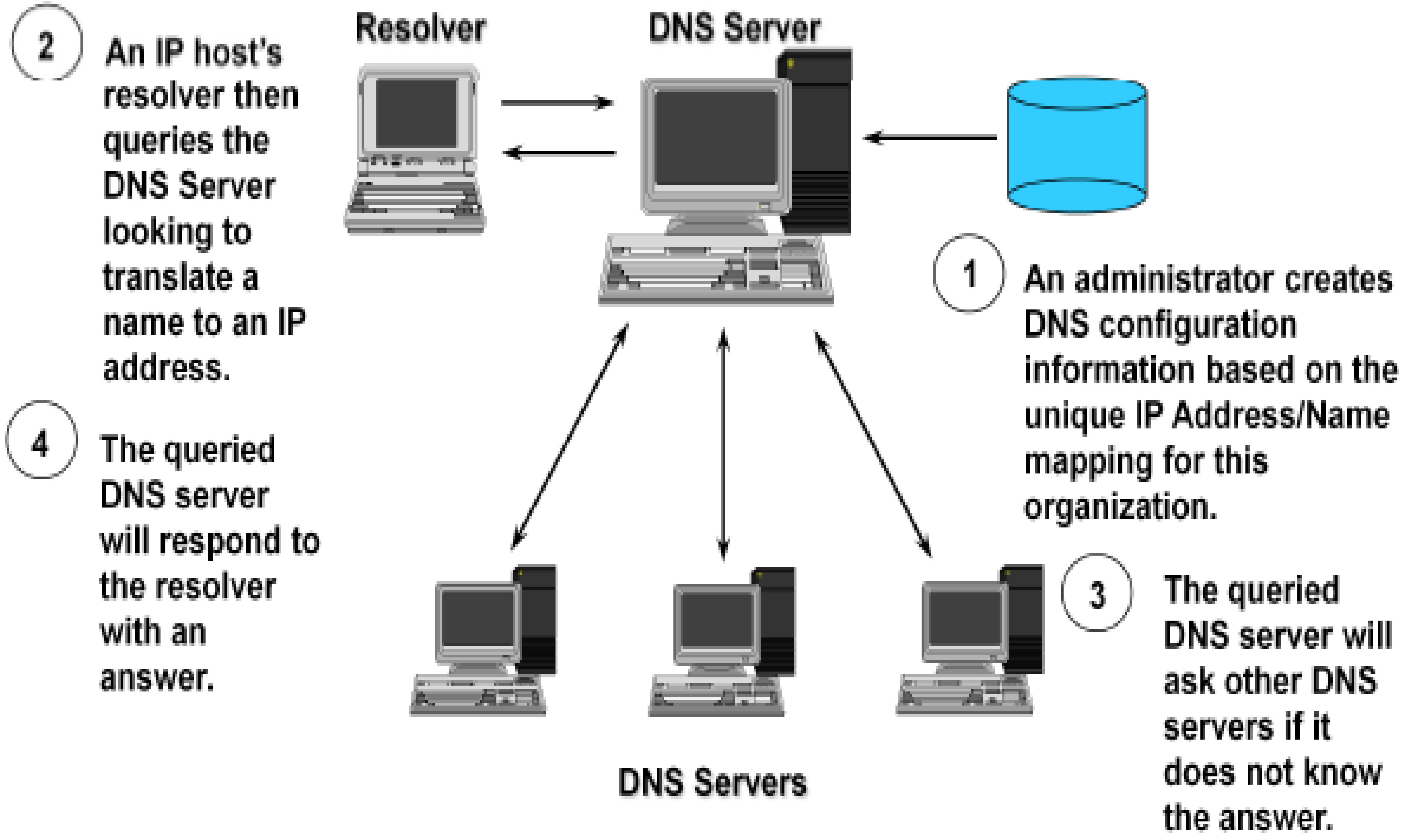
Name Space Structure



The Servers

- A DNS Server has responsibility (“authority”) for a part of the DNS Name Space hierarchy
- Maintains information about Domains in “Zones”
- Zones contain the information used to answer questions
  - –Questions are called “queries”
  - –Answers are called “responses”

Servers



An IP host's resolver then queries the DNS Server looking to translate a name to an IP address.

The queried DNS server will respond to the resolver with an answer.

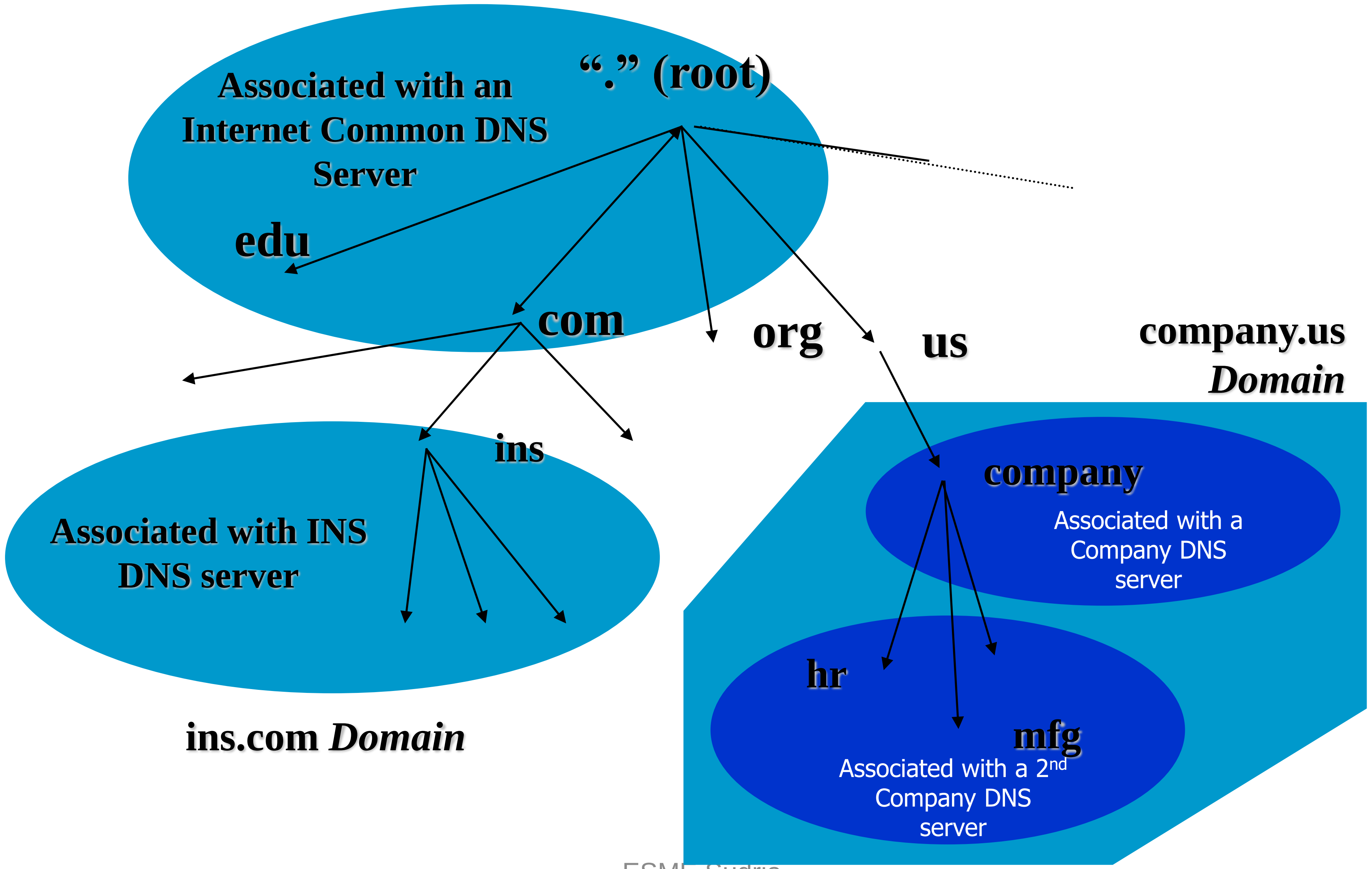
An administrator creates DNS configuration information based on the unique IP Address/Name mapping for this organization.

The queried DNS server will ask other DNS servers if it does not know the answer.

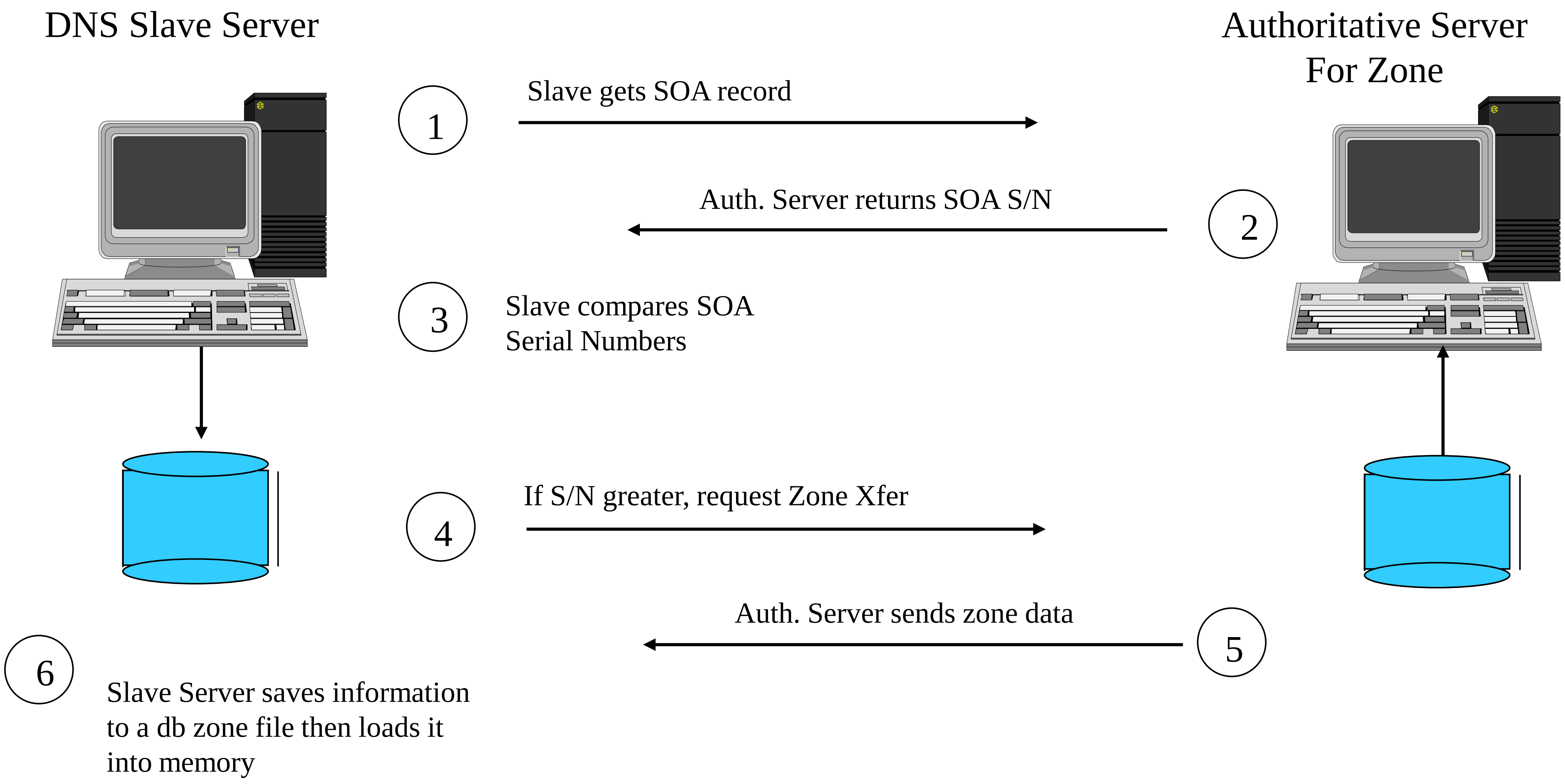
Domains and Zones

- Domains represent the conceptual aspect of the DNS system
- Zones are the embodiment of one or more domains on a DNS Server
- Zone Types
  - Master, Slave, Forward, Stub
- Delegation
  - A parent domain turns over responsibility of a child domain
- A configuration file (boot file) defines the zones
  - Called “named.conf”, in the /etc directory by default

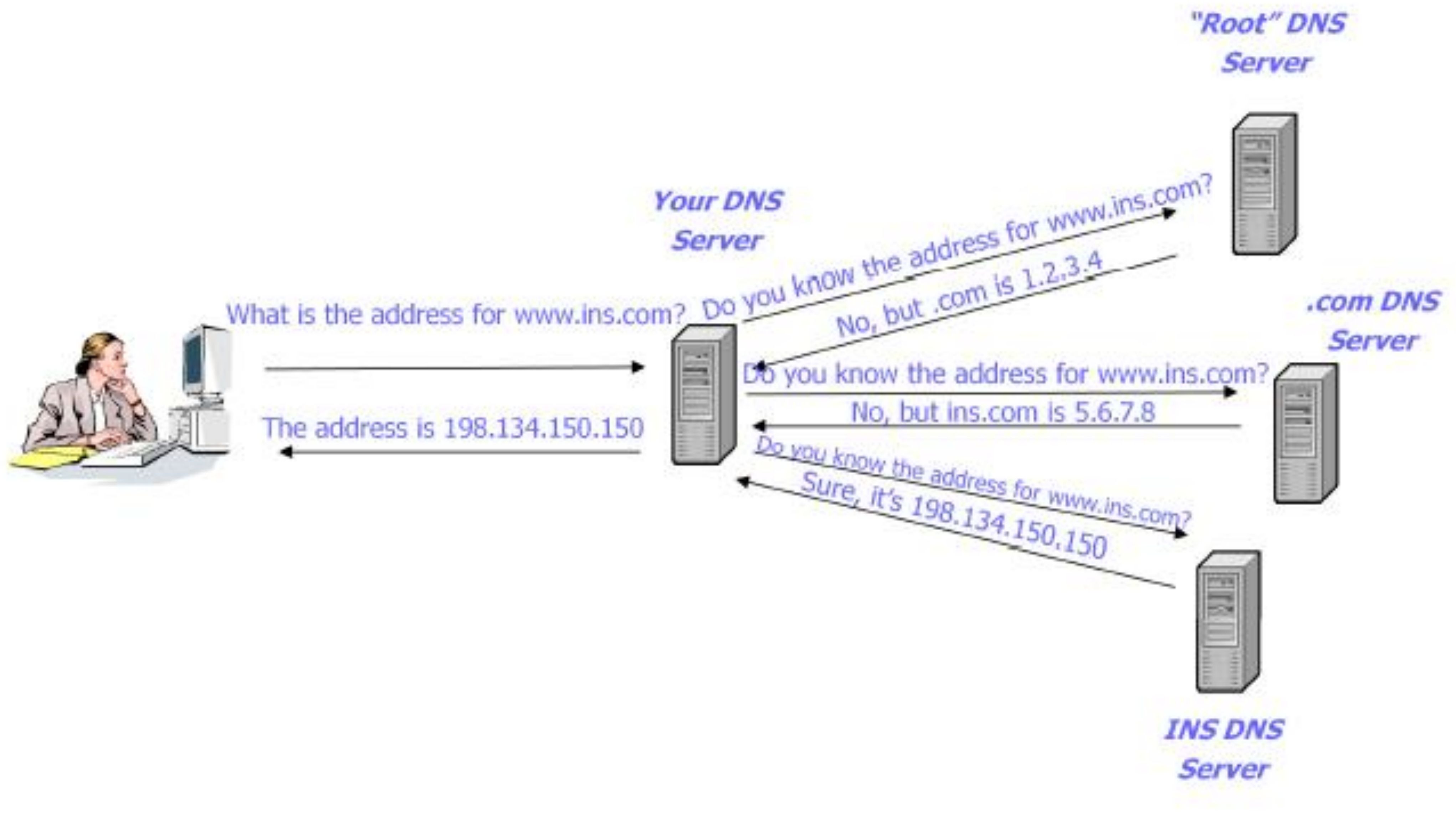
Zones to Domains


"." (root)
Associated with an Internet Common DNS Server
edu
com
org
us
company.us *Domain*
ins
Associated with INS DNS server
ins.com *Domain*
company
Associated with a Company DNS server
hr
mfg
Associated with a 2nd Company DNS server

Slave Zones


DNS Slave Server
Authoritative Server For Zone
1
Slave gets SOA record
2
Auth. Server returns SOA S/N
3
Slave compares SOA Serial Numbers
4
If S/N greater, request Zone Xfer
5
Auth. Server sends zone data
6
Slave Server saves information to a db zone file then loads it into memory

DNS Concepts – Resolution


"Root" DNS Server
Your DNS Server
.com DNS Server
INS DNS Server
What is the address for www.ins.com?
The address is 198.134.150.150
Do you know the address for www.ins.com?
No, but .com is 1.2.3.4
Do you know the address for www.ins.com?
No, but ins.com is 5.6.7.8
Do you know the address for www.ins.com?
Sure, it's 198.134.150.150

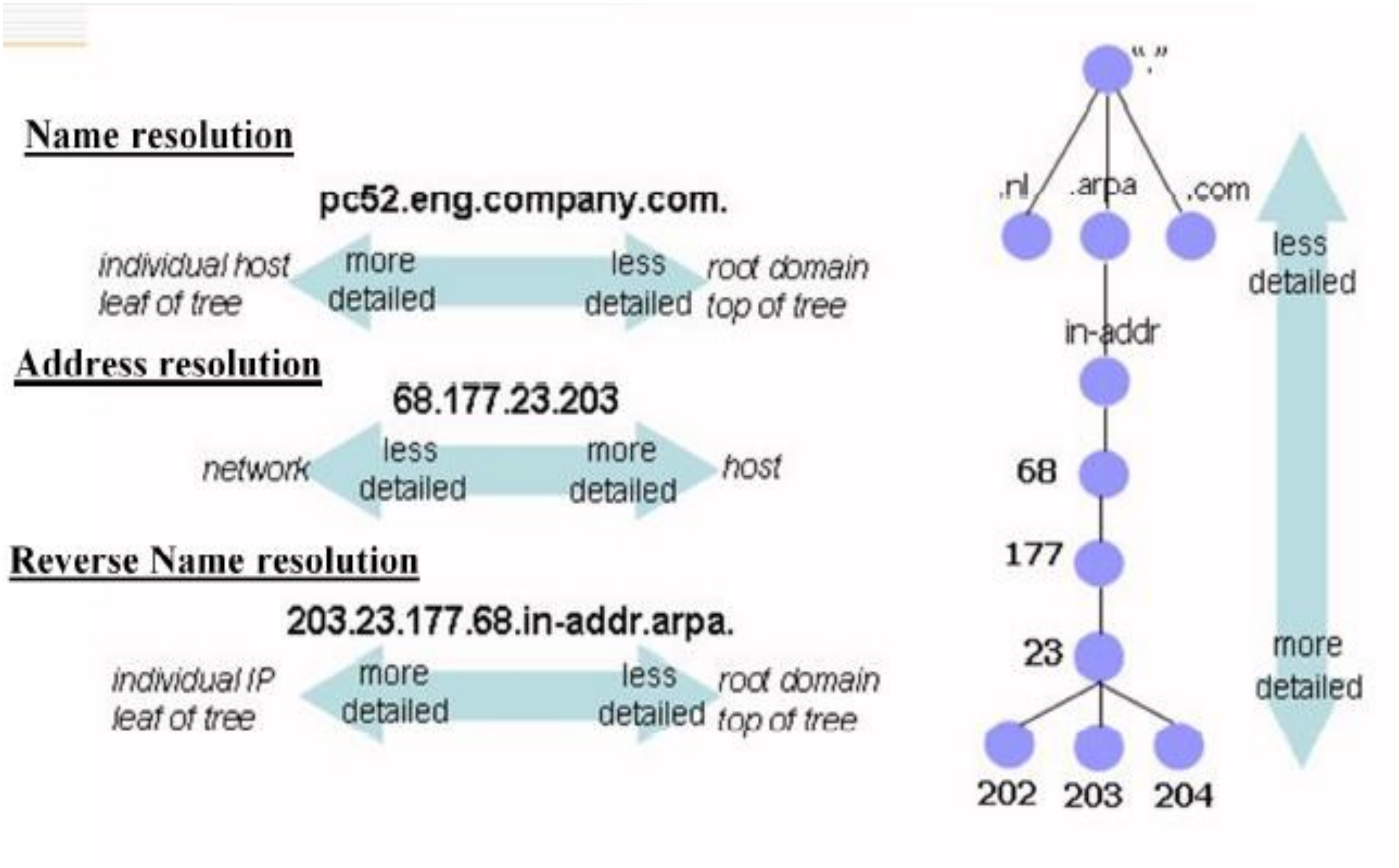
Caching

- Learning Information from other DNS servers is called “caching”
- The authoritative DNS indicate how long information can be retained
  - this is called Time-To-Live for a DNS response, or “TTL”
- A Master or Slave DNS server response is “Authoritative”
- A DNS server response from cache is “non-Authoritative”

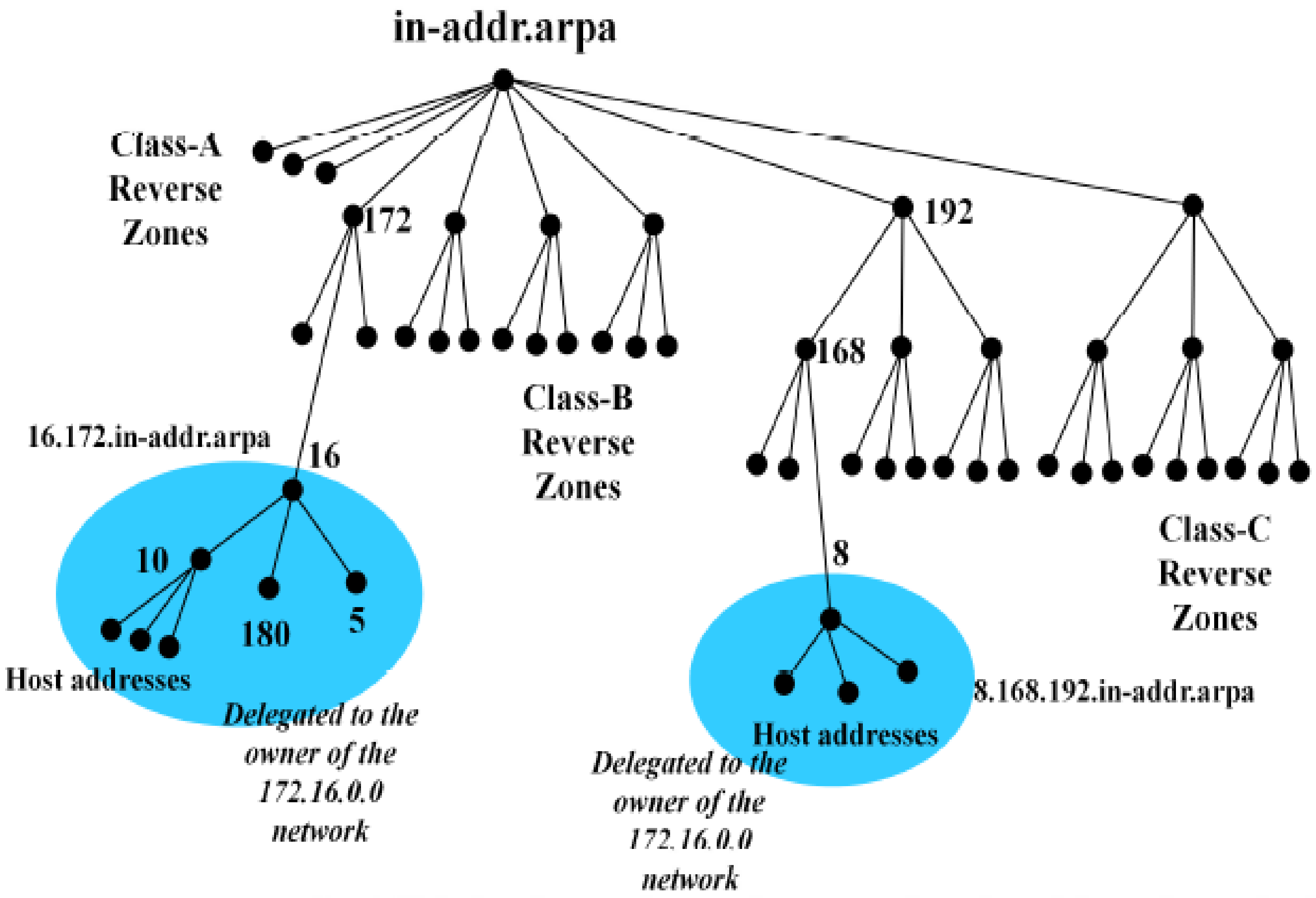
DNS Reverse Zones

- Name to IP Address translation is straight forward
- Address to Name Translation, called “reverse name resolution”
- DNS operates exclusively with character data; an IP Address is a 32-bit binary number
- FQDNs/Domain names are evaluated from right to left
- IP Addresses are interpreted from left to right
- IP Addresses must be “converted” to be interpreted like a domain name (right to left)

DNS Reverse Zones: Structure



DNS Reverse Zones: Delegation

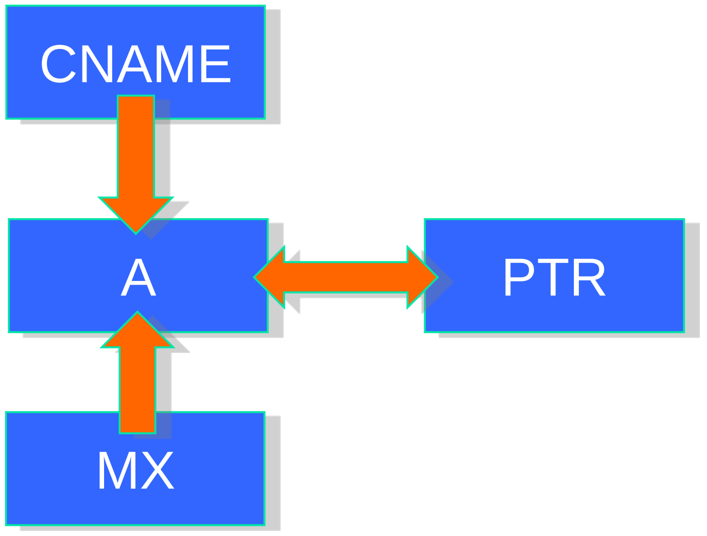


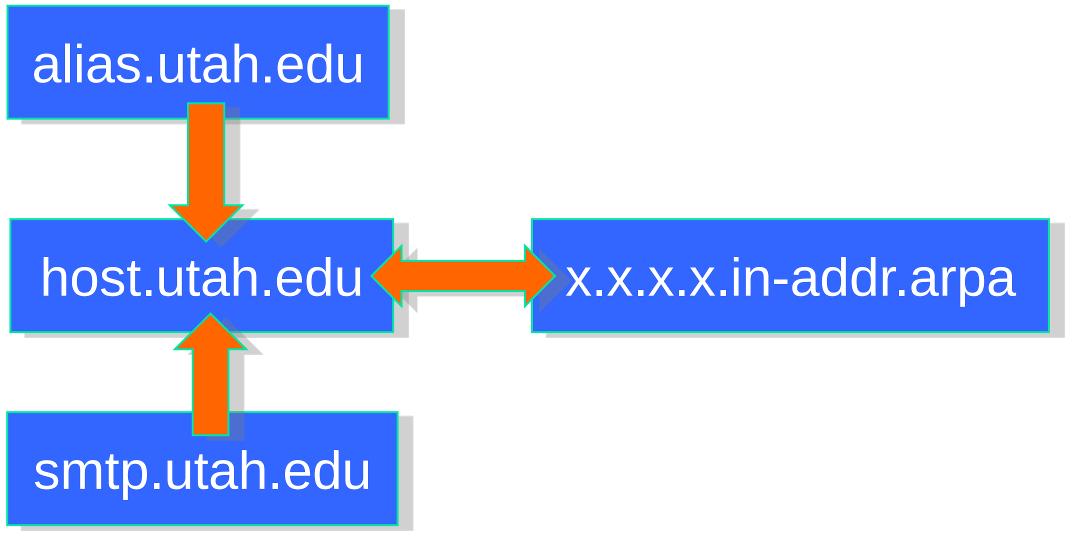
DNS Resource Records

- The Question:
  - Class: “IN” which stands for Internet
  - Type: the type of data being requested
  - Owner: a name to address mapping
- The Answer:
  - Rdata: the actual answer to the question
  - TTL: Time-To-Live; how long this answer is valid
- Record types: SOA, NS, A, PTR, CNAME, MX
- A DNS server looks for an exact match of Class, Type and Owner
  - Resource Records are stored on zone db files

DNS Resource Records

- Resource Records relationships





Lab–configuring Linux to use the DNS

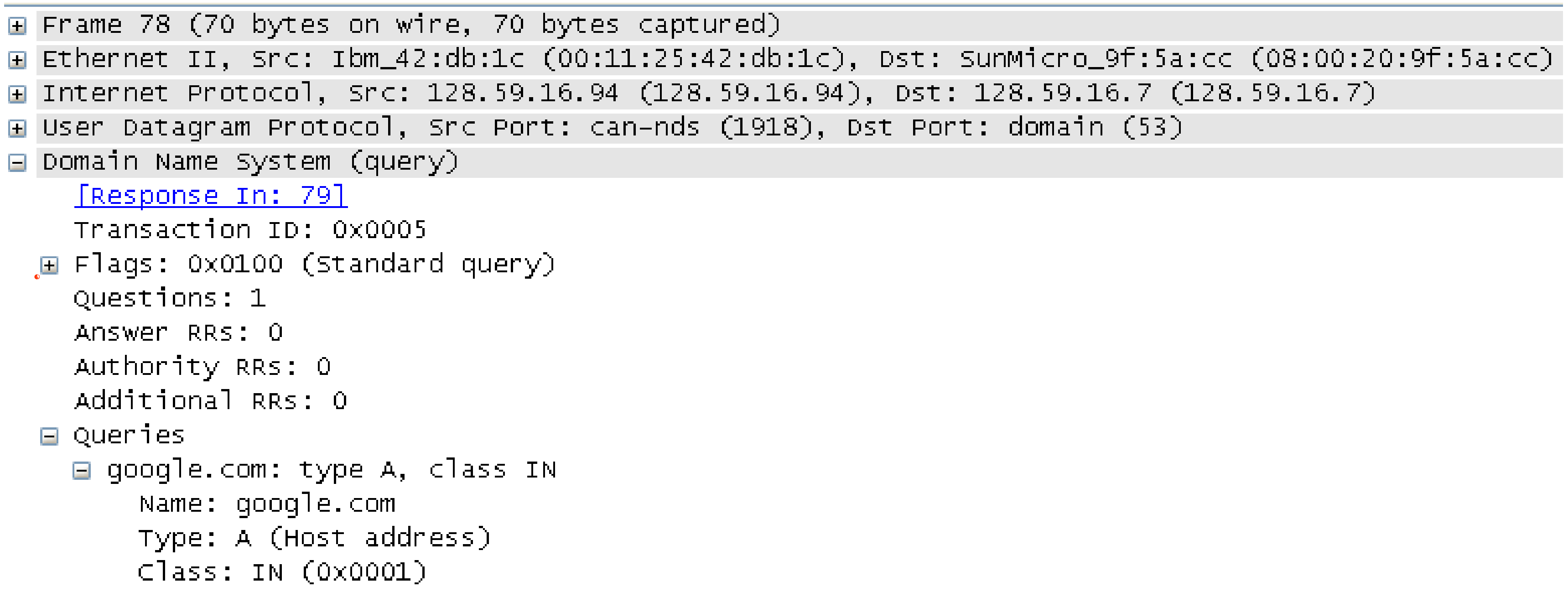
- Add a DNS server in /etc/resolv.conf
  - nameserver 192.168.0.100
- Associates this computer with the votredomaine.com domain
  - domain votredomaine.com
- systemd-resolve in Ubuntu
  - /etc/resolv.conf: nameserver 127.0.0.53
  - /run/systemd/resolve/resolv.conf: nameserver 192.168.1.254

Lab–interrogating the DNS– nslookup and host

- nslookup www.esme.fr
  - Prints the IP address corresponding to www.esme.fr
  - Control Panel > Network or /etc/resolv.conf
- Nslookup 178.170.115.96
  - prints the name corresponding to 178.170.115.96
- Host www.esme.fr 8.8.8.8
  - use non-default DNS server (8.8.8.8)

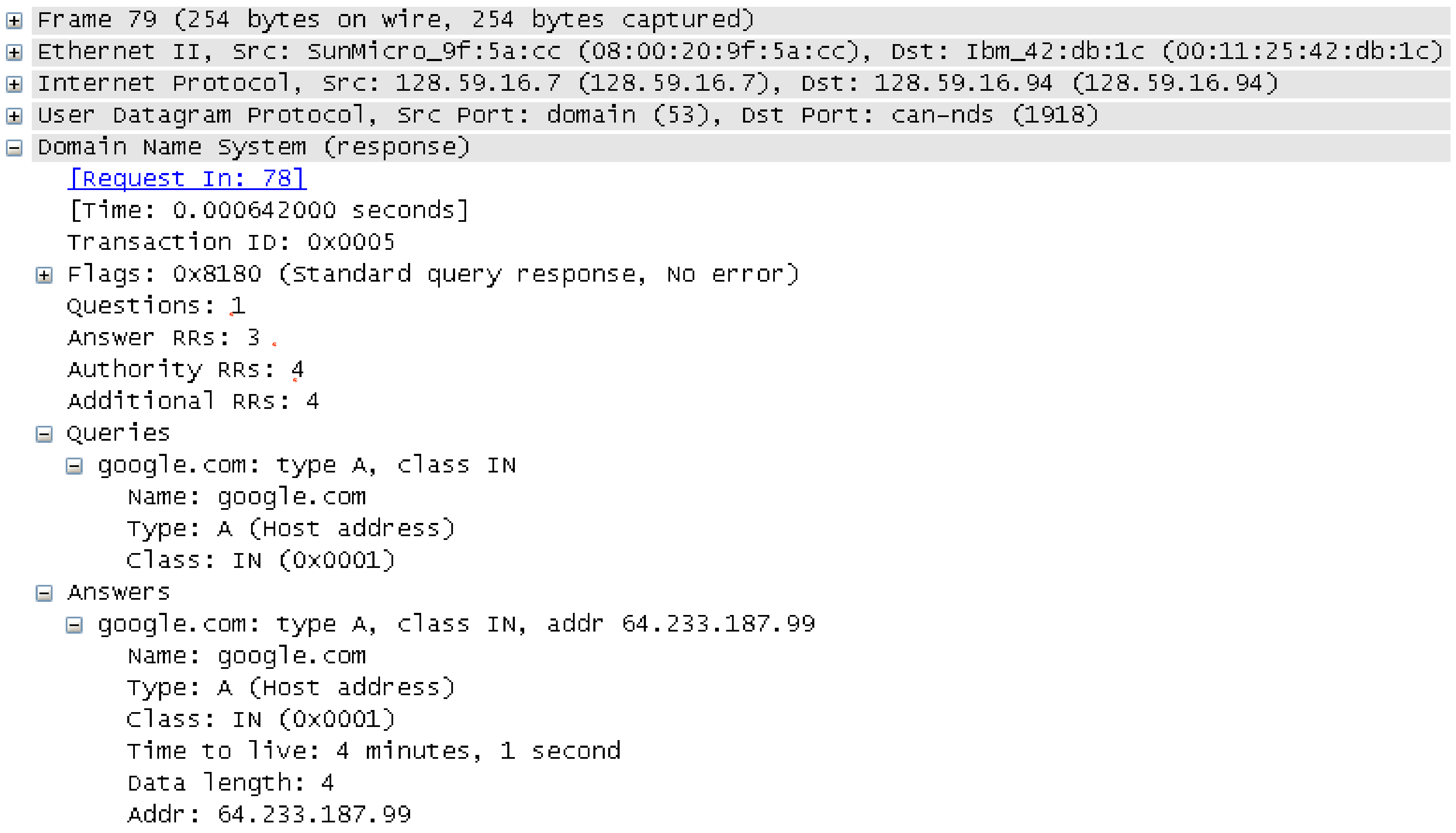
Lab–interrogating the DNS–nslookup and host

- nslookup google.com



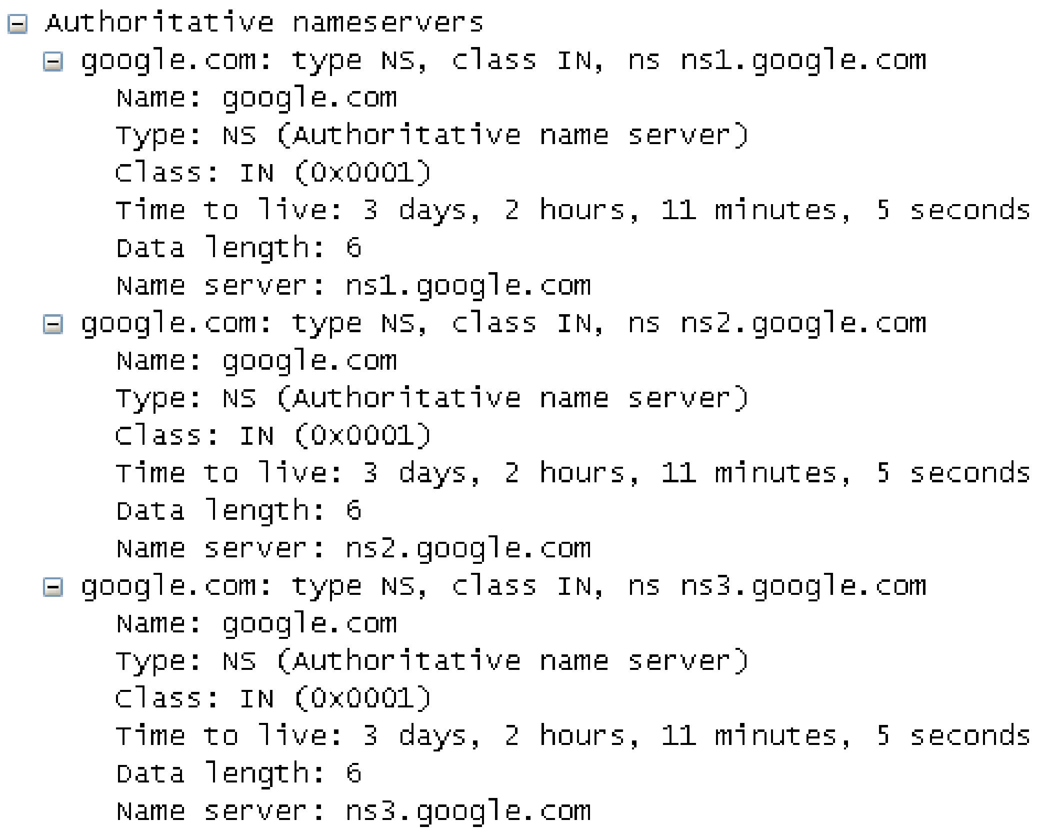
```
⊞ Frame 78 (70 bytes on wire, 70 bytes captured)
⊞ Ethernet II, Src: Ibm_42:db:1c (00:11:25:42:db:1c), Dst: SunMicro_9f:5a:cc (08:00:20:9f:5a:cc)
⊞ Internet Protocol, Src: 128.59.16.94 (128.59.16.94), Dst: 128.59.16.7 (128.59.16.7)
⊞ User Datagram Protocol, Src Port: can-nds (1918), Dst Port: domain (53)
⊟ Domain Name System (query)
    [Response In: 79]
    Transaction ID: 0x0005
  ⊞ Flags: 0x0100 (Standard query)
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  ⊟ Queries
    ⊟ google.com: type A, class IN
        Name: google.com
        Type: A (Host address)
        Class: IN (0x0001)
```

Lab–interrogating the DNS–nslookup and host



```
⊞ Frame 79 (254 bytes on wire, 254 bytes captured)
⊞ Ethernet II, Src: SunMicro_9f:5a:cc (08:00:20:9f:5a:cc), Dst: Ibm_42:db:1c (00:11:25:42:db:1c)
⊞ Internet Protocol, Src: 128.59.16.7 (128.59.16.7), Dst: 128.59.16.94 (128.59.16.94)
⊞ User Datagram Protocol, Src Port: domain (53), Dst Port: can-nds (1918)
⊟ Domain Name System (response)
    [Request In: 78]
    [Time: 0.000642000 seconds]
    Transaction ID: 0x0005
  ⊞ Flags: 0x8180 (Standard query response, No error)
    Questions: 1
    Answer RRs: 3
    Authority RRs: 4
    Additional RRs: 4
  ⊟ Queries
    ⊟ google.com: type A, class IN
        Name: google.com
        Type: A (Host address)
        Class: IN (0x0001)
  ⊟ Answers
    ⊟ google.com: type A, class IN, addr 64.233.187.99
        Name: google.com
        Type: A (Host address)
        Class: IN (0x0001)
        Time to live: 4 minutes, 1 second
        Data length: 4
        Addr: 64.233.187.99
```

Lab–interrogating the DNS–nslookup and host



```
⊟ Authoritative nameservers
  ⊟ google.com: type NS, class IN, ns ns1.google.com
      Name: google.com
      Type: NS (Authoritative name server)
      Class: IN (0x0001)
      Time to live: 3 days, 2 hours, 11 minutes, 5 seconds
      Data length: 6
      Name server: ns1.google.com
  ⊟ google.com: type NS, class IN, ns ns2.google.com
      Name: google.com
      Type: NS (Authoritative name server)
      Class: IN (0x0001)
      Time to live: 3 days, 2 hours, 11 minutes, 5 seconds
      Data length: 6
      Name server: ns2.google.com
  ⊟ google.com: type NS, class IN, ns ns3.google.com
      Name: google.com
      Type: NS (Authoritative name server)
      Class: IN (0x0001)
      Time to live: 3 days, 2 hours, 11 minutes, 5 seconds
      Data length: 6
      Name server: ns3.google.com
```

Lab–interrogating the DNS–nslookup and host

- $ nslookup google.com ns.google.com

```
Server:         ns.google.com
Address:  216.239.32.10#53

Name:    google.com
Address: 216.58.198.206
Name:    google.com
Address: 2a00:1450:4007:809::200e
```

- $ nslookup google.com

```
Server:         127.0.0.53
Address:  127.0.0.53#53

Non-authoritative answer:
Name:    google.com
Address: 172.217.22.142
Name:    google.com
Address: 2a00:1450:4007:810::200e
```

Lab–DNS–host command debug options

```
$ sudo host -v www.esme.fr
Trying "www.esme.fr"
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 48165
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 3,
ADDITIONAL: 0

;; QUESTION SECTION:
;www.esme.fr.                  IN      A

;; ANSWER SECTION:
www.esme.fr.             3453  IN      CNAME      esme.fr.
esme.fr.           3453  IN      A       178.170.115.96

;; AUTHORITY SECTION:
esme.fr.           10653         IN      NS     a.dns.gandi.net.
esme.fr.           10653         IN      NS     c.dns.gandi.net.
esme.fr.           10653         IN      NS     b.dns.gandi.net.

Received 120 bytes from 127.0.1.1#53 in 11 ms
Trying "esme.fr"
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 6647
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1,
ADDITIONAL: 0

;; QUESTION SECTION:
;esme.fr.                IN      AAAA

;; AUTHORITY SECTION:
esme.fr.           110   IN      SOA   a.dns.gandi.net.
hostmaster.gandi.net. 1509114607 10800 3600 604800
10800

Received 87 bytes from 127.0.1.1#53 in 11 ms
Trying "esme.fr"
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 15456
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 3,
ADDITIONAL: 0

;; QUESTION SECTION:
;esme.fr.                IN      MX

;; ANSWER SECTION:
esme.fr.           3600  IN      MX    0 esme-
fr.mail.protection.outlook.com.

;; AUTHORITY SECTION:
esme.fr.           10800         IN      NS     b.dns.gandi.net.
esme.fr.           10800         IN      NS     c.dns.gandi.net.
esme.fr.           10800         IN      NS     a.dns.gandi.net.

Received 137 bytes from 127.0.1.1#53 in 25 ms
```

Lab–configuring your PC to use DNS

- display the current hostname
  - hostname
- set name to string
  - hostname <string>
- To set the domain name, edit the file /etc/resolv.conf and insert your domain
  - domain example.com
- Exercise:
  - configure your domain name on your Linux box to be .example.com. What happens when you “ping alice”?

Lab–configuring DNS – the "searchlist"

- In the file /etc/resolv.conf enter searchlist entries
  - search ops.example.com example.co.uk partnerco.com
- Exercise:
  - Set your searchlist as above
  - Then start tcpdump
  - ping badname, where badname doesn't exist on your LAN
  - This will force the resolver to try every entry in the searchlist

Troubleshooting – identifying a problem

- DNS can’t find the IP address for a name
  - run host or nslookup
- Example :
  - $ host junk.example.com
  - Host junk.example.com not found: 3(NXDOMAIN)

Troubleshooting – identifying a problem

- if DNS fails to resolve an internal name
  - a likely cause is that the domain-name and searchlist settings on your machine are empty
  - Your site's DNS administrator hasn't entered the name (alice) in your DNS database
- the DNS can't find the name corresponding to an IP number
  - the DNS administrator for the site in question didn't set up the PTR records properly

Troubleshooting – identifying a problem

- Internet Explorer itself caches the result
  - IE remembers that it couldn’t resolve the name and doesn’t try again
  - Solution : exit IE and restart it
- DNS packets can get dropped, especially on a busy network
  - see if the same problem happens on another machine
  - If it doesn’t, you do have a problem with the first machine

Troubleshooting – identifying a problem

- weird names show up in queries, e.g. junk.ibm.com.uit.co.uk
    - When you : ping junk.ibm.com
- host can give different values for the same query if you run it more than once
    - “type=ANY” query can return only what’s in the queried server’s cache, rather than causing a full recursive query to be done

Debug output for a failed name resolution

- run “tcpdump -n”
- Use ping, not nslookup or host, to generate the lookup.
  - Ping uses your machine’s standard resolver so it will generate the same queries that any other standard application would
- Append a dot to the name you’re querying,
  - e.g. “host junk.ibm.com.” so that it’s interpreted as an FQDN
  - the domain or explicit searchlist activity is bypassed
  - For internal name append your domain name: “host alice.example.com.”

DNS problems and causes

- DNS relies on underlying networking infrastructure
  - Bypass the DNS by pinging any host by IP number
- Client is wrongly configured
  - see if the same problem occurs on another machine
  - no DNS packets: Check that you configured the resolver with the IP address of a DNS server
- DNS server on your LAN is down
  - tcpdump will show DNS queries sent to the server but no DNS replies, only ICMP error messages

DNS problems and causes

- your ISP’s servers are down
  - packets going out to the server but none coming back
  - wrong IP address for the server in the resolver configuration
  - the server machine is dead. Try pinging it
  - you are not allowed to use this nameserver and it therefore refuses to reply
  - the machine is alive but the DNS server software isn’t running.
    - an easy way to check if the nameserver on is running :
    - telnet hostaddr 53

DNS problems and causes

- The DNS server that contains the data for the site you requested is down or inaccessible (“nameserver not responding.”)
  - use a different nameserver instead of your ISP’s servers
    - nslookup hostaddr server_IP_number
  - use a different ISP, so that you’re using different DNS servers
  - connect to another remote site, run the usual host or nslookup commands from there
- Whenever you get unusually long delays in an application
  - e.g. in telnet or ping or your e-mail client, it’s often due to DNS problems

Conclusions

- Some tools are essential for network troubleshooting :
  - Ping, traceroute, tcpdump
- Packets move from network to networks thanks to the routing mechanism which uses IP address / netmask information to choose the next hop
- Routing can be static or dynamic
- Basic rules can be used to troubleshoot netmasks and addresses
- DNS is a helpful tool that can be subject to malfunctions