Simplicity of Networking

Command Line Interface

ls	list dir content
cd	change current dir
cp	copy
rm	remove/delete
nano	text editor
vi	text editor (difficult)
dmesg	output kernel messages
tail	last contents of a file
more	file pager
less	file pager (nicer)
cat	binary output a file
man	manuals
mkdir	create a directory
touch	update/create a file (access time)
pwd	present working dir
ln	create (symbolic) link
du	disk use
chmod	change permission
fg/bg	fore-/background a process

CONTROL:

CTRL-C	Kill
CTRL-D	Exit
CTRL-Z	suspend
CTRL-S	stop input (undo CTRL-Q)
TAB	command completion!!!!

Commands can be put in a file for batching:

```
#!/bin/bash
pwd
cd /tmp
touch hoi
echo hallo > hoi
tail hoi
cd -
```

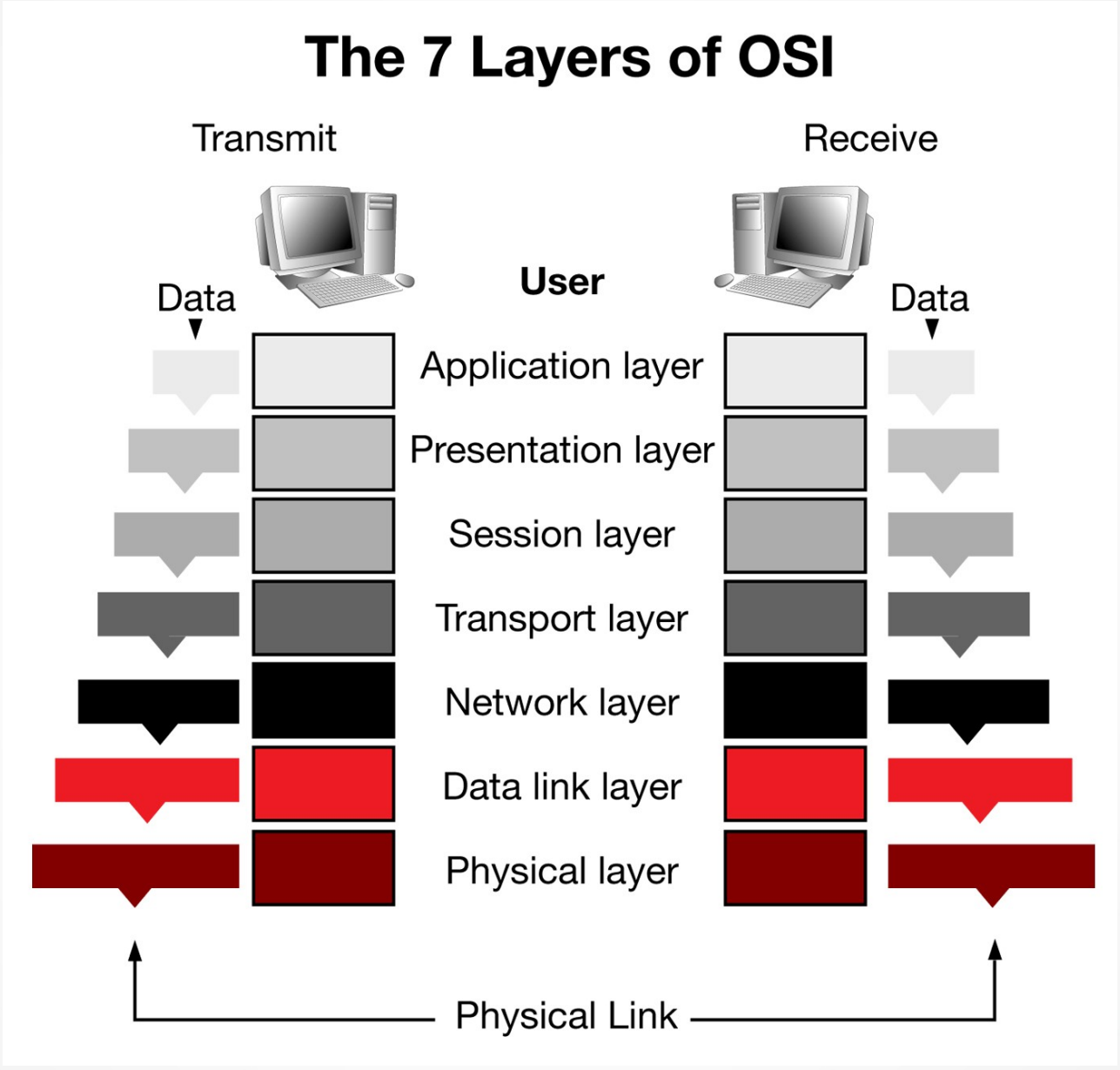
A practical guide to IP Networking

- OSI Model
- Layer 1 (Wires)
- Layer 2 (MAC Addresses)
- Layer 3 (IP addresses)
- Layer 4 (IP Services)
- Play Time
- Trouble

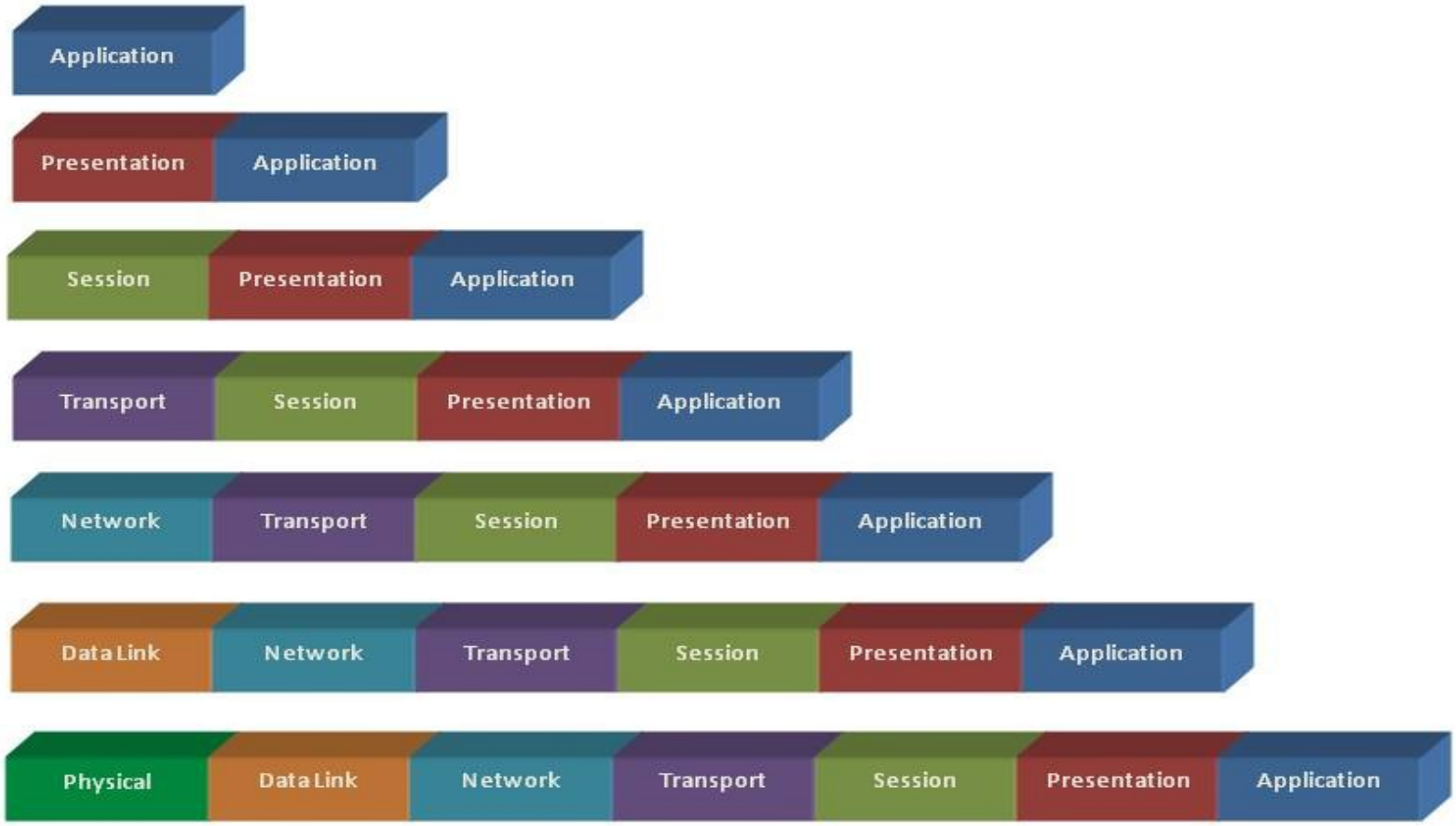
A practical guide to IP Networking

> *There's no place like 127.0.0.1*

OSI Model


The 7 Layers of OSI
Transmit
Receive
Data
User
Data
Application layer
Presentation layer
Session layer
Transport layer
Network layer
Data link layer
Physical layer
Physical Link

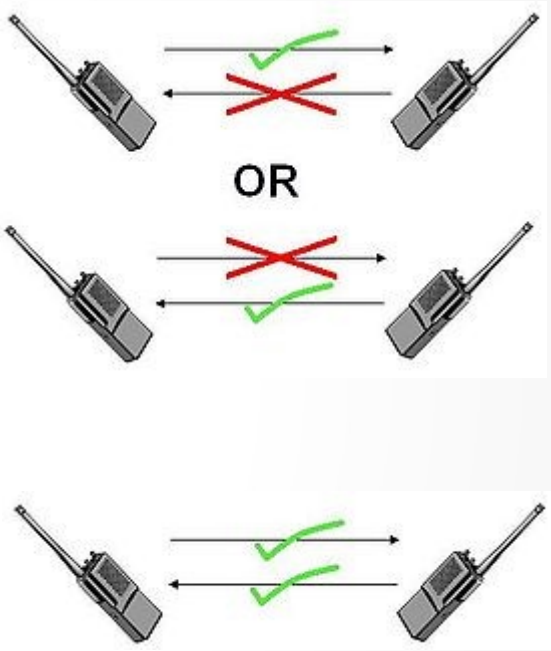
OSI Model



OSI Model Packet Encapsulation

Layer 1

- UTP Cables
- Fiber optic cables
- Wireless
- The bits of data!
- Full-Duplex/Half Duplex

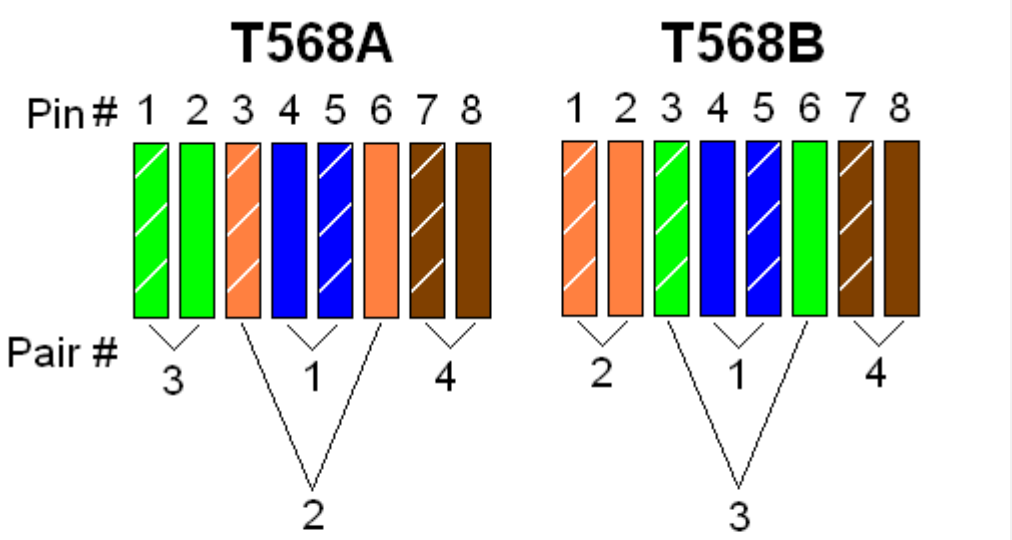


Layer 1

A word about UTP Cables (CAT 5/5e/6/7/etc)

„Twisted pair cabling is a type of wiring in which two conductors of a single circuit are twisted together for the purposes of canceling out electromagnetic interference“

- Unshielded **Twisted** Pair
- Shielded Twisted Pair
- 'Straight-through' or 'Cross'
- Wiring according to: T568A/T568B



Wire Pair	Wire Colors	T568A Standard	T568B Standard
1	Solid blue/blue with white stripe	Pins 4 and 5	Pins 4 and 5
2	Solid orange/orange with white stripe	Pins 3 and 6	Pins 1 and 2
3	Solid green/green with white stripe	Pins 1 and 2	Pins 3 and 6
4	Solid brown/brown with white stripe	Pins 7 and 8	Pins 7 and 8

Layer 1

Grab an UTP cable and connect it to your laptop or computer

Plug the other end into your assigned switch!

Layer 1

Linux

'**ethtool <DEV>**'

check it's output

wireless:

'iwconfig <DEV>'

OSX

'**Network Utility**'

info tab - link speed

Windows

'**ipconfig/all**'

Layer 1

```
Settings for p10p1:
    Supported ports: [ TP ]
    Supported link modes:   10baseT/Half 10baseT/Full
                            100baseT/Half 100baseT/Full
                            1000baseT/Full
    Supported pause frame use: No
    Supports auto-negotiation: Yes
    Advertised link modes:  Not reported
    Advertised pause frame use: No
    Advertised auto-negotiation: Yes
    Speed: 100Mb/s
    Duplex: Full
    Port: Twisted Pair
    PHYAD: 0
    Transceiver: internal
    Auto-negotiation: on
    MDI-X: Unknown
    Supports Wake-on: pg
    Wake-on: g
    Current message level: 0x0000003f (63)
                           drv probe link timer ifdown ifup
    Link detected: yes
```

Layer 2

To Know:

- MAC Address – a unique 64bit address which identifies your NIC (Network Interface Card)
- MAC first 24bit == manufacturer ID
- Happens before routing! *(or after)* *????? (later more)*
- Subnet / Broadcast domain – often used names for the layer 2 network
- Switches and hubs operate at layer 2

Layer 2

Linux

'**ip address**'

Try to find 'link/ether'

OSX

'**networksetup -listallhardwareports**'

Try to find 'Ethernet Address'

Windows

'**ipconfig/all**'

Try to find 'Physical address'

Layer 2

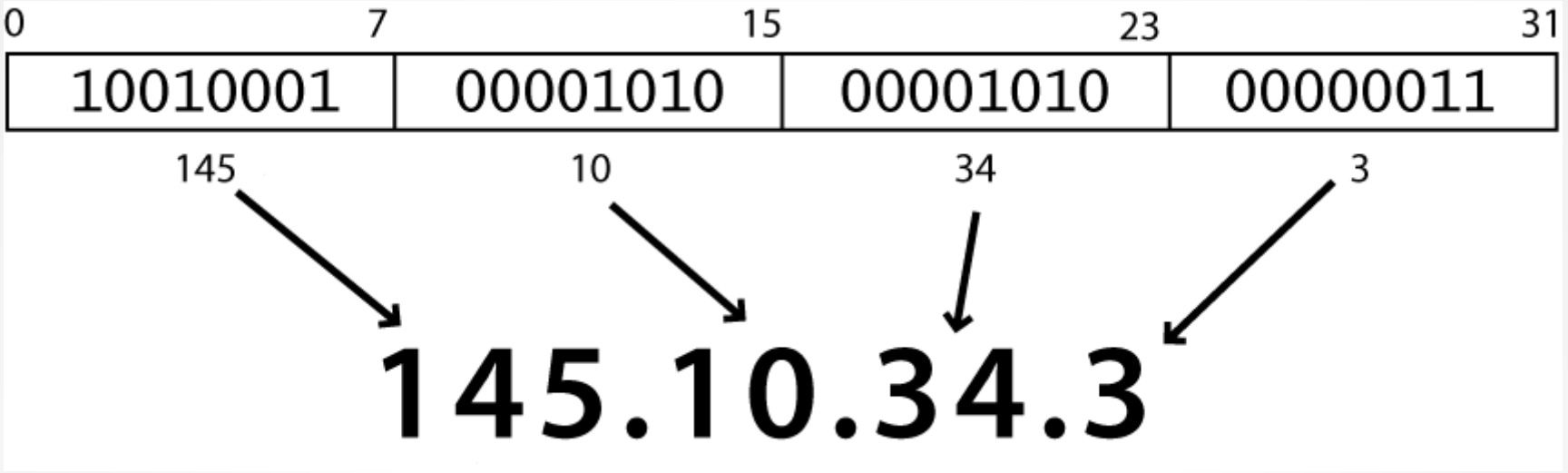
```
$ ip address show dev wlp2s0
2: wlp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group
default qlen 1000
    link/ether 9c:b6:d0:eb:70:73 brd ff:ff:ff:ff:ff:ff
    inet 10.2.4.160/24 brd 10.2.4.255 scope global dynamic noprefixroute wlp2s0
       valid_lft 7136sec preferred_lft 7136sec
    inet 10.2.4.158/24 brd 10.2.4.255 scope global secondary noprefixroute wlp2s0
       valid_lft forever preferred_lft forever
    inet6 fe80::9da0:69fa:3a10:247d/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

Layer 3

To Know:

- IP Address – a 32bit address which identifies your computer
- Subnet Mask – identifies the scope of your layer 2 network! (netmask)
- ARP – resolve ipaddresses to MAC addresses
- Route – a table to find your way out of the layer 2 network/subnet/broadcast domain

Layer 3




```
128  64  32  16 8 4 2 1
  1   0   0   1 0 0 0 1
128   +     16     + 1 = 145
```

Layer 3

Private IP adress ranges:

- 10.0.0.0/255.0.0.0 (/8)
- 172.16.0.0/255.240.0.0 (/12)
- 192.168.0.0/255.255.0.0 (/16)
- 169.254.0.0/255.255.0.0 (/16) *used for non dhcp networks/autoconf*

Special Ranges

- 224.0.0.0/240.0.0.0 (/4) multicast address space
- 127.0.0.1/255.0.0.0 (/8) loopback address space
- 0.0.0.0/255.0.0.0 (/8) broadcast addresses
- 255.255.255.255/255.255.255.255 (/32) limited broadcast

All address space is maintained by IANA

Layer 3

Ip address and it's subnet mask???

Imagine 192.168.1.1 with a subnet mask 255.255.255.0 (/24)

In bits

Ip: **1 1 0 0 0 0 0 0 . 1 0 1 0 1 0 0 0 . 0 0 0 0 0 0 0 1 . 0 0 0 0 0 0 0 1**

Layer 3

Ip address and it's subnet mask???

Imagine 192.168.1.1 with a subnet mask 255.255.255.0 (/24)

In bits

Mask: **1 1 1 1 1 1 1 1 . 1 1 1 1 1 1 1 1 . 1 1 1 1 1 1 1 1 . 0 0 0 0 0 0 0 0**

Layer 3

Ip address and it's subnet mask???

Imagine 192.168.1.1 with a subnet mask 255.255.255.0 (/24)

In bits

Mask: **1 1 1 1 1 1 1 1 . 1 1 1 1 1 1 1 1 . 1 1 1 1 1 1 1 1 . 0 0 0 0 0 0 0 0**
8 bits + 8 bits + 8 bits + 0 bits = 24bits

Layer 3

Ip address and it's subnet mask???

Imagine 192.168.1.1 with a subnet mask 255.255.255.0 (/24)

In bits

Ip: **1 1 0 0 0 0 0 0 . 1 0 1 0 1 0 0 0 . 0 0 0 0 0 0 0 1 . 0 0 0 0 0 0 0 1**
Mask: **1 1 1 1 1 1 1 1 . 1 1 1 1 1 1 1 1 . 1 1 1 1 1 1 1 1 . 0 0 0 0 0 0 0 0**

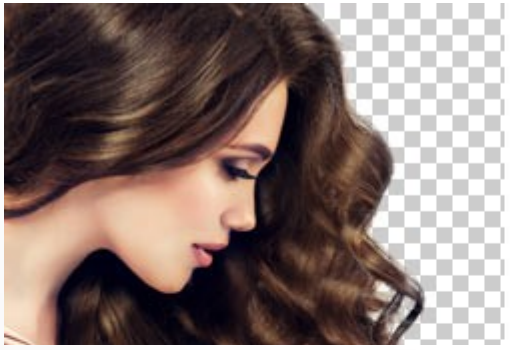
Layer 3

Ip address and it's subnet mask???

Imagine 192.168.1.1 with a subnet mask 255.255.255.0 (/24)

In bits

Ip: **1 1 0 0 0 0 0 0 . 1 0 1 0 1 0 0 0 . 0 0 0 0 0 0 0 1 . 0 0 0 0 0 0 0 1**
Mask: **1 1 1 1 1 1 1 1 . 1 1 1 1 1 1 1 1 . 1 1 1 1 1 1 1 1 . 0 0 0 0 0 0 0 0**



Layer 3

Ip address and it's subnet mask???

Imagine 192.168.1.1 with a subnet mask 255.255.255.0 (/24)

In bits

Ip:	**1 1 0 0 0 0 0 0 . 1 0 1 0 1 0 0 0 . 0 0 0 0 0 0 0 1 . 0 0 0 0 0 0 0 1**
Mask:	**1 1 1 1 1 1 1 1 . 1 1 1 1 1 1 1 1 . 1 1 1 1 1 1 1 1 . 0 0 0 0 0 0 0 0**
Bcast:	**1 1 0 0 0 0 0 0 . 1 0 1 0 1 0 0 0 . 0 0 0 0 0 0 0 1 . 1 1 1 1 1 1 1 1**
Net:	**1 1 0 0 0 0 0 0 . 1 0 1 0 1 0 0 0 . 0 0 0 0 0 0 0 1 . 0 0 0 0 0 0 0 0**

The very first address in the subnet is the network address : 192.168.1.0
The very last address in the subnet is the broadcast address : 192.168.1.255

They are reserved for the network to operate: you cannot use them!

Layer 3

OSX

Network preferences - NIC - advanced - manual

Windows

ncpa.cpl

adapter - properties - internet protocol version 4 (TCP/IPv4)

Linux

ip address add 192.168.99.37/24 dev eth0

Layer 3

Try a single ping to the address of your gateway (.1) in your subnet

ping -c 1 192.168.25.250

-c count: Stop after sending count packets.

Linux/OSX

ping -c 1 192.168.2.1

-c count: Stop after sending count packets.

Windows

ping 192.168.2.1

Layer 3

ARP, Request who-has 192.168.25.250 tell 192.168.25.176, length 28
ARP, Request who-has 192.168.25.250 tell 192.168.25.176, length 28
ARP, Request who-has 192.168.25.250 tell 192.168.25.176, length 28

ARP : Address Resolution Protocol.

Resolves ip addresses to MAC addresses

This is the glue between layer 2 and layer 3!

Layer 3

Check the ARP table on your computer

On your computer execute:

'**arp -an**'
-a display all
-n numeric, no resolving

Windows

'arp -a'

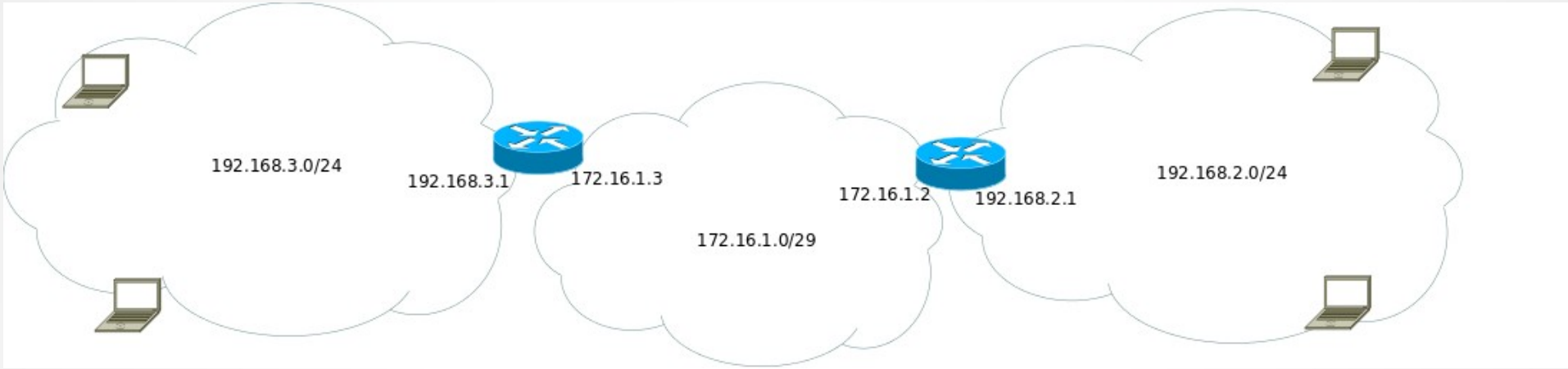
Layer 3

On your computer execute:

'**ping 8.8.8.8**'

Introducing routing

Layer 3



Have a look on the webinterface of your assigned router.

http://192.168.x.1

Layer 3

Linux

'**ip route add 172.16.1.0/29 via 192.168.x.1**'

OSX

'**sudo route -n add -net 172.16.1.0/29 192.168.x.1**'

Windows

'**route ADD 172.16.1.0 MASK 255.255.255.248 192.168.x.1'**

Layer 3

Linux

'**ip route**'

OSX

'**sudo netstat -nr**'

Windows

'**route PRINT**'

Layer 3

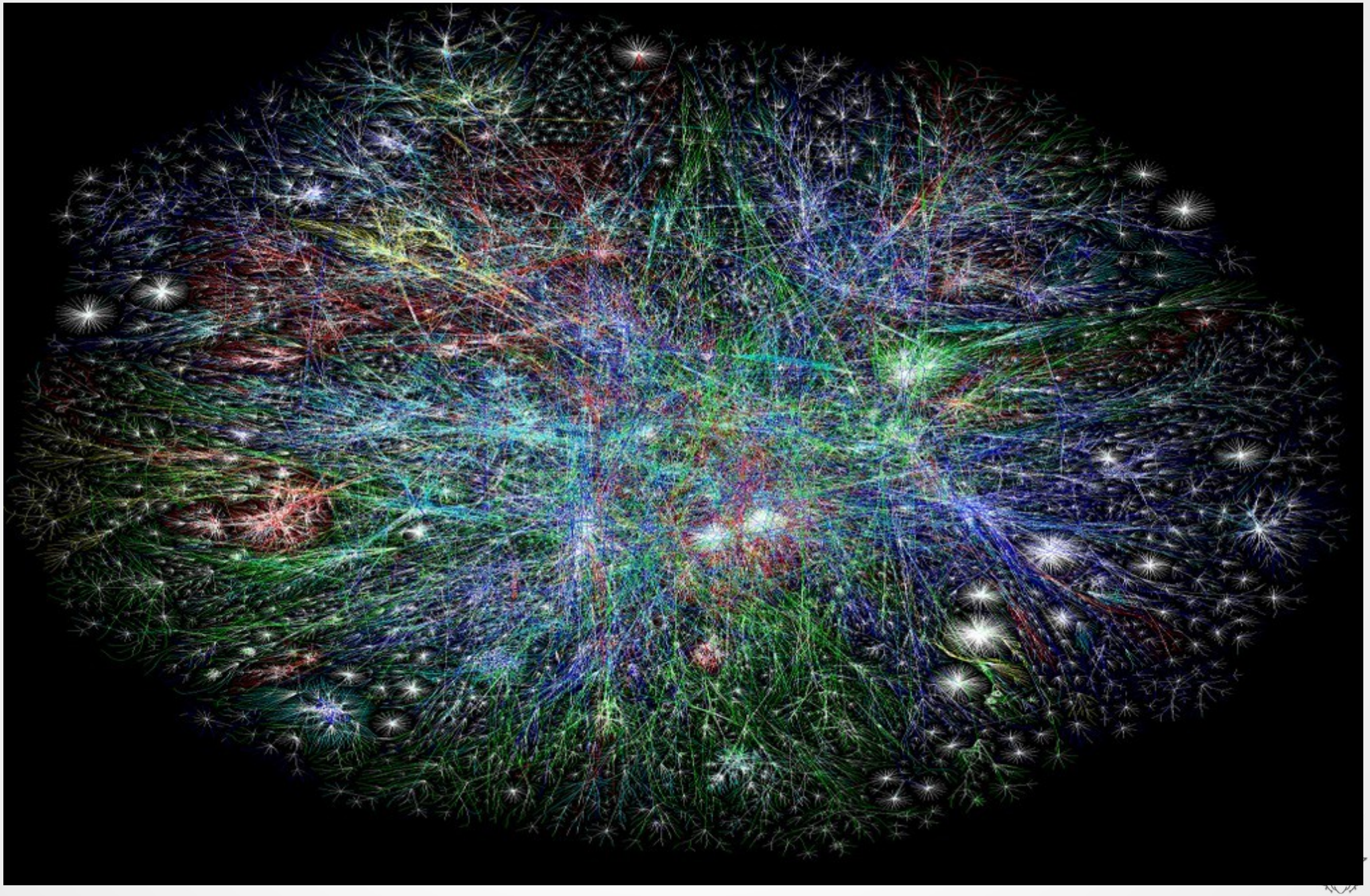
Remember:

- When you address a host within your subnet you'll
  - use MAC addresses to communicate
  - Use ARP to resolve an IP address to a MAC address

- When you address a host outside your subnet you'll
  - Look in the route table to find who knows how to find the host *(most specific)*
  - Use ARP to find the MAC address of the host who knows
  - Send your packet to that MAC address

This process continues until the host has been reached!

Layer 3



Layer 3

Linux

'**ip route add 192.168.y.0/24 via 192.168.x.1**'

OSX

'**sudo route -n add -net 192.168.y.0/24 192.168.x.1**'

Windows

'**route ADD 192.168.y.0 MASK 255.255.255.0 192.168.x.1**'

Layer 3

Linux

'**ip route add 8.8.8.8/32 via 192.168.x.1**'

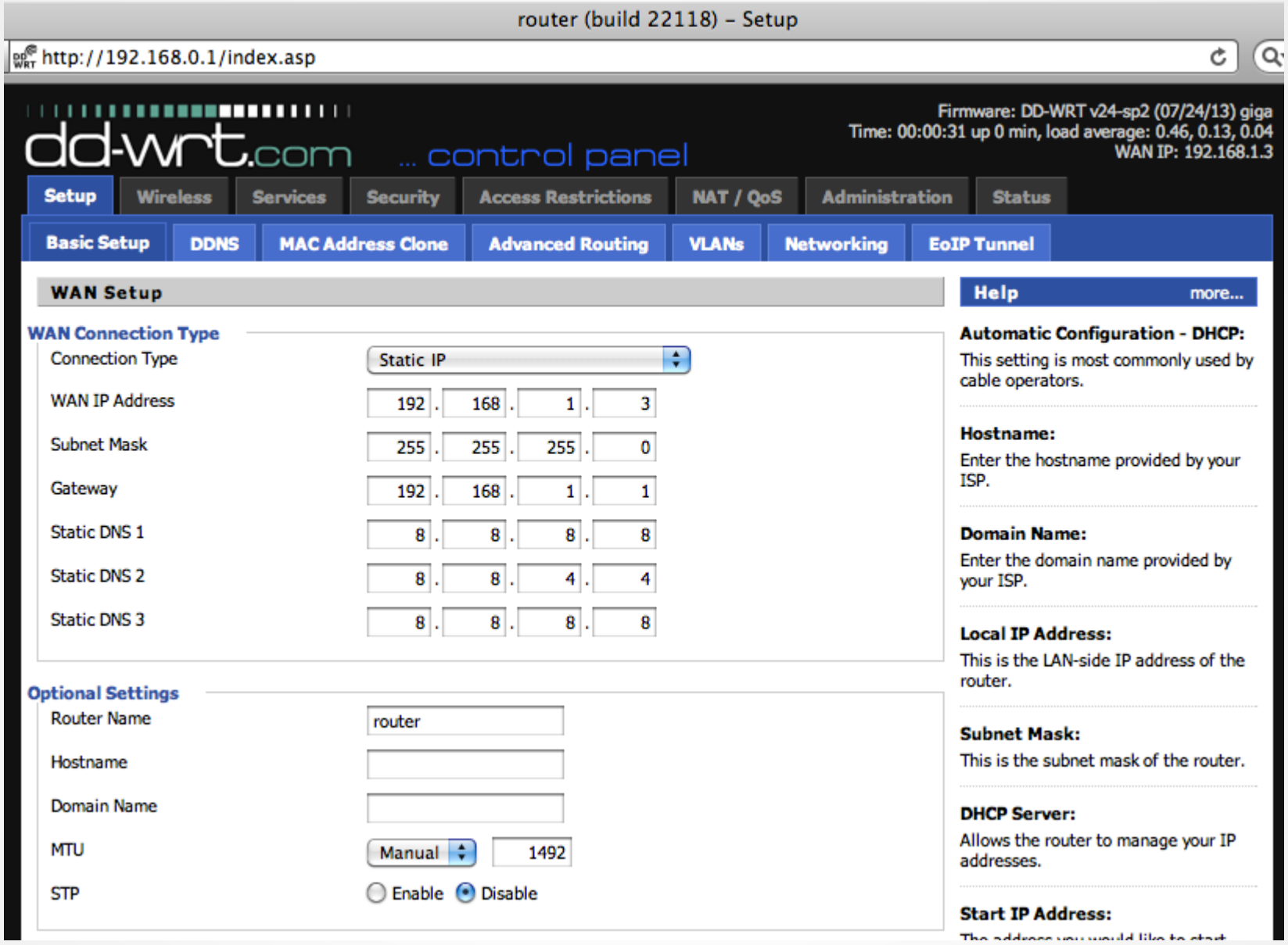
OSX

'**sudo route -n add -net 8.8.8.8/32 192.168.x.1**'

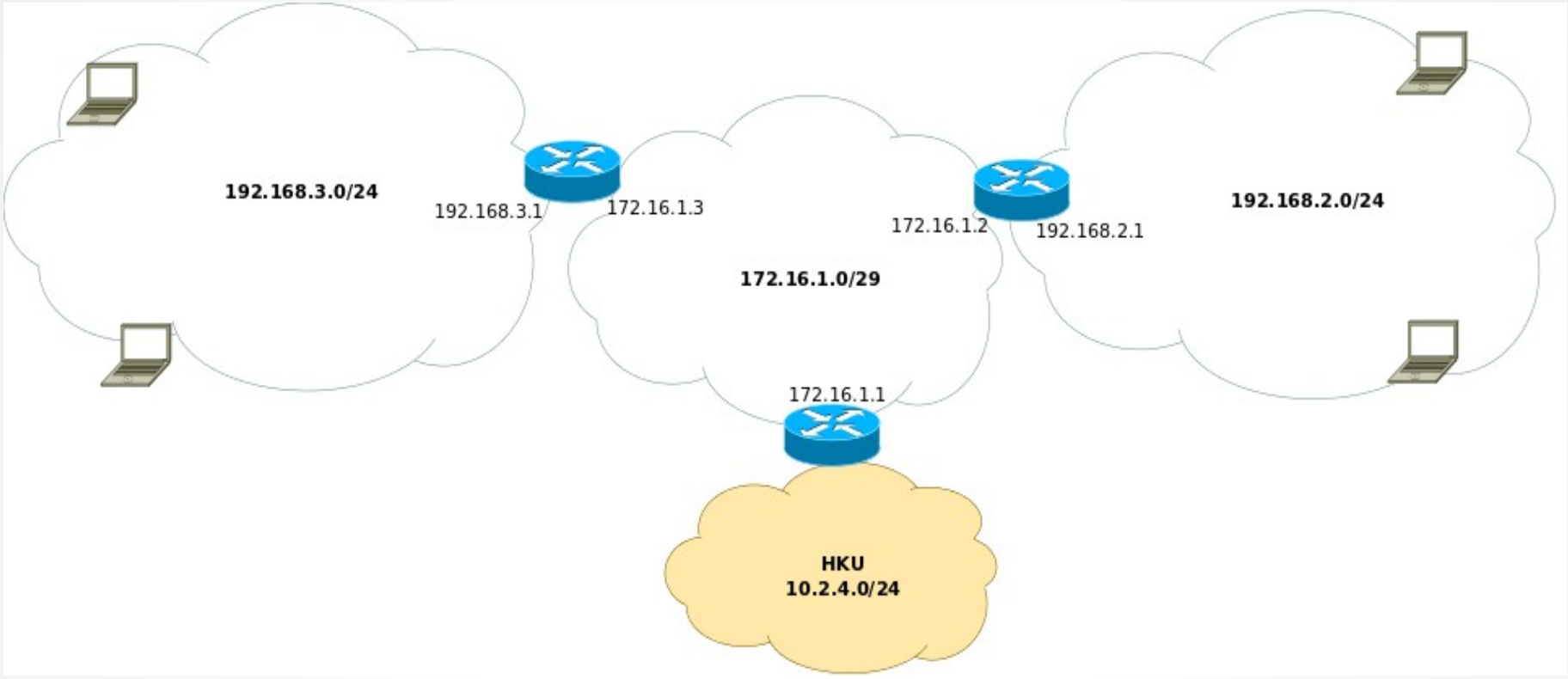
Windows

'**route ADD 8.8.8.8 MASK 255.255.255.255 192.168.x.1'**

Layer 3


router (build 22118) – Setup
http://192.168.0.1/index.asp
dd-wrt.com ... control panel
Firmware: DD-WRT v24-sp2 (07/24/13) giga
Time: 00:00:31 up 0 min, load average: 0.46, 0.13, 0.04
WAN IP: 192.168.1.3
Setup
Wireless
Services
Security
Access Restrictions
NAT / QoS
Administration
Status
Basic Setup
DDNS
MAC Address Clone
Advanced Routing
VLANs
Networking
EoIP Tunnel
WAN Setup
WAN Connection Type
Connection Type
Static IP
WAN IP Address
192 . 168 . 1 . 3
Subnet Mask
255 . 255 . 255 . 0
Gateway
192 . 168 . 1 . 1
Static DNS 1
8 . 8 . 8 . 8
Static DNS 2
8 . 8 . 4 . 4
Static DNS 3
8 . 8 . 8 . 8
Optional Settings
Router Name
router
Hostname
Domain Name
MTU
Manual
1492
STP
Enable
Disable
Help
more...
Automatic Configuration - DHCP:
This setting is most commonly used by cable operators.
Hostname:
Enter the hostname provided by your ISP.
Domain Name:
Enter the domain name provided by your ISP.
Local IP Address:
This is the LAN-side IP address of the router.
Subnet Mask:
This is the subnet mask of the router.
DHCP Server:
Allows the router to manage your IP addresses.
Start IP Address:

Layer 3



'ip route add default via 192.168.x.1'
'ip route add 0.0.0.0/0 via 192.168.x.1'
'sudo route -n add -net 0.0.0.0/0 192.168.x.1'
'route ADD 0.0.0.0 mask 0.0.0.0 192.168.x.1'

Layer 3

Linux

'**traceroute -n 8.8.8.8**'

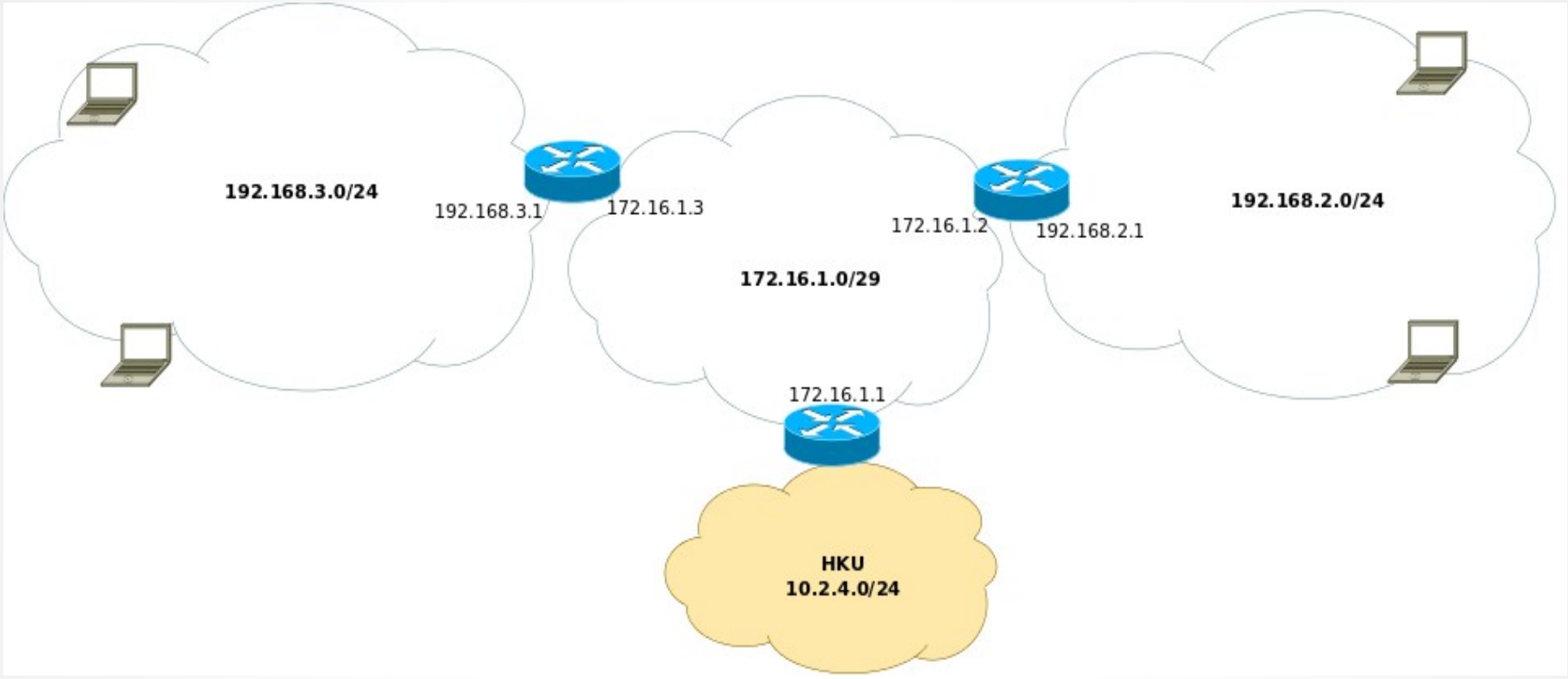
OSX

'**sudo traceroute -n 8.8.8.8**'
(also in Network Utility)

Windows

'**tracert 8.8.8.8**'

Layer 3


192.168.3.0/24
192.168.3.1
172.16.1.3
172.16.1.2
192.168.2.1
192.168.2.0/24
172.16.1.0/29
172.16.1.1
HKU
10.2.4.0/24

Layer 3

IP packets have a TTL (Time To Live)

This prevents infinite loops!

Watch the TTL attribute in the tcpdump output!

You have to use -v

Layer 3

This is all you need to know about how the internet works!!!

Remember internet and the network **does not go beyond layer 3**!

There are more features:

- Routing protocols like BGP/OSPF so your route table is managed
- Loop prevention through Spanning Tree, Link Aggregation, etc
- More specifics for Wireless networks aka 802.11*
- Redundancy and asynchronous routing

But let's focus on what we do on the network/internet

Layer 4

Layer 4

TCP/IP

As referred to very often however it's:

- IP = The IPv4 protocol *(protocol number 0)*
  - ICMP = Internet Control Messaging Protocol *(protocol number 1)*
  - UDP = User Datagram Protocol *(protocol number 17)*
  - TCP = Transmission Control Protocol *(protocol number 6)*

There are way more, see /etc/protocols!

Layer 4

ICMP

It controls the connection

If something goes wrong ICMP will tell you

0 Destination network unreachable
1 Destination host unreachable
2 Destination protocol unreachable
3 Destination port unreachable
4 Fragmentation required, and DF flag set
5 Source route failed
6 Destination network unknown
7 Destination host unknown
Etc

Used by:

- Routers
- Ping
- Traceroute

Layer 4

UDP

- It is simple,
- It is stateless

„fire and forget“

Usable for broadcast and multicast and very simple protocols

Used by:
- DHCP (port 67 and 68)
- DNS (port 53)
- OSC
- ...

Layer 4

TCP

- reliable,
- ordered,
- error-checked
- connection oriented

„it gets the message across reliably“

Not usable for broadcast and multicast, d'uh

Used by:
- SSH
- HTTP
- Lots more...

Layer 4

TCP & UDP ports

TCP and UDP have the same port system consisting of 65536 numbered ports

A program can listen on a port on your NIC!

Important port numbers:
0-1023 are reserved: (root only)

- TCP:22 = SSH
- TCP:23 = Telnet
- TCP:25 = SMTP
- UDP:53 = DNS
- TCP:80 = HTTP
- etc, see /etc/services

1024-65535 are unpriviliged

- TCP:1194 = OpenVPN
- UDP:1194 = OpenVPN
- TCP:3306 = MySQL
- TCP:3389 = RDP
- etc, see /etc/services

Layer 4

Important Services for your network:

DHCP: Dynamic Host Configuration Protocol

- Uses UDP port 67 (request) and port 68 (reply)
- Uses a handshake
  - Client broadcasts
  - DHCP server replies
  - Client request an ip address
  - DHCP server replies with an ipaddress
  - Client ACK the ipaddress
- More options are available through DHCP
  - Default gateway
  - Extra routes
  - Boot server (PXE/TFTP)
  - Many more

Layer 4

Important Services for your network:

DNS: Domain Name System

Resolves names to ipaddresses so you don't need the remember every ip address

Your network needs at least 1 DNS resolver

DNS is big but these are important to know:

- Every domain has a authorative name server (NS record)
- An hostname to an ip address is an A record
- An alias (to an A record) is a CNAME
- To find the mailserver of a domain you request an MX record
- SRV records are for more complicated services (Jabber/XMMP)

Layer 4

On your computer add DNS *(DNS Server 8.8.8.8)*

Linux

add the following to /etc/resolv.conf *(nano /etc/resolv.conf)*

search ws.ect.lan
nameserver 8.8.8.8

OSX

Network preferences - NIC - advanced - manual

Windows

ncpa.cpl

adapter - properties - internet protocol version 4 (TCP/IPv4)

Layer 4

DNS tests (first make sure you can ping the DNS server!)

Linux

'**dig www.hku.nl**'
'**host www.hku.nl**'

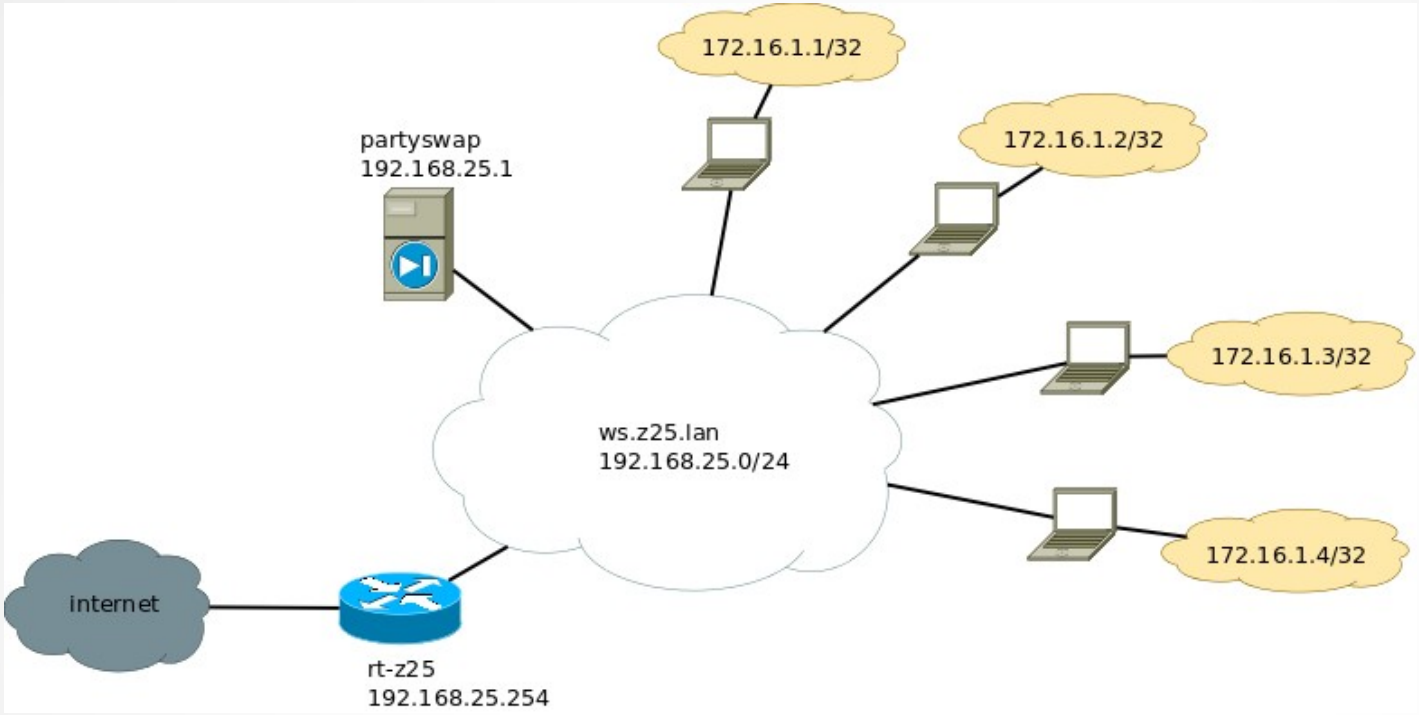
OSX

'**host www.hku.nl**'

Windows

'**nslookup www.hku.nl'**

Play time



ifconfig lo:1 172.16.1.1 netmask 255.255.255.255 up
echo 1 > /proc/sys/net/ipv4/ip_forward

Add route info for other hosts
route add -net 172.16.1.1 netmask 255.255.255.255 gateway 192.168.25.x

Trouble

We can ping to 8.8.8.8... from our host with ip 192.168.25.x

192.168.25.x is a **private address**, and IP packets addressed by them **cannot be transmitted** onto the public Internet

But how can 8.8.8.8 reply to us???

Trouble

NAT – Network Address Translation

„it is common to hide an entire IP address space, usually consisting of private IP addresses, behind a single IP address in another address space.

To avoid ambiguity in the handling of returned packets, a one-to-many NAT must alter higher level information such as TCP/UDP ports in outgoing communications and must maintain a translation table so that return packets can be correctly translated back.“ (wikipedia)

NAT:

- is a workaround for the exhaustion of IPv4 address space
- breaks real end-to-end connectivity across the internet
- makes it difficult to accept incoming connections
- Requires more capable routers in your home

Imagine a simple phone call....

Trouble

What's the solution.....

IPv6

We'll cover that next time ….

https://github.com/sphaero/workshops

TCPdump

„a powerful command-line packet analyzer“

„tcpdump - dump traffic on a network“

We'll use tcpdump to sniff the network in order to see what we are doing

TCPdump

Before we can do anything on the network we have to enable the network interface

Execute '**ifconfig eth0 up**' to bring your interface up

TCPdump

Access the second terminal on your computer (**ALT+F2**)

Login as root

Execute '**tcpdump -i eth0 -ntev**'

-i interface
-n no name resolving (DNS)
-t no timestamp
-e show link-level header (layer 2)
-v extra output (verbose)

Execute '**tcpdump -i eth0 -nte**'

Switch back and forth to the first (tty1) and second terminal (tty2) (**ALT+F1/ALT+F2**)

TCPdump

```
00:25:22:d8:dc:9f > b8:27:eb:eb:88:00, ethertype IPv4 (0x0800), length 110:
192.168.12.50.44717 > 192.168.12.90.4713: Flags [P.], seq 220:264, ack 516,
win 445, options [nop,nop,TS val 1852752 ecr 405490241], length 44

6c:40:08:90:9b:86 > ff:ff:ff:ff:ff:ff, ethertype IPv4 (0x0800), length 282:
192.168.12.118.17500 > 255.255.255.255.17500: UDP, length 240

00:25:22:d8:dc:9f > 08:96:d7:93:30:f6, ethertype IPv4 (0x0800), length 66:
192.168.12.50.35075 > 145.58.28.175.80: Flags [.], ack 737399, win 1444,
options [nop,nop,TS val 1937128 ecr 1193042399], length 0
```

OSI Model Packet Encapsulation

Thank you

 arnaud@z25.org

 twitter.com/sphaero

 github.com/sphaero