



Revision

CO1507 Introduction to Networking

Jonathan Edwards 2024

Network devices

- Host
- Switch
- Router
- Server
 - Web (http)
 - DHCP
 - DNS
- Wireless access point

Router Vs Switch

A **router** connected networks together

A router must maintain routing tables

A router is required in a Wide Area Network (WAN)

It works on IP addresses

A **switch** connected hosts together WITHIN a Local Area Network (LAN)

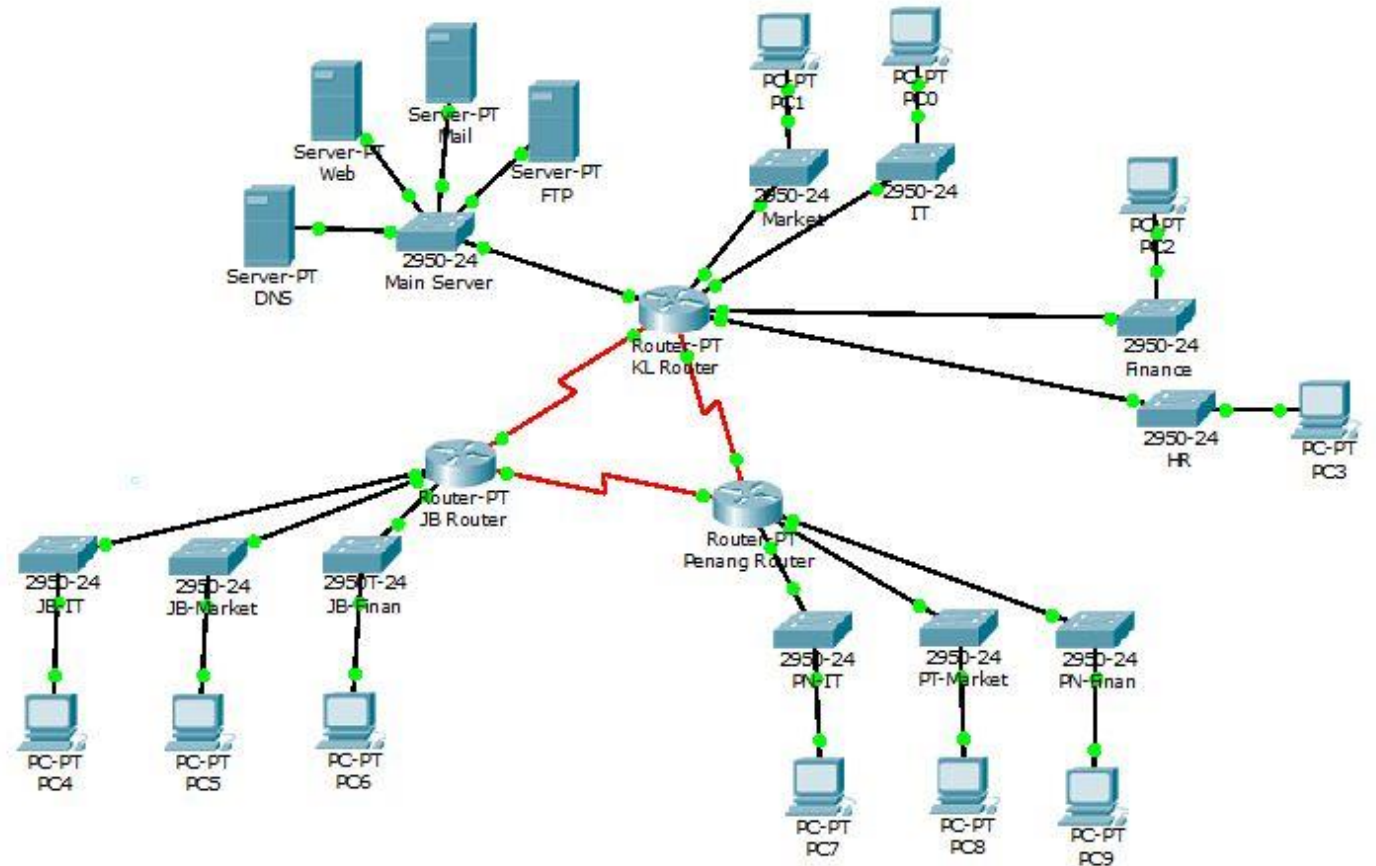
A switch forwards packets to the right port in a local area network

It works on MAC addresses

Network Types

- Local Area network (LAN)
 - Non-routed environment
 - One network address
 - One broadcast domain
 - No routers needed
- Wide Area network (WAN)
 - Routed environment
 - Multiple network / subnet addresses
 - Multiple broadcast domains
 - Includes routers

Wide Area Network



Addressing Requirements

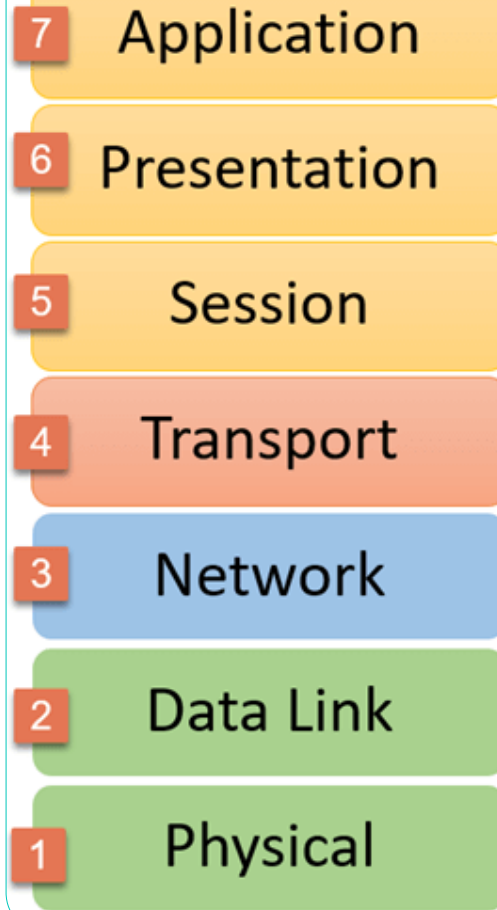
- **LAN** hosts need:
 - IP address and subnet mask
- **WAN** hosts need:
 - IP address, subnet mask **AND** default gateway
- If the **default gateway** is wrong or missing you will **not** be able to communicate with hosts on a **different** network (subnet)
- Communication **within** the LAN will be unaffected

OSI Model

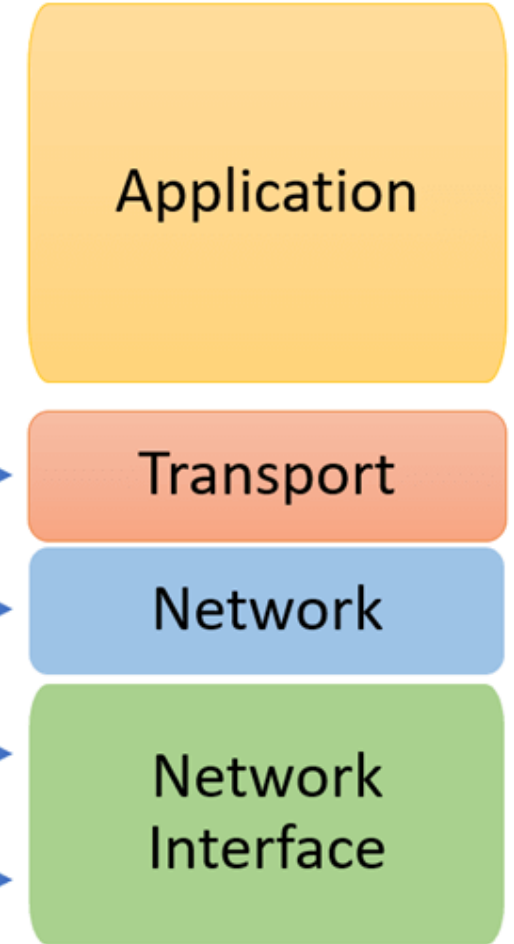
7	Application	Data	HTTP
6	Presentation	Data	Encoding, compression, decryption
5	Session	Data	Session management
4	Transport	Segment, Datagram	TCP/IP TCP UCP etc.
3	Network	Packet	Routing, ARP
2	Data-Link	Frame	Switching, MAC Address
1	Physical	Bit	Connections, Voltages

OSI Vs TCP/IP Model

OSI Reference Model



TCP/IP Conceptual Layers



© guru99.com

ARP

○ The ARP request is really saying:

“Who has the MAC address for this destination IP address?”

PC A broadcasts to all hosts on the network sending out a frame as seen below:

ARP Request	Destination IP address	Source IP address	Destination MAC address [broadcast]	Source MAC address
	192.168.10.5	192.168.10.2	FF-FF-FF-FF-FF-FF	6a-21-23-3d-27-11

ARP

- The PC with the IP address **192.168.20.3** will reply with its own MAC address using and ARP reply frame, notice how the broadcast address is replaced with **5c-45-21-e2-a7-23**
- **Note the destination address in the ARP reply becomes source address and vice versa.**

ARP	Destination IP address	Source IP address	Destination MAC address	Source MAC address
Response	192.168.10.2	192.168.10.5	6a-21-23-3d-27-11	5c-45-21-e2-a7-23

ARP

○ REQUEST:

ARP Request	Destination IP address	Source IP address	Destination MAC address [broadcast]	Source MAC address
	192.168.10.5	192.168.10.2	FF-FF-FF-FF-FF-FF	6a-21-23-3d-27-11

○ RESPONSE:

ARP Response	Destination IP address	Source IP address	Destination MAC address	Source MAC address
	192.168.10.2	192.168.10.5	6a-21-23-3d-27-11	5c-45-21-e2-a7-23

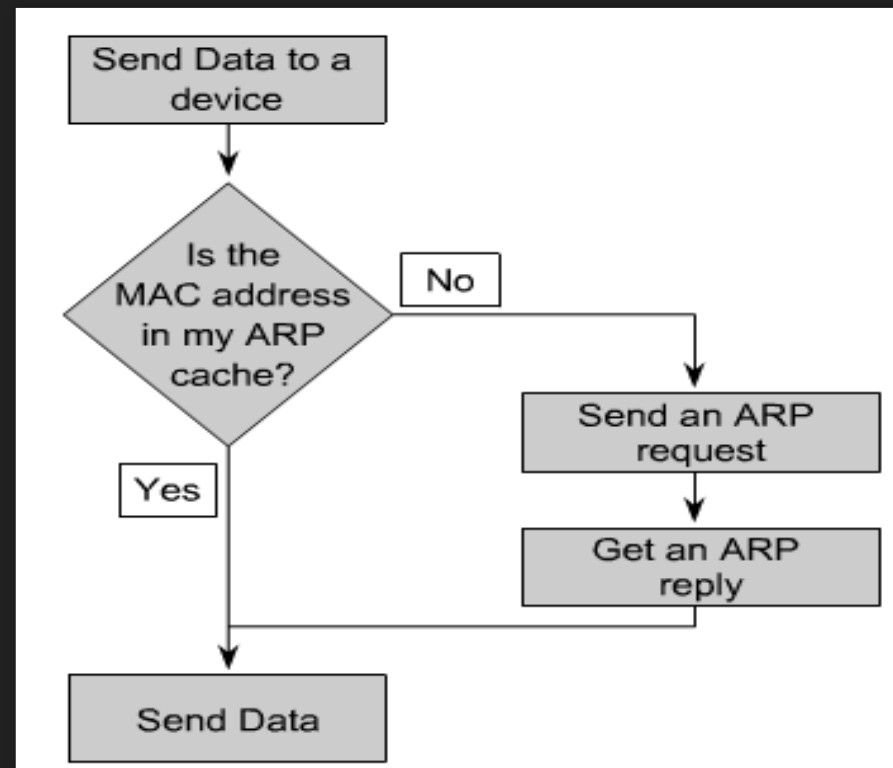
ARP

- Once PC A receives the destination MAC address, it inserts it in the frame for communication with PC B
- The destination IP address and the corresponding MAC address will then be stored in an ARP table as shown below

IP address	MAC address
192.168.10.5	6A-21-23-3D-27-11

- The above ARP table is from PC A. And now has an entry relating to PC D
- Every PC has its own ARP table to store data in
- From there on, every time PC A needs to communicate with PC D, it checks to see if the entry is in its ARP table. If it is, then it uses it, if not it uses a broadcast. See the flow chart on the next page

ARP Flow Chart



Custom Subnetting

- Remember a subnet mask simply tells the computer which bits (out of the 32 total) are the network address and which bits are the host ID
- Bits lining up with 1s are the network address, bits lining up with 0s are the host ID
- Default subnet masks only have 8 16 or 24 bit network addresses
- They only have 255 and 0 in them when written out in decimal e.g.
 - /24 = 11111111 11111111 11111111 00000000 = 255.255.255.0
- Custom subnet masks contain other numbers
 - /27 = 11111111 11111111 11111111 11100000 = 255.255.255.224

Network Address and Broadcast Address

- Spotting the network address and broadcast address with a default subnet mask is easy
- E.g. with 255.255.255.0 (/24)

- 192.168.1.**0** is a network address

Mask 11111111 11111111 11111111 **00000000**

IP 11000000 10101000 00000001 **00000000 - all zero**

- 192.168.1.**255** is a broadcast address

Mask 11111111 11111111 11111111 **00000000**

IP 11000000 10101000 00000001 **11111111 - all one**

It's easy because you just are looking for 0s or 255s

Network Address and Broadcast Address

- Spotting the network address and broadcast address with a custom subnet mask is tricky
- E.g. with 255.255.255.224 (/27)

- 192.168.1.**32** is a network address

Mask 11111111 11111111 11111111 11**100000**

IP 11000000 10101000 00000001 00**100000** - all zeros

- 192.168.1.**63** is a broadcast address

Mask 11111111 11111111 11111111 11**100000**

IP 11000000 10101000 00000001 00**111111** - all ones

It's hard because you can't spot obvious numbers when looking at the decimal

Though numbers that are powers of two or ONE LESS THAN a power of two should give you a hint

Every Subnet of 255.255.255.224 (/27)

- 8 subnets
- One occurs every 32 addresses
- 30 valid hosts on each subnet

- Subnet mask = 11111111 11111111 11111111 11100000
- Subnet mask = 255.255.255.224

- For the example we will use the network address 192.168.0.x

IP address	Last octet in binary host ID in red	
first subnet		
192.168.0.0	000 00000	network address
192.168.0.1	000 00001	first host
192.168.0.30	000 11110	last host
192.168.0.31	000 11111	broadcast address
second subnet		
192.168.0.32	001 00000	network address
192.168.0.33	001 00001	first host
192.168.0.62	001 11110	last host
192.168.0.63	001 11111	broadcast address
third subnet		
192.168.0.64	010 00000	network address
192.168.0.65	010 00001	first host
192.168.0.94	010 11110	last host
192.168.0.95	010 11111	broadcast address

255.255.255.224
(/27)

192.168.0.96	011 00000	network address
192.168.0.97	011 00001	first host
192.168.0.126	011 11110	last host
192.168.0.127	011 11111	broadcast address
fifth subnet		
192.168.0.128	100 00000	network address
192.168.0.129	100 00001	first host
192.168.0.158	100 11110	last host
192.168.0.159	100 11111	broadcast address
sixth subnet		
192.168.0.160	101 00000	network address
192.168.0.161	101 00001	first host
192.168.0.190	101 11110	last host
192.168.0.191	101 11111	broadcast address

255.255.255.224
(/27)

seventh subnet		
192.168.0.192	110 00000	network address
192.168.0.193	110 00001	first host
192.168.0.222	110 11110	last host
192.168.0.223	110 11111	broadcast address
eighth subnet		
192.168.0.224	111 00000	network address
192.168.0.225	111 00001	first host
192.168.0.254	111 11110	last host
192.168.0.255	111 11111	broadcast address

255.255.255.224
(/27)

Subnetting

Going in either direction..

- One **fewer** bit in the network address e.g. **/26** would give us:
 - Half as many subnets
 - Twice as far apart
 - Twice the addresses on each
- One **more** bit in the network address e.g. **/28** would give us:
 - Double the number of subnets
 - Half as far apart
 - Half the addresses on each

A Sample Question

- Is 192.168.1.15 /**28** a valid **host** IP address, or a **network** address, or a **broadcast** address
- How do you answer this?
- Work out the subnet mask:
 - **28 1s = 11111111 11111111 11111111 11110000 (255.255.255.240)**
- Look at the relevant octets in binary
(this is the last one at least in any question I will ask you because I won't ask about /23 or fewer bits)
- .240 = 1111**0000**
- .15 = 0000**1111** = **all ONES so the answer is it is a broadcast address**

A Sample Question

- Is 192.168.1.128 /27 a valid **host** IP address, or a **network** address, or a **broadcast** address
- How do you answer this?
- Work out the subnet mask;
 - **27 1s = 11111111 11111111 11111111 11100000 (255.255.255.224)**
- Look at the relevant octets in binary
(this is the last one at least in any question I will ask you because I won't ask about /23 or fewer bits)
- .224 = 111**00000**
- .128 = 100**00000** = **all ZEROS so the answer is it is a network address**

IPv6 Notation

IPv6 addresses are represented by **8 groups** of **16-bit** hexadecimal values separated by colons (:

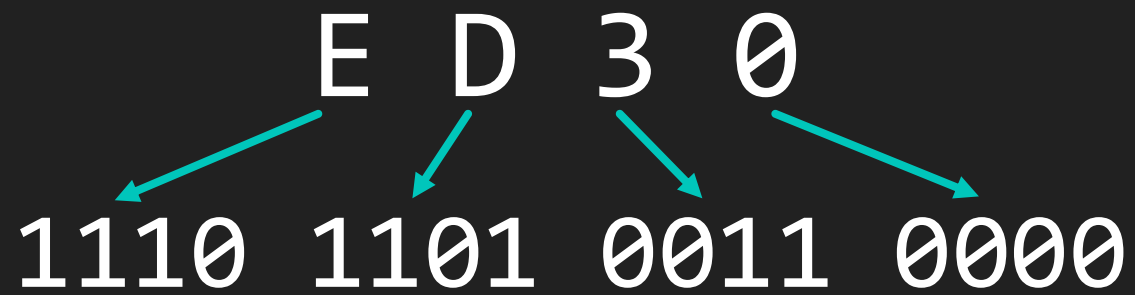
e.g. 2024:003A:CD3F:0000:023A:00CF:ED30:12FF

Leading zeros can be removed to give

2024:3A:CD3F:0:23A:CF:ED30:12FF

IPv6

2024:003A:CD3F:0000:023A:00CF:ED30:12FF



IPv6 Compression Summary

- Remove leading zeros

2024:003A:0000:0000:023A:00CF:ED30:12FF

To give :

2024:3A:0:0:23A:CF:ED30:12FF

- Remove longest section of adjacent zeros and replace with a double colon

2024:3A:0:0:23A:CF:ED30:12FF

To give:

2024:3A::23A:CF:ED30:12FF

Routing

- Dynamic routing - uses a protocol such as RIP, OSPF, etc.
- Static routing - programmed into the router manually
 - Recursive - specifies the **next hop** interface
 - E.g. `ip route 192.168.10.0 255.255.255.0 10.0.0.1`
 - Exit - interface - specifies the **interface** out which the packet should be forwarded by the router
 - E.g. `ip route 192.168.10.0 255.255.255.0 s0/1`
 - Default route (minimum precision) last resort
 - E.g. `ip route 0.0.0.0 0.0.0.0 10.0.0.1`
 - Or ..
 - E.g. `ip route 0.0.0.0 0.0.0.0 s0/1`

Linux

- 'UNIX-like' OS
- Distribution not a commercial package
- Distributed under GNU license hence GNU / Linux
- Examples include Ubuntu, Arch, Kali, etc. depending on user preferences or needs
- Powers many web servers, mobile devices, embedded devices, raspberry pi etc
- Command line is important because servers usually interfaced with remotely via SSH etc

Permissions

```
chmod file_name 754 -R
```

754

111101100

rwXr-Xr--

Owner Group Others

R = read
W = write
X = execute

7 = rwx
5 = r-x
4 = r--

-R = recursive

Other Common Commands

- `ls` *# list content of pwd*
- `pwd` *# present working directory*
- `ls -a` *# list content of pwd including hidden files*
- `ls -l` *# list content of pwd with long details*
- `rm` *# remove (delete)*
- `rmdir` *# remove directory (empty directory)*
- `mkdir` *# make directory*
- `touch` *# create file or if exists update timestamp*
- `cat` *# concatenate file to screen buffer*
- `grep` *# global regular expression and print*
- `less` *# terminal file perusal and searching*
- `ip addr` *# list ip addresses of interfaces*

This Weeks Lab

- Practice exam style questions
- Please complete to aid your revision
- Ask your tutor to go through questions you are stuck on