Lecture 4

Layered Communication Models

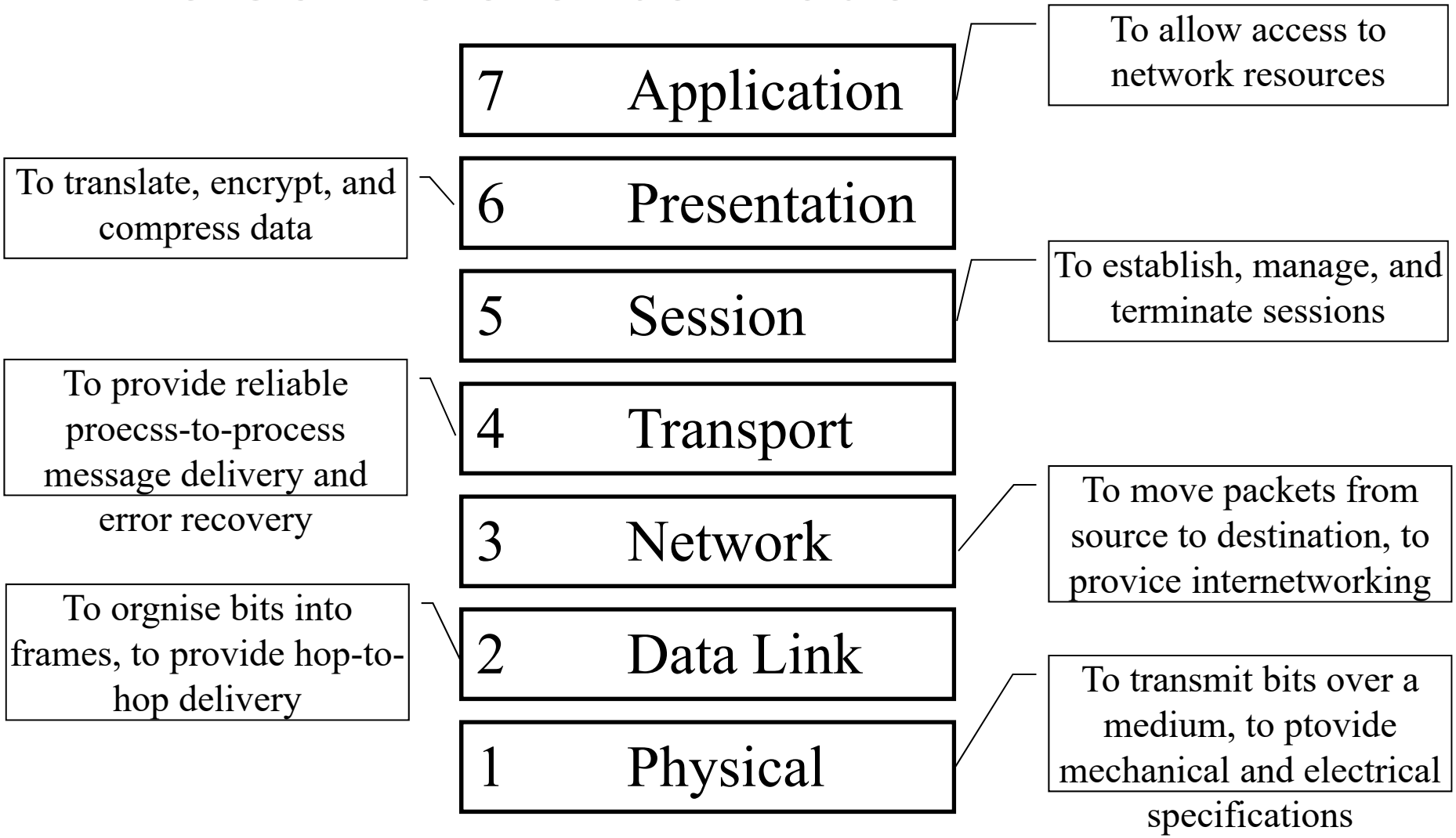
- In the 70's and 80's a number of vendors produced hardware for computing networking which could only work in homogenous computing environments and with particular kinds of equipments.
- Examples include IBM's networking technologies such as SNA, Apples Appletalk and even DEC's networking infrastructure.

- As a result a model was developed for communications which clearly articulated components of such system which could be modified within its parameters and have no effect on the remainder of the system.
- Layered models typically provide
  - independence.
  - flexibility.
  - standardisation.
  - simplified implementation and maintenance.

The OSI Reference Model

- Open Systems Interconnection (OSI) Reference Model
  - An ISO standard that covers all aspects of network communications
  - First introduced in the late 1970s
  - Is NOT a protocol
  - Is a model for understanding and designing a network architecture

The OSI Reference Model

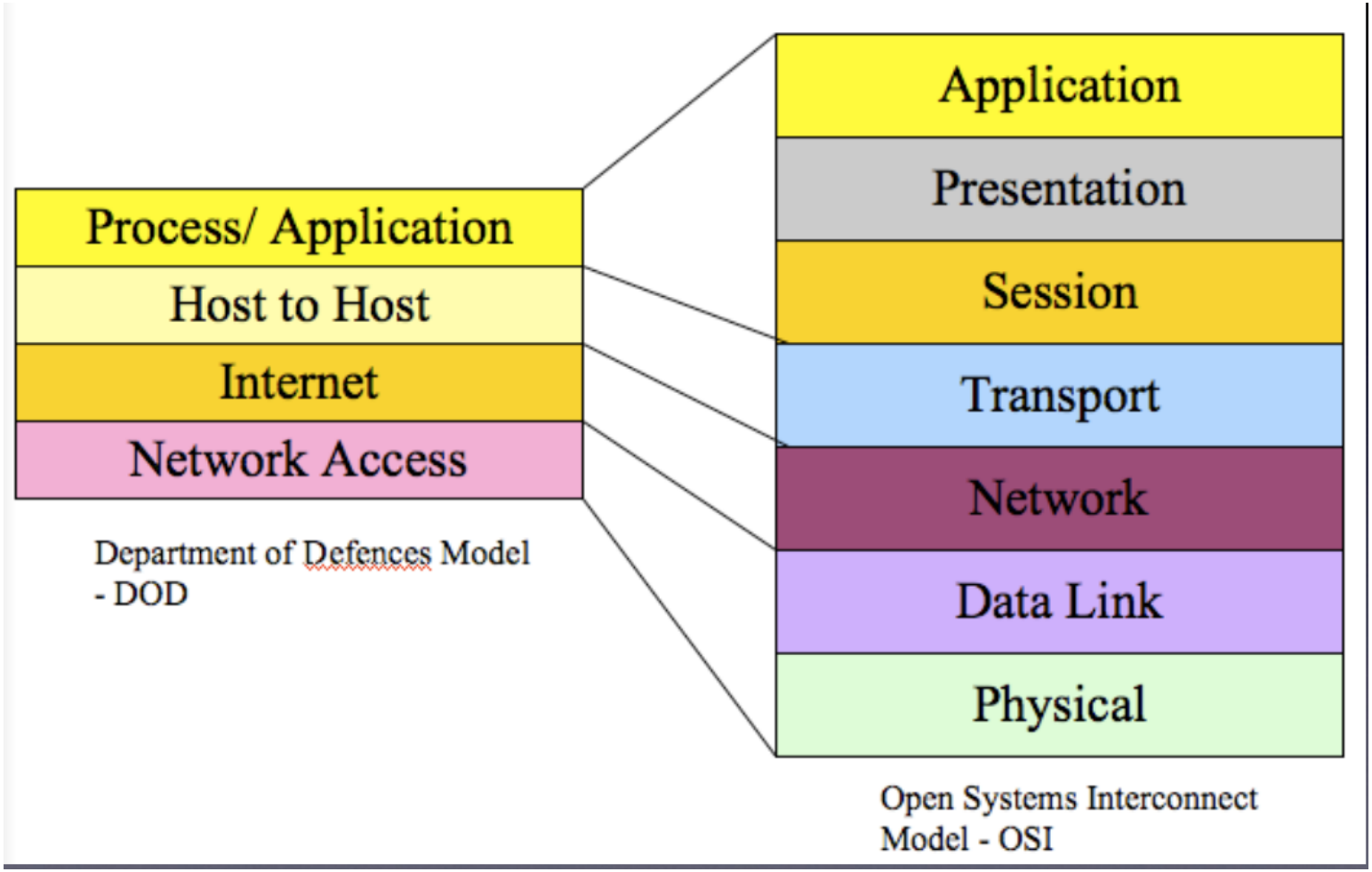


- Here is a brief description of each layer.
- Physical Layer (layer 1)
  - Provides the physical connection between the computer and the network wiring.
  - It specified cable pin assignment, voltage on the wire etc.
  - Data is transferred in bits.

- Datalink Layer (layer 2)
  - Provides the packaging and unpackaging of ‘bits’ of data for transmission - we commonly call this the structure of the hardware packet.
  - Some basic error checking is performed on transmission data.
  - There is also what we know as hardware level addressing at this layer.
  - Data is transferred in Frames. Things such as ATM and 802.3 Ethernet operate at this layer.

- Network layer (Layer 3)
  - Provides routing of data through the network (end to end). In this layer the most appropriate route is chosen for the data.
  - The layer is also concerned about packaging data so it can be routed between two points and addressing.
  - The data is referred to as datagram's/packets at this layer. IP, and IPX (Novells Internetwork Packet Exchange) are examples of protocols at this layer.
- Transport Layer (Layer 4)
  - This layer is largely concerned with error correction and synchronization (virtual circuit). TCP is an example of a protocol that operates at this layer.

- Session Layer (Layer 5)
  - The session layer is concerned with the establishment of connections between two points. It ensures a high level of reliability.
- Presentation Layer (Layer 6)
  - Basically involves the translation of data formats i.e. ASCII, Unicode etc. You may also see compression at this layer e.g. streams module implemented using gzip. Has no interaction with the user.
- Application Layer (Layer 7)
  - Provides functionality to the user, security and access to resources. Dependant on all layers beneath it. Examples include SMTP, POP, IMAP etc.

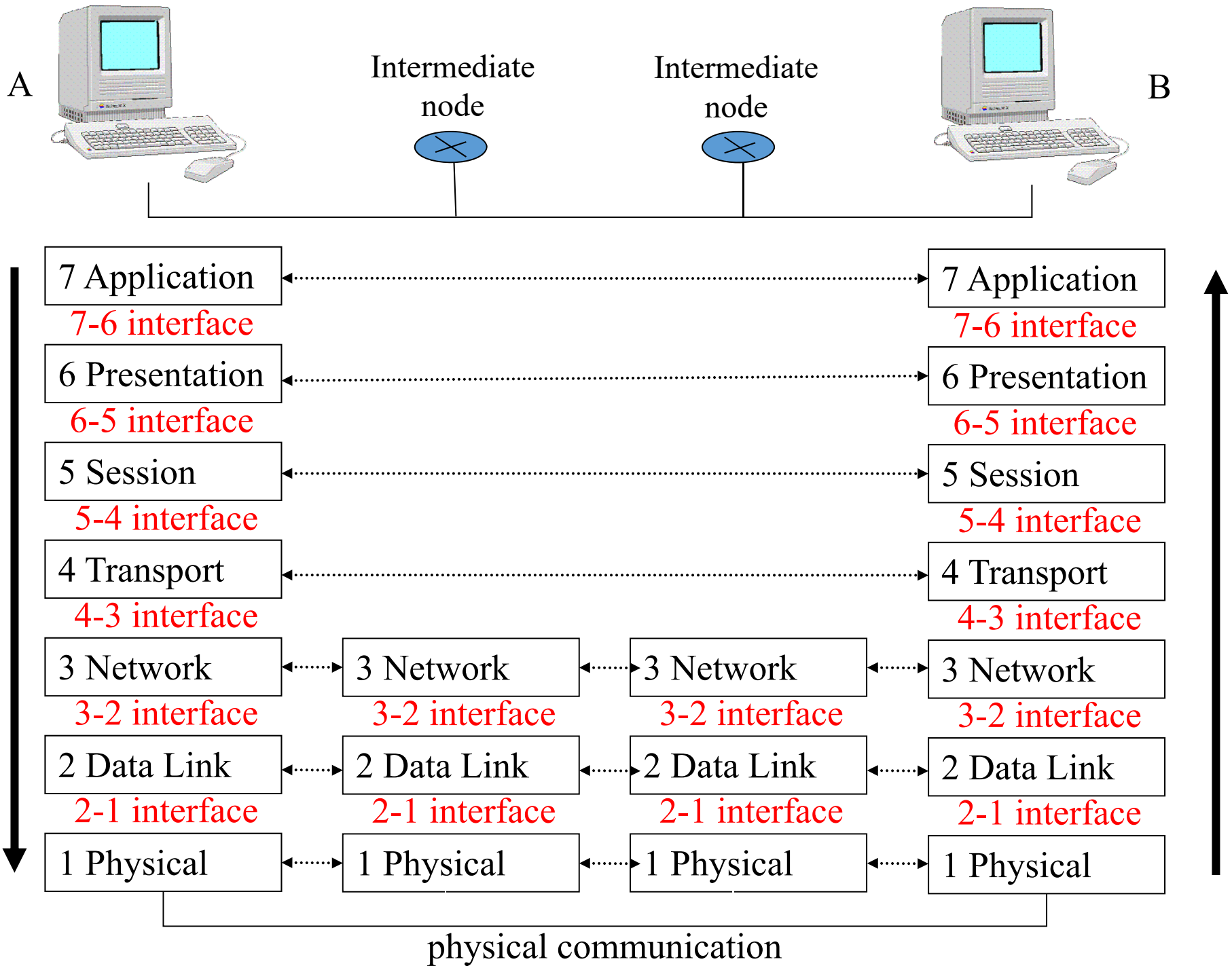

Process/ Application
Host to Host
Internet
Network Access
Department of Defences Model
- DOD
Application
Presentation
Session
Transport
Network
Data Link
Physical
Open Systems Interconnect
Model - OSI

Protocol data units (PDU)

- For data to move from source to destination it travel through each layer in the OSI model. Each layer communicates with its Peer Layer.
- Between layers Protocol Data Units are converted through a process of encapsulation/ de-encapsulation.
- For a layer to provide a service a upper layer it adds header information to perform its function. This is called encapsulation. (going down the OSI model)
- When a layer receives data, (going up the OSI model) it de-encapsulates the incoming stream into the layers PDU's, typically done by removing headers and checking before passing upwards.

Each layer's PDU

Layer 7 - Application	Data
Layer 6 - Presentation	Data
Layer 5 - Session	Data
Layer 4 - Transport	Segment
Layer 3 - Network	Packet
Layer 2 - Data Link	Frame (Media Dependant)
Layer 1 - Physical	Bit


A
Intermediate node
Intermediate node
B
7 Application
7-6 interface
6 Presentation
6-5 interface
5 Session
5-4 interface
4 Transport
4-3 interface
3 Network
3-2 interface
2 Data Link
2-1 interface
1 Physical
physical communication

Transmission at Layer 1

- Layer one is concerned primarily with media and how to send bits (0 and 1) over the wire or air.
- There are three ways in which this is done.
  - Voltage as is the case in Copper media.
  - Light as is the case in Optical media.
  - Radio waves over a spectrum as is the case in Wireless media.

- At this layer we tend to find cable specifications specified by organisations such as TIA and EIA. Some common examples of cabling standards include Cat 1 (Voice Grade), Cat 3 (10Mbps), Cat 5 and Cat5e (Data Grade - 100/1000Mbps).
- At this layer we also see rules of thumb for example;
  - 10 Base T
  - Refers to an Layer 2 Ethernet standard which operates at 10 Mbps, is base band (digital) and uses Twisted pair. Not more than 100m.

- Others include:
  - 10Base2, (10 Mbps over Coax Media (thinnet), Maximum of 200m)
  - 10Base5, (10 Mbps over Coax Media (thicknet), maximum of 500m)
- In general think of this layer a defining physical standards for how the media works.
- These rules are defined to minimize attenuation over the media so everything actually works.

Standards at the Data Link Layer - Ethernet and ATM

- Again standards are really important - as different vendors and organisations have different beliefs and opinions on the way things should be done.
- The IEEE have defined a number of the standards which describe how hardware can be used to group and transmit data in a sensible/ orderly fashion.

Ethernet

- A Ethernet frame may be up to 1500 bytes in size and contains a header and data.
  - 1000BaseX allows for systems to negotiate a larger maximum frame size if they want.
  - This size is commonly known as a MTU (Maximum Transfer Unit) and is dependant upon the media.
- Each packet has a source and destination address of 48 bits.
- Layer 2 of the OSI model defines addressing for hardware on a single network segment.

- Special addresses allow for broadcast (all 1s).
- Ethernet may be shared media but more commonly is point to point via a switch.
- Ethernet can be Half or Full duplex on twisted pair or fibre.

- Performs amazingly well, even on shared media when network built to correct specifications.
- Most problems caused by out of specification networks.
  - Length restrictions must be adhered to.
  - Avoid electrical interference from power, lights etc.
- Predominant network technology.
- Ethernet uses CSMA/CD (Carrier Sense Multiple Access/ Collision Detection) to detect and handle collisions.

- When two nodes transmit at the same time the following happens:
  - Prior to transmission a node looks to see if anyone is sending data (frames).
  - Nodes transmit data (frames)
  - Whilst transmitting the nodes also listen for other data on the segment.
  - If someone else is communicating at the same time then a collision occurs.
  - Devices transmitting will see the collision and then back off for a random period of time.

Ethernet Naming Rules

- The term Ethernet refers to a family of networking technologies including Fast Ethernet, Gigabit Ethernet and 10Gb Ethernet.
- The most common names are below, and are under the IEEE 802.3 standard:
  - 10 Base 2 (IEEE 802.3a)
  - 10 Base 5 (IEEE 802.3)
  - 100 Base T (IEEE 802.3i)
  - 1000 Base TX (IEEE 802.3X)

- Ethernet encapsulates data from high layers in the OSI model into frames.
- Within the Ethernet standard a standard called Media Access Control (MAC) defines how to encode information into frames and how to transmit it over the wire.

Ethernet Frame IEEE 802.2

- The lowest level of addressing is dictated by the network hardware and is encoded in Ethernet frames.
- Ethernet devices (NIC's - *Network Interface Card*) are assigned unique 6 byte addresses (48 bits) at the time of manufacturing. (sometimes this isn't true)
- For example the Ethernet controller on my Macintosh has the address 00:0a:95:9e:cd:76. This is used to uniquely identify my computer on the local network also known as segment.
- It is used by the link layer to transmit frames and helps in routing.

Layer 3 - Network Layer. The Internet Protocol

- The contents of a frame is a packet - an IP (RFC791) packet generally has the following elements.
  - Source Address.
  - Destination Address.
  - Time (TTL - time to live).
  - Options (define extra behavior).
  - Checksums.

- IP packets generally contain all the information essential to send a message to a destination.
- When a computer is preparing to transmit data the data is broken down into packets (layer 3), which are then framed (layer 2) and transmitted over a particular media.
- Framing means adding extra information.
- You need to understand that at the Network layer other protocols may exists e.g. IPX, Appletalk. Things like ICMP also operate at this layer.

- IP as a protocol also defines routing.
- The internet is a collection of networks which are interlinked using a number of varied links.
- Routing is the notion of moving a packet from one network to another and ultimately its destination.
- Each packet is independent and is routed independently.
- Packets may arrive via different routes and in any order.
- Packets may have to be transmitted as multiple frames with each frame containing an IP fragment. Frames may arrive out of order or be missing.

- If a packet cannot be assembled inside a reasonable time, it is discarded.
- There is no acknowledgement of packets or fragments.
- If you want reliability, you need higher levels in the protocol stack to arrange for acknowledgement and/or retransmission of packets.
- If whole IP packet is discarded.
  - The source must retransmit the entire packet.
  - There is no way to retransmit just the missing fragments.

- In addition to defining the layout of data in a packet, IP defines how hosts on a network are identified.
- Specifically IP defines a concept called IP addresses, which are used to uniquely identify hosts on a network or broader internet.
- The addressing scheme is hierarchal in nature.
- That said the hardware also plays a significant role here too.

IP addressing

- IP addresses are used to uniquely identify hosts on a network.
- Currently most use the IPv4 protocol which is 4 byte address. This means there can be a maximum of 2^{32} Nodes on the Internet (this can be fudged).
- The address is divided into two parts - the network and host.
- The network component identifies the grouping of addresses the host belongs too.

- The addresses are written in 4 octets (32 bits in groups
- 8) e.g.
  - 130.130.68.12
- A special address exists called *loopback* which refers to the host *127.0.0.1*

- A network controller can have one or more IP addresses bound to it.

IP address structure

- IP addresses have been broken into classes (groupings) since the dawn of time.
- The class determined the size of the network and host portion of each address.
- This has changed though due to the fact that addresses were being wasted.

Classful addressing

Class	1st Byte	Format	Use
A	1 - 126	N.H.H.H	Reserved for DOD
B	128 - 191	N.N.H.H	Large sites i.e. Universities
C	192 - 223	N.N.N.H	Commercial and Personal
D	224 - 239	-	Multicast
E	240 - 254	-	Experimental

Classless addressing

- Class A, B and C once had fixed subnet masks. This thus resulted in wastage of addresses.
- CIDR (Classless Inter-Domain Routing) was created allowing IP addresses to be broken down into any manner. Same can be said for VLSN (Variable Length Sub netting).

- Typically the netmask is bitwise AND with the IP address to identify its network address i.e. which grouping it belongs too.
- For example lets consider the example
  - 130.130.68.1
  - with a netmask of 255.255.255.0

- If we write it down on paper we get (notice the 4 octets):

  10000010 | 10000010 | 01000100 | 00001100

  AND

  11111111 | 11111111 | 11111111 | 00000000

  Produces

  10000010 | 10000010 | 01000100 | 00000000

  Meaning the network address is:

  130.130.68.0

ClDR notation

- An classless IP address is often represented using the slash notation, which is the IP address followed by a slash and the length of the mask, for example
  - 202.23.23.44/22
- The number of hosts in a subnet is calculated by using the following formula;
  - $2^{32 - \text{maskbits}} - 2$

- Broadcast address is reserved for each IP subnet. The address is used to speak to all IP's on the subnet with one packet.
- You can compute the broadcast address by bitwise OR'ing the IP address with the one's compliment of the mask.
- The host address bits are always set to 1. It is always the highest address in the subnet.

Calculating parameters

- Lets imagine we have the IP address 192.168.40.31 with the mask of 255.255.255.0 (/24).
- Lets calculate the network address, broadcast address

Subnetting

- By looking at an address we can identify what class it belongs to e.g. 151.30.22.12 is a class B network /16.
- Sometimes though these groupings are too restrictive. You may want to break a group of addresses into smaller chunks.
- To do this you have to choose the number of subnet you want OR the number of hosts you want in each subnet.

- Lets imagine we have the address space 192.168.18.0/24. This is a class C (Private) network.
- Lets say we want to divide it into 8 subnets.
- In order to do this we work out how many bits are needed to represent 8.
- In this case $2^3 = 8$.
- We need three additional bits to represent the subnet so we simply add 3 to the mask.

- The result is 8 subnet with the network addresses of 192.168.18.*n*/27.
- We now need to find the 8 values of *n* which identify each subnet.
- To do this we need to work out how many hosts we will have in each subnet.
  - $2^{32 - 27} = 32$
- However each subnet will have 30 hosts in it. (network and broadcast)

- This thus means that our class C is now broken down into 8 smaller subnet.
  - 192.168.18.0/27
  - 192.168.18.32/27
  - 192.168.18.64/27
  - 192.168.18.96/27
  - 192.168.18.128/27
  - 192.168.18.160/27
  - 192.168.18.192/27
  - 192.168.18.224/27

Supernetting

- Lets now consider a different example. In this example lets say we want to have at least 1400 hosts in each subnet.
- How many bits will we need to represent this - 11 (2^{11} = 2048).
- We now know that for a subnet to have 2048 address(a little more than we need but the only game in town) - we need 11 bits.

- Now if we have a class A or B - we have more than enough bits for the host. All you do is simply choose one network address and tell it that its now a /21.
- However sometimes we may need to use an aggregate of Private Class C's e.g. 192.168.*n.n.*
- *We know we need 21 bits for the network.* To do this we write down the subnet mask.

  11111111.11111111.11111000.00000000
- Once we have done this - we want to find a network address that does not spill into the last 11 bits.

- These would be;
  - 192.168.0.0/21
  - 192.168.8.0/21
  - 192.168.16.0/21
  - 192.168.24.0/21
  - . . .
  - 192.168.248.0/21
- Each of these networks has room for 2046 hosts. In this example we combined several private class C networks to form a larger network.

Network allocation

- Allocation of networks is controlled by organisations such as ICANN.
- In Asia Pacific we depend on APNIC to allocate addresses.

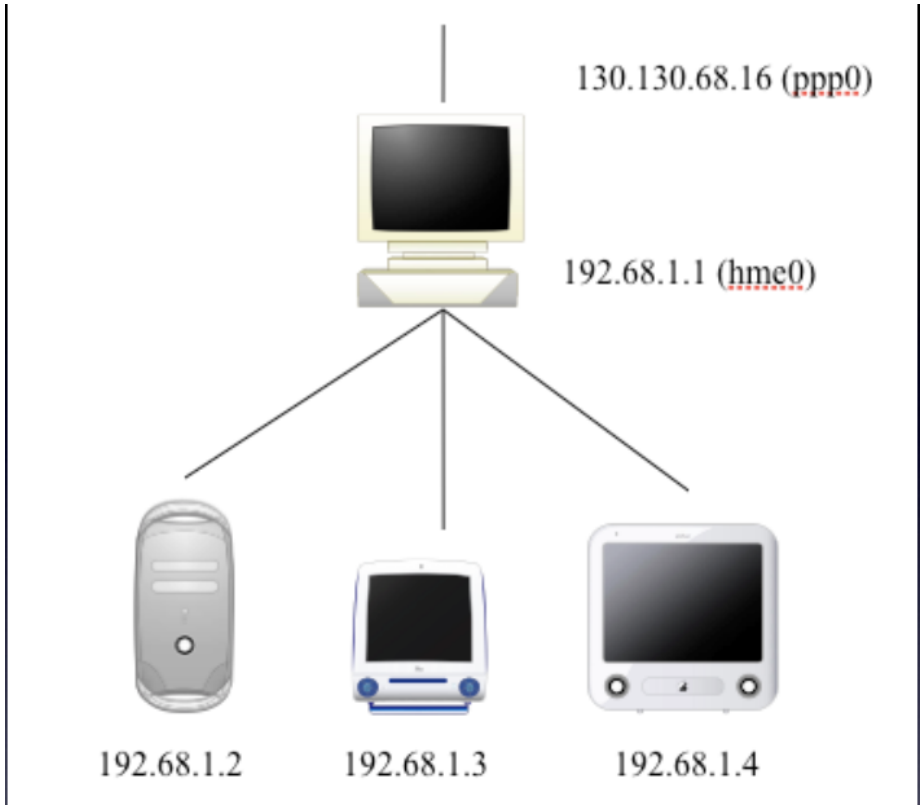
Private addresses

- RFC1918 defines a class of addresses which are Private.
- This thus means anyone can use them in there networks.
- The only problem is the are not routable i.e. they can not be reached from the Internet - for the very reason they are private.

  - **192.168.0.0 - 192.168.255.255** (65,536 IP addresses)
  - **172.16.0.0 - 172.31.255.255** (1,048,576 IP addresses)
  - **10.0.0.0 - 10.255.255.255** (16,777,216 IP addresses)

Network Address Translation

- What if you had a private network but wanted to talk to the Internet?
- you need a router (gateway) running a piece of software e.g. *ipf* (BSD Firewall Software).



- The machine providing the NAT service would be multi-homed i.e. have multiple network interfaces.
- In the previous diagram you saw that gateways machines Ethernet (hme0) interface has the address 192.68.1.1
- The gateway machine is connected to the Internet with a modem (ppp0). The address of the interface is 130.130.68.16 with a mask of 255.255.255.0.

- Network Address Translation works by rewriting packets.
- When a packet is sent from a host in the internal network it source address is rewritten by the gateway (if destination is outside).
- The gateway maintains a list of all rewrites so responses can be forward on.
- The outgoing packets port number may be altered (originating) allowing many internal addresses to be mapped onto one public address.

Routers and Routing Protocols

- Routers can determine the best route between two or more networks. A router operates at the network layer of the OSI model
- Because routers work at the network layer they do not care about Link layer protocols. Hence you can have a router with multiple interfaces using different protocols.
- Routers are commonly known as Gateways or forwarders.

IP Routing

- A router is a device for routing IP packets.
- It assembles the IP packet from incoming fragments, does a checksum validation, then sends the packet to the next destination, possibly as fragments.
- Actually, modern switch/routers may do cut through switching of fragments but conceptually this is the same.
- Each link in the network may have differing low level protocols (Ethernet, gig Ethernet, ATM, FDDI, SLIP, PPP, IP over SONET) with differing frame sizes.
- Thus a packet may be fragmented on some links but not others.

- Before an IP packet can be sent to its destination, a route must be determined.
- There are only 2 possibilities.
  - The destination address belongs to a machine on the same subnet as the source. The packet can be sent directly to the destination.
  - The destination address belongs to a machine on a different subnet from the source. The packet must be sent to a router which will arrange for the packet to be forwarded, possibly via additional routers, to the destination.

- Routers are also called gateways (they gateway from one network to another).
- A route maps a destination IP address to a local network (interface) or to a router.
- If the destination is not on a locally connected network, the route table will indicate a router to send the packet to.

- Remember that IP routing does not require any entity to know the entire network.
- A route in this case is not a path through the network to the destination.
- It is a single step to the destination or a single step to a router which will send the packet further on its way.

A simple example

- Lets consider a real simple example:
  - Machine A has a single interface with address 130.130.68.4 (netmask 255.255.255.0)
  - It wants to send a packet to 130.130.68.36
- The first step is to work out what network. Apply subnet mask to determine destination network.
  - 130.130.68.36 & 255.255.255.0 = 130.130.68.0

- The next step is to find destination in route table.
- Simplified route table is generated by the *netstat –r* command.

Destination	Mask	Gateway	Flags	Interface
130.130.68.0	255.255.255.0	130.130.68.4	U	hme0
default	0.0.0.0.	130.130.68.254	UG	

  - This sort of small route table is typical of hosts.
  - Since they usually have a single interface on a single subnet, they only require a few entries in the table and a default route to the router off the subnet.
  - The special entry "default" matches anything.
  - It is only used if no other match is found.

- First entry matches
- This entry has no G flag (means is not a gateway (router)) so it must be an interface. Thus destination is directly reachable.
- We can now send out an ARP packet on the interface to find the MAC address of the destination.
  If we get a reply, we can send the IP packet via Ethernet packet(s) to the destination.
- If we get no reply from the ARP request, the target machine cannot be reached.
  - Return a “destination unreachable” error code back to the client application.

- Lets consider a slightly different example where the destination is 192.56.24.5.
- Mask to find the network
  - 192.56.24.5 & 255.255.255.0 = 192.56.24
  - The entry "default" matches, since it matches everything.
- In this case, the G flag indicates the entry refers to a router. The address of the router is 130.130.68.254
- So, now we have the interface, send ARP request to get the MAC address of the router.
- In the more likely case that the destination address is not the router itself, it will use a very similar algorithm to send the packet to someone else.

A route table example

- Here is a simple routing table on a BSD/Linux machine. The machine in question has multiple interfaces.

```
# netstat -r
Destination        Mask          Gateway          Flags        If
132.236.227.0   255.255.255.0 0.0.0.0          U            e0
default         0.0.0.0       132.236.227.1    UG           e0
132.236.212.0   255.255.255.192 0.0.0.0        U            e1
132.236.220.64 255.255.255.192 132.236.212.6 UG            e1
127.0.0.1 255.255.255.255 255.0.0.0            U            lo0
```

- What do you think this network looks like diagramatically?

- The fourth route says to reach network 132.236.220.64/26 we must send packets to gateway 132.236.212.6 via interface e1.
- We can add routes by using the route command;

  *# route add -net 132.236.220.64 netmask 255.255.255.192 *

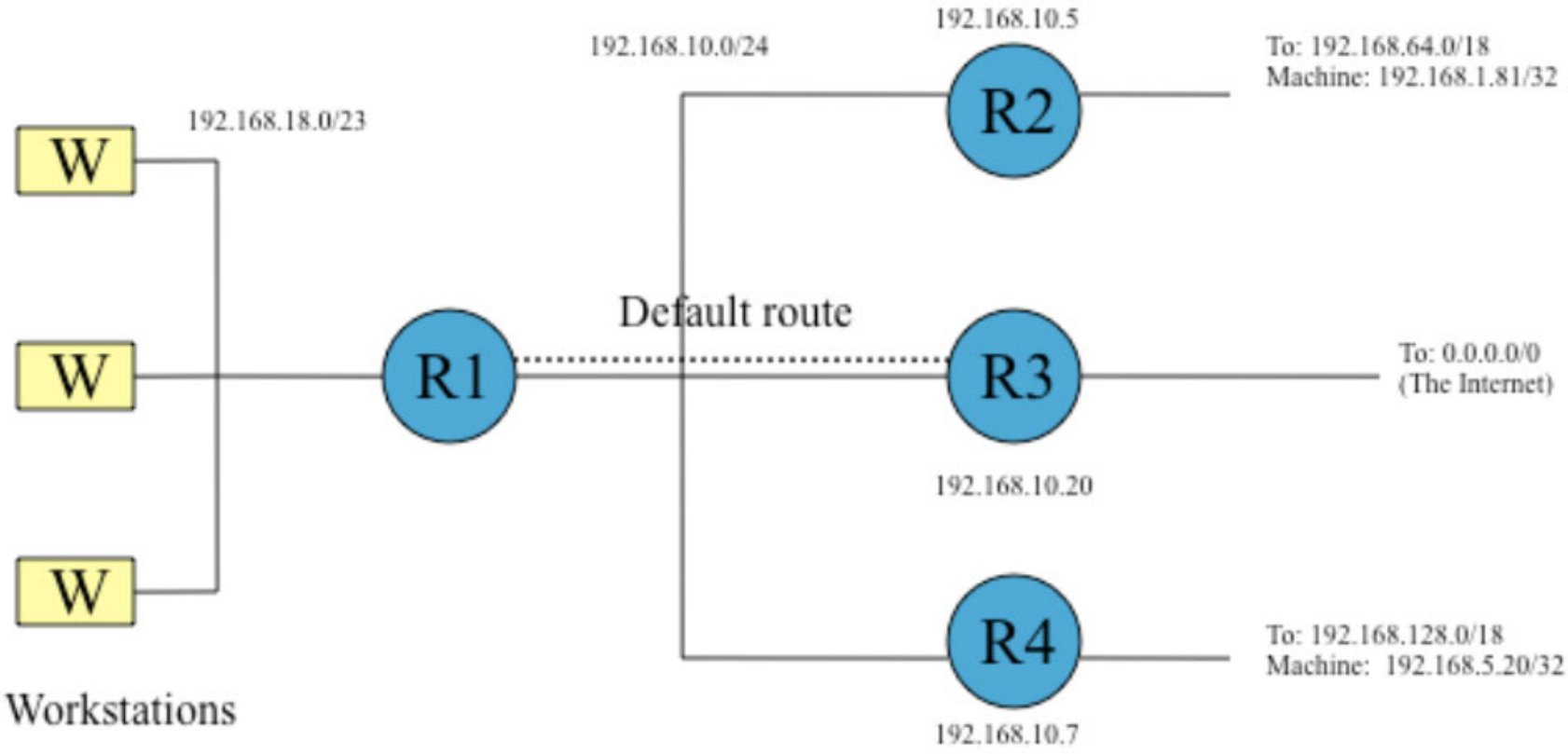
  132.236.212.6

  # route add default 132.236.227.1

- Routes can be dynamic - i.e. routers can learn from one another.
- To do this we typically use a daemon called *routed* or *gated*.
- These daemons communicate with one another to find out the routes to distant destinations.

Final example

- Lets consider the following network topology.



- Here is the routing table for this ‘mess’. This is the table for R1.

```
# netstat -r
Destination       Mask              Gateway         Flags     If
192.168.5.20  255.255.255.255   192.168.10.7    UGH       e1
192.168.1.81  255.255.255.255   192.168.10.5    UGH       e1
192.168.10.0  255.255.255.0     0.0.0.0         U         e1
192.168.18.0  255.255.254.0     0.0.0.0         U         e0
192.168.64.0  255.255.192.0     192.168.10.5    UG        e1
192.168.128.0 255.255.192.0     192.168.10.7    UG        e1
127.0.0.1       0.0.0.0           255.0.0.0       U         l0
Default         0.0.0.0           192.168.10.20   UG        e1
```

- First two entries are host routes (denoted by the H and the mask of 255.255.255.255), both these routes have a gateway on the 192.168.10.0/24 network. In effect they are routes to hosts. A U means the route is usable.
- The next two entries are typically access to a local network interface e1 and e0. The 0.0.0.0 implies a local interface. The router has legs into both networks and thus has two interfaces. This is implied by the lack of a G.
- The next two entries are connected routes which point out over gateways on the router network.

- The last route is the default route - all traffic that has not been passed on somewhere else goes here. In this example the default route points to the host 192.168.10.20.
- This machine could be an internet router or a firewall.

Route Propagation

- So far we have seen how we can manually encode routes into devices.
- This is great but does not scale in the enterprise.
- Normally in the enterprise we use routing protocols to ‘broadcast’ such route information to other nodes in the network.

Routing protocols

- There are two classes of routing protocols:
  - Intra Domain - within the confines of an organisation. Examples of such protocols are RIP and OSPF.
  - Inter Domain - routing between organisations/ networks. Examples include BGP.

The RIP protocol

- Routers and hosts talk to each other to exchange route information. The simplest protocol is RIP (routing information protocol).
- This is a distance vector routing algorithm. In this model each node in the network (capable of routing) maintains a table with the distance to each node.
- Such information is shared periodically and when there is a change to topology.

- Each router will broadcast a RIP packet every 30 seconds.
- If a router fails, nearby routers will discover this within 30 seconds. They will update their route tables accordingly.
- When they broadcast their route tables, other routers will be aware of the route changes.
- Obviously, this can take a few minutes to propagate to all the routers who need to know.
- Routes may also be set up statically by the SA and machines instructed not to listen to RIP for security of reliability reasons.

- RIP is too simple for large complex routing problems or systems requiring security.
- Other protocols such as BGP, IGRP and OSPF were developed to make the Internet work with large complex routing configurations.
- In the case of OSFP this is a Link State Routing Protocol which is much more efficient then others.
- The routing problems for Internet routers is beyond the scope of this course.

ICMP

- ICMP (Internet Control Message Protocol)
- Used for error or control (query) messages between hosts, not applications. Examples of such errors include;
  - destination unreachable
  - source quench
- ICMP is used to solve this problem. ICMP is part of the network layer and uses IP as its encapsulation PDU.

- Error Report:
  - Destination Unreachable. If a router can not route a packet to a destination it will return this ICMP error. Such messages indicate errors associated with fragmentation, unknown networks and network/ host unreachable.
  - Source Quench - flow control mechanism.
  - Time Exceeded - partly as a result of routing loops OR exceeded times on frame reassembly as a result of fragmentation.
- Query:
  - Echo Request/ Reply.
  - Timestamp Request / Reply.

Transmission Control Protocol

- Layered over the top of IP.
- Provides a reliable full duplex virtual circuit.
- Takes care of packets out of order, retransmit and variable data rates.
- Used by things like telnet, ftp and ssh where a reliable, full-duplex data stream is required.

TCP Session Establishment, Maintenance and Termination

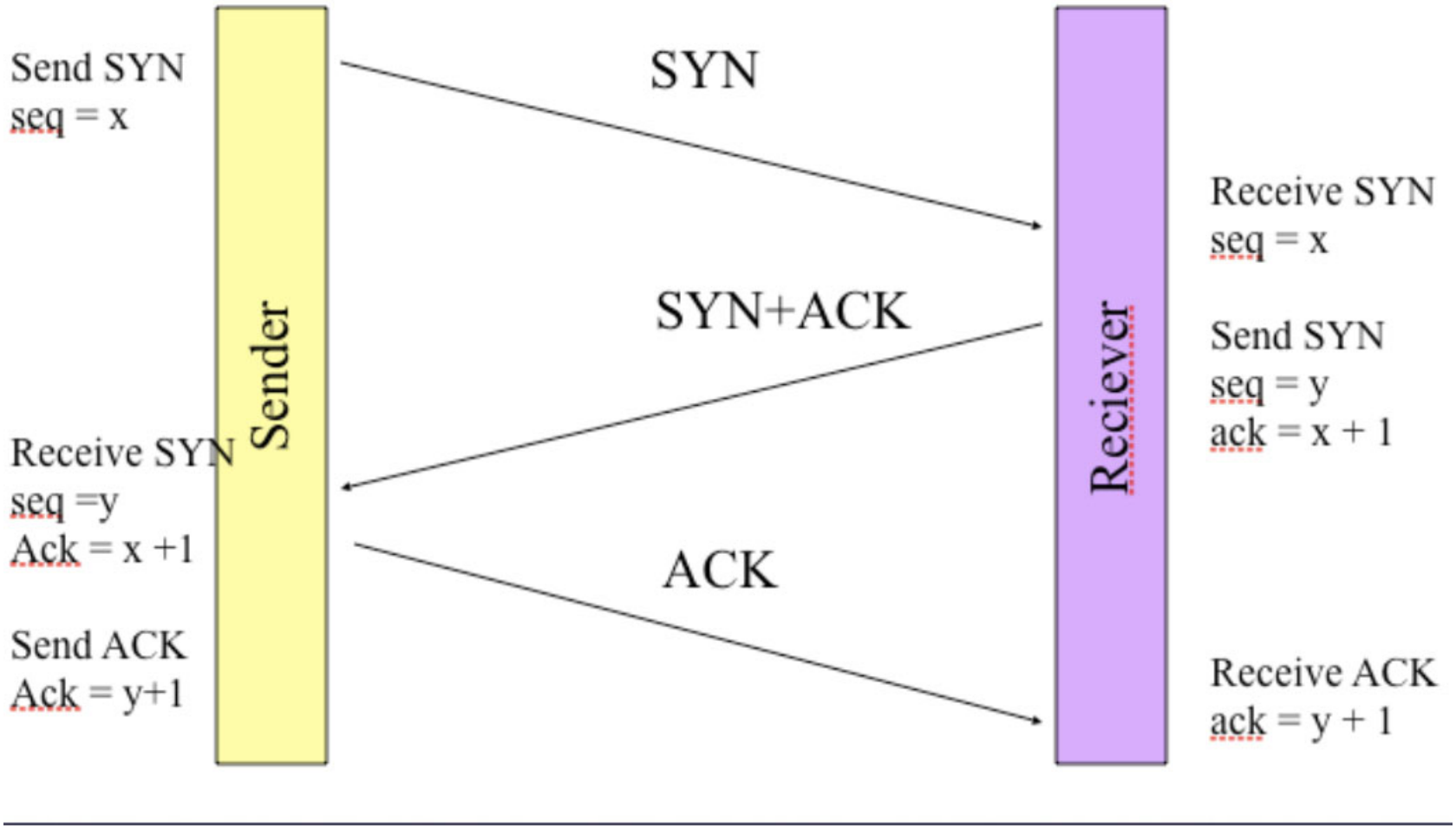
- One of the functions of the transport layer is to establish a connection oriented session with its peer.
- For data transfer to begin both sending and receiving applications inform their respective operating systems that a connection will be initiated.
- One machine will attempt to connect to the other after this is done.

- When the machine connects to its destination for the first time we call this a synchronisation handshake. The second and third handshakes negotiate the connection and confirm it from the other side.
- The last handshake is used to confirm the session which is then established.
- When the session is being negotiated a number of parameters are agreed upon one being the sliding window

Three way handshakes

- Before a connection is established, the two hosts must synchronize their initial sequence numbers.
- Packets have sequence numbers which identify them in a session. This allows us to resend them in the event they get lost.
- The sequence numbers are randomly generated by the hosts - this has the good side effect of making it slightly more difficult to hack e.g. injecting data into the session.

- The process requires both sides of the connection to send their own initial sequence number via a SYN packet. Each side must receive the other sides SYN and confirm it with a ACK (acknowledgment) packet.
- The three way handshake is important to not only negotiate the sequence number for packets but other parameters such as window size.
- We call this the SYN/ACK sequence. As I said it is used to negotiate the connection and set parameters e.g. window size and starting sequence number for


Send SYN
seq = x
SYN
Receive SYN
seq = x
Sender
SYN+ACK
Reciever
Send SYN
seq = y
ack = x + 1
Receive SYN
seq =y
Ack = x +1
ACK
Send ACK
Ack = y+1
Receive ACK
ack = y + 1

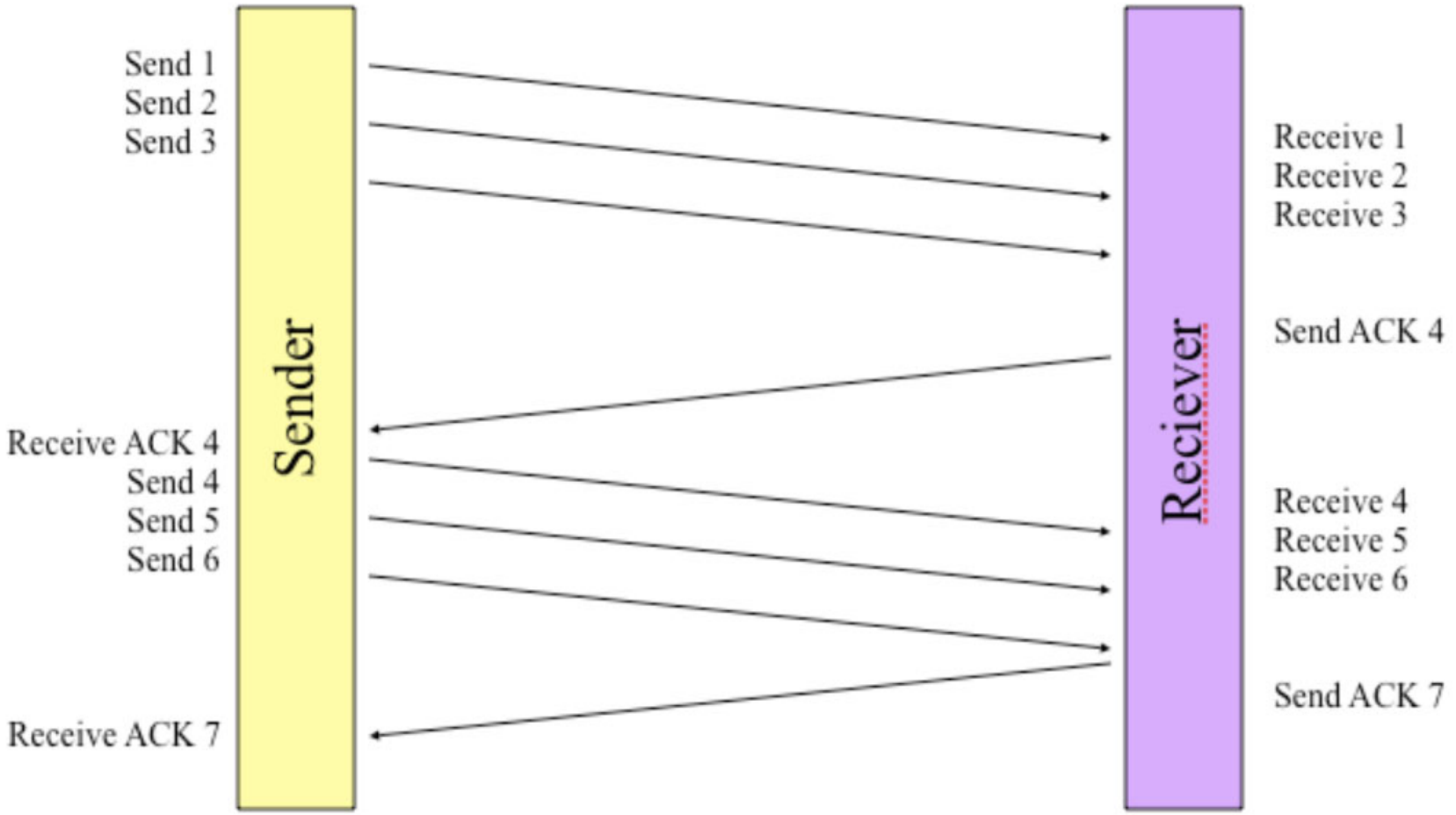
Congestion

- As data transfers between the two end points congestion can occur.
- This congestion occurs for a number of reasons e.g. throughput and network latencies.
- If too much traffic arrives quickly at the destination then there is the possibility that the buffer in which the data is stored before overflows from the internal buffer.
- TCP solves this problem by sending to the peer on the other end flags which indicate whether or not data should be sent. If TCP did not do this data would be lost hence corrupting a session.

Windowing in TCP

- TCP is a reliable protocol. It ensures packets are delivered to their destination.
- Packets must be delivered to host in the same way they are transmitted.
- If a sender has to wait for an acknowledgement on sending each packet/ segment then this would yield poor throughput.

- As a result most protocols allow more than a single packet/segment to be outstanding at a time.
- The number of data packets the sender is allowed to have outstanding without receiving an acknowledgement is known as a window.
- The window size is negotiated dynamically when the session is established.
- The window sizes can be dynamic - this means they can change during the connection for any number of reasons.



Window Size of 3

UDP

- No end-to-end connection. We typically refer to UDP as connectionless communication.
- Each packet is stand alone. In UDP there is no error checking like TCP.
- Used for situations where a virtual circuit is not required.
- Much less overhead than TCP. As a consequence we typically find it being used in applications such as
  - NFS (Network File System)
  - RIP (Route Information Protocol)
  - TFTP (Trivial File Transfer Protocol)
  - SNMP (Simple Network Management Protocol)

Services

- A service typically implemented by a protocol is usually offered on a port.
- Both UDP and TCP introduce into the addressing model the notion of ports.
- UDP ports are different to TCP ports.
- You should note that there are 65536 different ports on modern hosts.

- In order to use ports less than 1024 you need to be the *root* user. These ports are considered privileged.
- Remember we can use *lsof* to find out what ports processes have open.
- Ports are allocated to services through organisations such as ICANN. Standard port allocations are typically articulated in the */etc/services* file.
- Such services are called Well Known Services (WKS).

- Below is a sample from /etc/services.

```
ftp             21/udp                  # File Transfer [Control]
ftp             21/tcp                  # File Transfer [Control]
ssh                     22/udp
        # SSH Remote Login Protocol
ssh                     22/tcp          # SSH Remote Login Protocol
telnet          23/udp                  # Telnet
telnet          23/tcp                  # Telnet
```