Review Question

How long (in minutes) will it take to download a 3GB file over a 100Mb/s Fibre connection?

3GB * 1000(MB) = 3000MB
100Mb/s / 8 = 12.5MB/s
3000MB / 12.5MB/s = 240 seconds
240s / 60s = 4 minutes

Murdoch UNIVERSITY

Murdoch
UNIVERSITY

The OSI Application and Transport Layers

ICT169

Foundations of Data Communications

Admin – A Quick Update

- Most students should now be enrolled in a lab
  - Contact me if you're still unable to enrol
- All labs are now in 245.3.064 (SC3.064)
- Tutor emails will be distributed in the labs this week
- Student card access to the computer labs should be coming in the next two weeks

	Mon	Tue	Wed	Thu	Fri
8:00 AM					
8:30 AM	ICT169 Lecture	ICT169 Lab		ICT169 Lab	ICT169 Lab
9:00 AM	ECL4	SC3.064		SC3.064	SC3.064
9:30 AM			ICT169 Lab		
10:00 AM			SC3.064		
10:30 AM	ICT169 Lab	ICT169 Lab		ICT169 Lab	ICT169 Lab
11:00 AM	SC3.064	SC3.064		SC3.064	SC3.064
11:30 AM			ICT169 Lab		
12:00 PM			SC3.064		
12:30 PM		ICT169 Lab			ICT169 Lab
1:00 PM		SC3.064			SC3.064
1:30 PM				ICT169 Lab	
2:00 PM				SC3.064	
2:30 PM					
3:00 PM					
3:30 PM		ICT169 Lab		ICT169 Lab	ICT169 Lab
4:00 PM		SC3.064		SC3.064	SC3.064
4:30 PM					
5:00 PM					
5:30 PM				ICT169 Lab	
6:00 PM				SC3.064	
6:30 PM					
7:00 PM					
7:30 PM					

Murdoch UNIVERSITY

Participation Quizzes – A Reminder

- Participation is not assessed in the lab sessions, but through weekly LMS-based quizzes
- Around 3 long answer questions per quiz
- Not marked on correctness, but rather whether you have made a clear attempt to answer the question
  - Answers **must be your own work – no copying from Internet sources or other students will be permitted**
- While content will be from lectures and labs, some additional reading may be required
- Quizzes will be due the at end of the following week; Quiz 1 due Sunday (12 August)

Murdoch UNIVERSITY

Last Week

- Broad overview of data communications
- Introduced important data communications concepts:
  - The OSI and TCP/IP networking models
  - Circuit and Packet switching
  - Network convergence
  - Different types of networks and network traffic
- Network speeds and data storage
- Each week, we'll move further down the OSI model

Murdoch UNIVERSITY

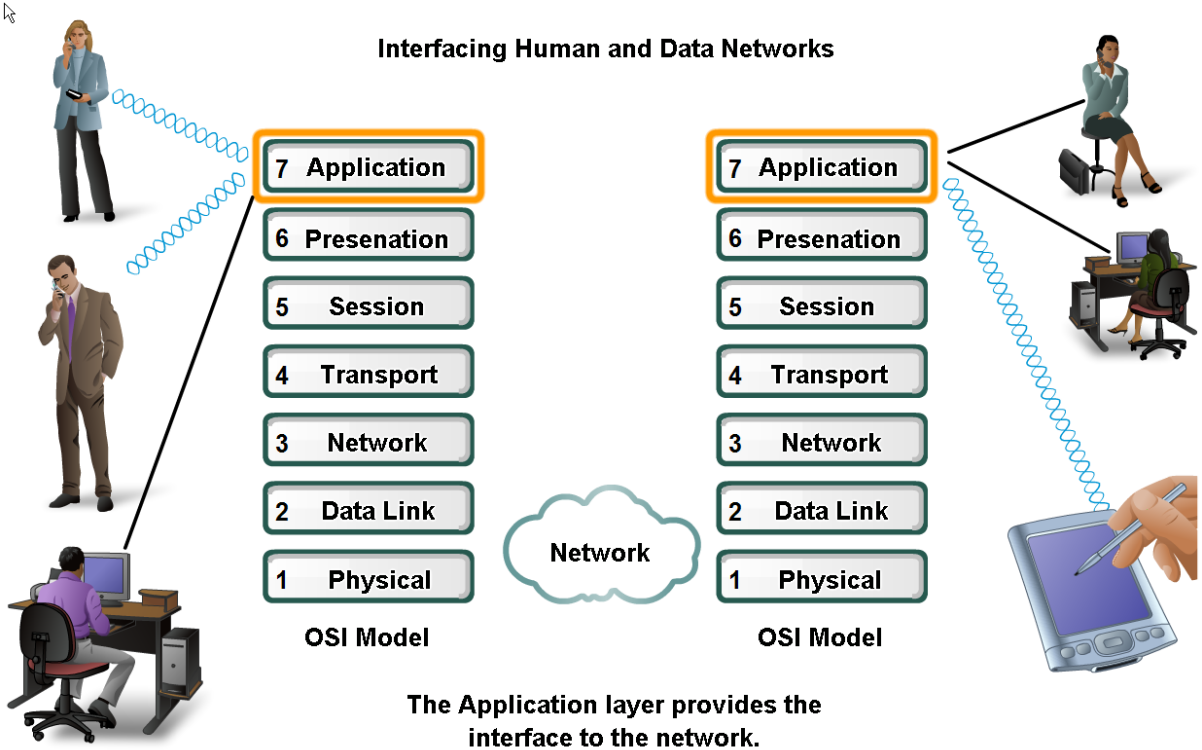
Lecture Overview

- More on the Application, Presentation, and Session layers of the OSI model
- Introduce widely used applications and services:
  - Domain Name System (DNS)
  - Hypertext Transfer Protocol (HTTP)
  - File Transfer Protocol (FTP)
  - Dynamic Host Configuration Protocol (DHCP)
- The Transport Layer and associated protocols (TCP and UDP)

Murdoch UNIVERSITY

Applications

- Applications provide the means to generate and receive data that is to be transported over the network
- Applications are the software tools end-users see interact with; the lower layers should be transparent to most users



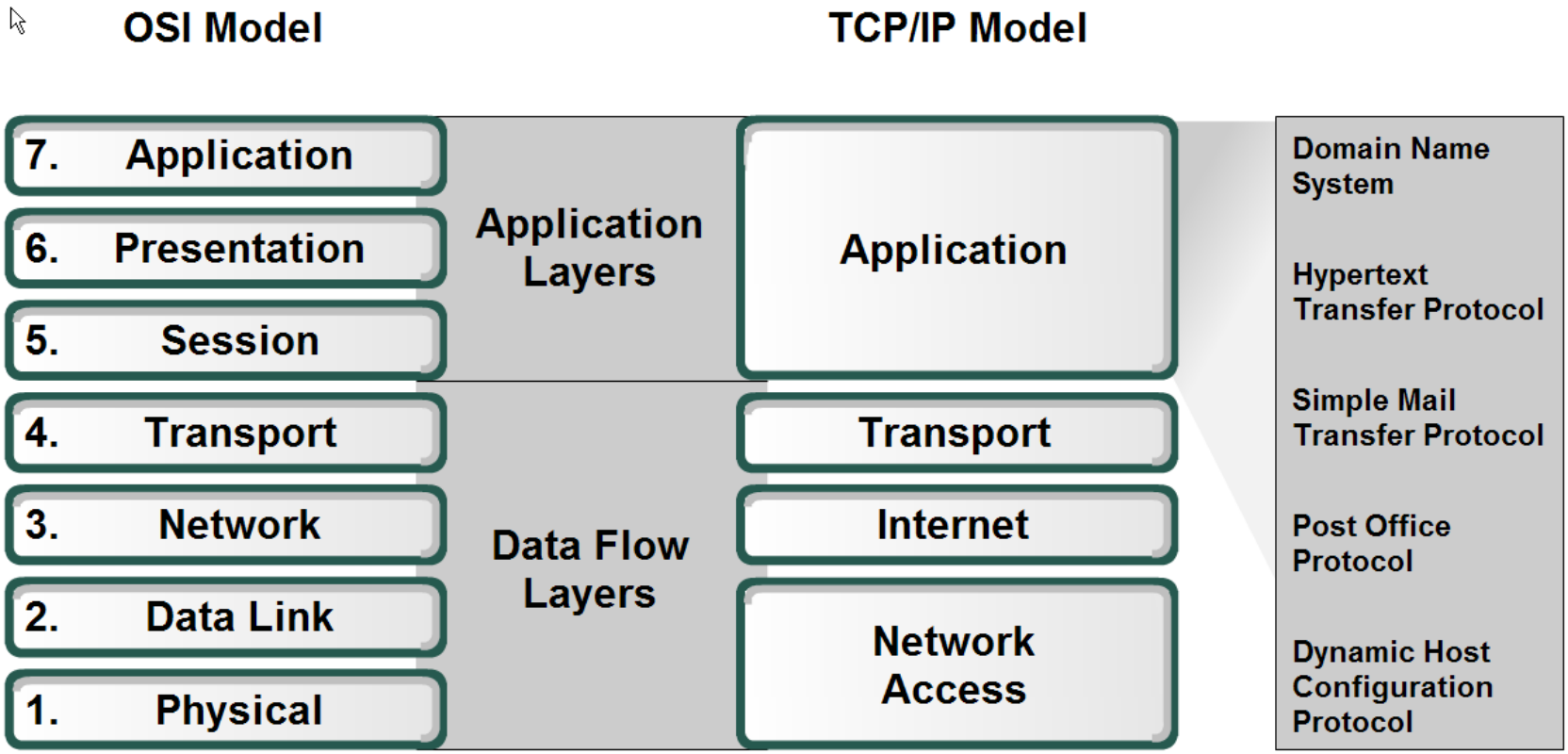
Murdoch UNIVERSITY

Presentation and Session Layers

- Presentation Layer
  - Coding and conversion of application layer data to ensure that data from source can be interpreted by destination
  - Think file formats: MPEG, JPEG, PNG, PDF, AVI, DOC, MP4, DOCX
- Session Layer
  - The session layer handles the exchange of information to initiate, restart, keep alive and terminate **conversations**
  - Examples: VPN protocols (PPTP, L2TP)

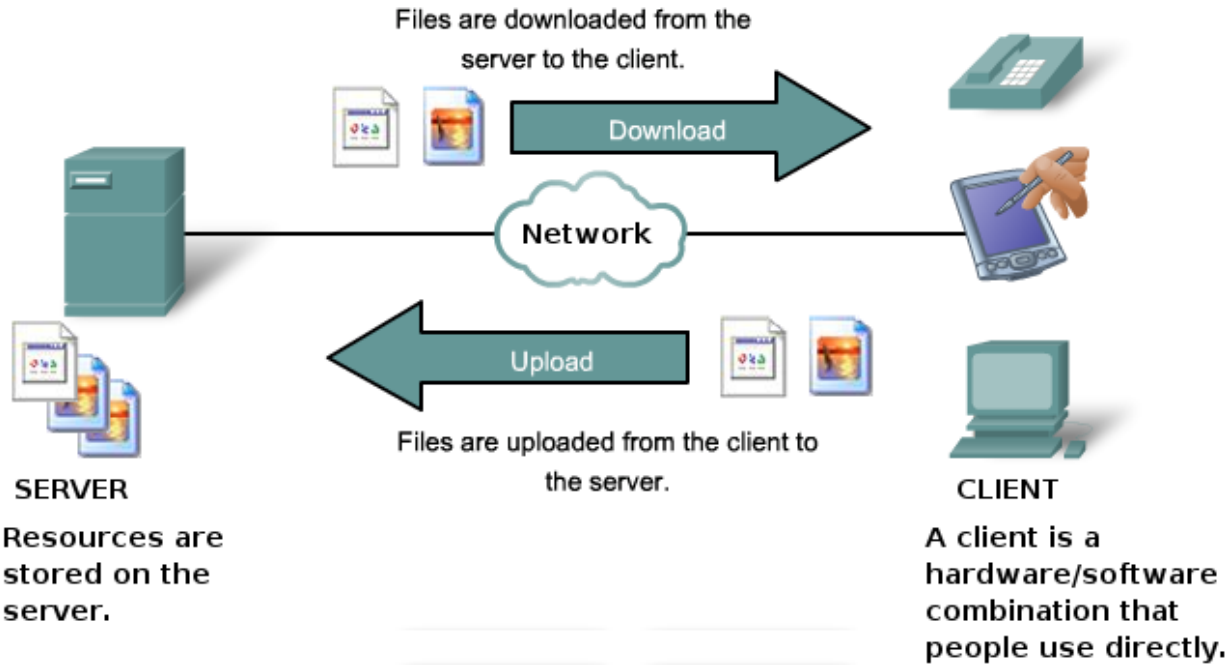
Applications in the Real World

- Most applications provide all application, presentation and session layer functions (similar to TCP/IP model)



Application Models – Client / Server

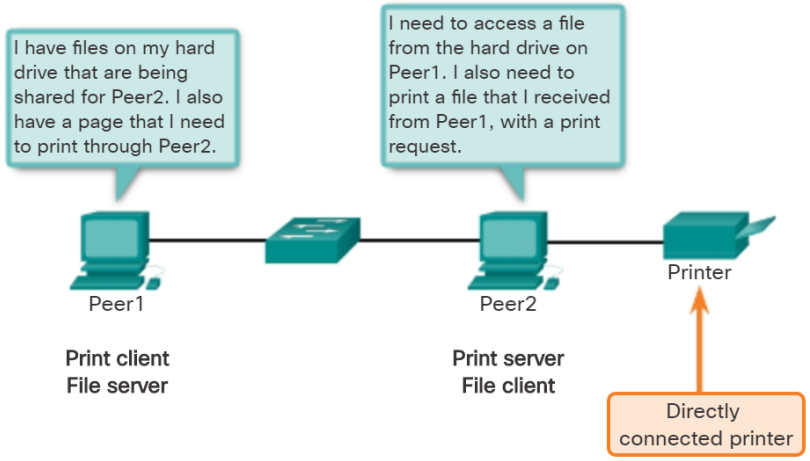
- Traditional (and still most common) communications model used in modern data networks
- End-user devices (eg. PCs, tablets, phones) generally assume the role of the client
- Dedicated servers provide content and services



Murdoch UNIVERSITY

Application Models – Peer-to-Peer (P2P)

- Comparatively new model for applications
- Devices act as both client and server.
- Examples: BitTorrent (filesharing), Skype (voice and video communication)
- P2P services still rely on centralized services (eg. BitTorrent trackers)
- Data is transmitted directly between machines; no server(s) needed.



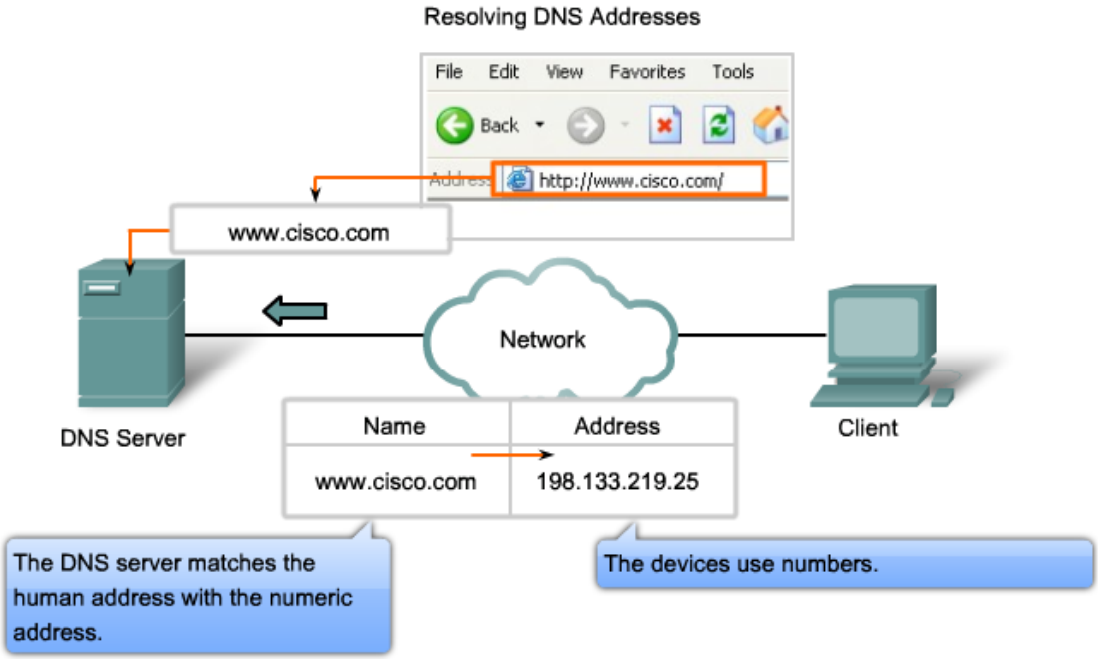
Murdoch UNIVERSITY

Applications

Murdoch UNIVERSITY

Domain Name System (DNS)

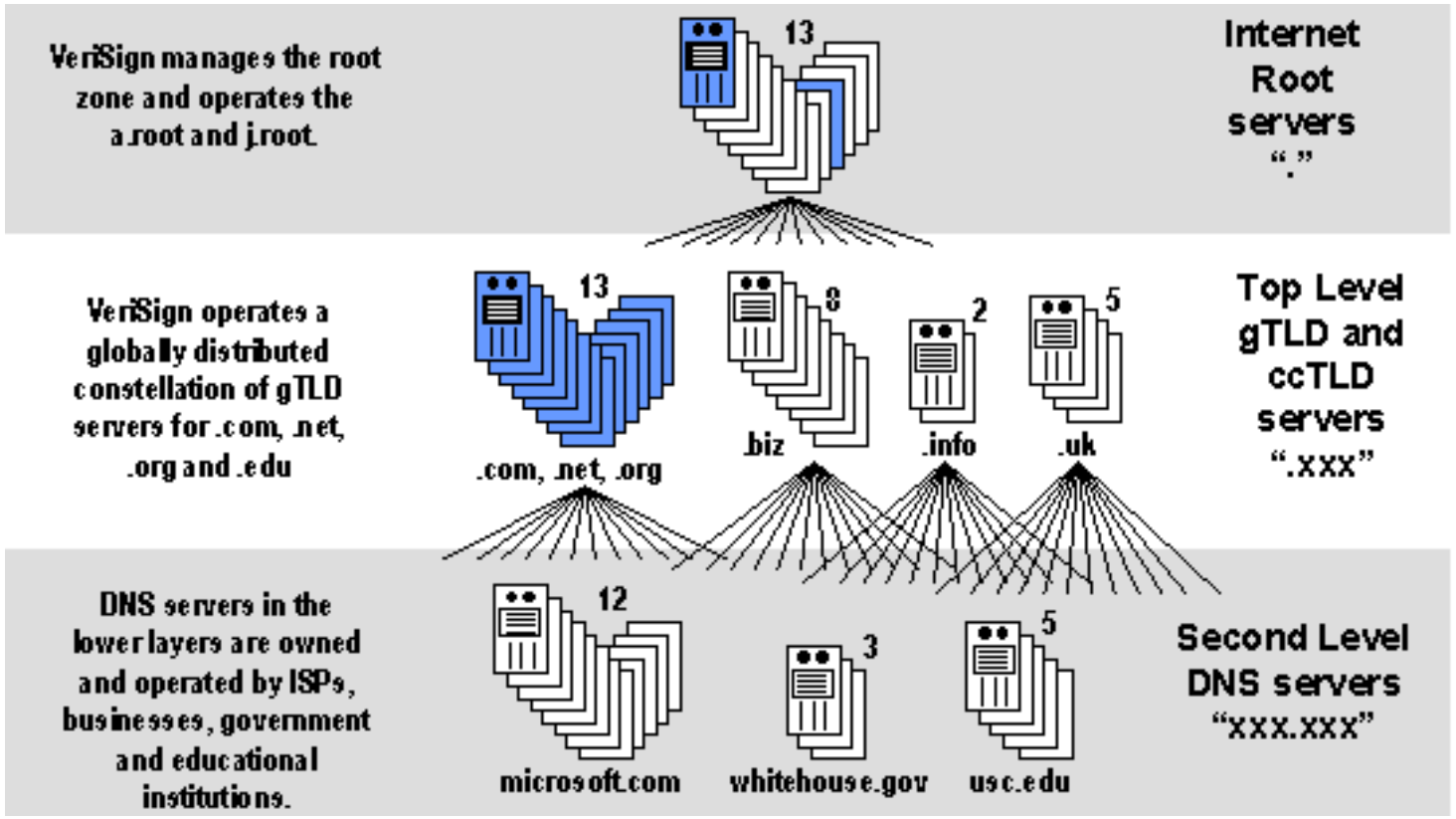
- Translates between human readable domain names (eg. “cisco.com”) and the associated network IP address (such as 198.133.219.25)
- DNS is a hierarchical system
- Uses UDP (or TCP) port 53



Murdoch UNIVERSITY

DNS Hierarchy

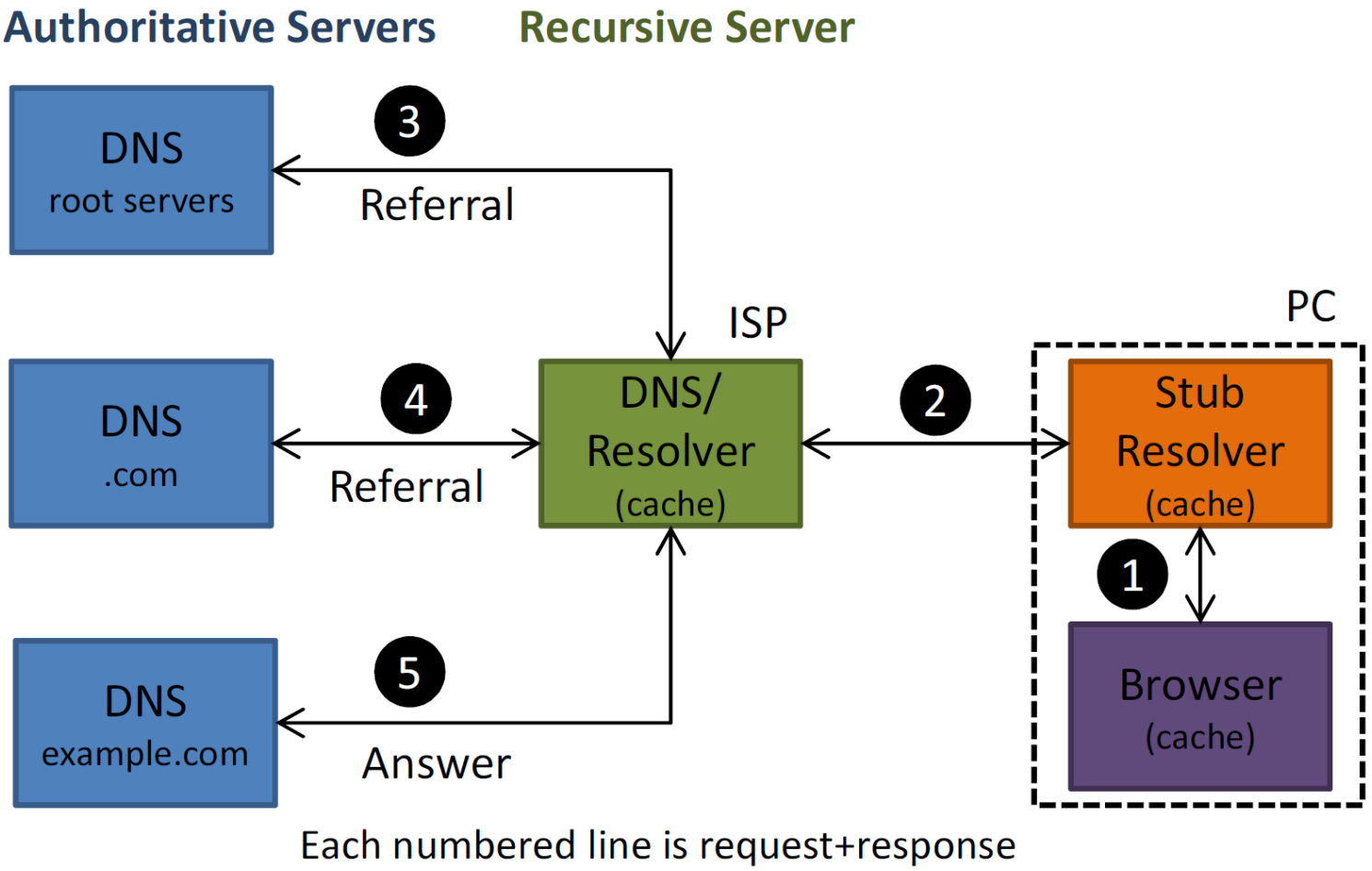
- Two different types of top level domain (TLD)
  - gTLD = general TLD, for example .com
  - ccTLD = country code TLD, for example .au



Source: http://archive.icann.org/en/tlds/org/applications/uia/C17_10.html

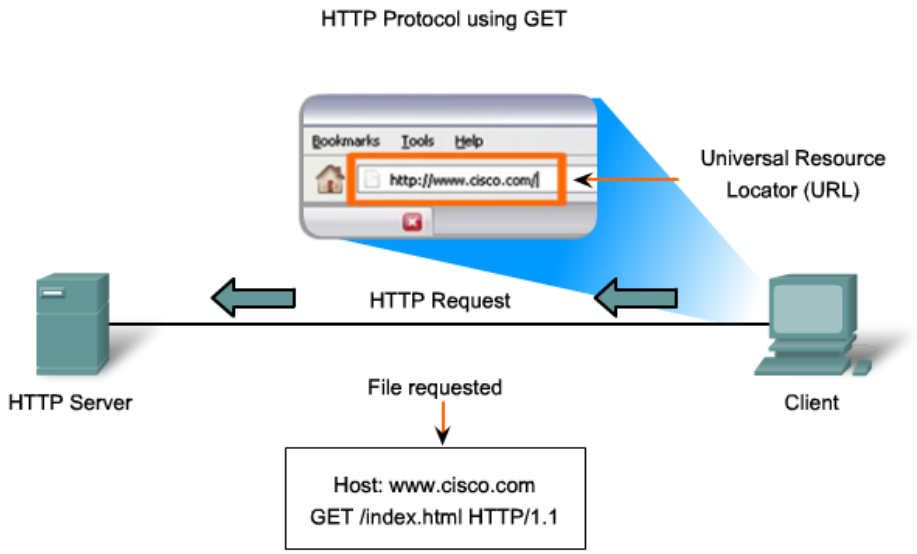
DNS Operation

- DNS lookup for **www.example.com**



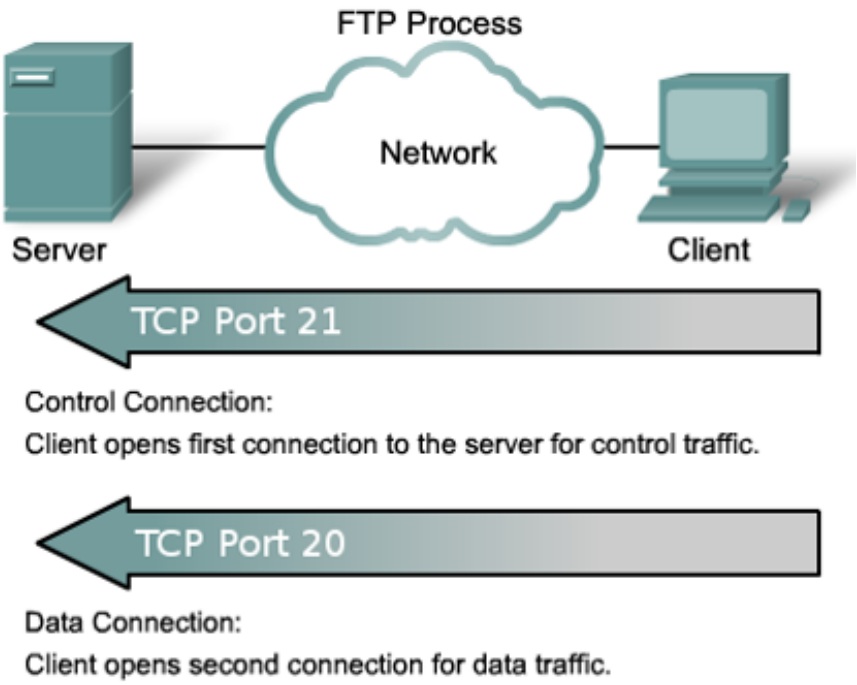
Hypertext Transfer Protocol (HTTP)

- Used to retrieve content from web servers
- HTTP GET is generated when URL is entered into the web browser
- Web server should respond with the requested page
- HTTP POST and PUT messages are used to send data to a web server (such as forms)
- Uses TCP port 80



File Transfer Protocol (FTP)

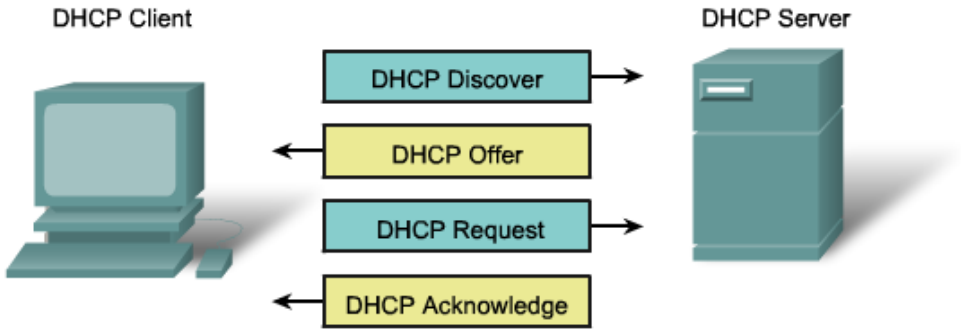
- Traditionally used for downloading and uploading large files over the Internet
- Designed for transferring data rather than displaying content
- Uses two ports:
  - TCP port 20 for the transfer of data
  - TCP port 21 for command and control



Murdoch UNIVERSITY

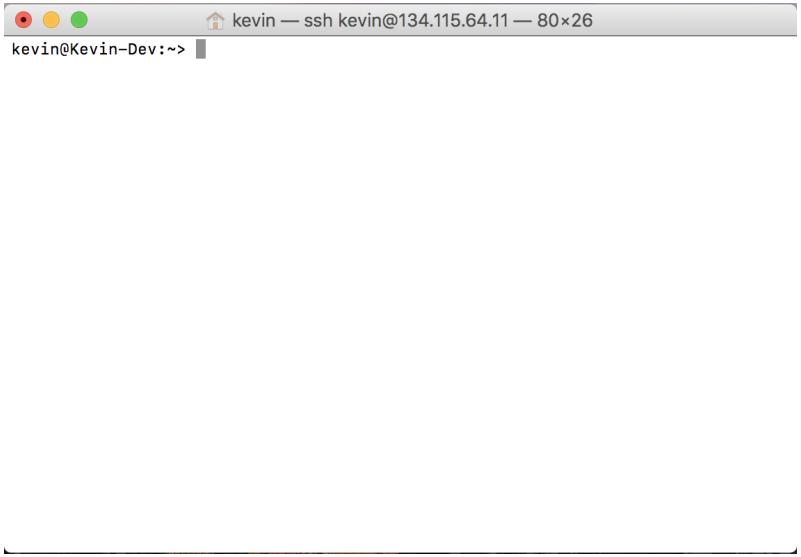
Dynamic Host Configuration Protocol (DHCP)

- Used to obtain an IP address 'automatically' from the network
- Four step process to obtain an IP address
- Most home routers have DHCP active by default
- Uses two ports:
  - UDP port 67 for requests sent by the client.
  - UDP port 68 for responses from the server.



Murdoch UNIVERSITY

Telnet

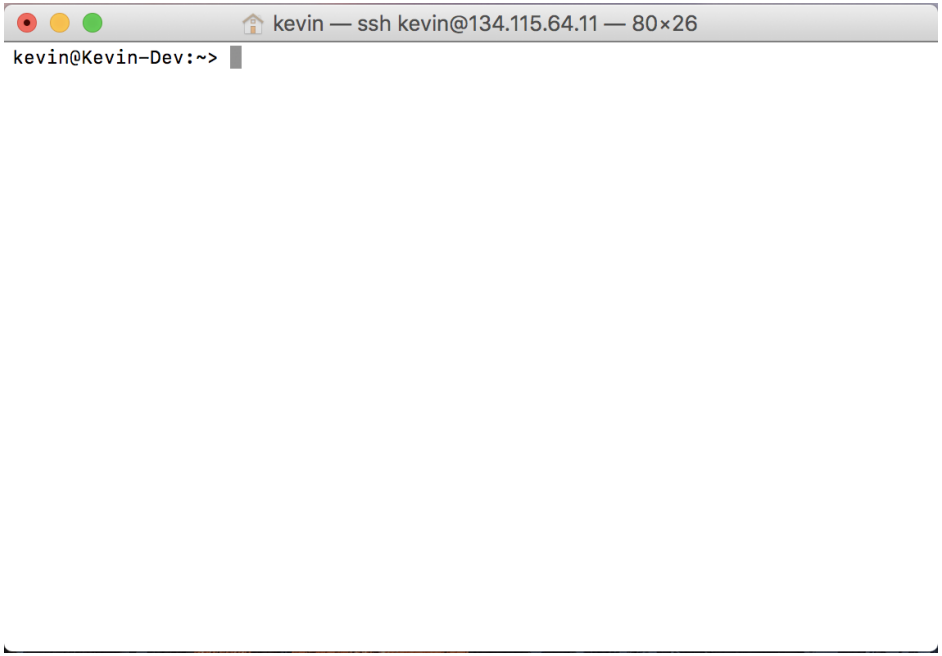


- Used to provide an interactive command line
- Typically used for remote management of network devices and servers
- Telnet is an old protocol (developed in 1969) and transmits in cleartext
  - Not secure and disabled in many operating systems
  - Superseded by SSH
- Uses TCP port 23

Murdoch UNIVERSITY

Secure Shell (SSH)

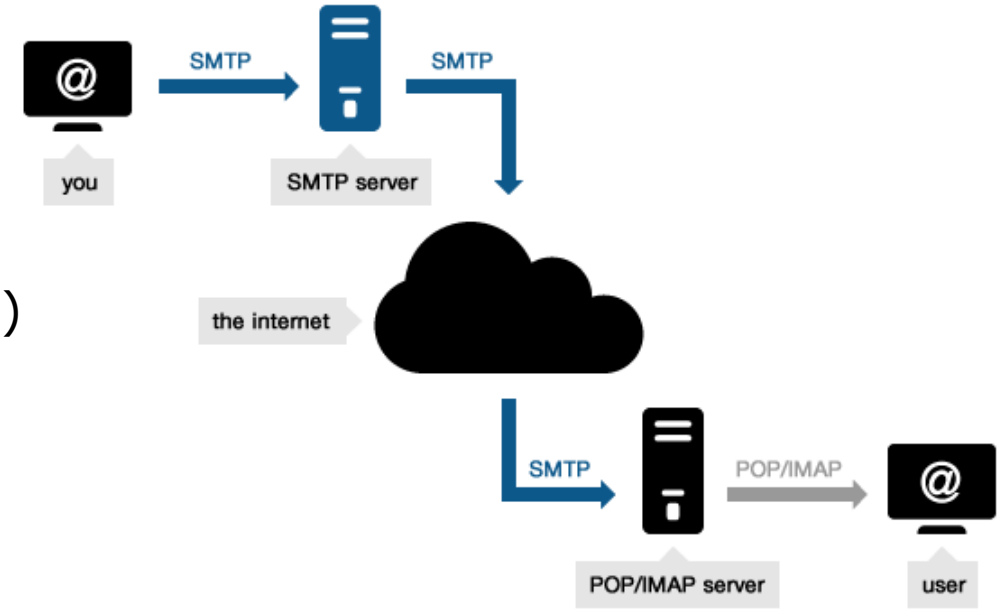
- Also used to provide an interactive command line for remote management
- Encrypts all session data to prevent third parties from reading any intercepted data
- Uses TCP port 22



Murdoch UNIVERSITY

Simple Mail Transfer Protocol (SMTP)

- Used for sending and receiving email between mail servers
- Also used for sending email from a client to a mail server
- Uses TCP port 25 (can also use TCP ports 465 and 587)
- For receiving mail, use:
  - Post Office Protocol (POP) on TCP port 110
  - Internet Message Access Protocol (IMAP) on TCP port 143



Source: www.serversmtp.com

Murdoch UNIVERSITY

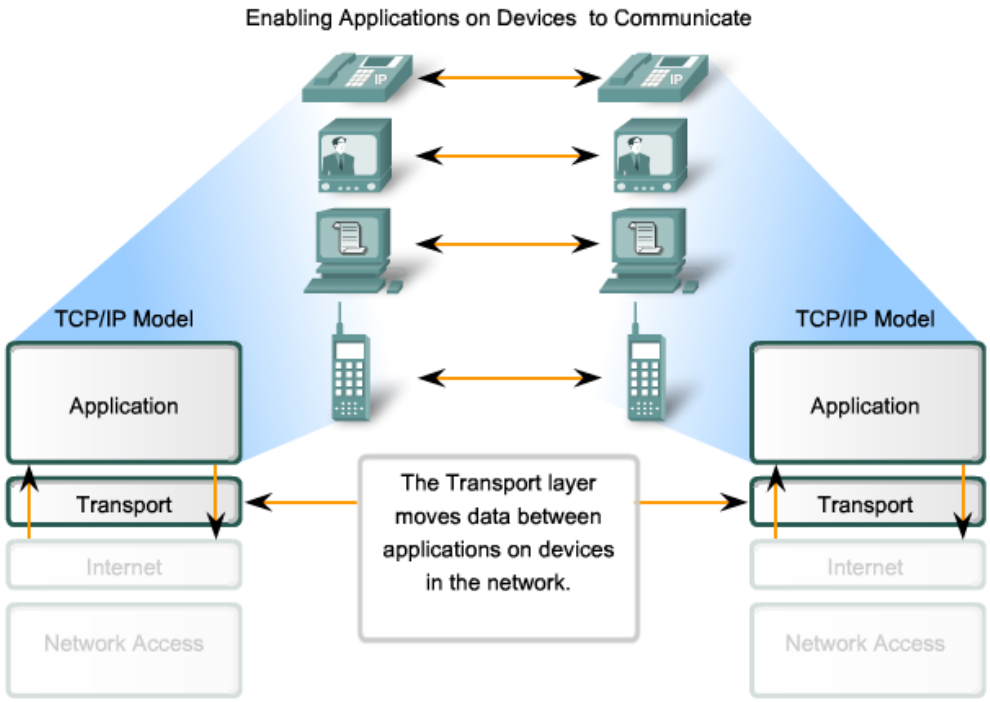
Break

When we return: The Transport Layer

Murdoch UNIVERSITY

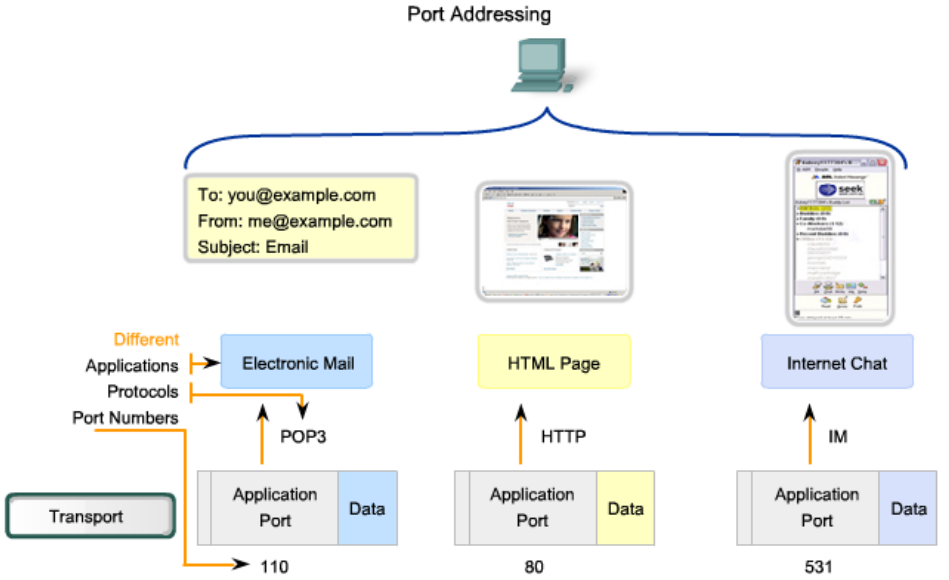
The Transport Layer

- Segments application data into transportable chunks for transmission
- Uses **ports** to track individual conversations and identify applications
- Can reassemble segments and provide reliability
- Different transport layer protocols are used to cater to differing requirements of applications
- Often referred to as an end-to-end concept



Ports – Addressing at the Transport Layer

- Don’t exist physically; ports are a logical concept used by operating systems for identification of different applications
- Ports are identical, but some are ‘recognised’ for use by specific applications (eg. TCP port 80 is recognised as HTTP)
- Some applications can have multiple port numbers (eg. HTTP can also used port 8080)



Data for different applications is directed to the correct application because each application has a unique port number.

Murdoch UNIVERSITY

Ports (cont.)

- Port number is a 16-bit integer value (0—65535)
- Three classes of ports:
  - **Well known ports (0–1023)** for common services and applications.

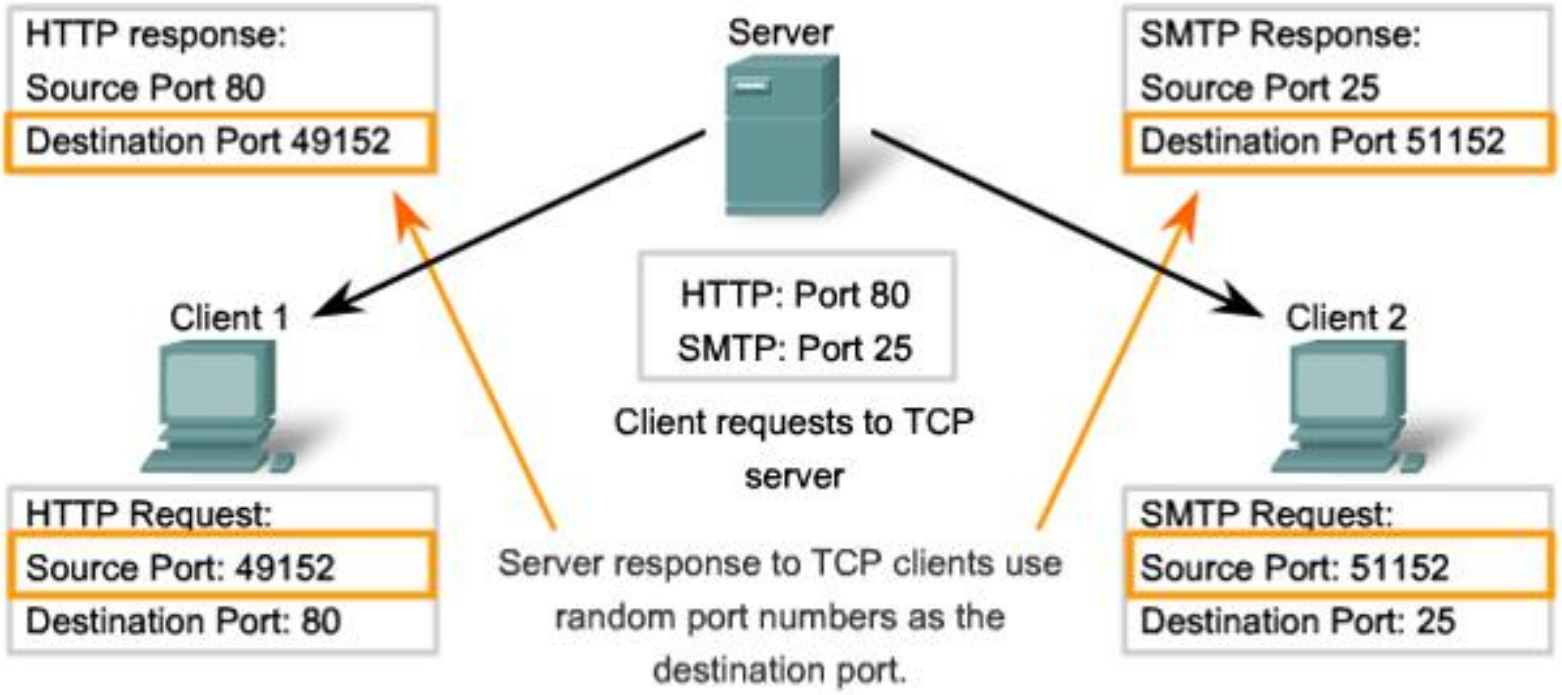
    Examples: HTTP → Port 80, SMTP → Port 25
  - **Registered ports (1024—49151)** for less commonly used services or applications

    Examples: OpenVPN → Port 1194, SIP → Port 5060
  - **Dynamic / Private ports (49152—65535)** for client-initiated sessions
- Full list available: http://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers

Murdoch UNIVERSITY

Ports Use Example

- Clients use private ports to initiate sessions
- The source port does not need to match the destination port

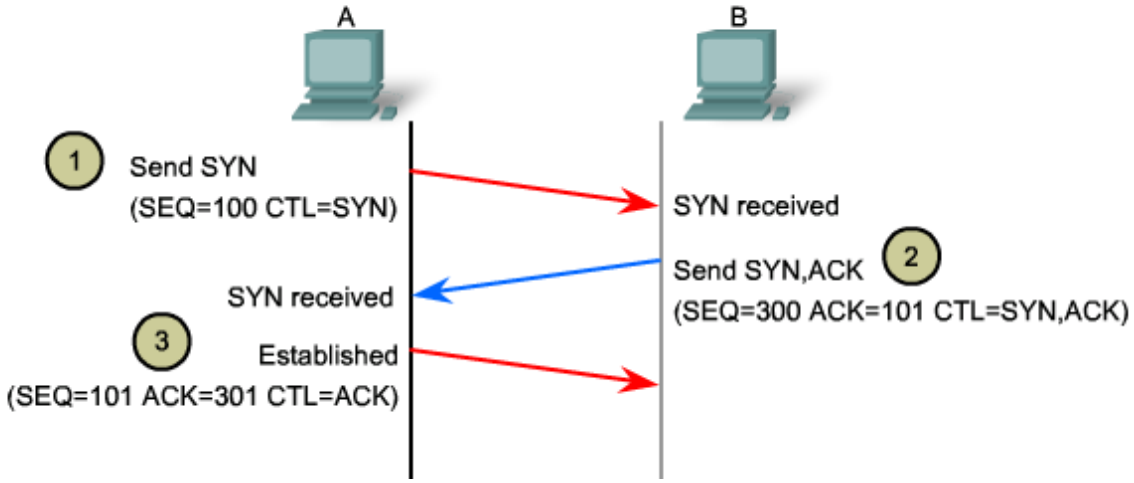


Transport Layer Protocols

- Two most common transport layer protocols: Transmission Control Protocol (TCP) and User Datagram Protocol (UDP)
- TCP is used when the delivery of data must be **reliable**
  - File downloads
  - Loading webpages
- UDP is used when the delivery of data must be **timely**
  - Voice and video communications
  - Online games
- Other protocols exist but are not widely used
  - Stream Control Transmission Protocol (SCTP)
  - Datagram Congestion Control Protocol (DCCP)

Transmission Control Protocol (TCP)

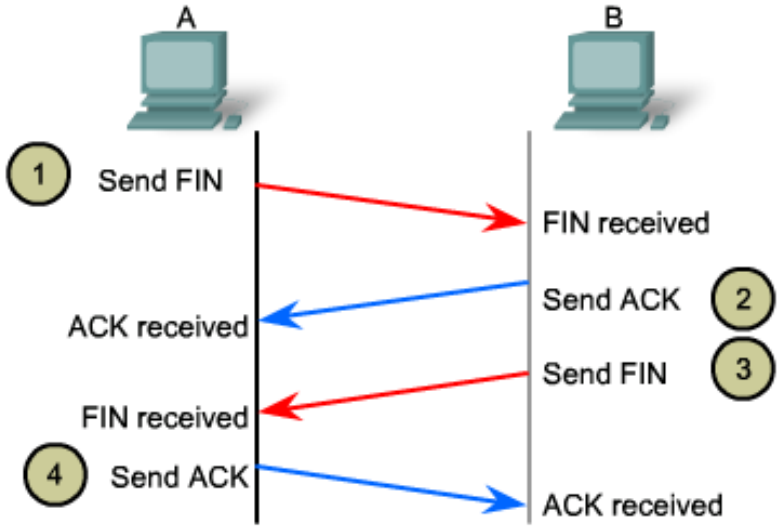
- TCP is a **connection-oriented protocol**, meaning that communications between two devices must be explicitly initiated and terminated
- Not all transport layer protocols are connection-oriented
- Connection setup consists of three steps: SYN, SYN-ACK, and ACK
- Process known as **three-way handshake**



Murdoch UNIVERSITY

Transmission Control Protocol (cont.)

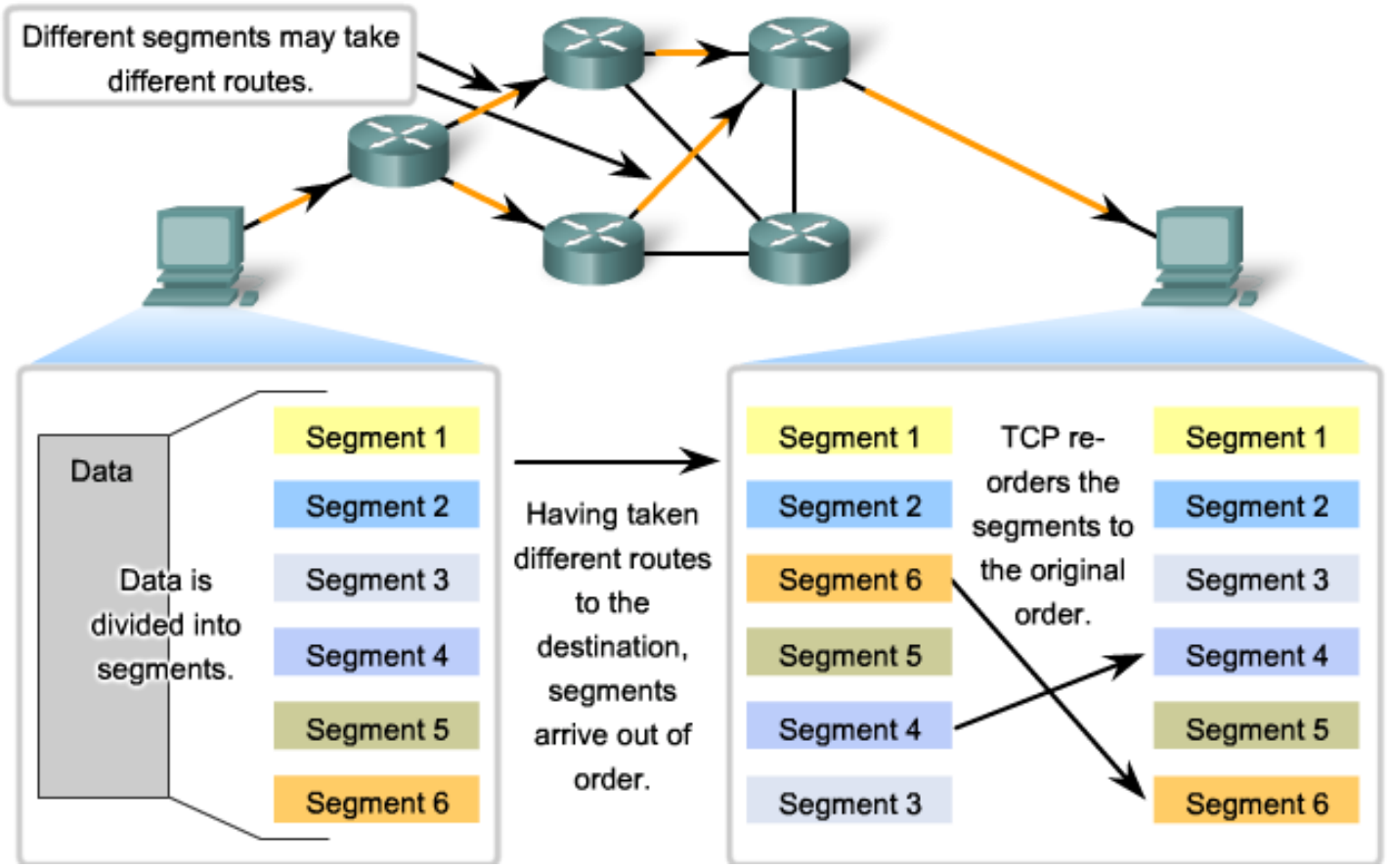
- When a conversation is complete, the connection is terminated using a four step process
- Consists of FIN, ACK, FIN, and ACK
- Unlike in the three-way handshake, the FIN and ACK sent by host B are actually two separate steps



Murdoch UNIVERSITY

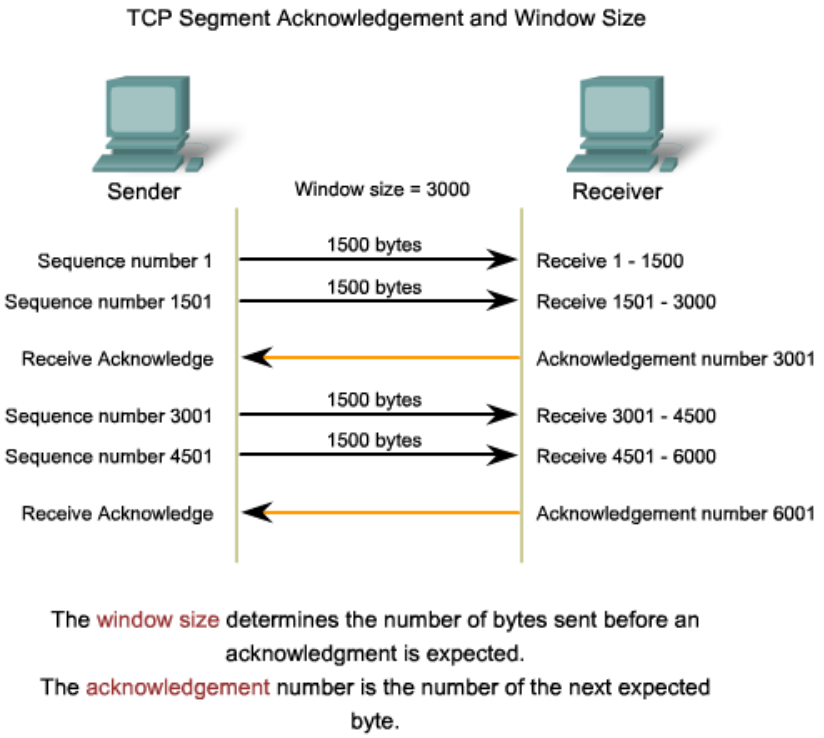
TCP In-Order Delivery

- TCP provides in-order delivery of segments to application by using **sequence numbers**



TCP Reliable Transport

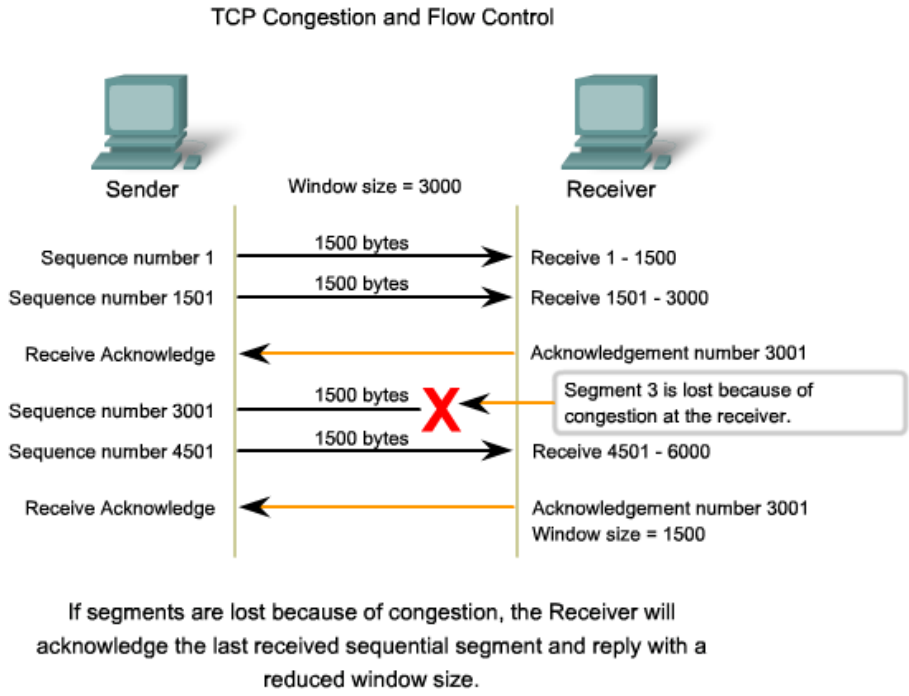
- TCP **sequence numbers** are used in conjunction with **acknowledgements** (ACKs) to provide reliability
- All data transmitted using TCP must be acknowledged
- An ACK is cumulative, meaning that it also acknowledges all preceding segments
- Receivers always acknowledge **next expected byte**



Murdoch UNIVERSITY

TCP Reliable Transport (cont.)

- When segments are not ACKed, they must be retransmitted by the sender
- Segments can be lost due to network congestion or interruptions to the medium



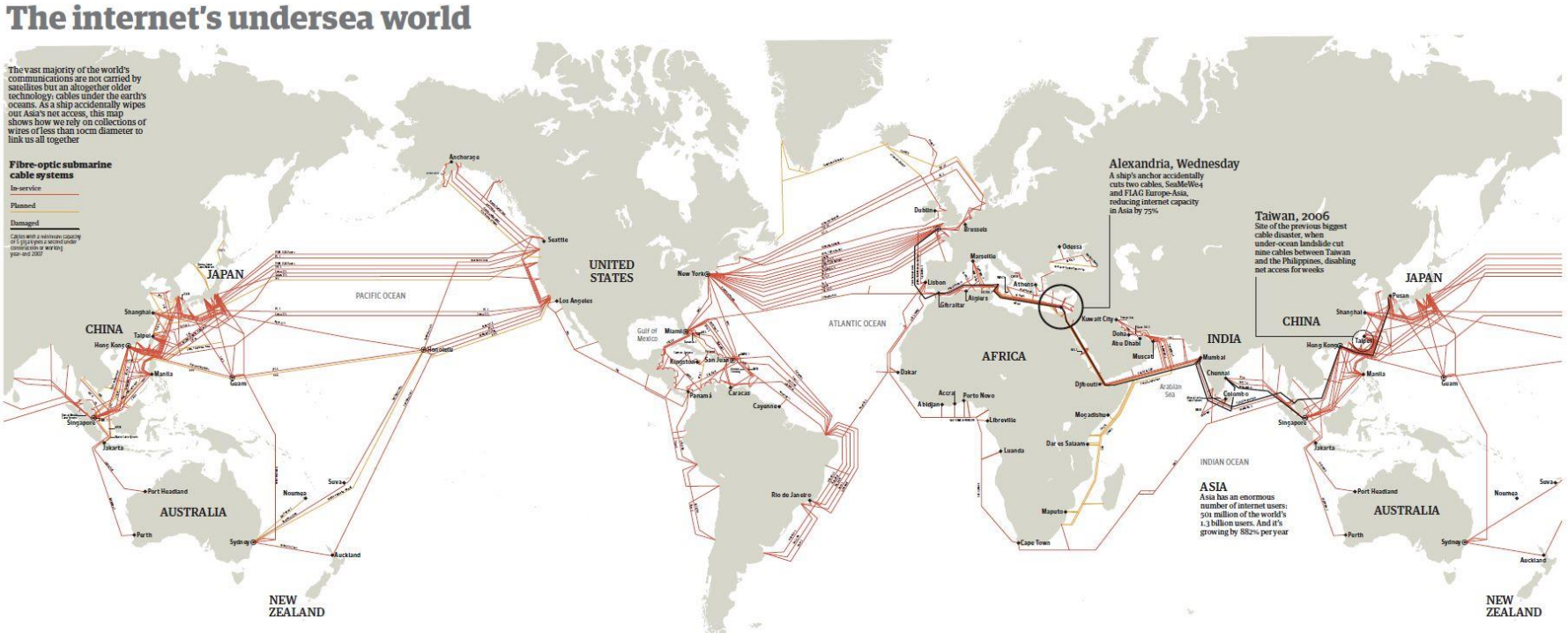
Murdoch UNIVERSITY

TCP Congestion Control

- TCP uses congestion control to manage the rate of transmission
- TCP congestion window specifies number of unacknowledged segments that can be in-flight from sender to receiver
- Why use congestion control?
  - What if TCP senders could only send one packet at a time without ack? If round trip delay between sender and receiver was 200ms, TCP could send 5 packets per second. Very slow.
  - What if TCP senders could send as fast as LAN connection permits (eg. 1000Mb/s)? Gateway to Internet is **bottleneck** and unable to handle load. **Congestion** (packets may be dropped)

TCP Congestion Control (cont.)

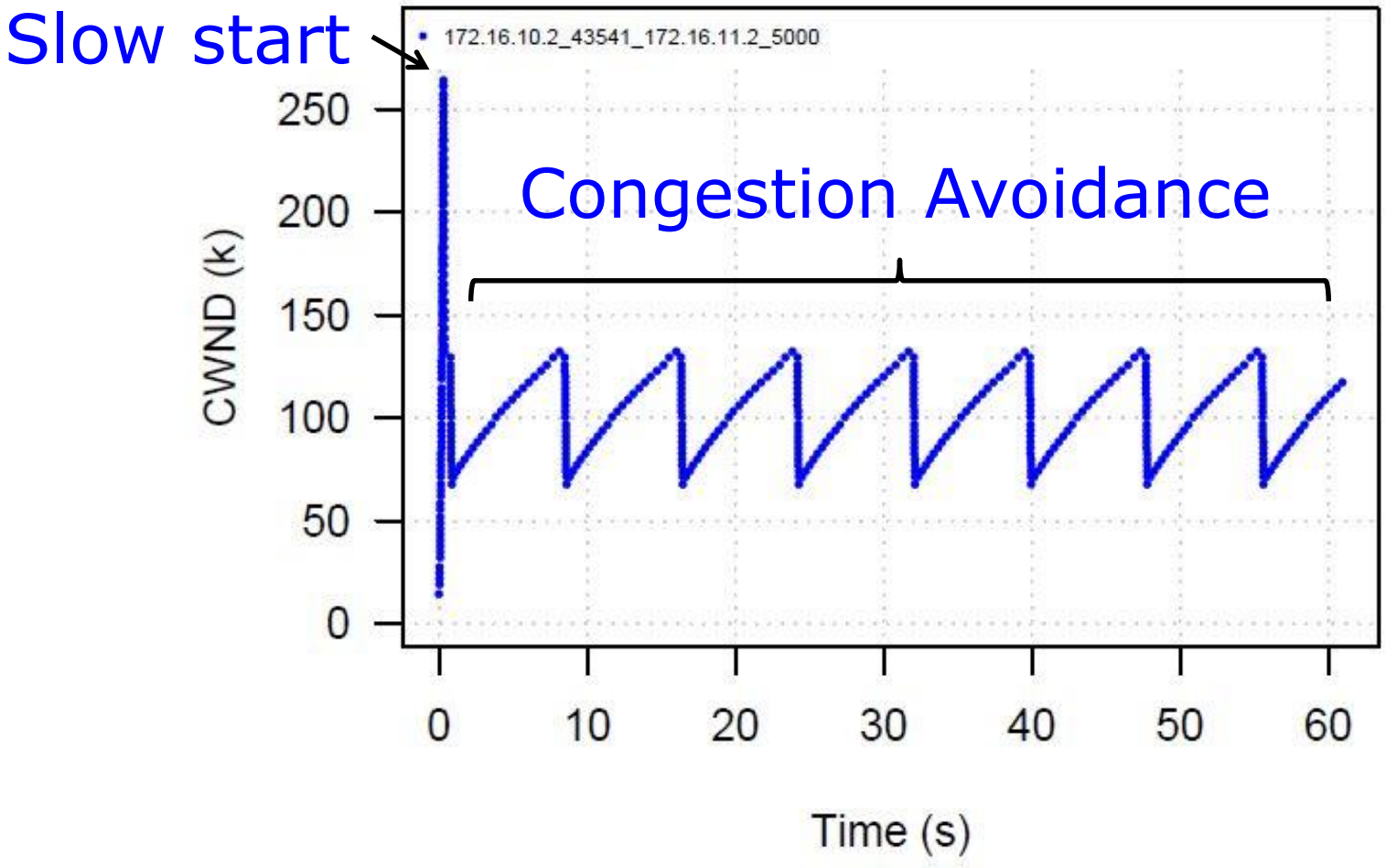
- Consider that we share resources with many other users
- Fibre links between continents carry traffic from millions of concurrent TCP connections
- How does a TCP sender finds perfect rate – Fair share of 100% utilised bottleneck link speed?



TCP Congestion Control Algorithm – NewReno

- **Slow Start** (not that slow)
  - Start with initial congestion window of 2 (or 10) segments
  - Sender increases congestion window by 1 segment for every packet acknowledged by receiver
  - Quickly increase throughput up to max possible fair share
  - When sender detects packet loss it halves window and goes into
- **Congestion Avoidance**
  - Without loss increase window by 1 segment each round trip time
  - When sender transmits too fast and congests low speed link, router connected to low speed link will drop packets
  - When sender detects packet loss it halves window
  - Quickly shrinking window will quickly reduce throughput of connection and congestion on the bottleneck

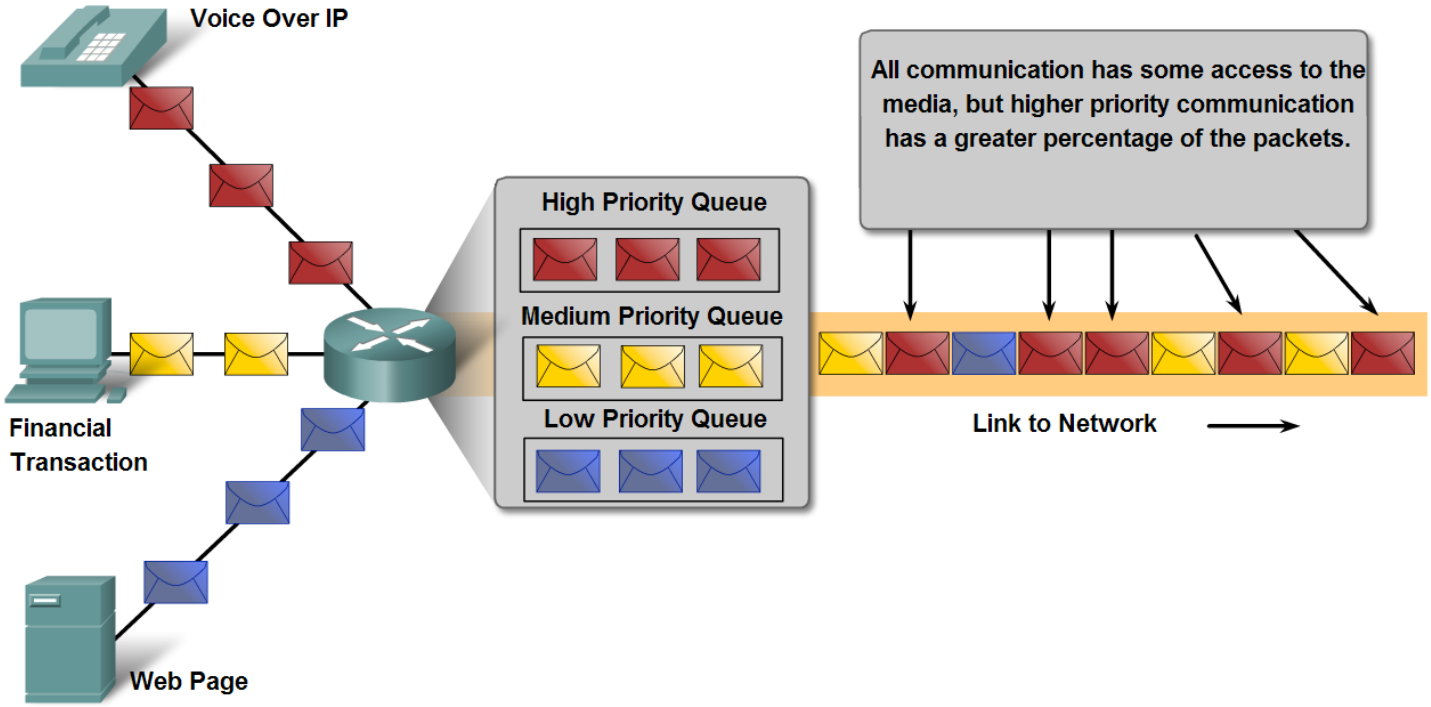
TCP Congestion Control in Action



Murdoch UNIVERSITY

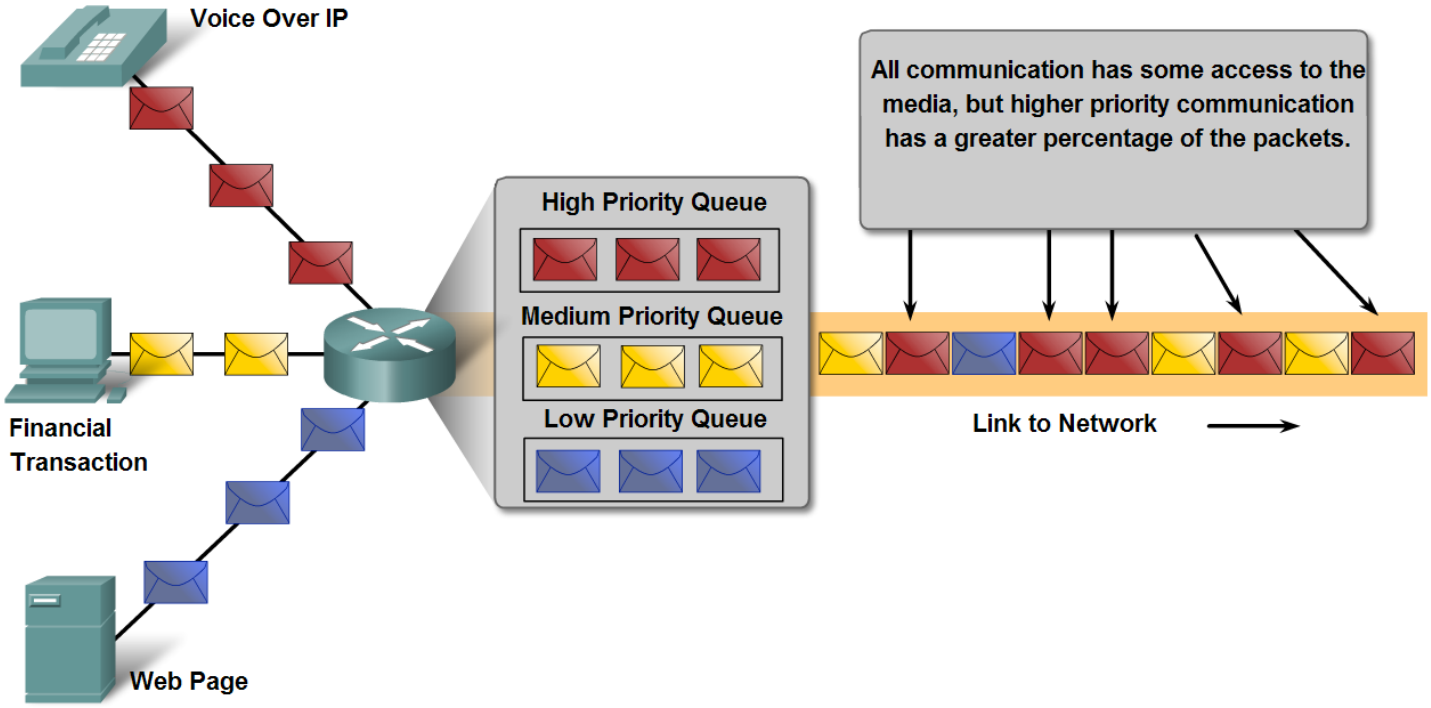
Buffering on the Network

- Congestion occurs when number of packets arriving at router is higher than number of packets that can be send on next link
- Router buffers outgoing packets, but when buffer is filled the router must drop packets



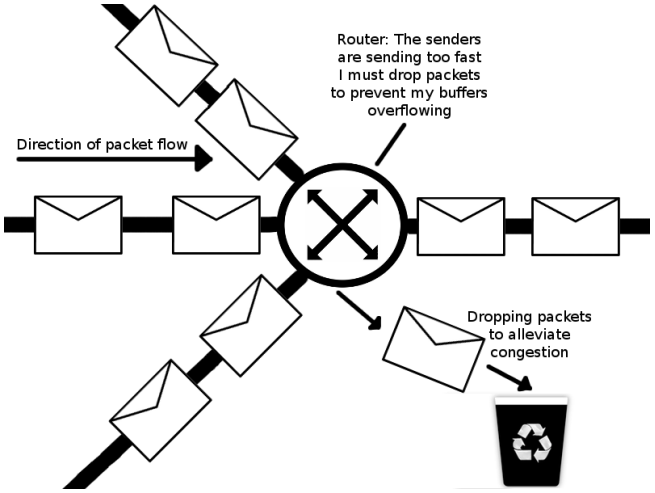
Buffering on the Network (cont.)

- Router buffers are usually quite small, so let's increase them! Problem solved!
- This is what people actually used to think for years
- Leads to problem called **Bufferbloat**



Bufferbloat

- If we increase buffer sizes, network latency will also be increased
  - NewReno will always fill buffers to capacity
  - Larger buffers take longer to clear
- Applications that require reliability might also need low latency (eg. stock trading)
- Could also interfere with some TCP congestion control mechanisms



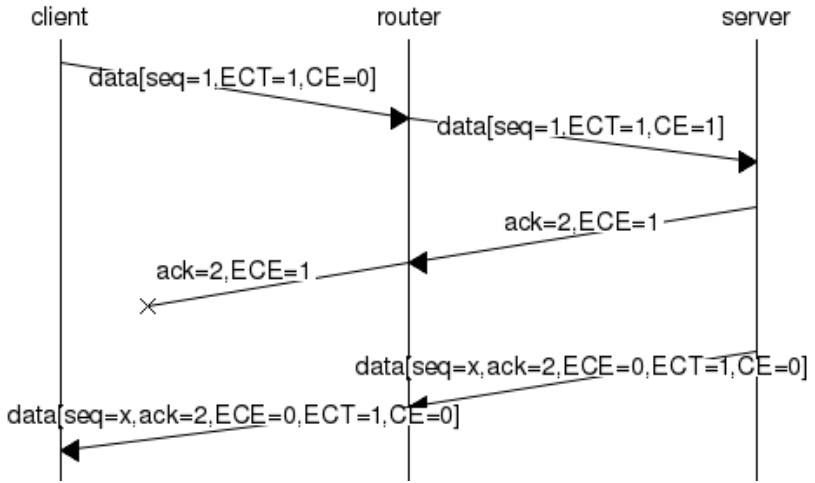
Murdoch UNIVERSITY

TCP Congestion Control

- You've heard about NewReno, which has been the standard congestion control mechanism for some time
  - Still used by Windows (and macOS until recently)
- There are dozens (if not hundreds) of different algorithms
  - Linux and macOS use CUBIC by default
  - Not all aim for maximum performance
- Some algorithms use estimates of network delay as an indicator of congestion
- This is a highly active research area in data communications

Active Queue Management

- What if routers actively tell TCP senders that there is congestion at configurable buffer delays?
- Active Queue Management (AQM) and TCP Early Congestion Notification (ECN)
  - Router marks packets when queue length or estimated delay is above threshold
  - TCP receiver echoes marks to TCP sender
  - TCP sender reduces congestion window



Murdoch UNIVERSITY

TCP Flow Control

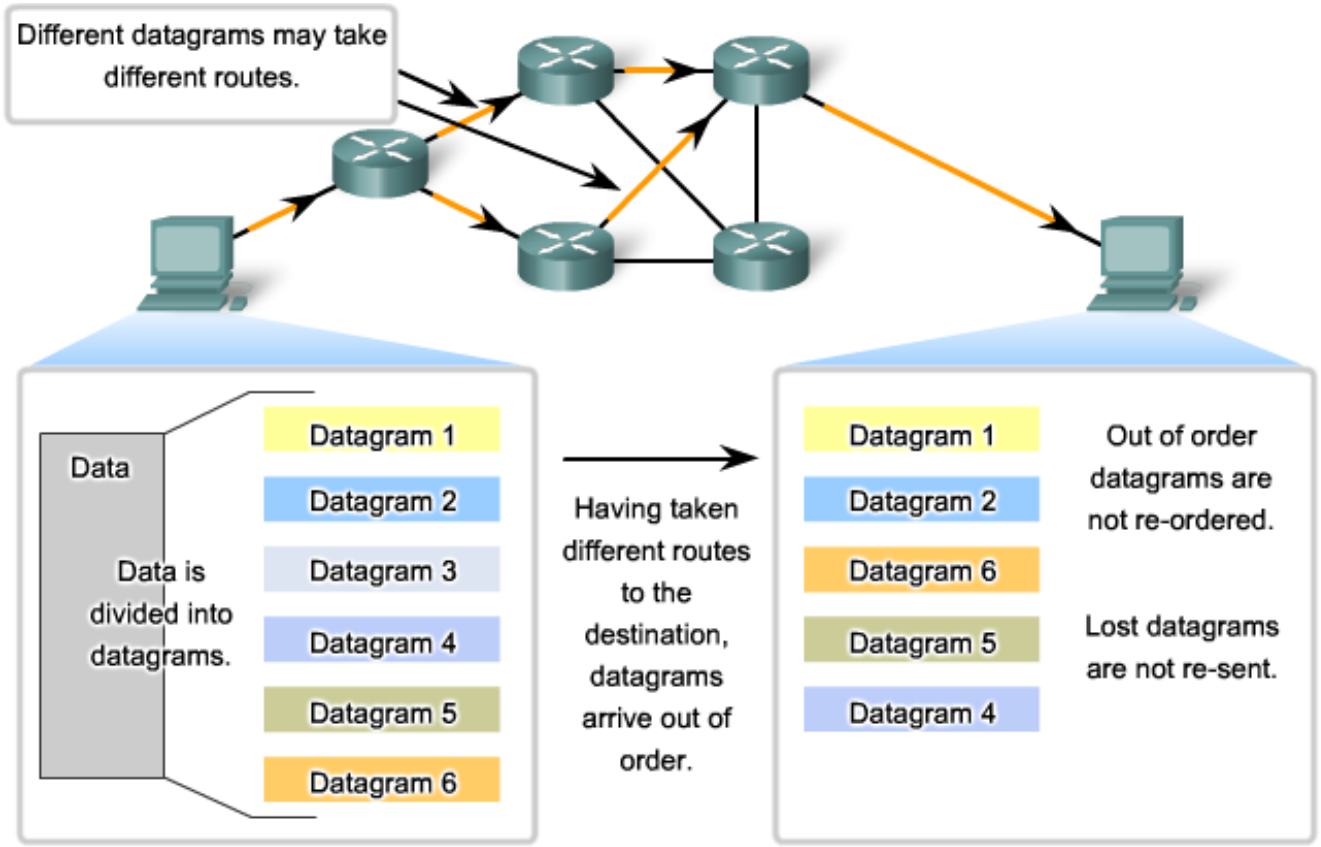
- Similar to congestion control, but prevents the sender from overflowing the receiver (instead of the whole network)
- The receiver advertises a **receive window**; the number of bytes it will accept before the next acknowledgement or window update
- Uses a windowing mechanism much like congestion control
- Sender will be restricted to the **minimum** of the congestion window and receive window

User Datagram Protocol (UDP)

- Recall that UDP is used when data must arrive in a timely manner
- Unlike TCP, UDP is a **connectionless** protocol
- 'Best effort' protocol, and has no equivalent to TCP acknowledgements
  - Not necessarily less reliable, just not guaranteed
- Datagrams may also arrive out of order
- No congestion or flow control
- Low per-packet overhead (simpler and smaller header)

UDP Reliability

UDP will not reassemble data back into order and will not resend lost datagrams because it is **connectionless** and **unreliable**



Why Use UDP?

- So why would an application use an unreliable service?
- **Answer 1:** Because resent data is useless and additional delay should be avoided
  - Teleconferencing/Skype – Additional delay for retransmissions more annoying than little dropouts
  - Online Games – No point in resending packets after action has already happened. Or would you like "laggy" game that pauses on packet loss?
- **Answer 2:** TCP is too complex or has too many overheads
  - Full TCP implementation may be too complex for machine with slow CPU, low RAM
  - Application can implement a simpler acknowledgement scheme on top of UDP to transport data
  - Example: Trivial File Transfer Protocol (TFTP)

Why Use UDP? (cont.)

- So why would an application use an unreliable service?
- **Answer 3:** UDP is connectionless and will not setup and teardown connection like TCP
  - Setting up and tearing down a TCP connection requires a minimum of 6 packets - unnecessary bandwidth usage
  - Setting up connection requires server to maintain state of connection - unnecessary CPU/RAM usage
  - For frequent short message exchanges, it is more efficient (and cheaper) in terms of bandwidth and server resources to use UDP
  - Example: DNS – Servers have to deal with thousands of requests per second. DNS request plus reply is only two packets, 1/4 of packet we would need with TCP. Plus flow and congestion control are useless for these short flows

UDP and TCP Protocol Headers

TCP Header

Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Source port																Destination port															
4	32	Sequence number																															
8	64	Acknowledgment number (if ACK set)																															
12	96	Data offset				Reserved 0 0 0				NS	CWR	ECE	URG	ACK	PSH	RST	SYN	FIN	Window Size														
16	128	Checksum																Urgent pointer (if URG set)															
20 ...	160 ...	Options (if *data offset* > 5. Padded at the end with "0" bytes if necessary.) ...																															

UDP Header

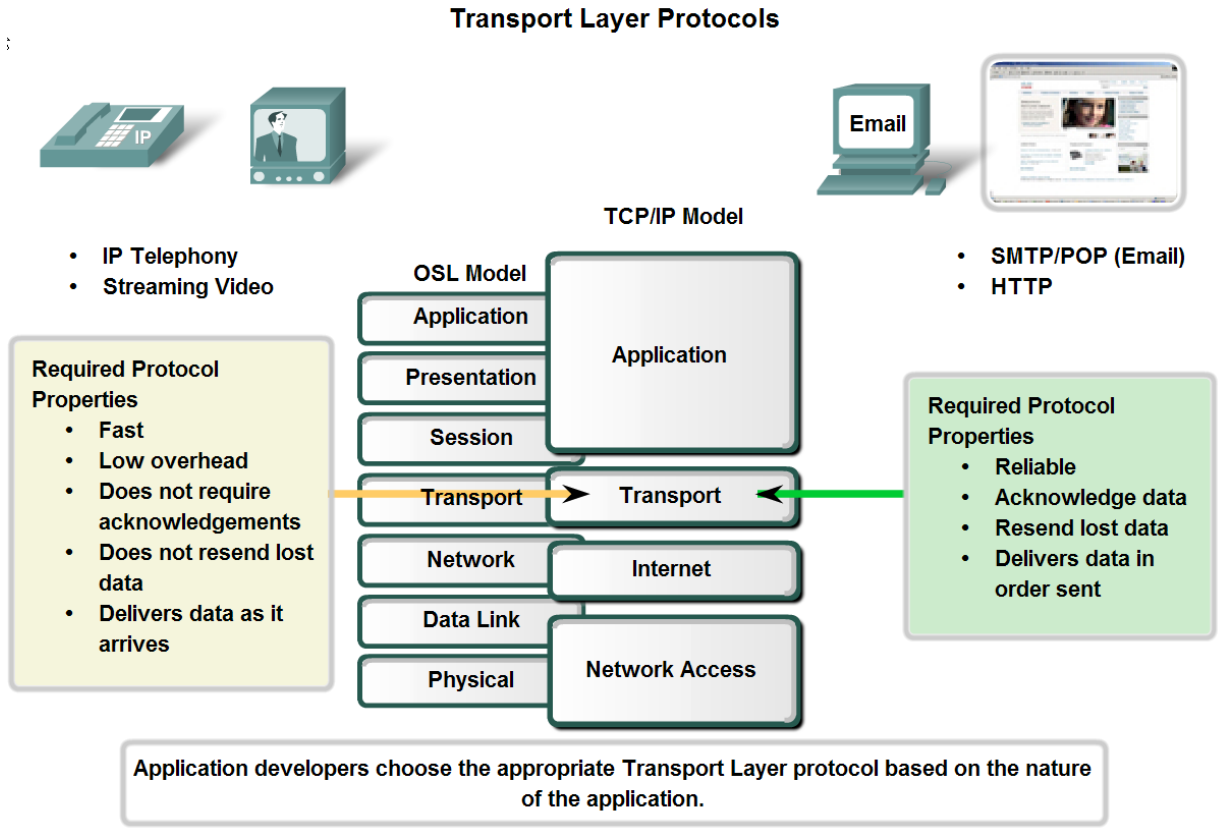
Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Source port																Destination port															
4	32	Length																Checksum															

Source: Wikipedia

Murdoch UNIVERSITY

TCP vs. UDP

- Neither protocol is better, it is about what's appropriate
- Application developers must decide what to use



Murdoch UNIVERSITY

Lecture Objectives

You should now be able to:

- Describe the role of the OSI Application, Presentation, and Session layers.
- Describe the Client/Server and Peer-to-Peer architectures.
- Describe the purpose of some widely used application layer protocols.
- Describe the purpose of the OSI Transport layer.
- Define ports with respect to the transport layer.
- Differentiate between the reliable and unreliable delivery of data.
- Describe the operation of the Transmission Control Protocol and User Datagram Protocol.
- Identify when it is appropriate to use each transport layer protocol.

Lecture Summary and the Week Ahead

- Today's lecture has examined the roles of the Application, Presentation and Session layers. The two major architectures for communications were also described.
- The purpose of the Transport layer was also examined. Specifically, in relation to TCP and UDP
- The readings for this week are Introduction to Networks – Chapters 9 and 10
- Participation Quiz 1 due this Sunday!
- In the labs: examining network traffic using Wireshark

Murdoch UNIVERSITY

Next Week

- We will continue descending the OSI model. Next up, the Network layer.
- Specifically, we will focus on IP addressing and subnetting. We will also start discussing the role of routers in data communications.
- Please make sure you bring a pen and paper!

Murdoch UNIVERSITY