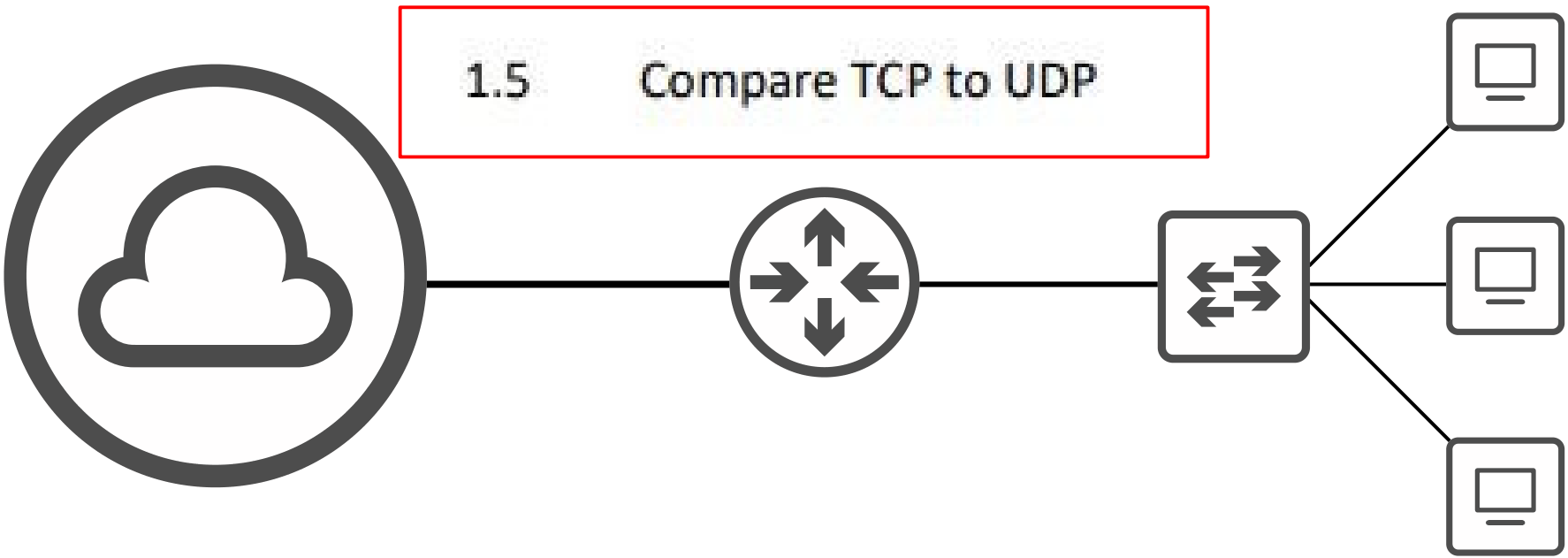
CCNA 200-301 Day 30

Comparing TCP & UDP



Things we'll cover

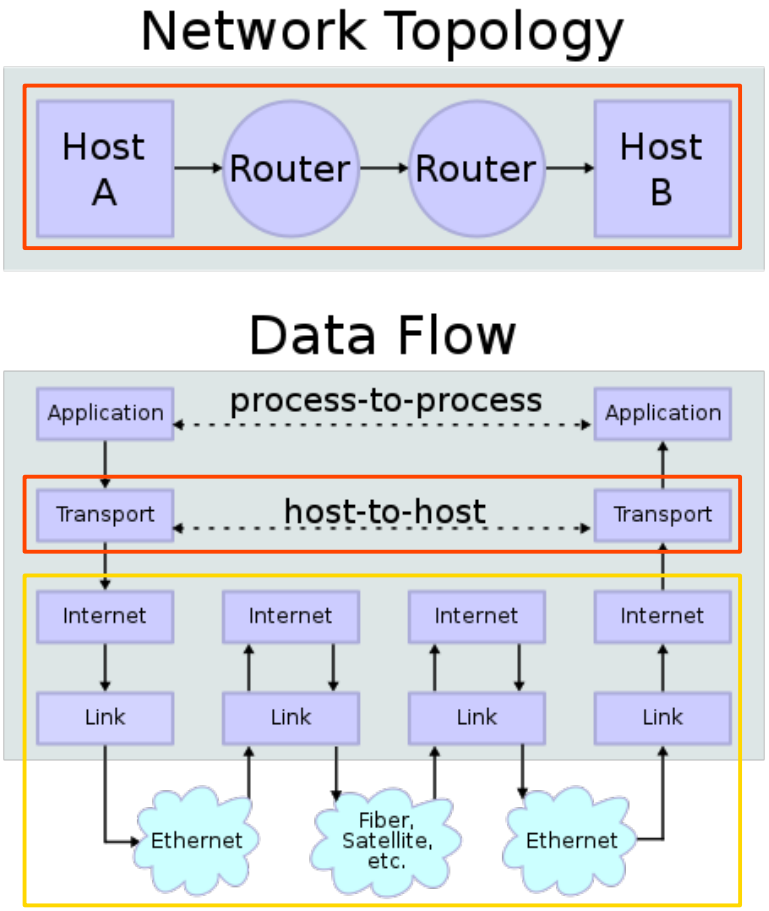
- Basics of Layer 4

1.5 Compare TCP to UDP

- TCP (Transmission Control Protocol)
- UDP (User Datagram Protocol)
- Comparing TCP & UDP

Functions of Layer 4 (Transport Layer)

- Provides transparent transfer of data between end hosts.



en:User:Kbrose (https://commons.wikimedia.org/wiki/File:IP_stack_connections.svg), „IP stack connections“, https://creativecommons.org/licenses/by-sa/3.0/legalcode

Functions of Layer 4 (Transport Layer)

- Provides transparent transfer of data between end hosts.

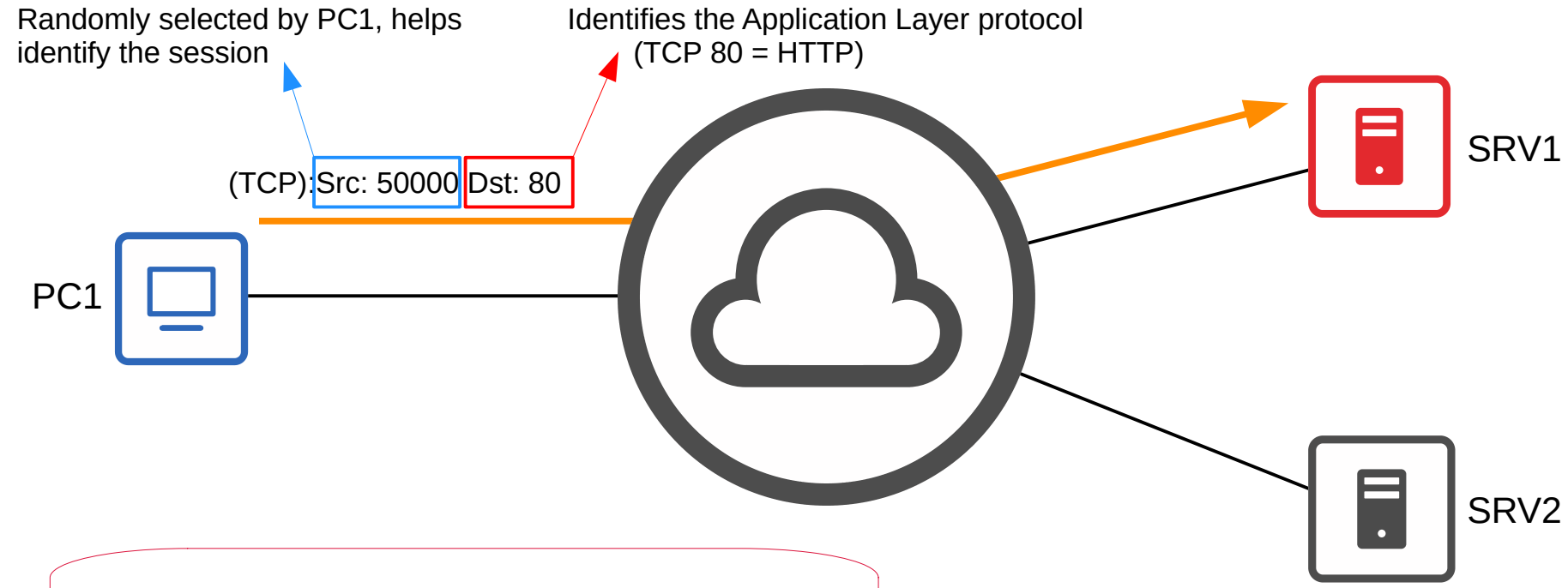
- Provides (or doesn't provide) various services to applications:
  - → reliable data transfer
  - → error recovery
  - → data sequencing
  - → flow control

- Provides Layer 4 addressing (**port** numbers).

  ↳Identify the Application Layer protocol
  ↳Provides session multiplexing.

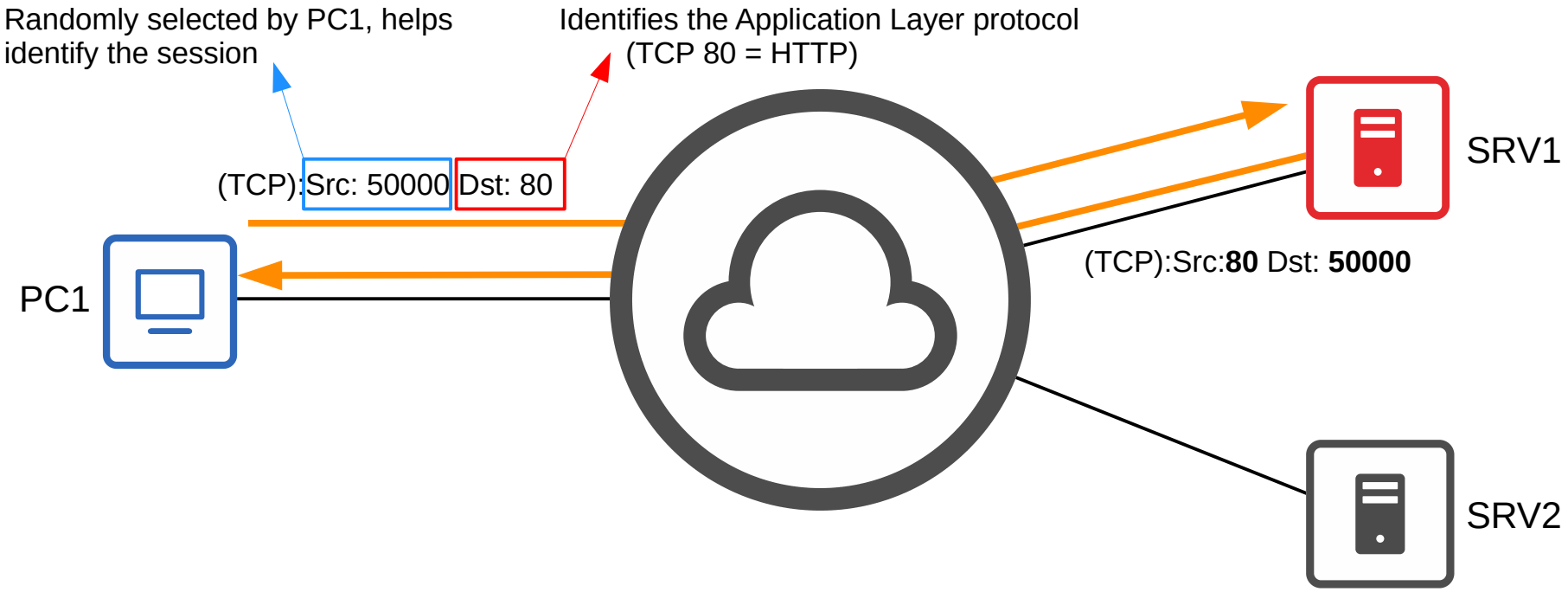
NOT the physical interfaces/ports on network devices

Port Numbers / Session Multiplexing

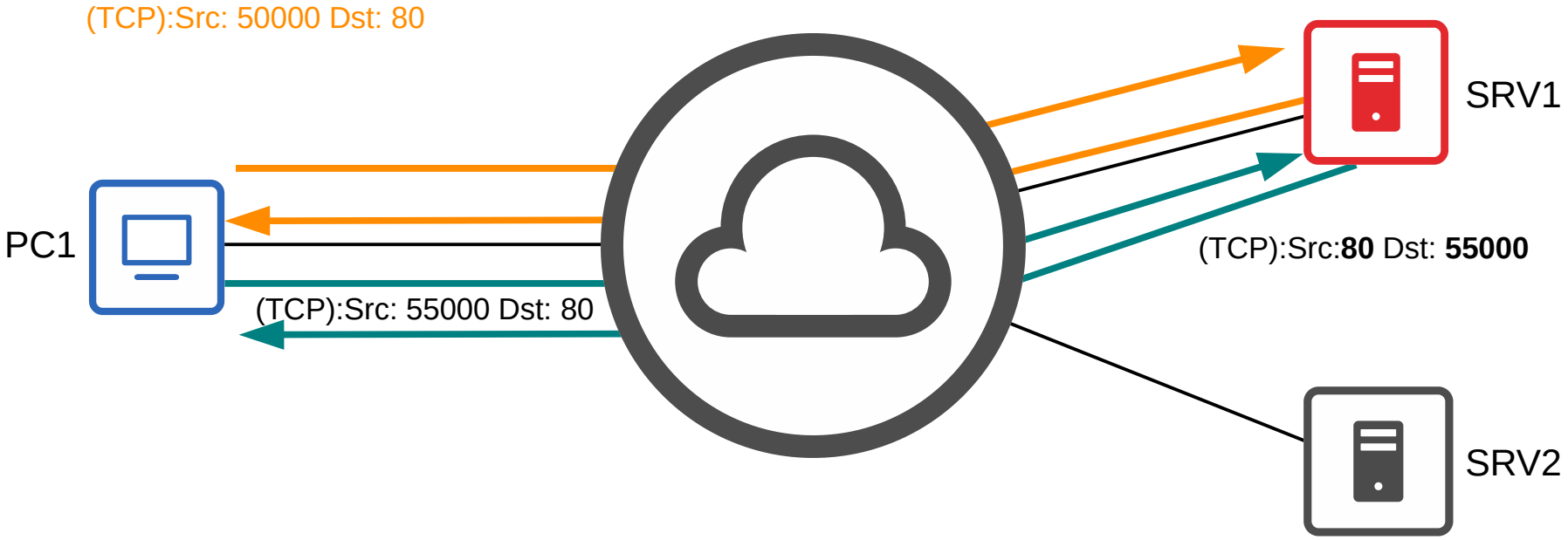


A session is an exchange of data between two or more communicating devices.

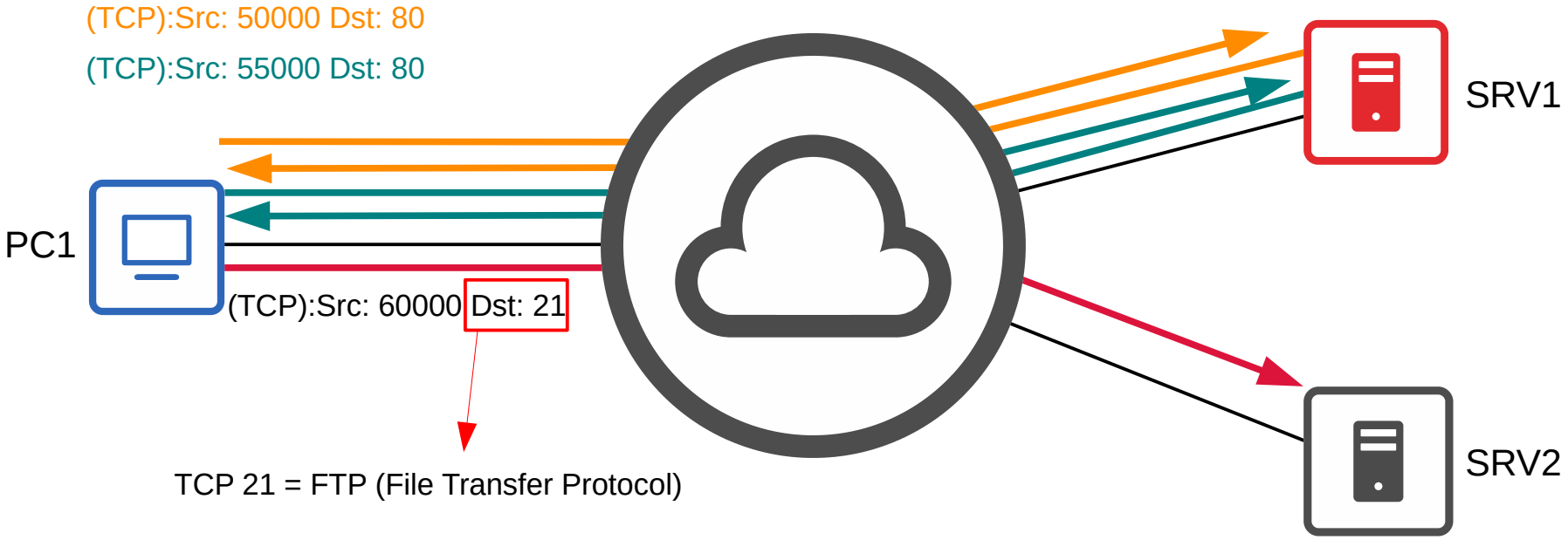
Port Numbers / Session Multiplexing



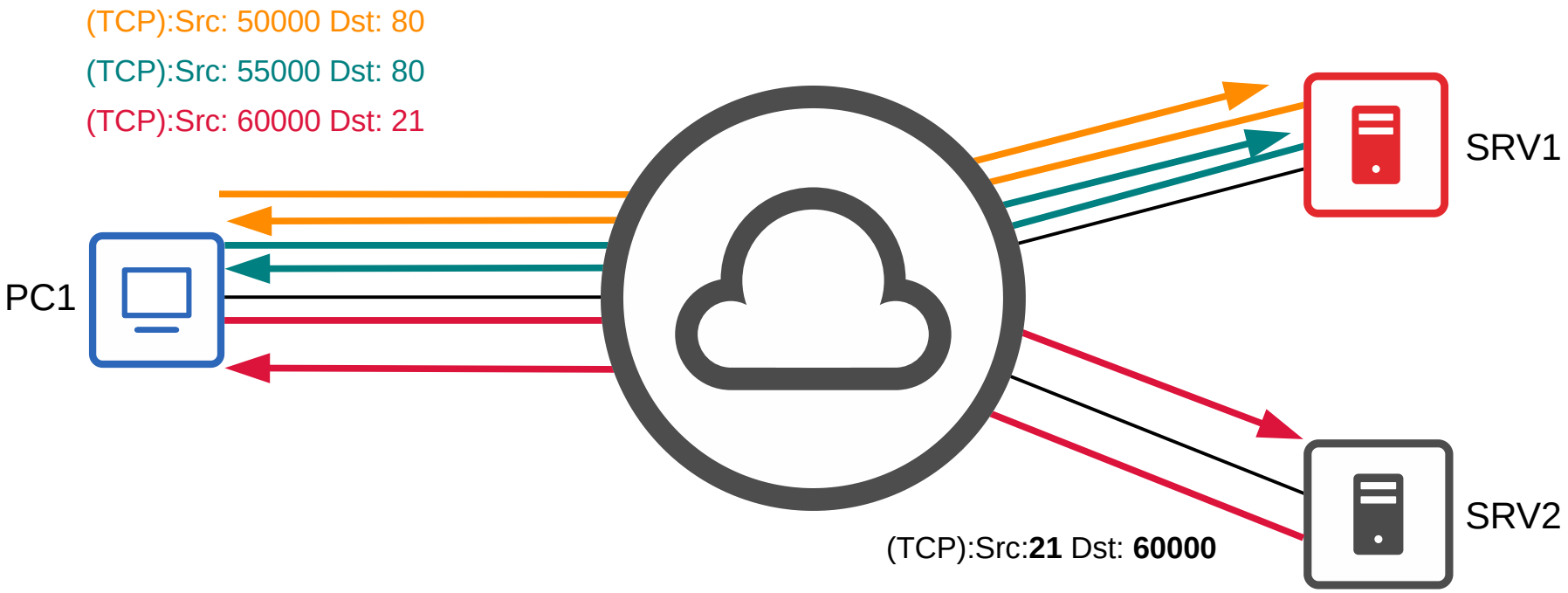
Port Numbers / Session Multiplexing



Port Numbers / Session Multiplexing



Port Numbers / Session Multiplexing



Functions of Layer 4 (Transport Layer)

- Provides transparent transfer of data between end hosts.

- Provides (or doesn't provide) various services to applications:
  - → reliable data transfer
  - → error recovery
  - → data sequencing
  - → flow control

- Provides Layer 4 addressing (**port** numbers).

  ↪Identify the Application Layer protocol
  ↪Provides session multiplexing.

  ↪The following ranges have been designated by IANA (Internet Assigned Numbers Authority)
  Well-known port numbers: 0 – 1023
  Registered port numbers: 1024 – 49151
  Ephemeral/private/dynamic port numbers: 49152 – 65535

(TCP):Src: **50000** Dst: 80
(TCP):Src: **55000** Dst: 80
(TCP):Src: **60000** Dst: 21

TCP (Transmission Control Protocol)

- TCP is connection-oriented.
  ↪ Before actually sending data to the destination host, the two hosts communicate to establish a connection. Once the connection is established, the data exchange begins.

- TCP provides reliable communication.
  ↪ The destination host must acknowledge that it received each TCP segment.
  ↪ If a segment isn't acknowledged, it is sent again.

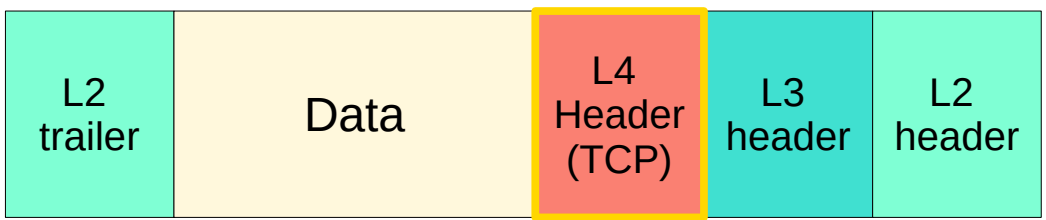
- TCP provides sequencing.
  ↪ Sequence numbers in the TCP header allow destination hosts to put segments in the correct order even if they arrive out of order.

- TCP provides flow control.
  ↪ The destination host can tell the source host to increase/decrease the rate that data is sent.

TCP Header

TCP segment header

Offsets	Octet	0								1								2								3							
Octet	Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
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.) ...																															



TCP Header

16 bits = 65536(2^{16}) available port numbers

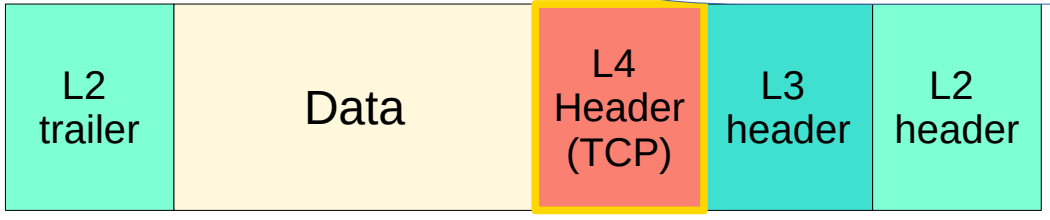
TCP segment header

Offsets	Octet	0								1								2								3							
Octet	Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
0	0	Source port																Destination port															
4	32	Sequence number																															
8	64	Acknowledgment number (if ACK set)																															
12	96	Data				Reserved 0 0 0			NS	CWR	ECE	URG	ACK	PSH	RST	SYN	FIN	Window Size															
16	128	Checksum																Urgent pointer (if URG set)															
2		ed at th																															

These two fields provide sequencing and reliable communication.

The window size is used for flow control.

These three flags are used to establish and terminate connections.



Establishing Connections: Three-Way Handshake

SYN flag

SYN flag, ACK flag

ACK flag

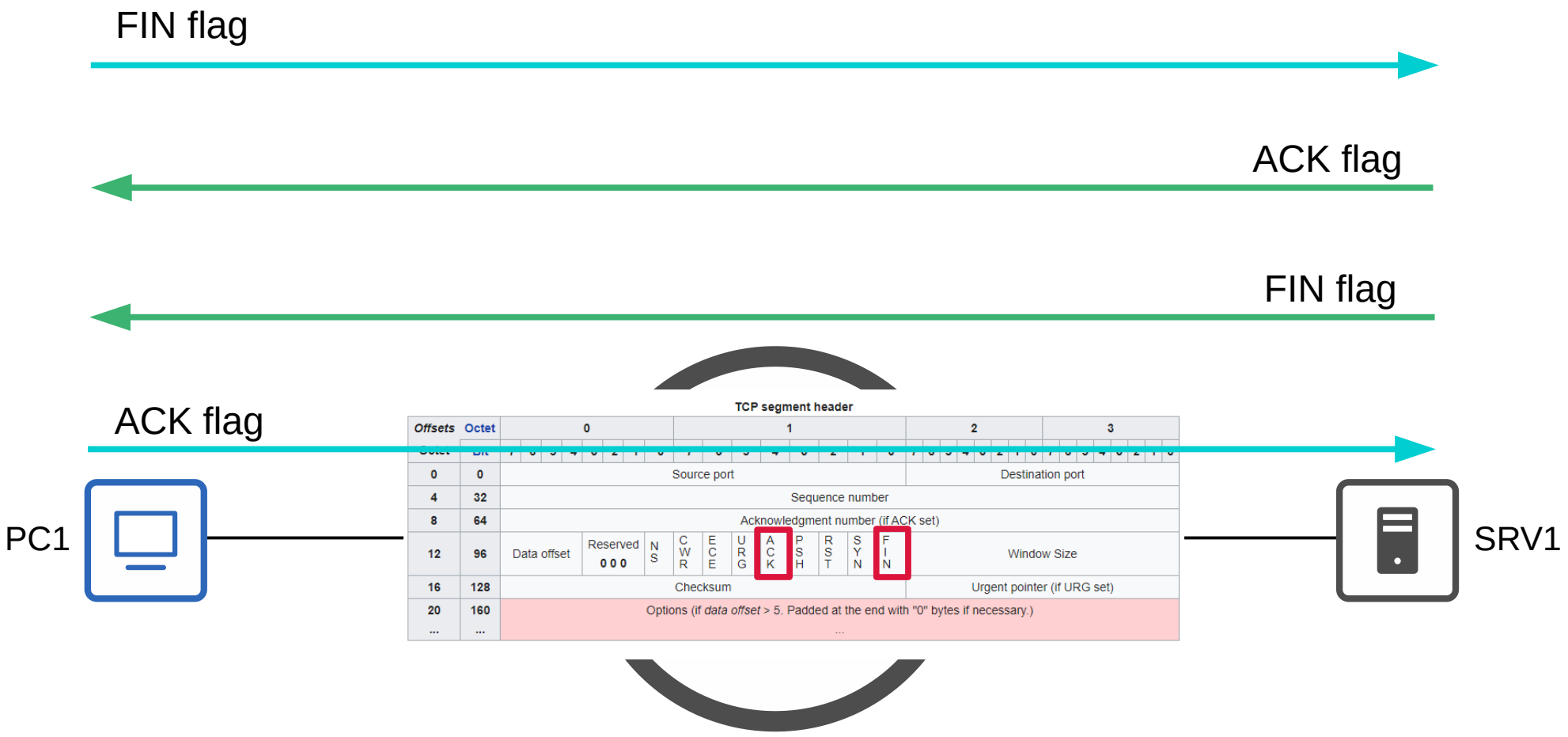
PC1

SRV1

TCP segment header

Offsets	Octet	0								1								2								3							
Octet	Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
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.) ...																															

Terminating Connections: Four-Way Handshake



TCP (Transmission Control Protocol)

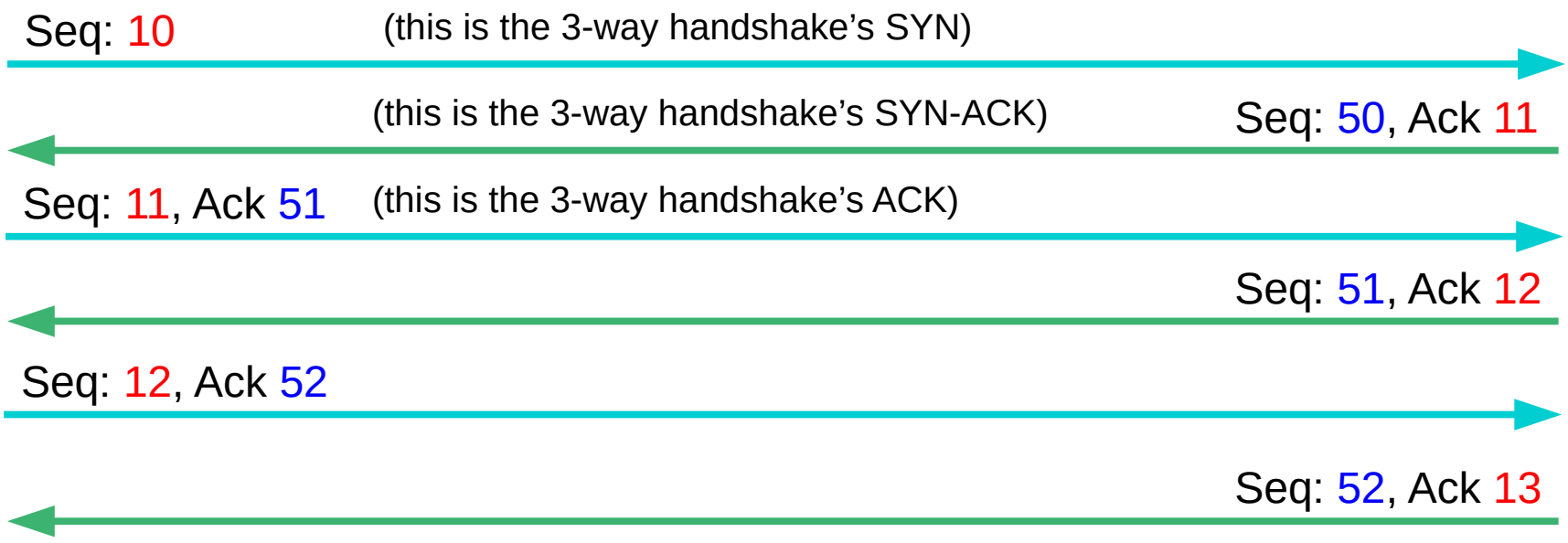
- TCP is connection-oriented.
  ↪ Before actually sending data to the destination host, the two hosts communicate to establish a connection. Once the connection is established, the data exchange begins.

- TCP provides reliable communication.
  ↪ The destination host must acknowledge that it received each TCP segment.
  ↪ If a segment isn't acknowledged, it is sent again.

- TCP provides sequencing.
  ↪ Sequence numbers in the TCP header allow destination hosts to put segments in the correct order even if they arrive out of order.

- TCP provides flow control.
  ↪ The destination host can tell the source host to increase/decrease the rate that data is sent.

TCP: Sequencing / Acknowledgment



- Hosts set a random initial sequence number.
- **Forward acknowledgment** is used to indicate the sequence number of the next segment the host expects to receive.

TCP (Transmission Control Protocol)

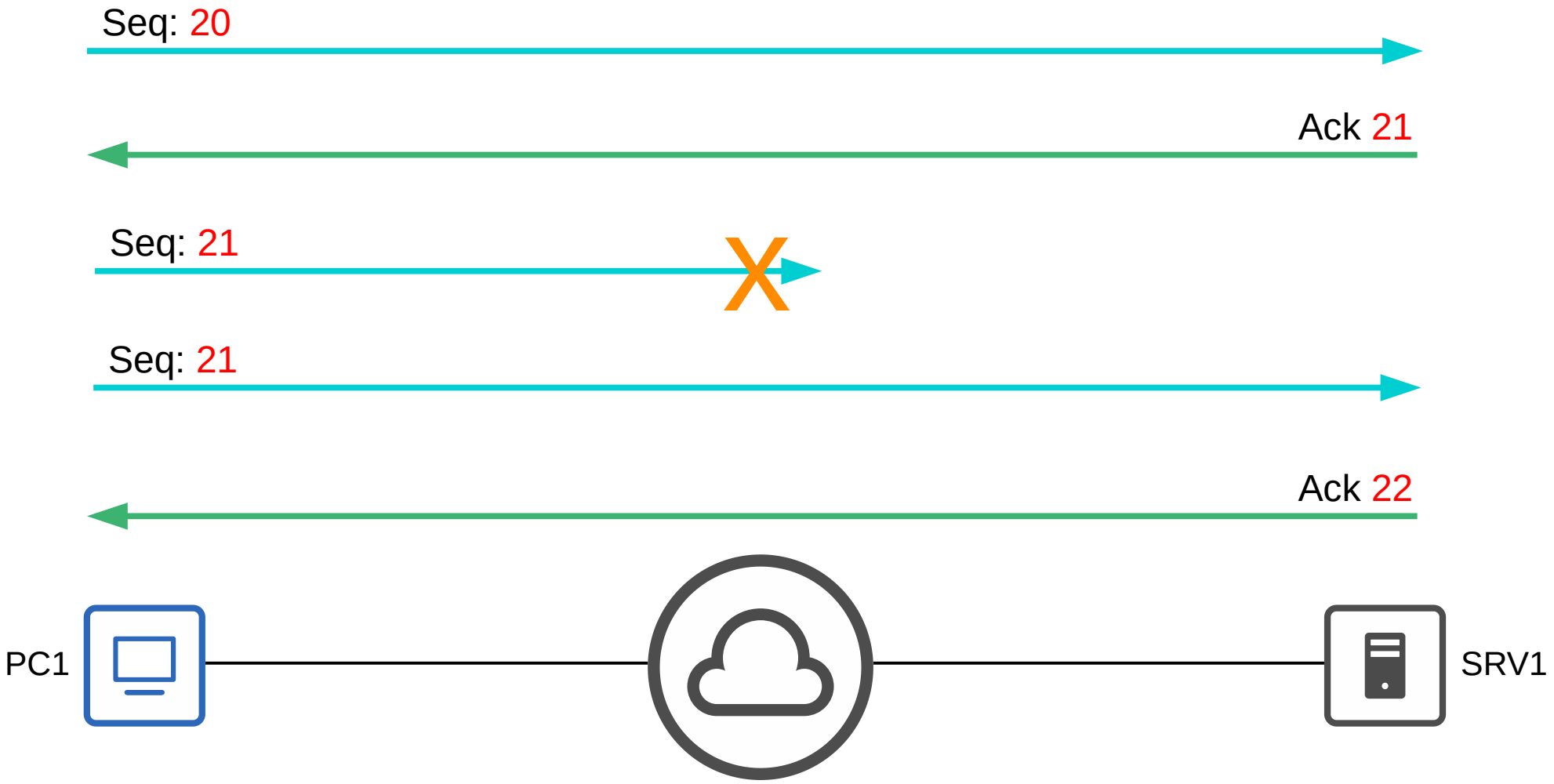
- TCP is connection-oriented.
  ↳ Before actually sending data to the destination host, the two hosts communicate to establish a connection. Once the connection is established, the data exchange begins.

- TCP provides reliable communication.
  ↳ The destination host must acknowledge that it received each TCP segment.
  ↳ If a segment isn't acknowledged, it is sent again.

- TCP provides sequencing.
  ↳ Sequence numbers in the TCP header allow destination hosts to put segments in the correct order even if they arrive out of order.

- TCP provides flow control.
  ↳ The destination host can tell the source host to increase/decrease the rate that data is sent.

TCP Retransmission



TCP (Transmission Control Protocol)

- TCP is connection-oriented.
  ↪ Before actually sending data to the destination host, the two hosts communicate to establish a connection. Once the connection is established, the data exchange begins.

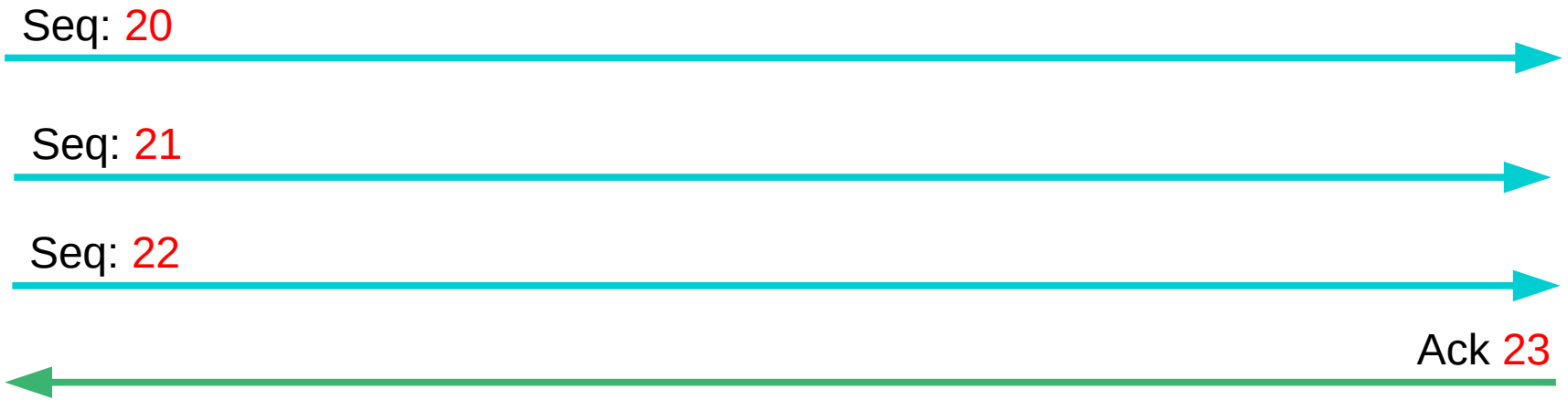
- TCP provides reliable communication.
  ↪ The destination host must acknowledge that it received each TCP segment.
  ↪ If a segment isn't acknowledged, it is sent again.

- TCP provides sequencing.
  ↪ Sequence numbers in the TCP header allow destination hosts to put segments in the correct order even if they arrive out of order.

- TCP provides flow control.
  ↪ The destination host can tell the source host to increase/decrease the rate that data is sent.

TCP Flow Control: Window Size

- Acknowledging every single segment, no matter what size, is inefficient.
- The TCP header's **Window Size** field allows more data to be sent before an acknowledgment is required.
- A 'sliding window' can be used to dynamically adjust how large the window size is.



In all of these examples, I used very simple sequence numbers. In real situations, the sequence numbers get much larger and do not increase by 1 with each message.
For the CCNA, just understand the concepts and don't worry about the exact numbers.

TCP (Transmission Control Protocol)

- TCP is connection-oriented.
  ↳ Before actually sending data to the destination host, the two hosts communicate to establish a connection. Once the connection is established, the data exchange begins.

- TCP provides reliable communication.
  ↳ The destination host must acknowledge that it received each TCP segment.
  ↳ If a segment isn't acknowledged, it is sent again.

- TCP provides sequencing.
  ↳ Sequence numbers in the TCP header allow destination hosts to put segments in the correct order even if they arrive out of order.

- TCP provides flow control.
  ↳ The destination host can tell the source host to increase/decrease the rate that data is sent.

1.5 Compare TCP to UDP

UDP (User Datagram Protocol)

- UDP is **not** connection-oriented.
  ↪ The sending host does not establish a connection with the destination host before sending data. The data is simply sent.

- UDP **does not** provide reliable communication.
  ↪ When UDP is used, acknowledgments are not sent for received segments. If a segment is lost, UDP has no mechanism to re-transmit it. Segments are sent 'best-effort'.

- UDP **does not** provide sequencing.
  ↪ There is no sequence number field in the UDP header. If segments arrive out of order, UDP has no mechanism to put them back in order.

- UDP **does not** provide flow control.
  ↪ UDP has no mechanism like TCP's window size to control the flow of data.

UDP datagram 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															

JEREMY'S IT LAB

Comparing TCP & UDP

TCP segment header

Offsets	Octet	0								1								2								3							
Octet	Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
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 datagram 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															

Comparing TCP & UDP

- TCP provides more features than UDP, but at the cost of additional **overhead**.
- For applications that require reliable communications (for example downloading a file), TCP is preferred.
- For applications like real-time voice and video, UDP is preferred.
- There are some applications that use UDP, but provide reliability etc within the application itself.
- Some applications use both TCP & UDP, depending on the situation.

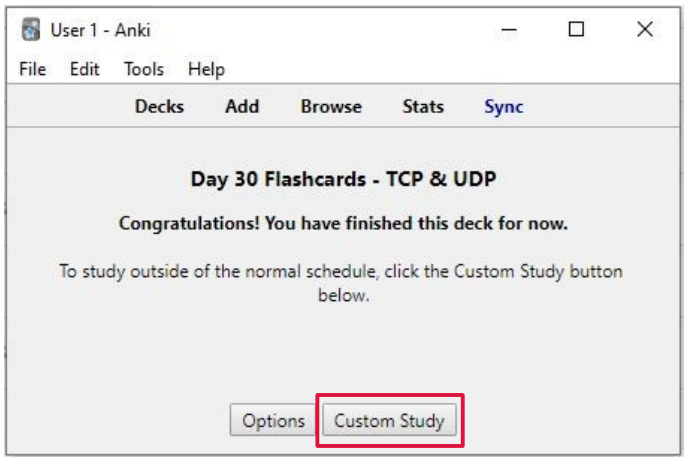
Comparing TCP & UDP

TCP	UDP
Connection-oriented	Connectionless
Reliable	Unreliable
Sequencing	No sequencing
Flow control	No flow control
Use for downloads, file sharing, etc	Used for VoIP, live video, etc

Port Numbers

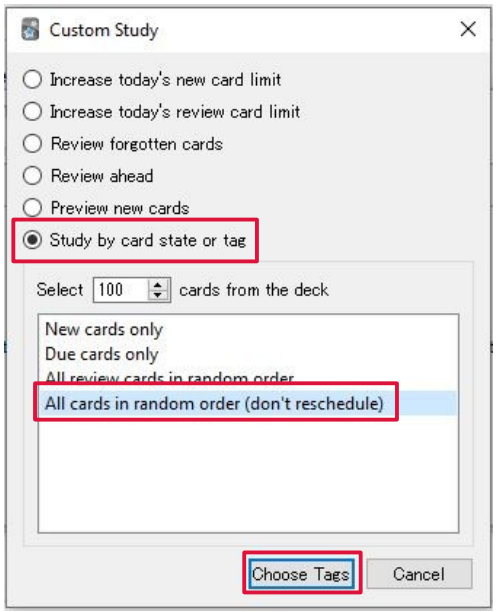
TCP

- FTP data (20)
- FTP control (21)
- SSH (22)
- Telnet (23)
- SMTP (25)
- HTTP (80)
- POP3 (110)
- HTTPS (443)



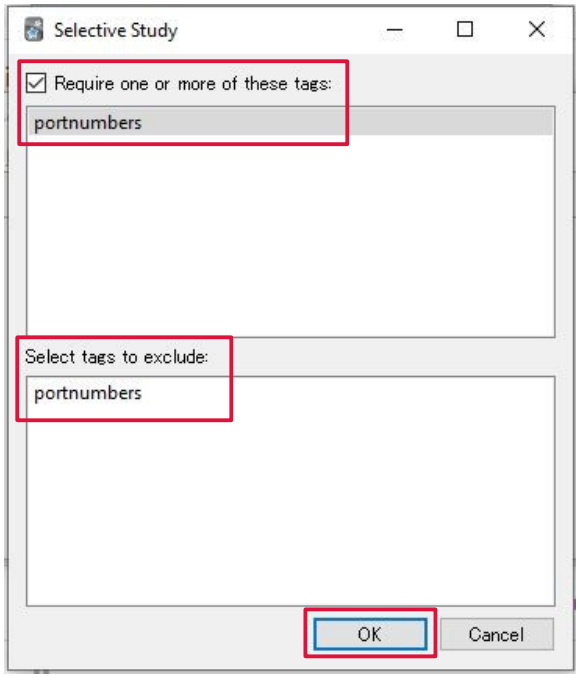
UDP

- DHCP server (67)
- DHCP client (68)
- TFTP (69)
- SNMP agent (161)
- SNMP manager (162)
- Syslog (514)



TCP & UDP

- DNS (53)



Things we covered

- Basics of Layer 4

1.5 Compare TCP to UDP

- TCP (Transmission Control Protocol)
- UDP (User Datagram Protocol)
- Comparing TCP & UDP

Quiz 1

Which of the following is a well-known port number, as defined by IANA?

a) 1010

b) 2001

c) 4023

d) 65000

↪The following ranges have been designated by IANA (Internet Assigned Numbers Authority)
Well-known port numbers: 0 – 1023
Registered port numbers: 1024 – 49151
Ephemeral/private/dynamic port numbers: 49152 – 65535

Quiz 2

According to IANA specifications, what range of port numbers should hosts select from when randomly selecting a source Layer 4 port number?

a) Well-known

b) Registered

c) Ephemeral

d) Reserved

↳The following ranges have been designated by IANA (Internet Assigned Numbers Authority)
Well-known port numbers: 0 – 1023
Registered port numbers: 1024 – 49151
Ephemeral/private/dynamic port numbers: 49152 – 65535

Quiz 3

Which of the following are features of TCP but not UDP? (select three)

a) Layer 4 addressing

b) Error recovery

c) Session multiplexing

d) Flow control

e) Sequencing

Quiz 4

Which of the following Application Layer protocols use TCP to provide reliable communications? (select three)

a) SMTP

b) SNMP

c) HTTPS

d) DHCP

e) Syslog

f) SSH

Quiz 5

PC1 and SRV1 have an active TCP connection. SRV1 receives a TCP segment from PC1 with a sequence number of 27. When SRV1 acknowledges the segment, what will the value of the Acknowledgment field in the TCP header be? Assume a TCP window size of 1.

a) 26

b) 27

c) 28