Jared Ralola Part2

HOST A $\xleftrightarrow{TCP}$ HOST B

i) First segment sequence # is 227 and since the first segment is 128 bytes:

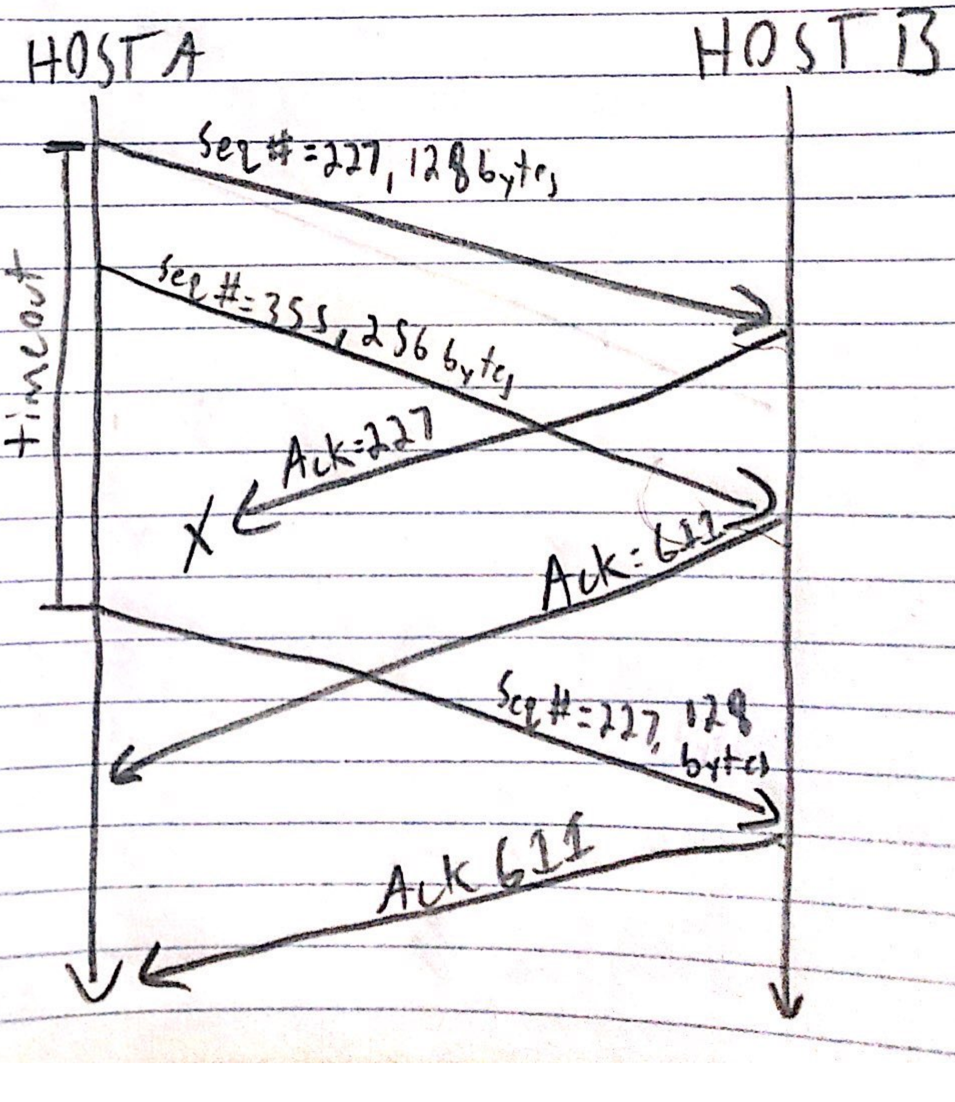
227 + 128 = 355

$\boxed{\text{Second segment sequence \# is 355}}$

ii) First acknowledgement # is $\boxed{355}$

iii) The acknowledgement # is $\boxed{227}$ as it's still waiting for the first segment.

iv)



Jared Rolola Part 3

$$\text{Timeout Interval} = \text{EstimatedRTT} + 4(\text{DevRTT})$$

$$\text{EstimatedRTT} = (1-\alpha)\,\text{EstimatedRTT} + \alpha(\text{SampleRTT})$$

$$\text{DevRTT} = (1-\beta)\,\text{DevRTT} + \beta\,|\text{SampleRTT} - \text{EstimatedRTT}|$$

SampleRTT	EstimatedRTT	DevRTT	Timeout Interval
10ms	→ Example, calculated below		
10ms	15.625ms	2.8125ms	26.875ms
10ms	14.2188ms	3.1641ms	26.8752ms
10ms	13.1641ms	3.1641ms	25.8205ms
10ms	12.3731ms	2.9664ms	24.2387ms
100ms	34.2798ms	18.6549ms	108.8994ms
100ms	50.7099ms	26.3137ms	155.9647ms
100ms	63.0324ms	28.9772ms	178.9412ms
100ms	72.2743ms	28.6643ms	186.9315ms
100ms	79.2057ms	26.6968ms	185.9929ms
10ms	61.9043ms	32.9987ms	193.8992ms
100ms	71.9282ms	31.8920ms	198.9962ms
1000ms	303.5712ms	198.0262ms	1095.676ms
1000ms	477.6784ms	279.1001ms	1594.0788ms
1000ms	609.2588ms	307.2604ms	1837.3004ms
10ms	458.6941ms	342.6188ms	1829.1693ms

Example) $\text{EstimatedRTT} = (1-0.25)\cdot 20\text{ms} + 0.25(10\text{ms}) = \boxed{17.5\text{ms}}$

$\text{DevRTT} = (1-0.25)\cdot 0\text{ms} + 0.25\cdot|10\text{ms} - 17.5\text{ms}|$
$\boxed{= 1.875\text{ms}}$

$\text{Timeout Interval} = 17.5\text{ms} + 4\cdot(1.875\text{ms}) = \boxed{25\text{ms}}$

Jared Raiola Part 4

a) TCP Checksum is computed over the TCP header and the TCP pseudoheader. This gives protection against misrouted segments, as the pseudoheader contains the source address, the destination address, the protocol and the TCP length.
(Pages 15 and 16 of TCP RFC 793)

b) A TCB (transmission control block) stores all the variables used to maintain a TCP connection. This is known as a connection record, which includes:

↳ Local and remote socket numbers

↳ Security and precedence of the connection

↳ Pointers to the users send and receive buffers

↳ Pointers to the retransmit queue and to the current segment

↳ Variables related to the send and receive sequence numbers

(Page 19 of TCP RFC 793)

Jared Raiola Part 4 Continued

c) i) Interface Primitives:

↳ Open → new connection
creates a TCB

↳ Send → send data over the connection

↳ Receive → Receive data over the connection

↳ Close → Close the connection

↳ Status → status of the connection
from the TCB

↳ Abort → All operations cancelled.
Connection closes and resets

(Pages 45 - 50 of TCP RFC 793)

ii)

RFC calls		Socket calls
Open	←→	socket
send	←→	send, sentto
receive	←→	recv, recvfrom
close	←→	close
status	←→	poll
abort	x	N/A
N/A	x	bind
	x	listen
	x	connect
	x	accept
	x	gethostbyname
	x	select
	x	getsockopt
↓	x	setsockopt