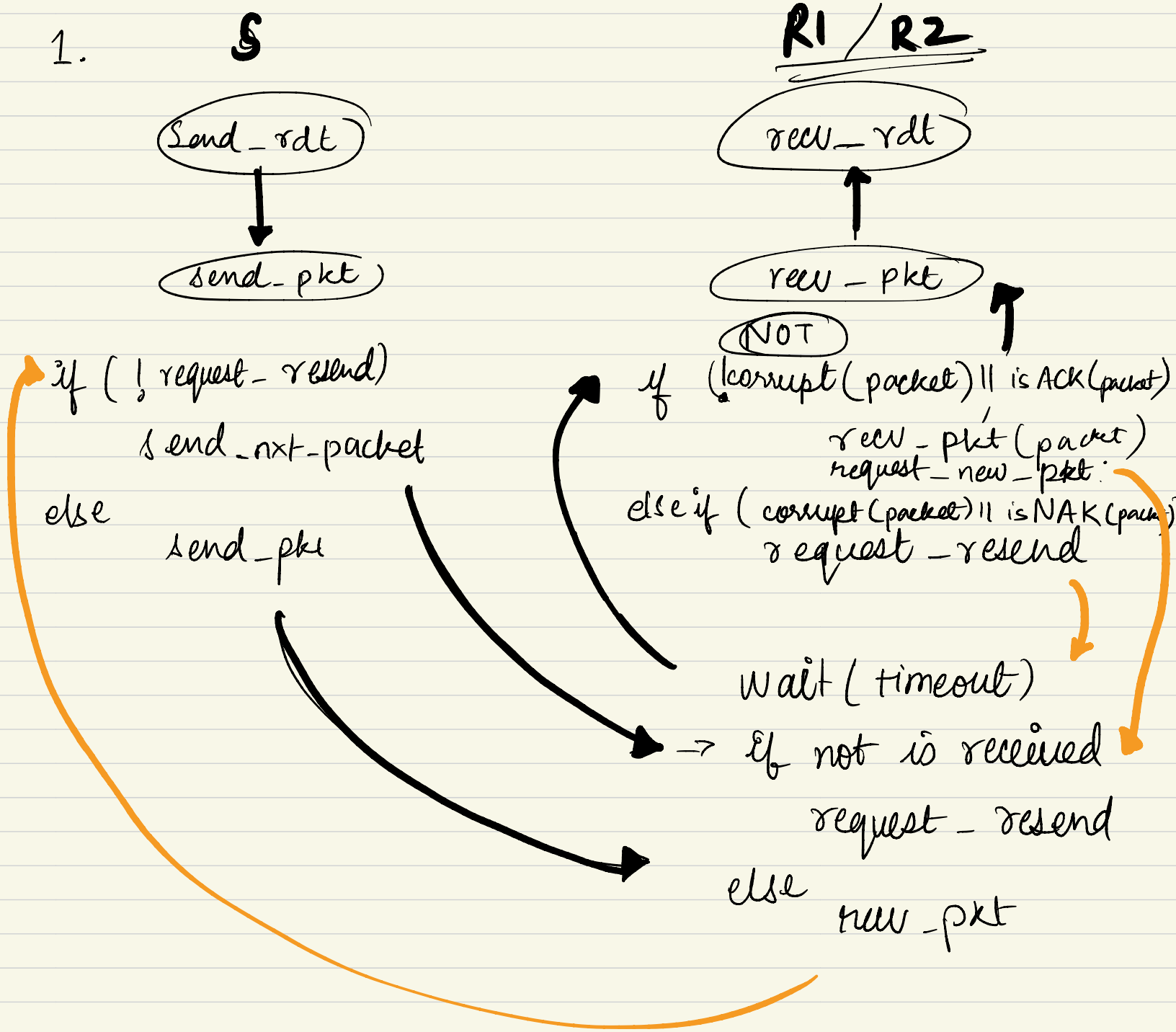
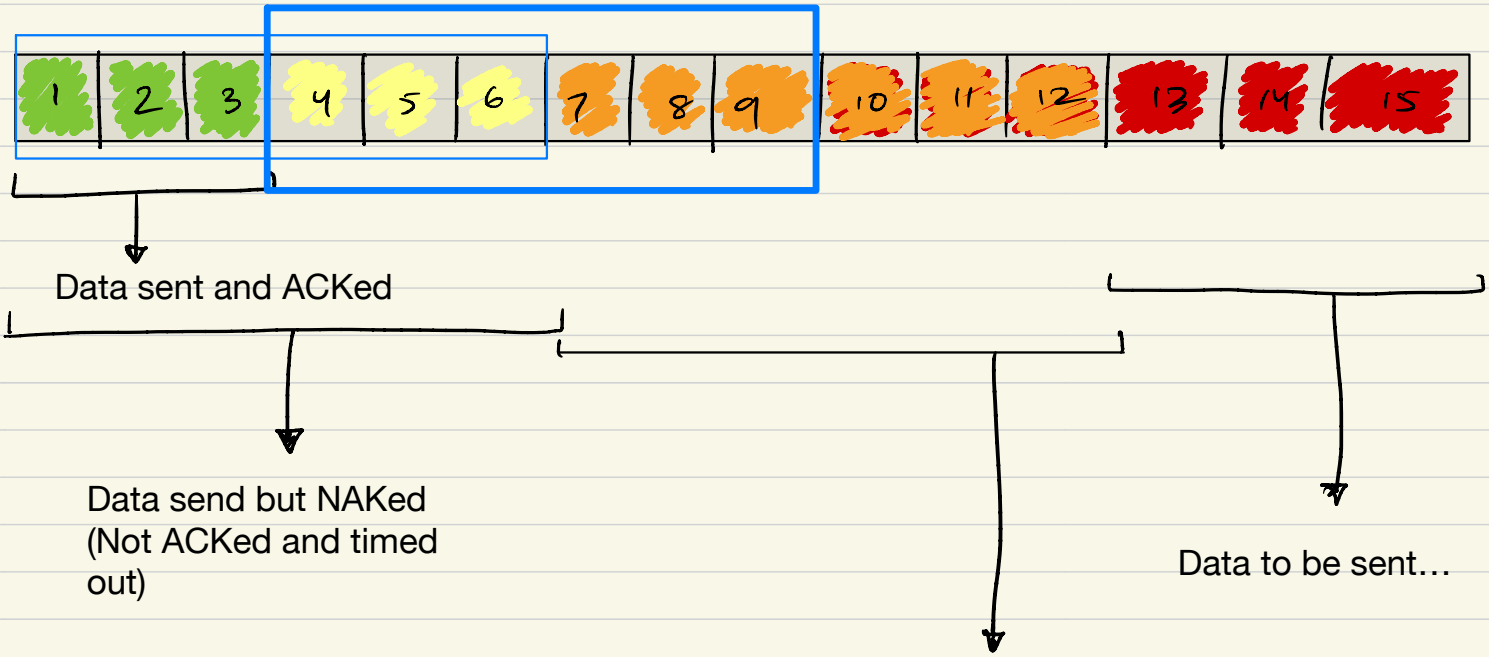
1.


S
R1 / R2
send_rdt
send_pkt
recv_rdt
recv_pkt
if (! request_resend)
send_nxt_packet
else
send_pkt
NOT
if (!corrupt(packet) || isACK(packet)
recv_pkt(packet)
request_new_pkt;
else if (corrupt(packet) || isNAK(packet)
request_resend
wait (timeout)
-> if not is received
request_resend
else
new_pkt

2. Flow control is the process of controlling the the rate of data transmission between two nodes to prevent a faster sender to overwhelm a slower receiver. Whereas, congestion control is the process of controlling the flow of data only when there is congestion in the network. In simple words, congestion control prevents the senders from overwhelming the network whereas flow control prevents a sender from overwhelming a receiver.

TCP implements flow control through sliding window.



The green packets were sent successfully therefore the minimum bound of the sliding window will shift to packet 4. And since 3 packets were sent successfully, the maximum bound will increase to packet 9. Please refer to the thicker blue box in the figure above. This is the window that keeps sliding forward along the stream of packets until the data has been fully transmitted. The receiver has to advertise the maximum receive size based on which the sliding window can be calculated. This determines the transmission rate. And sliding door controls the flow of data.

TCP congestion control uses congestion window (also just a sliding window) but in addition to congestion policies.

Congestion policy is broken into 3 phases:

1. Slow Start Phase - Starts with a small number and then increases exponentially to reach the maximum possible threshold in the network.
2. Congestion Avoidance Phase - After reaching the threshold, the rate is only increased by 1.
3. Congestion Detection Phase - If congestion is detected, i.e. based on 2 cases, 1. Retransmission due to timeout, 2. Retransmission due to 3 duplicate ACKs. The CDP resets to Phase 1 if case 1 takes place, and Phase 2 if case 2 takes place.

3.

	SOURCE	PORT	DESTINATION	PORT
i)	10.0.0.1	1025	1.2.3.4	80
ii)	10.0.0.2	1026	1.2.3.4	80
iii)	5.6.7.8	11111	1.2.3.4	80
iv)	5.6.7.8	11111	1.2.3.4	80
v)	1.2.3.4	80	5.6.7.8	11111
vi)	1.2.3.4	80	10.0.0.1	1025

The ports circled are ports assumed

4. 1. 3 subnets are present on the network. The smallest prefix is 1.1.x.y where ‘x’ is the subnet and ‘y’ will be the devices in that subnet.

2. It really depends on the requirement of the business. If they want every device in every subnet to have a static IP on the internet, they will have to purchase 768 addresses. But if they want every subnet to have on static IP then it would be 3 public IPs, but the network can function on just 1 public IP address too as the network connection can be divided internally. But since the question asks the number of IP prefixes, I think it is one prefix can be bought and divided into creating an internal network structure with 256 static public addresses that can directly access the internet and doesn’t require NAT, but this network since there are 3 subnets and a theoretical max of 768 devices.

3.

Network Destination	Subnet	Gateway
1.1.1.0/24	255.255.255.0	1.1.1.0/24
1.1.2.0/24	255.255.255.0	1.1.4.1/32
1.1.3.0/24	255.255.255.0	1.1.5.1/32
0.0.0.0/0	0.0.0.0 (N/A)	1.1.1.0/0