A0232935A

Liu Weihao

Q1

Link layer

pros:

1. Efficiency. Link layer retransmissions avoid having to retransmit over the entire path. Transport and application layer retransmissions must be sent over the entire path, thus incurring higher latency and wasting bandwidth.
2. Link layer retransmissions can hide non-congestion losses from higher layer protocols.

cons:

1. End host to end host retransmissions are still necessary. A packet can be lost can after being successfully received at a router (due to congestion or errors). A route flap can send packets on a path that causes their TTL to expire before arriving at the destination. Some routers may not implement link layer retransmissions.
2. Link layer retransmissions add some complexity and overhead.
3. A packet may be retransmitted at multiple layers simultaneously. For example, a packet may be delayed before reaching a wireless link. The link layer protocol retransmits the packet, but the sender has already timed out and retransmits the packet. The link layer's retransmission is superfluous.

Transport layer

pros:

1. Transport layer retransmissions avoid having to reimplement retransmission functionality in applications. Many applications can use the same transport protocol implemented in a library or in the operating system, reducing bugs, implementation time, etc.

Application layer

pros:

1. Some applications can tolerate loss better than the increased delay of retransmissions.
2. Data may be lost above the transport layer.

Q2

If we use implicit message, the sender will retransmit the message after timeout, or use RTT the estimate the package lose. After that, the sender will use AIMD protocol to achieve the optimal point.

If we use explicit, the router will indicate that there is congestion by setting a bit in the packet and receiver send that information back in ACK, so receiver can reduce the transmit rate to avoid the package lose.

Q3

End-to-end principle. This layer guarantee that the data can be transmitted from sender to it's destination.

Q4

Host receives IP datagrams. Each datagram has source and destination IP address, each datagram carries one transport-layer segment, each segment has source and destination port number. Then host uses IP addresses and port numbers to direct the segment to appropriate socket

Q5

We can assume that, there are 4 packets, $N-1,\ N,\ N+1,\ N+2$. The sending order is $N-1 \rightarrow N+2$. If the order in which receiver received the packets was different, the receiver will receive different ACK signal.

Receive order:

$N-1,\ N,\ N+1,\ N+2$, sender will receive 1 ACK(n)

$N-1,\ N,\ N+2,\ N+1$, sender will receive 1 ACK(n)

$N-1,\ N+1,\ N,\ N+2$, sender will receive 2 ACK(n)

$N-1,\ N+1,\ N+2,\ N$**, sender will receive 3 ACK(n)**

$N-1,\ N+2,\ N,\ N+1$, sender will receive 2 ACK(n)

$N-1,\ N+2,\ N+1,\ N$**, sender will receive 3 ACK(n)**

If packet N is loss,

$N-1,\ N+1\ ,\ N+2$**, sender will receive 3 ACK(n)**

$N-1,\ N+2\ ,\ N+1$**, sender will receive 3 ACK(n)**

In conclusion, if the packet out of order, there is 33% probability of receiving 3 ACK(n). But if the packet is lost, that probability will be 100%.

Q6

TCP sets timeout as a function of the RTT. We can estimate the RTT by watching the ACKs.

1. Smooth estimate: keep a running average of the RTT:

$$EstimatedRTT = a * EstimatedRTT + (1-a) * SampleRTT$$

2. Compute timeout:

$$Timeout = 2 * EstimatedRTT$$

3.

Q7

No matter which point we start from in the plane, AIMD can reach the optimal point. But MIMD, AIAD can't

Q8

TCP fairness requires that a new protocol receive a no larger share of the network than a comparable TCP flow. This is important as TCP is the dominant transport protocol on the Internet, and if new protocols acquire unfair capacity they tend to cause problems such as congestion collapse.

The resources TCP allocates in a fair manner is bandwidth

Q9

The throughput is the average rate that packets are successfully decoded at the receiver. Note that the rate that packets are sent by the sender is an upper bound on the actual throughput and since it is easily computable, we use it to estimate the throughput.

Q10

We need both of them because of efficiency. We don't need all of them, end-to-end is enough. But, when packet lose or network congestion happened, if we only use end-to-end reliability, the data needs to be retransmitted from sender to it's destination. If we use hop-by-hop reliability, the data only needs to be retransmitted from the congestion node. That more efficiency and will save a lot of time.