COMP90007 Internet Technologies
Week 10 Workshop

Semester 2, 2020

Suggested solutions

Question 1

Why does UDP exist? Would it not have been enough to just let the user processes send raw IP packets?

Answer:

No. IP packets contain IP addresses, which specify a destination machine. Once such a packet arrived, how would the network handler know which process to give it to? UDP packets contain a destination port. This information is essential so they can be delivered to the correct process.

Question 2

Both UDP and TCP use port numbers to identify the destination entity when delivering a message. Discuss possible reasons for why these protocols invented a new abstract ID (port numbers), instead of using process IDs (which already existed when these protocols were designed?)

Answer:

Here are three reasons.

- First, process IDs are OS-specific. Using process IDs would have made these protocols OS-dependent.

- Second, a single process may establish multiple channels of communications. A single process ID (per process) as the destination identifier cannot be used to distinguish between these channels.

- Third, having processes listen on well known ports is easy, but well-known process IDs are impossible.

Question 3

What is the key difference between TCP Tahoe and TCP Reno?

Answer:

- The key difference, and benefit of Reno, is that Reno avoids Slow start when it can and can do Fast recovery at certain cases.

Question 4

Recall the Leaky Bucket algorithm we saw in class. If a sender has a burst data rate of 20KB/s for 20 seconds as data to send and we have a bucket size of 100KB with a output rate of 10KB/s: Does the Leaky Bucket algorithm achieve its aim of regulating output properly? If so explain how and show your calculations. If not, find the appropriate bucket size and discuss why we need more/less of a bucket size.

Answer: The bucket is too small, we should have had a bucket size of 200KB. The input data is 20x20=400KB, in the same time the bucket can empty only 20x10=200KB and can hold another 100KB. So another 100KB bucket size is needed to deal with 400KB in total.

Question 5

TCP relies on timers for resending in case some ACKs are missing. If we set such timers to a fixed value of say 100ms, discuss what would be the advantages and disadvantages of such a static protocol design.

Answer: A fixed 100ms timer is easy to implement and does not require monitoring network status and load. If conditions are stable and 100ms is set accordingly then all should work fine with little overhead. However, in most cases, network conditions change. Thus 100ms could lead to prematurely timers going off, which means many redundant resends, especially when network is busy... When network conditions indicate fast traffic, then the opposite would be true, i.e., static timers will go off too late for missing segments and applications would wait extra for no good reason. In case of dynamic conditions in the network, timers should be set based on measurements rather than to a static value.