

Solutions to Homework#4

2. (a) In the following, the client receives file “foo” when it thinks it has requested “bar”.
1. The client sends a request for file “foo”, and immediately aborts locally. The request, however, arrives at the server.
 2. The client sends a new request, for file “bar”. It is lost.
 3. The server responds with first data packet of “foo”, answering the only request it has actually seen.
- (b) Requiring the client to use a new port number for each separate request would solve the problem. To do this, however, the client would have to trust the underlying operating system to assign a new port number each time a new socket was opened. Having the client attach a timestamp or random number to the file request, to be echoed back in each data packet from the server, would be another approach fully under the application’s control.
9. (a) The advertised window should be large enough to keep the pipe full; delay (RTT) \times bandwidth here is $100 \text{ ms} \times 100 \text{ Mbps} = 10 \text{ Mb} = 1.25 \text{ MB}$ of data. This requires 21 bits ($2^{21} = 2,097,152$) for the **AdvertisedWindow** field. The sequence number field must not wrap around in the maximum segment lifetime. In 60 seconds, 750 MB can be transmitted. 30 bits allows a sequence space of 1024 MB, and so will not wrap in 60 seconds. (If the maximum segment lifetime were not an issue, the sequence number field would still need to be large enough to support twice the maximum window size; see “Finite Sequence Numbers and Sliding Window” in Section 2.5.)
12. (a) This is 125MB/sec; the sequence numbers wrap around when we send $2^{32} \text{ B} = 4 \text{ GB}$. This would take $4\text{GB}/(125\text{MB}/\text{sec}) = 32$ seconds.
- (b) Incrementing every 32 ms, it would take about $32 \times 4 \times 10^9$ ms, or about four years, for the timestamp field to wrap.
39. Incrementing the Ack number for a FIN is essential, so that the sender of the FIN can determine that the FIN was received and not just the preceding data.
- For a SYN, any ACK of subsequent data would increment the acknowledgement number, and any such ACK would implicitly acknowledge the SYN as well (data cannot be ACKed until the connection is established). Thus, the incrementing of the sequence number here is a matter of convention and consistency rather than design necessity.

- 46.
- (a) If a client has only sent the request once, and has received a reply, and if the underlying network never duplicates packets, then the client can be sure its request was only executed once.
 - (b) To ensure at-most-once semantics a server would have to buffer a reply with a given transaction `XID` until it had received an acknowledgement from the client that the reply had been received properly. This would entail adding such `ACKs` to the protocol, and also adding the appropriate buffering mechanism to the implementation.