Computer Networks (Sheet #4)

Marius Gavrilescu

1. Sequence numbers need to be exact. If we have x bits of sequence number, that means $8 \cdot 2^x \geq msl \cdot 100Mbps = 6000 \cdot 10^6$. So we need at least 30 bits of sequence number.

   Window size can be approximate. We want to be able to represent window sizes up to $100Mbps \cdot rtt = 10Mb = 1250000$ bytes. That requires a minimum of 21 bits if we are to represent each window size exactly, but we can use fewer bits if we scale the window by a constant factor (for example, we can use 16 bits and always shift the window left by 5).

2. We observe slow start on startup at the beginning (0.2s-0.6s). Slow start after timeout is also visible at around 5.6s-5.7s.
   Linear-increase congestion avoidance is used between 2s-5.5s and after 5.7s.

   Between 0.5s and 3s we see slow start at startup, then some packages are lost, so when transmission is resumed at 2s the congestion window has been halved several times. We use fast recovery and therefore skip straight to congestion avoidance instead of doing slow start.

3. This trace shows the TCP implementation is using a congestion avoidance algorithm without fast recovery (e.g. TCP Tahoe). This means that when three duplicate ACKs are received we reset the congestion window to its initial value and use slow start, instead of trying to skip slow start and going straight to congestion avoidance.

4. `noggin_` sends a SYN packet with source 192.168.1.7:49157 and destination 129.67.151.1:80 to 192.168.1.1.
   The switch picks a port (say 12345), remembers the mapping `192.168.1.7:49157 <-> 12345`, rewrites the SYN packet's source address to 92.21.153.82:12345, and sends it on.

   `www.cs.ox.ac.uk` sends a SYN-ACK packet with source 129.67.151.1:80 and destination 92.21.153.82:12345
   The switch sees 12345 is mapped to 192.168.1.7:49157, rewrites the SYN-ACK packet's destination address to 192.168.1.7:49157 and sends it on.

   `noggin_` sends an ACK packet with source 192.168.1.7:49157 and destination 129.67.151.1:80 to 192.168.1.1.
   The switch sees 12345 is mapped to 192.168.1.7:49157, rewrites the ACK packet's source address to 92.21.153.82:12345, and sends it on.

   The connection is now established.

5. The TCP connections are almost guaranteed to break. If the NAT is reset, then its translation table is lost. Therefore a package arriving from the external host to the NAT would be dropped (as it refers to a address:port combination that is not mapped) or sent to the DMZ, whereas a package arriving from the internal host would create a new address:port mapping, most likely different from the one that was in place before (and thus the external host will drop the package as it does not know about that connection and the packet is not a SYN).

6. Yes. At the application level DNS is one example: the A record type represents IPv4 addresses, so a new record type (AAAA) was added to represent IPv6 addresses.

   At the transport level, the checksum field in UDP/IPv4 can be set to 0 (meaning "no checksum") but as IPv6 no longer includes a header checksum UDP/IPv6 must have a checksum.

   None of these are conceptual changes.

7. Without a Content-Type the user-agent would not know what type of content it is receiving, nor would it know the charset used (if the content is text). While type and charset can be ineferred from looking at the content (looking for magic numbers, a Unicode BOM) or looking at the file extension in the URL if there is one this only works in the simplest cases.

8. One advantage is speed: if we don't have to incur the overhead of a new TCP connection (and maybe TLS negotiation) for each object, the user-agent will receive every object faster.

   One disadvantage is resource consumption on the server-side:
   Connections are now long-lived, so the server must be able to handle a significant number of connections being kept open. This can be alleviated by making the server drop a connection after a few seconds of inactivity (letting a browser use it to download some HTML and every object referenced in the HTML, and then closing it to save resources).