**Compare and Contrast TCP congestion and flow control**

According to Marsic (2010), problems arise when a sender sends more segments than a receiver can accept, called congestion. Segments could be dropped and then when resent become out-of-order. Both Flow Control and Congestion Control attempt to mitigate congestion so that segments of data will not be dropped or otherwise rejected. However, there are differences in how each of these work to do this.

Flow Control is a method by which the number of segments passed between sender and receiver are kept in check. Since a TCP receiver allows out-of-order segments and holds them in a buffer, it is prudent to make sure that the buffer does not get overfull (Marsic, 2010). As such TCP receivers will constantly send open buffer space information back to the sender. In this way, the sender knows how many more segments it can send to the receiver without exceeding its buffer. Regulating how many segments get sent helps to prevent congestion.

Congestion Control also seeks to prevent congestion, but it will also attempt to mitigate the damage done once congestion has occurred (Marsic, 2010). With Flow Control keeping the sender from sending more segments than the receiver could accept, one would think that all congestion would be mitigated. This is not the case. Since there are intermediary nodes between the sender and receiver, there is still a possibility that congestion can occur if, for example, a router between them is congested. In such circumstances, the sender would take notice that the receiver is not acknowledging previously segments, which are being dropped in route, and reduce the number of segments it will send accordingly (Marsic, 2010).

**Describe how TCP handles packets received out of sequence**

According to Marsic (2010), TCP receiving nodes will accept segments of data from the sender even if they are out of order. The receiver will hold these out-of-order segments in a buffer. When the receiver has enough segments such that it has filled in the gaps of segments, it will finally send the segments to the layer above the TCP layer. In this way the application layer receives the segments in order. A potential problem with holding out-of order segments in a buffer is that the sender could continue sending segments while the receiver is holding too many in buffer, which would eventually make the system fail. To avoid this, the receiver continually announces the buffer space that is available for more segments to be sent (Marsic, 2010).

**Explain three advantages of IPv6 versus IPv4**

IPv4, the Internet Protocol which has been used since the 1970’s, is becoming obsolete due to the extensive growth of the internet over time (Marsic, 2010). A newer version of the protocol IPv6 has been developed and has been gradually replacing IPv4. Although the migration to the new protocol will take time, it has several advantages:

1. The most critical advantage of IPv6 over IPv4, and the one for which it was developed, is that it can accommodate many more internet addresses than IPv4 (Marsic, 2010). IPv4 has 32-bit addresses, which equates to 4,294,967,296 addresses. As the internet has grown and addresses have been assigned to more and more organizations, the pool of available addresses has been depleted. IPv6 has an address space of 128 bits, which translates to 340,282,366,920,938,463,463,374,607,431,768,211,456 addresses. No doubt, the address space for IPv6 will not be used up any time soon.
2. Compared to IPv4 headers, IPv6 headers are much simpler because of the removal of some features which were rarely used or not needed (Marsic, 2010). Instead, IPv6 has a shorter mandatory header and allows for an optional extension header, which can hold extra information.
3. Since the extension header might be critical in some circumstances, IPv6 also incorporates a Hop-by-Hop Options header. Inclusion of this header forces every node between sender and receiver to see and process the extra header information (Marsic, 2010).

**Explain how BGP facilitates the routing of packets between autonomous systems**

According to Marsic (2010), Border Gateway Protocol, or BGP, is a routing protocol used between different domains, called autonomous systems. Within an autonomous system, routing decisions are made relatively easily because the pathway information for routing tables is easily available due to being broadcast to the network. However, routing tables may not be as easy to generate between autonomous systems because not all routing information may be available outside of the autonomous system, perhaps for security reasons. When communication spans two autonomous systems, the session is called an external BGP session (eBGP). The eBGP sessions are more prone to congestion due to not having full visibility of pathways between sender and receiver. As such the eBGP sessions are more complicated than internal routing sessions (Marsic, 2010).

**References**

Marsic, I. (2010). Computer Networks: Performance and Quality of Service. Retrieved from http://www.ece.rutgers.edu/~marsic/books/QoS/