1. Consider the simplified network configuration pictured below, with three paths from the workstation to the server, a fast path (ABCDEF), a medium path (ALMNF), and a slow path (AXYF).

A. (5 points) Assuming that this network uses a traditional store-and-forward approach with no message segmentation, calculate the total amount of time that it would take to move a 16 megabit message across each route from the workstation to the server.

B. (5 points) Assuming that this network uses a traditional store-and-forward approach and message segmentation (i.e., splitting the 16 megabit message into 8000 2-kilobit packets), calculate the total amount of time that it would take to move the 16 megabit message across each route from the workstation to the server.

C. (10 points) The packet switching approach in part (B), in which all segments of a message are sent via the same path across the network, is considered to be connection-oriented. An alternative connectionless packet switching approach sends the message segments by means of a variety of paths to the destination. Assuming again that the network’s switches are store-and-forward and that the message is split into 2-kilobit packets, calculate the optimal distribution of packets being sent across the three routes.
2. (5 points) When two packets at a switch simultaneously opt for the same outgoing link, there are two common methods for dealing with the problem. The most common policy is the **store-and-forward** approach, in which one packet is buffered at the switch until the other packet is finished with the link. The other approach avoids the expense of buffering by deflecting one of the packets onto another available link, forcing it to use an alternative path while the other packet uses the desired link. Assuming that the switch can determine how close to its destination any packet is, **explain** whether it would be better to deflect the packet that is closest to its destination or the packet that is furthest from its destination.

3. The grid at right displays the comparative delay components associated with the use of lightly and heavily loaded LANs and WANs. These delay components are:
   - \( D_{\text{processing}} \): the delay associated with processing the information within the packet headers (e.g., addressing, error-checking)
   - \( D_{\text{propagation}} \): the delay associated with traversing the links between the source and the destination
   - \( D_{\text{queuing}} \): the delay associated with buffering as packets await being forwarded onto their output line
   - \( D_{\text{transmission}} \): the delay associated with packets being copied from a buffer onto the outgoing transmission line

   **A.** (5 points) **Explain** why the \( D_{\text{processing}} \) is always the shortest delay, regardless of the size of the network or how heavily loaded it is.

   **B.** (5 points) **Explain** \( D_{\text{propagation}} \) is the second shortest delay on local area networks, while \( D_{\text{transmission}} \) is the second shortest delay on wide area networks.

   **C.** (5 points) **Explain** why \( D_{\text{queuing}} \) is the longest delay on heavily loaded networks, but not on lightly loaded networks.

4. When packets are sent through various protocol layers prior to being transmitted, each layer attaches a header to the front of the packet and sometimes a footer to the rear of the packet, containing information that will be useful to the relevant protocol at the network nodes between the source of the message and its destination.

   **A.** (5 points) **Addressing information** (e.g. IP addresses) is always placed in a header. While the destination address is included to ensure appropriate packet delivery, a forged source address can be used to implement “spoofing”, the practice of hiding the true identity of the packet’s source. **Explain** how gateway devices (entry/exit points for specific sub-networks) may be used to detect and inhibit this type of spoofing.

   **B.** (5 points) **Error checking information** (e.g. checksums obtained by performing mathematical operations on the packet payload, used by the intervening nodes to confirm that no bits have been corrupted) is usually placed in a footer, since it can then be calculated for insertion as the payload streams out of a transmitting node or for verification as the payload streams into a receiving node. **Explain** circumstances when the error-checking checksum should be placed in the header instead of the footer.