1. Answers:
   1. A circuit-switched network would be well suited to the application described, because the application involves long sessions with predictable smooth bandwidth requirements. Since the transmission rate is known and not bursty, bandwidth can be reserved for each application session circuit with no significant waste. In addition, we need not worry greatly about the overhead costs of setting up and tearing down a circuit connection, which are amortized over the lengthy duration of a typical application session.
   2. Given such generous link capacities, the network needs no congestion control mechanism. In the worst (most potentially congested) case, all the applications simultaneously transmit over one or more particular network links. However, since each link offers sufficient bandwidth to handle the sum of all of the applications' data rates, no congestion (very little queuing) will occur.
2. Answers:
   1. 5 Mbps / 200 kbps= 25, so this means 20 people is the max number of supports users
   2. Probability that a given user is transmitting = 0.05
3. Answers:
   1. The first end system requires B/R1 to transmit the packet onto the first link; the packet propagates over the first link in d1/s1; the packet switch adds a processing delay of dproc; after receiving the entire packet, the packet switch connecting the first and the second link requires L/R2 to transmit the packet onto the second link; the packet propagates over the second link in d2/s2. Similarly, we can find the delay caused by the second switch and the third link: B/R3, dproc, and d3/s3. Adding these five delays gives

dend-end = B/R1 + B/R2 + B/R3 + d1/s1 + d2/s2 + d3/s3+ dproc+ dproc

* 1. Utilizing the equation in the last step, we get:

dend-end = 10 + 10 + 10 + 18 + 24 + 8.

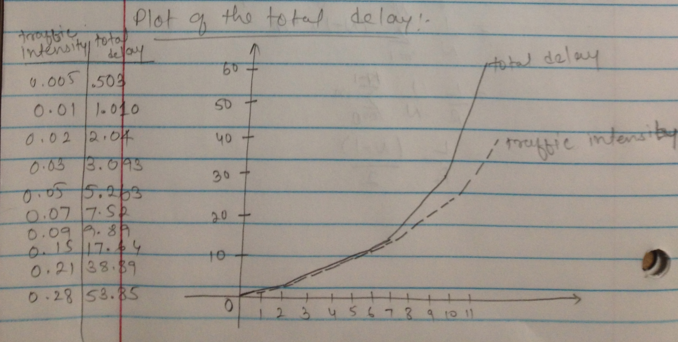
1. Answers:
   1. The arriving packet must first wait for the link to transmit 24,800 bytes or 198,400 bits. Since these bits are transmitted at 10 Mbps, the queuing delay is 19.84 msec. Generally, the queuing delay is (nL + (L - x))/R.
2. Answers:
   1. If only one path : min (Rk1, Rk2, … , RkN)
   2. If all M paths : min (Rk1, Rk2, … , RkN) / M as the bottleneck is shared by all links.
3. Answers:
   1. Total delay = [QL/R(1 – Q)] + L/Q

= L / R [Q + 1 – Q / 1 – Q ]

= L / R [ 1 / 1 – Q ] seconds

* 1. Assume that x is transmission delay = L/R

Traffic intensity = Q = aL/R = xa

Total delay = x / ( 1 – xa)

* 1. Answer:

1. Answers:
   1. Time to send message from source host to first packet switch = 15\*10^6 / 8\*10^6 = 1.875 sec. With store-and-forward switching, the total time to move message from source host to destination host = 1.375 sec× 3 hops = 5.625 sec.
   2. Time to send 1st packet from source host to first packet switch = 1.5\*10^3/8\*10^6 = 0.1875 msec. Time at which second packet is received at the first switch = time at which 1 st packet is received at the second switch = 2 × 0.1875 msec = 0.375 msec
   3. Time at which 1st packet is received at the destination host = 0.1875 msec × 3 hops = 0.5625 msec . After this, every 0.1875 msec one packet will be received; thus time at which last (10000 th) packet is received = 0.5625 msec + 9999 \* 0.1875 msec = 1.875375 sec. It can be seen that delay in using message segmentation is significantly less (almost 1/3rd; 5.625/1.875=3).
   4. Packets have to be put in sequence at the destination and message segmentation results in many smaller packets. Since header size is usually the same for all packets regardless of their size, with message segmentation the total amount of header bytes is more.