1) The canonical name of mydream's web server is asi.mydream.com and its IP address is 178.198.41.98. The contents of this web server should be replicated in the new server, which should be up and running at another IP address. We would use redirection of HTTP requests that arrive at the old server to the new server. Then we would update the A record belonging to asi.mydream.com with the new web server's IP address in the authoritative DNS server. After all this, the old server can be taken down after having waited for twice the TTL value in the A record and having inspected its web logs for inactivity.

2) \( R = 12 \text{ Mbps} \)
\[
\text{RTT} = \frac{2 \times 1050}{2 \times 10^5} \text{ sec} = 0.5 \text{ msec}
\]
\[
\text{Utilization} = \frac{N \times \frac{L}{R}}{L + \text{RTT}} = \frac{N}{11.5}
\]
\[
\frac{L}{R} = \text{Segment Transmission time} = \frac{1500 \times 8}{12 \times 10^6} = 1 \text{ msec}
\]
\[
\text{File is } \left[ \frac{18,080}{1460} \right] = 13 \text{ segments}
\]
\[
W > L + R \times \text{RTT} = 138,000 \text{ bits} = 17,250 \text{ bytes}
\]

\( N \) should be 12 for a utilization of 1, hence, maximum throughput.

3) [Diagram showing packet transmission and timeout handling]

14 will be retransmitted after its timeout at 30.5 + 45 = 45.5, thus completing the transfer at 45.5 + 10.5 = 56.5 msec.

4) [Diagram showing packet transmission and timeout handling]

13 will be transmitted at 44 and its transmission will be over at 44 + \( \frac{600 \times 8}{12 \times 10^6} \times 10^3 \) = 44.4 msec, thus completing the transfer at 44.4 + 10.5 = 54.9 msec.
5) The sample round-trip time, SampleRTT, is not used in the computation of estimated RTT, EstimatedRTT, for retransmitted segments. If SampleRTT were recorded for retransmitted segments, it could mislead EstimatedRTT since the sender can receive an ACK for an earlier transmission of the same segment after the segment is retransmitted and consider the ACK to be associated with the retransmission.

6) Since the sender's congestion control algorithm increases CongWin each time it receives a new ACK, CongWin increases much more rapidly in this modified algorithm and thus the client's download speed is significantly higher compared to original TCP.

7) i) 4000, 5500, 7000, 8500
   ii) After 1st ACK: CongWin ← CongWin + MSS = 9000 bytes
       After 2nd ACK: CongWin ← CongWin + \frac{MSS^2}{CongWin} = 9250 bytes
       After 3rd ACK: CongWin doesn't change since it's duplicate ACK
       Win = \min \{CongWin, RcvWin\} = 9250 bytes
       # of unacknowledged bytes = 5500 - 3000 = 2500 bytes
       Sender can send 9250 - 2500 = 6750 bytes

8) i) CongWin = 4 MSS + 4 MSS = 8 MSS
    \text{Initial window size per RTT in Congestion Avoidance phase}
   ii) CongWin = 2^4 MSS since CongWin doubles every RTT in Slow Start phase
        = 16 MSS

9) \frac{T_4}{\text{RTT}_4} \Rightarrow T_4 = 5T_4, T_2 = 4T_4, T_3 = 2T_4

   \begin{align*}
   \sum_{i=1}^{4} T_i &= 12T_4 = 120 \text{ Mbps} \Rightarrow T_4 = 50 \text{ Mbps, } T_2 = 40 \text{ Mbps} \\
   T_3 &= 20 \text{ Mbps, } T_4 = 10 \text{ Mbps}
   \end{align*}