**Question 1 (35 points):**

Consider the sequencer protocol for total ordering shown in class.

a. Does it also preserve causal ordering when there are no failures? Explain.

**Answer:** Yes. Let $m_1$ and $m_2$ be two messages whose send events are ordered by the happens-before relation such that $m_1 \rightarrow m_2$ and assume that there is no message $m_3$ such that $m_1 \rightarrow m_3 \rightarrow m_2$. If $m_1$ and $m_2$ are sent by the same process, then $m_1$ was sent to the sequencer before $m_2$ and due to the FIFO reliable links and the sequencer's protocol, $m_1$ will be received and delivered by all processes in this order. Otherwise, $m_2$ was sent after its sender received and delivered $m_1$. Hence, the sender of $m_2$ also received and delivered $m_2$ after $m_1$, and by total ordering all processes did the same. By induction (or negation), we can show this to be true for all messages.

b. List two performance benefits of using the sequencer protocol over the vector timestamp causal ordering protocol shown in class.

**Answer:** The first benefit is that using the sequencer protocol there is no need to maintain and attach to each message a vector of size $n$. The second benefit is that messages are never buffered waiting for causally prior messages.

There are of course also shortcomings to this approach, but you were asked only about the benefits.

**Question 2 (35 points):**

a. Why do most live streaming protocols avoid building a random dissemination overlay like Araneola?

**Answer:** In live streaming, it is important to ensure that the overlay takes into account the upload bandwidth of nodes and their reliability (it has been determined that the longer a node stays in the system, the longer it is likely to continue staying in the system). In a random overlay like Araneola, it cannot be done. Also, in Araneola there is always message redundancy. Notice that it is possible to embed a tree in an Araneola overlay and do push only on tree edges and pull on the other overlay edges. This alleviates the redundancy problem, but does not help with the bandwidth issue.

b. List 2 benefits of disseminating information by flooding over a random semi-constant overlay like Araneola vs. relying on push based gossip.

**Answer:** First, push based gossip requires $O(\log n)$ gossips by each node whereas in an overlay like Araneola it is only a small constant (as low as 3-4). Second, when the overlay is semi-constant, connections are relatively long-lived meaning that it makes sense to use TCP connection. On the other hand, in push based gossip, the sending target changes in each gossip round and therefore it is not efficient to use TCP. The problem with UDP is that the sender does not know if the message was received or not.
Question 3 (30 points):
Which architecture, BigTable or Dynamo, is likely to return the results of a read that can be satisfied entirely from main memory more quickly on average? Which is likely to return more quickly in the worse case? Explain your answer and list explicitly any assumptions you make in your answer.

Answer:

(a) Average case: Assuming all information is in memory, and assuming a workload that exhibits a large degree of access locality and skewed access distribution, in most accesses the client already knows the tablet server holding the data in BigTable, and therefore these accesses will be completed by contacting a single node.

On the other hand, with Dynamo, when using a load balancer, the access will go to a random node who will forward it to the members of the preferred list and wait for R replies. This is expected to take longer. Yet, when using a client side Dynamo-aware library and setting R=1, Dynamo will also involve a single server access in such cases.

(b) Worst case: In the worst case, in BigTable a client has stale information in its caches. So it first tries the wrong tablet server. Then the wrong meta-tablet server, then the wrong root tablet server, then chubby (or the master who accesses chubby). Then gradually the root tablet server, meta tablet server, and finally the correct tablet server. Also, if one of these servers is unresponsive, there is a need to wait until it either recovers or another server takes over.

On the other hand, in Dynamo the path is always the same and thus shorter in the worst case.