Time: 75 min (strictly enforced)
Points: 50

YOUR NAME:

Be brief, but **DO NOT** omit necessary detail

{Note: Simply copying text directly from the slides or notes will not earn (partial) credit. Brief, clear and consistent explanation will.}
I. Below, you are given a true or false statement and asked a follow up question.

1. [7 pts] **True statement:** Direct connectivity does not scale, while indirect connectivity can potentially scale.

(4 pt) Explain what we mean by direct and indirect connectivity. Why does direct connectivity not scale?

Ans: Direct Connectivity means, point-to-point links between nodes. This entails the network design to be a full mesh thereby putting the requirements on number of links as \( n(n-1)/2 \). Clearly this is of order \( O(N^2) \) and does not scale. On the other hand indirect connectivity implies connecting to other nodes in the network through a bridge, router etc. Indirect connectivity thus does not require the network to be a full mesh, however it does require some forwarding agents at intermediate nodes.

(3 pts) Why and how can indirect connectivity potentially scale?

Since we do not require the network to be full mesh, just by placing some forwarding functionalities in the nodes themselves we can connect to any node in the network.
2. (8 pts) **True statement:** The IP is not just an interface ID like an IEEE 802 LAN (eg: Ethernet) address.

(4 pts) What else does the IP address encode? Why?

IP address is hierarchal, i.e. its address space is divided into two parts, i) network-id and ii) host-id. This allows for hierarchy and hence intelligent forwarding can be implemented at the nodes. This also leads to smaller routing tables because of address aggregation.

(4 pts) Explain why and how the subnet and supernet masks play a role in this encoding.

Because they allow for “compaction” of address space i.e. more efficient use of addresses. This allows faster routing table lookup (longest prefix match) thus also increasing the efficiency.
II. [12 pts] **Statistical Multiplexing, MAC protocols**

(4 pts) Why is “statistical” multiplexing different from simple multiplexing, and a system that does not allow multiplexing? What are the tradeoffs made by statistical multiplexing?

Statistical multiplexing allows the total multiplexed capacity to be greater than the actual capacity. It depends on the statistical property of the input, i.e. the average and the peak rates. The service rate in Statistical Multiplexing is generally set as

\[
\text{average rate} \leq \text{service rate} \leq \text{peak rate}.
\]

Tradeoffs: buffering, queuing delays and losses. Also the system becomes unstable (i.e. infinite delays and unbounded queues) if service rate < average rate.

Simple multiplexing allows multiplexing up to the “actual capacity”.

(4 pts) Of the three types of MAC protocols (Channel partitioning, Random access and “Taking turns” protocols), which classes allow statistical multiplexing? Why or why not?

**Taking Turns and Random Access allow statistical multiplexing.**

**Taking turns:** Is similar to Round-robin scheduling hence it allows statistical multiplexing. If a user has data to send then he will get a turn. Thus it can be understood as a node getting more slots if it has more data to send and no slots if it has no data to send.

**Random Access:** Users grab the channel whenever they have data to send. This implies that if a user has data to send he can utilize the whole channel, given the channel is not in use.
Channel Partitioning: Each user is allocated a “fixed length” slot in each round. This implies that if a user has more data to send then he has to wait for the next round even if there is an empty slot going (because some other user didn’t have anything to transfer). Thus we are doing a simple multiplexing here.

(4 pts) Why would the statistical multiplexing efficiency offered by these MAC protocols diminish with either increasing in number of hosts or distance spanned.

Random Access: Number of collisions increase as the number of hosts increases. Also the collision detection will fail if the length of the medium is beyond a certain threshold. (eg. 500m in Ethernet).

Taking Turns: If the number of hosts in the network increase there is an overhead in “polling” and latency.
III. [12 pts] ROUTING: In the network below:

a) (4 pts) By examination, find the shortest paths (and their costs) from A to every other node. For example: (A-E, 1) is answer for node E.

b) (6 pts) The initial distance vector (DV) for A is [0 9 INFINITY INFINITY INFINITY 1]. What are the values of the DV after every iteration of the Bellman Ford algorithm till it does not change anymore (I.e. the algorithm terminates). You need NOT write out the DVs of other nodes.

c) (2 pts) In the next page, you are given a forwarding table for A. Fill up the next-hop entries for each node based upon your answers above. A sample entry is filled up for you.

<table>
<thead>
<tr>
<th>Dest</th>
<th>Next Hop</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>E</td>
<td>7</td>
</tr>
<tr>
<td>C</td>
<td>E</td>
<td>6</td>
</tr>
<tr>
<td>D</td>
<td>E</td>
<td>3</td>
</tr>
<tr>
<td>E</td>
<td>E</td>
<td>1</td>
</tr>
</tbody>
</table>
### Forwarding Table at node A

<table>
<thead>
<tr>
<th>Destination</th>
<th>Next-Hop</th>
<th>Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>E</td>
<td>7</td>
</tr>
<tr>
<td>C</td>
<td>E</td>
<td>6</td>
</tr>
<tr>
<td>D</td>
<td>E</td>
<td>3</td>
</tr>
<tr>
<td>E</td>
<td>E</td>
<td>1</td>
</tr>
</tbody>
</table>
IV. [11 pts] End-to-End Principle

a) (3 pts) Speculate BRIEFLY why the end-to-end design principle focuses only on function placement, correctness and completeness, and does not focus on performance issues?

End-to-end principle does deal with correctness and completeness and it did not consider performance. If performance requirements are stringent then we cant do it at end-to-end. (eg. If delay guarantees are needed then we need some kind of network support in form of scheduling). On the other hand, if the requirements are not stringent then the complexity can be pushed to the network edges. Thus the end-to-end principle did not consider the case where functional specification also included performance issues.

b) (4 pts) Explain how the end-to-end principle has led to a diametrically opposite design of the Internet as compared to the telephone system.

The telephone system is circuit switched and homogenous. It placed all the complexity in the network while the end system was dumb. This implied that any incremental upgrade was costly.

The internet is packet switched, heterogeneous. This allows for indirect connectivity. The Internet design entailed putting all the complexity at the end system thus making the end system intelligent. This design feature facilitates an easy and cheap incremental upgrade.
c) (4 pts) Routing is a fairly complex function placed inside the network and not in end-systems. Explain why it is still consistent with end-to-end principle.

Because, Routing is a network layer function and hence had to be inside the network !! End-to-end principle only talks about taking the application layer functionality at the end.