Please write clearly. Make your answers concise, but complete.

1. (12 points) The figure below shows a small switched Ethernet network. Assume that initially, none of the routing tables has any valid entries. Show the contents of the routing tables following the transmission of a packet by the host with address 39 to the host with address f2, followed by the transmission of a packet by host b1 to host 6f and a third packet from host aa to host 39.

If host 39 now starts sending traffic at 1 Mb/s to host b1 and host c3 sends to host f2 at 5 Mb/s, how much traffic will each of the inter-switch links carry in each direction?

A→B traffic: B→A traffic: 
B→C traffic: C→B traffic: 
B→D traffic: D→B traffic:
2. (14 points) The left hand diagram below shows a queue configuration for the most recent version of the wunet router. The first link is configured with a bandwidth of 8 Mb/s and has two queues, with WDRR quanta of 30 and 20, respectively. The second link is configured with a bandwidth of 16 Mb/s and has one queue with quantum of 40. The links’ next-send-times (in μs) and the queues’ current credit values (in bytes) are shown on the diagram, and the current queue is marked by a shaded dot. Each packet is labeled with a letter, and its length (you may assume these are the true packet lengths) are shown in bytes.

![Diagram](image)

Which packet is sent next and at what time is it sent?

Which packet is sent after that and at what time?

Which packet is sent after that and at what time?

In the right hand diagram, show the state of the queues and links after the first three packets have been sent.

If a new packet arrives for link 2 at time 80, when will it be sent?
3. (12 points) The charts below show the average queue length and packet loss probability for the simple queueing model discussed in the lecture notes.

Consider the following scenario for a 1 Gb/s link with a queue that can hold 100 packets.

At time 0, the queue is empty and there is no traffic arriving at the queue.

At time 5 ms, the input traffic jumps to 900 Mb/s and stays at that level for 5 ms.

At 10 ms, the input traffic increases to 1.1 Gb/s and stays at that level for 20 ms.

At 30 ms, the input traffic decreases to 800 Mb/s and then stays at that level.

Assume that the average packet length is 10,000 bits (so it takes an average of 10 μs to send one packet on the link).

Complete the chart below, so that it shows how the queue length varies over time. Keep in mind that it takes time for the queue to fill or empty, and your sketch should reflect this.

Approximately how many packets are discarded during this 50 ms period?
4. (12 points) The chart below shows the results of a performance simulation of $d$-left hashing for different values of $d$. These results are for a hash table that stores 100,000 items in 10,000 buckets.

Suppose we constructed a real hash table for the $d=1$, in which each bucket had room to store 25 items and we inserted 100,000 items into the table, producing the distribution of bucket occupancies shown in the chart. Now, suppose we go to insert one more item. Estimate the probability that the insertion fails because the bucket that the item hashes to is full. Explain your answer.

Give a similar estimate for a hash table with $d=2$ and a bucket size of 12 items. Remember that for $d=2$, we perform two independent hash probes.

Revise your estimate for the $d=2$ case, assuming that the bucket has room to store 13 items, instead of 12.
5. (15 points) The diagram at left below shows the state of a crossbar switch with virtual output queues controlled by the \( i \)-SLIP crossbar scheduling algorithm. The dark circles represent VOQs with one or more cells at the input for the row, going to the output for the column (so input 3 has cells for outputs 0 and 4). The horizontal and vertical lines in the small white squares represent the \( i \)-SLIP pointers. Specifically, each output (column) has a horizontal line for the input corresponding to its pointer value, and each input (row) has a vertical line representing its pointer value. In the left hand figure, circle all the dots corresponding to VOQs that would be selected by the \( i \)-SLIP scheduler in its first iteration in this situation. In the figure at right, show the state of all the pointers after the first iteration, and draw circles for all input-output pairs that would continue to the next iteration.

Now, mark the cells that would be selected in the second iteration in the right-hand diagram above. In the diagram below, show the state of the scheduler after the second iteration completes.
6. (12 points) Consider an overlay router (such as the wunet router) implemented by software running on a conventional processor, with a single 1 Gb/s interface. Assume that the processor can receive and forward a unicast packet in $900+10L$ nanoseconds, where $L$ is the length of the packet in bytes. Assume that the packets range in length from 10 bytes to 1010 bytes.

What is the maximum output packet rate for the router?

What is the maximum output packet rate for maximum length packets?

Suppose that the time required to send multicast packets with a fanout of $F$ is $900+10L+(50+5L)(F-1)$ nanoseconds. What is the maximum output packet rate in this case for packets with a fanout of 11?

For fanout 11 packets, make an estimate for the smallest value of $L$ for which the router can saturate the output interface.
7. (15 points) An instance of the binary-search-on-prefix-lengths data structure is shown below. Show how you would modify the data structure to add the prefixes x5.01 and x5.0101

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>10</td>
<td>01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>101</td>
<td>010/2</td>
<td>110/0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0101</td>
<td>1100</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>xC.1</td>
<td>x1.0</td>
<td>x3.0/0</td>
<td>xB.0/3</td>
<td>xA.0/3</td>
<td>x4.1/2</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>x3.00</td>
<td>xC.10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>xB.011</td>
<td>xA.010</td>
<td>x4.110</td>
<td>xB.001/3</td>
<td>xC.100/6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>xB.0011</td>
<td>xC.1000</td>
<td>xC.1001</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Now, consider a tree bitmap implementation of the original prefix set, with a stride of 3. Complete the internal bitmap and external bitmaps for the root node shown below.

Internal bitmap: _ _ 0 _ _ 1 _ _

External bitmap: _ 1 _ 0 _ _ 1 _

Consider a general version of the tree bitmap data structure for prefixes of up to 64 bits. How many memory accesses would be needed to find the longest prefix in this data structure if the stride length were 4?

How many hash lookups would be needed if we used the binary search on prefix length structure?

Which do you think is the more practical alternative? Justify your answer.