1. Suppose you are required to design an ATM network to support 4096 users, but that the only switches available have just 32 ports. Give a diagram of a network similar to the one on page 3-2 of the notes, with 1:1 concentration. Label your diagram clearly to indicate the number of switches at each level.

What is the cost of the switches in this network, assuming a per port cost of $800?

There are 640 switches with 32 ports each, so the cost is (640)(32)(800)=$16,384,000.

How would you redesign the network if you had 256 port switches instead? How do the costs compare?

With 256 port switches, we would have a two level hierarchy, with 32 switches in the bottom row and 16 in the top row, with each switch in the bottom row having 8 links to each switch in the top row. Since this would give us 48 switches, the cost would be (48)(256)(800)=9,830,400 or 60% of the cost of the original network.

2. Show by induction, that a binary tree with \( n > 2 \) leaves, and in which every internal node has two children, has exactly \( n-2 \) internal nodes.

Actually, if we include the root as an internal node, there are \( n-1 \) internal nodes. We can show this for \( n > 1 \) leaves. Let the basis of the induction be \( n=2 \). There is only one binary tree with \( n=2 \). It has a root and the two leaves. Clearly, this tree satisfies the statement.

Now, consider any tree with \( n > 2 \) leaves. Every such tree has at least one internal node with two children, both of which are leaves. If we remove these two leaves, we get a tree with \( n-1 \) leaves (since the former internal node is now a leaf), and which by the induction hypothesis has \( n-2 \) internal nodes. Since the pruning operation also reduced the number of internal nodes by 1, the original tree must have had \( n-1 \) internal nodes.
Suppose we have a switch that uses the cell recycling technique with binary copying for multicast and that there are three connections set up, one with fanout 1 and a bandwidth of 50 Mb/s, one with fanout 2 and bandwidth 100 Mb/s and one with fanout 6 and bandwidth 60 Mb/s. How much bandwidth is used on all the input ports? How many on all the outputs? How many on the recycling data paths?

The bandwidth used on the input ports is 50+100+60=210 Mb/s. The bandwidth used on the output ports is 50+(100)(2)+(60)(6)=610 Mb/s. Since only the last connection requires the use of any recycling ports and since it only uses four, the recycling bandwidth is (60)(4)=240 Mb/s.

3. Consider a 64 port version of the WU gigabit switch in which 8 ports are connected to 2.4 Gb/s links, 48 ports are connected to 1.2 Gb/s links and 8 are connected to 600 Mb/s links. Any unused bandwidth at a port is available for use by the recycling traffic. What fraction of the outgoing traffic can be multicast if the maximum virtual circuit rate is 50 Mb/s? 150 Mb/s? 600 Mb/s? 1.2 Gb/s? Assume that the maximum load allowed on a switch port (between the switch elements and an IPP or OPP) is 2.4 Gb/s.

The recycling data paths on the eight ports connecting to 2.4 Gb/s external links can't be used at all. For the 48 ports with 1.2 Gb/s external links, we have 1.2 Gb/s of bandwidth per port and for the last 8 we have 1.8 Gb/s per port. We can always add a new connection so long as there is at least one of these ports with enough unused bandwidth to handle a connection of bandwidth B. When B=50 Mb/s, we can do this so long as the total load on the recycling data path is at most (48)(1.15)+8(1.75)=69.2 Gb/s. Since the total output bandwidth is 81.6 Gb/s, we can have up to 85% of the output traffic associated with multicast connections.

For B=150, the available recycling bandwidth is 63.6 Gb/s so we can have up to 78% of the traffic belonging to multicast connections.

For B=600, the available recycling bandwidth is 38.4 Gb/s so we can have up to 47% of the traffic belonging to multicast connections.

For B=1200, the available recycling bandwidth is 4.8 Gb/s so we can have up to 6% of the traffic belonging to multicast connections.

4. Consider a switch that uses cell recycling for copying, together with a switching network that can produce three distinct copies in one pass, instead of two. How would you change the cell format on page 3-23 to accommodate this?

First, we would need to add a third VXI field and BDI field. I would do this by adding a word to the main part of the cell (the 32 bit part) that is identical to the current second and third words, giving fields labeled VXI1, VXI2 and VXI3, along with BDI1, 2, 3. Next we would need to add a third bit to the CYC field. We also need to extend the ADR field to contain three interleaved addresses instead of two. Since the cell is now one word longer
than before, we have room for 11 nibbles in the ADR field, but since we have to fit three
addresses into this space, we can only really use nine of them and can only have switches
with up to 512 ports. Reducing the ADR field to nine nibbles allows us to expand the
timestamp field to 20 bits, instead of the current 12 bits. The last change needed is to the
RC field. Since, as a cell moves through the network, each of the three copies may peel off at
a different point and go in a different direction, the RC field must have seven distinct
values to identify all the distinct possibilities. We can actually reduce this to six, if we
require that the software always sort the three address values in increasing order. This will
leave two codes available, one to specify a specific path cell and the other to specify a range-
copy using the first of the two address values to define the range.

We could increase the possible switch size to 4096 ports by making the ADR field 12
nibbles long and moving four bits of the time stamp field into four of the currently unused
bits in the first word of the cell. This is a bit ugly, but is probably a worthwhile thing to do.

Describe a general procedure for adding and removing endpoints from multicast
connections in this system?

To add an endpoint, we can either add it at any internal node that currently has fewer than
three children (modifying one table entry) or if there is no such internal node, using the
same procedure as in the notes: that is, pick a shallow leaf and replace it by a new recycling
node, making the new endpoint and the selected leaf, children of that new recycling node.
Removing an endpoint is a bit more involved. If the parent of the endpoint has two
children, we could proceed as with the algorithm in the notes. If the parent has three, then
we could just drop the endpoint that is to be removed and leave it at that, or we could
attempt to restructure the tree so that we don’t end up with lots of internal nodes with just
two children. Doing this in a completely general way is possible, but may not be worth the
trouble, since as endpoints get added back in, they will first fill these un-full internal
nodes. Perhaps the best approach is to do this restructuring in a lazy fashion. That is, do it
only when you have to in order to free up bandwidth at recycling ports, or do it when there
is no other useful work to do. The trick is to do this in a way that does not cause cells to get
out of order, as the nodes in the tree change in depth.

How many recycling ports are required for a multicast connection that produces m
copies?

Assuming no rearrangements are done, then in the worst-case, we could end up with trees
in which all the internal nodes have just two children. In this case a multicast connection
that produces m copies would need m-2 recycling ports. If all the internal nodes had
exactly three children (the best case), we would need (m-3)/2 recycling ports.

If h of a system’s n ports are used for recycling and the remaining n-h ports for
input and output traffic only, how many ports are needed to handle a specified
amount of multicast traffic without blocking? (Express your answer as a function
of n, β, B and δ.)

If we want to handle the worst-case, in which all the internal nodes of the tree have two
children, we get exactly the same result as before. If we assume (optimistically) that we can always have trees in which the internal nodes (or almost all of them) have three children, we can get away with about half as many recycling ports. In particular, it’s enough to have $h \geq n(t/2)/((1/\beta)+(t/2)-(B/\beta))$.

What are the advantages of this approach over binary copying? What are the drawbacks?

The big advantage is that it allows a reduction in the number of recycling ports, at least under best case conditions. And arguably, the normal condition may be close to the best case, especially when you consider that many multicast connections will have a small fanout and consequently will use no multicast ports or just one. A second advantage is that it reduces the depth of the tree somewhat (for example, we need 10 passes to produce 10 copies using binary copying, but only 7 with three-way copying).

The drawbacks are that you need to expand the cell format to include the extra information required, which forces an increase in the system bandwidth. In our case, we added one word, which increases the number of clock ticks in a cell cycle from the current 16 to 17. If we want to maintain the same data rate, this implies that we have to increase the system bandwidth by about 6%. In the common case where most of the traffic isn’t multicast in the first place, this cost may be a more serious consideration than the improved efficiency in the use of the recycling bandwidth.

5. The table below shows a simplified version of the virtual circuit tables for an eight port configuration of the WU gigabit switch. In this version, virtual paths are omitted and a pair of the form (x,y) in the table represents a (output port,VCI) pair and a pair of the form (x,y) represents a (recycling port,VCI) pair. Make a list of the different connections represented by the table entries shown. For each point-to-point connection, list the (input port, input VCI) and (output port, output VCI). For each one-to-many connection, give the (input port, input VCI) and the set of all (output port, output VCI) pairs to which copies of cells are forwarded. For each many-to-many connection give the set of all (input port, input VCI) pairs from which cells are received and the set of all (output port, output VCI) pairs to which copies of cells are sent.

<table>
<thead>
<tr>
<th>VCI</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>input 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(3,3),(2,0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(3,1)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td>(5,2)</td>
<td>(3,2),(4,0)</td>
<td>(1,4),(2,8)</td>
<td>(5,1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>(1,7),(2,0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>(7,7),(5,9)</td>
<td></td>
<td>(6,2)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>(2,6),(6,1)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>(6,7),(7,2)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(5,1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>(5,9),(7,1)</td>
<td>(5,1)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Unicast virtual circuits: $(2,3) \rightarrow (5,2)$ and $(4,3) \rightarrow (6,2)$

One-to-many multicast virtual circuits: $(0,9) \rightarrow (1,7), (2,0), (5,9), (7,1)$
(2,5) → (3,2), (7,7), (5,9)

Many-to-many multicast: (1,4), (2,8), (6,7), (7,2) → (1,4), (2,8), (6,7), (7,2)

6.