

## Exercise Set 10: Switching Fabric Topologies

Assigned: Wed. 12 May 2004 (week 10) - Due: Fri. 21 May 2004 (week 11)

### 10.1 Rearrangably Non-Blocking Benes Network

Consider an 8x8 Benes network made of 2x2 switches (i.e. apply both steps of the recursive construction --not just the first step), in an old-style telephony (circuit switching) context. Make the following input-->output connections (a permutation of the numbers 0 through 7):

- 0-->3, 1-->5, 2-->6, 3-->0, 4-->1, 5-->2, 6-->7, 7-->4.

Next, verify that if inputs 0 and 3 wish to mutually exchange destinations (i.e. their previous connections are torn down and we wish to make the new connections 0-->0 and 3-->3), then we also need to modify the routing of several other connections as well.

### 10.2 Clos/Benes made of 4x4 Switching Elements

Draw a 64x64 rearrangeably non-blocking fabric made of 4x4 switching elements. Repeatedly use the Clos network construction, with parameters  $(4, x, 4, x, 4)$ . You will end-up with a Benes-style network (I am not sure if the Benes network is only defined using 2x2 switches, or if it is also generalized for larger switching elements; if the latter is true, then I guess you will end up with a Benes network --not just a Benes-style network). *Reminder:* a Clos network with parameters  $(IN, N1, N2, N3, OUT)$  has  $N1$  swithing elements in its first stage,  $N2$  elements in its second stage,  $N3$  elements in its third stage, and each first-stage swithing element has  $IN$  input ports, while each third-stage swithing element has  $OUT$  output ports.

### 10.3 Non-Blocking Clos Network using Inverse Multiplexing

Show that a packet-switching  $(k, m, k, m, k)$  Clos Network, made out of  $k \times k$  and  $m \times m$  non-blocking switches, is itself non-blocking, assuming that the first- and third-stage switches implement  $k$ -way inverse multiplexing through the  $k$  middle-stage switches. Assume that inverse multiplexing is at the granularity of individual flows, where flows are defined by the pair of input *and* output port numbers. Your proof should be a simple generalization of the proof given in class for the Benes network --which happens to be the case  $k=2$ .

### 10.4 Maximum Degree of Internal Blocking in Banyan Networks

Draw a 16x16 banyan fabric made of 2x2 switching elements. [*Reminder:* an  $N \times N$  banyan network made of 2x2 switches has  $\log_2 N$  stages made of  $N/2$  switches each, and looks like  $N$  balanced binary trees with  $N/2$  leaf-switches (and  $N$  output ports) each, where the  $N$  trees share their leaf nodes as well as a number of other internal nodes, with the amount of sharing progressively decreasing from the leaves to the root].

Following that, find and draw on the fabric a set of telephony-style connections from the  $N=16$  inputs to the  $N=16$  outputs (i.e. a permutation of the numbers 0 through 15) that produces the worst possible amount of internal blocking that you can come up with, i.e. where there is a link through which as many different connections as possible need to pass. You should be able to find such a permutation that requires 4 different connections to go through a single link in the 16x16 banyan. In general, the theory says that this worst possible congestion is equal to the square root of  $N$  for an  $N \times N$  banyan; try to prove this, or give an argument for it.

## 10.5 16-Port Full Fat Tree

Draw a 16-port non-blocking fat tree made of 4x4 switching elements. In order for the fat tree to be non-blocking, each switching element must dedicate as many links to the "upwards" direction (the direction to "remote" ports) as the number of links that it dedicates to the "downwards" direction (the direction to "local" ports) --hence, in our case, two (2) links upwards and two (2) links downwards.

Show a set of telephony-style connections (a permutation) that uses up all of the links up to the third level of switches; then, show another permutation that uses up all of the links across the entire tree.

## 10.6 Cost Comparison of Full Fat Tree versus Benes Network

A "full" fat tree, i.e. a non-blocking fat tree, resembles very much a folded Benes network and has approximately half the number of levels (or stages) when compared to its corresponding Benes network; remote traffic traverses these levels twice --once going upwards and once going downwards-- so remote traffic traverses as many stages as in the Benes network, but local traffic traverses much fewer levels (or switches). Because of this relationship in the number of levels (stages), fat trees give the impression that they offer non-blocking operation at a lower cost, when compared to Benes networks; this impression is **not true**, however, as this exercise shows. The reason is that fat trees use switching elements with **twice** the number of links, as compared to their corresponding Benes network.

Consider the ratio:  $(\text{Number of switching elements in a network}) / (\text{Number of ports of the network})$  as a function of the number of ports of the network, for networks built out of switching elements of a given size  $(2k) \times (2k)$ . Plot this ratio for  $N \times N$  full fat trees and Benes networks, as a function of  $N$ , on a logarithmic horizontal scale; assume that both networks are made of 4x4 switching elements (notice that a fat tree that uses 4x4 elements is a *binary* tree, whereas the Benes network made of 4x4 elements has a *quad* fan-out). The comparison is not straightforward because not all sizes  $N$  produced by one network are possible sizes of the other network too; however, a general trend should be observable once you compute and plot a few points on each curve. If you have enough time, repeat for 6x6 or 8x8 switching elements, and/or compute a general formula that shows the relationship.

Give an intuitive **explanation** why an input port can "see" more output ports, when it goes through a given number of switches, in a Benes (or banyan) network, as compared to the number of output ports that it can "see" when it goes up and down a *variable* number of similar switches --up to the same maximum number of switches-- in a tree.

## 10.7 3/5-Ratio Fat Tree

Use 8x8 switching elements to make a fat tree network where each switching node dedicates 5 of its 8 ports to the "downwards" direction, and the remaining 3 of its ports to the "upwards" direction --hence, this network will have *some* internal blocking. Actually, notice that this network will exhibit no internal blocking as long as the non-local traffic out of or into each subtree does not exceed  $3^h$  times the port throughput, where  $h$  is the height of the subtree.