CS6040: Router Architectures and Algorithms

Jul.-Nov. 2024, Prof. Krishna Sivalingam
Lab 2: Multi-stage Interconnection Network Switching Configuration
Due date: Sep. 13, 2024, 11PM, On Moodle

Aug. 24, 2024

The objective of this assignment is to determine the switching element configurations of an $N \times N$ *Omega*, *Delta* and *Benes* interconnection networks, given a set of ordered inputs.

The command line will specify the following parameters:

```
% ./program2 -in inputfile -sw Omega / Delta / Benes
```

The program will process the inputs and then determine the configurations (straight and cross) of all the 2×2 switching elements in all the stages. The program will then output the configurations as shown in examples later.

The output will be stored in file named **out.txt**. The first line will specify the switching network type (Omega or Delta or Benes).

1 Omega and Delta MINs

For both network types, assume that there is a perfect-shuffle connection preceding the left-most stage.

1.1 Inputs

The inputfile contains:

- The first line specifies N. The second line specifies the number of active inputs, $A \leq N$.

  Assume that N is a power of 2 and $N \leq 32$.

- There are A entries (one integer per line). A_i specifies the desired output port number of the packet at input port i of the Perfect Shuffle switch. Here, $0 \leq i \leq N-1$ and $A_i \in \{0,1,2,\ldots,N-1\}$. The A_i entries are in monotonically increasing order.

  Assume that the Sort-Trap elements are not needed, and that the inputs presented are in sorted order (but with some output port numbers not present). For example, one set of inputs is $\{0,2,4,5,6,7\}$, in an 8×8 switch for $A = 6$. That is, there are no packets destined for output ports 1 and 3.

- Assume that there is no output port contention, i.e. all the requests are made to unique output ports.

An example file is shown below:

```
8
5
1
3
5
6
7
```

1.2 Output

For an 8×8 switch, here is an example output for some input (not related to the input file above).

```
Omega
C T C
T C C
T T T
C T C
```

Here, C denotes cross-connect and T denotes through connect setting of the corresponding 2×2 switch. The first and last column correspond to the left-most (i.e. input) and the right-most (i.e. output) stages of the interconnection network. Note that there is ONE space between the entries on a row and no space at the end of the row (make sure that you print accordingly).

2 Benes MIN

For Benes network, there is NO perfect-shuffle connection preceding the left-most stage.

2.1 Inputs

The inputfile contains:

- The first line specifies N. The second line specifies the number of active inputs, $A \leq N$.

  Assume that N is a power of 2 and $N \in \{4, 8\}$.

- There are A entries (one integer per line). A_i specifies the desired output port number of the packet at input port i of the Perfect Shuffle switch. Here, $0 \leq i \leq N-1$ and $A_i \in \{0, 1, 2, \ldots, N-1\}$. The A_i entries are NOT in any particular order.

  For example, one set of inputs is $\{1, 4, 6, 2, 5, 7\}$, in an 8×8 switch for $A = 6$. That is, there are no packets destined for output ports 0 and 3.

  The file format is as shown above.

- Assume that there is no output port contention, i.e. all the requests are made to unique output ports.

The packets are processed in the order that they appear in the input file. For example, in the above input, the path for the packet from input 0 to output 0 is first determined; then, the path for $1 \rightarrow 4$; then, for $2 \rightarrow 6$, etc.

Let an intermediate switch is configured for a given packet. When another packet has to go through this same switch, it is possible that there will be internal contention. In this case, your algorithm has to choose an alternate route for this packet till there is no internal contention in its path from source to destination port.

You could model this as a capacity-constrained shortest-cost path, where all link costs are initially 1. Whenever a link is used for a packet, the costs of all links on this packet's path are set to 0. Note that you can also choose to implement more sophisticated algorithms.

If a given packet cannot be routed to its destined output port due to internal contention, the packet is treated as dropped.

2.2 Output

The output is as shown below: for each switch, the configuration is printed – one line corresponds to one row in the switch. This is followed by the list of output packets that could not be routed due to internal contention – one destination port per line.

For an 4×4 switch, here is an example output for input of: $\{1,2,3,0\}$.

```
Benes
T T C
T C C
```

Note that this is one possible configuration; multiple configurations are possible. Your program should print at least one correct configuration.

3 What to Submit on IITM Moodle

A single tar.gz file containing:

- Source files, compiled executable
- Example Input files used and Corresponding Output Files generated
- Makefile
- README File that explains how to compile and run the program; whether your programs works correctly or whether there are any known bugs/errors in your program.

4 Grading

- Benes Implementation: 40 points
- Delta Implementation: 25 points
- Omega Implementation: 25 points
- Viva Voce: 10 points

5 Policies

- This is an INDIVIDUAL assignment. Please refer to first day handout (on Moodle) regarding penalties for any form of academic dishonesty, plagiarism, etc. There should be no downloaded code.

  Copying in a programming assignment will result in 0 marks for this assignment and one grade less in this course, for the first violation (during studies at IIT Madras).

  For second-time violation (during studies at IIT Madras), the penalty will be 'U' grade in this course and one grade less for all other courses taken in this semester.

  For further violations, IIT Madras may choose to rusticate the student for a suitable duration of time or cancel the registration.

- Code Plagiarism Checking Software will be used.

- You MUST NOT USE any Generative AI software (ChatGPT, etc.) Such assignments will be reported to IITM Student Disciplinary and Welfare Committee.

- Please start on the assignment right away – you can start working on implementing the input file parsing, etc.

- The program can be written in C/C++/Java/Python.