Feb 6 Lecture Notes

Jiahe Tan,   Sihan Ding
[jt7gu, sd3rf]@virginia.edu

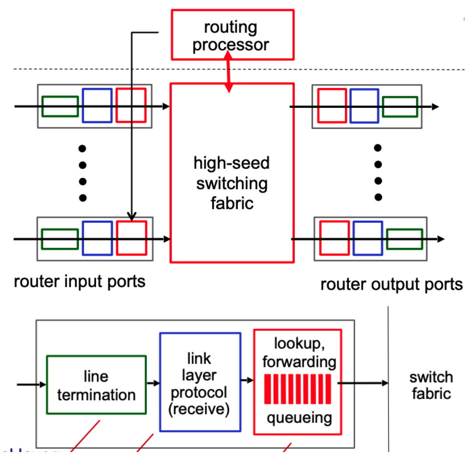


Figure 1. Router Architecture

1. Review Router architecture

A router consists with three components: Input ports, Switching Fabric and Output ports.
The input port, also known as link cards, has three components:

- Line termination: Read in the signal
- Link layer protocol: Error detection (CRC), flow control
- Lookup, forwarding and queuing: Lookup the output port. Temporally store package information if the output port is busy

The switching fabric connects the input port to the correct output port.
The output port sends the package out

2. Destination-based forwarding

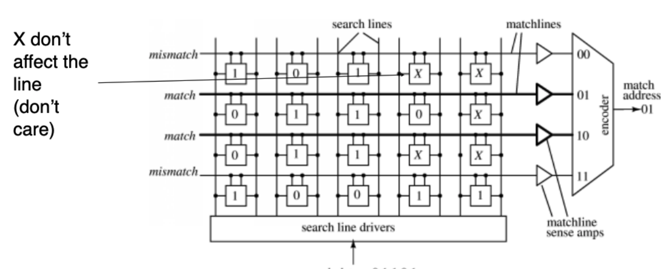
The router forward the package by looking at the forwarding table. The package will be sent to the link interface by using longest address prefix that matches destination address.
Consider the forwarding table shown in Figure 2. If a package has destination IP: 11001000 00010111 00010110 10100001
It will be sent to interface 0 since only interface 0 matches 21 digits of the destination address.
Another package with destination IP: 11001000 00010111 00011000 10101010
It matches 20 digits with interface 0, 24 digits with interface 1 and 21 digits with interface 2. So the package will be forwarded to interface 1.

Destination Address Range	Link interface
11001000 00010111 00010*** *********	0
11001000 00010111 00011000 *********	1
11001000 00010111 00011*** *********	2
otherwise	3

Figure 2. Forwarding table

2.1. TCAM

Ternary content addressable memories is used to retrieve address in one clock cycle. This device stores the forwarding table in the rectangle boxes shown in Figure 3. Each row of rectangle boxes represents an IP address. These IP address are arranges such that the row with less "X" are at the top. The package's destination IP is passed into search line drivers and the driver will pass each digit of the destination IP into search lines. If the digit on a search line does not match with the digit in a rectangle, the rectangle will disable the entire matchline. Then the priority encoder at the right will select the uppermost mamtchline that is not disabled.

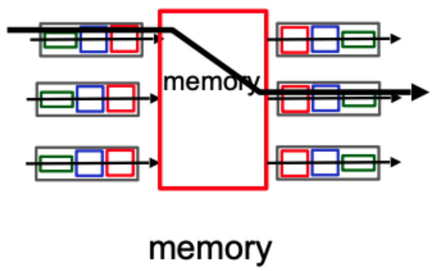


3. Switching fabrics

The switching fabric transfer packet from input buffer to appropriate output buffer.
There are three types of switching fabrics.

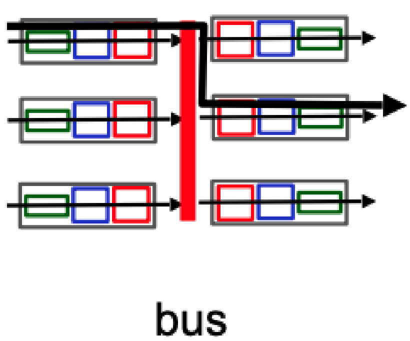
3.1. Memory Based Routers

This kind of switching fabrics is used on switches under direct control of CPU. The packet are copied to system's memory so the speed is limited by memory's bandwidth.



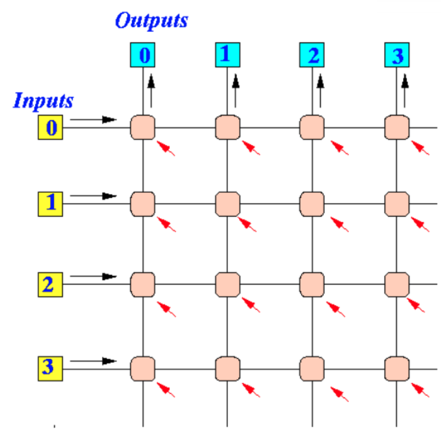
3.2. BUS

All input and output ports are connected to one single bus. The input port broadcast on the bus so only one input and output connection is allowed at a time. The speed is limited by bus bandwidth.



3.3. Interconnection Network

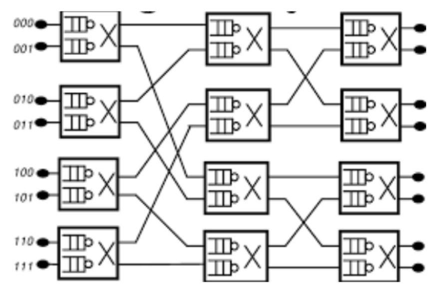
This kind of fabric overcomes bus bandwidth limitation by allowing multiple connections at same time.



The crossbar implementation shown below uses a number of tri-state buffers located at the intersection point of each row and column to control the traffic.

3.4. Banyan Networks

The switches acts like Mux to choose which way the



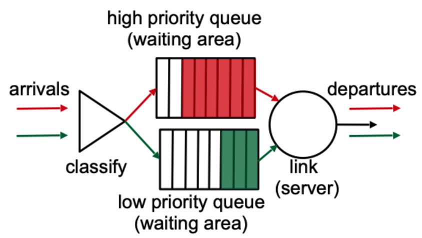
package goes

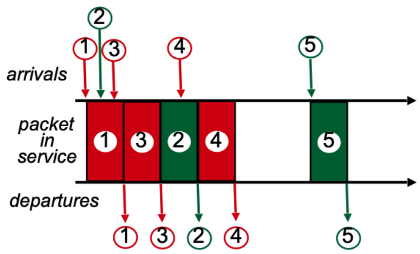
4. Queuing

Input port: Head of line Blocking, a package needs to wait in queue before it gets send out.
Output port: Packages are stored in buffer if output speed is not quick enough. Priority scheduling choose which package gets to send first.

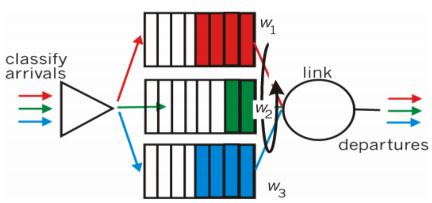
Scheduling policy:

- First in first out: If the queue is full, the buffer needs to discard a package.
  - tail drop: Drop the arriving package
  - priority: Drop the package with least priority
  - random: Randomly drop a package
- Priority scheduling: Having multiple buffers with different priority level and store packages into buffer according to their priority level. Send packages in priority buffer first.





- Round Robin: Distribute packages into different classes and send one package from each class.
- Weighted Fair Queuing: Package with higher priority gets more change to send



5. Distance Vector Algorithm

$D_x(y)$ = Estimate of least cost from x to y .

$C(x, v)$ = Node x knows cost to each neighbor v.

$D_x = [D_x(y) : yN]$ =Node x maintains distance vector.

Node x also maintains its neighbors' distance vectors – For each neighbor v, x maintains $D_v = [D_v(y) : yN]$.

1.A router transmits its distance vector to each of its neighbors in a routing packet.

2.Each router receives and saves the most recently received distance vector from each of its neighbors.

3.A router recalculates its distance vector when it receives a distance vector from a neighbor containing different information than before.

Bellman Ford: Each router maintains a Distance Vector table containing the distance between itself and ALL possible destination nodes. Distances,based on a chosen metric, are computed using information from the neighbors' distance vectors.

From time-to-time, each node sends its own distance vector estimate to neighbors.

When a node x receives new DV estimate from any neighbor v, it saves v's distance vector and it updates its own DV using B-F equation: $D_x(y) = \min C(x, v) + D_v(y), D_x(y)$ for each node $y \in N$