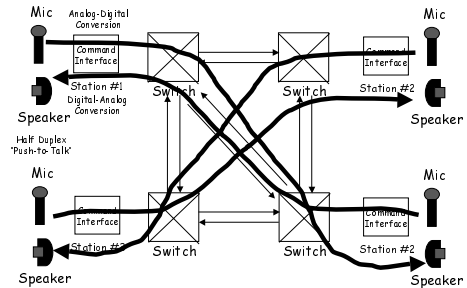


Multimedia Network Switch Project

- Project Concept and Background
- Basic concept of a network router
- Streaming audio application
- Bells and Whistles

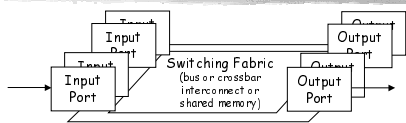
CS 150 - Spring 2004 - Lec. #10 Net Switch Proj -1

Basic Project Concept



CS 150 - Spring 2004 - Lec. #10 Net Switch Proj -2

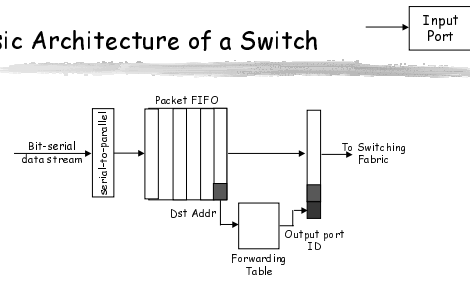
Basic Architecture of a Network Switch



- Input Port
 - (Queue) incoming packet
 - Lookup destination in forwarding table to determine output port
- Output Port
 - (Queue) outgoing packet
- Switching Fabric
 - High bandwidth connection between input and output ports
 - Carry packets from input to output port

CS 150 - Spring 2004 - Lec. #10 Net Switch Proj -3

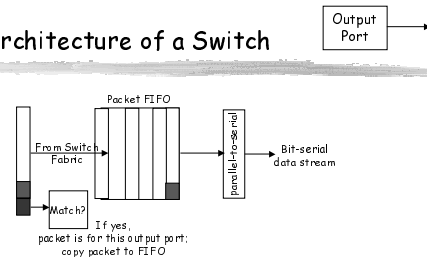
Basic Architecture of a Switch



- Input Port
 - (Queue) incoming packet
 - Lookup destination in forwarding table to determine the output port that is the best to relay the packet to
 - Forwarding table has one entry for each possible destination and contains the relay output port ID

CS 150 - Spring 2004 - Lec. #10 Net Switch Proj -4

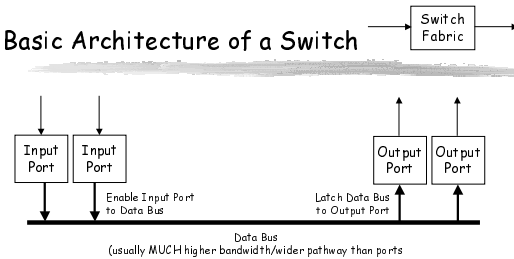
Basic Architecture of a Switch



- Output Port
 - (Queue) outgoing packet

CS 150 - Spring 2004 - Lec. #10 Net Switch Proj -5

Basic Architecture of a Switch



- Switching Fabric
 - Implemented as a bus interconnection
 - Simple, but limited bandwidth
 - One and only one input port connected to (multiple) output port(s)
 - *Time-division multiplexing* with queuing (FIFOs) in ports

CS 150 - Spring 2004 - Lec. #10 Net Switch Proj -6

Basic Architecture of a Switch

Switching Fabric

- Implemented as a crossbar interconnection (also known as *space division multiplexing*)
- $N \times N$ array with connections enabled at each of the crosspoints interconnected by a horizontal and vertical bus (any-to-any connectivity)
- Much higher aggregate bandwidth possible than a single bus system (figure shows the equivalent of two independent buses)
- More than one interconnection simultaneously possible (in figure, two independent connections)

CS 150 - Spring 2004 - Lec. #10 Net Switch Proj - 7

Basic Architecture of a Switch

Switching Fabric

- Implemented as a *packet memory* (yet another way to build time division multiplexing)
- Packet memory organized as a deque, with packets added to the tail and packets removed from the head
- Arbitrated access to memory among input and output ports

CS 150 - Spring 2004 - Lec. #10 Net Switch Proj - 8

Audio Application

- Establish end-to-end connection between speaker and listener in both directions
 - Call set-up sequence: identify destination station through user control panel, establish call path
- For each direction, if push-to-talk
 - Mute incoming audio stream (speaker disabled, mic active)
 - Sample mic input analog audio signal to obtain digital approximation (analog-to-digital conversion)
 - Insert sample into packet, send to destination station
- If listen mode (NOT push-to-talk)
 - Speaker enabled, mic inactive
 - Extract sample from packet, convert from digital to analog, drive speaker output

CS 150 - Spring 2004 - Lec. #10 Net Switch Proj - 9

Block Diagram

CS 150 - Spring 2004 - Lec. #10 Net Switch Proj - 10

Basic Forwarding Table: Statically Configured

Station ID	Output Port
1	0
2	6
3	2
4	4

Forwarding table for Station 1
Note Port 0 is local (i.e., local loopback)

CS 150 - Spring 2004 - Lec. #10 Net Switch Proj - 11

Basic Forwarding Table: Statically Configured

Station ID	Output Port
1	0
2	2
3	2
4	2

Forwarding table for Station 1

Station ID	Output Port
1	2
2	4
3	0
4	4

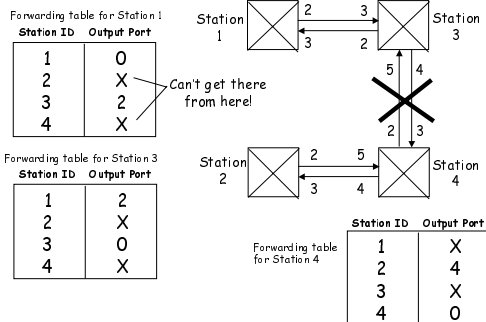
Forwarding table for Station 3

Station ID	Output Port
1	2
2	4
3	2
4	0

Forwarding table for Station 4

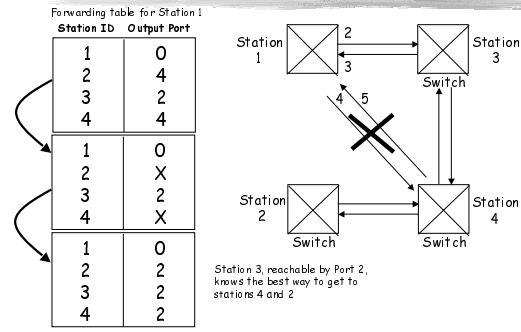
CS 150 - Spring 2004 - Lec. #10 Net Switch Proj - 12

Basic Forwarding Table: Statically Reconfigured



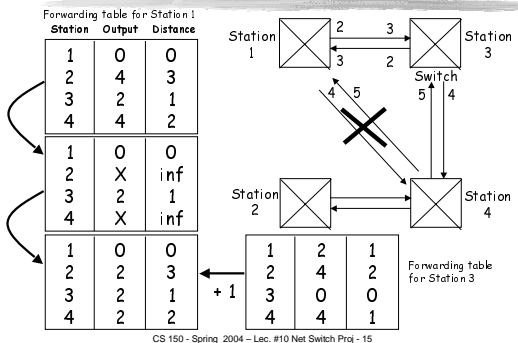
CS 150 - Spring 2004 - Lec. #10 Net Switch Proj - 13

Advanced Forwarding Table: Dynamically Reconfigured



CS 150 - Spring 2004 - Lec. #10 Net Switch Proj - 14

Advanced Forwarding Table: Dynamically Reconfigured



CS 150 - Spring 2004 - Lec. #10 Net Switch Proj - 15

Advanced Forwarding Table

- Keep-alive packets among nearest neighbors
 - Special packet format sent every once in a while to each switches direct (distance 1) neighbors just so they know that you are alive
 - If you have not heard from a direct neighbor in "a while", assume that the direct link has failed; this will cause some forwarding table entries to get "X" and "inf"
 - You should also be able to detect a new neighbor on some direct link by this mechanism
- If link has failed:
 - Poll live direct neighbor for its forwarding table
 - Add one to distances in the received table
 - Try another neighbor if this one doesn't know how to get there

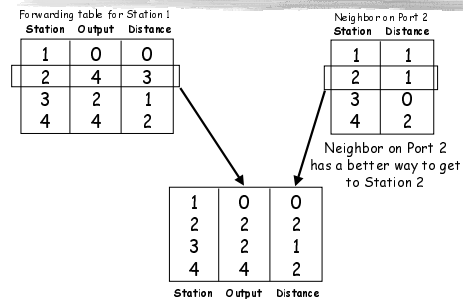
CS 150 - Spring 2004 - Lec. #10 Net Switch Proj - 16

Even More Advanced Forwarding Table

- Build on the mechanism just described BUT ...
 - Don't poll! Piggyback distance information on the keep alive messages
 - For every keep alive message:
 - Compare existing forwarding table to just received distance information
 - Add one to received distance information
 - If neighbor has a BETTER (i.e., shorter!) way to get to a given station, then update the local forwarding table to relay packets for that station through that port
 - But beware of loops!

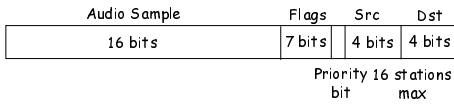
CS 150 - Spring 2004 - Lec. #10 Net Switch Proj - 17

More Advanced Forwarding Table



CS 150 - Spring 2004 - Lec. #10 Net Switch Proj - 18

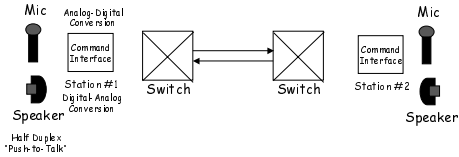
Possible Packet Format



- Usually a good idea to pad packets to be 32 bits or multiples of such words
- Src and Dst fields define 1 of 16 possible stations
- Priority bit useful if mix real time traffic with messaging traffic in same system
- Flags allow special functions to be defined, like keep alive messages or exchange of forwarding tables
- 32 Kbps (2000 samples per second) yields high quality voice

CS 150 - Spring 2004 - Lec. #10 Net Switch Proj - 19

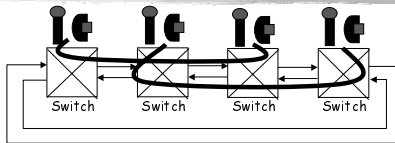
Minimum Project Configuration



- At the very minimum ...
 - Demonstrate point-to-point communications
 - Your design loaded into two connected CalX boards
 - Note: this is NOT an A+ project!

CS 150 - Spring 2004 - Lec. #10 Net Switch Proj - 20

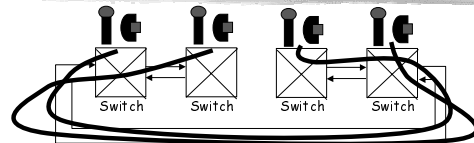
More Interesting Project Configuration



- Ring topology
- Two simultaneous calls to different nodes
- Break the middle links and see what happens

CS 150 - Spring 2004 - Lec. #10 Net Switch Proj - 21

More Interesting Project Configuration



- Dynamic rerouting to reestablish call connectivity
- Especially impressive if this can be accomplished while the call is in progress!

CS 150 - Spring 2004 - Lec. #10 Net Switch Proj - 22

Extreme Project Configuration

- More stations
- More complex interconnection of stations
- Demonstrate multiple simultaneous calls within same network
- Disconnect a station, route around the removed station
- Can you do make this work even in the middle of a call in progress?

CS 150 - Spring 2004 - Lec. #10 Net Switch Proj - 23

Some Possible Bells and Whistles

- Multicast/Broadcast: all stations listen to the same source stream
- Group Communications: form different subgroups of stations to receive same source stream
- Basic specification: static configuration of forwarding table
 - Dynamic reconfiguration: break link and system still works
 - Plug in new station and system still works
 - Rearrange links and system still works
- Implement packet priority and demonstrate that it works ... with non-priority cross traffic
- Other ideas to come ...

CS 150 - Spring 2004 - Lec. #10 Net Switch Proj - 24