

Switch Implementation and Performance

 Simple switch - general purpose workstation with multiple network adapters





Simple Switch (cont.)

- Advantages
 - Simple
 - Inexpensive
 - Flexible
- Disadvantages
 - Limited performance due to single point of contention: I/O bus and main bus each traversed twice for each packet





Simple Switch Performance

- Upper bound on total throughput half min(main memory bandwidth, I/O bus bandwidth)
 - 66 MHz, 32 bit I/O bus allows 66 * 10^6 * 32 / 2 ≈ 1 Gbps
 - This is peak; practical limit is less
 - Could only support one Gigabit Ethernet adapter, or less than 10 Fast Ethernet adapters





Simple Switch Performance (cont.)

- For short packets, time to process packet could also be limiting factor
 - Max packets/sec (pps) * avg. packet length also limits max throughput
- Solution to achieve higher performance custom hardware, able to switch packets between different ports in parallel





Switch Construction



- Ports include network interface, buffering, possibly other logic
- Fabric routes packets to correct output port



Switch Ports

- Components of ports
 - Virtual circuit tables
 - Bridge forwarding tables



- Logic to analyze packet headers to decide how to route packets (input ports)
- Buffers
- Input ports are potential bottleneck need to inspect every received packet to determine where to forward



Buffering Issues

- Problem with buffering in the input ports: head-of-line-blocking
 - Packets destined for available output ports can be held up behind a packet destined for a busy output port
- Buffering in output ports or possibly in fabric (*internal buffering*) preferable
- Buffer management might need to take into account QoS guarantees





Switching Fabric

- Different types of fabrics
 - Shared-bus: basic fabric as described for simple workstation-based switch
 - Shared-memory: input ports write to memory, output ports read from memory.
 - Wide, fast memory required
 - Multi-port memory can increase speed
 - Crossbar: connect any input port to any output port



Switching Fabric (cont.)

- Fabric can either be controlled by input port or control processor, or can access packet header to decide how to route (*self-routing*)
 - Input port can add routing header to start of packet; this is removed when packet delivered to output port





Self-Routing Fabrics

- Self-routing fabric typically constructed of layers of simple 2x2 switches
 - Most scalable approach





Banyan Networks

- Need to ensure that packets on two inputs to the same switch don't need to go to the same output
 - Banyan networks eliminate possibility of collisions
 - Each column of switches looks at next bit of routing header – route to top output if bit is 0, to bottom output if bit is 1





Banyan Networks (cont.)

 Switches wired in "perfect shuffle" pattern so that if packets are presented on input ports in ascending order, they are delivered with no collisions





Banyan Networks (cont.)

 Can use a sorting network (like the Batcher network) to order packets on input to Banyan network

