Packet Switching

Guevara Noubir

Fundamentals of Computer Networks

Textbook: Computer Networks: A Systems Approach,
L. Peterson, B. Davie, Morgan Kaufmann

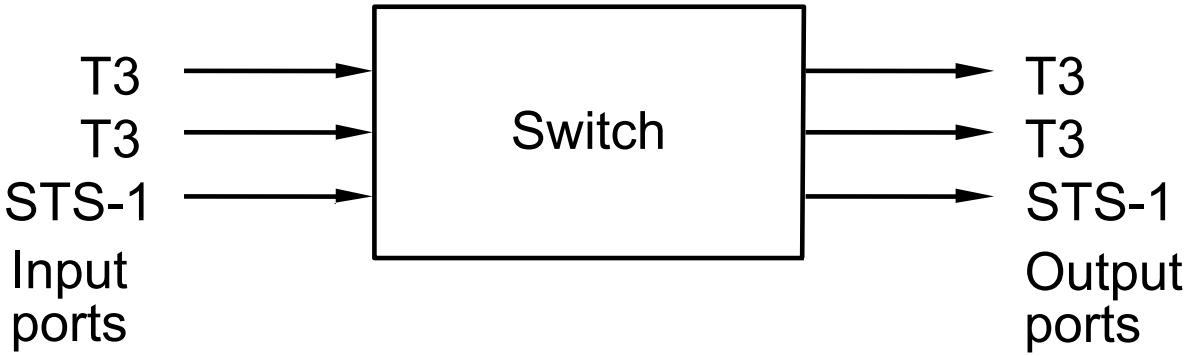
Chapter 3.

Outline

Store-and-Forward Switches
Cell Switching
Segmentation and Reassembly
Bridges and Extended LANs
Switch Design

Scalable Networks

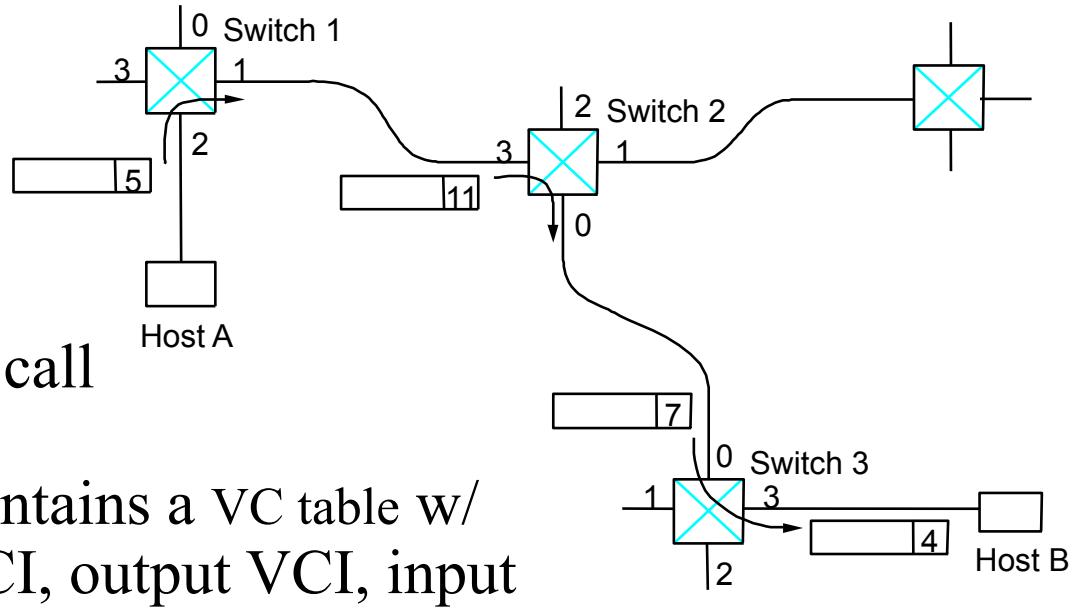
- Switch
  - Forwards packets from input port to output port
  - Port selected based on address in packet header
    - Virtual circuit (connection-oriented) vs. Datagram (connectionless)



- Advantages
  - Cover large geographic area (tolerate latency)
  - Support large numbers of hosts (scalable bandwidth)

Virtual Circuit (VC) Switching

- Explicit connection setup (and tear-down) phase
- Subsequent packets follow same circuit
- Sometimes called *connection-oriented* model



- Analogy: phone call
- Each switch maintains a VC table w/ entries: input VCI, output VCI, input port, output port

Virtual Circuit Switching

- Connection Setup approaches:
  - Permanent Virtual Circuits (PVC): manually setup/removed by network administrators
  - Switched Virtual Circuits (SVC): dynamically setup through signaling over some control channels
- Connection state => VC table
  - Incoming interface, VC Identifier (VCI), outgoing interface, outgoing VCI
- SVC:
  - The setup message is forwarded over the network
  - New entries are created in the VC table and destination switches choose incoming VCI
  - When the setup message reaches the destination, connection acknowledgements and chosen VCI are communicated back to the source

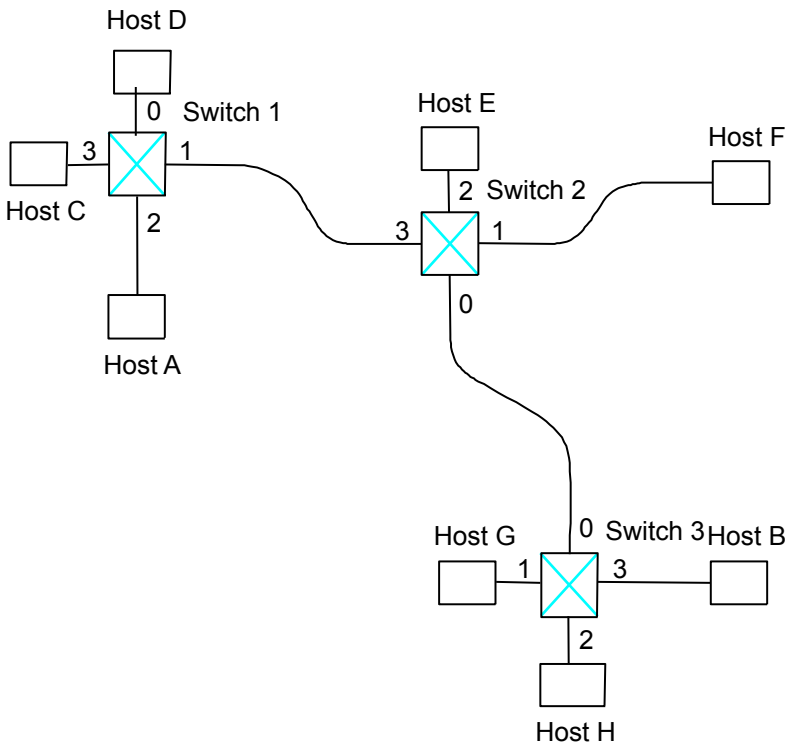
Virtual Circuits

- Examples of Virtual Circuit Technology:
  - Frame Relay, X.25, Asynchronous Transfer Mode (ATM)
- Frame Relay was popular for creating virtual private networks (VPNs) using PVC.
- ATM is a more complex technology that provides mechanisms for supporting quality of service
  - More success in Wide Area Networks, DSL but not on LAN

Datagram Switching

- No connection setup phase
- Each packet forwarded independently
- Sometimes called *connectionless* model

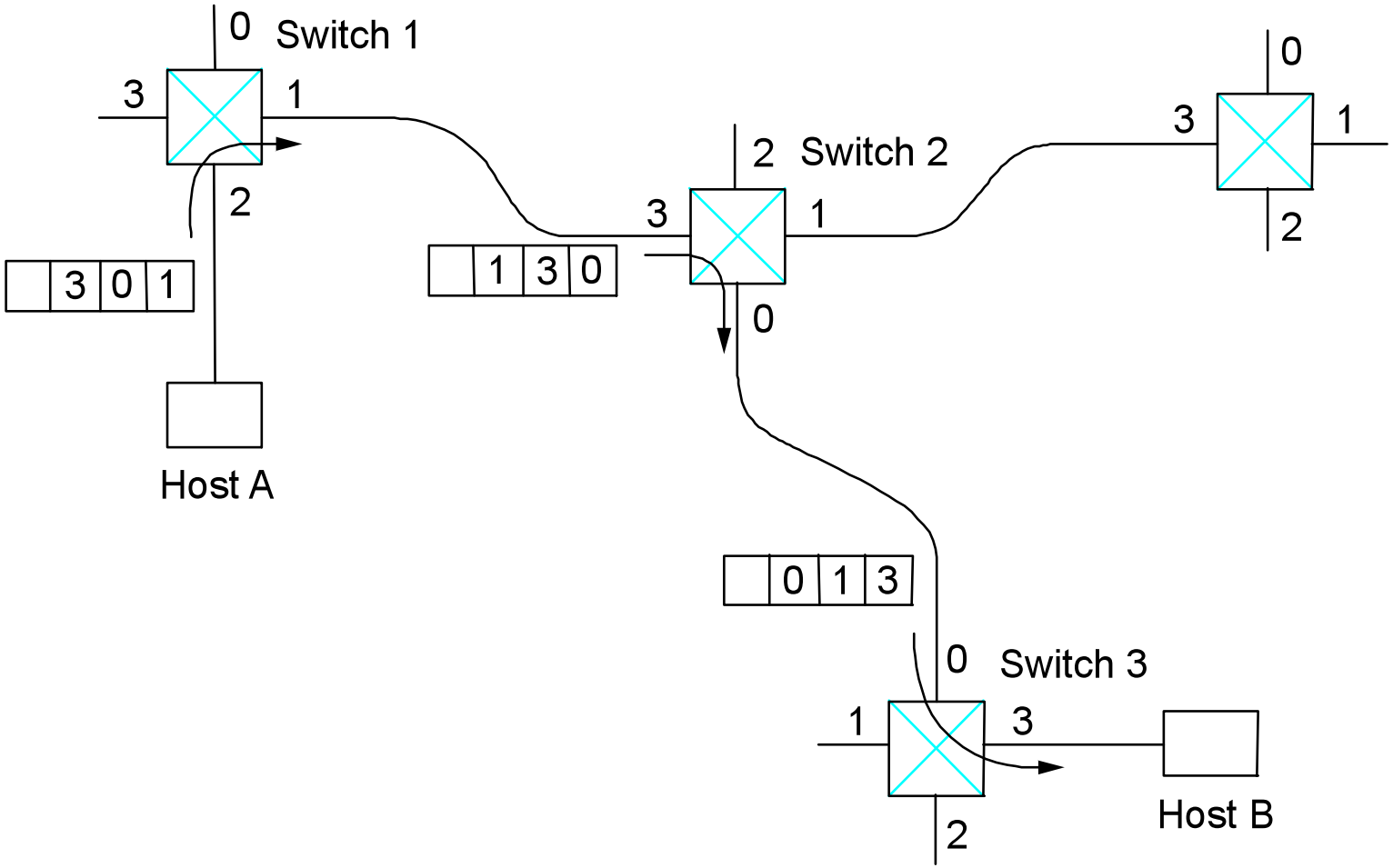
- Analogy: postal system
- Each switch maintains a forwarding (routing) table



Source Routing

- The information to route the packet is provided by the source host and included in the packet
- Example of implementing source routing:
  - Assign a number to each switch output port
  - Include the list of output ports that the packet has to go through
  - The list is rotated by the intermediate switches before forwarding
- Disadvantage:
  - Packet initiators need to have a sufficient information about the network topology
  - The header has a variable length

Source Routing



Virtual Circuit Model

- Typically wait full RTT for connection setup before sending first data packet.
- While the connection request contains the full address for destination, each data packet contains only a small identifier, making the per-packet header overhead small.
- If a switch or a link in a connection fails, the connection is broken and a new one needs to be established.
- Connection setup provides an opportunity to reserve resources.

Datagram Model

- There is no round trip time delay waiting for connection setup; a host can send data as soon as it is ready.
- Source host has no way of knowing if the network is capable of delivering a packet or if the destination host is even up.
- Since packets are treated independently, it is possible to route around link and node failures.
- Successive packets may follow different paths and be received out of order.
- Since every packet must carry the full address of the destination, the overhead per packet is higher than for the connection-oriented model.

Cell Switching (ATM)

- Connection-oriented packet-switched network
- Used in both WAN and LAN settings
- Signaling (connection setup) Protocol: Q.2931
- Specified by ATM forum (www.atmforum.com)
- Packets are called *cells*
  - 5-byte header + 48-byte payload
- Commonly transmitted over SONET (Synchronous Optical NETwork)
  - other physical layers possible: SDH, Wireless, DSL

Variable vs Fixed-Length Packets

- No Optimal Length
  - if small: high header-to-data overhead
  - if large: low utilization for small messages
- Fixed-Length Easier to Switch in Hardware
  - simpler
  - enables parallelism

Big vs Small Packets

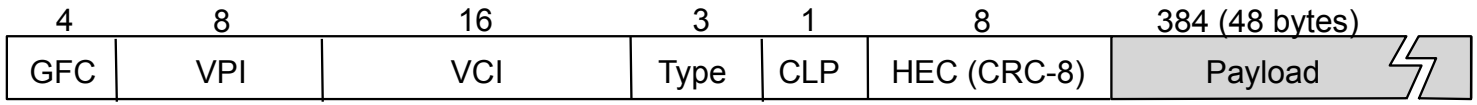
- Small Improves Queue behavior
  - finer-grained pre-emption point for scheduling link
    - maximum packet = 4KB
    - link speed = 100Mbps
    - transmission time = 4096 x 8/100 = 327.68us
    - high priority packet may sit in the queue 327.68us
    - in contrast, 53 x 8/100 = 4.24us for ATM
  - near cut-through behavior
    - two 4KB packets arrive at same time
    - link idle for 327.68us while both arrive
    - at end of 327.68us, still have 8KB to transmit
    - in contrast, can transmit first cell after 4.24us
    - at end of 327.68us, just over 4KB left in queue

Big vs Small (cont)

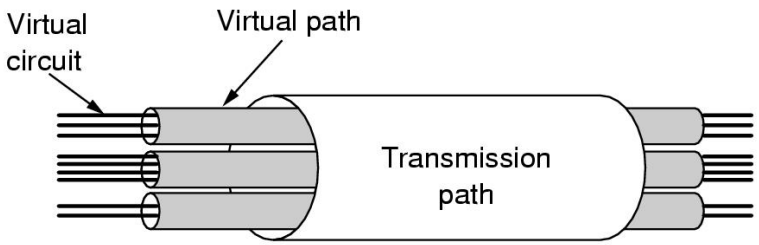
- Small Improves Latency (for voice)
  - voice digitally encoded at 64Kbps (8-bit samples at 8KHz)
  - need full cell's worth of samples before sending cell
  - example: 1000-byte cells implies 125ms per cell (too long)
  - smaller latency implies no need for echo cancellers
- ATM Compromise: 48 bytes = (32+64)/2

Cell Format

- User-Network Interface (UNI)

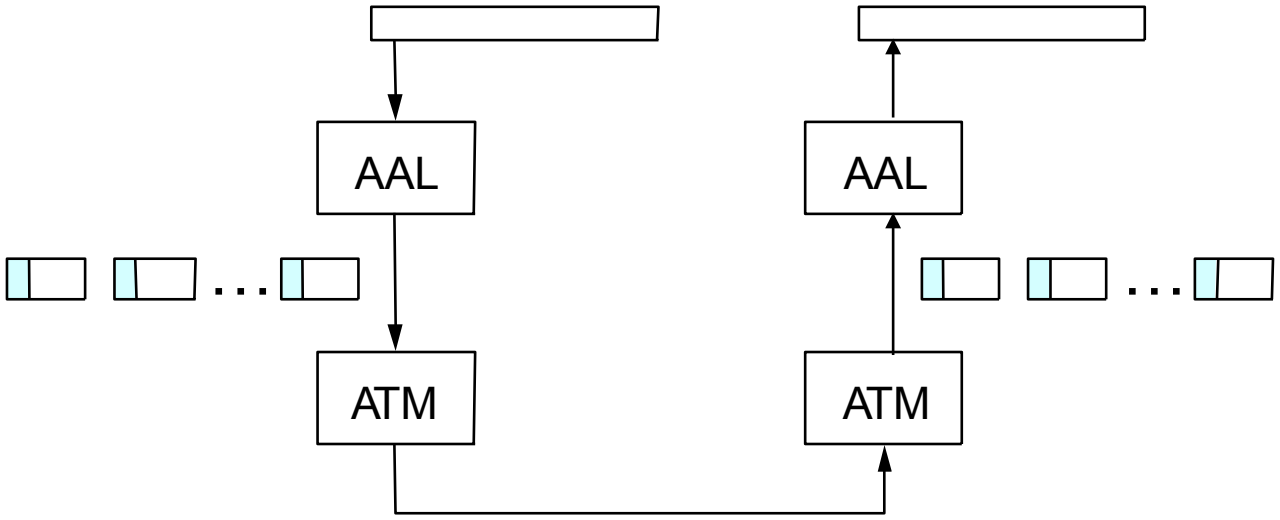


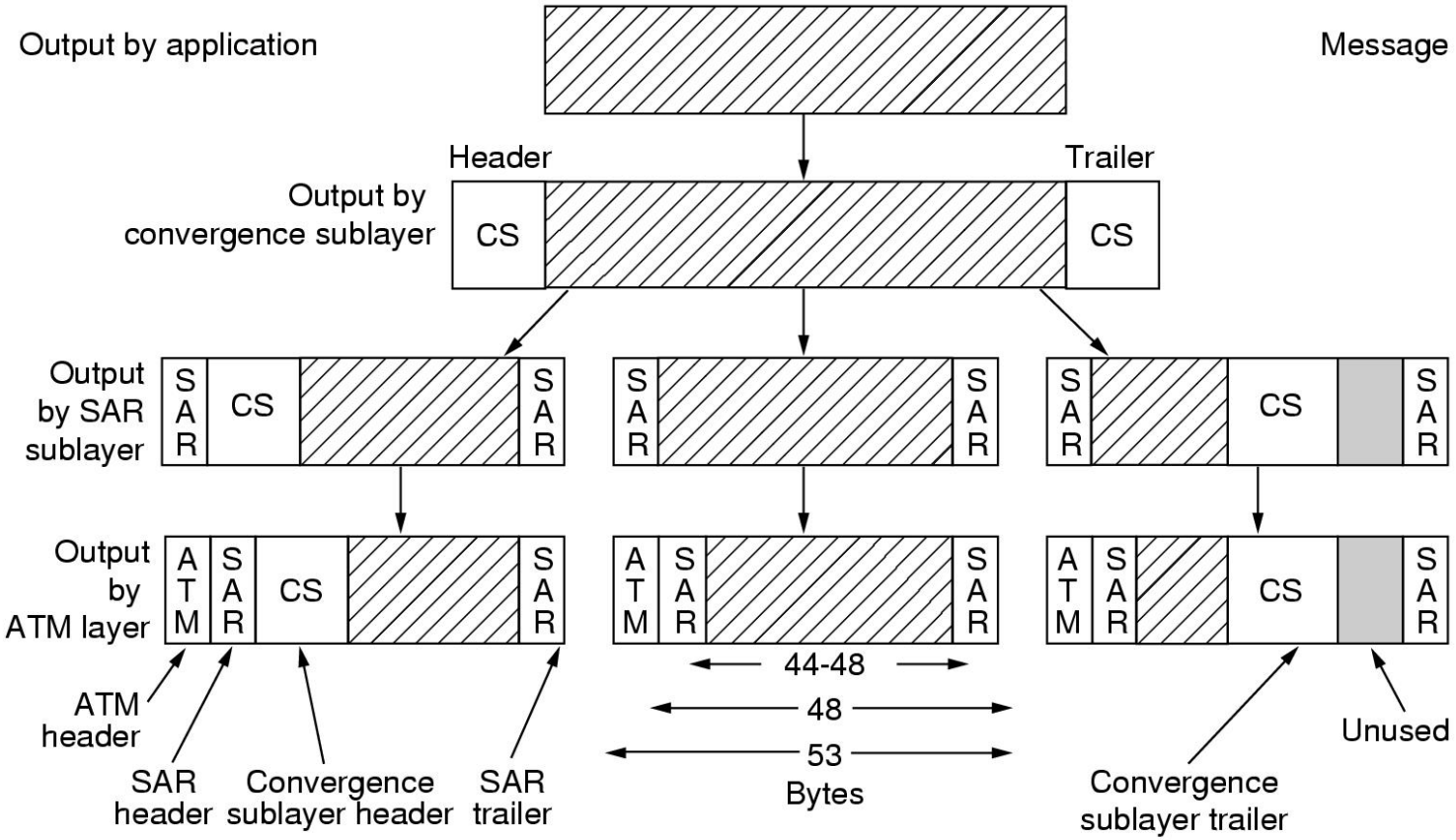
  - host-to-switch format
  - GFC: Generic Flow Control (not used)
  - VCI: Virtual Circuit Identifier
  - VPI: Virtual Path Identifier
  - Type: management, congestion control, AAL5 (later)
  - CLPL Cell Loss Priority
  - HEC: Header Error Check (CRC-8)
- Network-Network Interface (NNI)
  - switch-to-switch format
  - GFC becomes part of VPI field



Segmentation and Reassembly

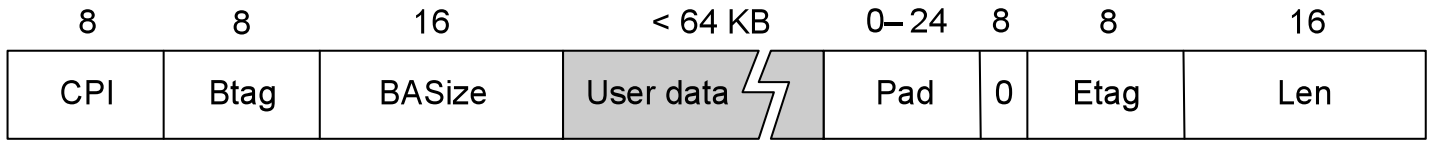
- ATM Adaptation Layer (AAL)
  - AAL 1 (CBR) and 2 (VBR) designed for applications that need guaranteed rate (e.g., voice, video)
  - AAL 3/4 designed for packet data
  - AAL 5 is an alternative standard for packet data. Designed by the computer industry. Most used interface to ATM.




Output by application
Message
Header
Trailer
Output by convergence sublayer
CS
CS
Output by SAR sublayer
S A R
CS
S A R
Output by ATM layer
A T M
S A R
CS
S A R
44-48
48
53
Bytes
ATM header
SAR header
Convergence sublayer header
SAR trailer
Unused
Convergence sublayer trailer

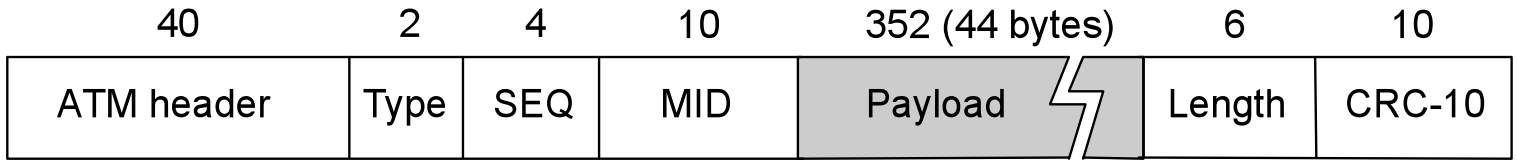
AAL 3/4

- Convergence Sublayer Protocol Data Unit (CS-PDU)



  - CPI: common part indicator (version field: currently 0)
  - Btag/Etag:beginning and ending tag
  - BAsize: hint on amount of buffer space to allocate
  - Length: size of whole PDU

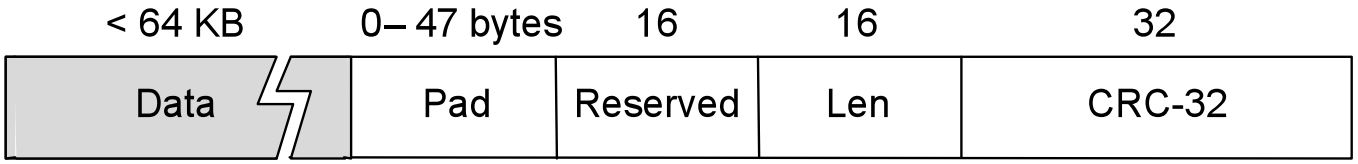
Cell Format



- Type
  - BOM: beginning of message
  - COM: continuation of message
  - EOM end of message
- SEQ: sequence number
- MID: message id
- Length: number of bytes of PDU in this cell

AAL5

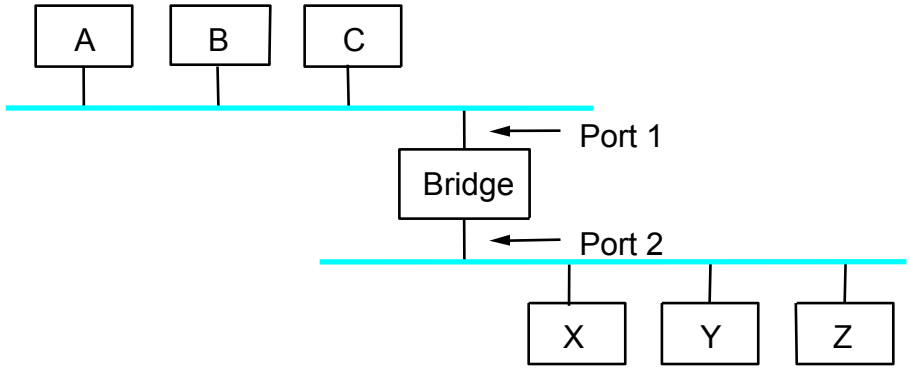
- CS-PDU Format



  - pad so trailer always falls at end of ATM cell
  - Length: size of PDU (data only)
  - CRC-32 (detects missing or misordered cells)
- Cell Format
  - end-of-PDU bit in Type field of ATM header

Bridges and Extended LANs

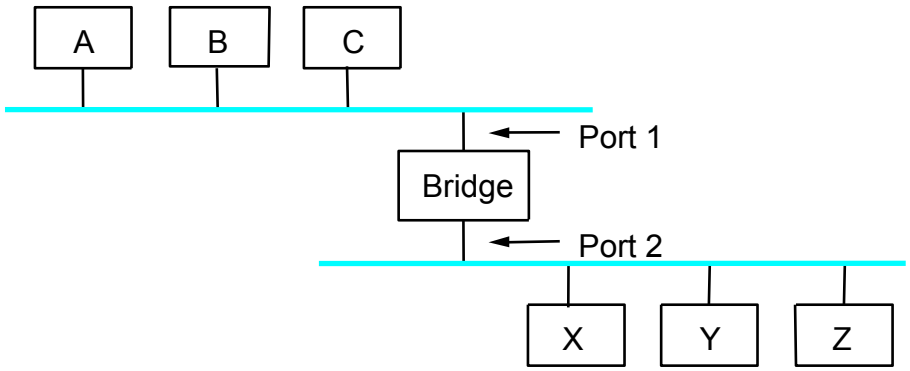
- LANs have physical limitations (e.g., 2500m)
- Connect two or more LANs with a *bridge*
  - Accept and forward strategy
  - Level 2 connection (does not add packet header)



- Ethernet Switch is a LAN Switch = Bridge

Learning Bridges

- Do not forward when unnecessary
- Maintain forwarding table

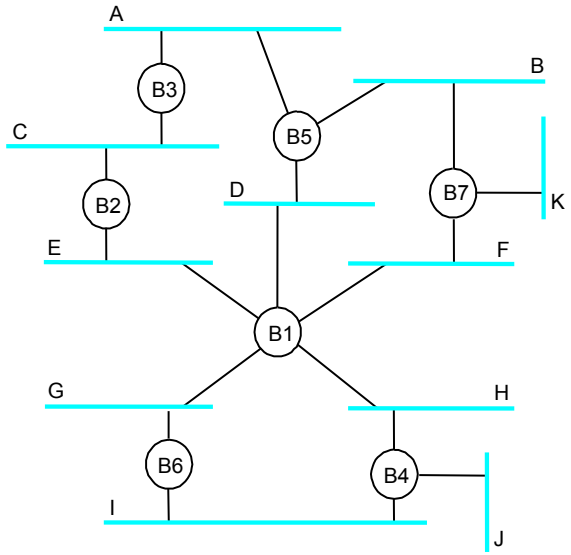


Host	Port
A	1
B	1
C	1
X	2
Y	2
Z	2

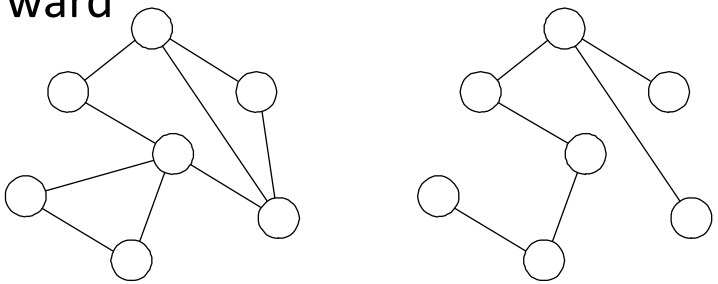
- Learn table entries based on source address
- Table is an optimization; need not be complete
- Always forward broadcast frames

Spanning Tree Algorithm

- Problem: loops

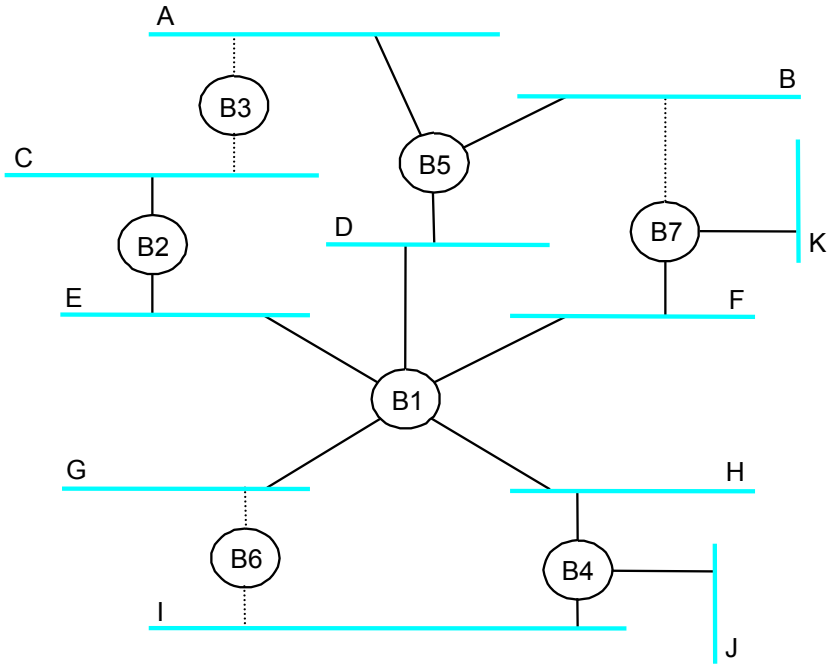


- Bridges run a distributed spanning tree algorithm
  - select which bridges actively forward
  - developed by Radia Perlman
  - now IEEE 802.1 specification



Algorithm Overview

- Each bridge has unique id (e.g., B1, B2, B3)
- Select bridge with smallest id as root
- Select bridge on each LAN closest to root as designated bridge (use id to break ties)
- Each bridge forwards frames over each LAN for which it is the designated bridge



Algorithm Details

- Bridges exchange configuration messages
  - id for bridge sending the message
  - id for what the sending bridge believes to be root bridge
  - distance (hops) from sending bridge to root bridge
- Each bridge records current best configuration message for each port
- Initially, each bridge believes it is the root

Algorithm Detail (cont)

- When learn not root, stop generating config messages
  - in steady state, only root generates configuration messages
- When learn not designated bridge, stop forwarding config messages
  - in steady state, only designated bridges forward config messages
- Root continues to periodically send config messages
- If any bridge does not receive config message after a period of time, it starts generating config messages claiming to be the root

Broadcast and Multicast

- Forward all broadcast/multicast frames
  - On all active ports except the one on which the frame was received
- Learn when no group members downstream
- Accomplished by having each member of group G send a frame to bridge multicast address with G in source field

Limitations of Bridges

- Do not scale
  - Spanning tree algorithm does not scale
  - Broadcast does not scale
- Do not accommodate heterogeneity
- Caution: beware of transparency
  - Bridged LANs do not always behave as single shared medium LAN: they drop packets when congested, higher latency

Virtual LANs (VLAN)

- VLANs are used to:
  - Increase scalability: reduce broadcast messages
  - Provide some basic security by separating LANs
- VLANs have an ID (color).
- Bridges insert the VLAN ID between the ethernet header and its payload
- Packets (unicast and multicast) are only forwarded to VLAN with the same ID as the source VLAN

Design of Switches

- Design goals: Throughput, Scalability, Cost
- Throughput:
  - Is not equal to the sum of speeds of input/output links
  - Depends also on packet size (some operations have to be executed for all packets independently of their size): *packet per second* metric

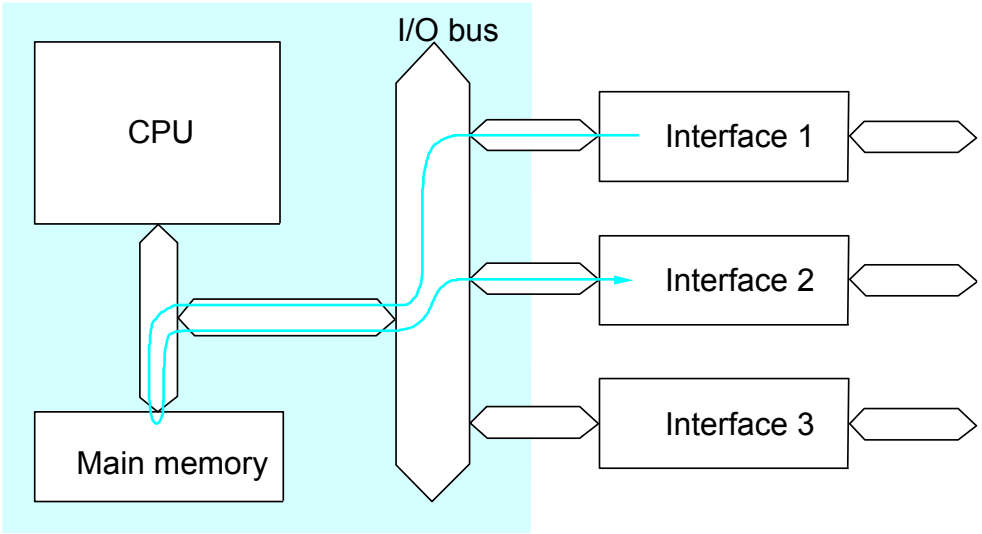
  => Throughput is a function of traffic
- Scalability:
  - How does hardware cost increase as a function of IN/OUT

Ports and Fabrics

- Ports:
  - Functions: Interface with links, buffer packets, maintain tables for VCI (incoming/outgoing VCI)
  - FIFO buffers are not suitable because of *head-of-line* blocking
  - QoS policies have to be embedded in the buffer management (e.g., scheduling, discarding)
- Fabrics:
  - Function: deliver packet to the right output

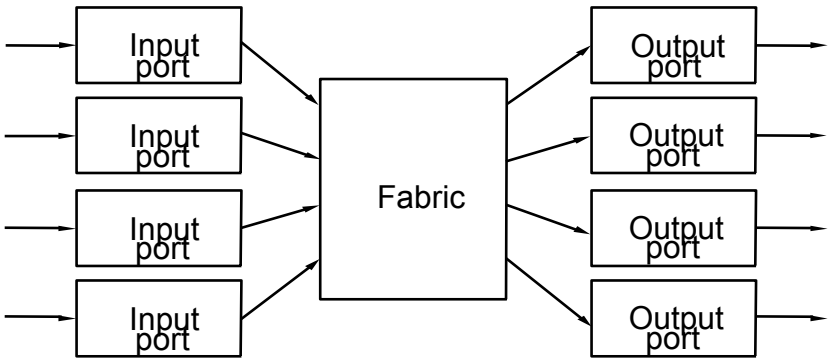
Workstation-Based

- Aggregate bandwidth
  - 1/2 of the I/O bus bandwidth
  - capacity shared among all hosts connected to switch
  - example: 133MHz, 64 bits bus => 8Gbps/2 => few 100MHz ports
- Packets-per-second
  - must be able to switch small packets
  - 1000,000 packets-per-second is achievable for a PC
  - e.g., 64-byte packets implies 512Mbps which is too small for a switch



Switching Hardware

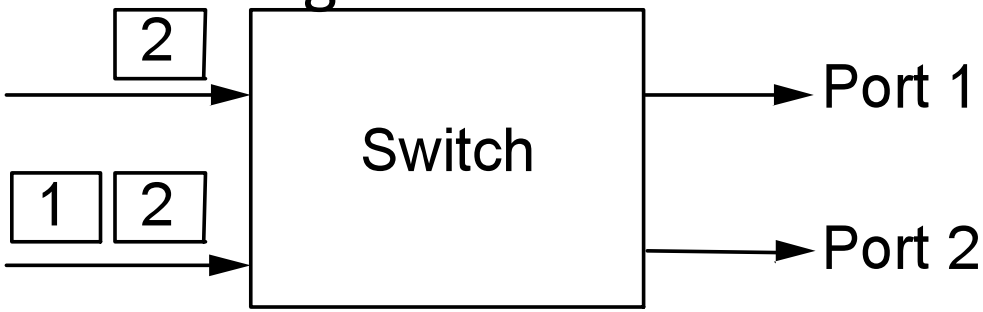
- Design Goals
  - Throughput (depends on traffic model)
  - Scalability (a function of *n*)



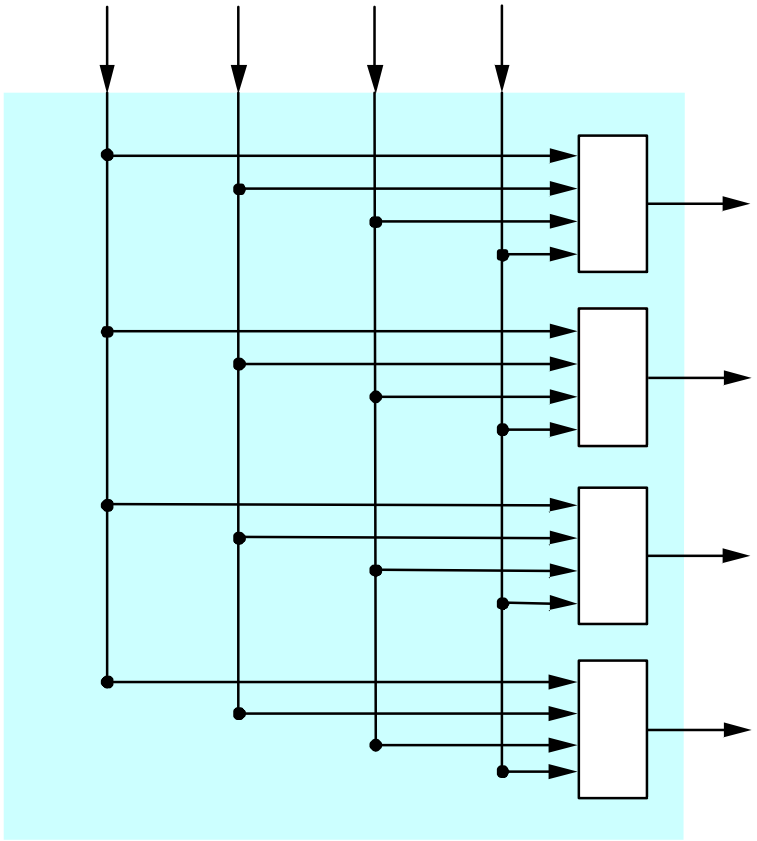
- Ports
  - Circuit management (e.g., map VCIs, route datagrams)
  - Buffering (input and/or output)
- Fabric
  - As simple as possible
  - Sometimes do buffering (internal)

Buffering

- Wherever contention is possible
  - input port (contend for fabric)
  - internal (contend for output port)
  - output port (contend for link)
- Head-of-Line Blocking
  - input buffering: avoid FIFO



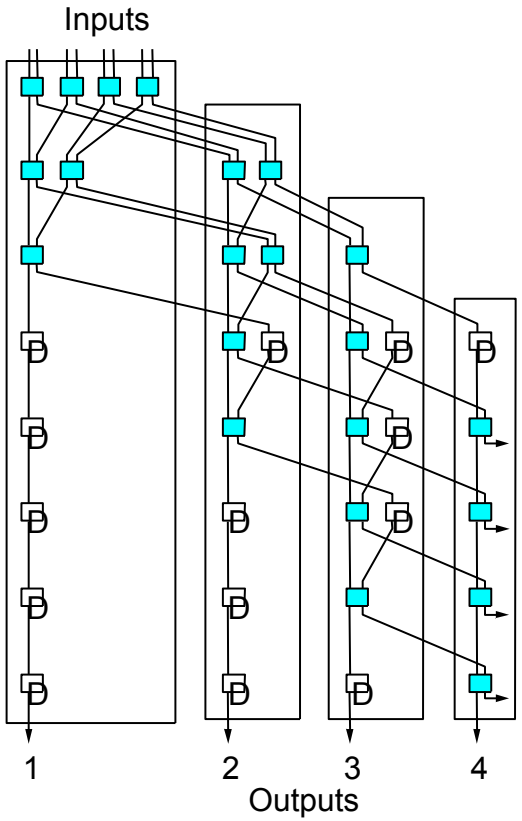
Crossbar Switches



Knockout Switch

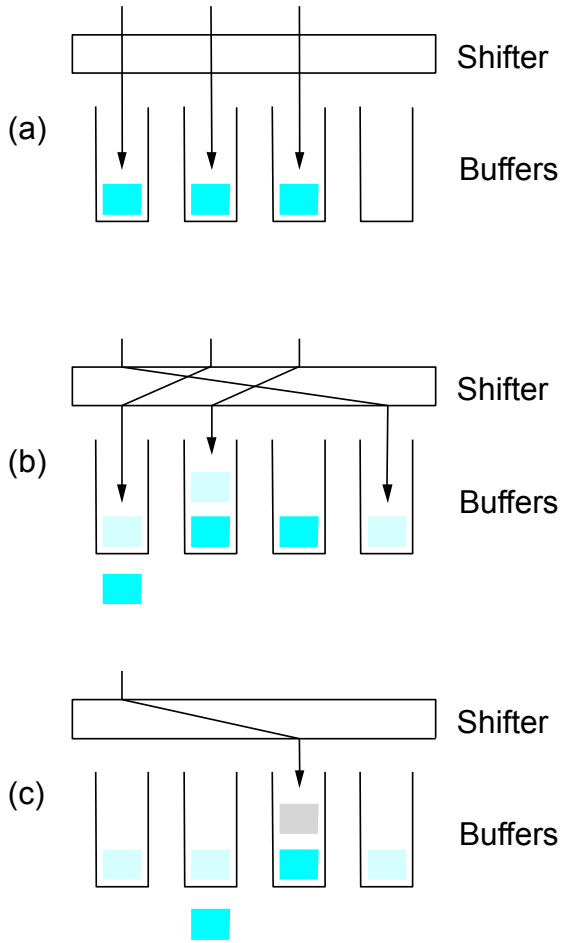
- Example crossbar
- Concentrator
  - select *l* of *n* packets
- 2x2 switches randomly select a winner

- Complexity is sill: n^2



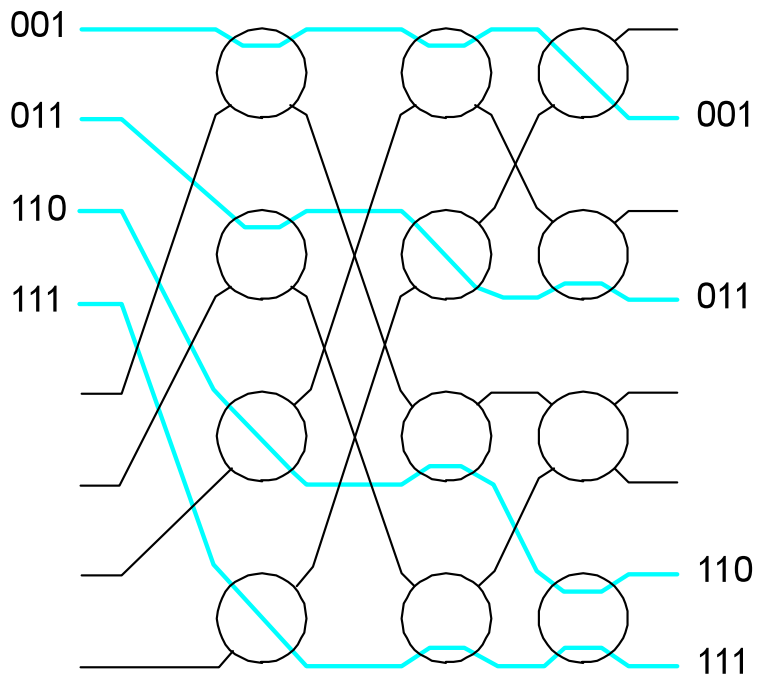
Knockout Switch (cont)

- Output Buffer



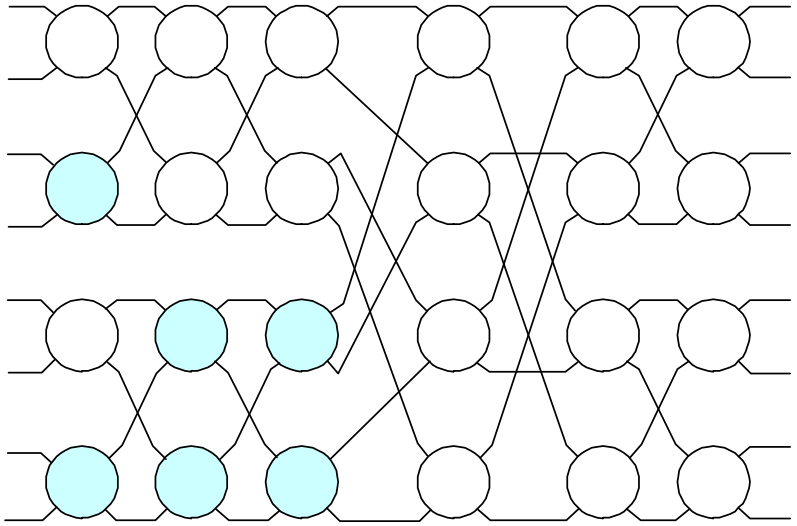
Self-Routing Fabrics

- Banyan Network
  - constructed from simple 2 x 2 switching elements
  - self-routing header attached to each packet
  - elements arranged to route based on this header
  - no collisions if input packets sorted into ascending order
  - complexity: $n \log_2 n$



Self-Routing Fabrics (cont)

- Batcher Network
  - switching elements sort two numbers
    - some elements sort into ascending (clear)
    - some elements sort into descending (shaded)
  - elements arranged to implement merge sort
  - complexity: $n \log^2_2 n$

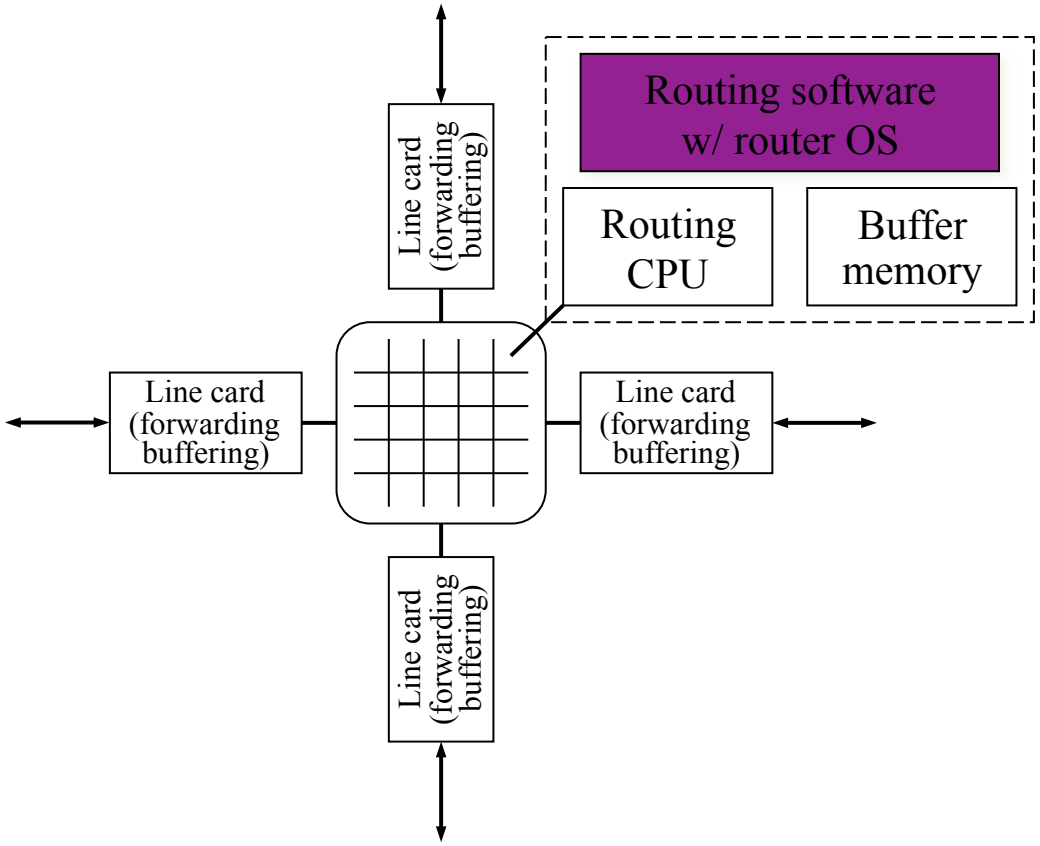


- Common Design: Batcher-Banyan Switch

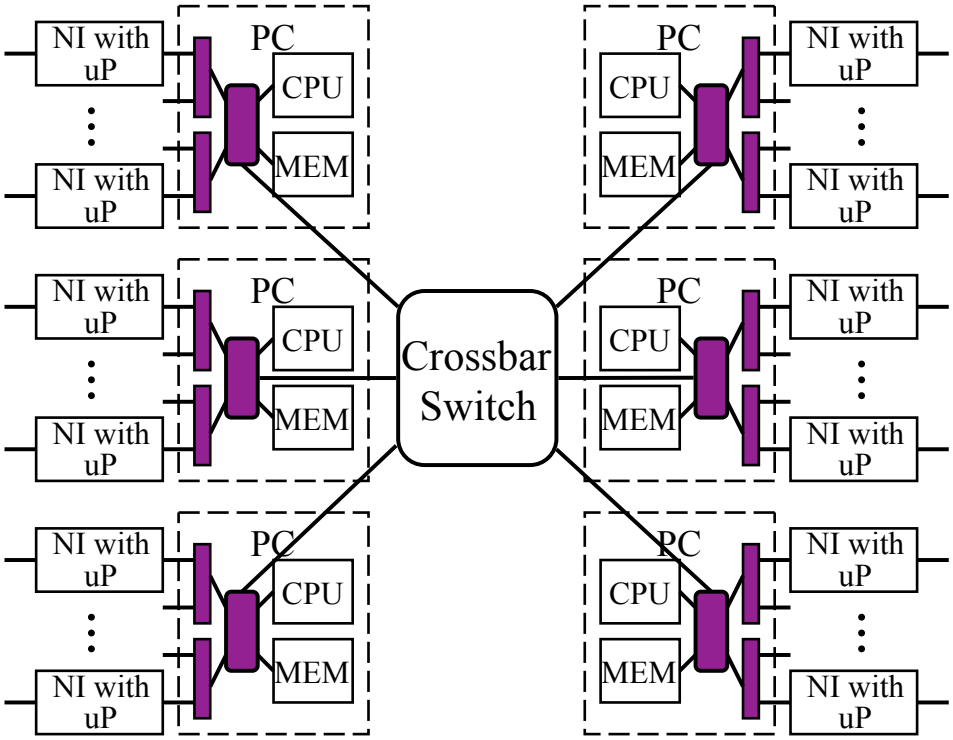
High-Speed IP Router

- Switch (possibly ATM)
- Line Cards + Forwarding Engines
  - link interface
  - router lookup (input)
  - common IP path (input)
  - packet queue (output)
- Network Processor
  - routing protocol(s)
  - exceptional cases

High-Speed Router



Alternative Design



ATM in the LAN

- ATM is used generally used for backbones
- ATM can also be used for LAN but requires special mechanisms to emulate LAN characteristics (e.g., broadcast used by ARP)
- Solutions:
  - New protocols that do not require broadcast (e.g., ATMARP)
  - Emulate shared media LAN: LAN Emulation (LANE)

LANE

- LANE servers:
  - LAN Emulation Configuration Server (LECS): *configuration*
  - LAN Emulation Server (LES): *configuration*
  - Broadcast and Unknown Server (BUS): *data transfer*
- LAN Emulation Client (LEC):
  - Is connected to the LECS through a predefined VC
  - Gets config info from LECS (e.g., type of LAN, maximum packet size, ATM address of the LES)
  - LEC registers with LES (ATMADDR, MACADDR), and gets the BUS ATMADDR
  - Broadcast is sent to BUS
  - Unicast: first packet sent to BUS + Address resolution request to LES, subsequent packets are directly sent to the destination over a newly established VC