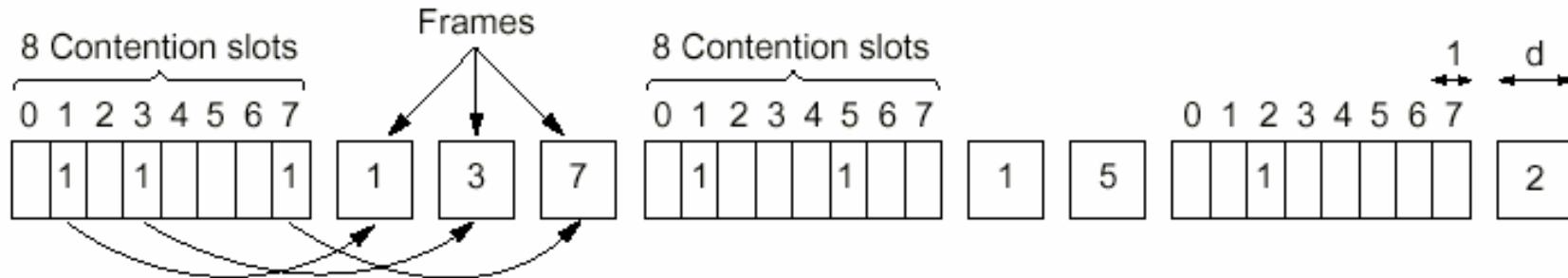


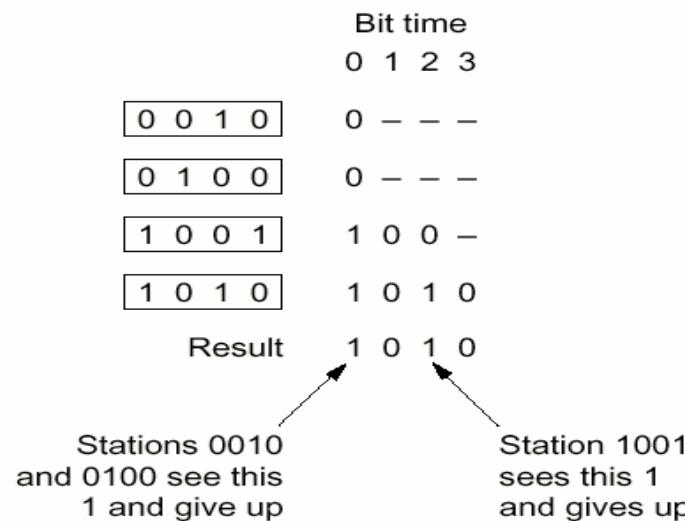
MAC Sublayer (9)

The basic Bit - Map Protocol ($N = 8$)



This protocol belongs to the class called **reservation protocols**.

The Binary Countdown Protocol



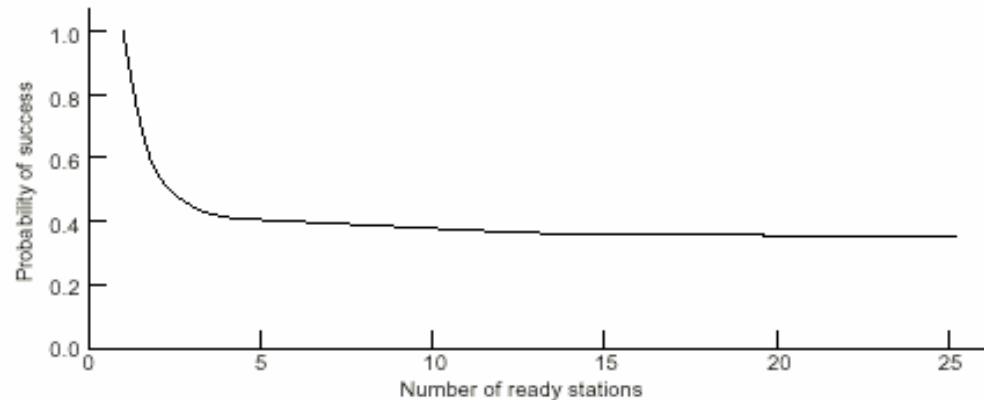
Limited Contention Protocols

Idea:

Combine the best properties of the contention and the collision-free protocols

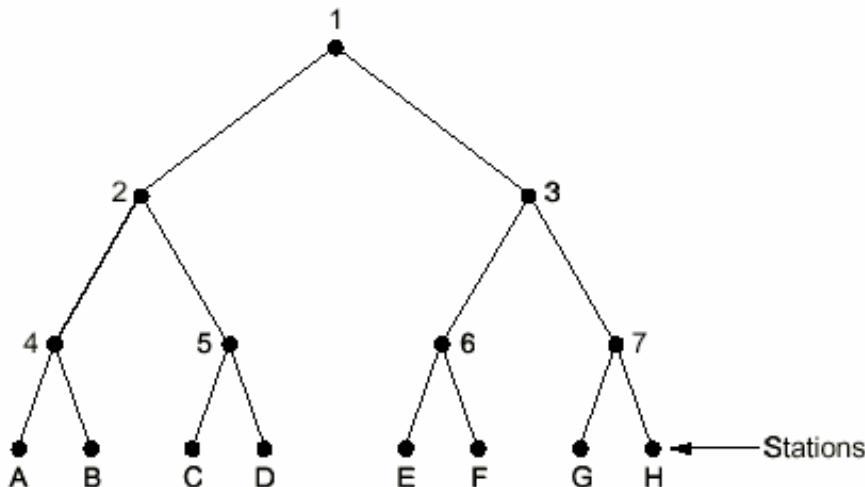
MAC Sublayer(10)

Probability of channel acquisition for a symmetric contention protocol



Consequence: Dynamic station assignment to slots depending on the load.

Example: The Adaptive Tree Walk Protocol



MAC Sublayer(28)

Summary of channel allocation methods:

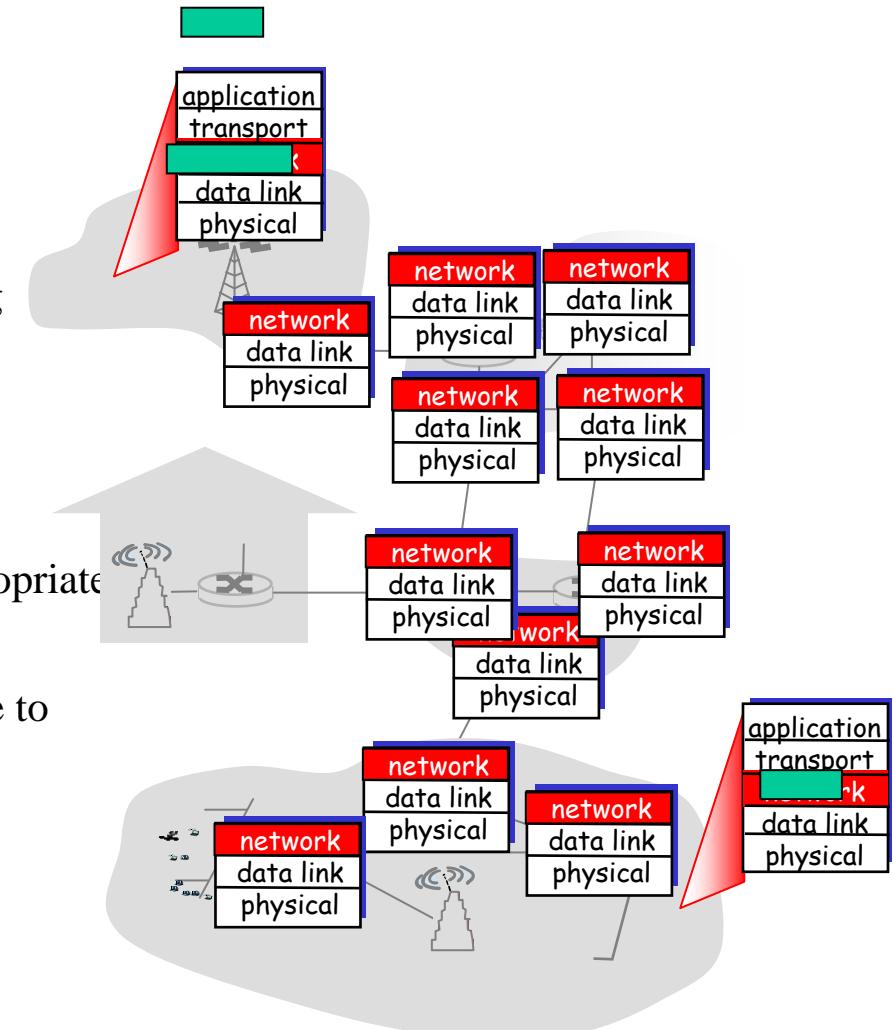
Method	Description
FDM	Dedicate a frequency band to each station
TDM	Dedicate a time slot to each station
Pure ALOHA	Unsynchronized transmission at any instant
Slotted ALOHA	Random transmission in well-defined time slots
1-persistent CSMA	Standard carrier sense multiple access
Nonpersistent CSMA	Random delay when channel is sensed busy
P-persistent CSMA	CSMA, but with a probability of p of persisting
CSMA/CD	CSMA, but abort on detecting a collision
Bit map	Round robin scheduling using a bit map
Binary countdown	Highest numbered ready station goes next
Tree walk	Reduced contention by selective enabling
Wavelength division	A dynamic FDM scheme for fiber
MACA, MACAW	Wireless LAN protocols
GSM	FDM plus TDM for cellular radio
CDPD	Packet radio within an AMPS channel
CDMA	Everybody speak at once but in a different language
Ethernet	CSMA/CD with binary exponential backoff
Token bus	Logical ring on a physical bus
Token ring	Capture the token to send a frame
DQDB	Distributed queuing on a two-bus MAN
FDDI	Fiber-optic token ring
HIPPI	Crossbar using 50-100 twisted pairs
Fibre channel	Crossbar using fiber optics
SPADE	FDM with dynamic channel allocation
ACTS	TDM with centralized slot allocation
Binder	TDM with ALOHA when slot owner is not interested
Crowther	ALOHA with slot owner getting to keep it
Roberts	Channel time reserved in advance by ALOHA

Network layer

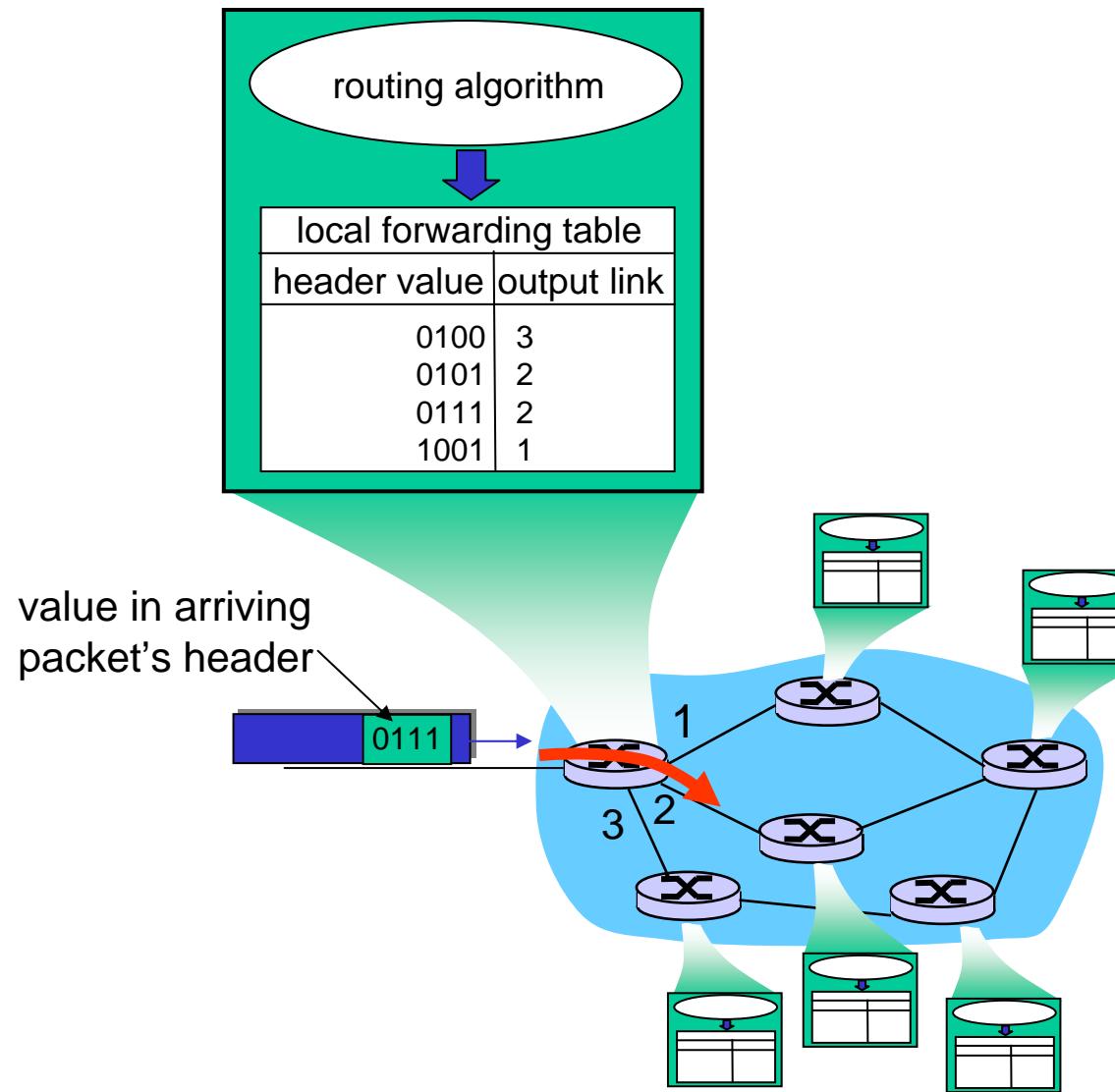
- transport segment from sending to receiving host
- on sending side encapsulates segments into datagrams
- on receiving side, delivers segments to transport layer
- network layer protocols in *every* host, router
- router checks header fields in all IP datagrams passing through it

Two Key Network-Layer Functions

- *forwarding*: move packets from router's input to appropriate router output
- *routing*: determine route taken by packets from source to destination
 - *routing algorithms*

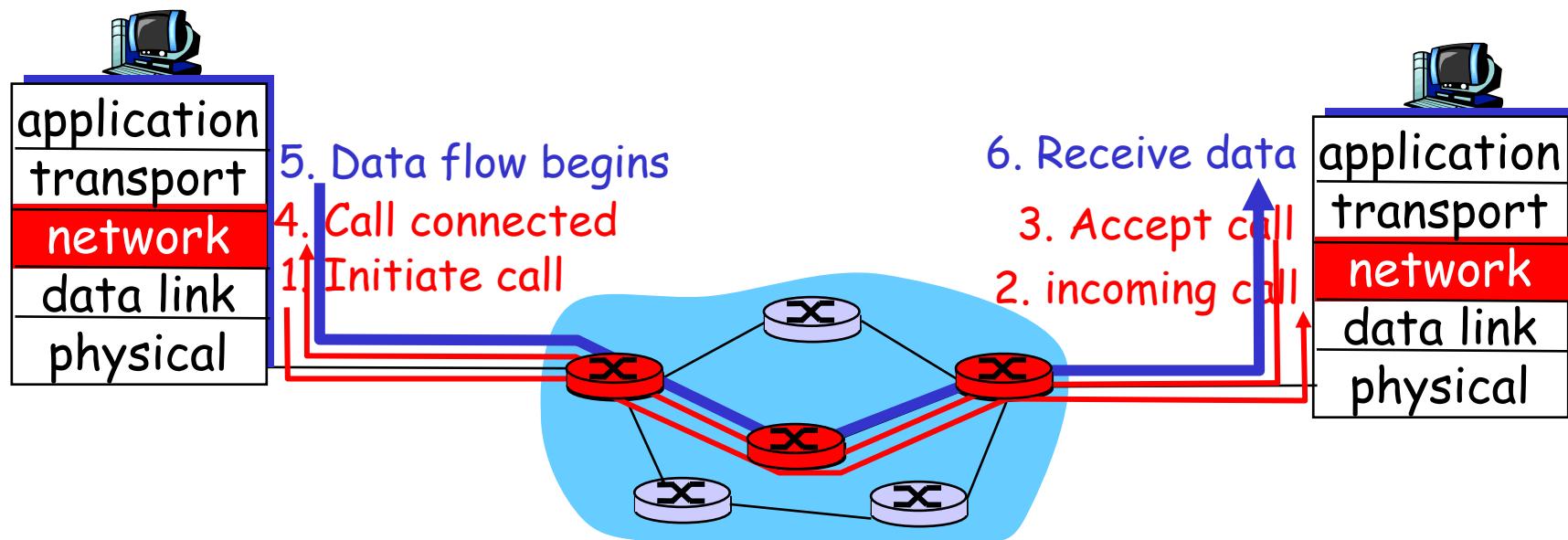


Interplay between routing and forwarding



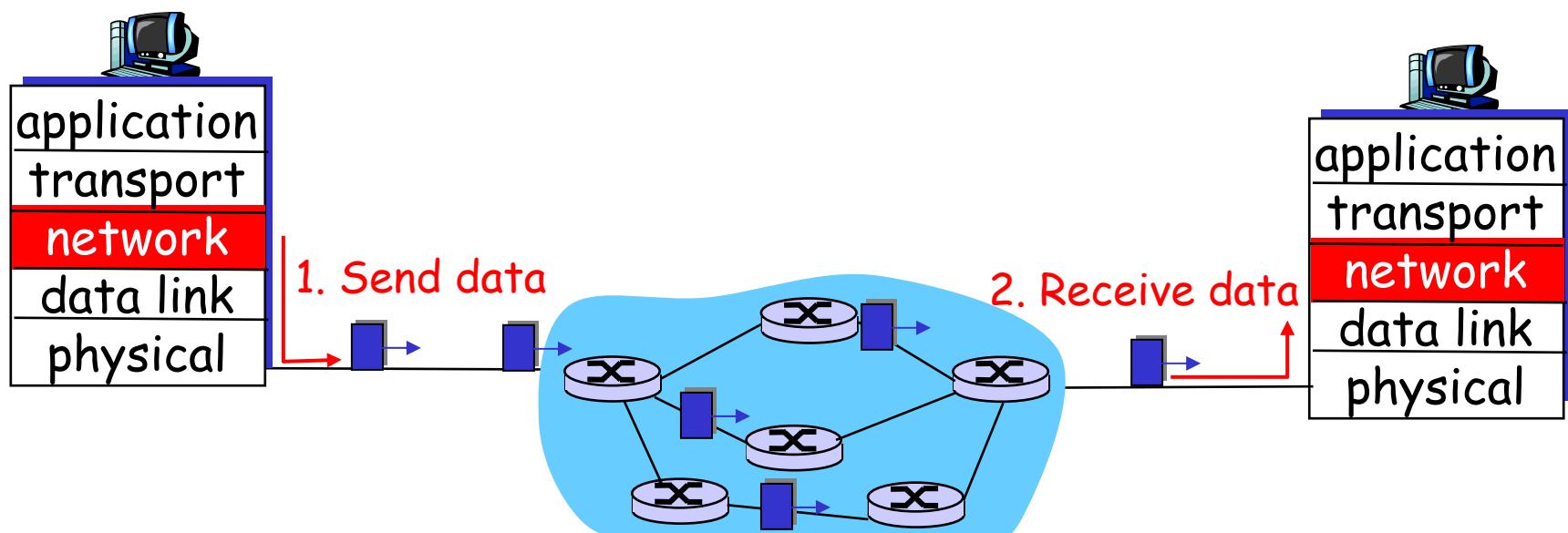
Virtual circuits

- used to setup, maintain, teardown VC
- used in ATM, X.25
- not used in today's Internet



Datagram networks

- no call setup at network layer
- routers: no state about end-to-end connections
 - no network-level concept of “connection”
- packets forwarded using destination host address
 - packets between same source-dest pair may take different paths



Forwarding table

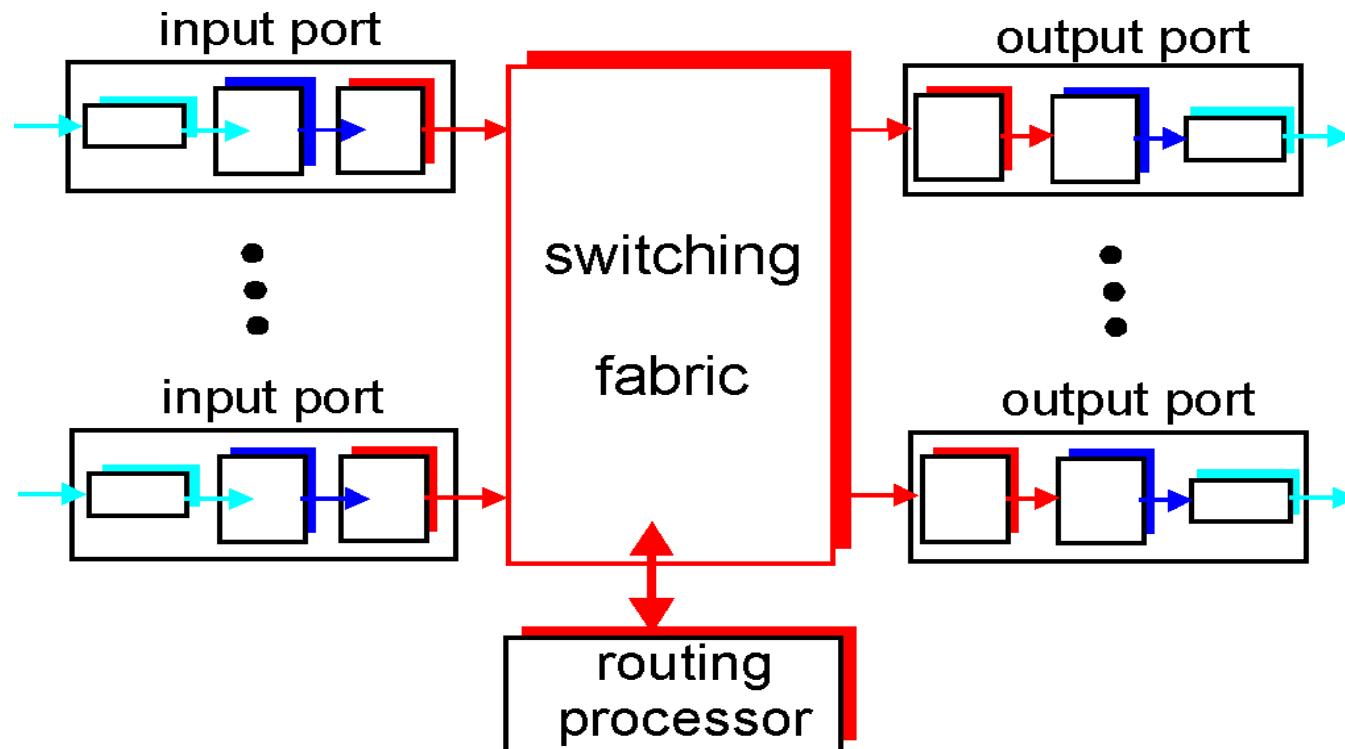
4 billion
possible entries

<u>Destination Address Range</u>	<u>Link Interface</u>
$11001000\ 00010111\ 00010000\ 00000000$ through $11001000\ 00010111\ 00010111\ 11111111$	0
$11001000\ 00010111\ 00011000\ 00000000$ through $11001000\ 00010111\ 00011000\ 11111111$	1
$11001000\ 00010111\ 00011001\ 00000000$ through $11001000\ 00010111\ 00011111\ 11111111$	2
otherwise	3

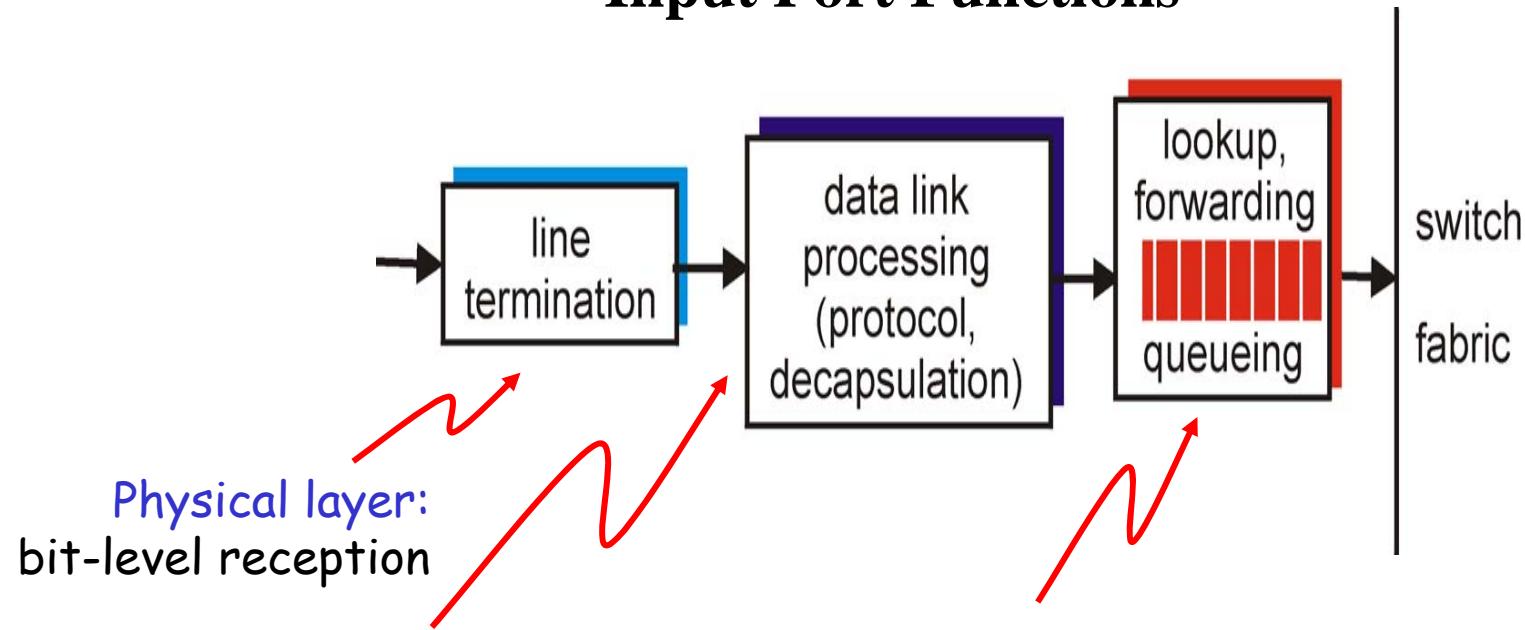
Router Architecture Overview

Two key router functions:

- *lookup*, run routing algorithms/protocol, i.e. setting the state of the switching fabric (RIP, OSPF, BGP)
- *forwarding* datagrams from incoming to outgoing link according to the actual state of the switching fabric

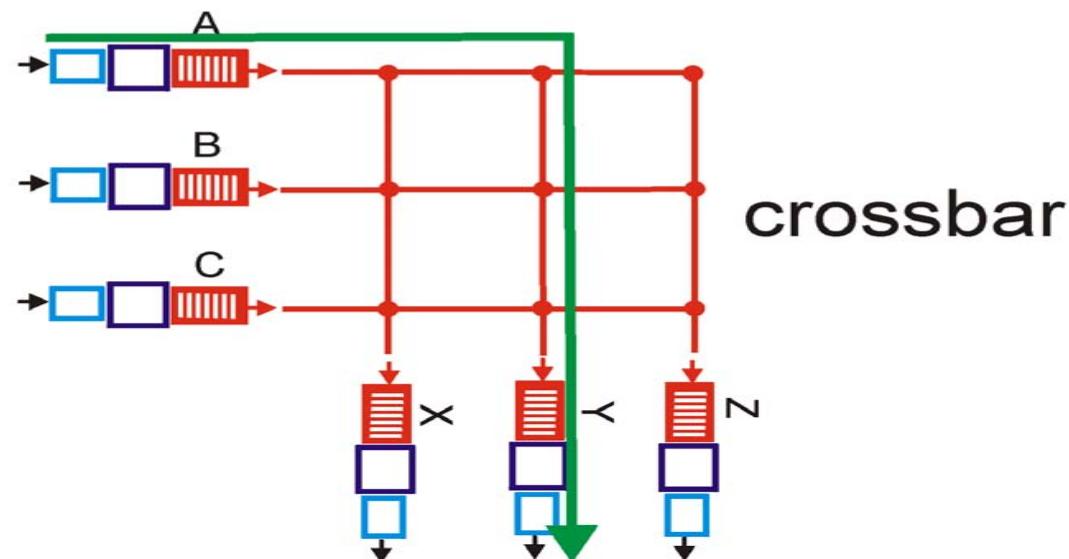
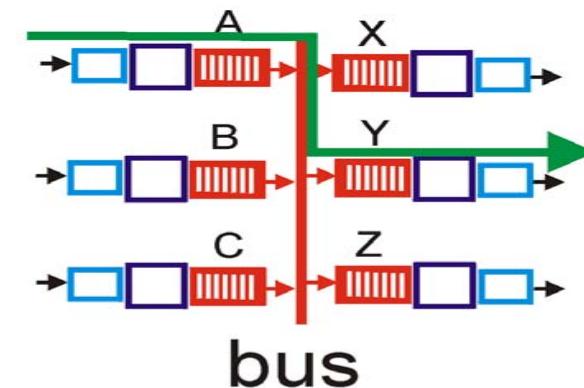
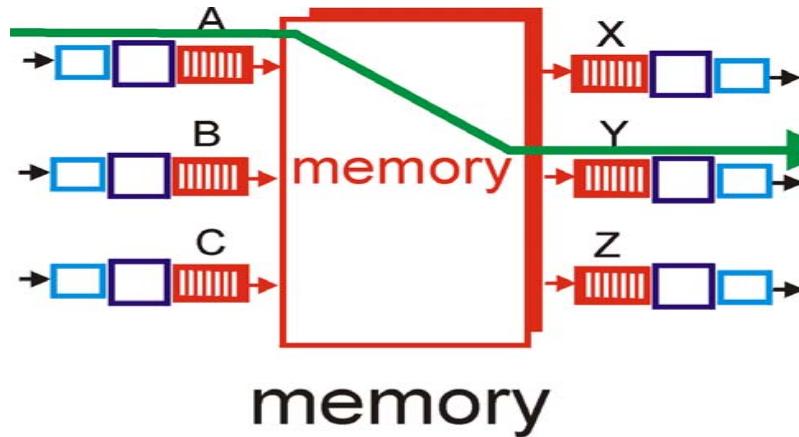


Input Port Functions



- given datagram destination, lookup output port using forwarding table in input port memory
- goal: complete input port processing at ‘line speed’ (transmission rate)
- queuing: if datagrams arrive faster than forwarding rate into switch fabric

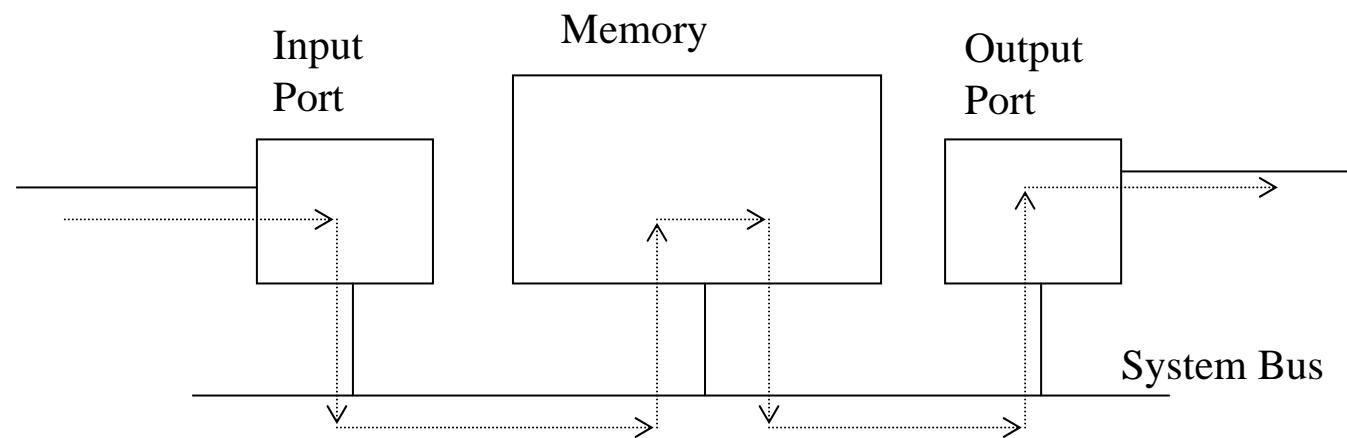
Three types of switching fabrics



Switching Via Memory

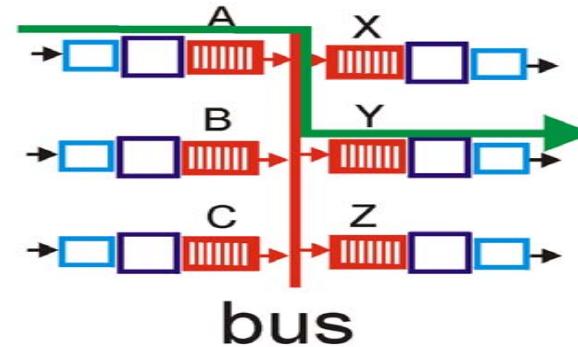
First generation routers:

- traditional computers with switching under direct control of CPU
- packet copied to system's memory
- speed limited by memory bandwidth



Switching Via a Bus

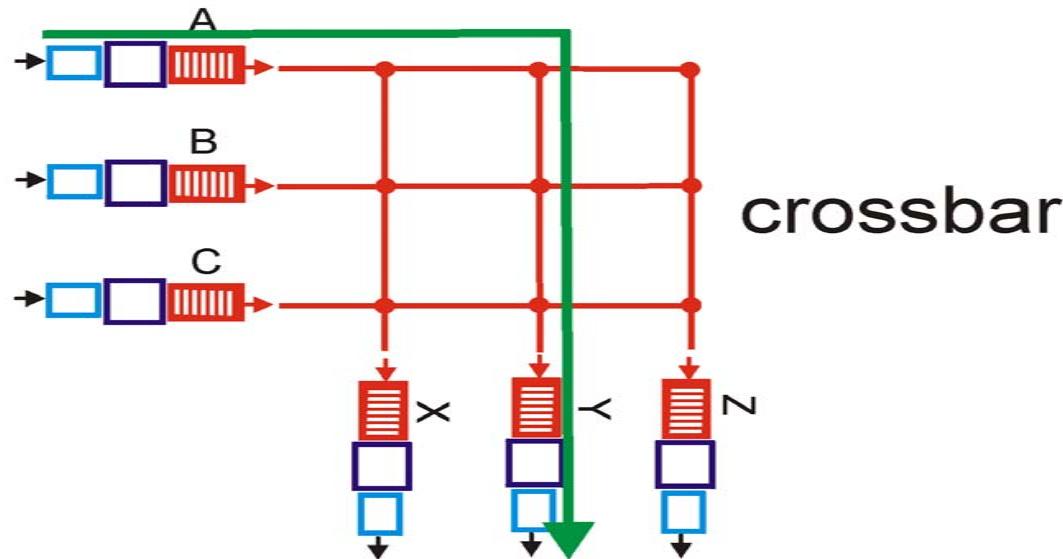
-



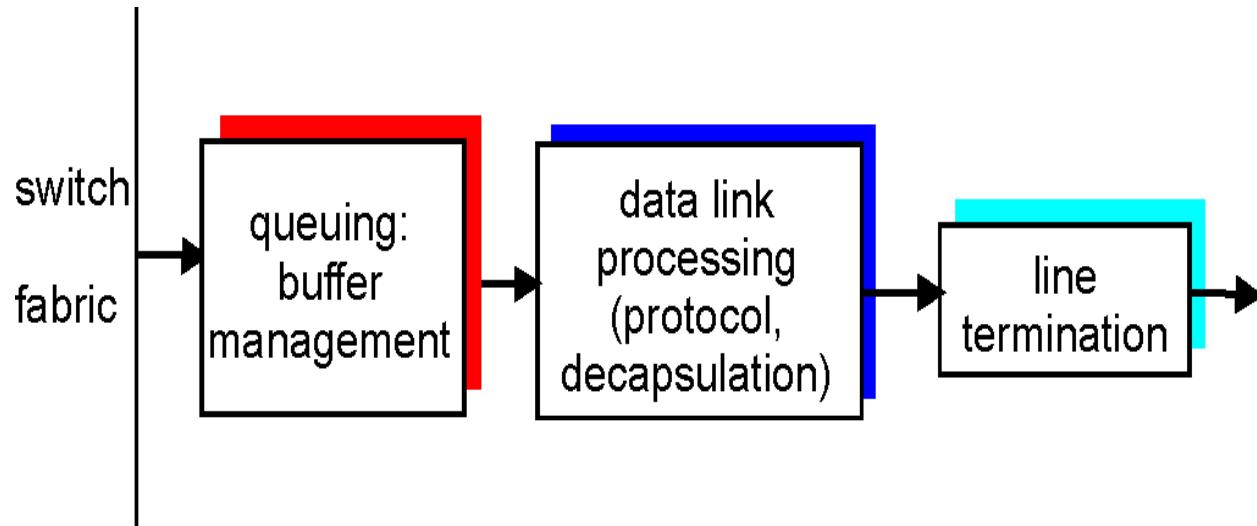
- datagram from input port memory to output port memory via a shared bus
- **bus contention:** switching speed limited by bus bandwidth
- 32 Gbps bus, Cisco 5600: sufficient speed for access and enterprise routers

Switching Via An Interconnection Network

- overcome bus bandwidth limitations
- crossbar switch: interconnection network of $2n$ buses connecting n input ports to n output ports.
- Cisco 12000: switches 60 Gbps through the interconnection network

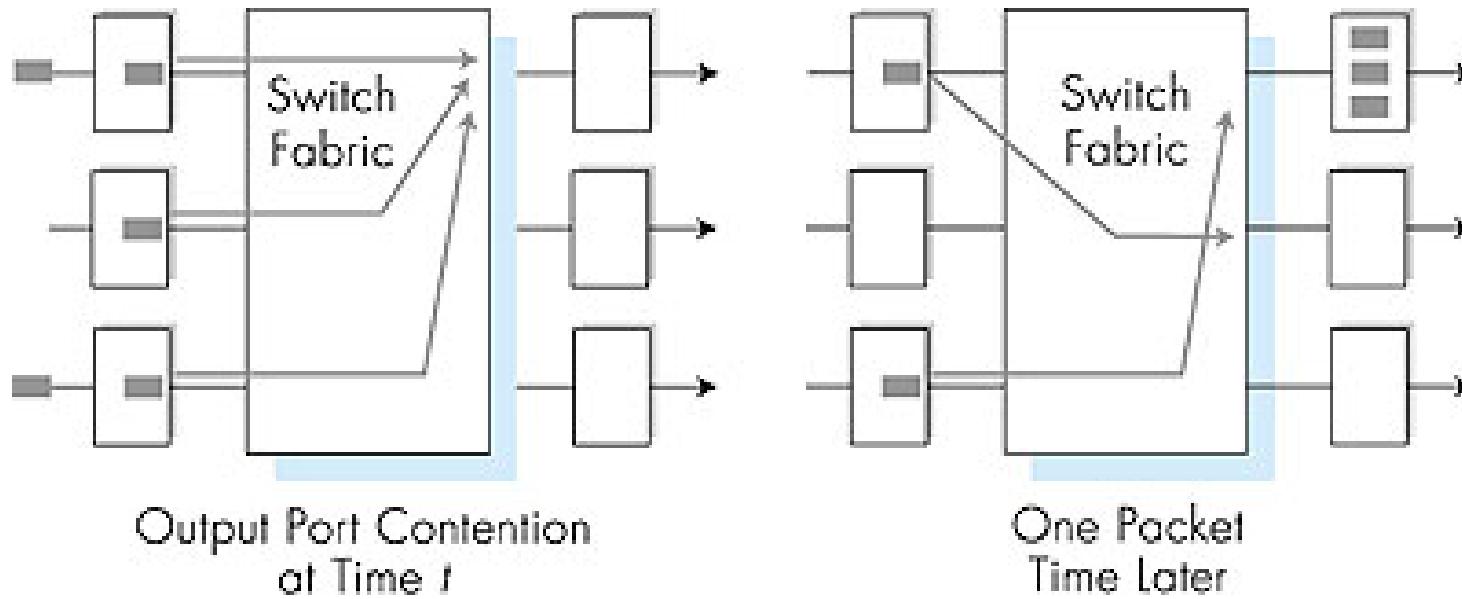


Output Ports



- *Buffering* required when datagrams arrive from fabric faster than the transmission rate
- *Scheduling discipline* chooses among queued datagrams for transmission

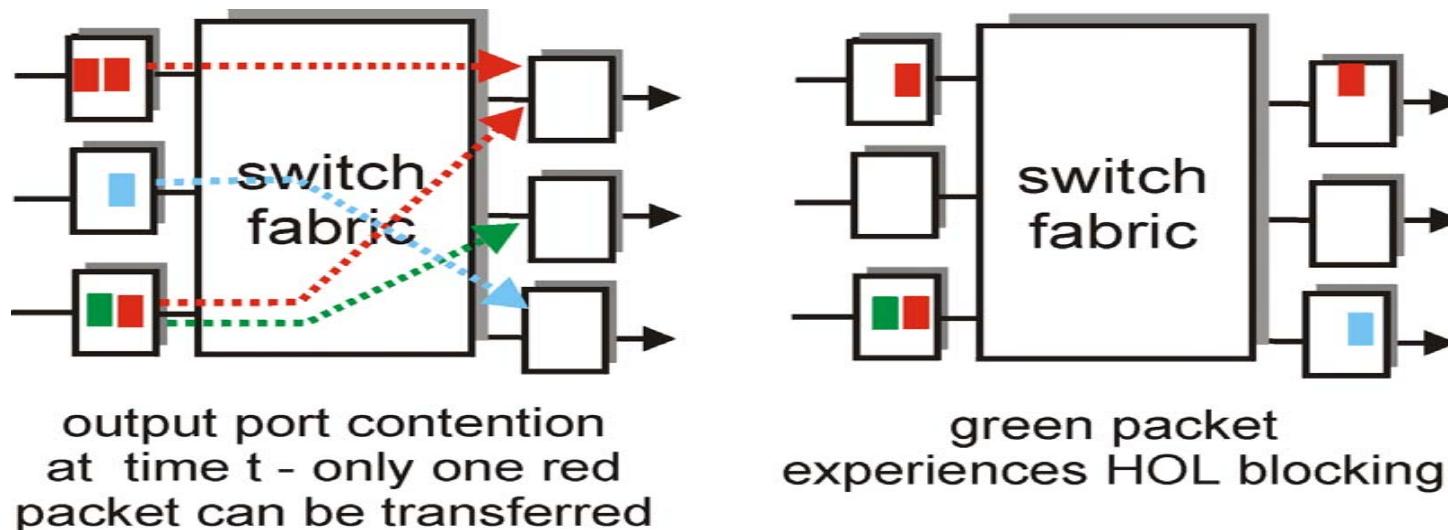
Output port queueing



- buffering when arrival rate via switch exceeds output line speed
- *queueing (delay) and loss due to output port buffer overflow!*

Input Port Queuing

- Fabric slower than input ports combined -> queueing may occur at input queues
- **Head-of-the-Line (HOL) blocking:** queued datagram at front of queue prevents others in queue from moving forward
- *queueing delay and loss due to input buffer overflow!*



Network Layer(7)

Distance Vector Routing

Idea:

Each router maintains a table (vector) giving a periodically updated best known distance to each destination and which line to use to get there

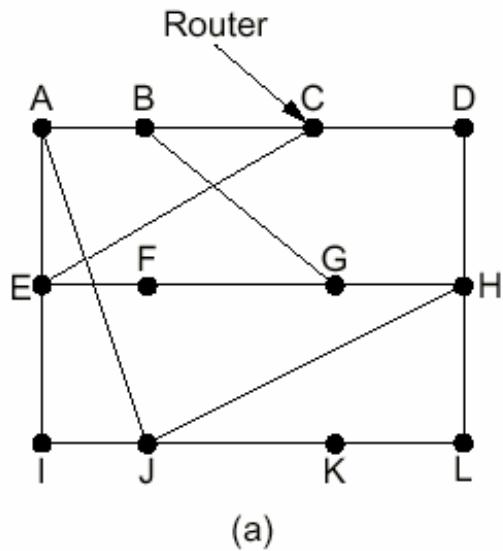
---> It is a *dynamic* routing algorithm

Used in early versions of

- ARPANET
- Internet
- DECnet
- Novell's IPX
- AppleTalk
- Cisco routers

Network Layer(8)

Example of the updating process



New estimated delay from J

Line

To	A	I	H	K	Line
A	0	24	20	21	8 A
B	12	36	31	28	20 A
C	25	18	19	36	28 I
D	40	27	8	24	20 H
E	14	7	30	22	17 I
F	23	20	19	40	30 I
G	18	31	6	31	18 H
H	17	20	0	19	12 H
I	21	0	14	22	10 I
J	9	11	7	10	0 -
K	24	22	22	0	6 K
L	29	33	9	9	15 K

JA delay is 8 JI delay is 10 JH delay is 12 JK delay is 6

Vectors received from J's four neighbors

New routing table for J

(b)