Network Layer - Switching

Dr. T. Venkatesh
Dept of CSE, IIT Guwahati

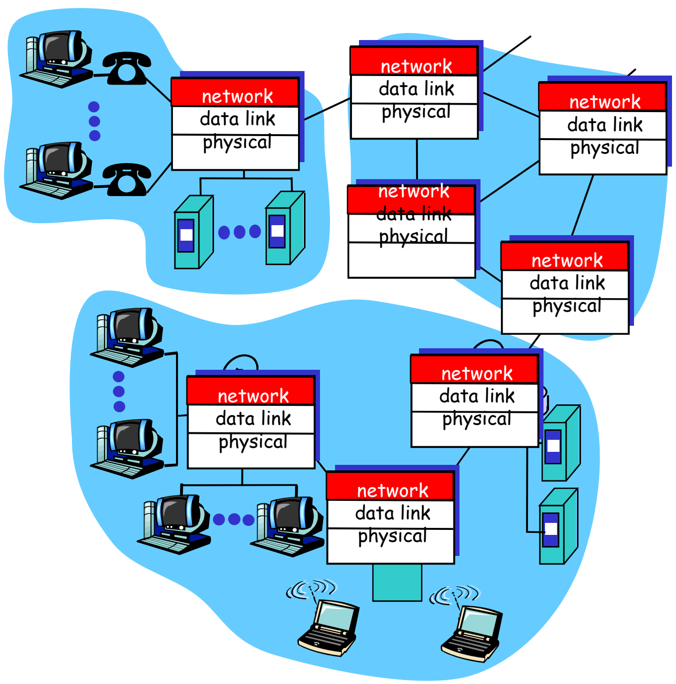
Slides from lectures of Prof. Srini Seshan, CMU and Prof. Jim Kurose, UMass, Amherst

Overview

- Principles behind network layer services
  - network layer service models
  - forwarding versus routing
  - Addressing for scalable routing
- How a router works
  - routing (path selection)
  - dealing with scale
- Advanced topics: IPv6, mobility
- Instantiation, implementation in the Internet

Network Layer

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



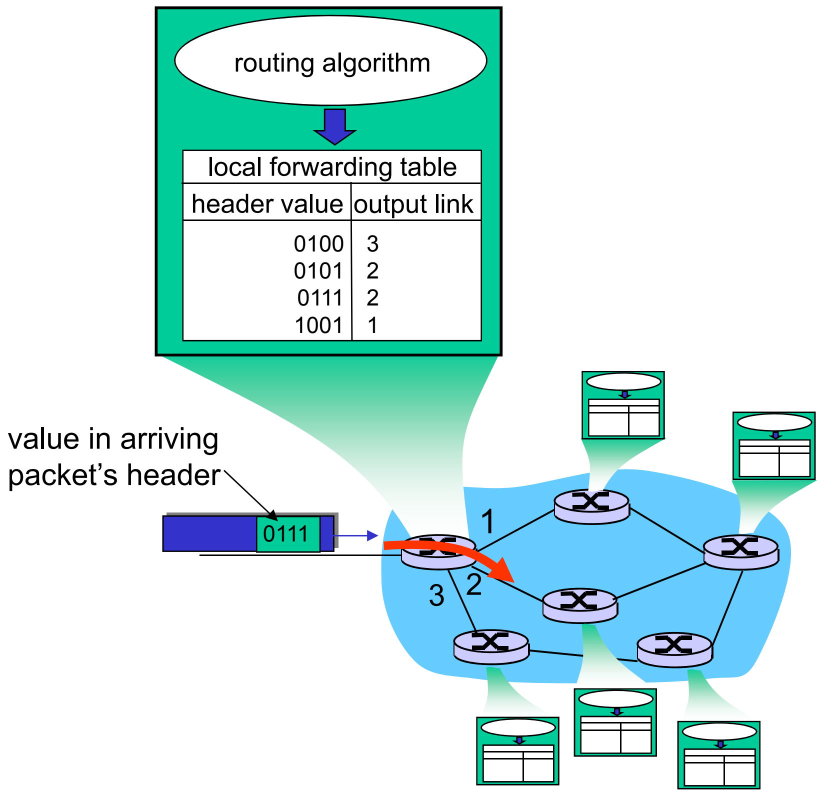
Key Network-Layer Functions

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

analogy:

- routing: process of planning trip from source to dest
- forwarding: process of correct left turns, right turns, exits, etc.

Interplay between routing and forwarding



Routing vs. Forwarding

- **Routing:** control plane
  - Computing paths the packets will follow
  - Routers talking amongst themselves
  - Creating the forwarding tables
- **Forwarding:** data plane
  - Directing a data packet to an outgoing link
  - Using the forwarding tables

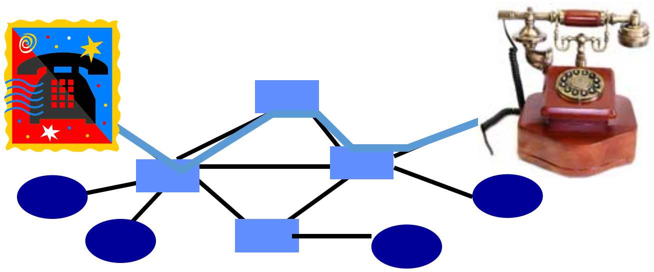
Service View: Connection setup

Q: What *service model* for "channel" transporting datagrams from sender to receiver?

- Example services for a datagram
  - Guaranteed delivery, guaranteed delivery with delay less than 100ms
- Example services for a flow of datagrams
  - In-order datagram delivery
  - Guaranteed minimum bandwidth of flow
  - Bounds on delay and jitter within the packet flow
- Before datagrams flow, two hosts and intervening routers are configured to a specific service model (*no choice*)
  - Routers get involved or Routers do not get involved
- Network and transport layer connection service
  - Network: between two hosts
  - Transport: between two processes

Guaranteed Service – Connection-oriented

- Source establishes connection
  - Reserve resources along hops in the path
- Source sends data
  - Transmit data over the established connection
- Source tears down connection
  - Free the resources for future connections
- Bandwidth and path fixed for every pair of users by the network

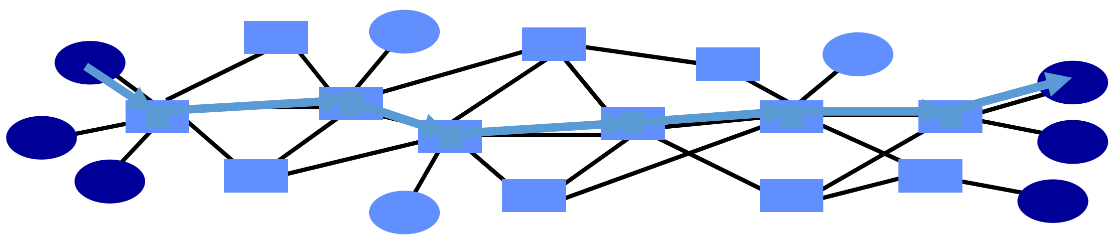


Pros and Cons

- Advantages
  - Predictable performance
  - Reliable, in-order delivery
  - Simple forwarding
  - No overhead for packet headers
- Disadvantages
  - Wasted bandwidth
  - Blocked connections
  - Connection set-up delay
  - Per-connection state inside the network

Connection-less or Datagram Service

- Message divided into packets
  - Header identifies the destination address
- Packets travel separately through the network
  - Forwarding based on the destination address
  - Packets may be buffered temporarily
- Destination reconstructs the message
- Resources not reserved – no guarantee of packet delivery in congestion



Best-Effort: Good Enough?

- Packet loss and delay
  - Sender can resend
- Packet corruption
  - Receiver can detect, and sender can resend
- Out-of-order delivery
  - Receiver can put the data back in order
- Packets follow different paths
  - Doesn’t matter
- Network failure
  - Drop the packet
- Network congestion
  - Drop the packet

Virtual circuits

"source-to-dest path behaves much like telephone circuit"

- performance-wise
- network actions along source-to-dest path

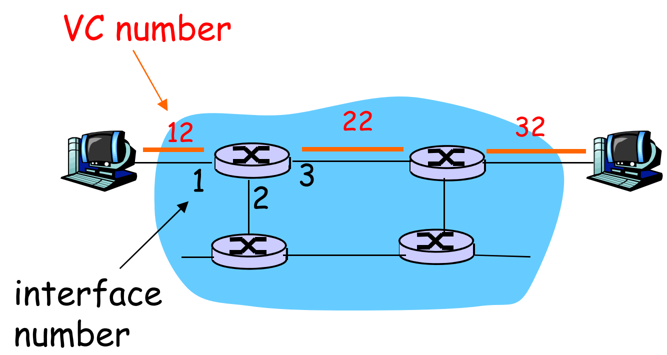
- call setup, teardown for each call *before* data can flow
- each packet carries VC identifier (not destination host address)
- *every* router on source-dest path maintains "state" for each passing connection
- link, router resources (bandwidth, buffers) may be *allocated* to VC

A VC consists of:

1. Path from source to destination
2. VC numbers, one number for each link along path
3. Entries in forwarding tables in routers along path

- VC numbers are configured as a part of forwarding table
  - Signaling protocol used to configure VC paths

Forwarding table



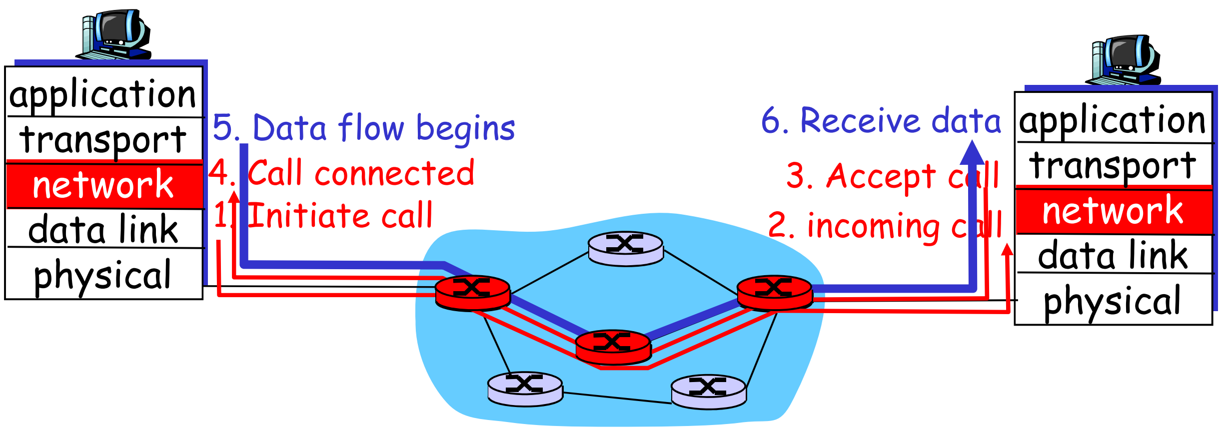
Forwarding table in northwest router:

Incoming interface	Incoming VC #	Outgoing interface	Outgoing VC #
1	12	3	22
2	63	1	18
3	7	2	17
1	97	3	87
…	…	…	…

Routers maintain connection state information!

Virtual circuits: signaling protocols

- used to setup, maintain teardown VC
- used in ATM, frame-relay, X.25 (not used in today's Internet)
- Multi-protocol label switching uses similar ideas



Datagram or VC network: why?

Internet

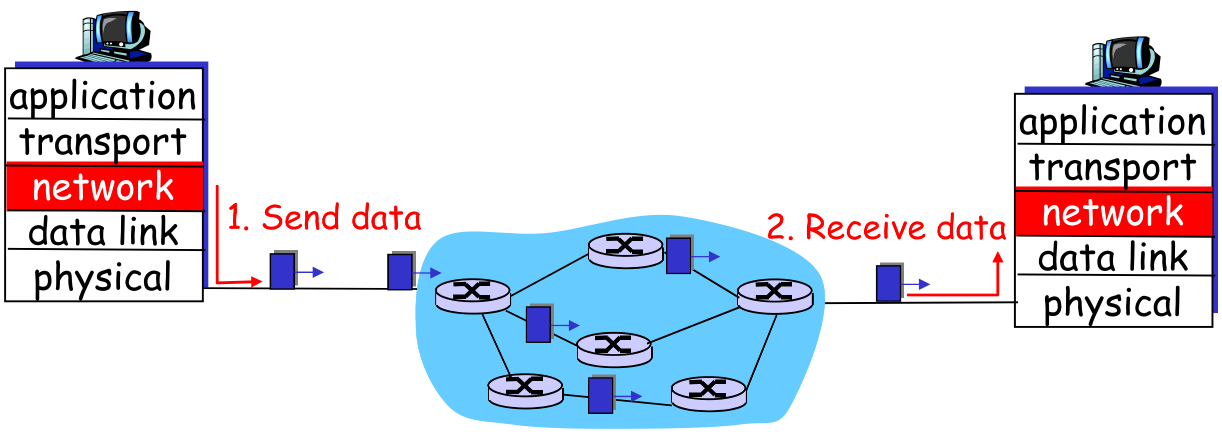
- data exchange among computers
  - "elastic" service, no strict timing req.
- "smart" end systems (computers)
  - can adapt, perform control, error recovery
  - simple inside network, complexity at "edge"
- many link types
  - different characteristics
  - uniform service difficult

ATM (VC)

- evolved from telephony
- human conversation:
  - strict timing, reliability requirements
  - need for guaranteed service
- "dumb" end systems
  - telephones
  - complexity inside network

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

Destination Address Range	Link Interface
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

Longest prefix matching

Prefix Match	Link Interface
11001000 00010111 00010	0
11001000 00010111 00011000	1
11001000 00010111 00011	2
otherwise	3

Examples

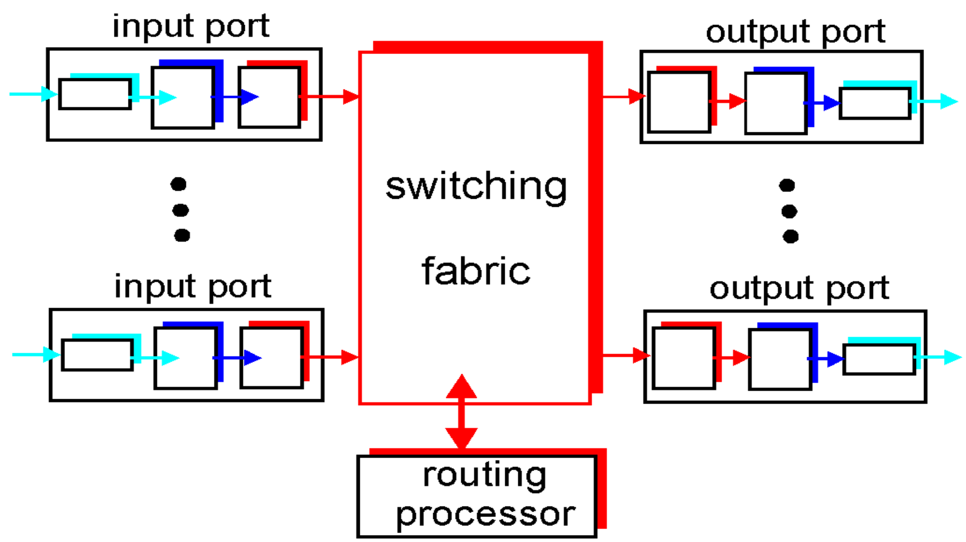
DA: 11001000 00010111 00010110 10100001 Which interface?

DA: 11001000 00010111 00011000 10101010 Which interface?

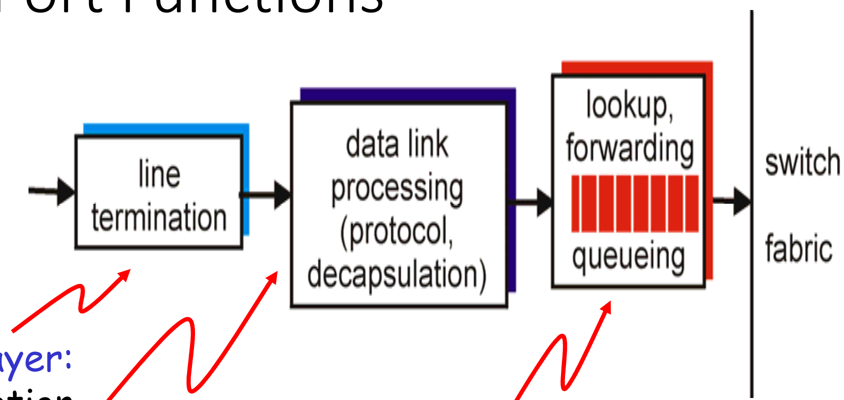
Router Architecture Overview

Two key router functions:

- run routing algorithms/protocol (RIP, OSPF, BGP)
- *forwarding* datagrams from incoming to outgoing link



Input Port Functions



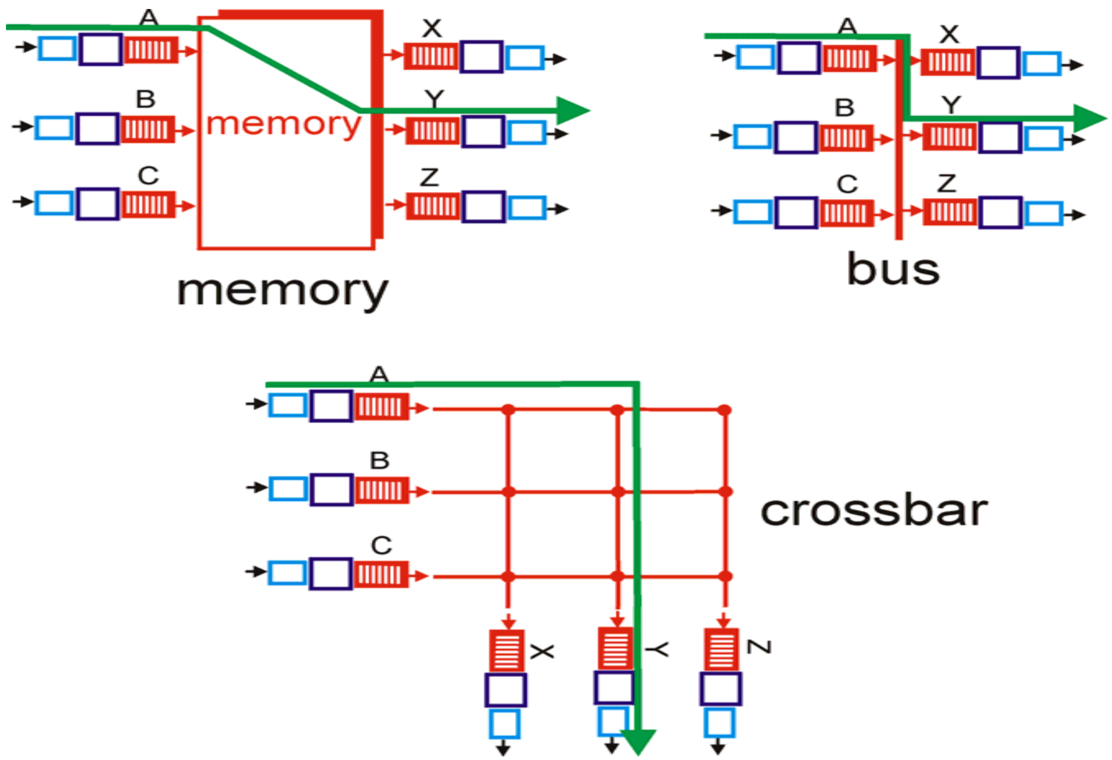
Physical layer:
bit-level reception

Data link layer:
e.g., Ethernet
see chapter 5

Decentralized switching*:*

- given datagram dest., lookup output port using forwarding table in input port memory
- goal: complete input port processing at 'line speed'
- queuing: if datagrams arrive faster than forwarding rate into switch fabric

Three types of switching fabrics



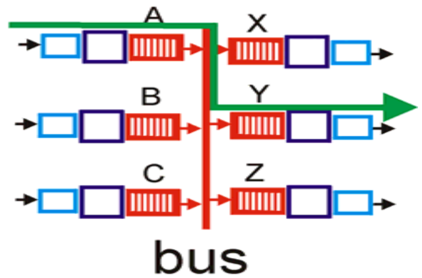
Switching Via Memory

First generation routers:

- traditional computers with switching under direct control of CPU
- packet copied to system's memory
- speed limited by memory bandwidth (2 bus crossings per datagram)

- Improvisation: Each line card has copy of forwarding table
  - Update the forwarding table on the fly at each line card.

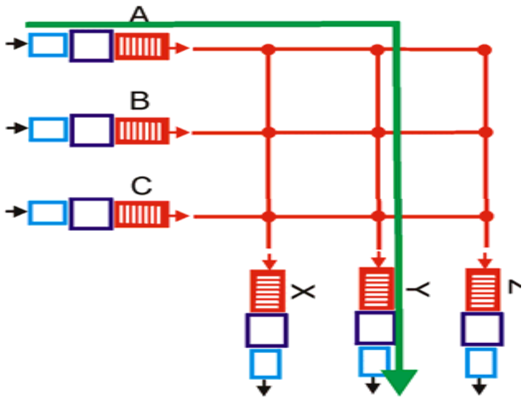
Switching Via a Bus



- datagram from input port memory
  to output port memory via a shared bus
  - Cache bits of forwarding table in line cards, send directly over bus to outbound line card
- bus contention: switching speed limited by bus bandwidth
- 1 Gbps bus, Cisco 1900: sufficient speed for access and enterprise routers (not regional or backbone)

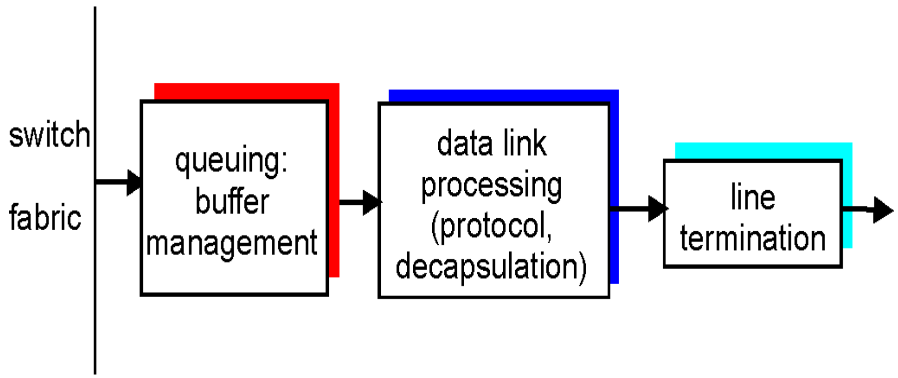
Switching Via An Interconnection Network

- overcome bus bandwidth limitations
- interconnection nets developed to connect processors in multiprocessor
- Advanced design: fragmenting datagram into fixed length cells, switch cells through the fabric.
- Cisco 12000: switches Gbps through the interconnection network



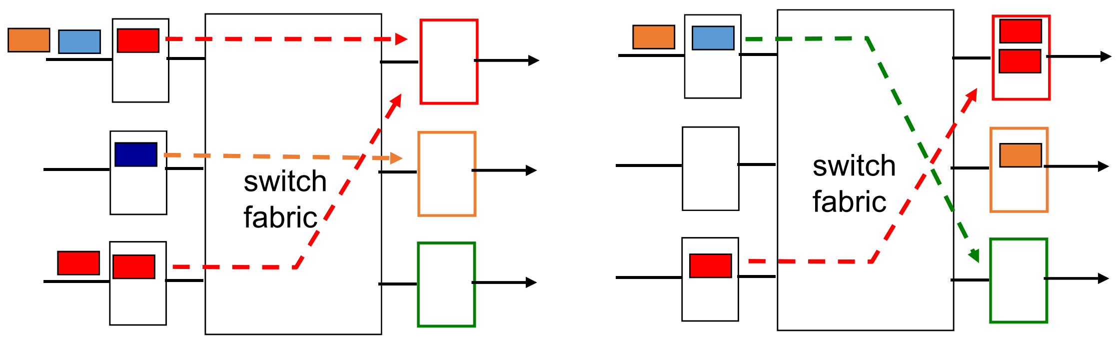
- Every input port has a connection to every output port
- During each timeslot, each input connected to zero or one outputs
- **Advantage:** Exploits parallelism
- **Disadvantage:** Need scheduling algorithm

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



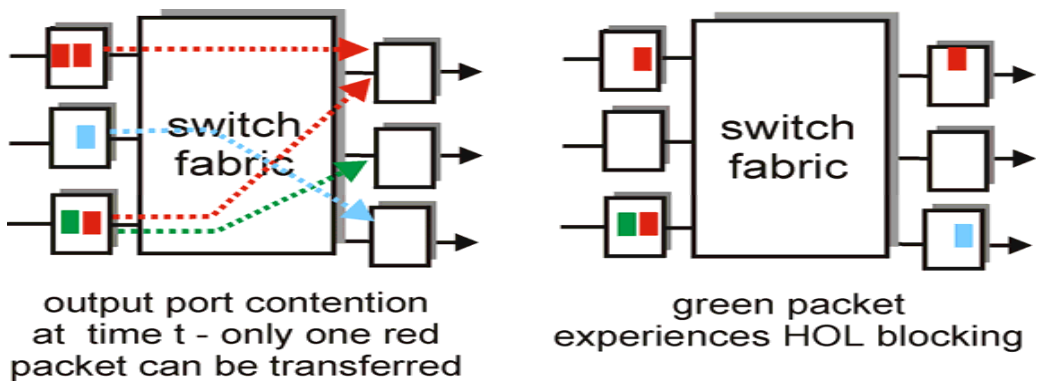
at *t,* packets more from input to output

one packet time later

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

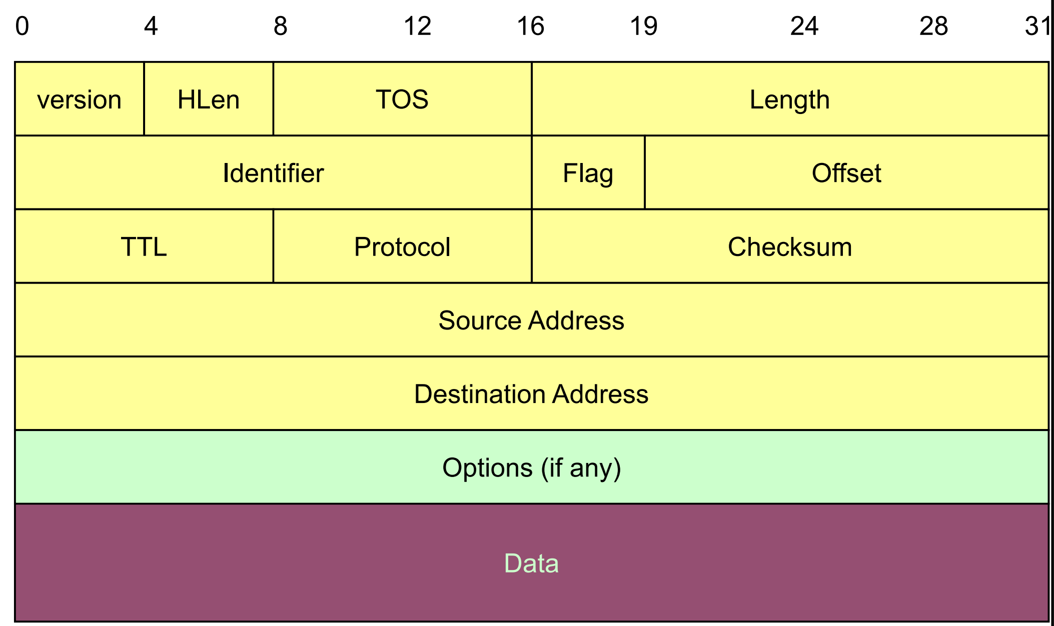
Input Port Queuing

- 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!*

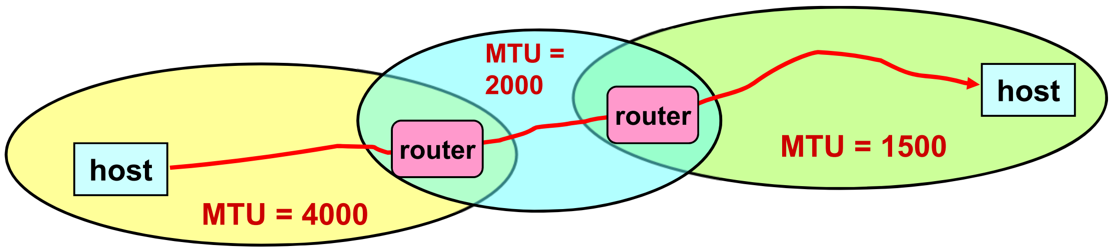


IPv4 Datagram

- Hlen: Indicated in 32-bit words (typically 5)
- TOS: Used in differentiated services
- Identifier, flags, fragment offset → used primarily for fragmentation
- Time to live (number of hops)
  - Must be decremented at each router
  - Packets with TTL=0 are thrown away
  - Ensure packets exit the network
- Protocol
  - Demultiplexing to higher layer protocols
  - TCP = 6, ICMP = 1, UDP = 17…
- Header checksum
  - Ensures some degree of header integrity
  - Relatively weak – 16 bit
- Options
  - E.g. Source routing, record route, etc.
  - Performance issues
    - Poorly supported



IP Fragmentation



- Every network has own Maximum Transmission Unit (MTU)
  - Largest IP datagram it can carry within its own packet frame
    - E.g., Ethernet is 1500 bytes
  - Don't know MTUs of all intermediate networks in advance
- IP Solution
  - When hit network with small MTU, fragment packets

Reassembly

- Where to do reassembly?
  - End nodes or at routers?
- End nodes
  - Avoids unnecessary work where large packets are fragmented multiple times
  - If any fragment missing, delete entire packet
- Dangerous to do at intermediate nodes
  - How much buffer space required at routers?
  - What if routes in network change?
    - Multiple paths through network
    - All fragments only required to go through destination

Fragmentation Related Fields

- Length
  - Length of IP fragment
- Identification
  - To match up with other fragments
- Flags
  - Don’t fragment flag
  - More fragments flag
- Fragment offset
  - Where this fragment lies in entire IP datagram
  - Measured in 8 octet units (13 bit field)

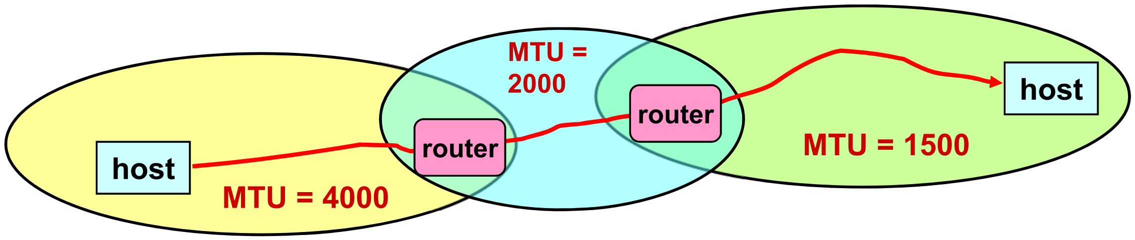
Fragmentation and Reassembly Concepts

- Demonstrates many Internet concepts
- Decentralized
  - Every network can choose MTU
- Connectionless
  - Each (fragment of) packet contains full routing information
  - Fragments can proceed independently and along different routes
- Best effort
  - Fail by dropping packet
  - Destination can give up on reassembly
  - No need to signal sender that failure occurred
- Complex endpoints and simple routers
  - Reassembly at endpoints

Internet Control Message Protocol (ICMP)

- Short messages used to send error & other control information
- Examples
  - Ping request / response
    - Can use to check whether remote host reachable
  - Destination unreachable
    - Indicates how packet got & why couldn't go further
  - Flow control
    - Slow down packet delivery rate
  - Redirect
    - Suggest alternate routing path for future messages
  - Router solicitation / advertisement
    - Helps newly connected host discover local router
  - Timeout
    - Packet exceeded maximum hop limit

IP MTU Discovery with ICMP



- Typically send series of packets from one host to another
- Typically, all will follow same route
  - Routes remain stable for minutes at a time
- Makes sense to determine path MTU before sending real packets
- Operation
  - Send max-sized packet with “do not fragment” flag set
  - If encounters problem, ICMP message will be returned
    - “Destination unreachable: Fragmentation needed”
    - Usually indicates MTU encountered