# Computer Networks

Wenzhong Li, Chen Tian

Nanjing University

*Material with thanks to James F. Kurose, Mosharaf Chowdhury, and other colleagues.*


# Chapter 3. Network Layer

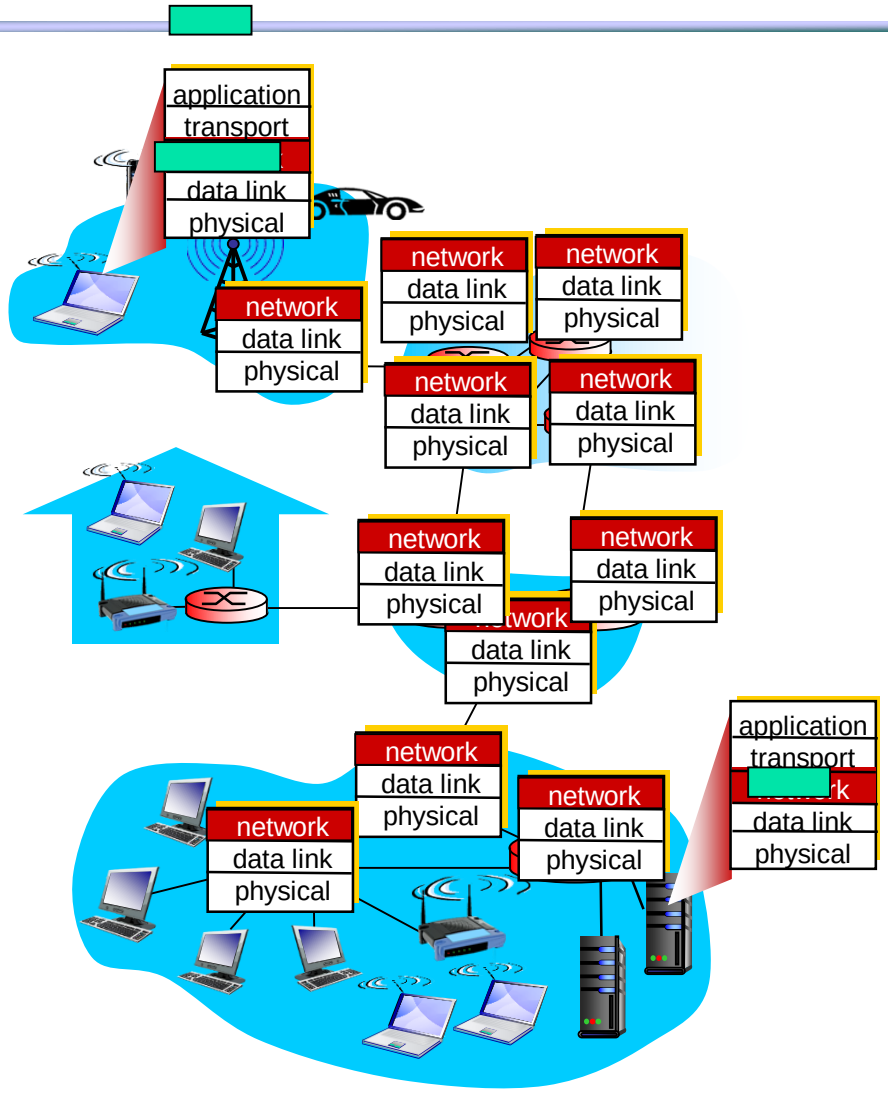
- Network Layer Functions
- IP Protocol Basic
- IP Protocol Suit
- Routing Fundamentals
- Internet Routing Protocols
- IP Multicasting


- Network Layer Functions
- IP Routers
- Virtual Circuit and Datagram Networks


# Network Layer Functions

# 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 examines header fields in all IP datagrams passing through it



# Two Key Network-layer Functions

- OSI network-layer functions:
- Switching / Routing
  - Determine route taken by packets from source to destination (multiple nodes)
  - Shortest path from source to destination
  - Routing algorithms
- Forwarding
  - Move packets from input to designated output determined by switching (single node)
  - Error handling, queuing and scheduling

analogy: taking a trip

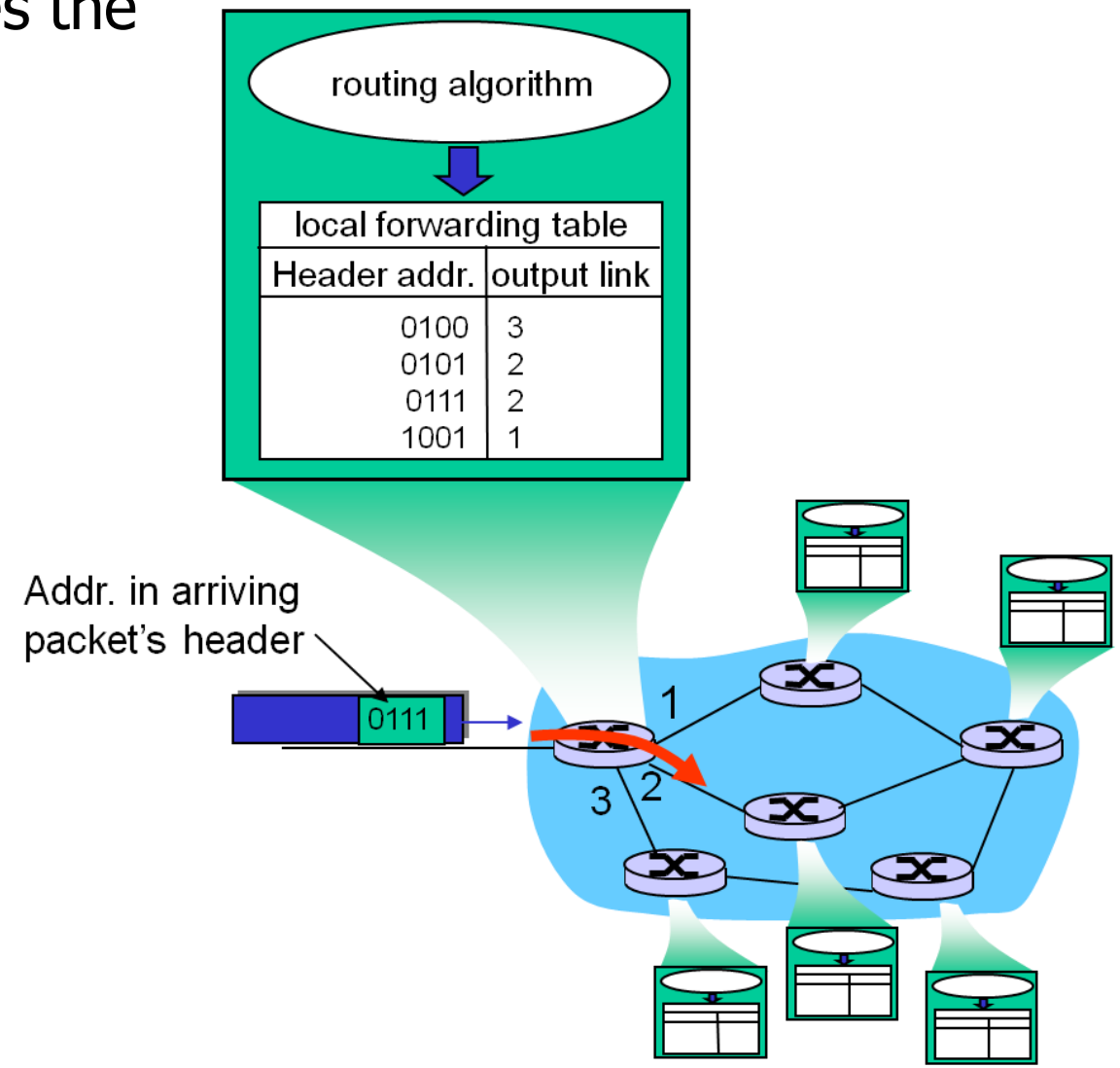
- *forwarding*: process of getting through single interchange
- *routing*: process of planning trip from source to destination

forwarding

routing

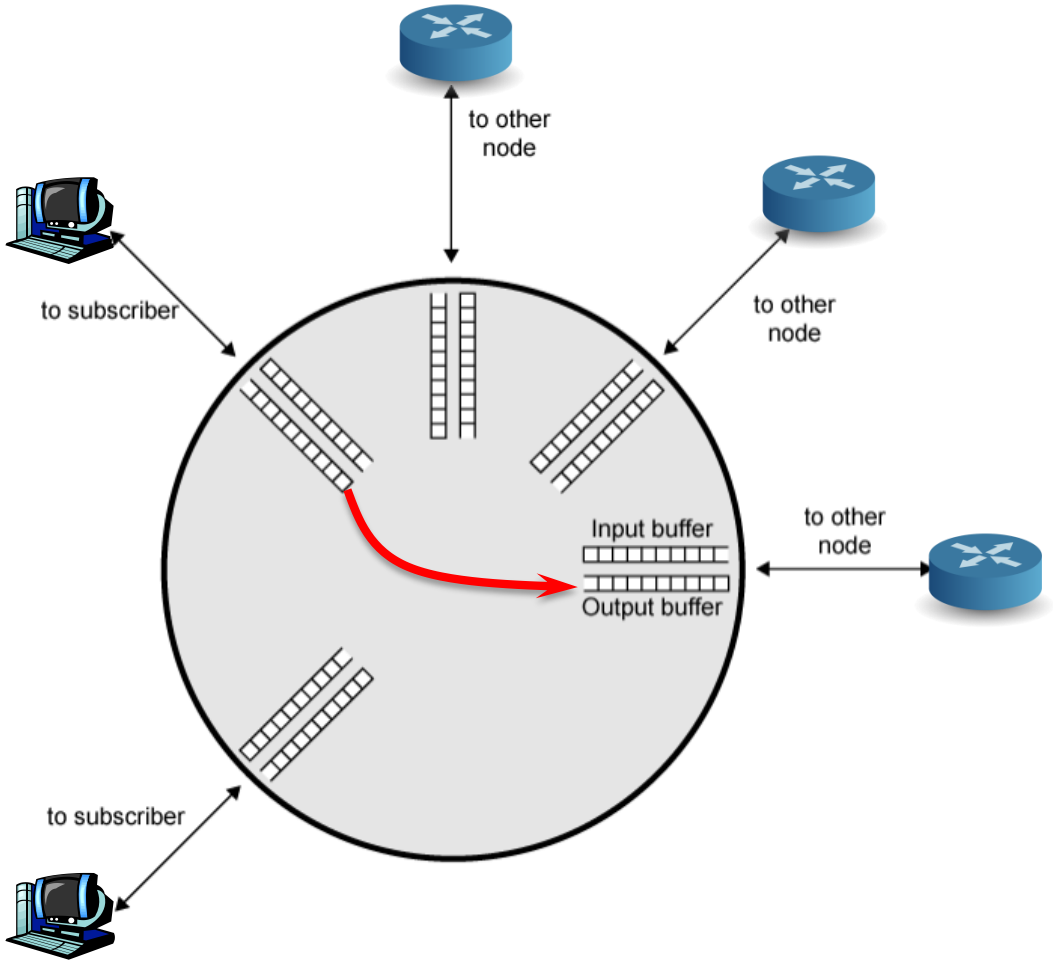
# Switch Functions

- Routing determines the forwarding table



# Forwarding Functions

- Queuing and scheduling
  - Host to Switch
  - Switch to Host
  - Switch to Switch



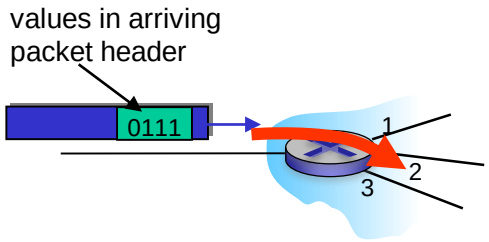
# Connection setup

- 3rd important function in *some* network architectures:
  - ATM, frame relay, X.25
- Before datagrams flow, two end hosts *and* intervening routers establish virtual connection
  - Routers get involved
- Network vs transport layer connection service:
  - *network:* between two hosts (may also involve intervening routers in case of VCs)
  - *transport:* between two processes

# Network layer: data plane, control plane

Data plane:

- *local*, per-router function
- determines how datagram arriving on router input port is forwarded to router output port



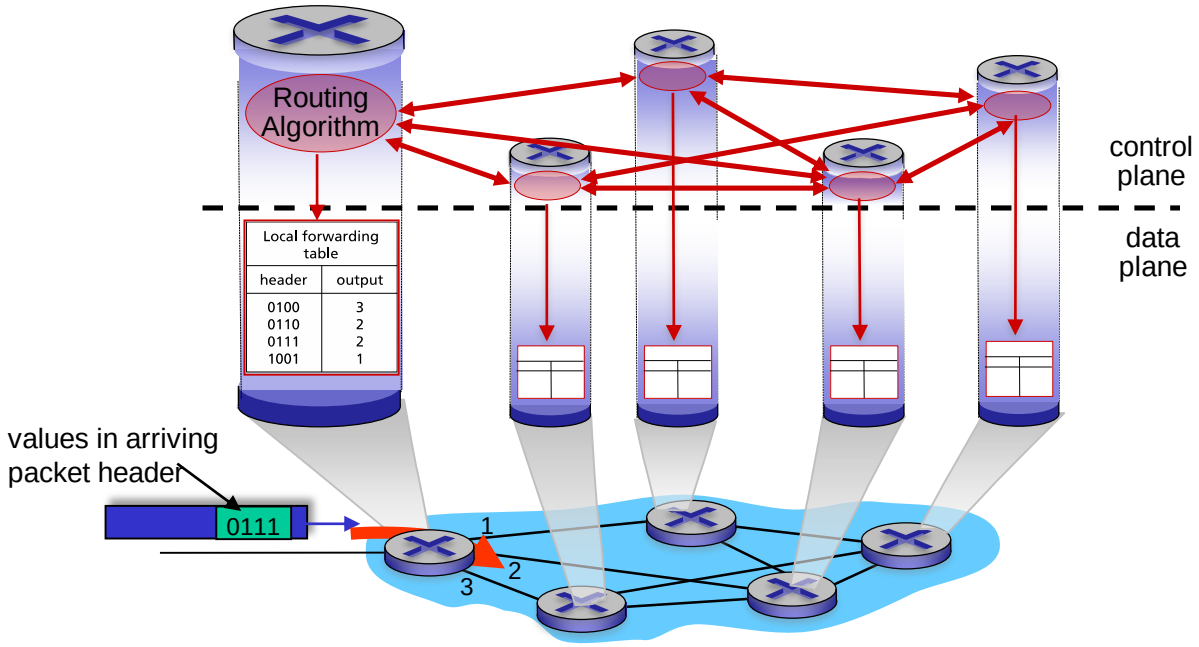

Control plane

- *network-wide* logic
- determines how datagram is routed among routers along end-end path from source host to destination host

- two control-plane approaches:
  - *traditional routing algorithms:* implemented in routers
  - *software-defined networking (SDN)*: implemented in (remote) servers

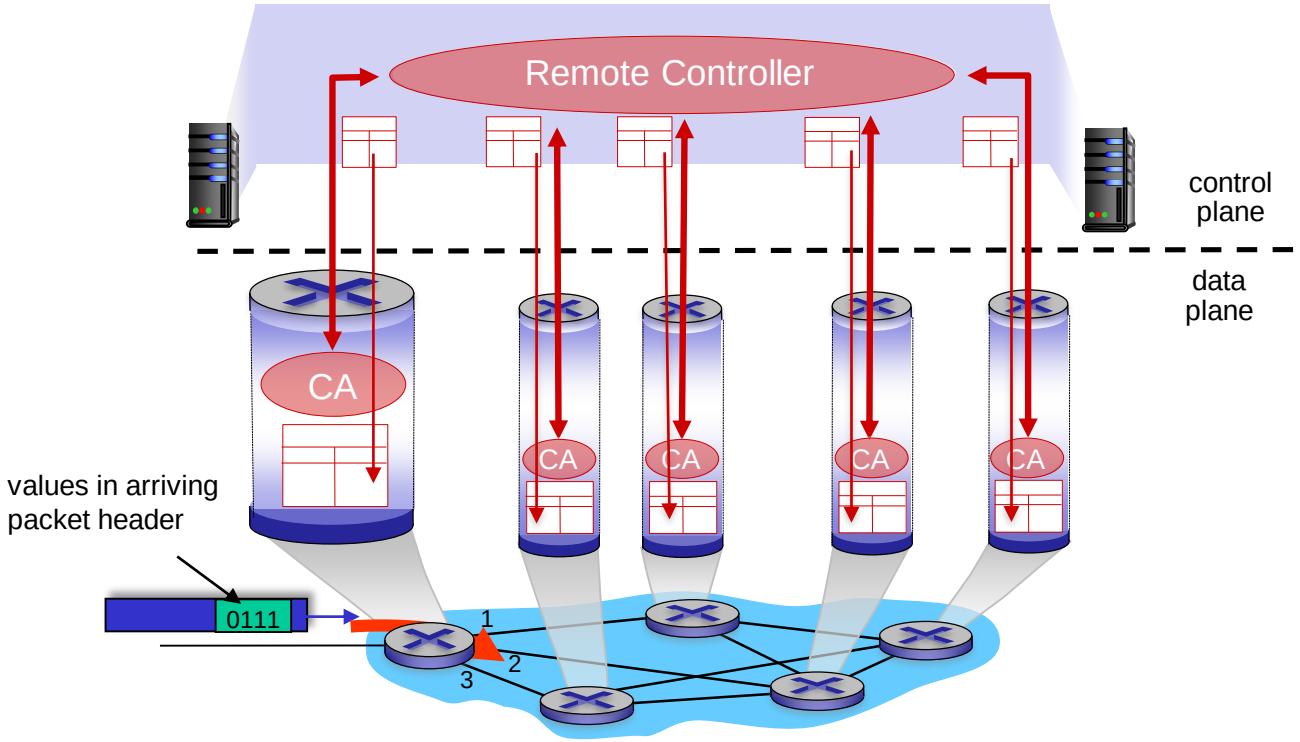
# Per-router control plane

Individual routing algorithm components *in each and every router* interact in the control plane



# Software-Defined Networking (SDN) control plane

Remote controller computes, installs forwarding tables in routers



# Network Service Model

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

- Network service model
  - Service model for "channel" transporting packets from sender to receiver
  - Called Quality of Service from host perspective

Example services for individual packets

- Guaranteed delivery
- Guaranteed delivery with less than 40 msec delay

Example services for a flow of packets

- In-order packet delivery
- Guaranteed minimum bandwidth to flow
- Restrictions on changes in inter-packet spacing

# Example：Network Service Model of ATM

In decreasing priority

- Constant Bit Rate (CBR) and Variable Bit Rate (VBR)
- Available Bit Rate (ABR) and Unspecified Bit Rate (UBR)

| Network Architecture | Service Model | Guarantees ? | | | | Congestion feedback |
|---|---|---|---|---|---|---|
| | | Bandwidth | Loss | Order | Timing | |
| Internet | best effort | none | no | no | no | no (inferred via loss) |
| ATM | CBR | constant rate | yes | yes | yes | no congestion |
| ATM | VBR | guaranteed rate | yes | yes | yes | no congestion |
| ATM | ABR | guaranteed minimum | no | yes | no | yes |
| ATM | UBR | none | no | yes | no | no |

# Example：Network Service Model of IP

- Best effort

| Network Architecture | Service Model | Bandwidth Guarantee | No-Loss Guarantee | Ordering | Timing | Congestion Indication |
|---|---|---|---|---|---|---|
| Internet | Best Effort | None | None | Any order possible | Not maintained | None |
| ATM | CBR | Guaranteed constant rate | Yes | In order | Maintained | Congestion will not occur |
| ATM | ABR | Guaranteed minimum | None | In order | Not maintained | Congestion indication provided |


# IP Routers

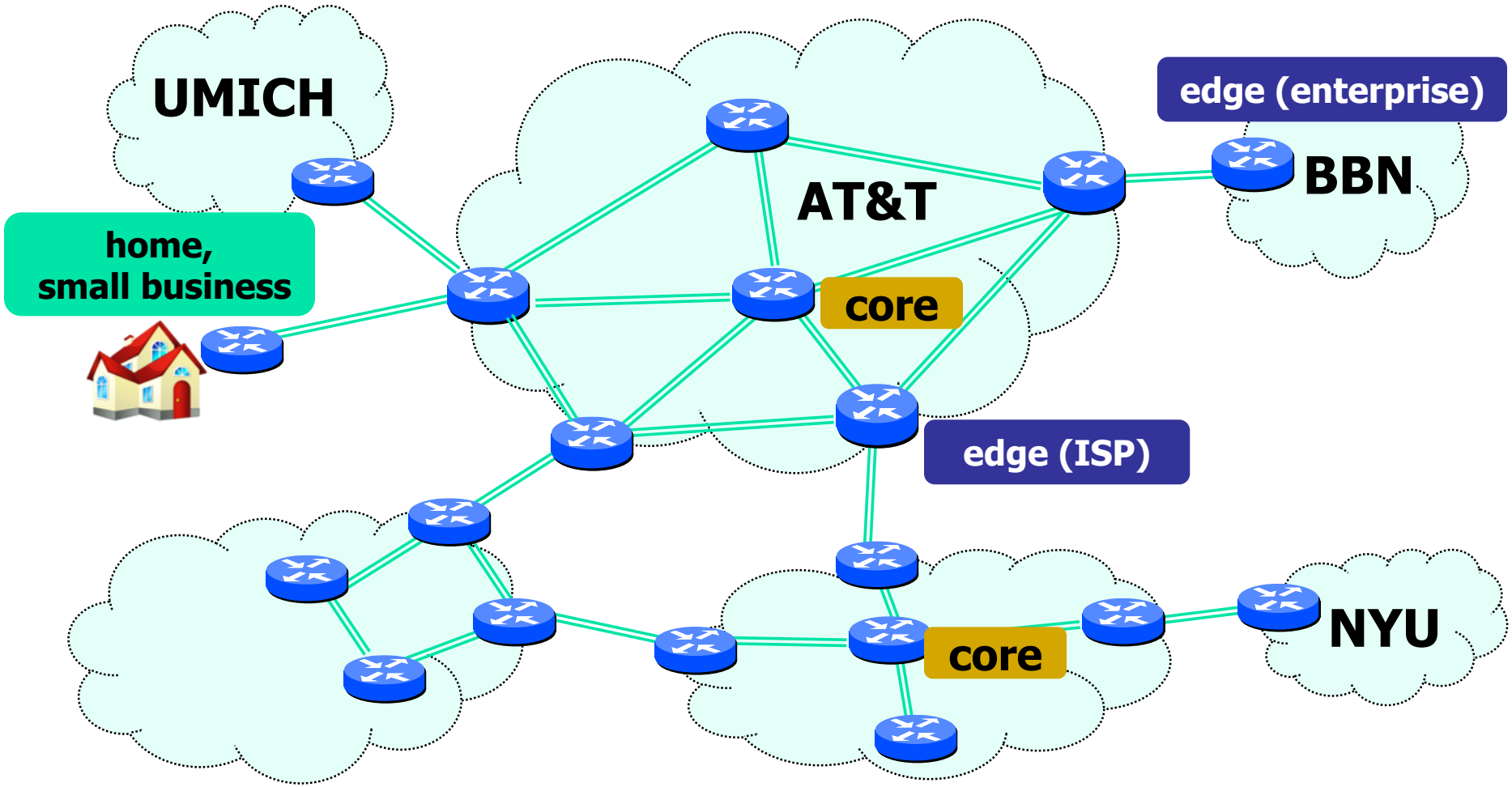

# IP routers

- Core building block of the Internet infrastructure
- $120B+ industry
- Vendors: Cisco, Huawei, Juniper, Alcatel-Lucent (account for >90%)

# Router definitions

- Router capacity = N x R
- N = Number of external router "ports"
- R = Speed ("line rate") of a port

# Networks and routers



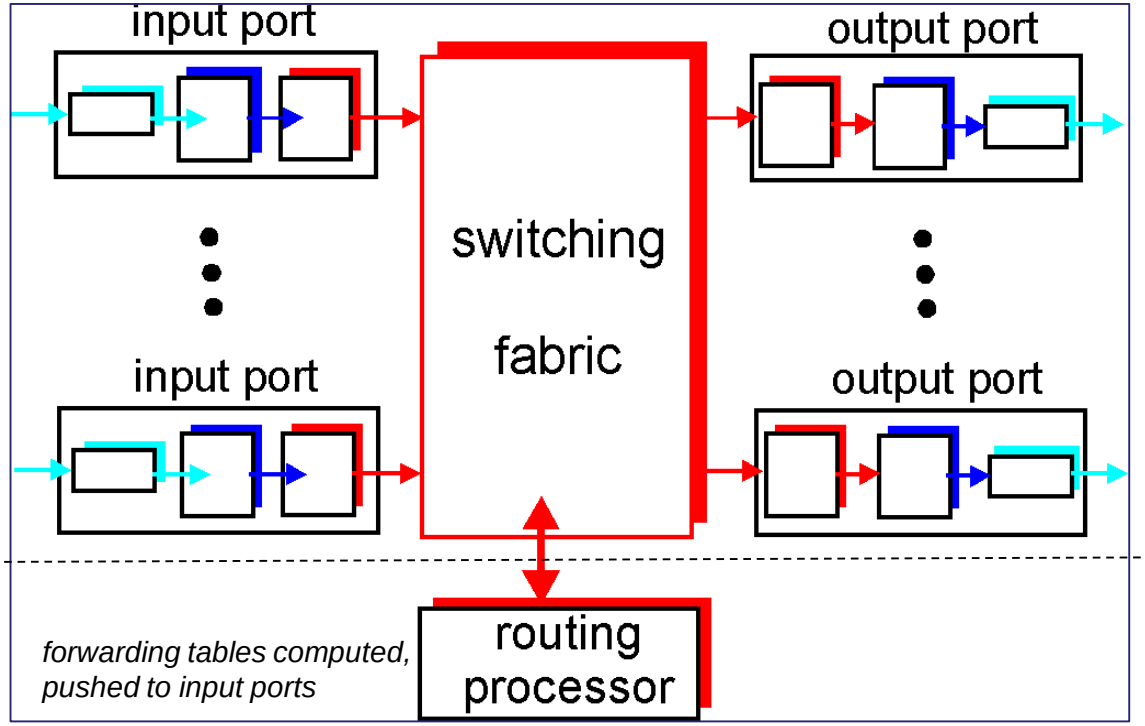
# Many types of routers

- Core
  - R = 10/40/100/200/400 Gbps
  - NR = O(100) Tbps (Aggregated)
- Edge
  - R = 1/10/40/100 Gbps
  - NR = O(100) Gbps
- Small business
  - R = 1 Gbps
  - NR < 10 Gbps

# Inside a Router: Architecture Overview

Two key switch functions:

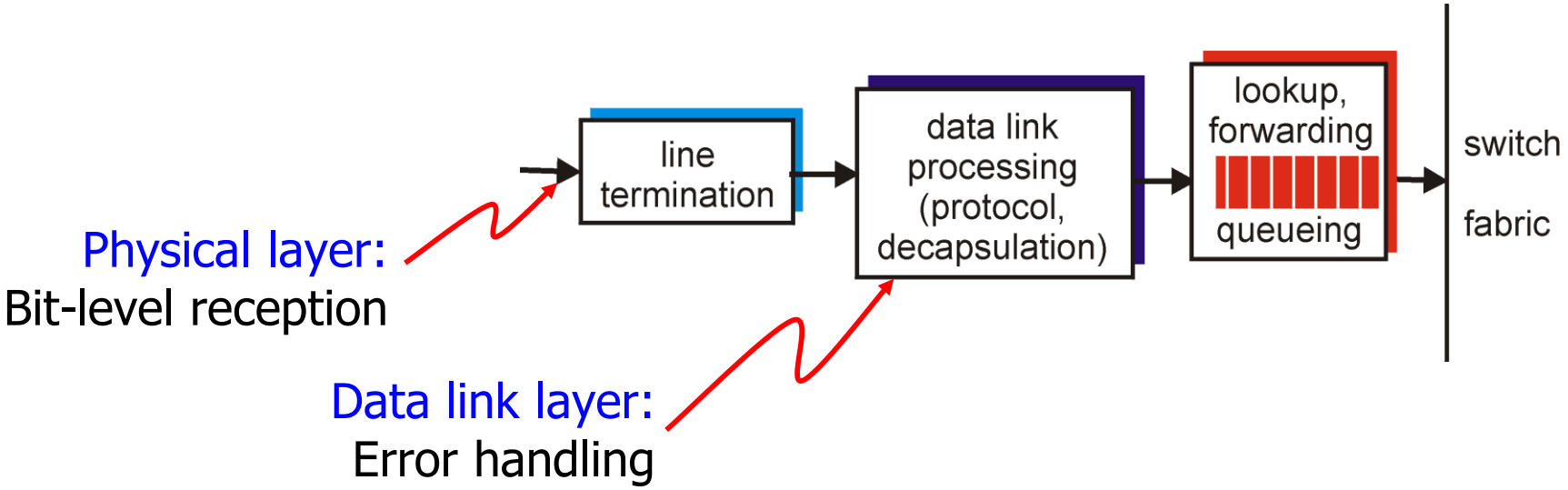
- Run routing algorithms/protocol
- Forwarding packets from incoming to outgoing link




forwarding data plane (hardware)

routing, management control plane (hardware&software)

# Input Port Functions




## Tasks

- Receive incoming packets (physical layer stuff)
- Update the IP header
  - TTL, Checksum, Options and Fragment (maybe)
- Lookup the output port for the destination IP address
- Queuing: if packets arrive faster than forwarding rate into switch fabric


# Input Port

- Challenge: speed!
  - 100B packets @ 40Gbps → new packet every 20 nano secs!
  - Typically implemented with specialized ASICs (network processors)


# Looking up the output port

- One entry for each address → 4 billion entries!
- For scalability, addresses are aggregated


# Example

- Router with 4 ports
- Destination address range mapping
  - 11 00 00 00 to 11 00 00 11: Port 1
  - 11 00 01 00 to 11 00 01 11: Port 2
  - 11 00 10 00 to 11 00 11 11: Port 3
  - 11 01 00 00 to 11 01 11 11: Port 4

# Example

- Router with 4 ports

| Destination Address Range | Link interface |
|---|---|
| 11001000 00010111 00010*** ******** | 0 |
| 11001000 00010111 00011000 ******** | 1 |
| 11001000 00010111 00011*** ******** | 2 |
| otherwise | 3 |

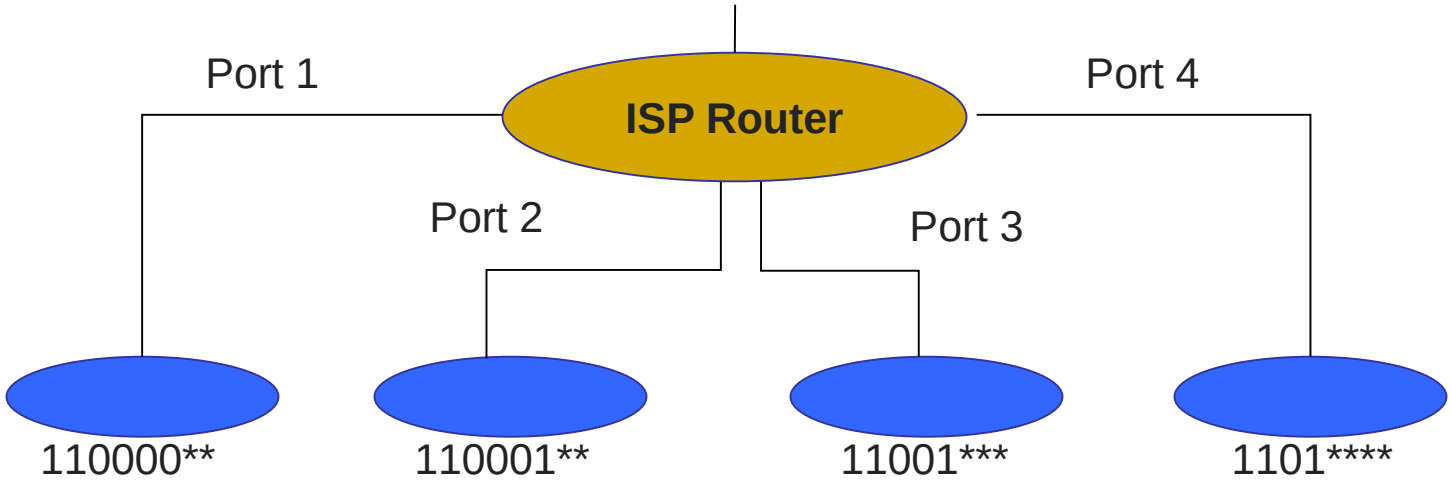
**Longest prefix matching rule:** when looking for forwarding table entry for given destination address, use longest address prefix that matches destination address.

**examples:**

11001000 00010111 00010110 10100001 **which interface?**

11001000 00010111 00011000 10101010 **which interface?**

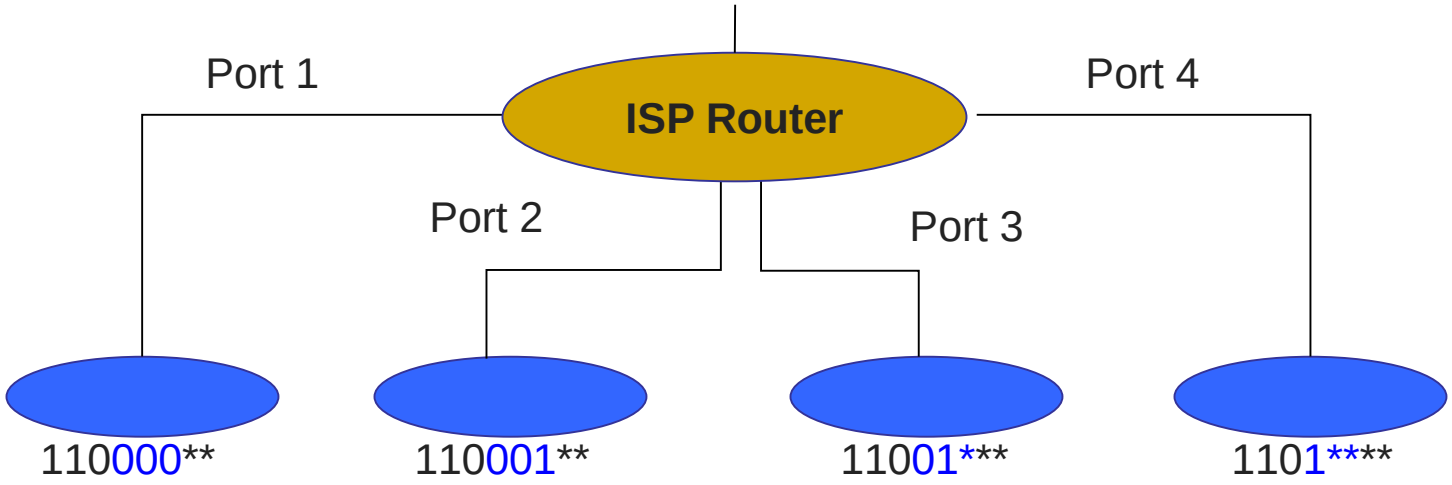
# Longest prefix matching




# Finding match efficiently

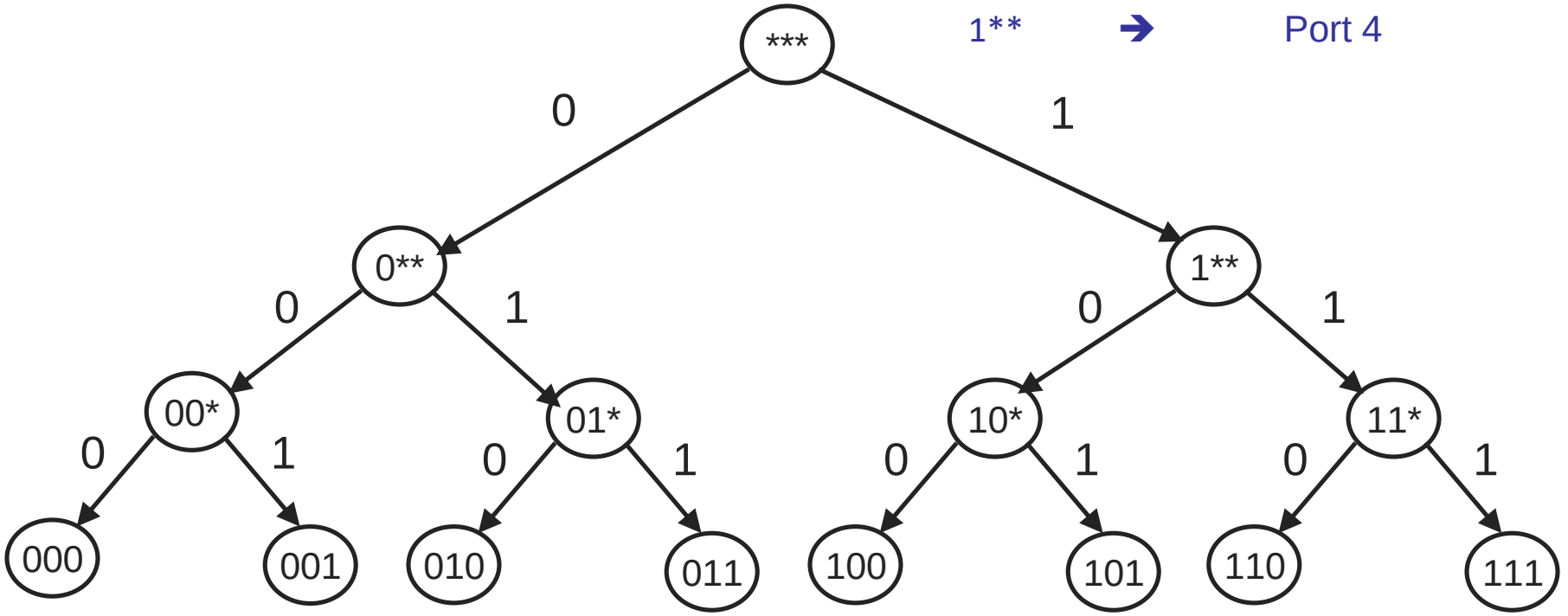
- Testing each entry to find a match scales poorly
  - On average: O(number of entries)
- Leverage tree structure of binary strings
  - Set up tree-like data structure

# Longest prefix matching



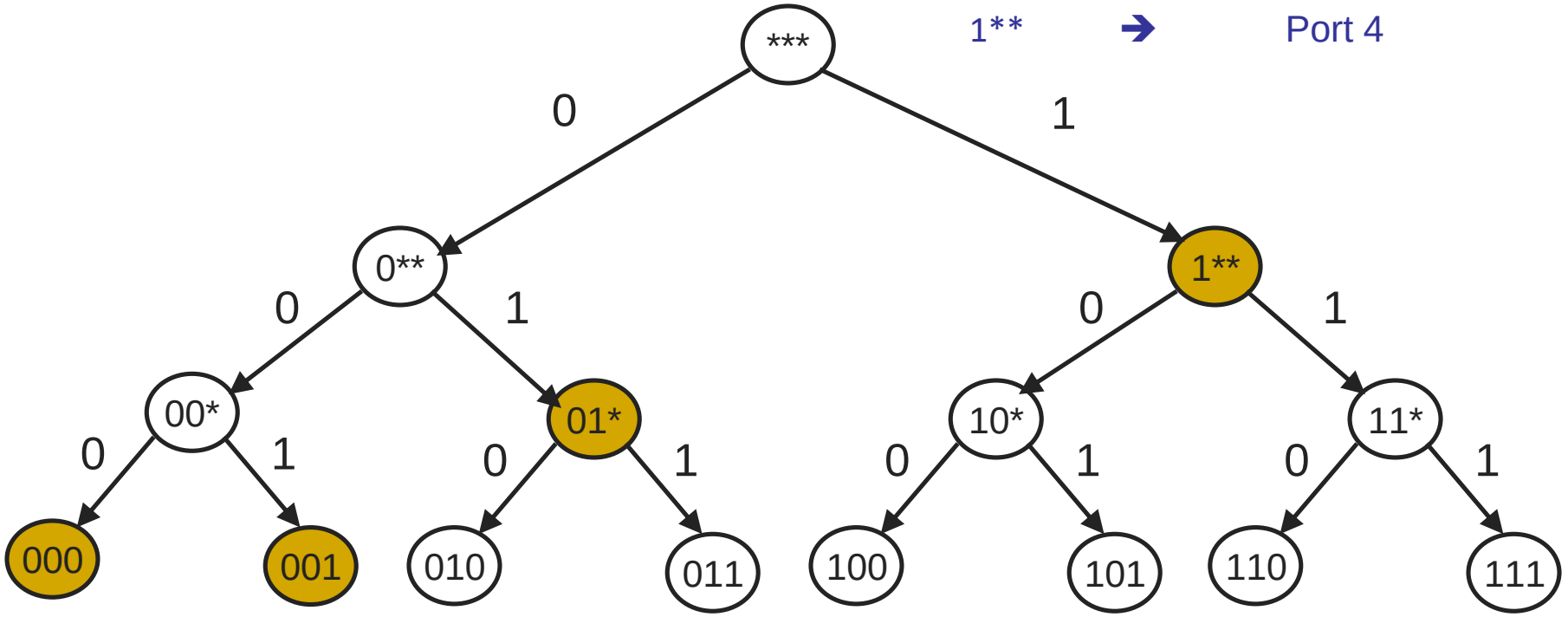
# Tree structure

000 → Port 1
001 → Port 2
01* → Port 3
1** → Port 4



# Tree structure

000 ➔ Port 1
001 ➔ Port 2
01* ➔ Port 3
1** ➔ Port 4




Record port associated with latest match, and only override when it matches another prefix during walk down tree

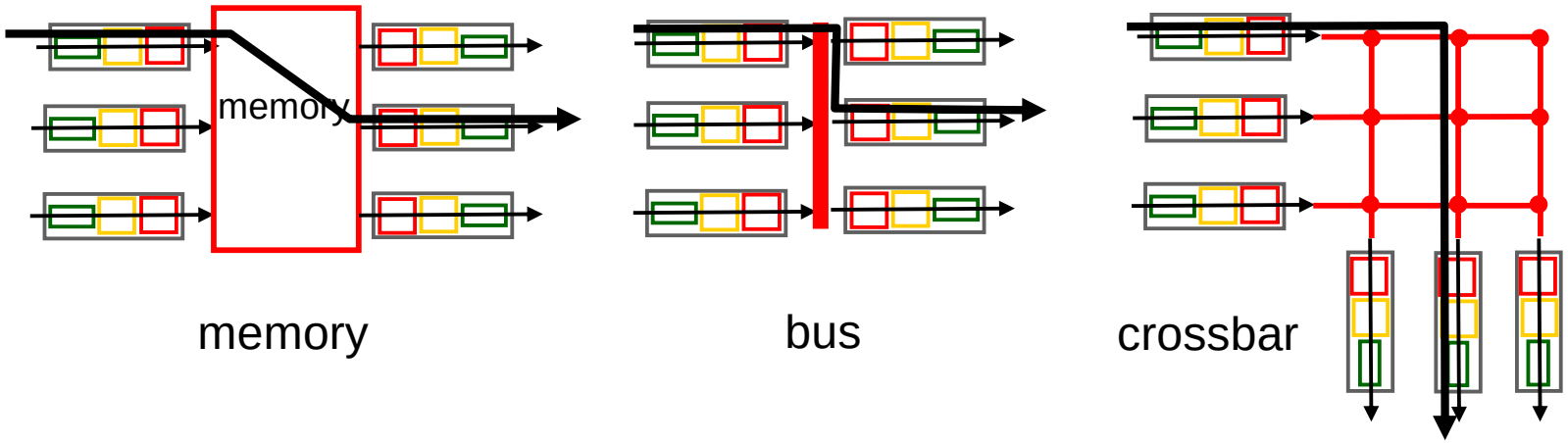

# Input Port

- Main challenge is processing speeds
- Tasks involved:
  - Update packet header (easy)
  - LPM lookup on destination address (harder)
- Mostly implemented with specialized hardware

# Connecting inputs to outputs: Switching fabric

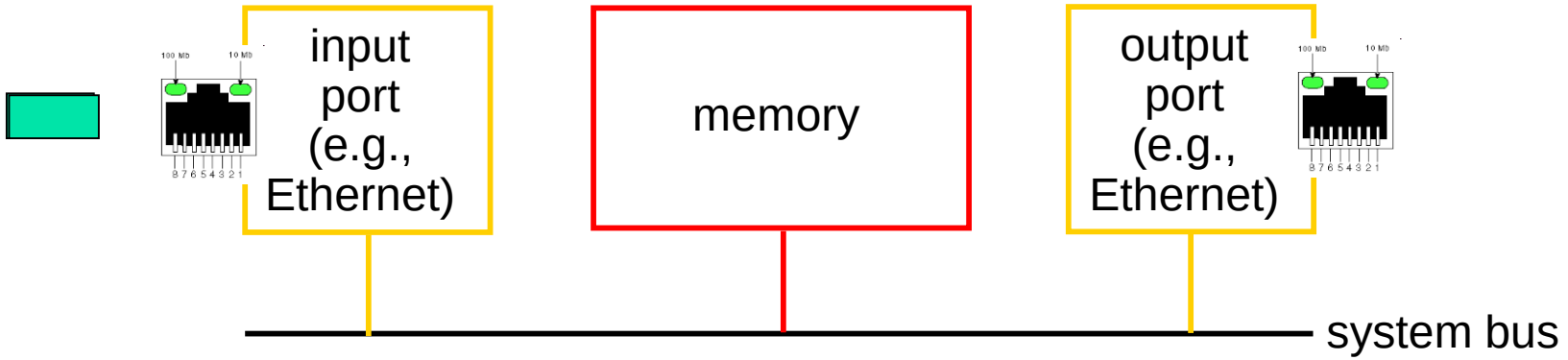
- Connecting inputs to outputs: Switching fabric
- Transfer packet from input buffer to appropriate output buffer
- Switching rate: rate at which packets can be transfer from inputs to outputs
  - often measured as multiple of input/output line rate
  - N inputs: switching rate N times line rate desirable
- Three types of switching fabrics

交换结构



# Switching via Memory

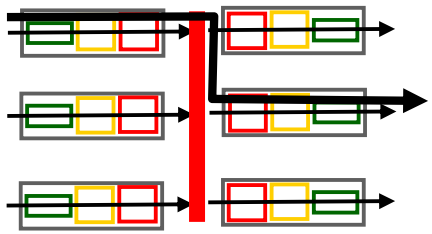
- First generation routers:
- Traditional computers with switching under direct control of CPU
- Packet copied to system's memory
- Speed limited by memory bandwidth (2 bus crossings per datagram)



# Switching via a Bus

- Datagram from input port memory

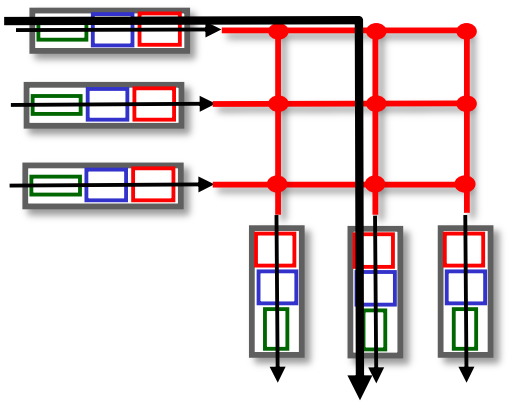
  to output port memory via a shared bus
- *Bus contention:* switching speed limited by bus bandwidth
- 32 Gbps bus, Cisco 5600: sufficient speed for access and enterprise routers



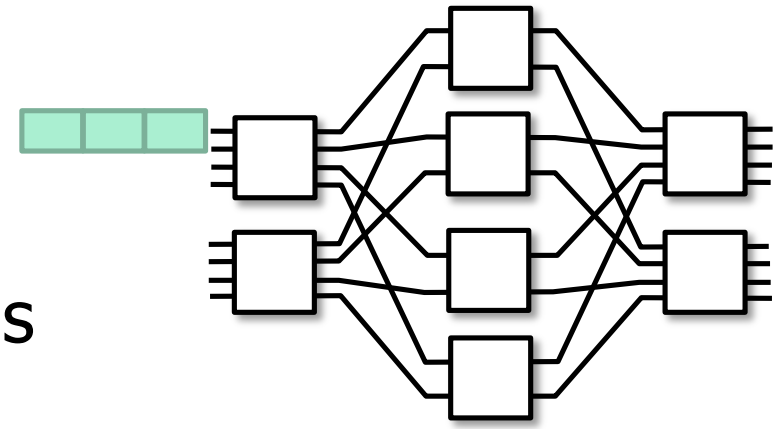
bus

# Switching via a Mesh

- Overcome bus bandwidth limitations
- Crossbar and other interconnection nets initially developed to connect processors in multiprocessor
- Multistage switch: nxn switch from multiple stages of smaller switches
- Exploiting parallelism: fragmenting datagram into fixed length cells, switch cells through the fabric.

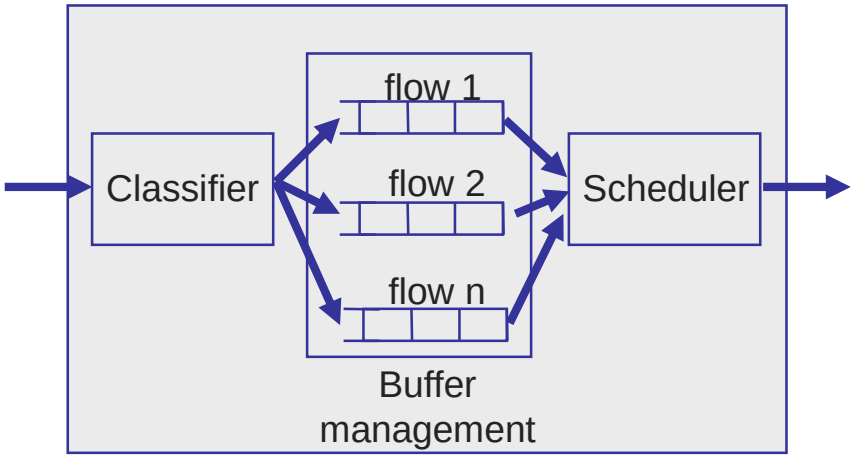


3x3 crossbar



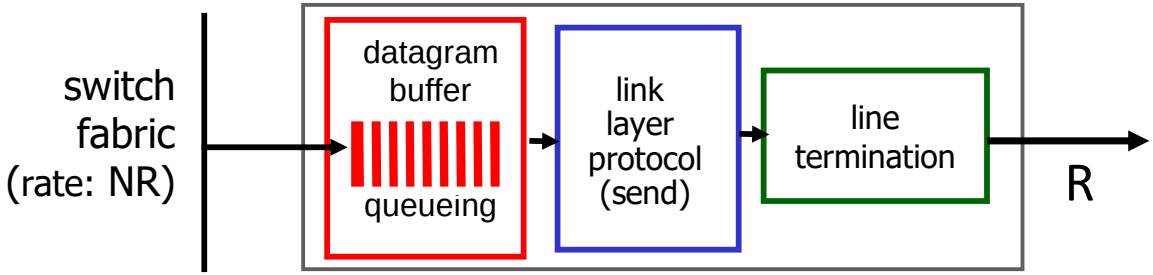
8x8 multistage switch
built from smaller-sized switches

# Output Port Functions




- Packet classification: map packets to flows
- Buffer management: decide when and which packet to drop
- Scheduler: decide when and which packet to transmit
  - Chooses among queued packets for transmission
  - Select packets to drop when buffer saturates

# Output port queuing




- *Buffering* required when datagrams arrive from fabric faster than link transmission rate. *Drop policy:* which datagrams to drop if no free buffers?

→ Datagrams can be lost due to congestion, lack of buffers

- *Scheduling discipline* chooses among queued datagrams for transmission

→ Priority scheduling – who gets best performance, network neutrality

# How much buffering?

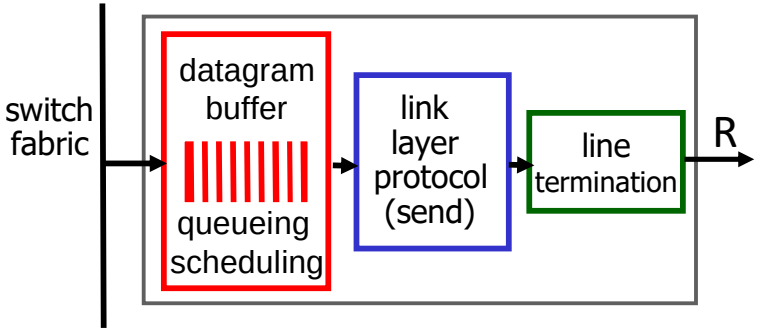
- RFC 3439 rule of thumb: average buffering equal to "typical" RTT (say 250 msec) times link capacity C
  - e.g., C = 10 Gbps link: 2.5 Gbit buffer

- but *too* much buffering can increase delays (particularly in home routers)
  - long RTTs: poor performance for real-time apps, sluggish TCP response

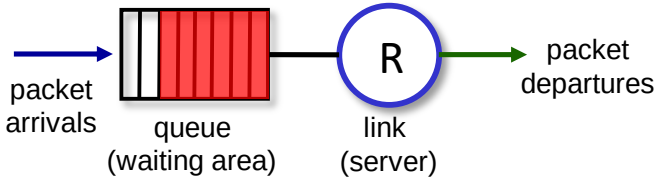
- more recent recommendation: with *N* flows, buffering equal to

$$\frac{RTT \cdot C}{\sqrt{N}}$$

# Buffer Management




Abstraction: queue




buffer management:

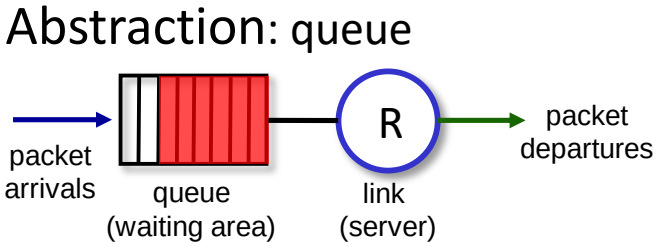
- drop: which packet to add, drop when buffers are full
  - tail drop: drop arriving packet
  - priority: drop/remove on priority basis

- marking: which packets to mark to signal congestion (ECN, RED)

# Packet Scheduling: FCFS

packet scheduling: deciding which packet to send next on link

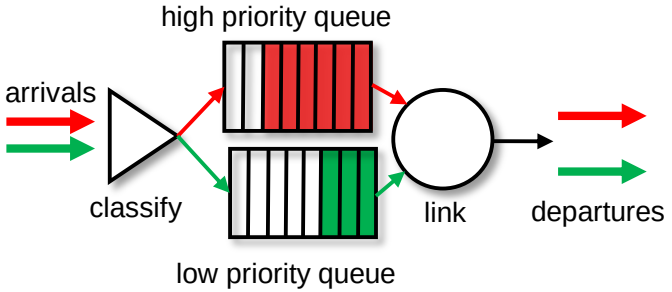
- first come, first served (FCFS)
- priority
- round robin
- weighted fair queueing

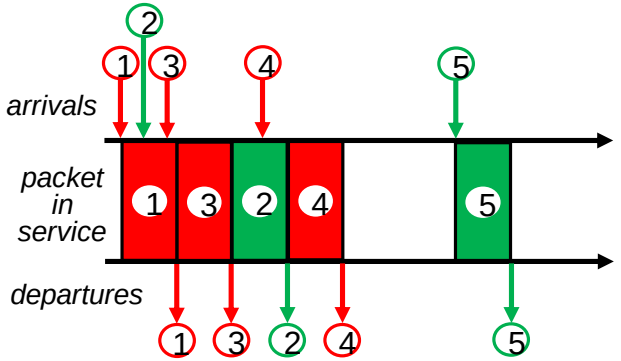


# Scheduling policies: priority

*Priority scheduling:*

- arriving traffic classified, queued by class
  - any header fields can be used for classification
- send packet from highest priority queue that has buffered packets
  - FCFS within priority class



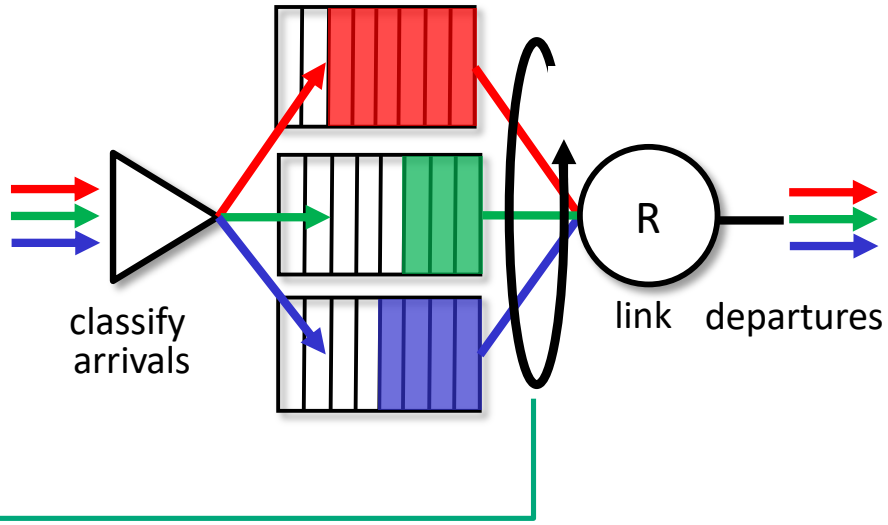



# Scheduling policies: round robin

*Round Robin (RR) scheduling:*

- arriving traffic classified, queued by class
  - any header fields can be used for classification

- server cyclically, repeatedly scans class queues, sending one complete packet from each class (if available) in turn



# Scheduling policies: WFQ

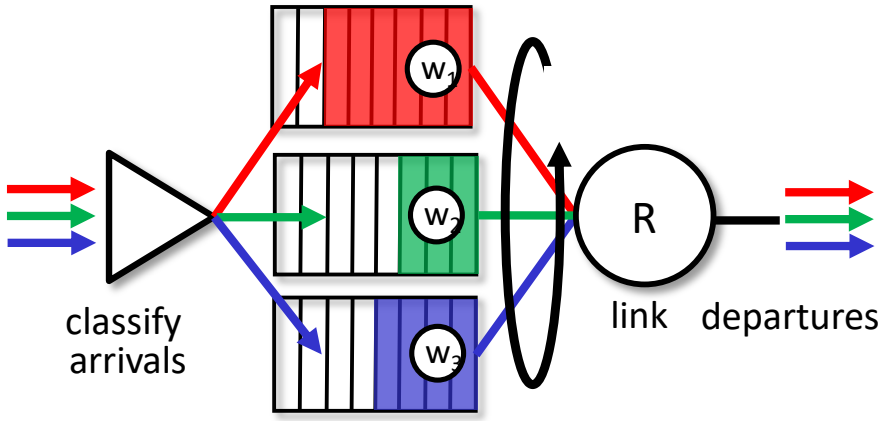
*Weighted Fair Queuing (WFQ):*

- generalized Round Robin

- each class, *i*, has weight, $w_i$, and gets weighted amount of service in each cycle:

$$\frac{w_i}{\sum_j w_j}$$

- minimum bandwidth guarantee (per-traffic-class)




# Virtual Circuit and Datagram Networks

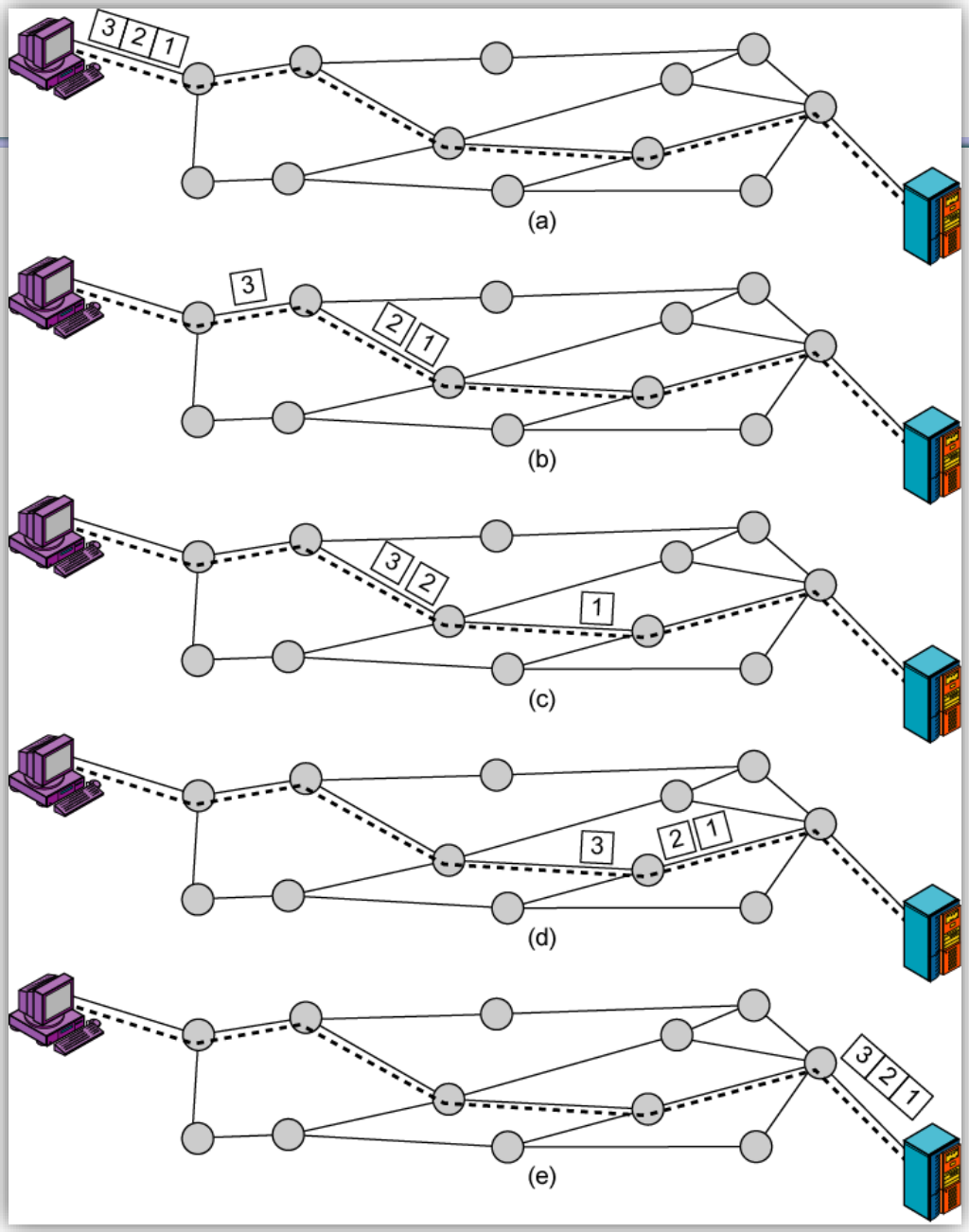
# Recap: Circuit Switching & Packet Switching

- Circuit Switching
  - End-to-end resources reserved for "call"
    - Link bandwidth, switch capacity
  - Dedicated resources: no sharing
  - Guaranteed performance
  - Call setup/teardown required
- Packet Switching
  - Each end-to-end data stream divided into packets
  - Application A, B packets share network resources
  - Store and forward: packets move one hop at a time, stored (queued) at switches
  - Resource contention: aggregate (burst-up) resource demand can exceed amount available
  - Congestion: packets queue and wait for link use

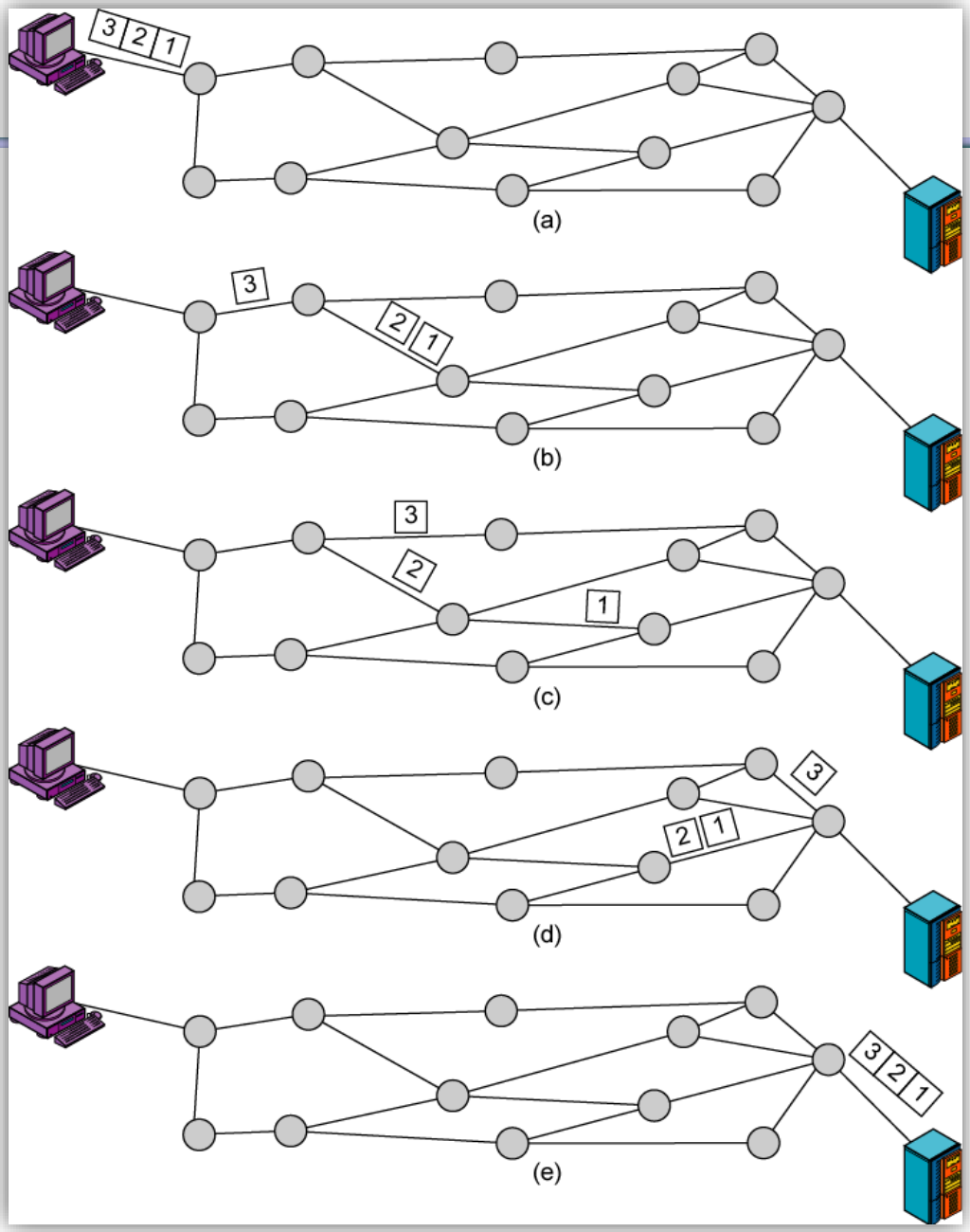
# Virtual Circuit and Datagram Networks

- Two types of Package Switch Networks
  - Virtual circuit networks
    - Network service provided on flow of packets
    - VC network provides network-layer connection oriented service
    - E.g., ATM, X.25, Frame Relay
  - Datagram networks
    - Network service provided on singular packet
    - Datagram network provides network-layer connectionless service
    - E.g., IP network

# Routing in Virtual Circuit



# Routing in Datagram Nets




# Virtual Circuit Networks

- Connection setup, teardown for each flow of packets
- Each packet carries VC identifier (not destination host address)

- Every switch on source-destination path maintains "state" for each passing connection
- Link, switch resources (bandwidth, buffers) may be allocated to VC
  - Dedicated resources = predictable quality of service


# Connection Setup

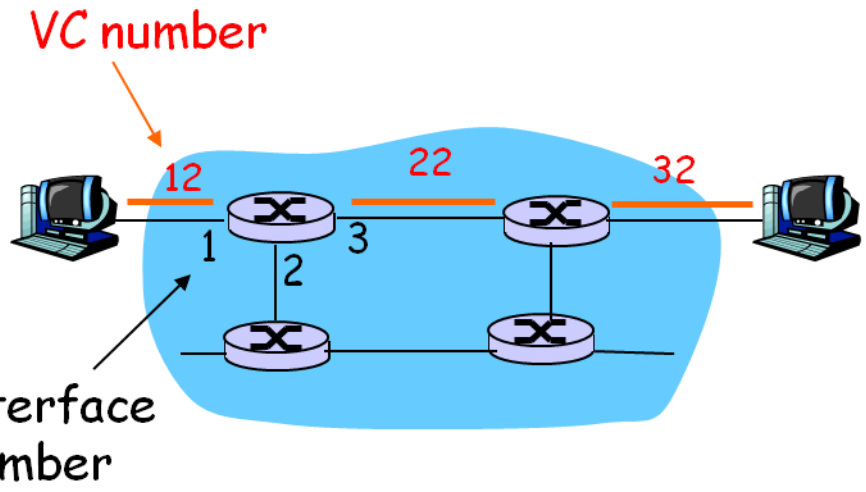
- Essential function for virtual circuit networks
  - E.g. ATM, frame relay, X.25
- Two end hosts and intervening switches pre-establish a path for virtual connection
- Routing is used for finding a suitable (shortest) path

# VC Implementation

- A VC consists of
  - Path from source to destination
  - VC numbers, maybe one number for each link along the path
  - Entries in forwarding tables in switches along the path

- Note:
  - Packet belonging to VC carries VC number (rather than addresses)
  - VC number can be changed on each link, forwarding table lists the new VC number

# A Forwarding Table for VC




Forwarding table in northwest switch

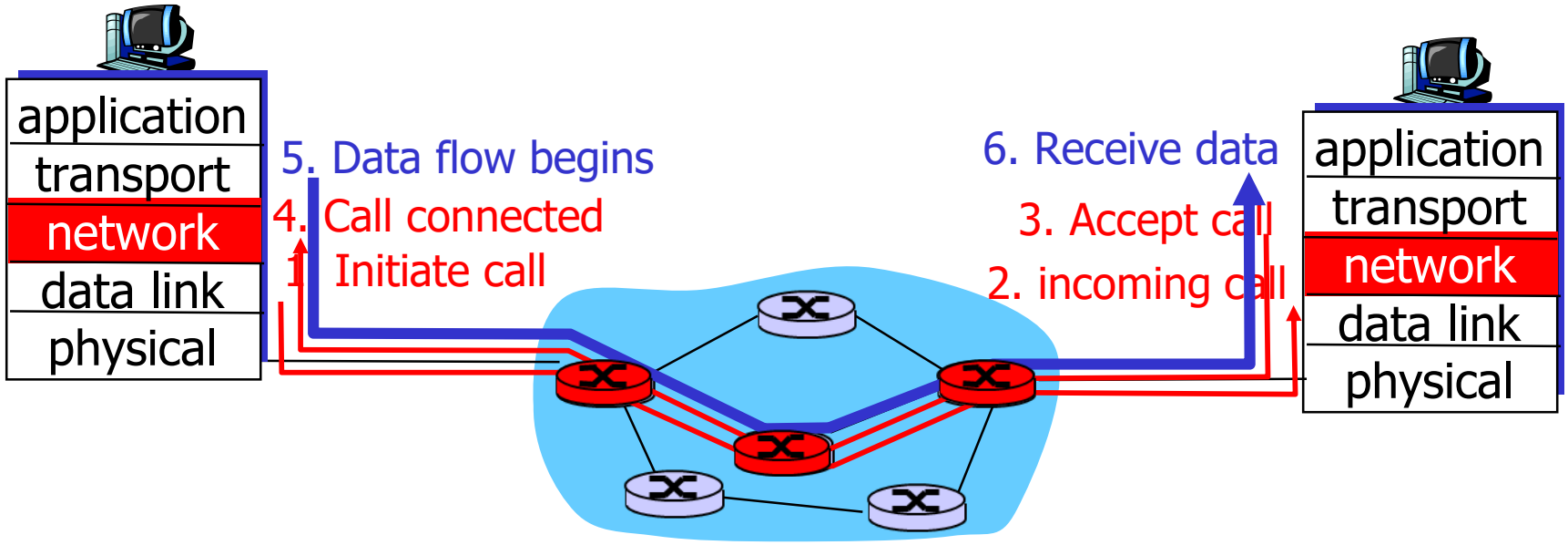
| Incoming interface | Incoming VC # | Outgoing interface | Outgoing VC # |
|---|---|---|---|
| 1 | 12 | 3 | 22 |
| 2 | 63 | 1 | 18 |
| 3 | 7 | 2 | 17 |
| 1 | 97 | 3 | 87 |
| … | … | … | … |

Table entries constitutes state information of a VC

# Virtual Circuits: Signaling Protocols

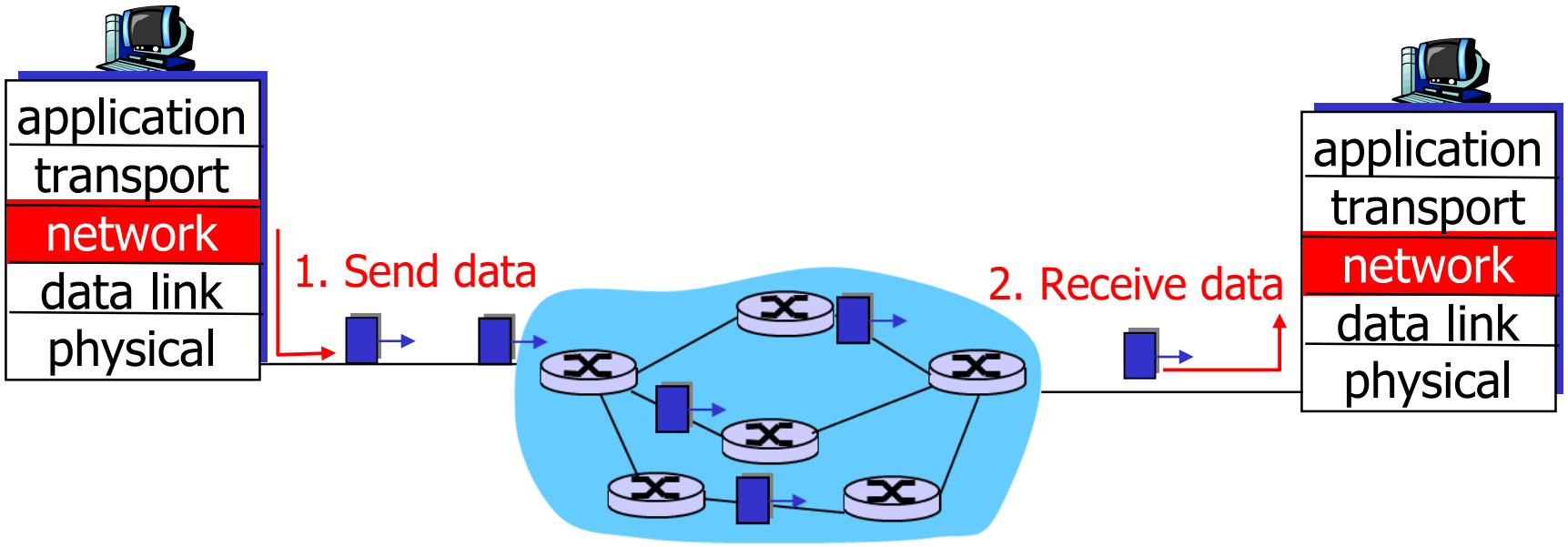
- Used to setup, maintain and teardown VC
- Used in ATM, frame-relay, X.25
- Not used in today's Internet

信令协议



# Datagram Networks

- No call setup at network layer
- No network-level concept of "connection"
- Switches: no state about end-to-end connections
- Packets forwarded using destination host address
- Packets between same source-dest pair may take different paths



# A Forwarding Table for Datagram Networks

- Also called routing table
- May reach 4 billion entries

- The destination address prefix may define a switch address or a subnet address

| Dest Address Prefix | Address Mask | Link Interface |
|---|---|---|
| 11001000 00010111 00010 | 11111111 11111111 11111000 00000000 | 0 |
| 11001000 00010111 00011000 | 11111111 11111111 11111111 00000000 | 1 |
| 11001000 00010111 000110 | 11111111 11111111 11111100 00000000 | 2 |
| default | * | 3 |

# Datagram vs. Virtual Circuit

## Datagram (Internet)

- Data exchange among computers
  - "Elastic" service, no strict timing
- "Smart" end systems (computers)
  - Can adapt, perform control, error recovery
  - Simple inside network, complexity at "edge"
- Many link types
  - Different characteristics
  - Uniform service difficult

## Virtual Circuit (ATM)

- Evolved from telephony
- Human conversation:
  - Strict timing, reliability requirements
  - Need guaranteed service
- "Dumb" end systems
  - Telephones
  - Complexity inside network (switches)
- Link type standardized

# Summary

- 网络层基本功能
  - 交换/路由，转发，建立连接
- 路由器的构成
- 两种分组交换网络
  - 虚电路网络
  - 数据报网络
    - IP网络