# ECE 158A: Lecture 7

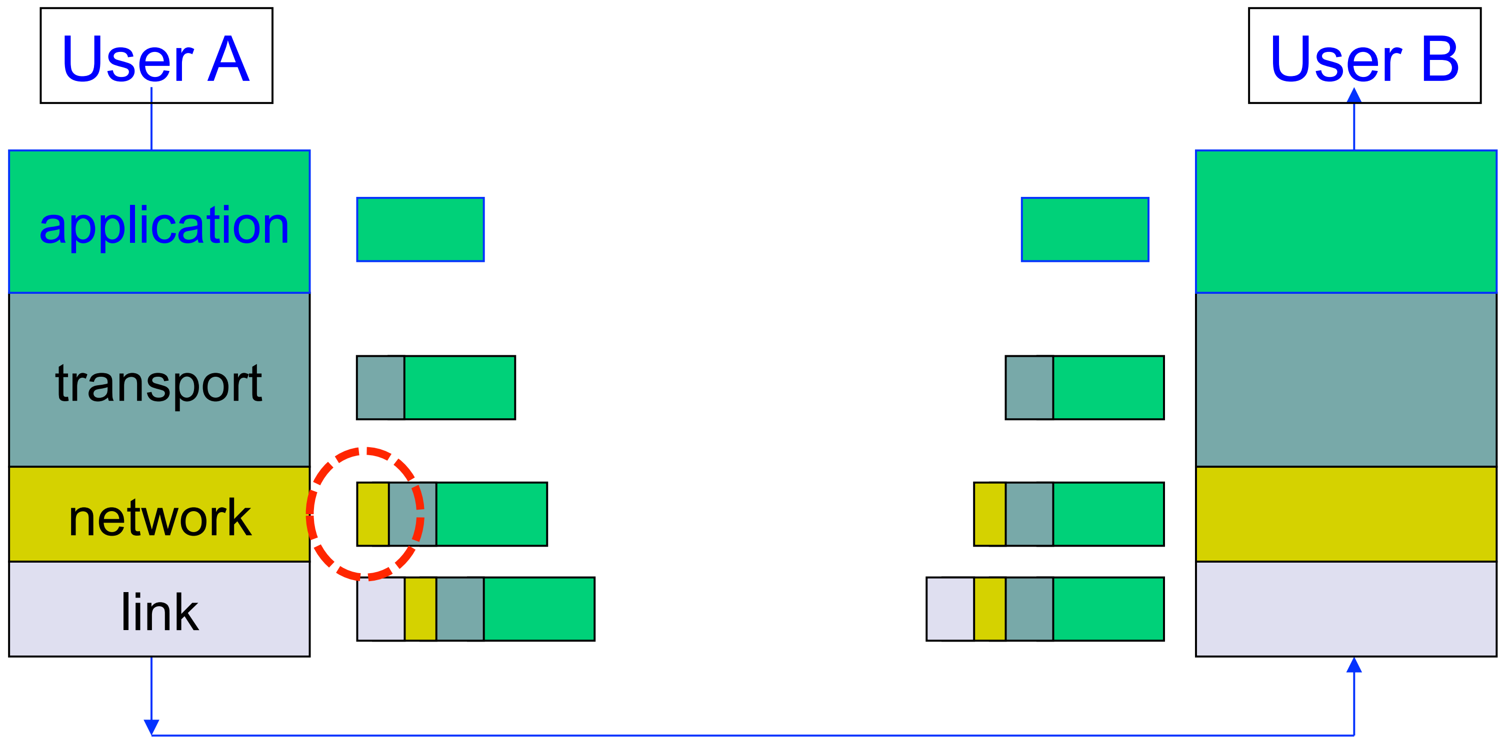
Fall 2015

# Outline

- We have discussed IP shortest path routing
- Now we have a closer look at the IP addressing mechanism
- We are still at the networking layer, we will examine:
  - IP Headers
  - IP Addressing
  - Routers

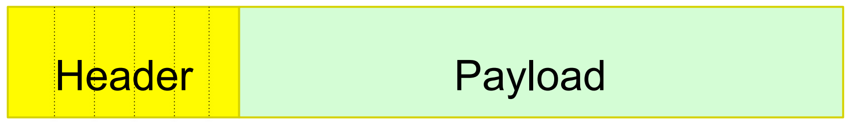
# Encapsulation

- IP, like any other protocol, adds control information as a packet header



# IP Header

- IP packet is divided in header and payload




- Payload contains TCP/UDP header

- IP routers only read the IP packet header

- IP header must contain all information needed to perform tasks of routing layer

# Routing Layer Tasks

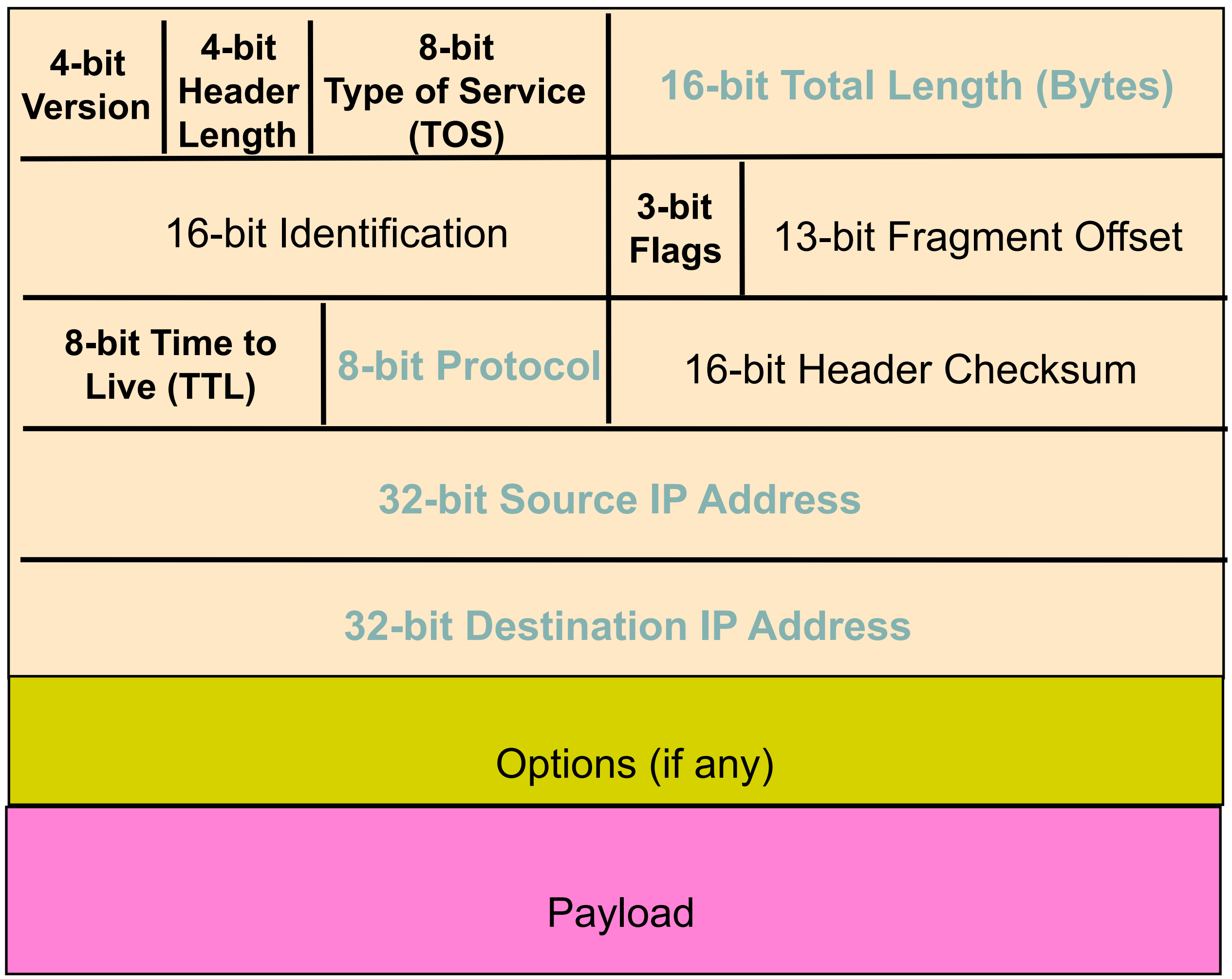
- Routing layer is responsible for
  - Reading packet correctly
  - Getting packet to the destination; getting responses back to the source
  - Carrying data
  - Telling host what to do with packet once arrived
  - Specifying any special network handling of the packet
  - Dealing with problems that arise along the path

- All the control information needed to achieve these tasks is contained in the IP header

# General philosophy

- Smart edge DUMB Core
- If something goes wrong don’t deal with it, just drop the packet!
- TPC will take care of it

# IPv4 Header Structure



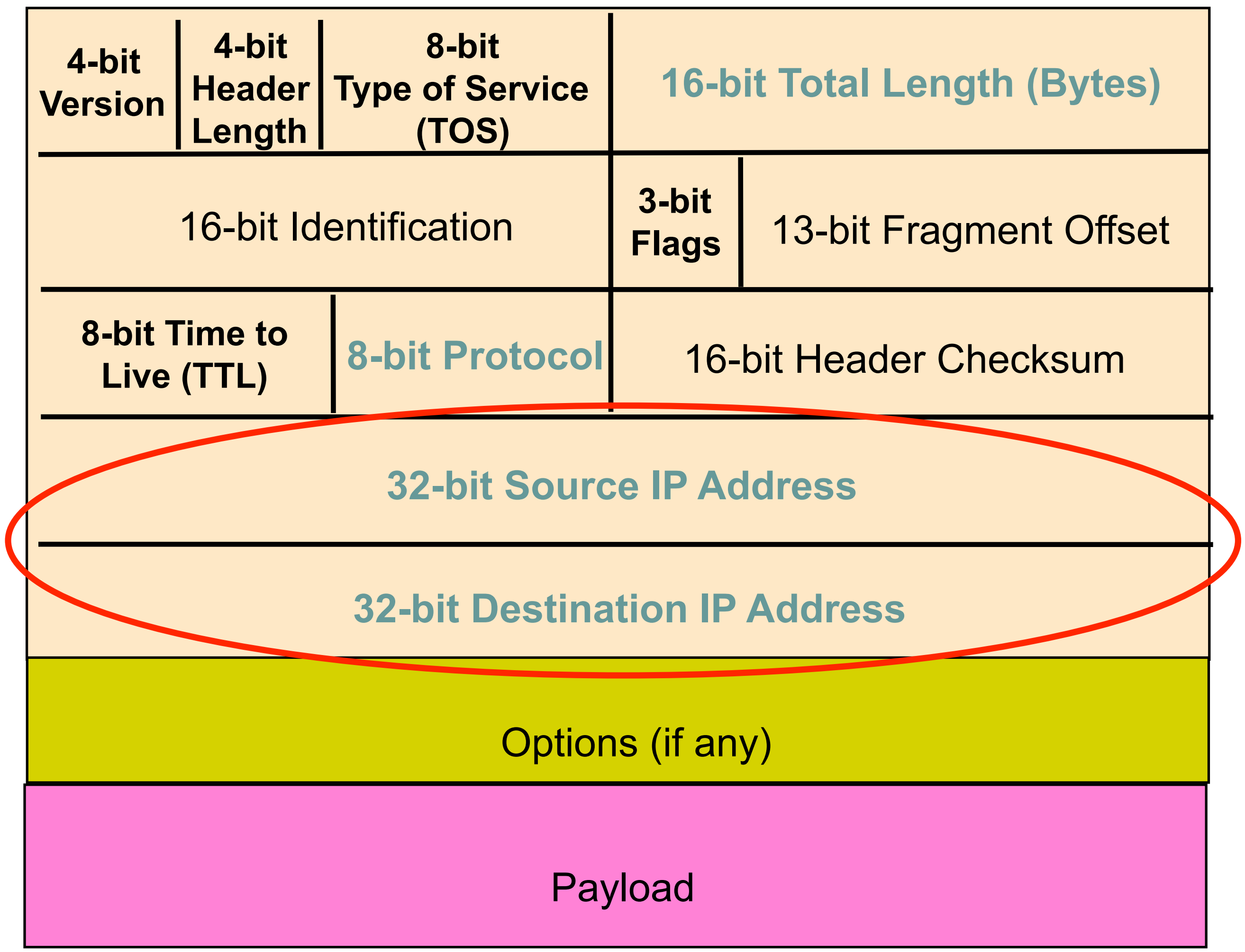
# Reading Packet Correctly

<table>
<tr><td>4-bit Version</td><td>4-bit Header Length</td><td>8-bit Type of Service (TOS)</td><td colspan="2">16-bit Total Length (Bytes)</td></tr>
<tr><td colspan="3">16-bit Identification</td><td>3-bit Flags</td><td>13-bit Fragment Offset</td></tr>
<tr><td colspan="2">8-bit Time to Live (TTL)</td><td>8-bit Protocol</td><td colspan="2">16-bit Header Checksum</td></tr>
<tr><td colspan="5">32-bit Source IP Address</td></tr>
<tr><td colspan="5">32-bit Destination IP Address</td></tr>
<tr><td colspan="5">Options (if any)</td></tr>
<tr><td colspan="5">Payload</td></tr>
</table>

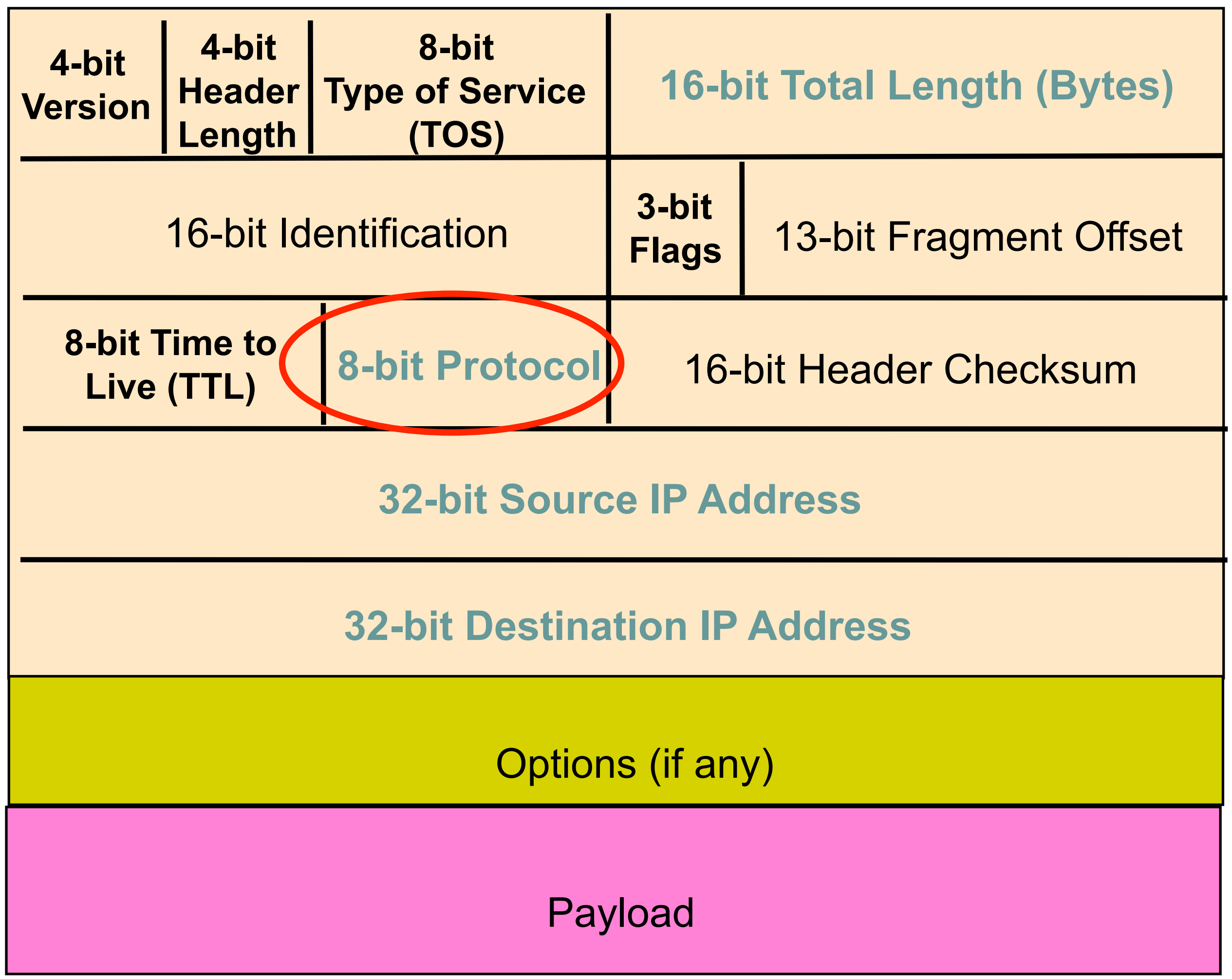
# Reading Packet Correctly

- Version number (4 bits)
  - Indicates the version of the IP protocol
  - Necessary to know what other fields to expect
  - Typically “4” (for IPv4), and sometimes “6” (for IPv6)
- Header length (4 bits)
  - Number of 32-bit words in the header
  - Typically “5” (for a 20-byte IPv4 header)
  - Can be more when IP options are used
- Total length (16 bits)
  - Number of bytes in the packet
  - Maximum size is 65,535 bytes
  - … though **underlying links may impose smaller limits**

# Get to the Destination (and Back)

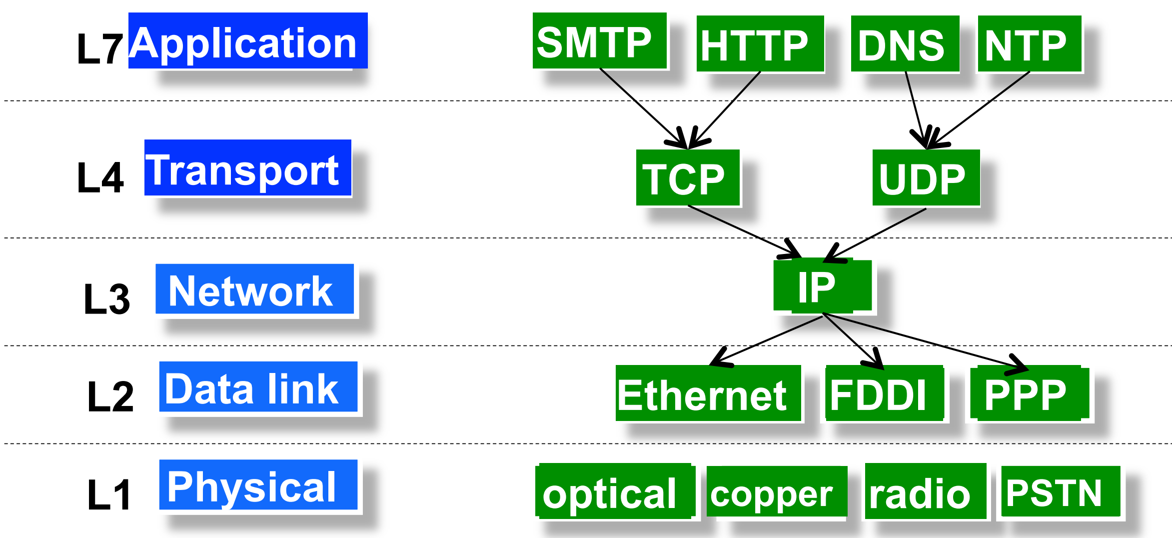


# How to Handle Packet



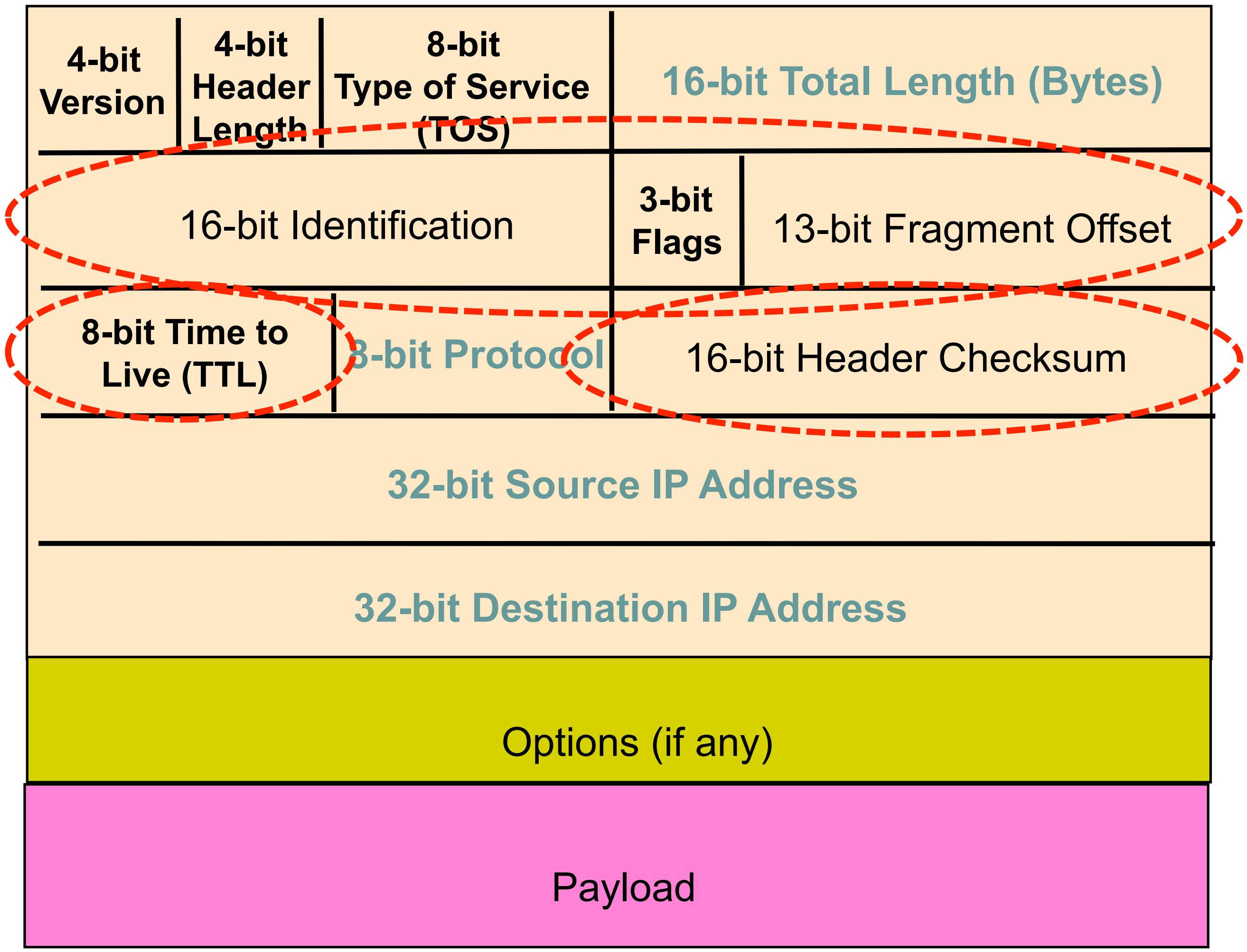
# How to Handle Packet

- Protocol (8 bits)
  - Identifies the higher-level protocol
  - Important for demultiplexing at receiving host




- Most common examples
  - E.g., “6” for the Transmission Control Protocol (TCP)
  - E.g., “17” for the User Datagram Protocol (UDP)

# Potential Problems



# Potential Problems

- Three potential problems

- Header Corrupted: Checksum, 16 bits
  - Checksum over packet header
  - If not correct, router discards packets
  - Recalculated at every router

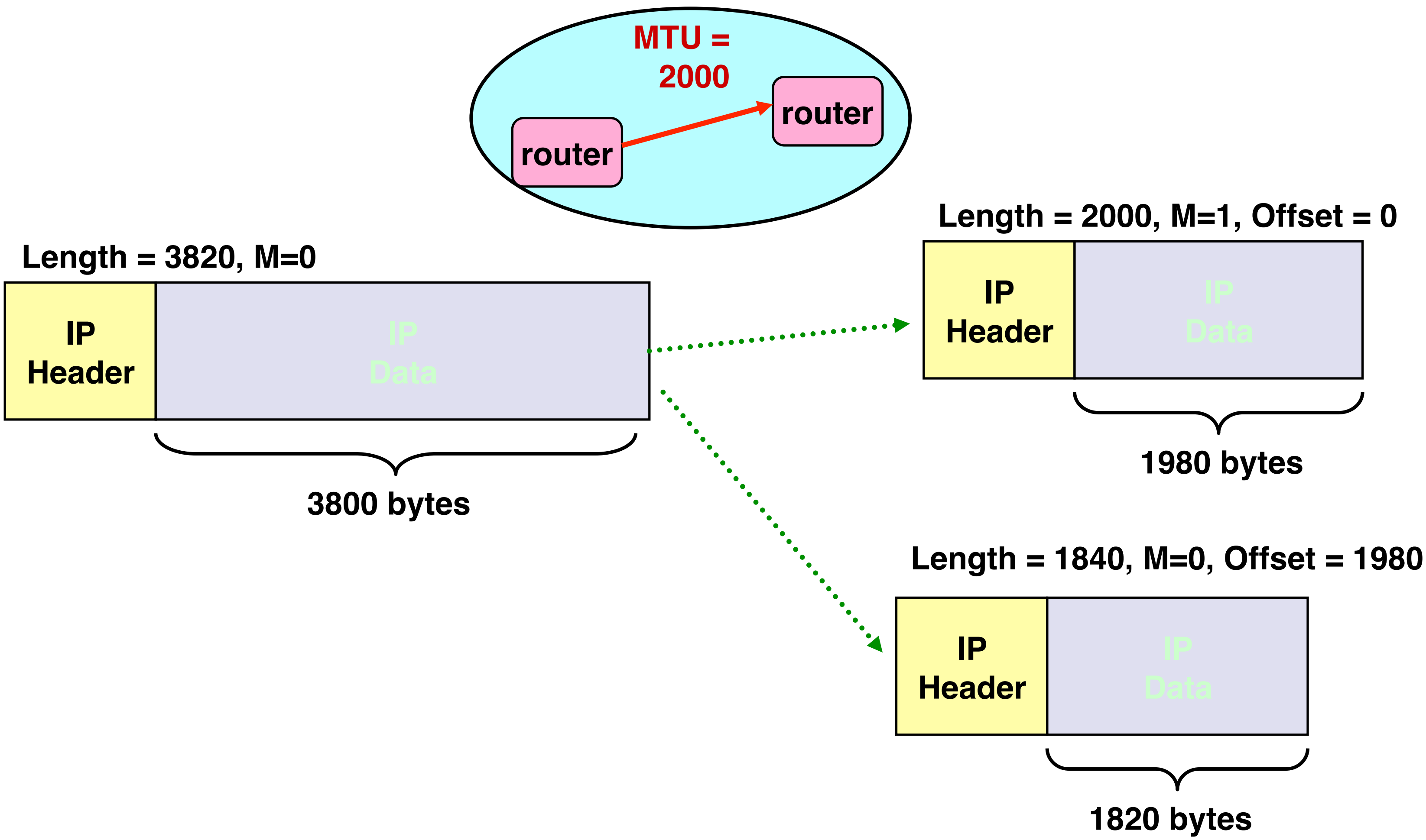
- Packet in a Loop: Time-to-Live (TTL), 8 bits
  - Decremented at each hop, packet discarded if reaches 0
  - …and "time exceeded" message is sent to the source
  - Used to avoid infinite loops

- Packet too large: Fragmentation

# Potential Problems

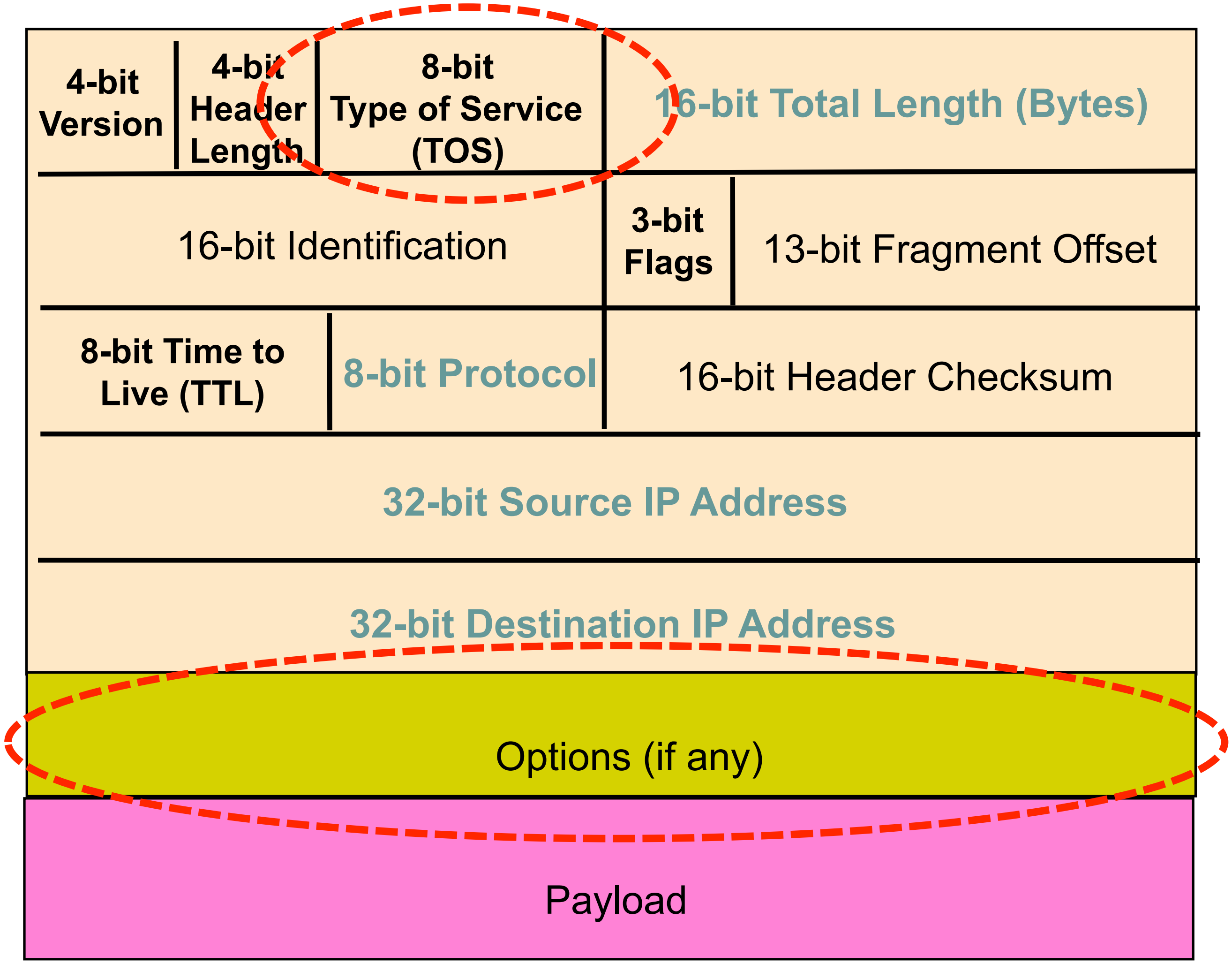
- When forwarding a packet, an Internet router can fragment it into multiple pieces (“fragments”) if the packet is too big for next hop link
  - too big = exceeds the link’s “Max Transmission Unit” (MTU)
- Must reassemble to recover original packet
  - Need fragmentation information (32 bits)
  - Packet identifier, flags, and fragment offset

# IP Fragmentation: Example



# Special Handling

- Not often used



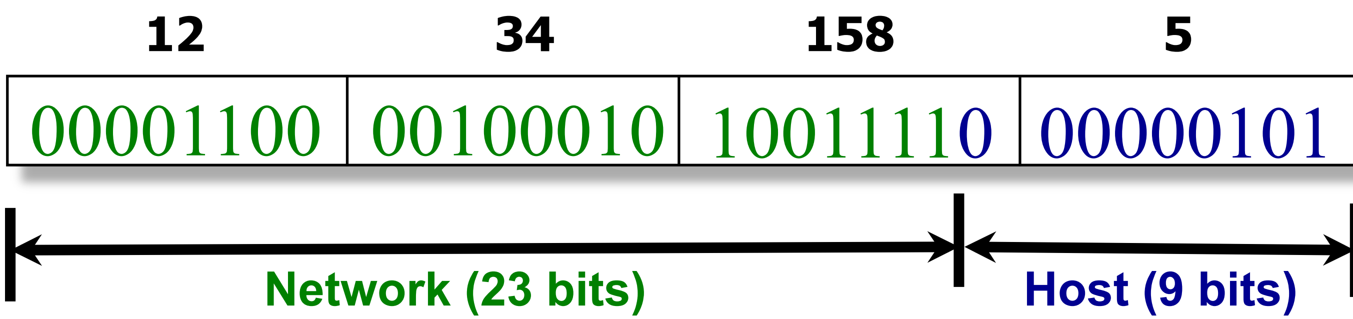
# IP Addresses

# IP Addresses (IPv4)

- Unique 32-bit, e.g., 12.34.158.5:

| 12 | 34 | 158 | 5 |
|---|---|---|---|
| 00001100 | 00100010 | 10011110 | 00000101 |

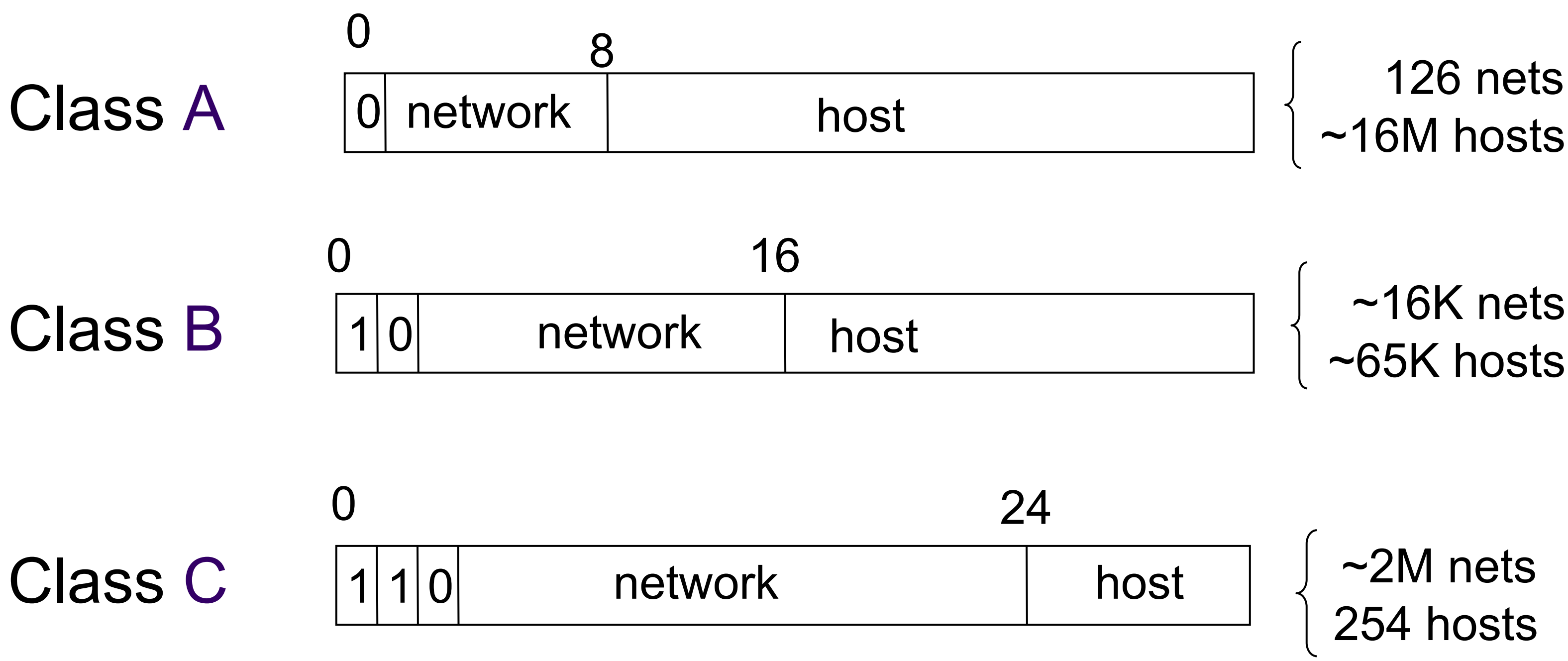
- Bits are partitioned into a prefix and suffix components
- Prefix is the network component; suffix is host component




- Notation: 12.34.158.0/23 represents a “slash 23” network with a 23 bit prefix and $2^9$ host addresses

# History of IP Addressing

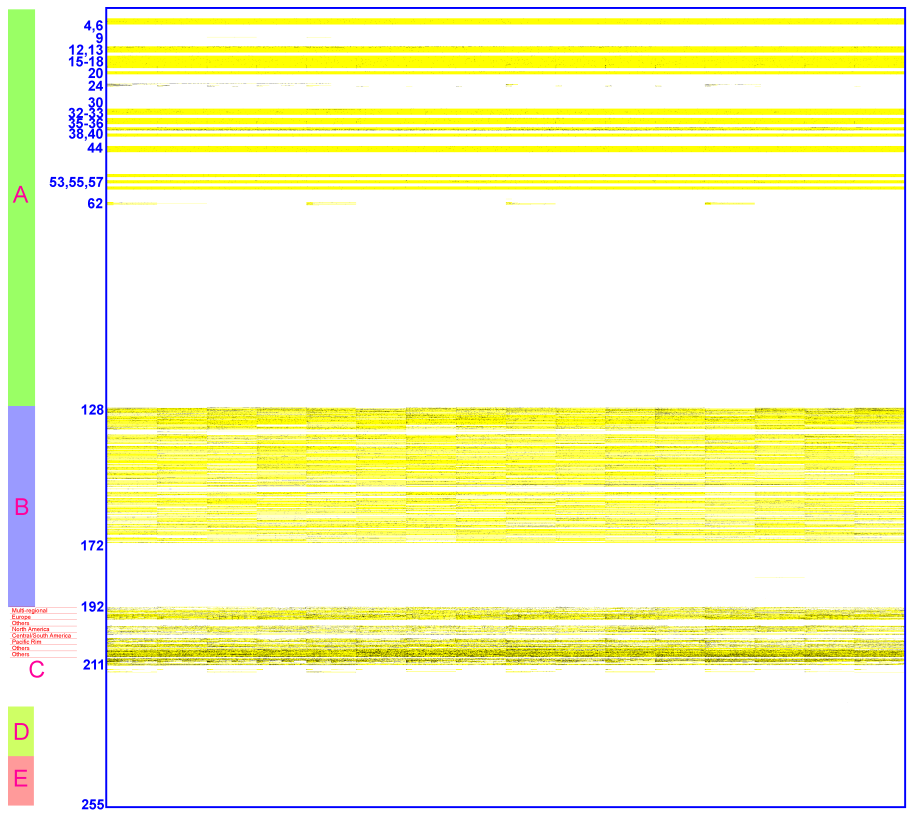
- First design: network 8 bits and host 24 bits
  - Can only index 256 networks!
- Next design: Partition addresses in classes
  - Can only support three sizes of networks




- Class D and E used for multicast and experiments

# IP Address Utilization ('97)

- http://www.caida.org/research/traffic-analysis/fix-west-1998/ipv4space/
- Map of available IPv4 address space with traffic activity superimposed (in yellow),



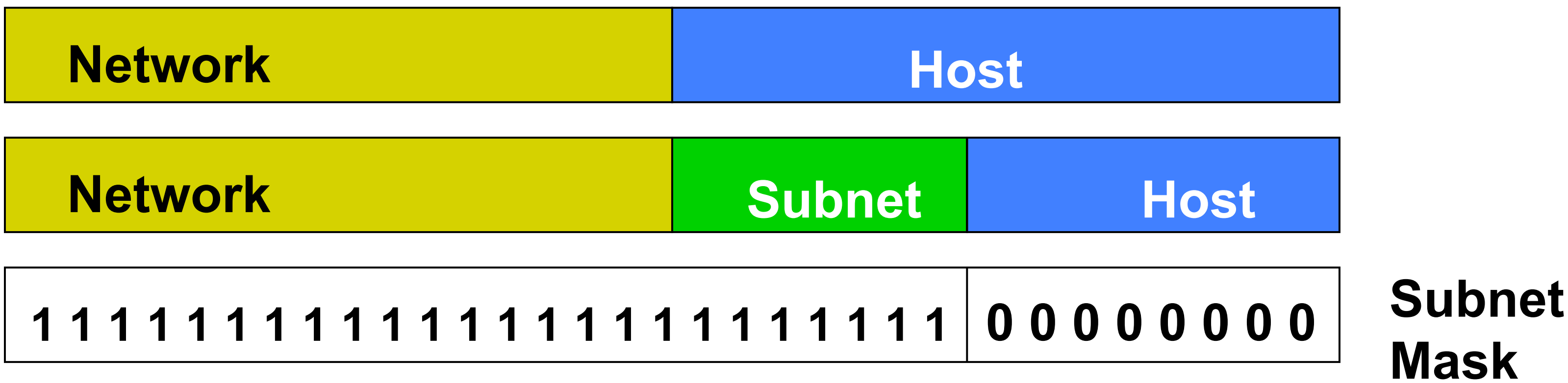
# Current IP Addressing

- To offer a better tradeoff between size of the routing table and efficient use of the IP address space, today we use CIDR
- CIDR = Classless Inter-Domain Routing

- Idea: Flexible division between network and host addresses
- Boundary between network and host must be explicitly specified
  - Informally, slash 26, i.e., 128.23.9/26
  - Formally, represent length of prefix with a 32-bit mask: 255.255.255.192 where all network prefix bits set to “1” and host suffix bits to “0”

# Subnets via IP Masks

- Subnet masks add another variable-length layer to hierarchy
  - Subnet a class B into several chunks



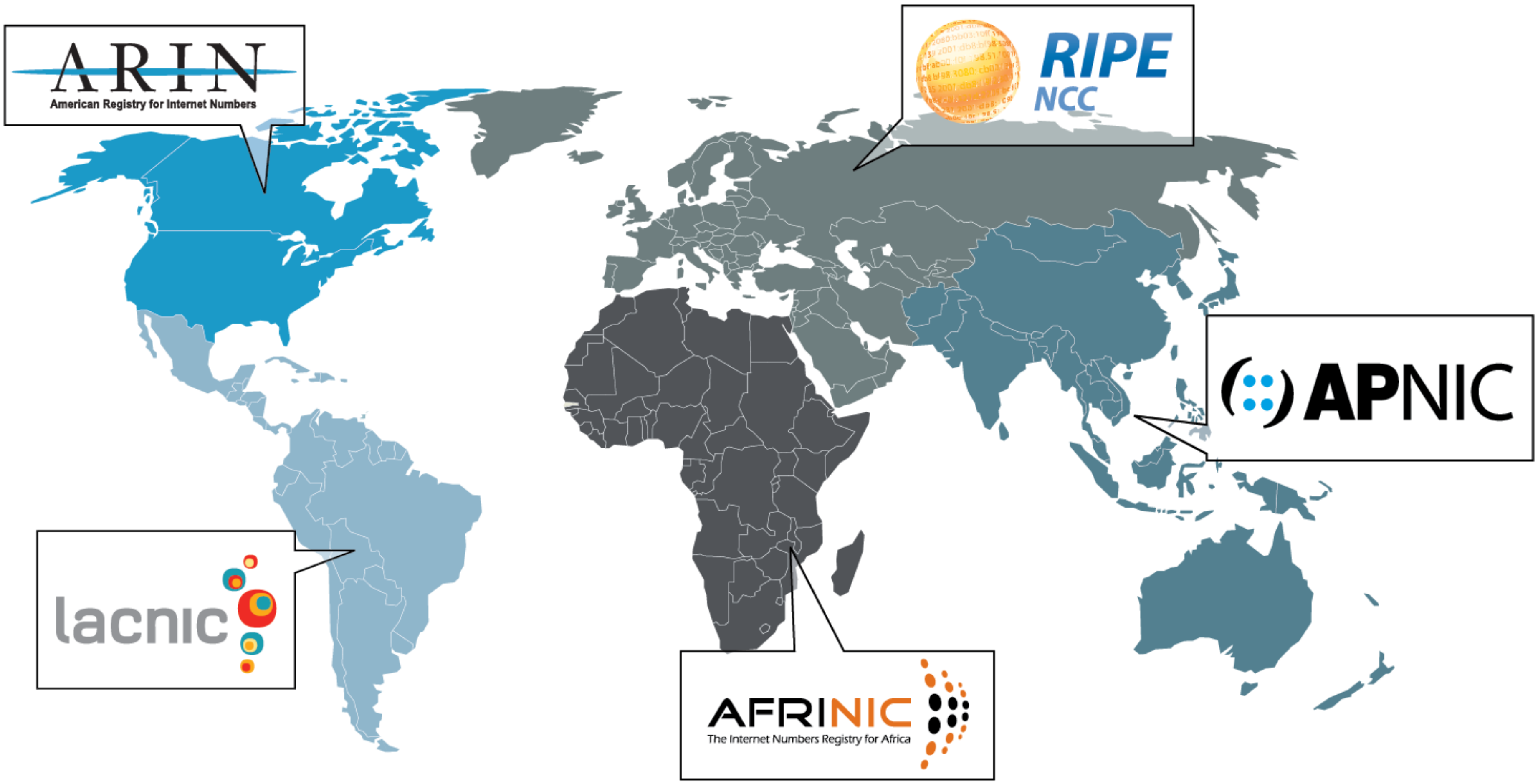

- Example: An organization needs 500 addresses.
  - A single class C address not enough (254 hosts).
  - Instead a class B address is allocated. (~65K hosts)
- CIDR allows an arbitrary prefix-suffix boundary
  - Hence, organization allocated a single /23 address

# Hierarchical Allocation of Addresses

- Internet Corporation for Assigned Names and Numbers (ICANN) gives large blocks to
- Regional Internet Registries (e.g., ARIN), which give blocks to
  - ARIN = American Registry for Internet Numbers
- Large institutions (ISPs), which give addresses to
- Individuals and smaller institutions
- Example: ICANN→ARIN→AT&T→UCSD→ECE

# Allocation of IP Addresses

- http://en.wikipedia.org/wiki/List_of_assigned_/8_IPv4_address_blocks



# Example in More Detail

- ICANN gives ARIN several /8s
- ARIN gives AT&T one /8, **12.0/8**
  - Network Prefix: **00001100**
- AT&T gives UCSD a /16, **12.197/16**
  - Network Prefix: **0000110011000101**
- UCSD gives ECE a /24, **12.197.45/24**
  - Network Prefix: **000011001100010100101101**
- ECE gives me a specific address **12.197.45.23**
  - Address: 00001100110001010010110100010111

# How to Get and IP Address

- In practice, there are two ways network administrators use to assign IP addresses
  - Hard-code assignments in a file
  - DHCP: Dynamic Host Configuration Protocol: dynamically get address: “plug-and-play”
    - Host broadcasts “DHCP discover” msg
    - DHCP server responds with “DHCP offer” msg
    - Host requests IP address: “DHCP request” msg
    - DHCP server sends address: “DHCP ack” msg

# Summary of Addressing

- Hierarchical addressing
  - Critical for scalable system
  - Do not require everyone to know everyone else
  - Reduces amount of updating when something changes

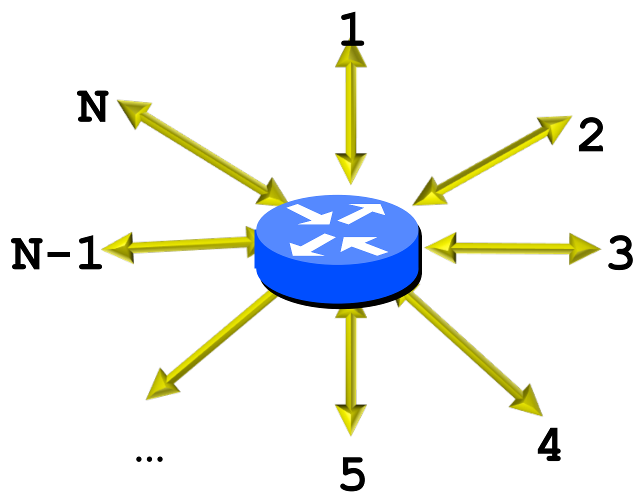
- Non-uniform hierarchy
  - Useful for heterogeneous networks of different sizes
  - Class-based addressing was far too coarse
  - Classless InterDomain Routing (CIDR) more flexible

# IP Routers

# IP Routers

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

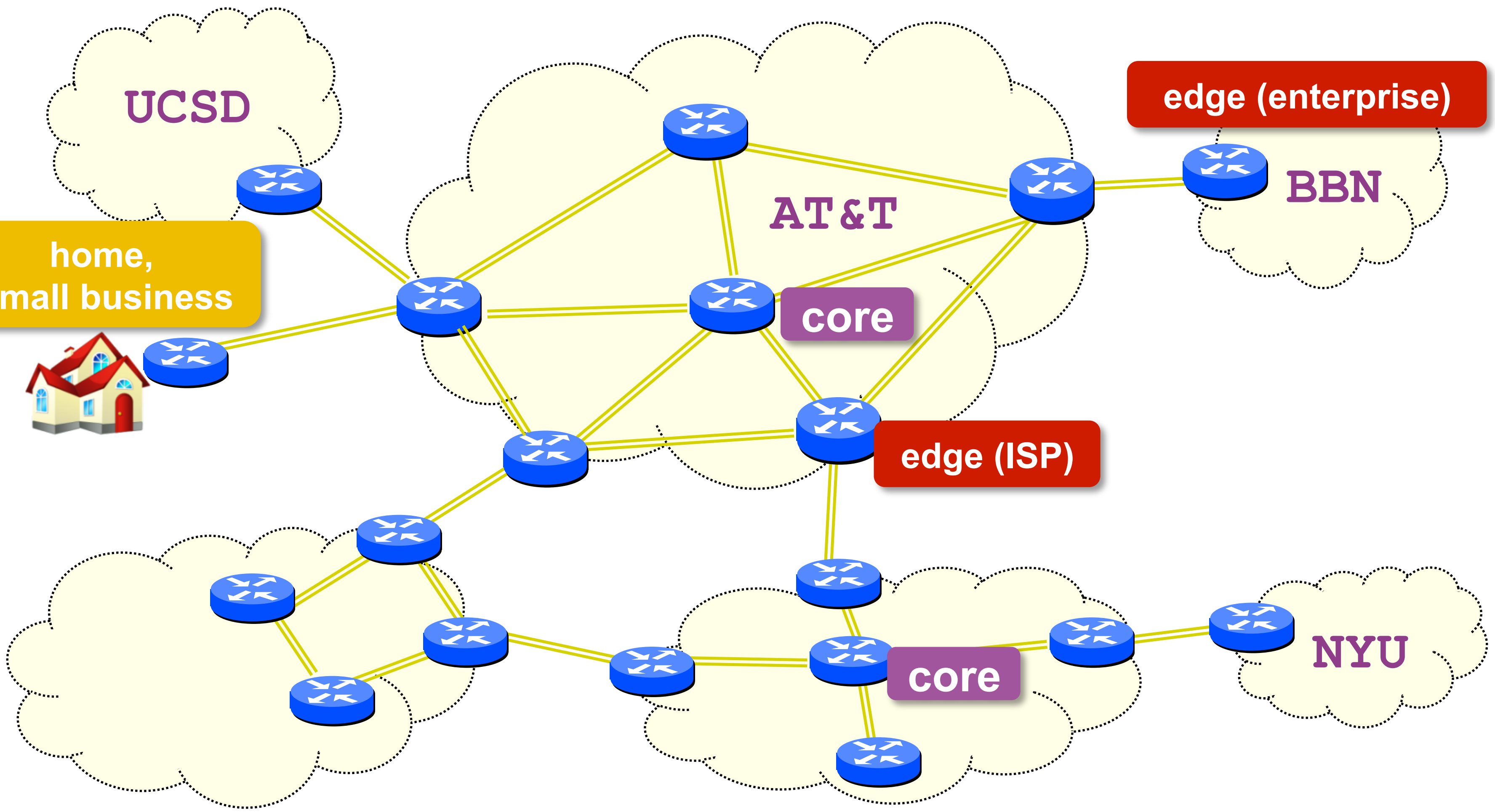
- Definitions:




R *bits/sec*

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

# Networks and routers



# Examples of routers (core)

**Juniper T4000**

- R= 10/40 Gbps
- NR = 4 Tbps

**Cisco CRS**

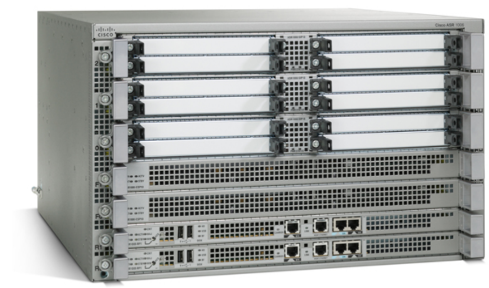
- R=10/40/100 Gbps
- NR = 322 Tbps

72 racks, 1MW

# Examples of routers (edge)

**Cisco ASR 1006**

- R=1/10 Gbps
- NR = 40 Gbps



**Juniper M120**

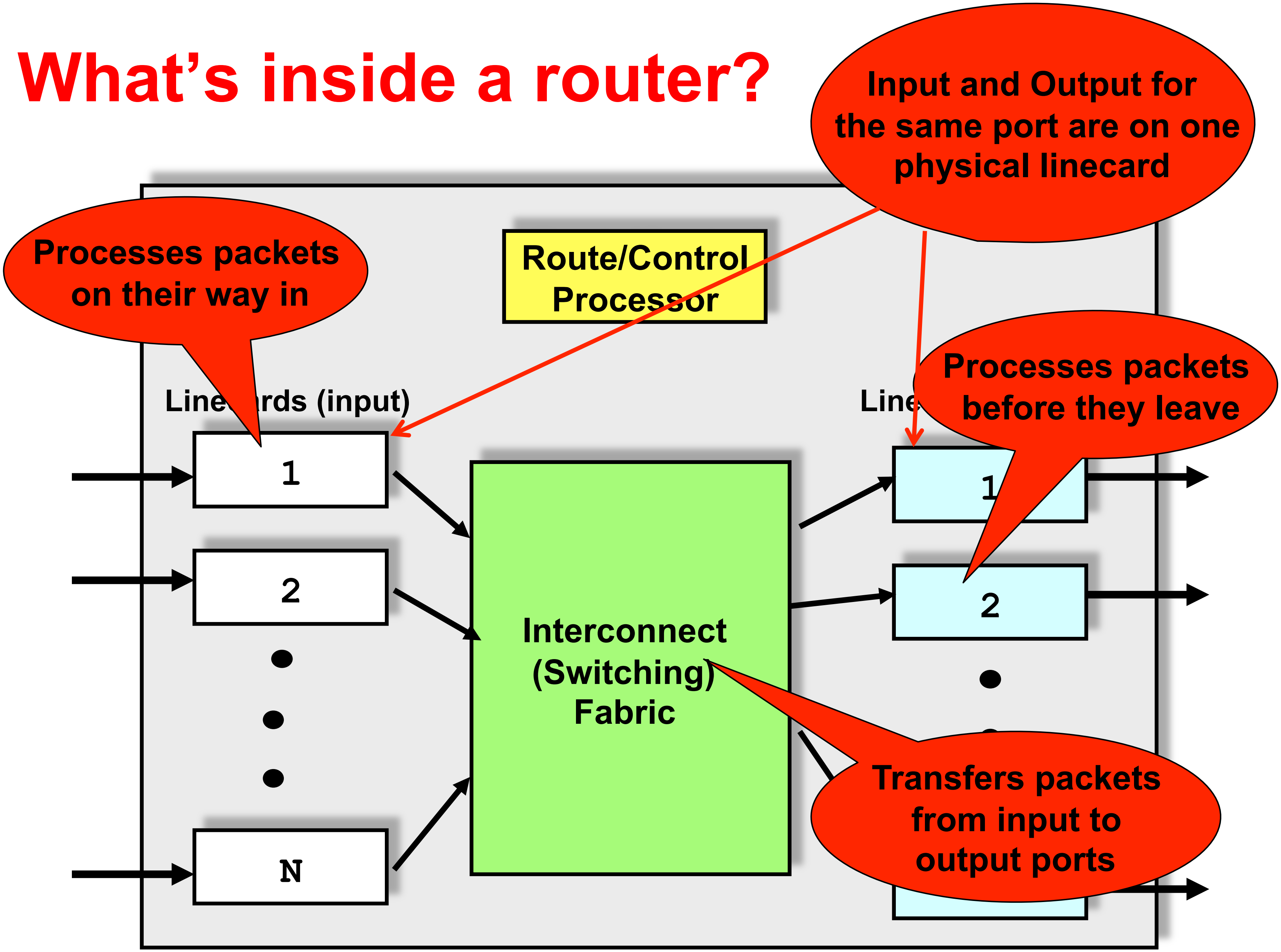
- R= 2.5/10 Gbps
- NR = 120 Gbps

# Examples of routers (small business)

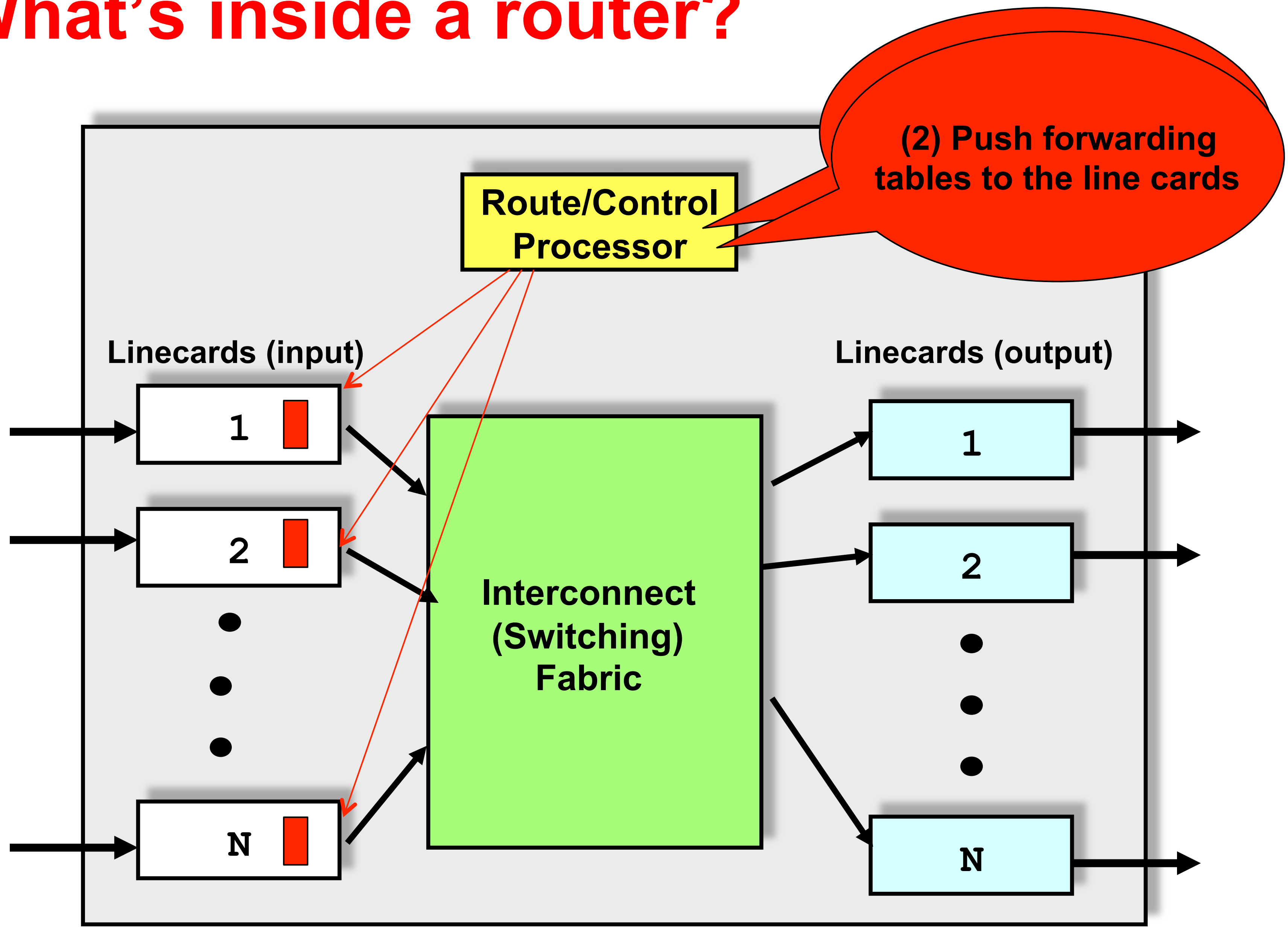
**Cisco 3945E**

- R = 10/100/1000 Mbps
- NR < 10 Gbps

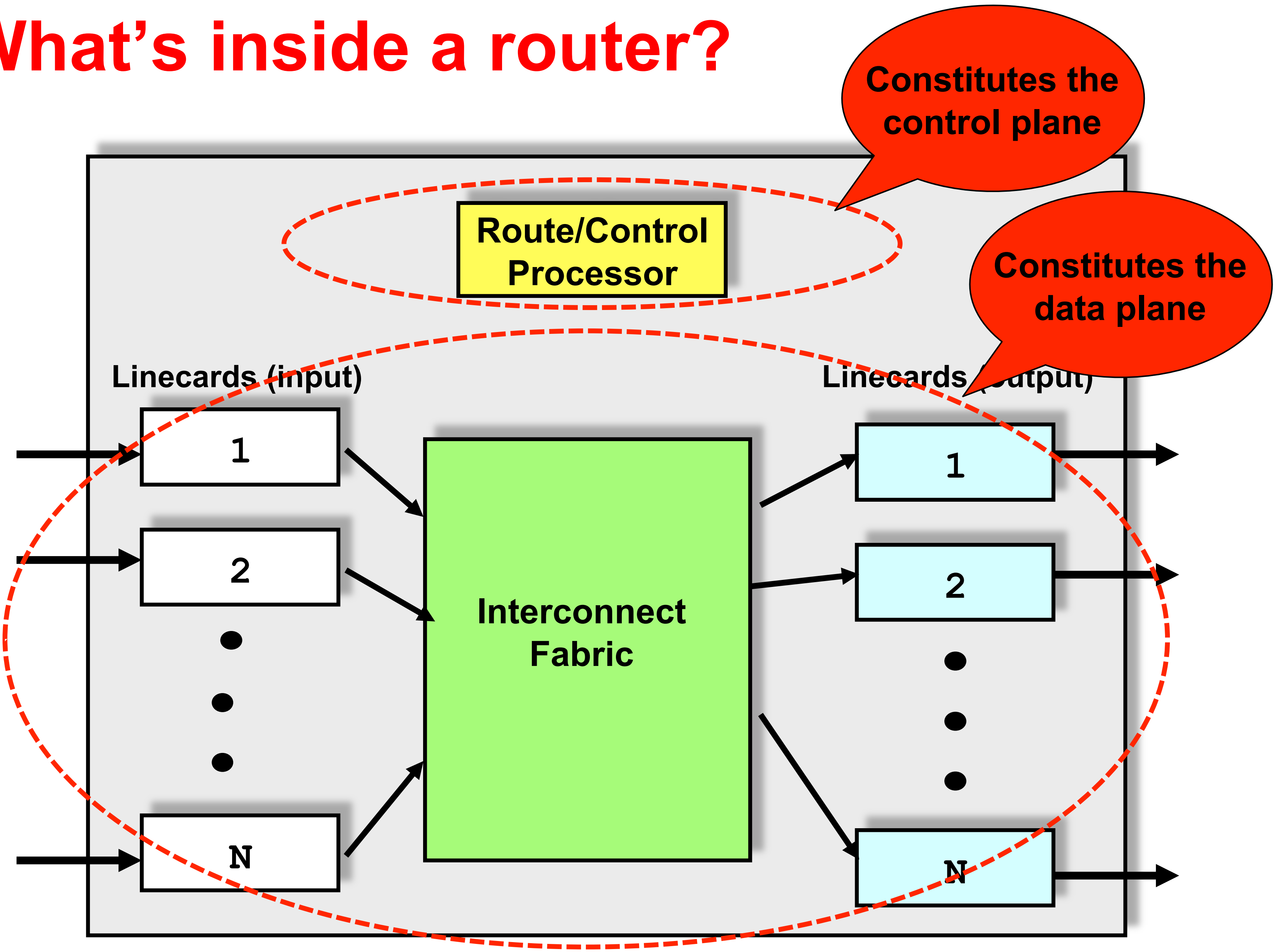
# What’s inside a router?



# What's inside a router?



# What's inside a router?



# In Summary

- Base-level protocol (IP) provides minimal service level
  - Allows highly decentralized implementation
  - Each step involves determining next hop
  - Most of the work at the endpoints

- IP forwarding → global addressing, alternatives, lookup tables
- IP addressing → hierarchical, CIDR
- IP service → best effort, simplicity of routers
- IP packets → header fields, fragmentation, ICMP