Last Lecture: Network Layer

1. *Design goals and issues* ✔
2. *Basic Routing Algorithms & Protocols* ✔
   - Packet Forwarding
   - Shortest-Path Algorithms
   - Routing Protocols
3. *Addressing, fragmentation and reassembly*
4. *Internet Routing Protocols and Inter-networking*
5. *Router design*
6. *Congestion Control, Quality of Service*
7. *More on the Internet's Network Layer*

This Lecture: Network Layer

1. *Design goals and issues*
2. *Basic Routing Algorithms & Protocols*
3. *Addressing, Fragmentation and reassembly* ✔
   - *Hierarchical addressing*
   - *Address allocation & CIDR*
   - *IP fragmentation and reassembly*
4. *Internet Routing Protocols and Inter-networking*
5. *Router design*
6. *Congestion Control, Quality of Service*
7. *More on the Internet's Network Layer*

1. IP Addressing

- Dotted-quad notation: here's timberlake.cse's *IP*

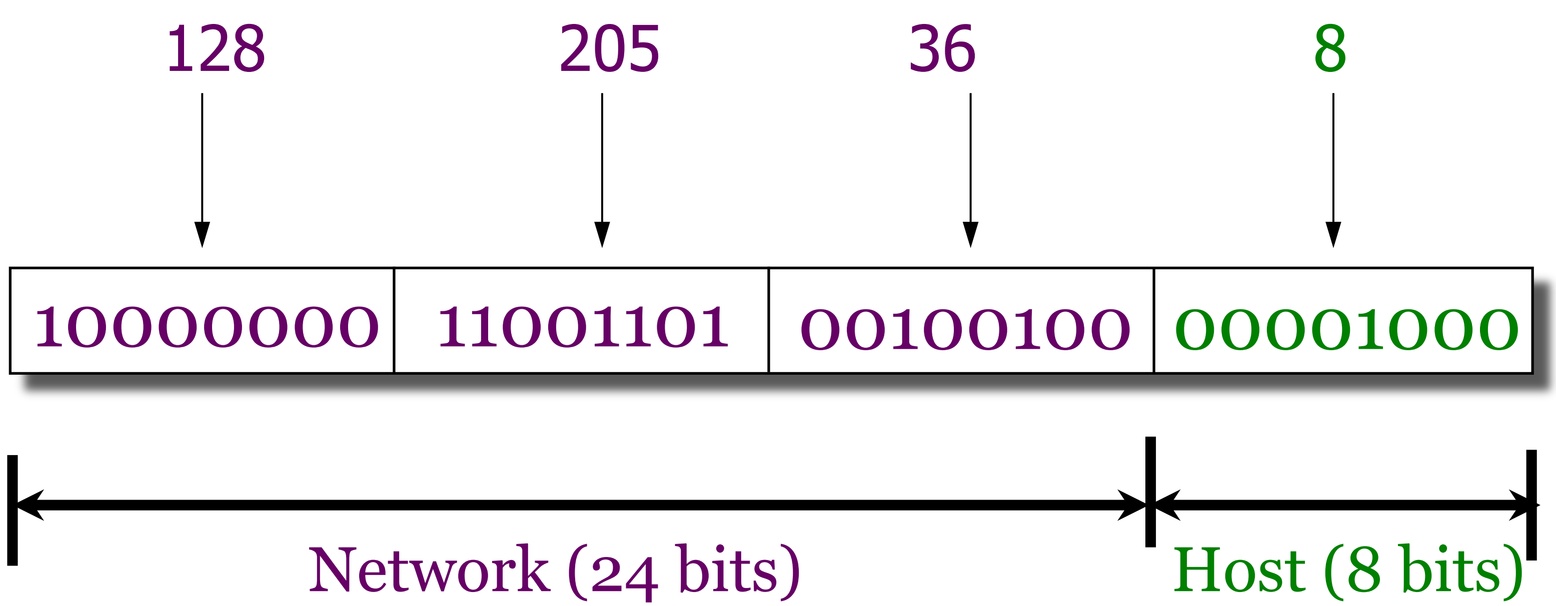
128	205	36	8
10000000	11001101	00100100	00001000

- *Theoretically*, up to $2^{32} \approx 4$ billion hosts
- *Practically*, about 768 millions (Jul 2010, ISC Survey), still huge!

- Routing table with 768M entries? No no.

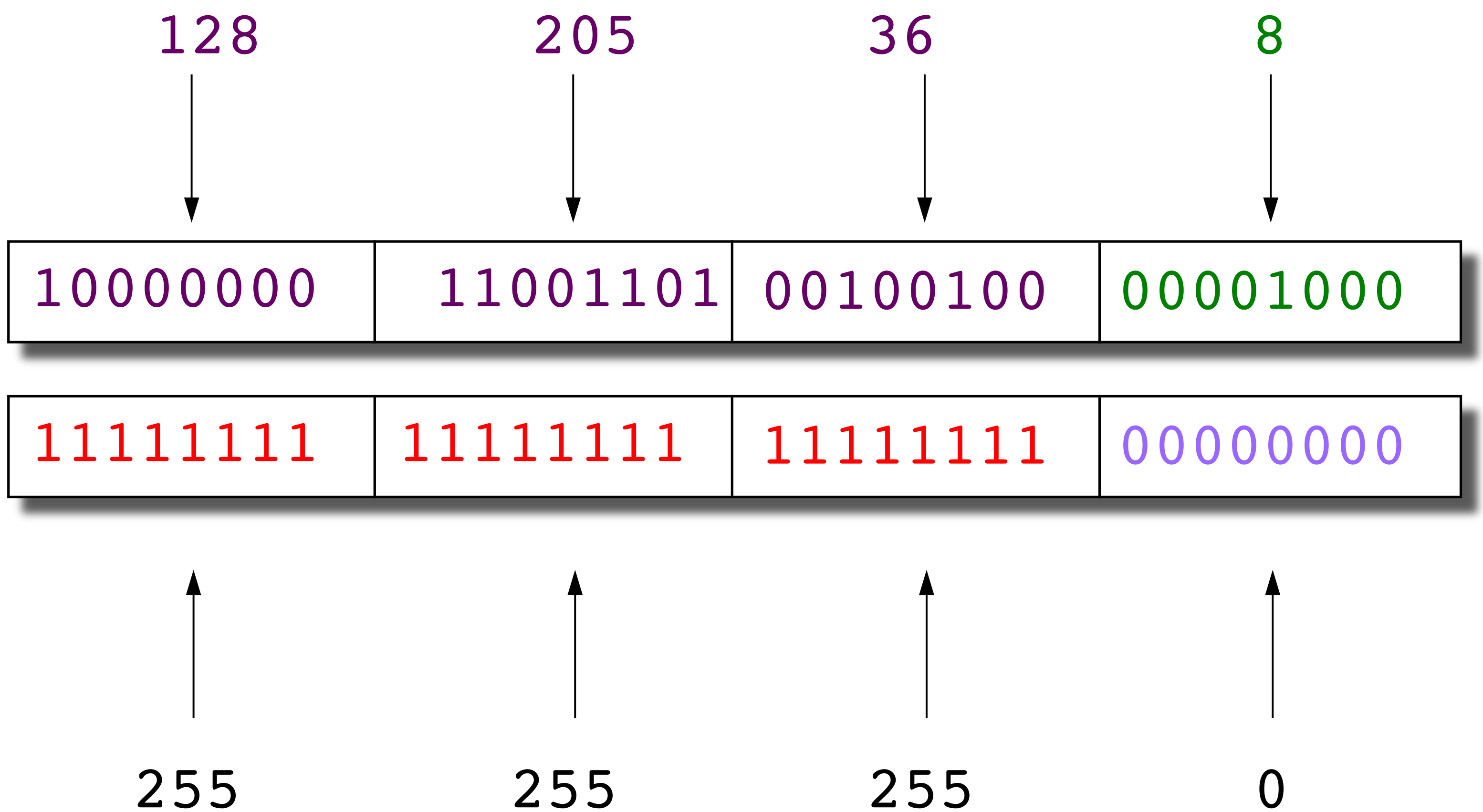
Hierarchical Addressing: Rough Idea

- Each "network" assigned a prefix
- Foreign routers' routing tables only need an entry for the entire "network"
  - The entry points to the network's "gateway(s)"



Subnet Mask: Extracting the Network Prefix

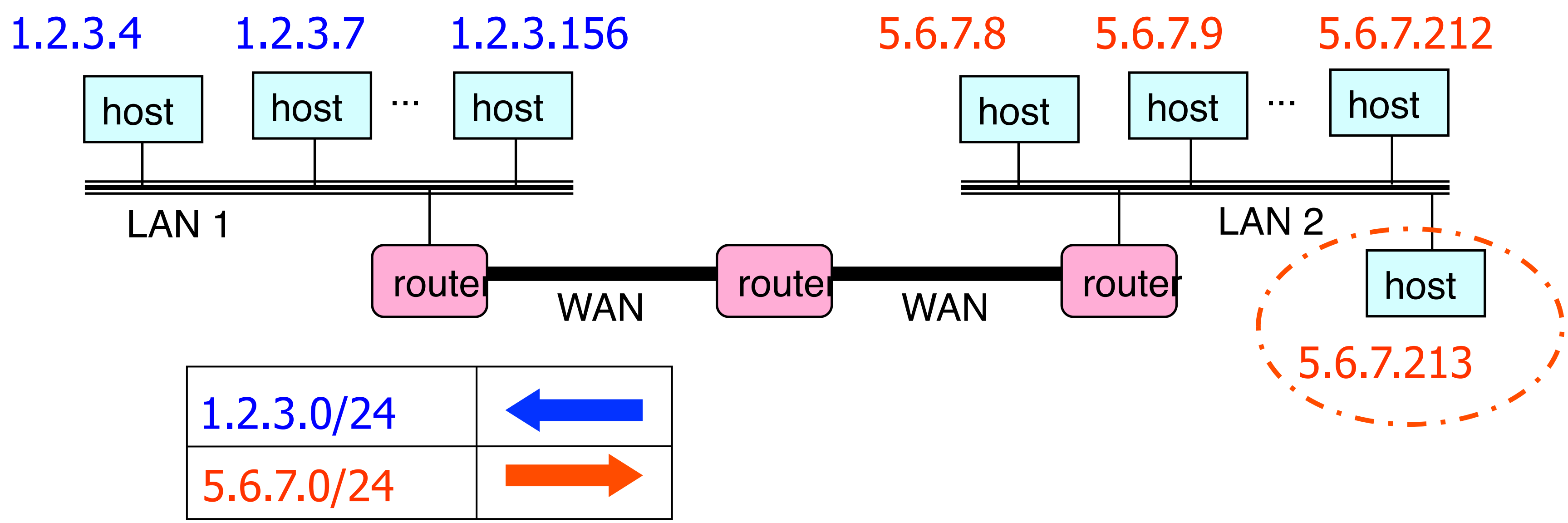
Address



Mask

Scalability Improved

- Routing tables are smaller (but still too big)
- No need to update the routers when new host added
  - E.g., adding a new host 5.6.7.213 on the right
  - Doesn't require adding a new forwarding-table entry

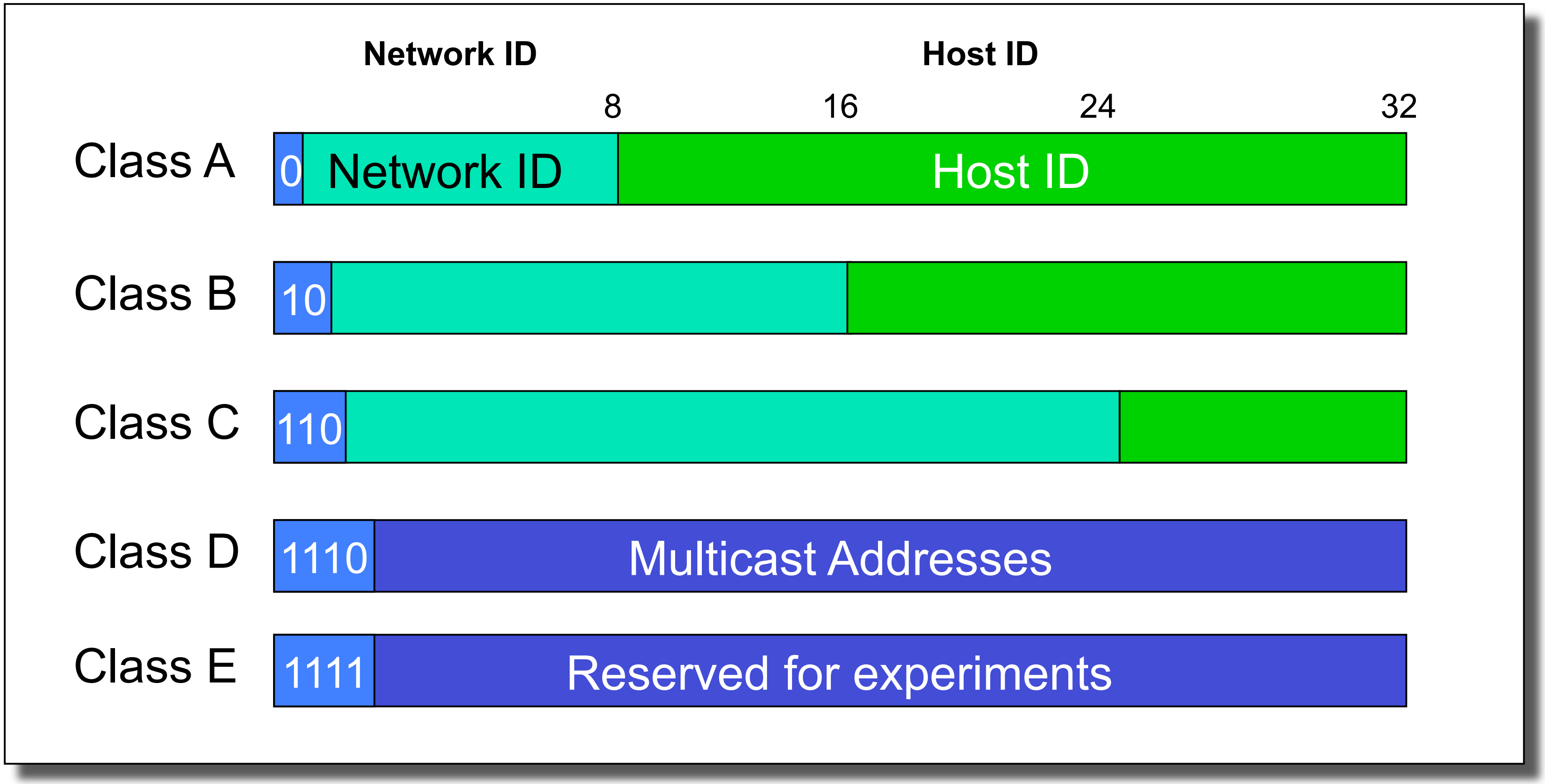


forwarding table

Address Allocation

- How to partition the address space into “blocks”
- Who gets which block?

Classful Allocation (The Old Way)

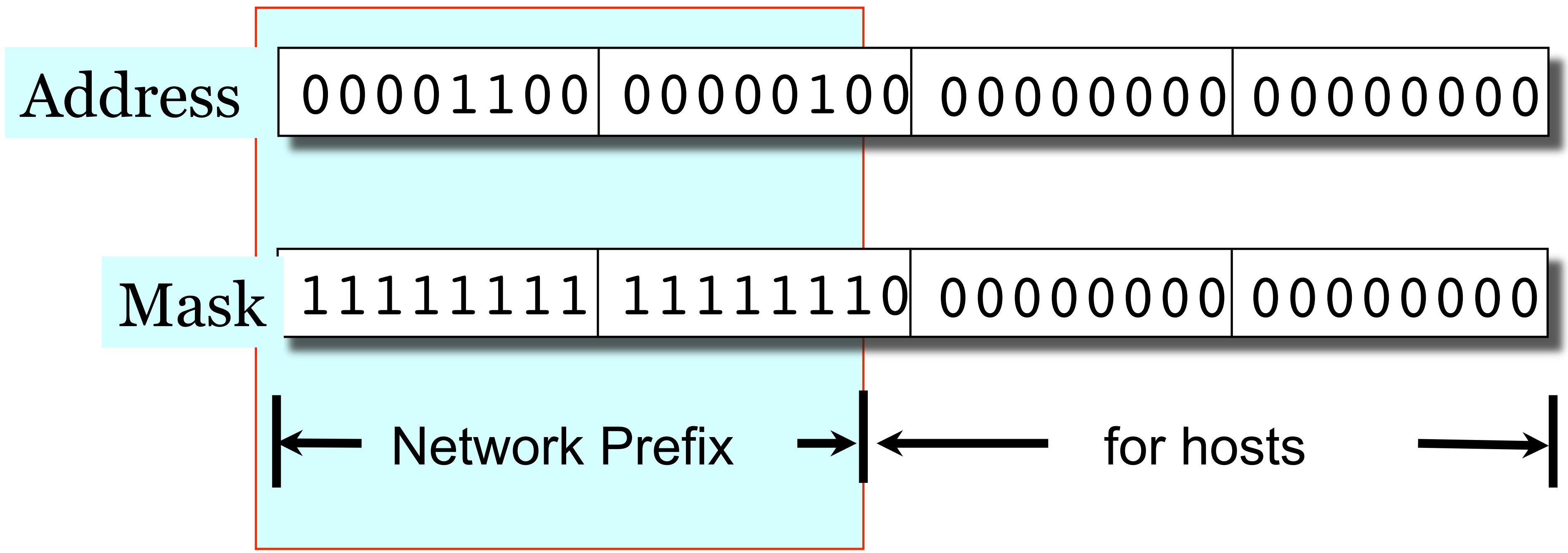


This is why dotted-quad notation is used

Classless Inter-Domain Routing (CIDR)

Use two 32-bit numbers to represent a network.
Network number = IP address + Mask

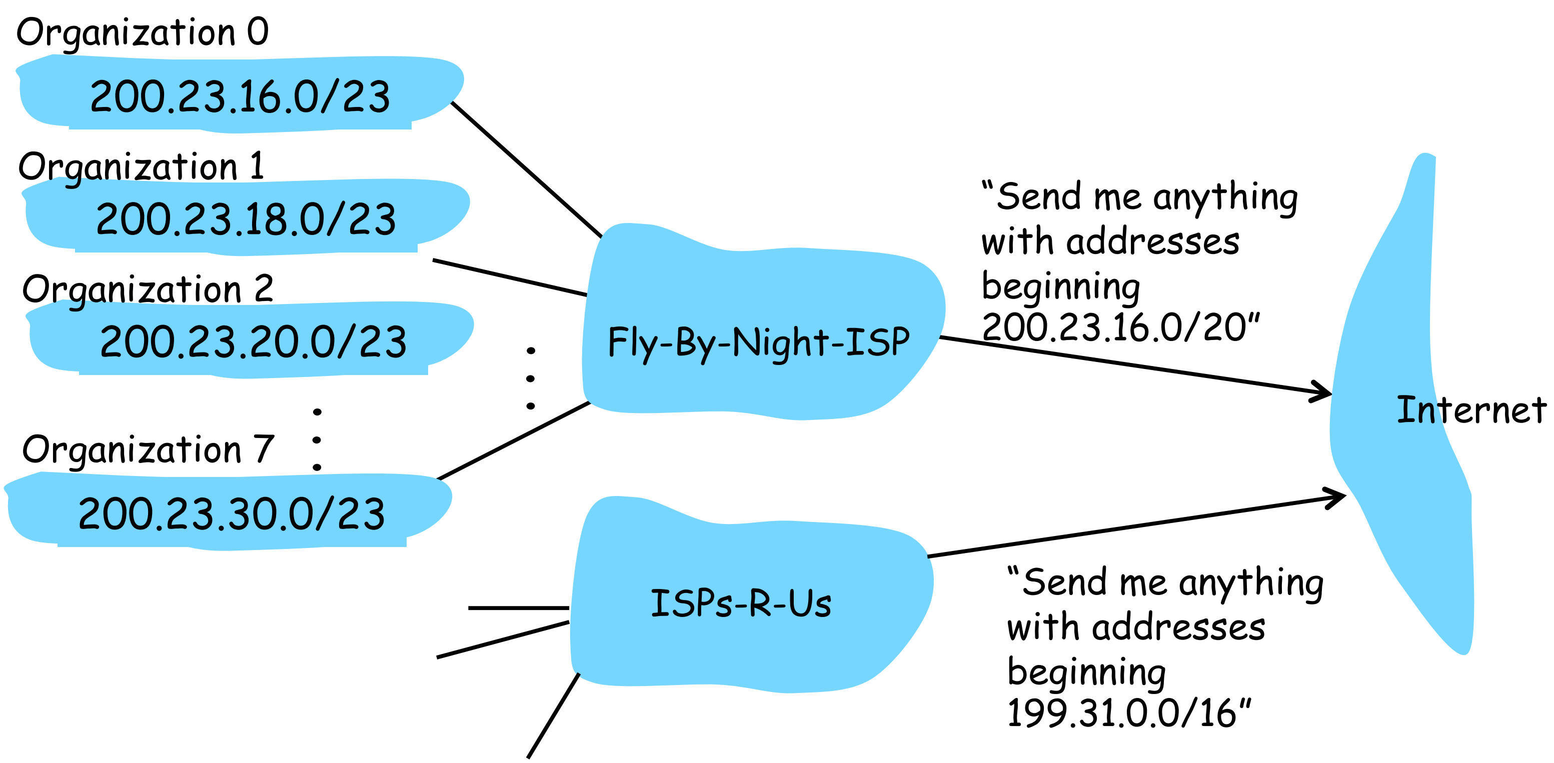
IP Address : 12.4.0.0   IP Mask: 255.254.0.0



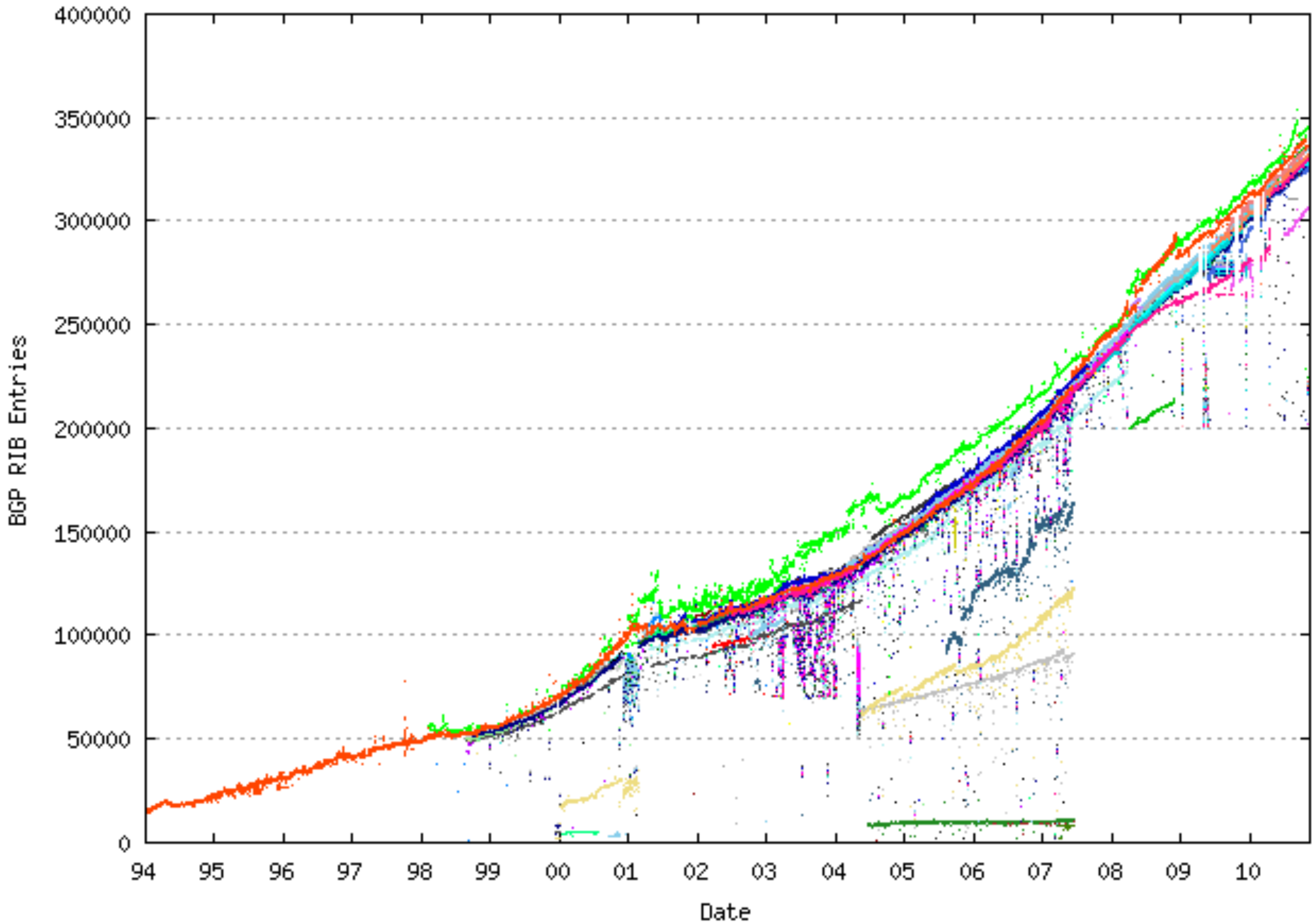
Written as 12.4.0.0/15

CIDR: Reduce Routing Table Sizes

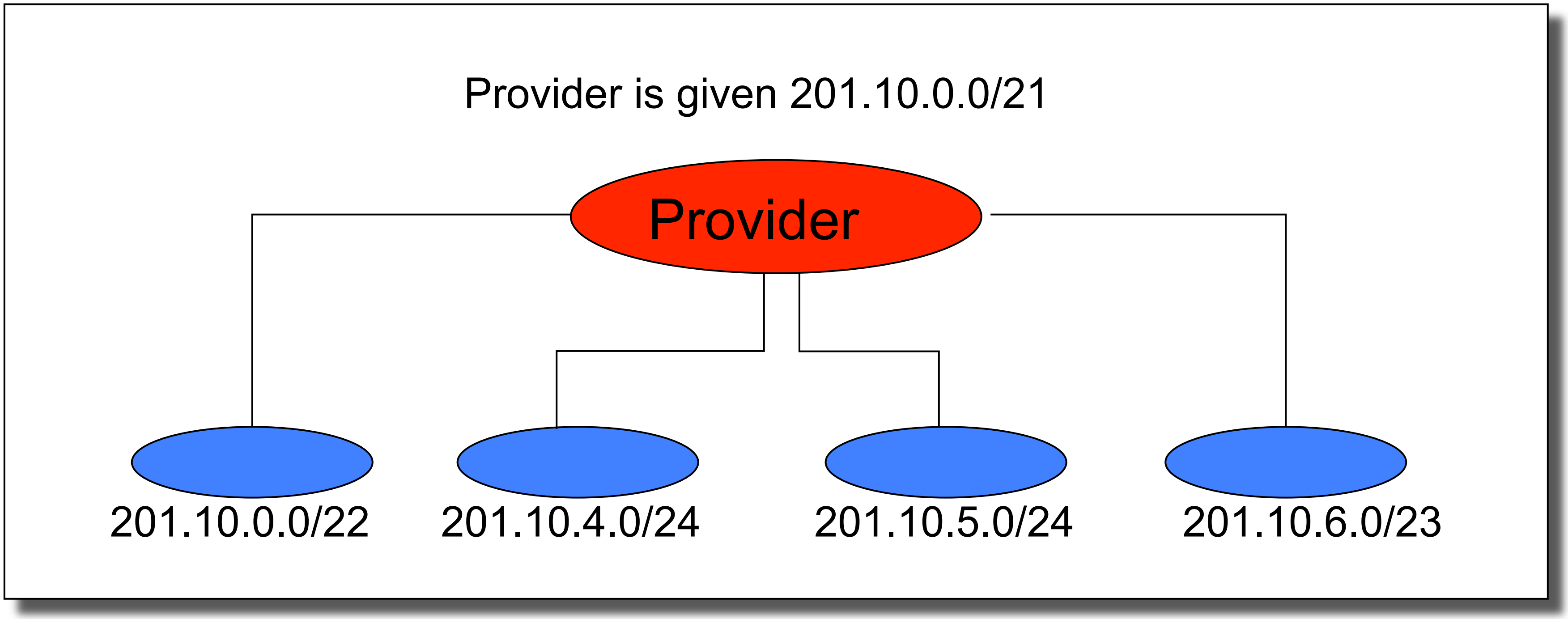
- About 350K entries to date



(BGP) Routing Table Size Growth

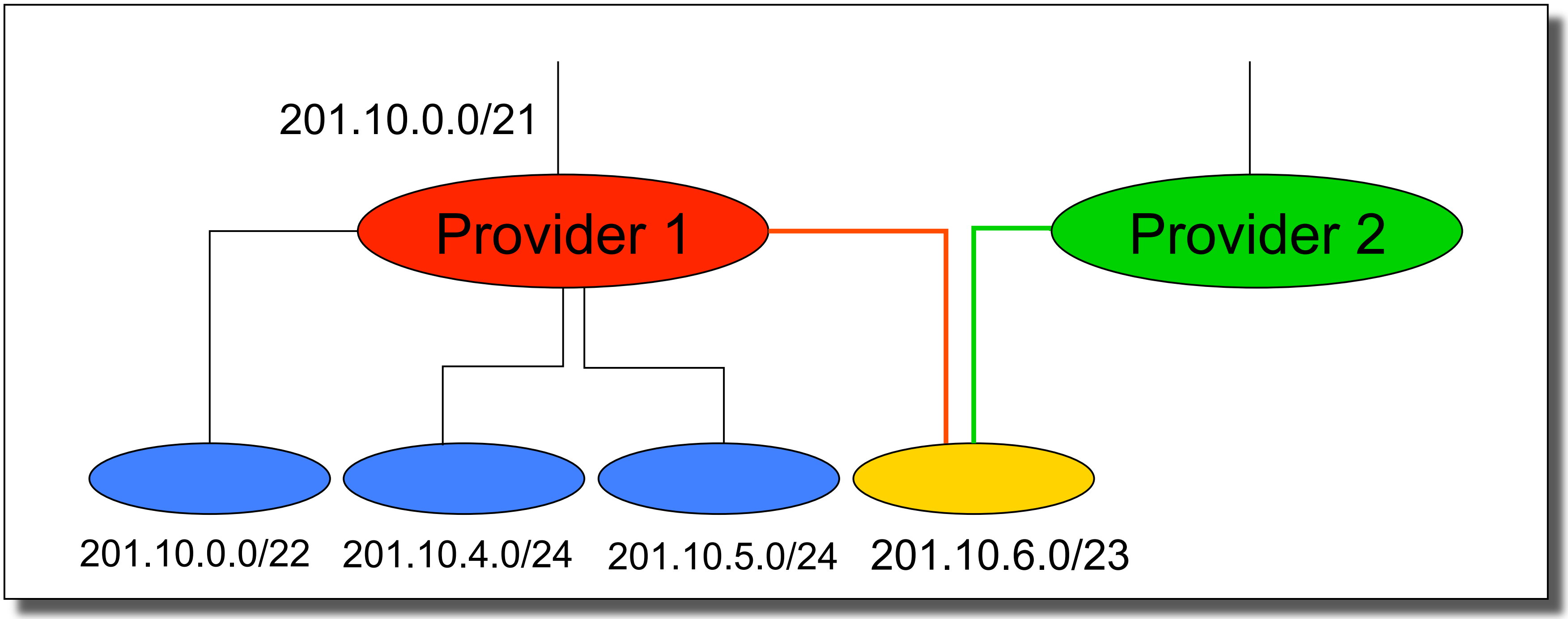


Scalability: Address Aggregation



Routers in the rest of the Internet just need to know how to reach 201.10.0.0/21. The provider can direct the IP packets to the appropriate customer.

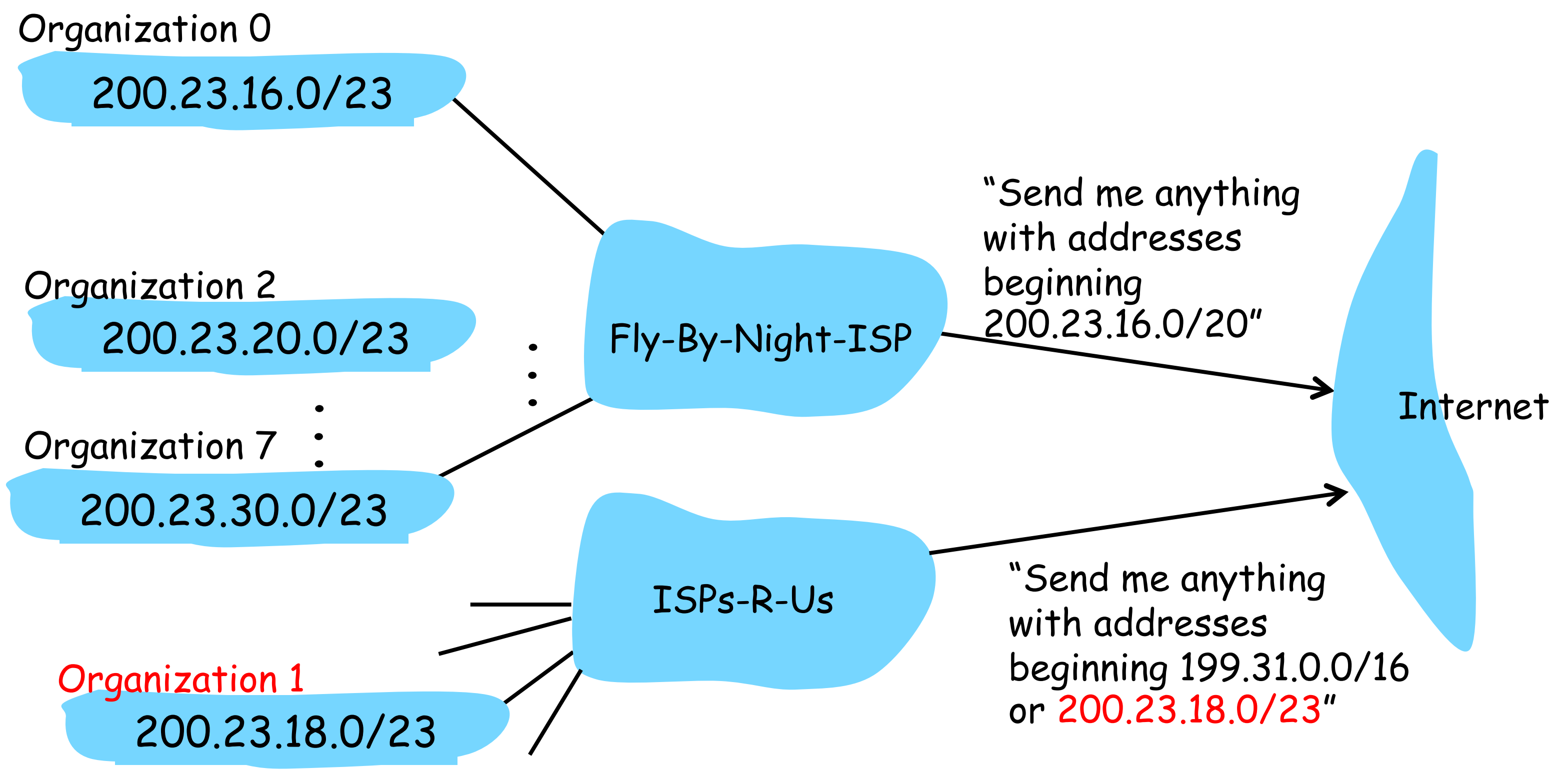
But, Aggregation Not Always Possible



Multi-homed customer with 201.10.6.0/23 has two providers. Other parts of the Internet need to know how to reach these destinations through *both* providers.

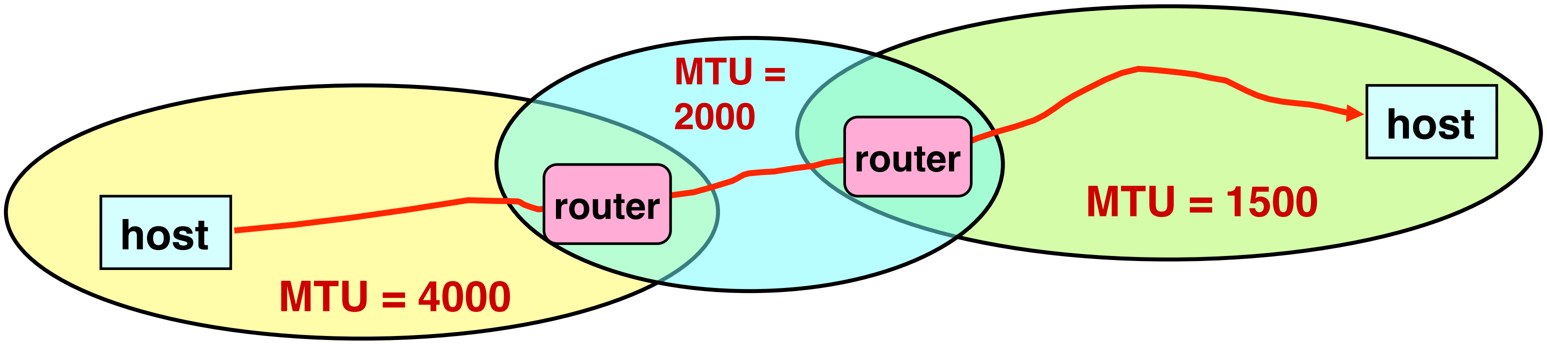
CIDR Not a Free Lunch

ISPs-R-Us has a more specific route to Organization 1



Requires routers to do *longest prefix match*, per packet, every few nanosecond

2. IP Fragmentation and Reassembly



- A packet may hit networks with different MTUs
- *Fragmentation* needed at networks whose MTUs are smaller than the packet
- *Reassemble* the packet after getting out

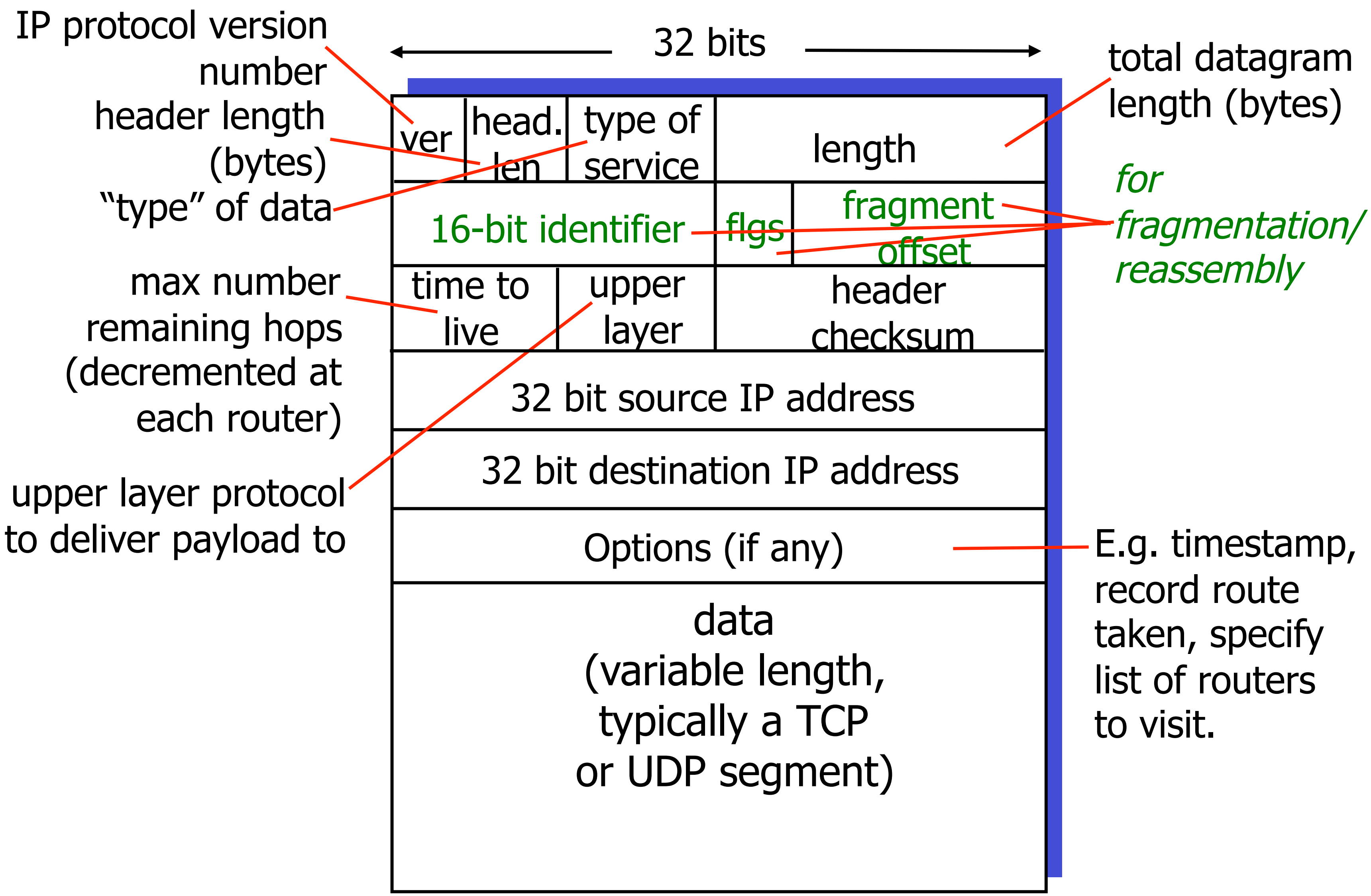
Where to do Reassembly

- At end nodes or routers?

- *At routers:*
  - Con: How much buffer space required at routers?
  - Con: What if routes in network change? Or there are multiple paths to the same destination?

- *At end (receiving) nodes*
  - Pro: avoids unnecessary work where large packets are fragmented multiple times
  - Pro: at routers, less buffer space & less computation
  - Con: if any fragment missing, retransmit entire packet through entire path, wasting bandwidth
  - *TCP/IP takes this approach*

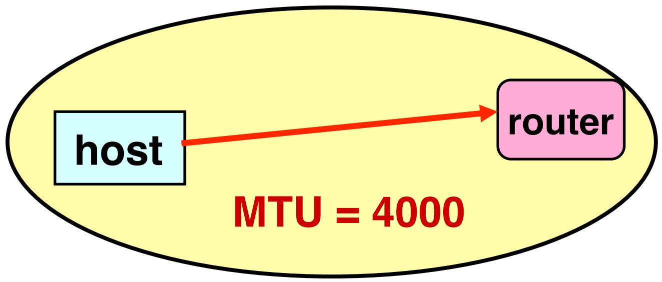
IP Packet Format

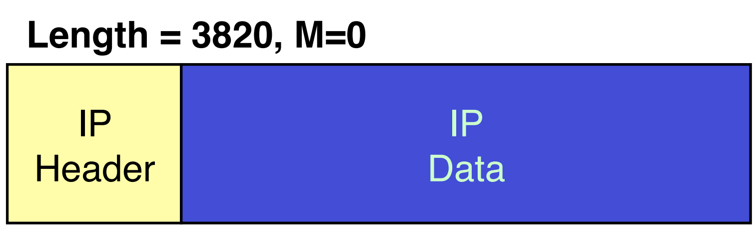


Fragmentation Related Fields

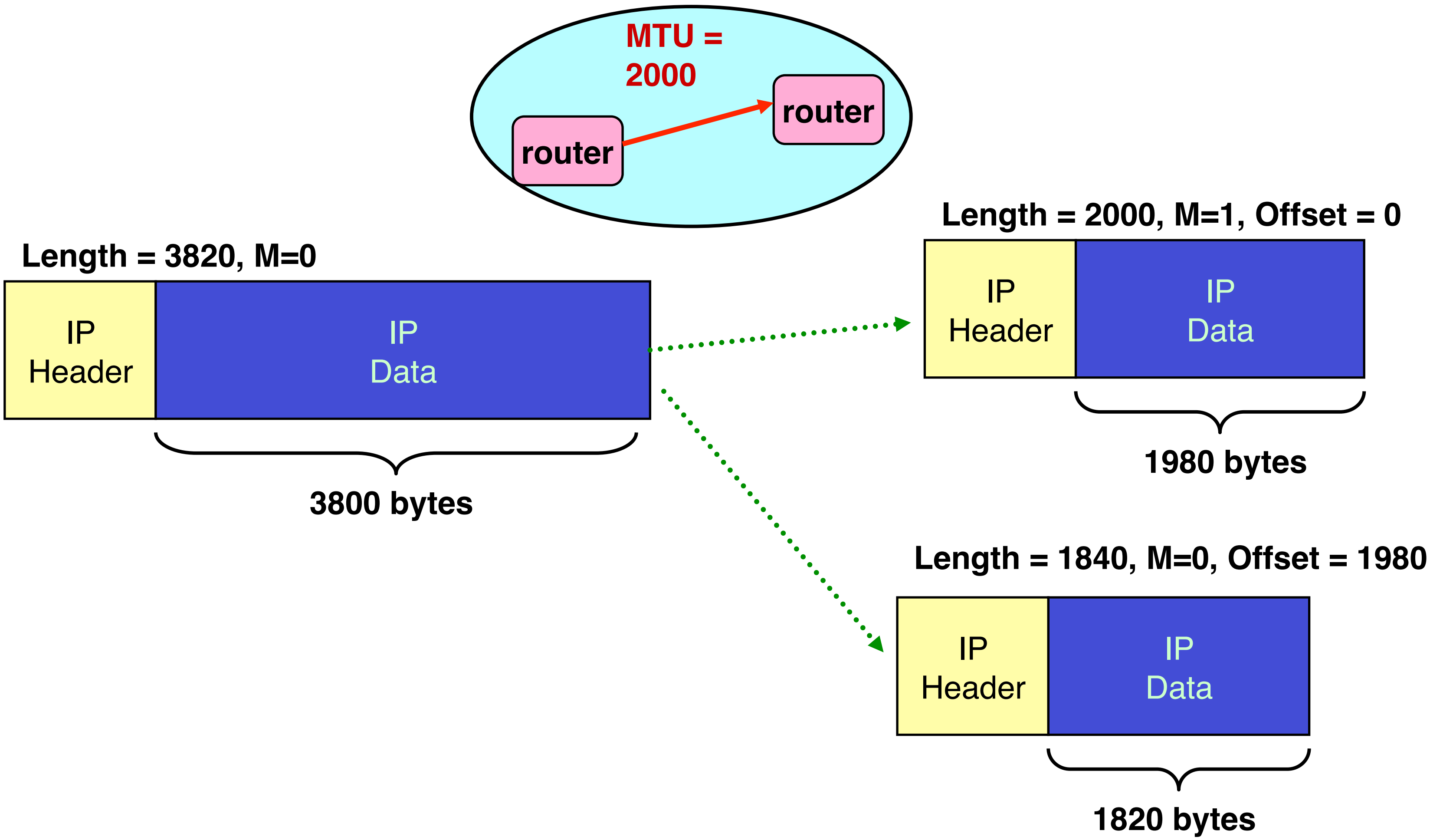
- *Length*
  - Length of IP fragment
- *Identifier*
  - To match up with other fragments
- *Flags*
  - **D**on't fragment flag
  - **M**ore fragments flag
- *Fragment offset*
  - Where this fragment lies in entire IP datagram
  - Measured in 8 octet units (13 bit field)

IP Fragmentation Example #1

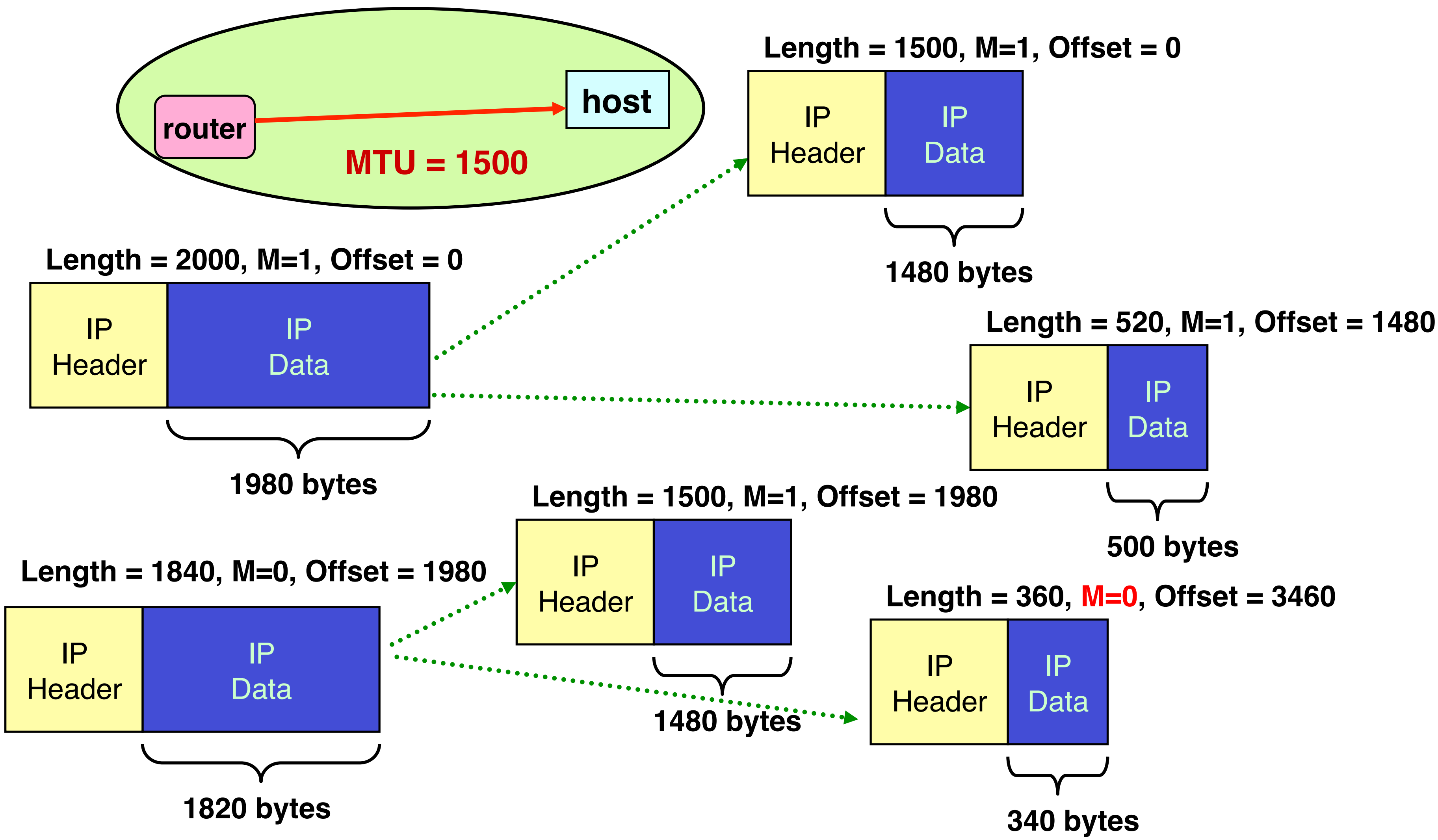




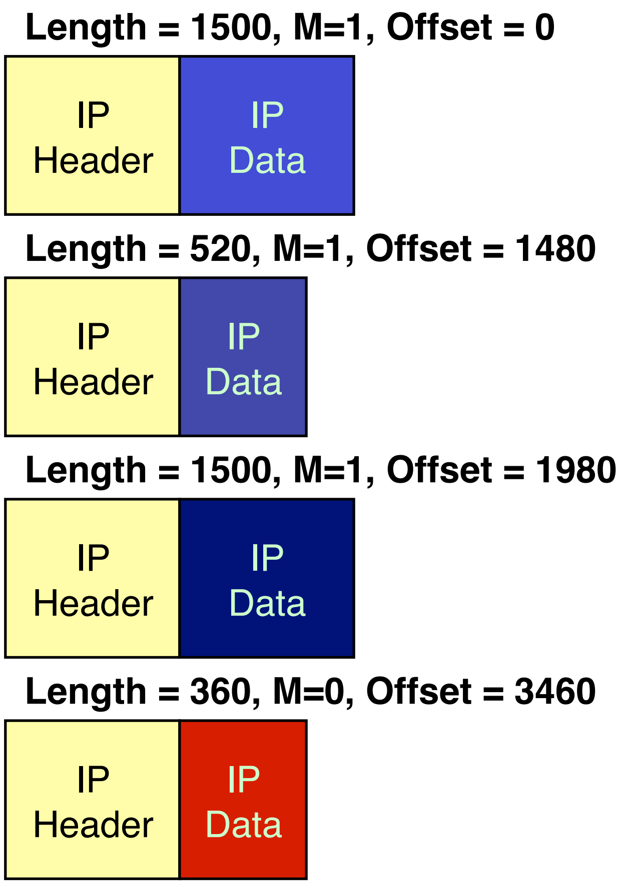
IP Fragmentation Example #2



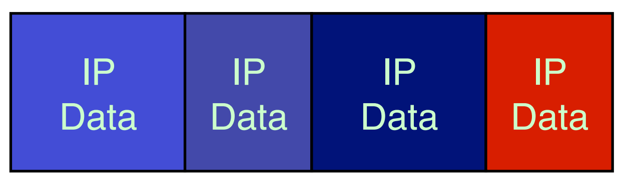
IP Fragmentation Example #3



IP Reassembly



- Fragments might arrive out-of-order
  - Don't know how much memory required until receive final fragment
- Some fragments may be duplicated
  - Keep only one copy
- Some fragments may never arrive
  - After a while, give up entire process



Fragmentation and Reassembly Concepts

- *Decentralized*: Every network can choose MTU
- *Connectionless*
  - Each (fragment of a) packet contains full routing information
  - Fragments travel independently
- *Best effort*
  - Fail by dropping packet
  - Destination can give up on reassembly
  - No need to signal sender that failure occurred
- *E2E principle*
  - Reassembly at endpoints
- *These are key networking principles!*

Fragmentation is Harmful

- Uses resources poorly
  - Forwarding costs per packet
  - Best if we can send large chunks of data
  - Worst case: packet just bigger than MTU
- Poor end-to-end performance
- Solution: *Path MTU discovery* protocol

- Common theme in system design
  - Assure correctness by implementing complete protocol
  - Optimize common cases to avoid full complexity