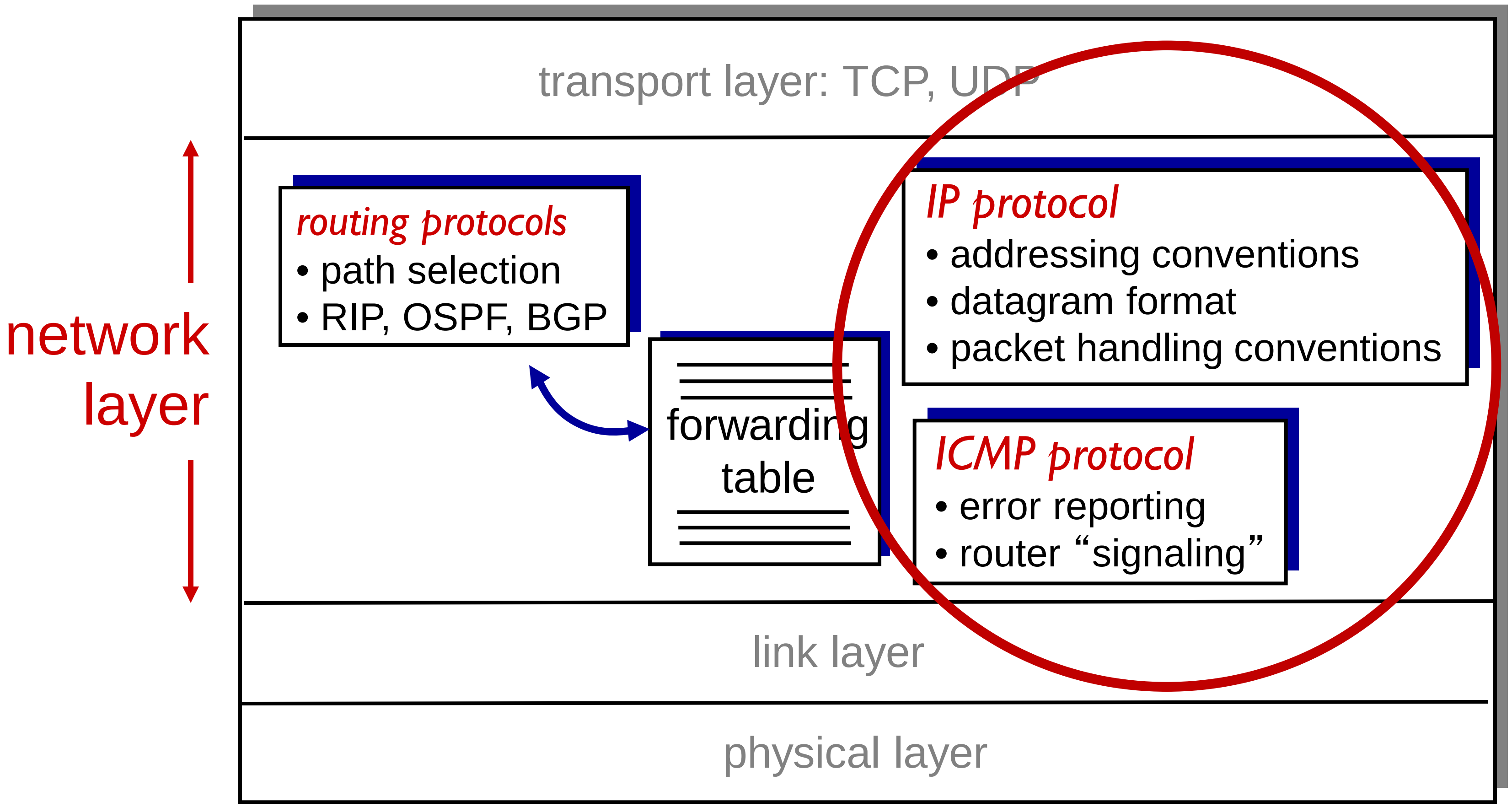
The Internet Protocol (IP)

Richard T. B. Ma
School of Computing
National University of Singapore

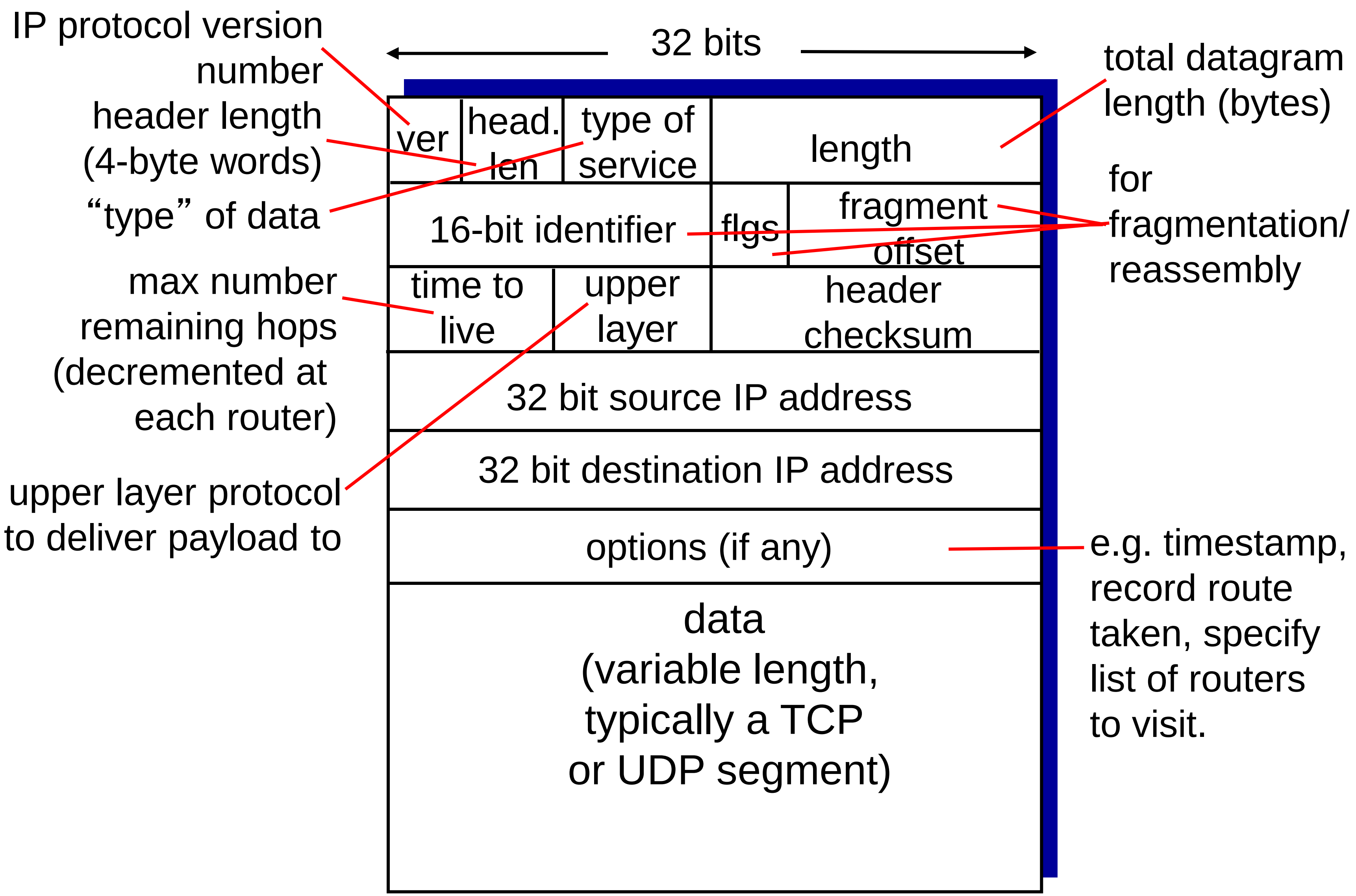
CS 3103: Compute Networks and Protocols

Internet network layer

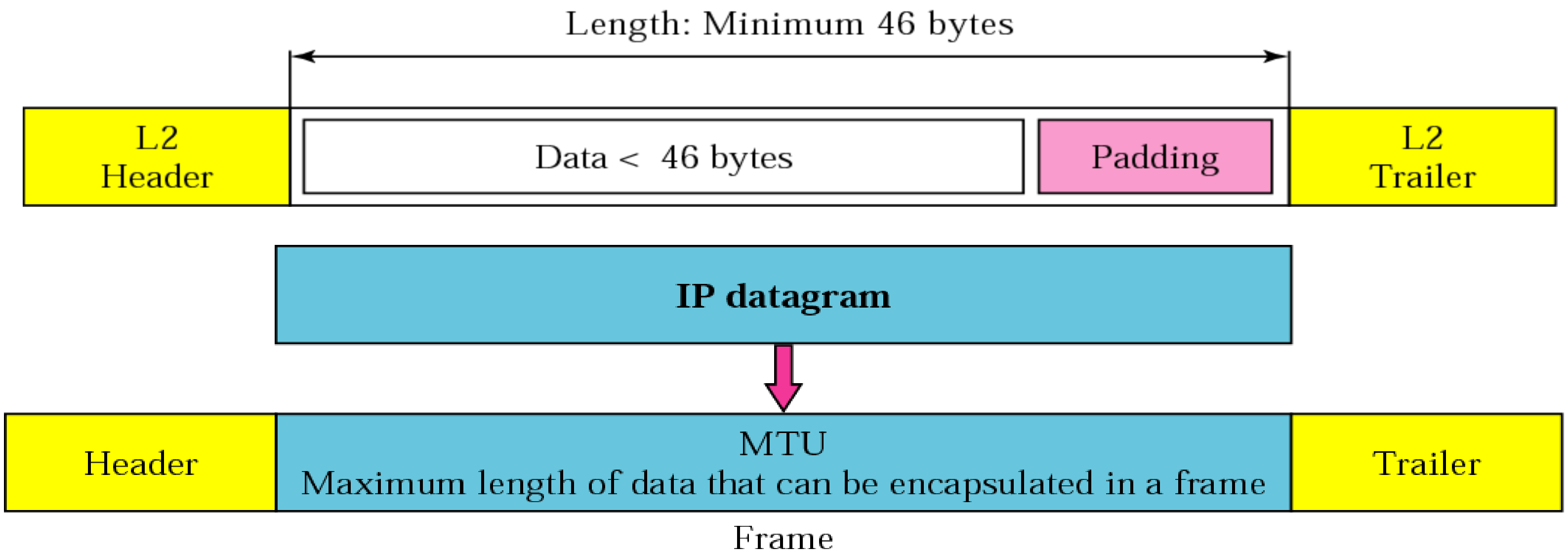
host, router network layer functions:



IP datagram format



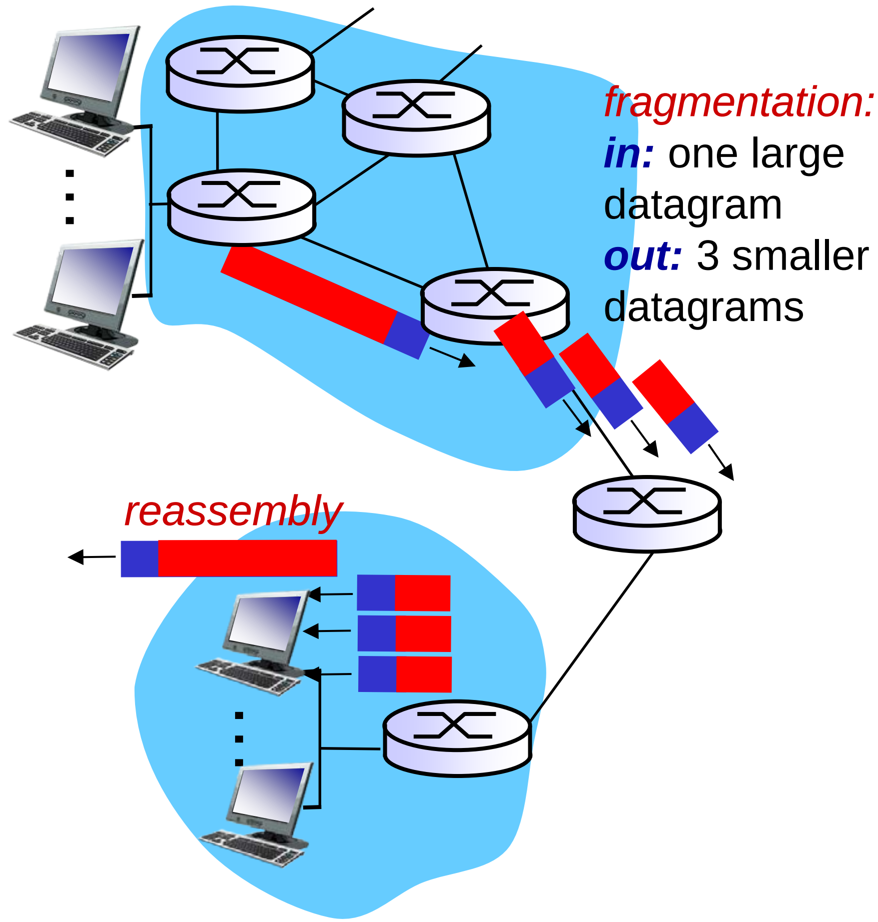
Encapsulation of datagram in frame



Protocol	*MTU*
Hyperchannel	65,535
Token Ring (16 Mbps)	17,914
Token Ring (4 Mbps)	4,464
FDDI	4,352
Ethernet	1,500
X.25	576
PPP	296

IP fragmentation, reassembly

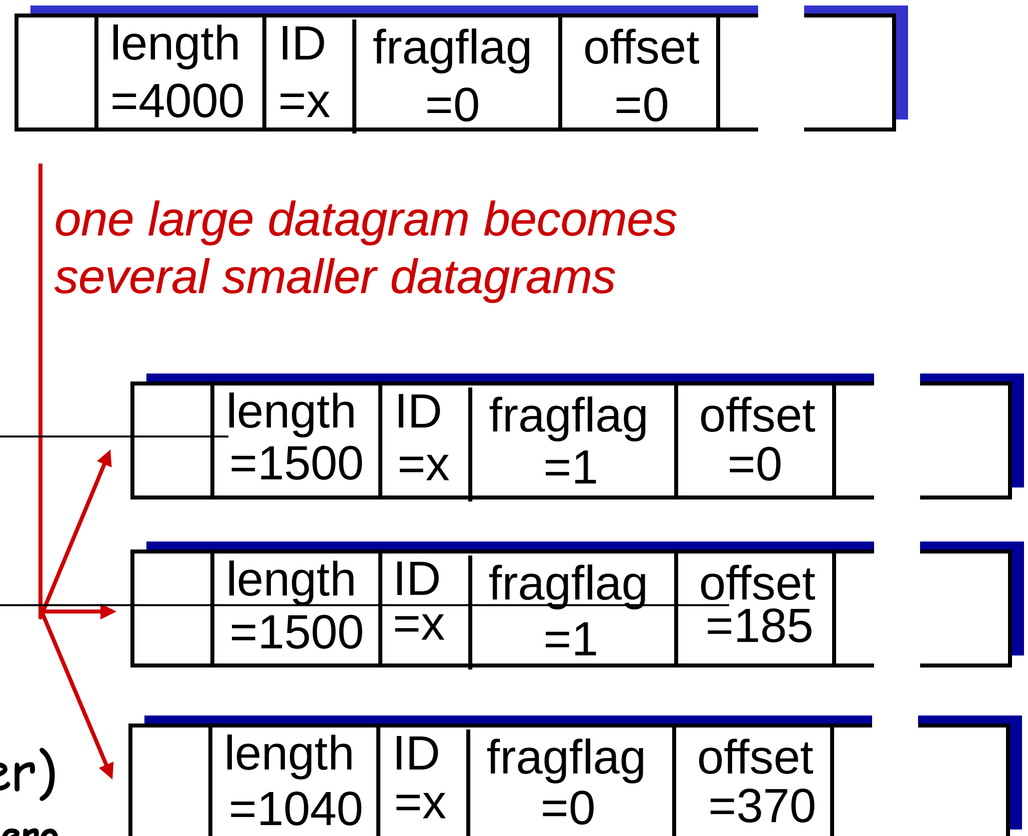
- network links have MTU (max.transfer size) - largest link-level frame
  - different link types, different MTUs
- large IP datagram divided ("fragmented") within net
  - one datagram becomes several datagrams
  - "reassembled" only at final destination
  - IP header bits used to identify, order related fragments



IP fragmentation, reassembly

example:

- 4000 byte datagram
- MTU = 1500 bytes



- Flags (from higher order)
  - **bit 0: Reserved; must be zero**
  - **bit 1: Don't Fragment (DF)**
  - **bit 2: More Fragments (MF)**

Disadvantages of fragmentation

- Lose 1 fragment, lose whole packet
- Kernel has limited buffer space
  - But IP doesn't know # of fragments per packet
  - Example: packets L and S are fragmented into 8 and 2 frames, respectively
  - Receiver has 8 buffer slots, fragments arrive as:

    L1, L2, L3, L4, L5, L6, L7, S1, L8, S2
- Inefficient transmission
  - 10 KB data, sent as 1024 byte TCP segments
  - Suppose MTU is 1006 bytes, each TCP packet is fragmented into 2 frames → sending 20 frames
  - If TCP had sent 966-byte segment, only 11 packets

Solutions of fragmentation

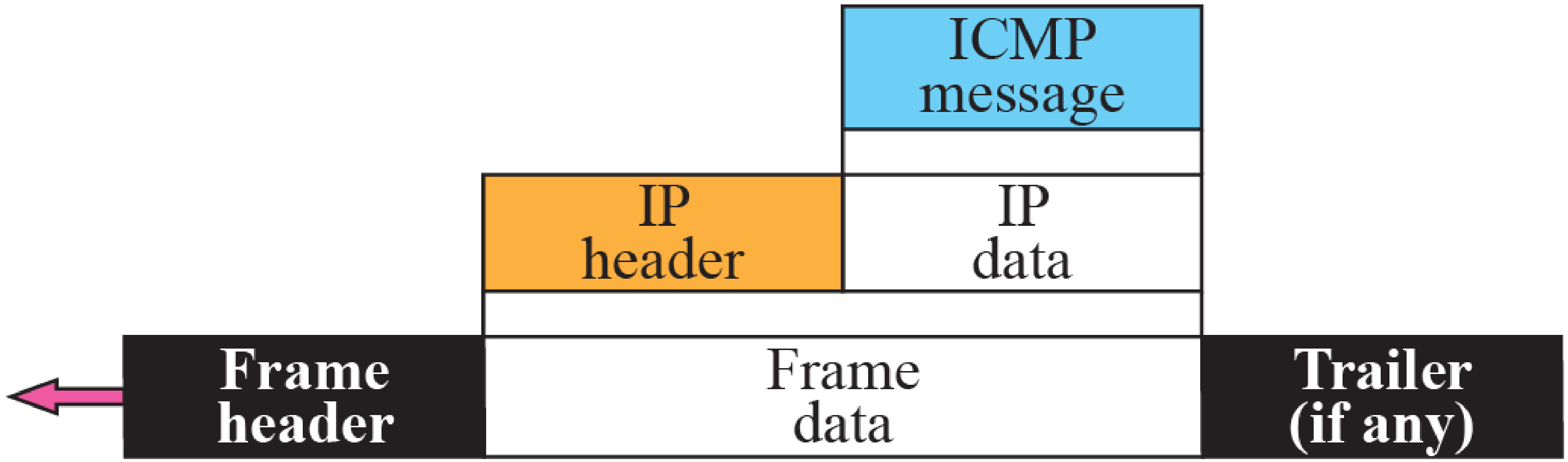
- Analysis
  - IP does not have control over # of fragments per packet
  - TCP can do buffer management better because it has more information
- Alternatives to fragmentation
  - Send only small datagrams (why not?)
  - Do path MTU discovery and let TCP send the appropriate segment sizes
    - Set DF flag
    - Router returns ICMP message (type 3, code 4) if fragmentation needed
  - IPv6 enforces min MTU of 576 bytes, no fragmentation at routers

Internet Control Message Protocol

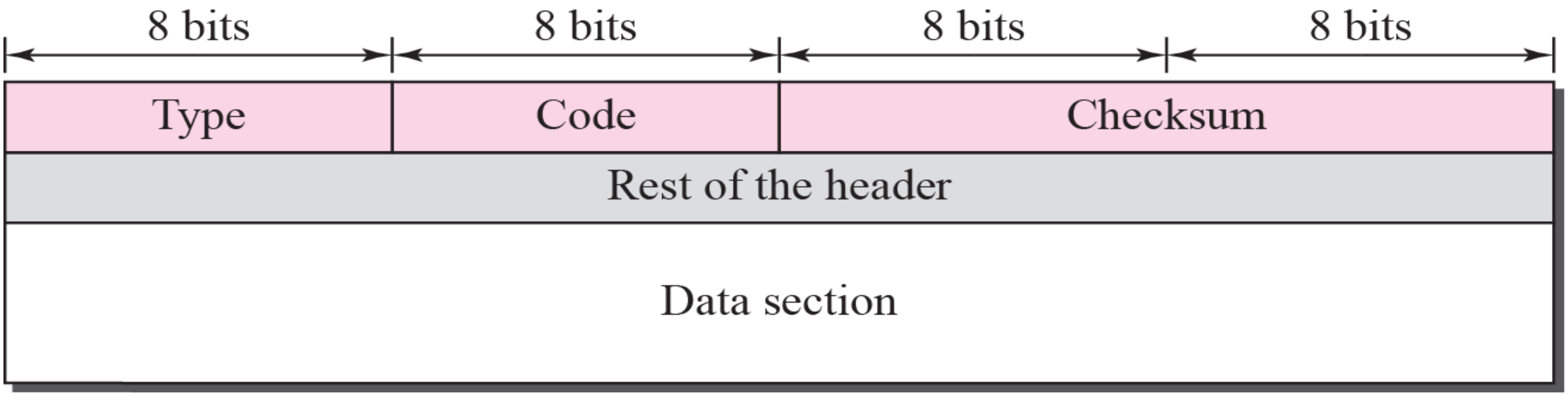
- Why do we need ICMP?
  - What if a router cannot route or deliver a packet?
  - What if a router experiences a congestion?
  - What if the TTL expires?
- Router needs to inform source to take action to avoid or correct the problem
- ICMP is used by host and routers to communicate network-level information
  - report error: unreachable host, net, port, protocol
  - make queries: echo request/reply (used by `ping`)
  - specified in RFC 792

Internet Control Message Protocol

- ICMP runs on network-layer, but "above" IP:
  - ICMP messages are carried inside IP datagrams
  - ICMP can only report condition back to the original source



ICMP message format



- 8 byte header, variable size data section
- format for first 4 bytes of header is common to all ICMP packets
- Type – ICMP message type
- Code – reason for the message type generated

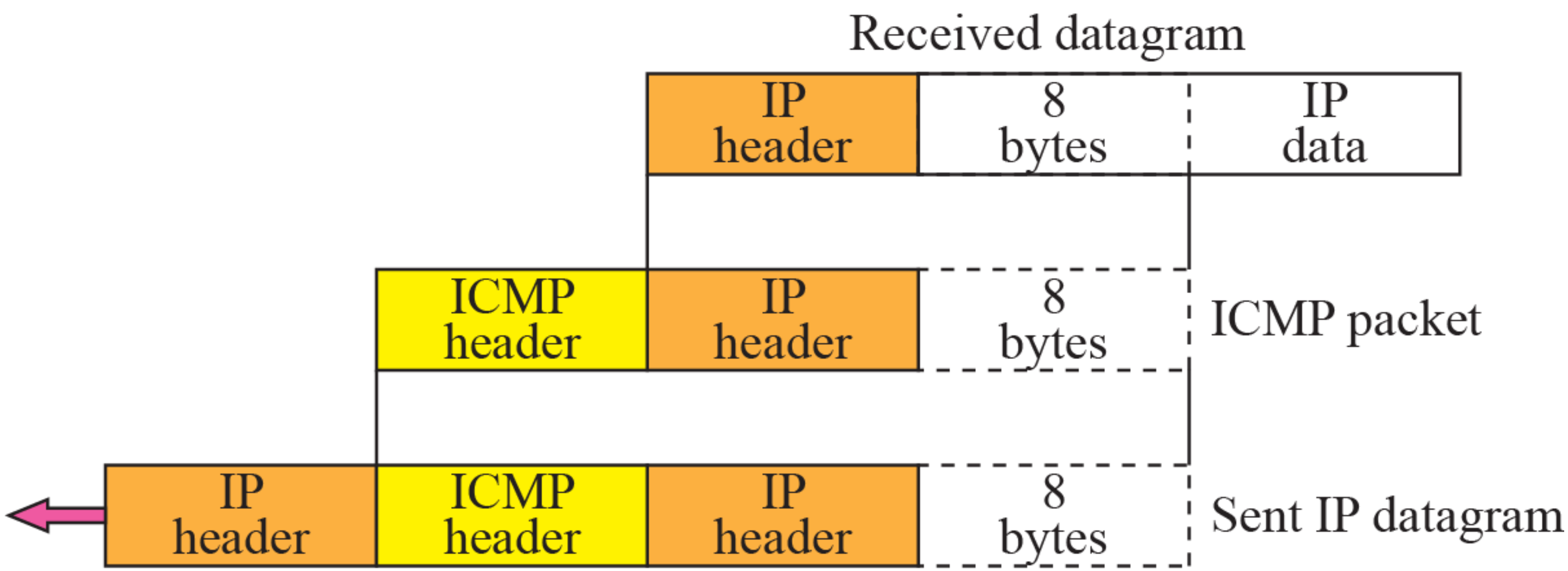
ICMP message types

Category	*Type*	*Message*
Error-reporting messages	3	Destination unreachable
	4	Source quench
	11	Time exceeded
	12	Parameter problem
	5	Redirection
Query messages	8 or 0	Echo request or reply
	13 or 14	Timestamp request or reply

ICMP message type and code

Type	Code	Description
0	0	echo reply (ping)
3	0	dest. network unreachable
3	1	dest. host unreachable
3	2	dest. protocol unreachable
3	3	dest. port unreachable
3	4	frag needed but DF set
3	6	dest. network unknown
3	7	dest. host unknown
8	0	echo request (ping)
9	0	route advertisement
10	0	router discovery
11	0	TTL expired
12	0	bad IP header

Construct ICMP error message



ICMP error messages

- No ICMP error message will be generated
  - in response to a datagram carrying an ICMP error message.
  - for a fragmented datagram that is not the first fragment.
  - for a datagram having a multicast address in the destination.
  - for a datagram whose source address is not a single host, 0.0.0.0, 127.x.x.x, broadcast or multicast address.
- Purpose: prevent Broadcast Storms

ICMP error messages examples

Type: 3	Code: 0 to 15	Checksum
Unused (All 0s)		
Part of the received IP datagram including IP header plus the first 8 bytes of datagram data		

Destination-unreachable format

- Codes:
  - **0: network unreachable**
  - **1: host unreachable**
  - **2: protocol unreachable**
  - **3: port unreachable**
  - **4: need fragmentation but DF flag is set**
  - …….
- Who generates these messages, hosts or routers?

ICMP error messages examples

Type: 4	Code: 0	Checksum
Unused (All 0s)		
Part of the received IP datagram including IP header plus the first 8 bytes of datagram data		

Source-quench format

- A source-quench message informs the source that a datagram has been discarded due to congestion in a router or the destination host.
- The source must slow down the sending of datagrams until the congestion is relieved.
- One message sent for each datagram discarded.

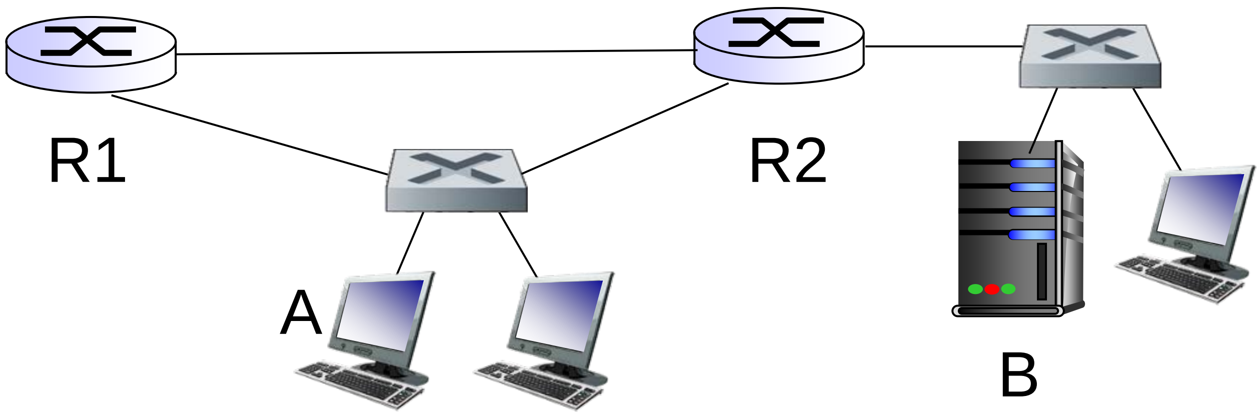
ICMP error messages examples

Type: 11	Code: 0 or 1	Checksum
Unused (All 0s)		
Part of the received IP datagram including IP header plus the first 8 bytes of datagram data		

Time-exceeded message format

- Code 0: When a router decreases a datagram's TTL to zero, it discards the datagram and sends a time-exceeded message to the original source.
- Code 1: When the final destination does not receive all of the fragments in a set time, it discards the received fragments and sends a time-exceeded message to the original source.

ICMP error messages examples



- Host also has a routing table when it is directly connected to multiple routers (multi-homing)
- Routing update process only for routers, not hosts
- Host uses static routing, e.g., host A has a default gateway R1.
- When A wants send packets to B, it sends it to R1.

ICMP error messages examples

Type: 5	Code: 0 to 3	Checksum
IP address of the target router		
Part of the received IP datagram including IP header plus the first 8 bytes of datagram data		

Redirection message format

- Redirection for
  - **Code 0: a network (obsolete)**
  - **Code 1: a host**
  - **Code 2: a specified type of service (ToS) and network**
  - **Code 3: a specified type of service (ToS) and host**
- Sent from a router to a host in the same LAN.

ICMP query messages

- Diagnose network problems through the query messages (two pairs of messages)
  - Echo request/reply
  - Timestamp request/reply
- A query is answered in a specific format by the destination node.

ICMP query messages examples

13: request
14: reply

Type: 13 or 14	Code: 0	Checksum
Identifier		Sequence number
Original timestamp		
Receive timestamp		
Transmit timestamp		

Timestamp-request and timestamp-reply message format

- Sender puts original timestamp
- Receiver copies original timestamp and put
  - **Receive timestamp upon receiving**
  - **Transmit timestamp upon sending it back**

ICMP query messages examples

- Timestamp query messages are used to
  - synchronize the clocks in two machines
  - determine the round-trip time between nodes
  - which of the above task is easier?

Sending time = receive timestamp – original timestamp

Receiving time = returned time – transmit timestamp

Round-trip time = Sending time + Receiving time

- What if nodes' clocks are not synchronized?

Calculating round-trip time

- Example:
  - Original timestamp: 46
  - Receive timestamp: 59
  - Transmit timestamp: 60
  - Return time: 67

  - Sending time = 59 – 46 = 13 milliseconds
  - Receiving time = 67 – 60 = 7 milliseconds
  - Round-trip time = 13 + 7 = 20 milliseconds

- Round-trip time can be accurately calculated even the clocks are not synchronized!

Synchronizing clocks

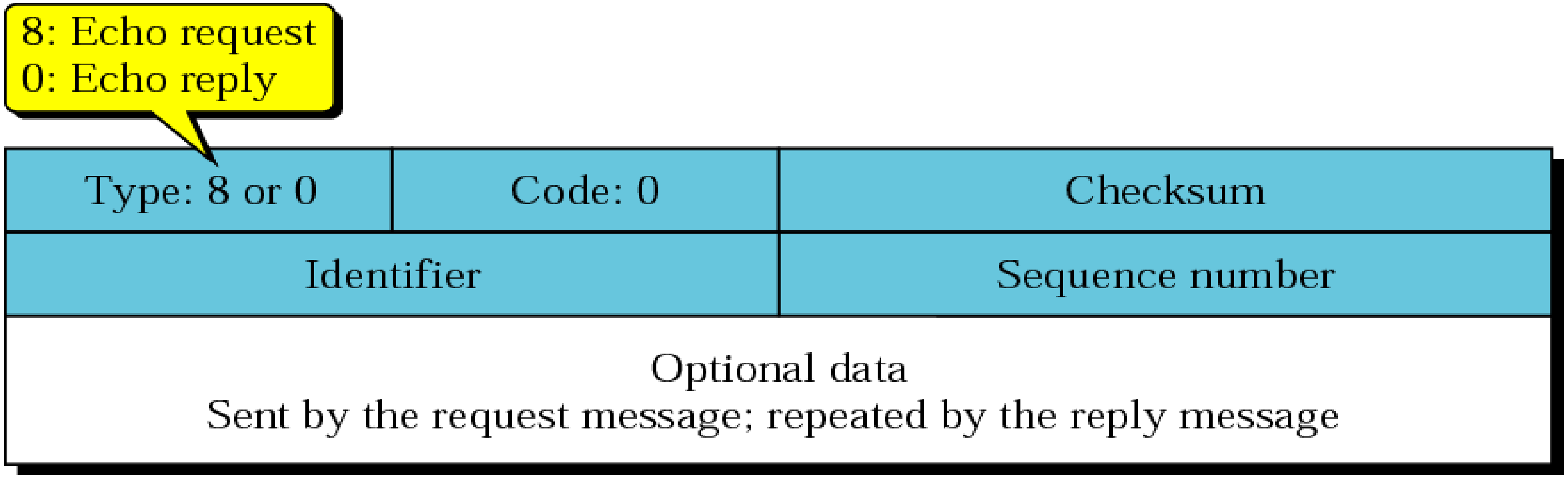
- Formula:

  Time difference = receive timestamp –
  (original timestamp + one-way time duration)

- Example:
  - Original timestamp: 46
  - Receive timestamp: 59
  - Transit timestamp: 60
  - Return time: 67
  - Time difference = 59 – (46 + 10) = 3 milliseconds

- One-way time duration can be estimated from round-trip time

ICMP query messages examples



Echo-request and echo-reply message format

- Host or router can send an echo-request message to another host or router
- Combination of request and reply can be used to determine if two systems can communicate, e.g., determine reachability of a host used by ***ping***.

Ping: testing reachability

```
$ ping fhda.edu
PING fhda.edu (153.18.8.1) 56 (84) bytes of data.
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=0 ttl=62 time=1.91 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=1 ttl=62 time=2.04 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=2 ttl=62 time=1.90 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=3 ttl=62 time=1.97 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=4 ttl=62 time=1.93 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=5 ttl=62 time=2.00 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=6 ttl=62 time=1.94 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=7 ttl=62 time=1.94 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=8 ttl=62 time=1.97 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=9 ttl=62 time=1.89 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=10 ttl=62 time=1.98 ms

--- fhda.edu ping statistics ---
11 packets transmitted, 11 received, 0% packet loss, time 10103ms

rtt min/avg/max = 1.899/1.955/2.041 ms
```

Traceroute: finding the route

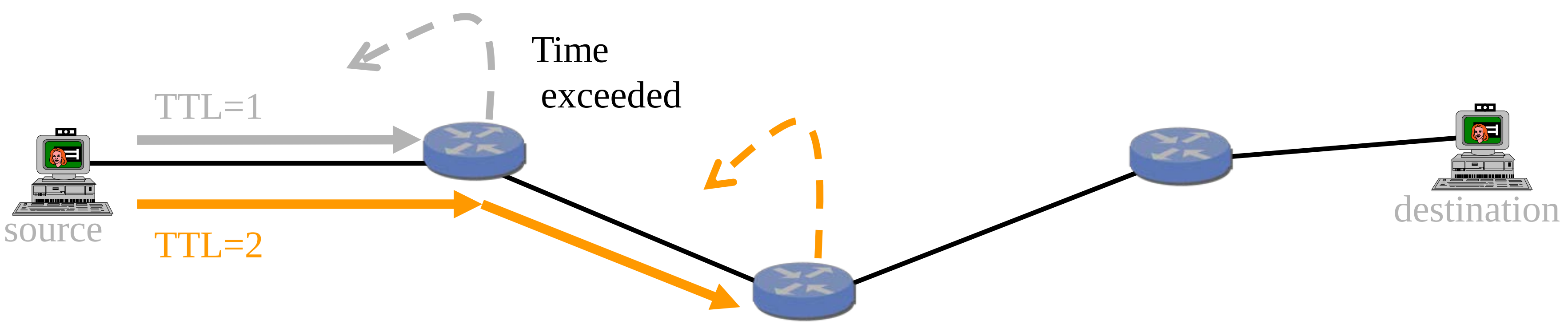
```
$ traceroute xerox.com
traceroute to xerox.com (13.1.64.93), 30 hops max, 38 byte packets
1 Dcore.fhda.edu (153.18.31.254) 0.622 ms 0.891 ms 0.875 ms
2 Ddmz.fhda.edu (153.18.251.40) 2.132 ms 2.266 ms 2.094 ms
...

18 alpha.Xerox.COM (13.1.64.93) 11.172 ms 11.048 ms 10.922 ms
```

- 20 bytes IP header + 8 bytes UDP header + 10 bytes of application data = 38 bytes
- Each entry shows 3 sample round-trip times from source to the router

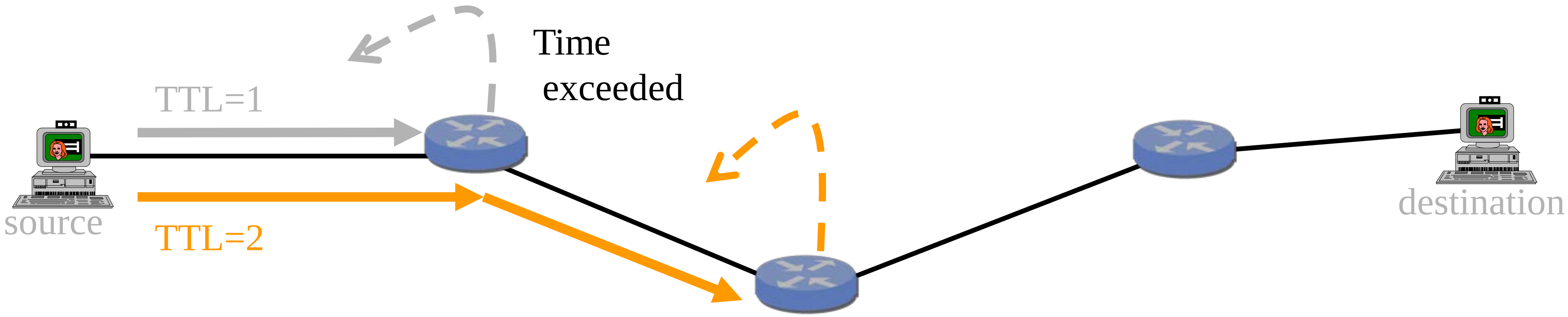
Traceroute: Exploiting TTL

- Host sends a series of UDP packets to destination
  - first 3 packets have TTL set to 1,
  - next 3 packets have TTL set to 2, etc.
  - each router decrements the time-to-live field
- If time-to-live field reaches 0
  - router sends a "TTL expired" message (type 11, code 0) back to the source



Traceroute: Exploiting TTL

- When ICMP arrives back at the source, source calculates round-trip time (RTT)
- Stopping criterion
  - UDP packets eventually arrive at destination host
  - destination returns ICMP "destination port unreachable" message (type 3, code 3)
  - when source gets this ICMP message, it stops sending UDP packets



IPv6

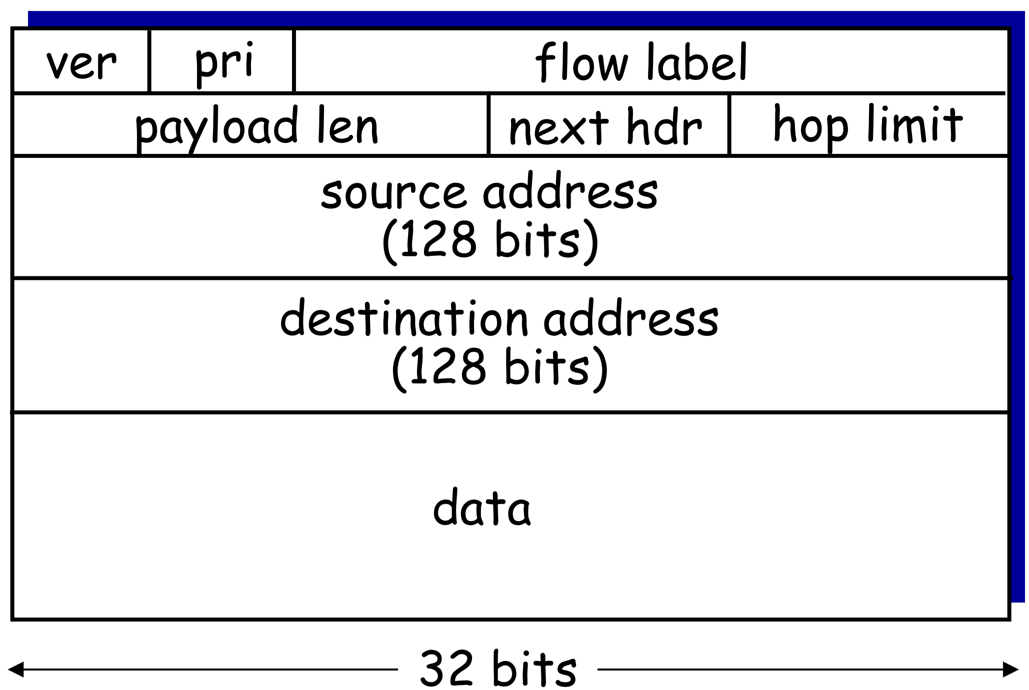
- Initial motivation: 32-bit address space soon to be completely allocated.

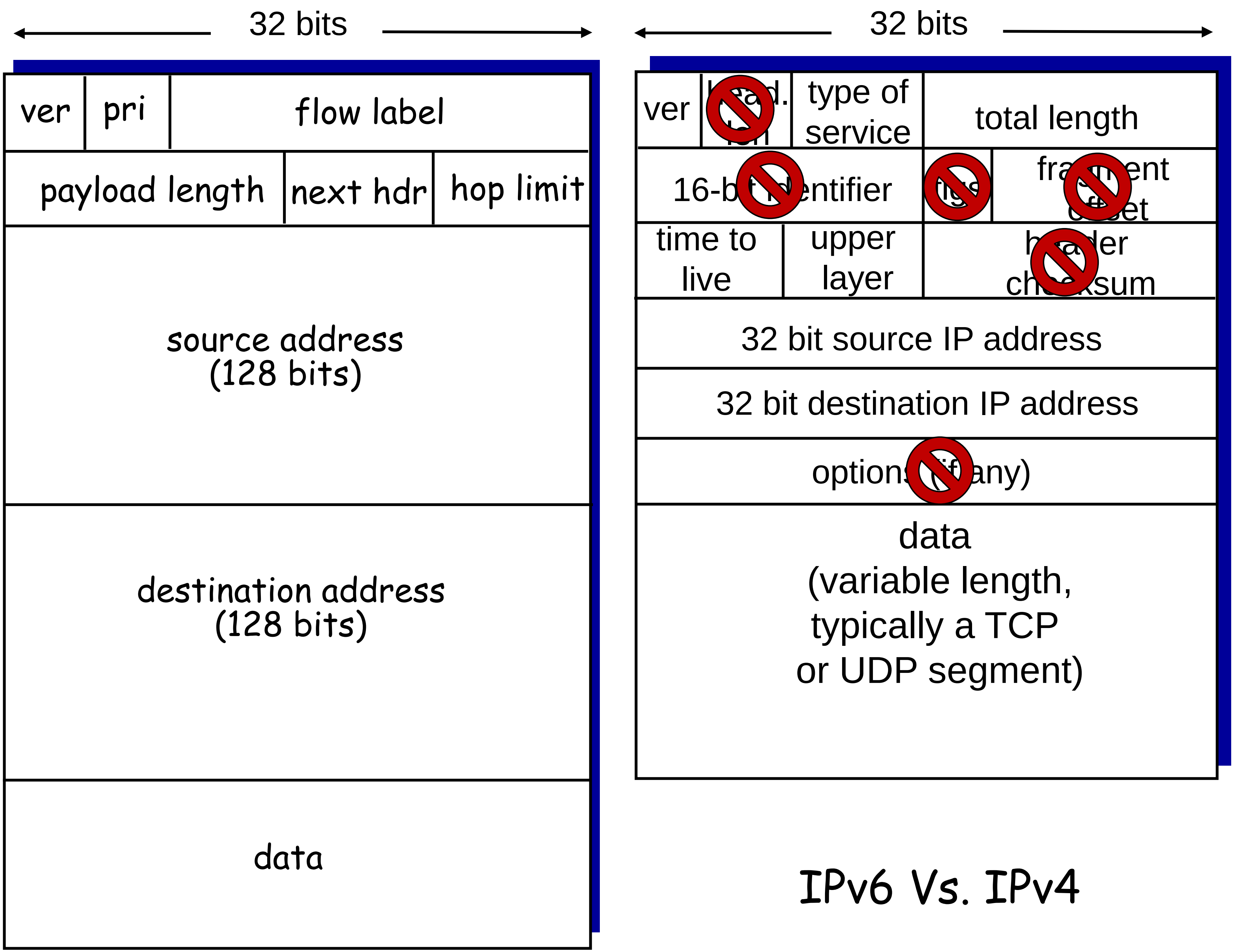
- Additional motivation:
  - header format helps speed processing/forwarding
  - header changes to facilitate QoS

  IPv6 datagram format:
  - fixed-length 40 byte header
  - no fragmentation allowed

IPv6 Header

Priority: identify priority among datagrams in flow
Flow Label: identify datagrams in same "flow."
(concept of "flow" not well defined).
Next header: identify upper layer protocol for data




32 bits
ver
pri
flow label
payload length
next hdr
hop limit
source address
(128 bits)
destination address
(128 bits)
data
32 bits
ver
type of
service
total length
time to
live
upper
layer
32 bit source IP address
32 bit destination IP address
data
(variable length,
typically a TCP
or UDP segment)

IPv6 Vs. IPv4

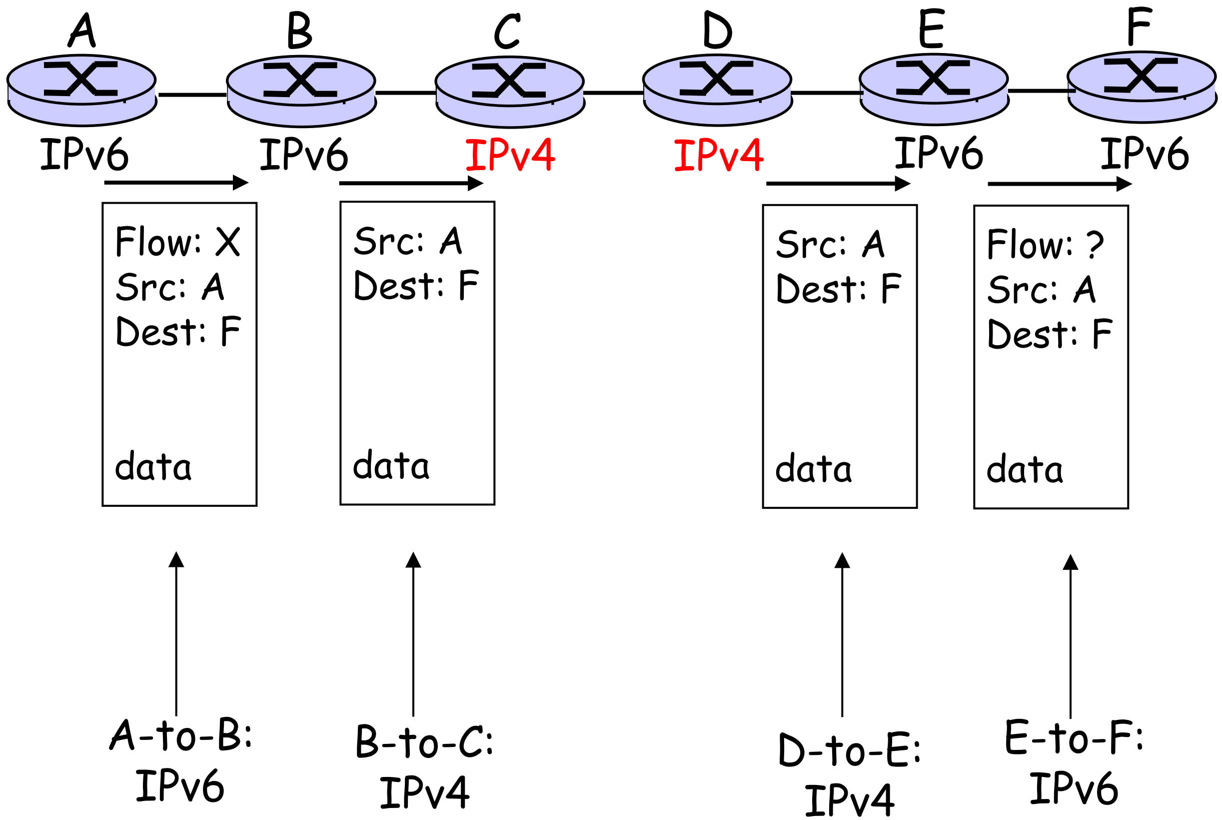
Other Changes from IPv4

- *Checksum*: removed entirely to reduce processing time at each hop
- *Options:* allowed, but outside of header, indicated by "Next Header" field
- *ICMPv6:* new version of ICMP
  - additional message types, e.g. "Packet Too Big"
  - multicast group management functions

Transition From IPv4 To IPv6

- Flag-day approach
  - Used by upgrading NCP to TCP 30 years ago
  - Not all routers can be upgraded simultaneous
- Dual-stack approach
  - How will the network operate with mixed IPv4 and IPv6 routers?

Dual-stack approach

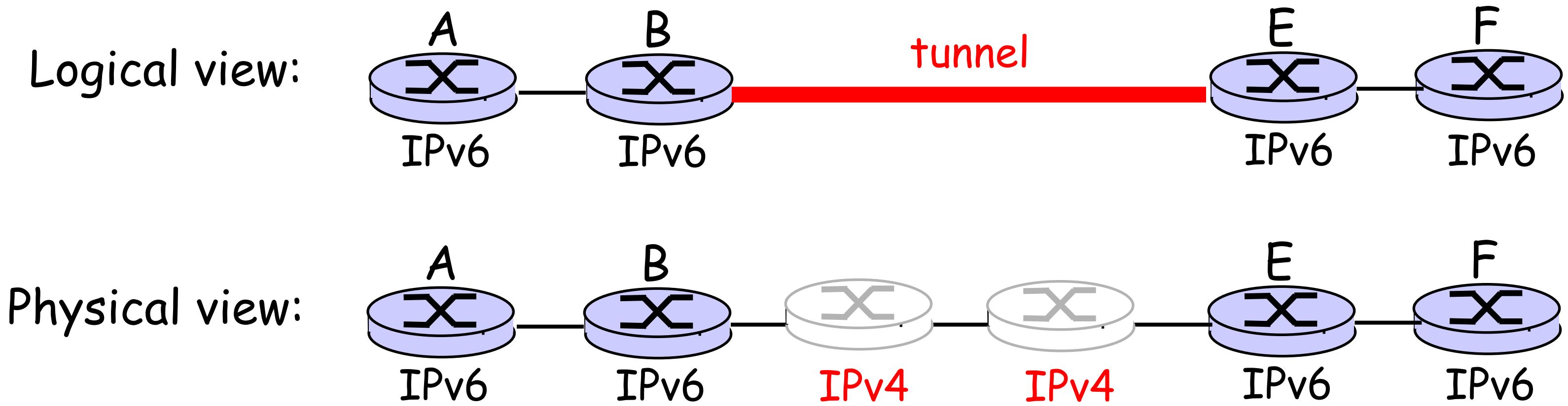


- IPv6 specific information lost in IPv4 routers!

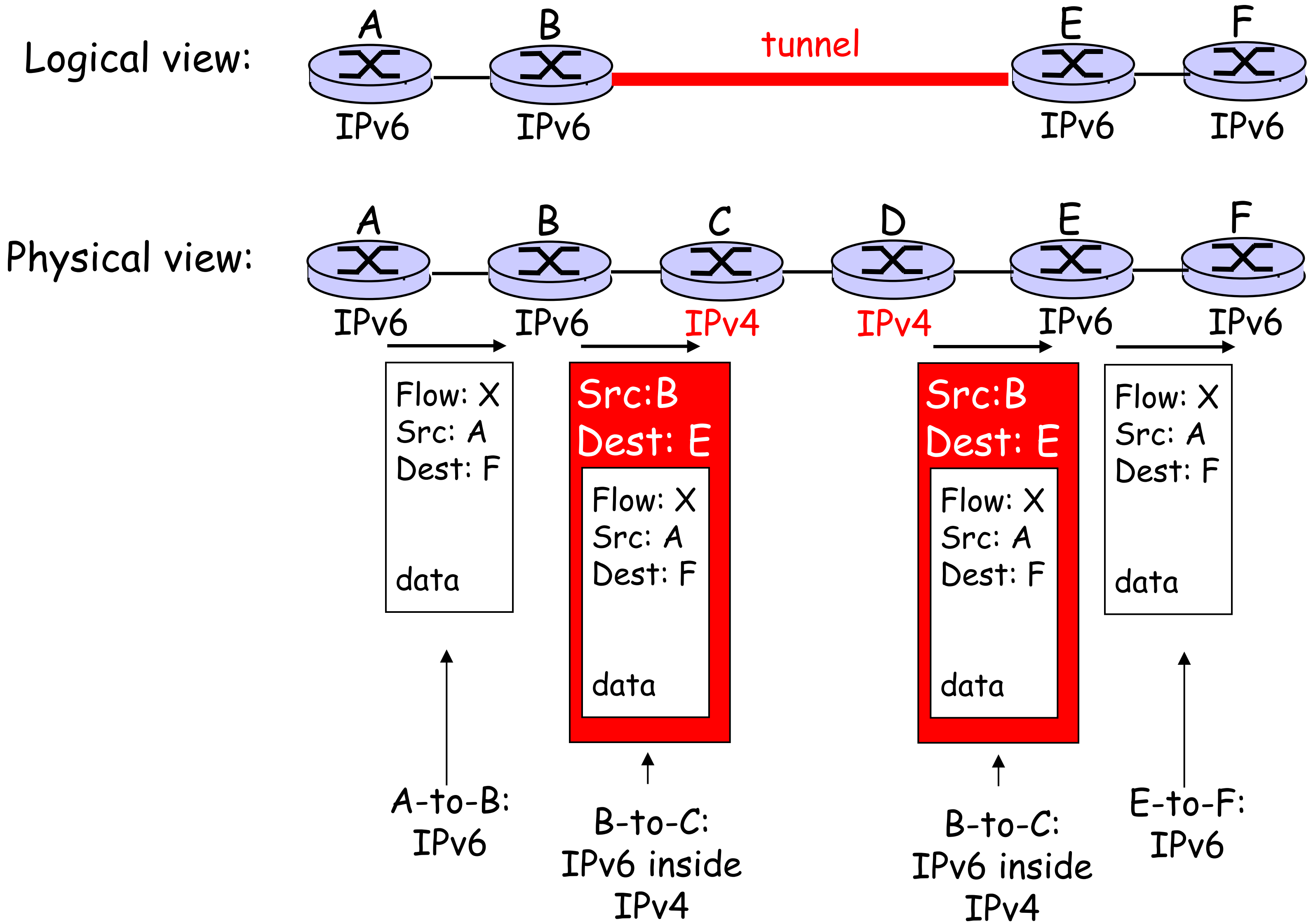
Transition From IPv4 To IPv6

- Flag-day approach
  - Used by upgrading NCP to TCP 30 years ago
  - Not all routers can be upgraded simultaneous
- Dual-stack approach
  - How will the network operate with mixed IPv4 and IPv6 routers?
- Tunneling approach
  - IPv6 carried as payload in IPv4 datagram among IPv4 routers

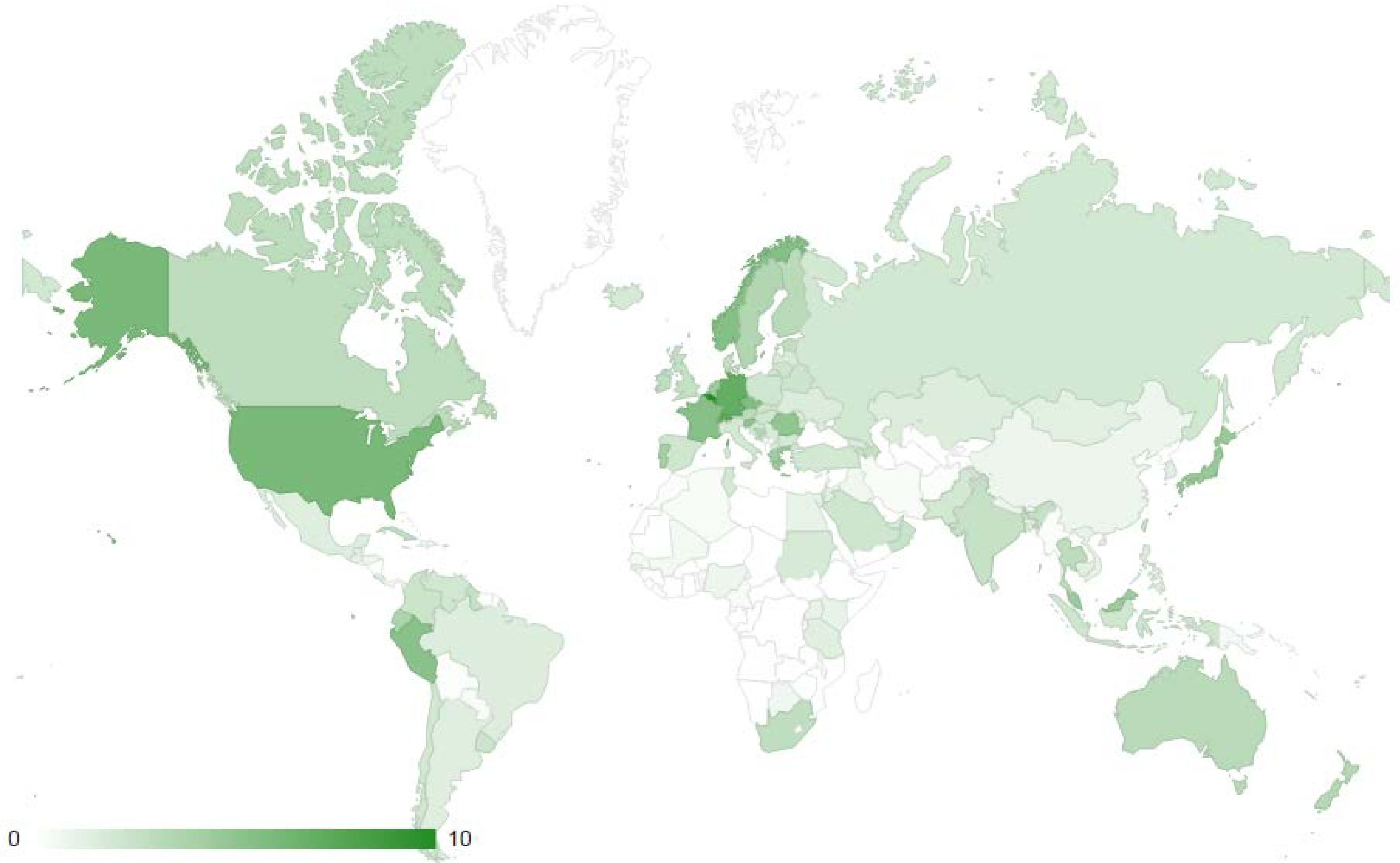
Tunneling approach



Tunneling approach



IPv6 adoption problem



http://6lab.cisco.com/stats/