

A decorative graphic consisting of a vertical black line and a horizontal grey line intersecting at a point to the left of the text.

## EECS 122: Introduction to Communication Networks

### Unit 15a IP add-ons

## Acknowledgements – slides coming from:

- **The book by Peterson/Davie**
- **The book by William Stallings**
- Several slides from the earlier issues of the EECS 122 taught by Prof Jean Walrand, A series of slides from lectures by **Peter Steenkiste** (CMU) (special acknowledgement for the illustration of routing algorithms!), also Anja Feldmann (TU Berlin), Nick McKeown (Stanford) , and D. Peterson (Princeton)



# Supporting Mobility

# Motivation for Mobile IP

- Routing

- based on IP destination address, network prefix (e.g. 129.13.42) determines physical subnet
- change of physical subnet implies change of IP address to have a topological correct address (standard IP) or needs special entries in the routing tables

- Specific routes to end-systems?

- change of all routing table entries to forward packets to the right destination
- does not scale with the number of mobile hosts and frequent changes in the location, security problems

- Changing the IP-address?

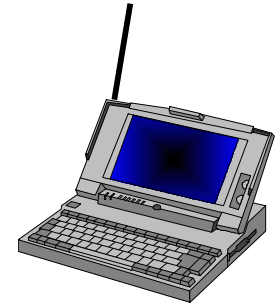
- adjust the host IP address depending on the current location
- almost impossible to find a mobile system, DNS updates take to long time

- TCP connections break, security problems

# Requirements to Mobile IP (RFC 2002)

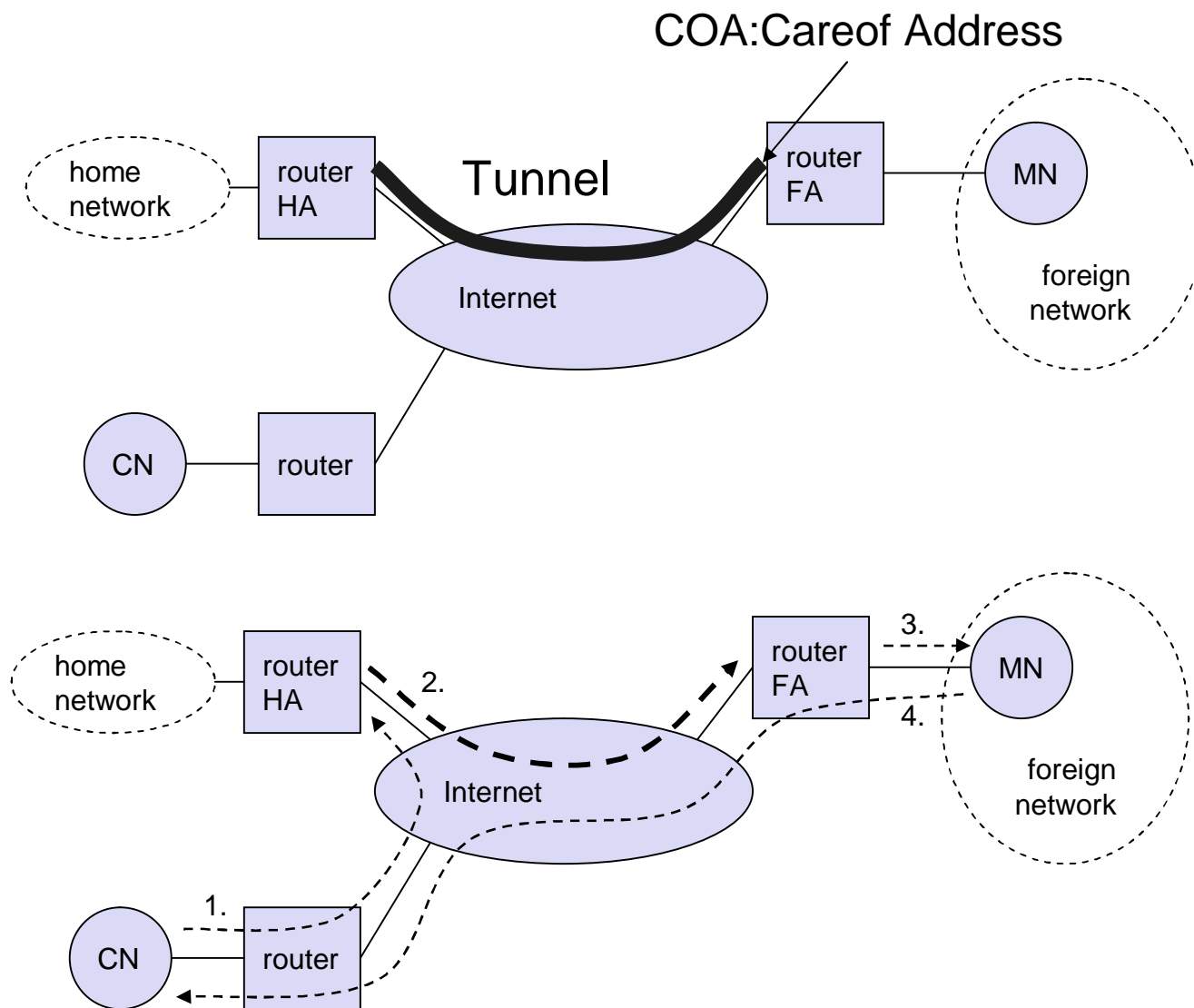
- Transparency
  - mobile end-systems keep their IP address
  - continuation of communication after interruption of link possible
  - point of connection to the fixed network can be changed
- Compatibility
  - usage of the same layer 2 protocols as IP
  - no changes to current end-systems and routers required
  - mobile end-systems can communicate with fixed systems
- Security
  - authentication of all registration messages
- Efficiency and scalability
  - only little additional messages to the mobile system required (connection typically via a low bandwidth radio link)
  - world-wide support of a large number of mobile systems in the whole Internet

# Terminology



- Mobile Node (MN)
  - system (node) that can change the point of connection to the network without changing its IP address
- Home Agent (HA)
  - system in the home network of the MN, typically a router
  - registers MN's location, tunnels IP datagrams to the COA
- Foreign Agent (FA)
  - system in the foreign network of the MN, typically a router
  - forwards the tunneled datagrams to the MN, typically also the default router for the MN
- Care-of Address (COA)
  - address of the current tunnel end-point for the MN (at FA or MN)
  - actual location of the MN from an IP point of view
  - can be chosen, e.g., via DHCP
- Correspondent Node (CN)
  - communication partner

# Data Transfer to the mobile system



## How does it work...

- Agent Advertisement

- HA and FA periodically send advertisement messages into their physical subnets
- MN listens to these messages and memorizes the HA
- -----➔ MN moves
- MN detects the Advertisement of the FA
- MN reads a COA from the FA advertisement messages

- Registration

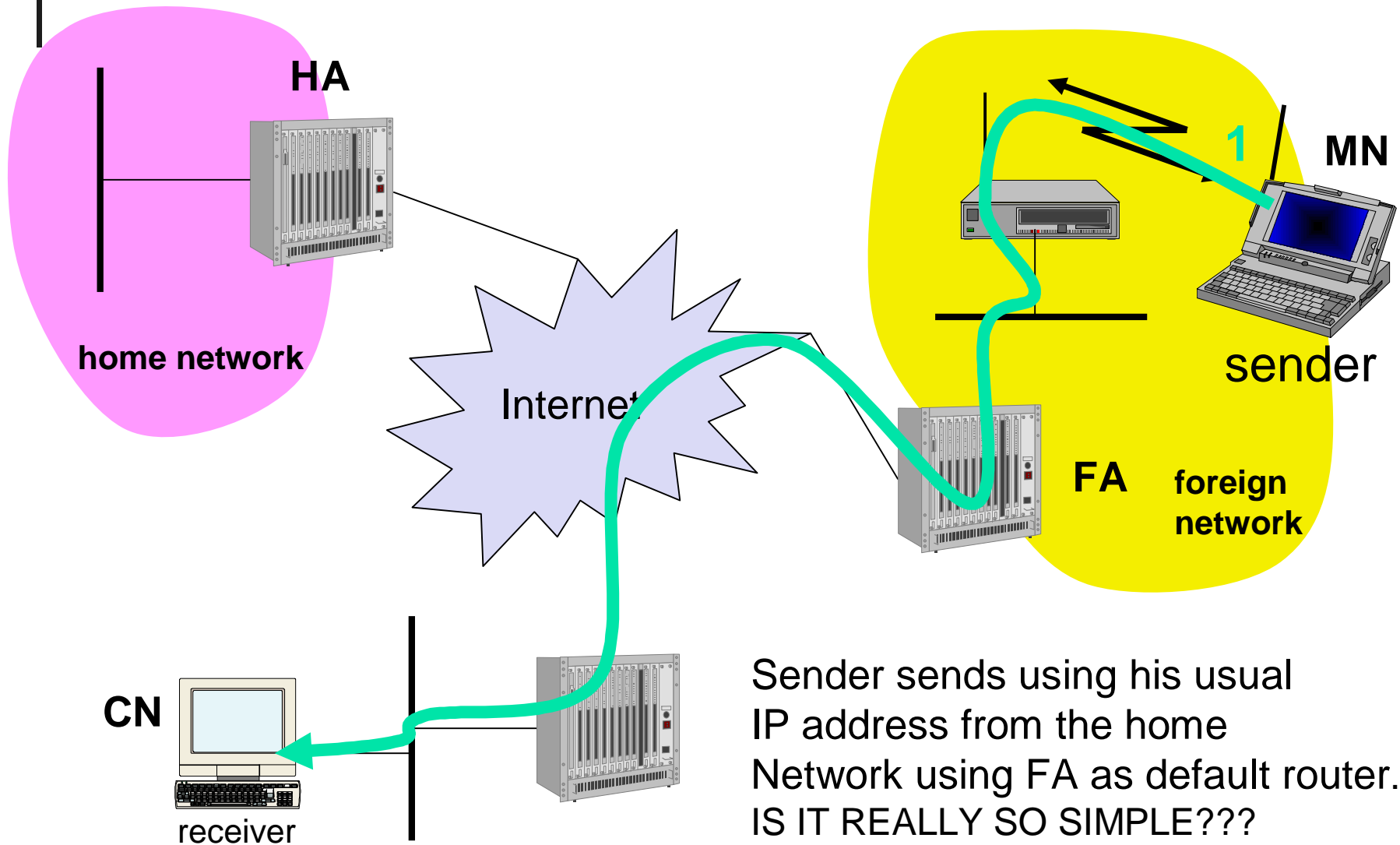
- MN registers within the FA (soft state)
- MN registers his move at the HA, disclosing the COA of the FA
- HA acknowledges via FA to MN

- Advertisement of the MN IP address by HA

- Within the home network HA handles the ARP requests for IP address of MN answering with the own MAC address
- To all others HA advertises the IP address of the MN in the home network.



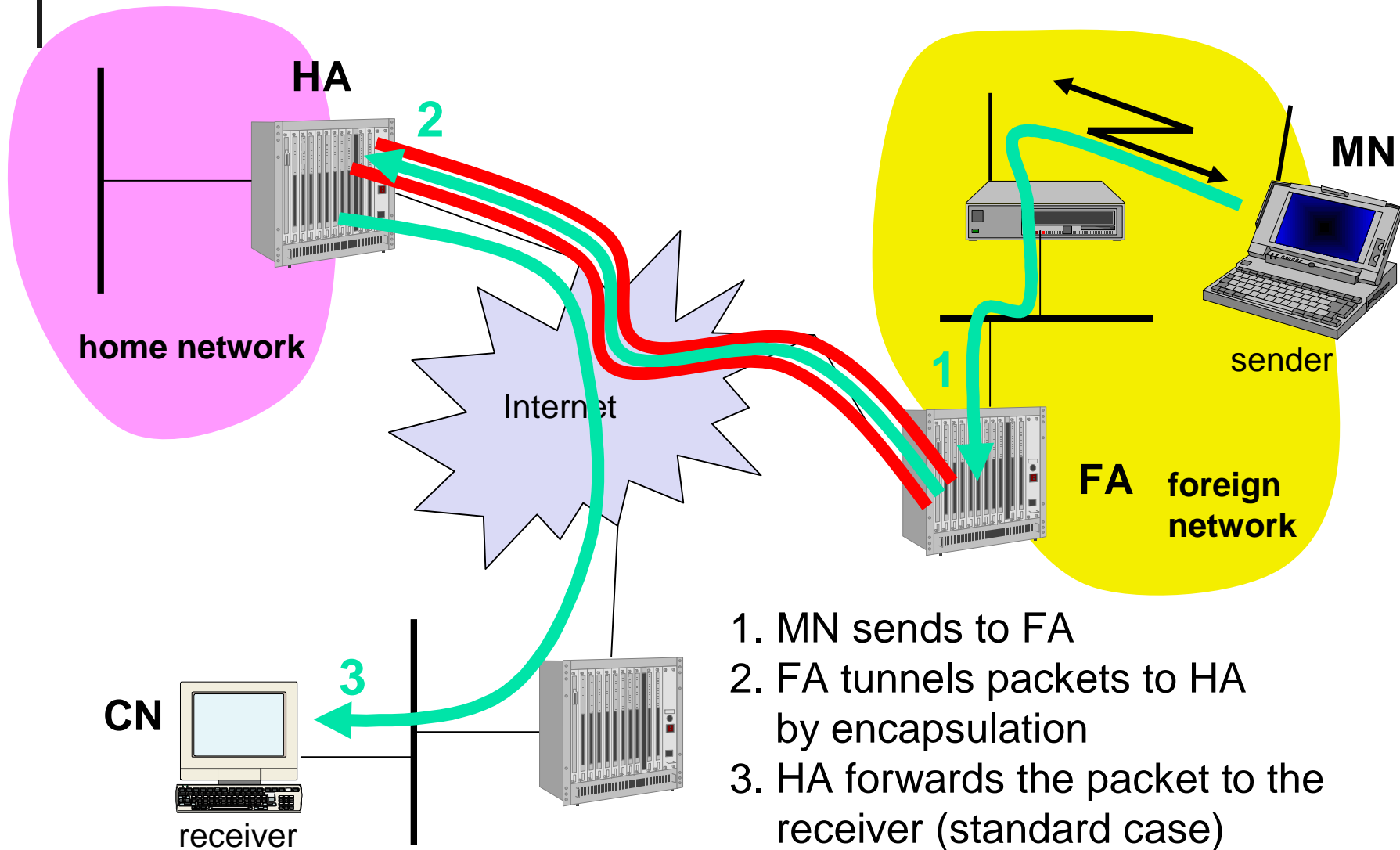
# Data transfer from the mobile system



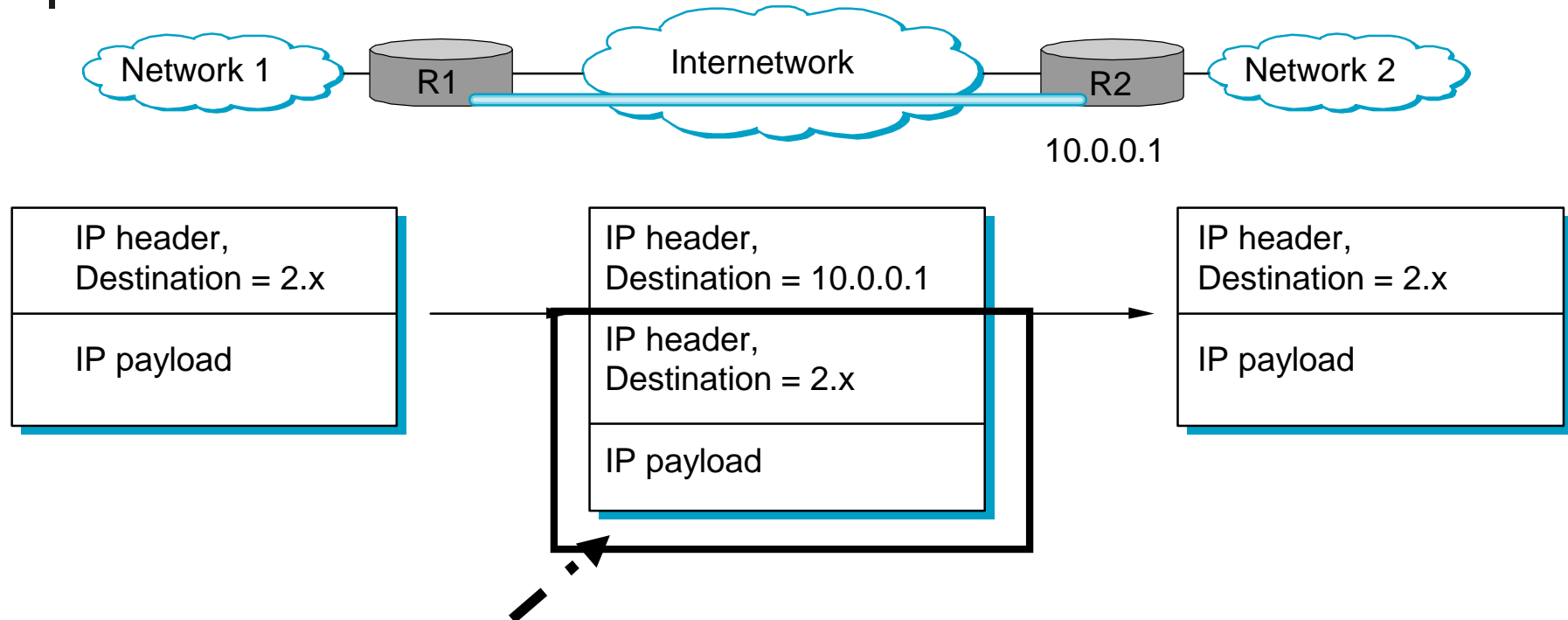
## Mobile IP with reverse tunneling

- Sending packet with his “old” IP address MN behaves topologically incorrect. Some firewalls might block this
- Reverse tunneling – solves the problem!
  - a packet from the MN with his IP address and the IP Address of the corresponding host is posted to the FA Using it MAC address.
  - FA encapsulates it with its IP Address and destination address HA -  
**→ Tunnel (IP packet in IP Packet!!!)**
  - Side effects: multicast and TTL problems solved (TTL in the home network correct, but MN is to far away from the receiver)
- Reverse tunneling does not solve
  - Security considerations completely, the reverse tunnel can be abused to circumvent security mechanisms (tunnel hijacking)
  - optimization of data paths, i.e. packets will be forwarded through the tunnel via the HA to a sender (double triangular routing)

# Reverse tunneling (RFC 2344)



## More about tunneling ... IMPORTANT



The complete (original header and payload) are now considered „new payload!“. The „new payload“ can be completely encrypted, and hidden from any kind of observation.

Router R1 can consider Network 2 as directly connected. It might be a subnetwork of location 1.

Virtual private networks use tunnels to connect locations....



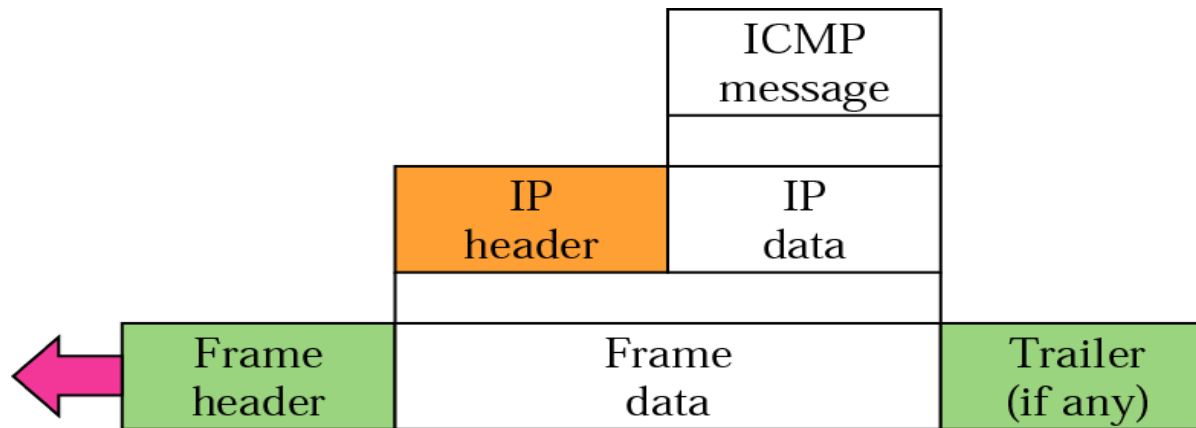
# ICMP

# ICMP – Internet Control Message Protocol

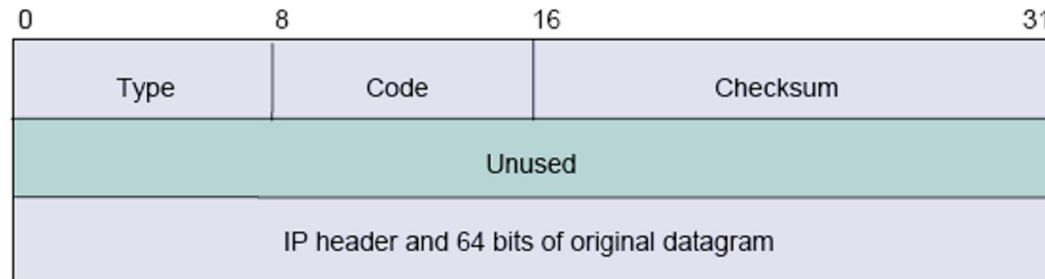
- Network-layer “above” IP:
- **ICMP Used by hosts, routers, gateways to communication network-level information**
  - **error reporting: unreachable host, network, port, protocol**
  - **echo request/reply (used by ping)**
- Other functions associated with ICMP are:
  - Reachability testing
  - Congestion Control
  - Route change information
  - Performance measuring
  - Subnet addressing

# ICMP – Internet Control Message Protocol

- ICMP messages are carried in IP datagrams



- **ICMP error message:** type, code plus first 8 bytes of IP datagram causing error



# ICMP - messages

Type	ICMP Message Type
0	Echo Reply
3	Destination Unreachable
4	Source Quench
5	Redirect (change a route)
8	Echo Request
11	Time Exceeded for a Datagram
12	Parameter problem on datagram
13	Timestamp Request
14	Timestamp Reply
15	Information request (obsolete)
16	Information reply (obsolete)
17	Address mask Request
18	Address Mask Reply

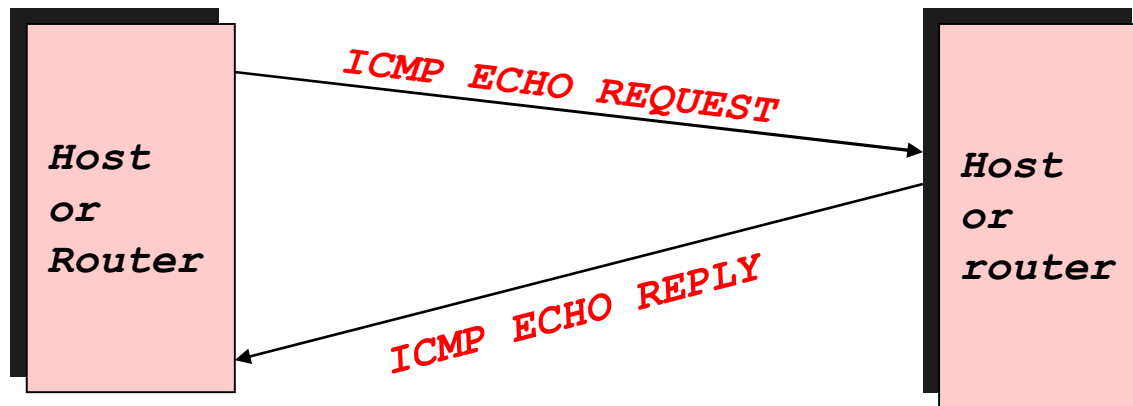


## ICMP – destination unreachable

Code Value	Meaning
0	Network unreachable
1	Host unreachable
2	Protocol unreachable
3	Port unreachable
4	Fragmentation needed and DF set
5	Source route failed
6	Destination network unknown
7	Destination host unknown
8	Source host isolated
9	Communication with destination network administratively prohibited
10	Communication with destination host administratively prohibited
11	Network unreachable for type of service
12	Host unreachable for type of service

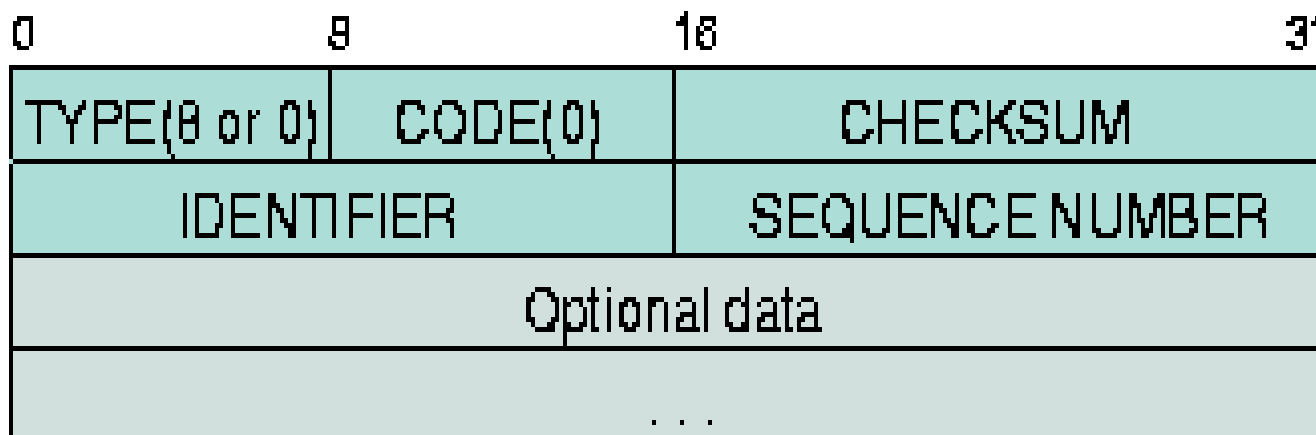
## Ping:

- Ping's are handled directly by the kernel
  - Each Ping is translated into an **ICMP Echo Request**
  - The Ping'ed host responds with an **ICMP Echo Reply**
- Executed three times...



## Ping – details...

“ Used to detect a hosts reachability by exchanging a request-response pair of ICMP packets



Type is 8 for request/0 for reply

Code is 0 (not used)

Identifier allows to differentiate different sessions

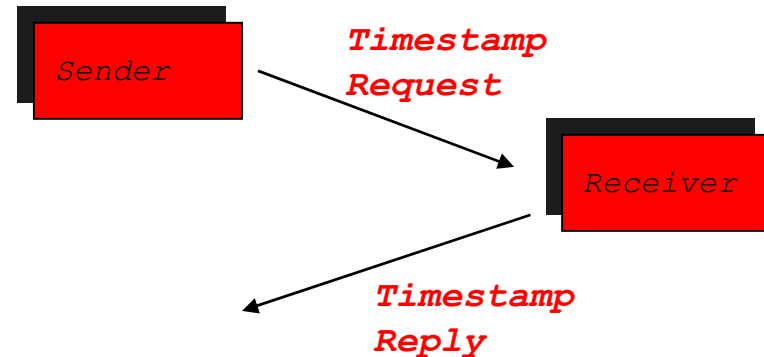
Sequence number . Supports matching the answer to the request.

# Traceroute and ICMP

- Source sends series of UDP segments to dest.  
(*UDP just IP plus something*😊)
    - First has TTL = 1
    - Second has TTL = 2, etc.
    - Unlikely port number
  - When n<sup>th</sup> datagram arrives to n<sup>th</sup> router:
    - Router discards datagram
    - And sends to source an ICMP message (type 11, code 0)
    - Message includes name of router & IP address
  - When ICMP message arrives, source calculates RTT
  - Traceroute does this 3 times
- Stopping criterion
- UDP segment eventually arrives at destination host
  - Destination returns ICMP “host unreachable” packet (type 3, code 3)
  - When source gets this ICMP, stops.

## Example of a Query: ICMP Timestamp

- A system (host or router) asks another system for the current time.
- Time is measured in milliseconds after midnight UTC (Universal Coordinated Time) of the current day
- Sender sends a **request**, receiver responds with **reply**



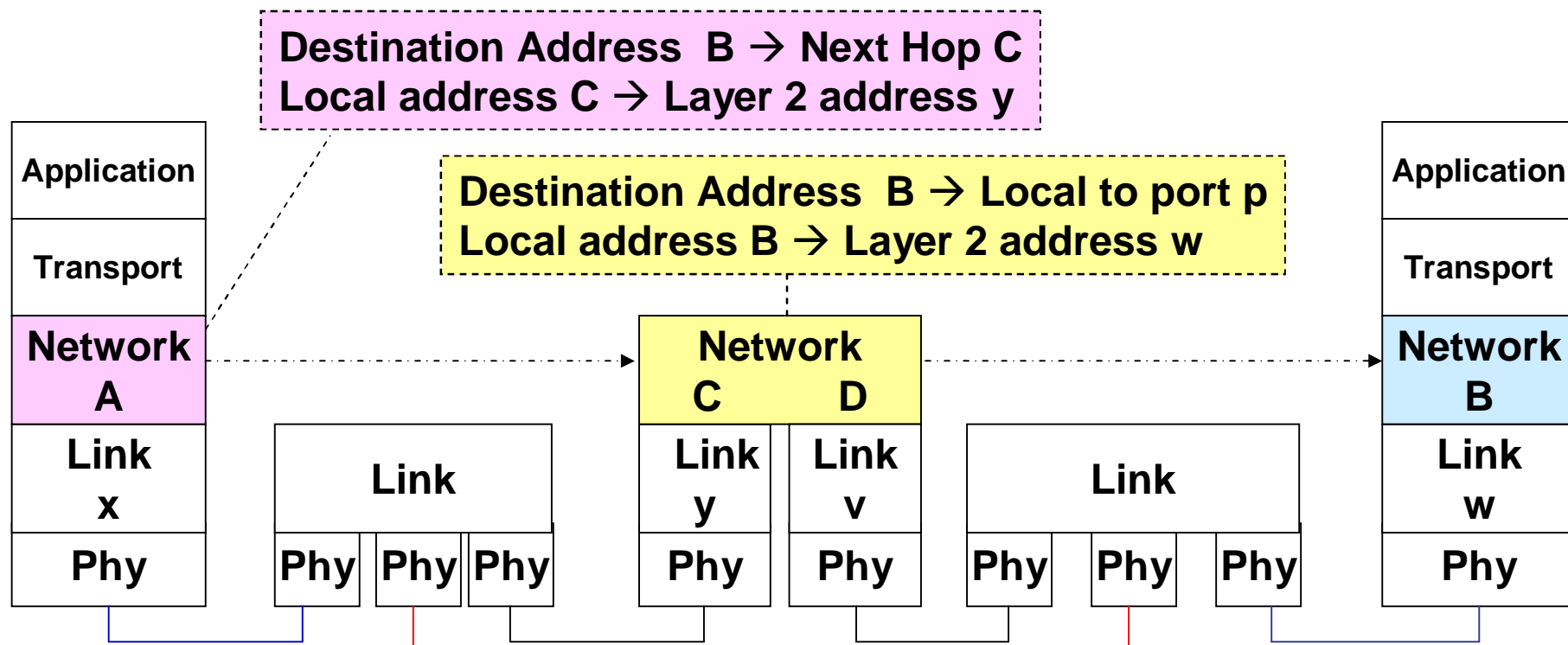
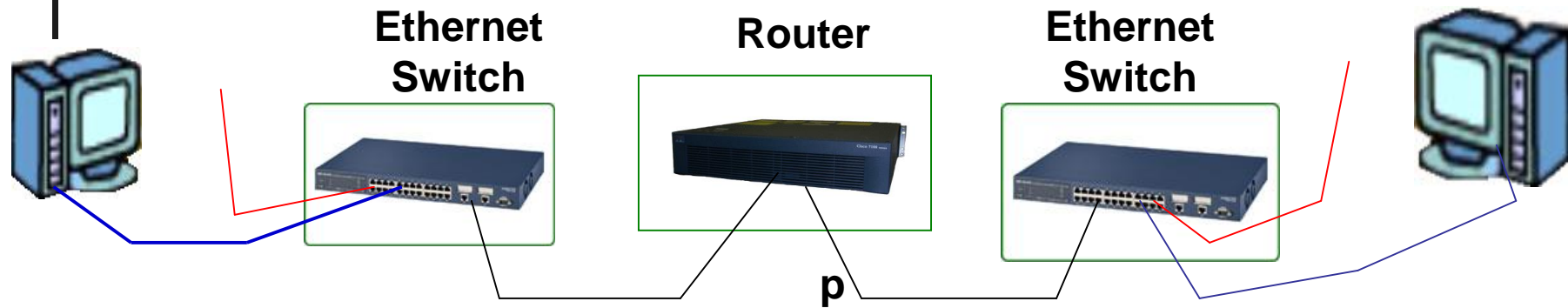
Type (= 17 or 18)	Code (=0)	Checksum
identifier		sequence number
32-bit sender timestamp		
32-bit receive timestamp		
32-bit transmit timestamp		



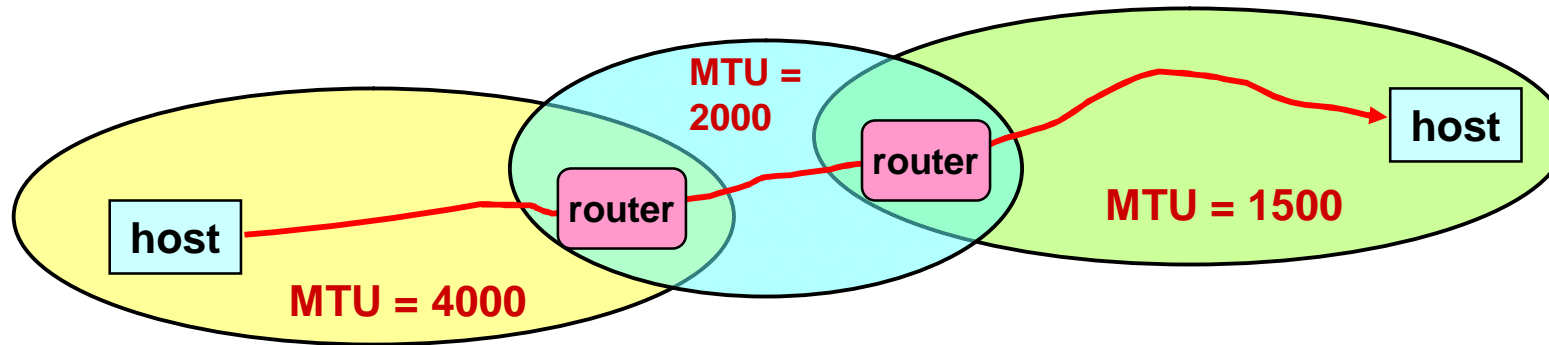
# FRAGMENTATION

Dealing with the different packet lengths  
on different networks

# Packet size: limitations of Layer 2 hold...



# IP Fragmentation





# Maximum Transmission Unit

- Maximum size of IP datagram is 65535, but the data link layer protocol generally imposes a limit that is much smaller
- Example:
  - Ethernet frames have a maximum payload of 1500 bytes  
→ IP datagrams encapsulated in Ethernet frame cannot be longer than 1500 bytes

“ Limits for various data link protocols:

Ethernet: 1500

FDDI: 4352

802.3: 1492

ATM AAL5: 9180

802.5: 4464

PPP: negotiated

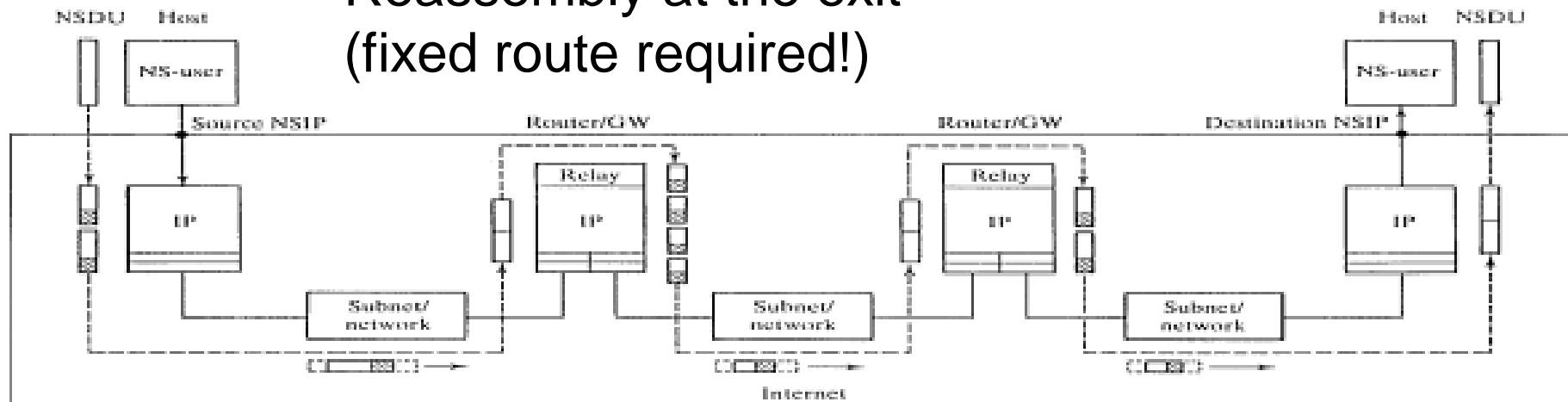
## Transferring an IP Packet on a network..

- IP packets are transported as a PAYLOAD on intermediate networks - e.g. Ethernets.
- The underlying network has limits on packet payload length – this is called MTU- maximum transfer unit ....
  - Every internet module **must** be able to forward a datagram of 68 octets;
  - Every internet destination must be able to receive a datagram of 576 octets;
- But: IP sender does not, in general, have to know which networks will transmit the packet... and use longer ones..
- Fragmentation – division of a long packet in „Pieces“..
- *Alternatively:*
  - *Use packet lengths known as „transportable“ (short enough)*
  - *Use path features discovery*

# Fragmentation Approaches

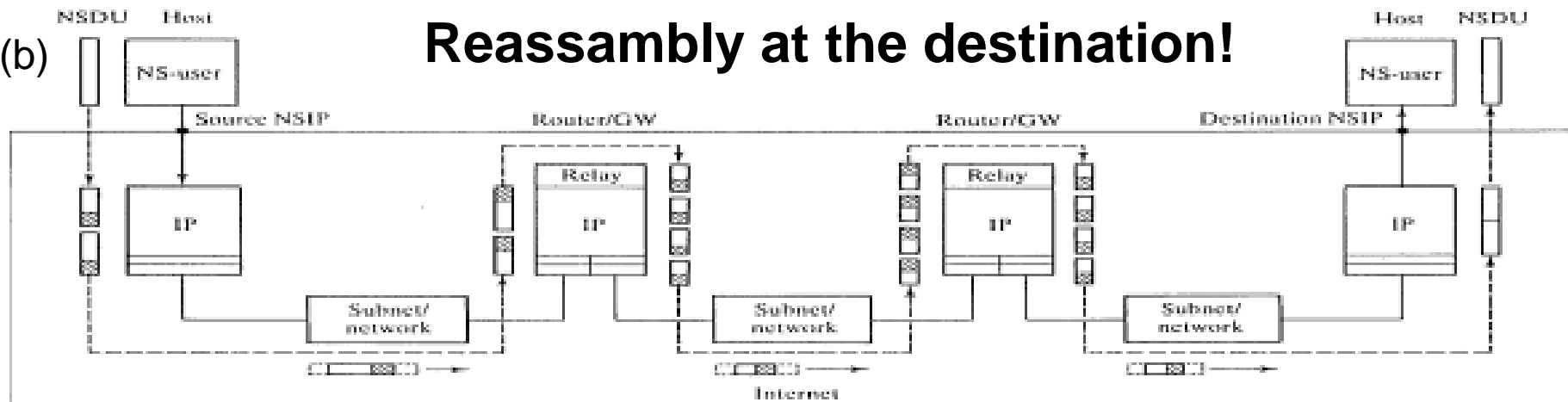
(a)

Reassembly at the exit  
(fixed route required!)



(b)

Reassembly at the destination!



GW = Gateway  
MTU = Message transfer unit

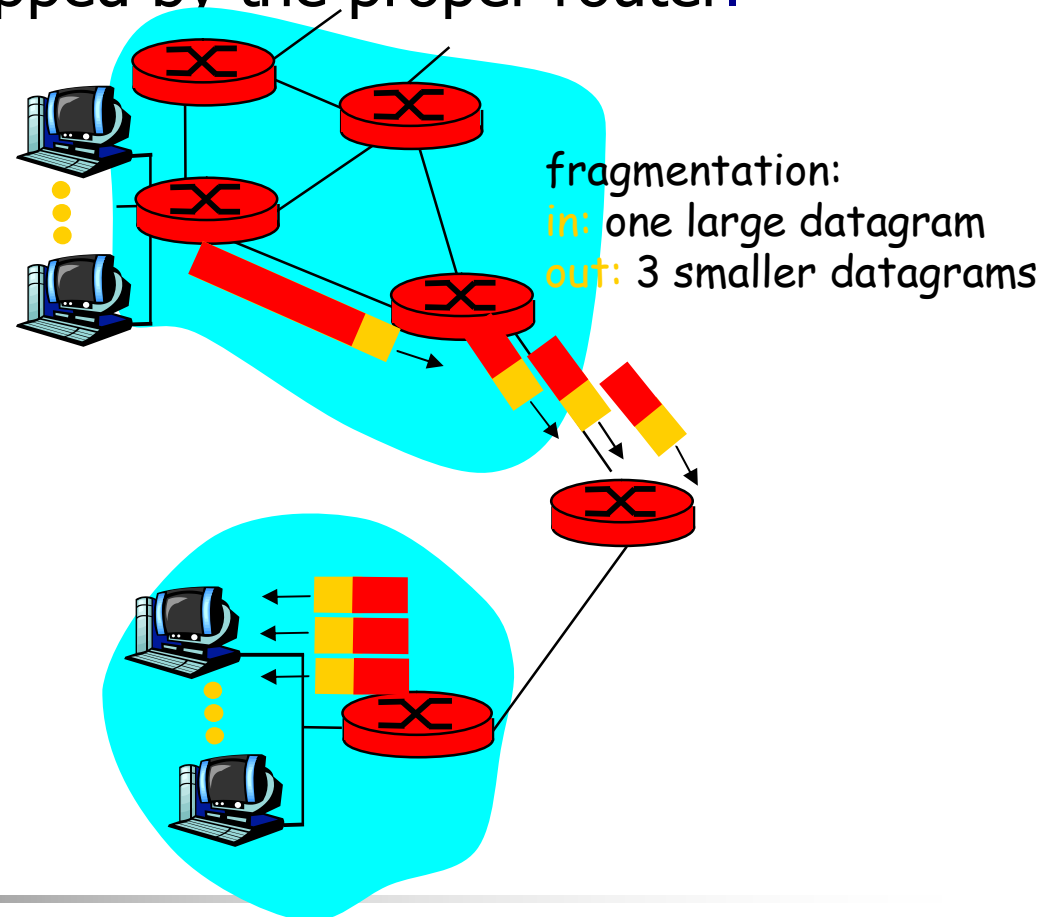
Header →  
Data →  
Subnet/network header  
Subnet/network trailer } MTU

# Fragmentation and reassembly - options

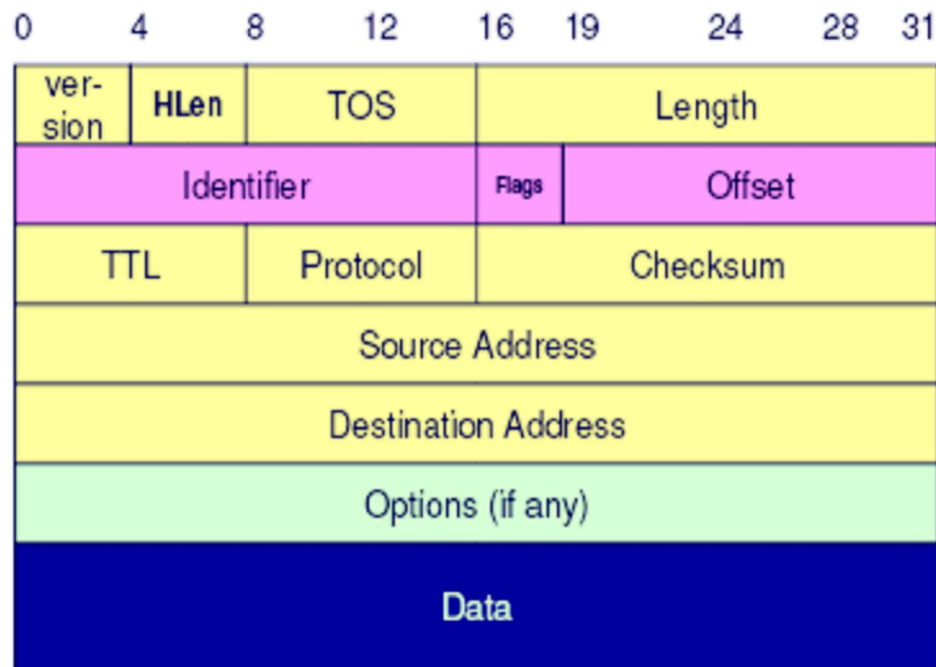
- In the Internet, re-assembling is done by the destination host, not by a router
- In case of a **non-fragment option**, packets with excessive length are dropped by the proper router.

**Alternative:** Variant (a)  
(in the previous slide)

- assures always maximum possible length of the pieces, no small packets on following networks (if they can support big ones)
- requires however that fragments leave a network using a **SINGLE router!!! (fixed route of all packets within the network!)**



# Fragmentation in IP – how to ?



## Identifier

- Unique identifier for original datagram
  - Historically, source increments counter every time sends packet

## Flags (3 bits)

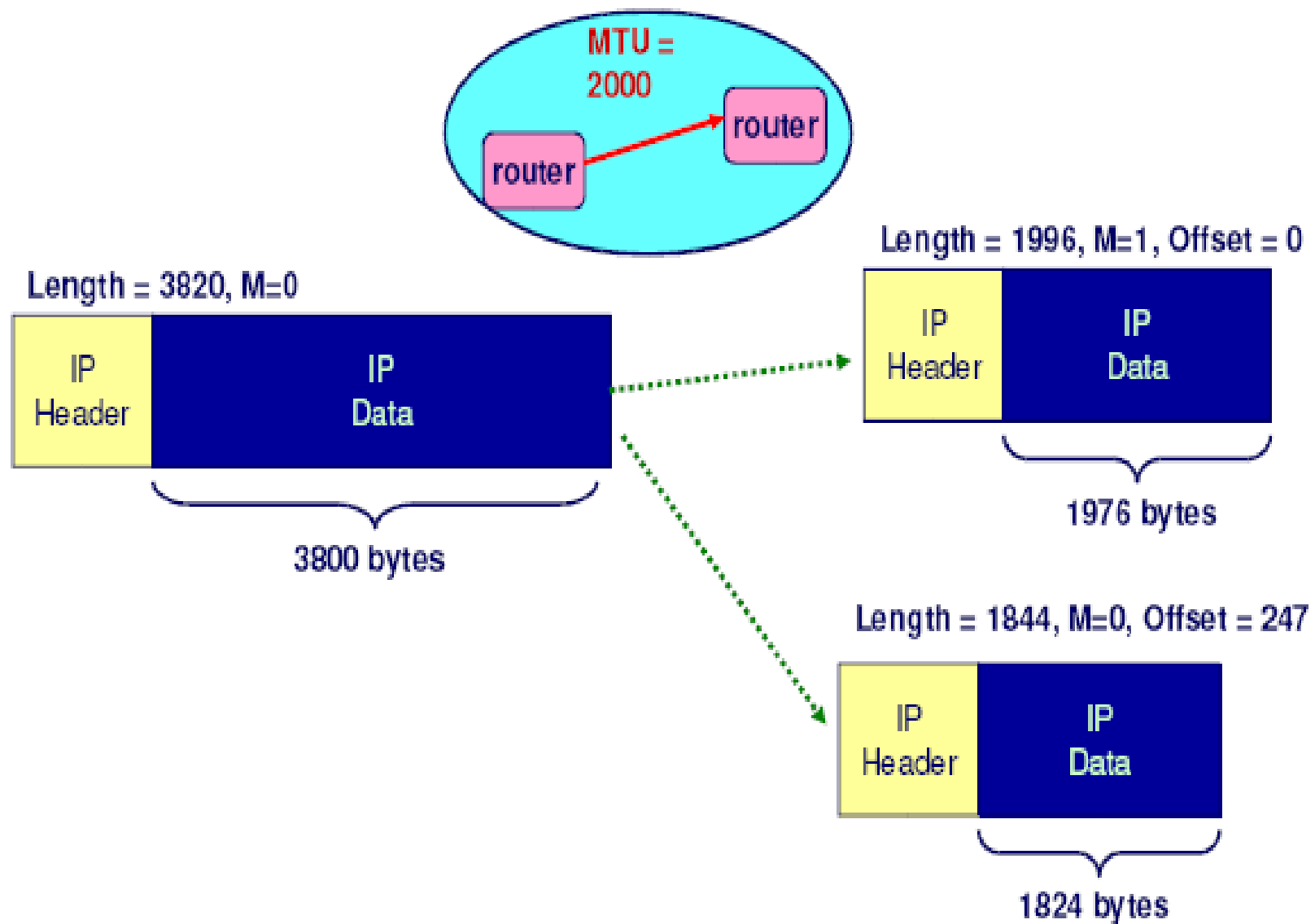
- “More fragments” flag: This is not the last fragment

## Offset

- Byte position of first byte in fragment  $\div 8$
- Byte position must be multiple of 8

- Each fragment carries copy of IP header
  - All information required for delivery to destination
- All fragments comprising original datagram have same identifier
- Offsets indicate positions within datagram

# Fragmentation - an example



# Reassembly

Length = 1500, M=1, Offset = 0



Length = 516, M=1, Offset = 185



Length = 1500, M=1, Offset = 247



Length = 364, M=0, Offset = 432



- Performed at final destination
- Fragment with M=0 determines overall length
  - $(432 * 8) + (364 - 20) = 3800$

## Challenges

- 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
- Significant memory management issues
  - See code in book

# The Internet style of fragmentation

## **Demonstrates Many Internet Concepts**

### **Decentralized**

- Every network can choose MTU

### **Connectionless Datagram Protocol**

- Each (fragment of) packet contains full routing information
- Fragments can proceed independently and along different routes

### **Fail by Dropping Packet**

- Destination can give up on reassembly
- No need to signal sender that failure occurred

### **Keep Most Work at Endpoints**

- Reassembly



## IP fragmentation and reassembly – processing

- Some options are copied, but others remain with the first fragment only;
- Fields which may be affected by fragmentation include:
  - options field
  - more fragments flag
  - fragment offset
  - internet header length field
  - total length field
  - header checksum
- Not so much used....

# 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
  - Loss of a fragment
- Path MTU discovery protocol → determines minimum MTU along route
  - Uses ICMP error messages
- Common theme in system design
  - Assure correctness by implementing complete protocol
  - Optimize common cases to avoid full complexity



# IPv6

# IPv6

---

- Motivated (prematurely) by address exhaustion
  - Addresses *four* times as big
- Steve Deering focused on simplifying IP
  - Got rid of all fields that were not absolutely necessary
  - “Spring Cleaning” for IP
- Result is an elegant, if unambitious, protocol

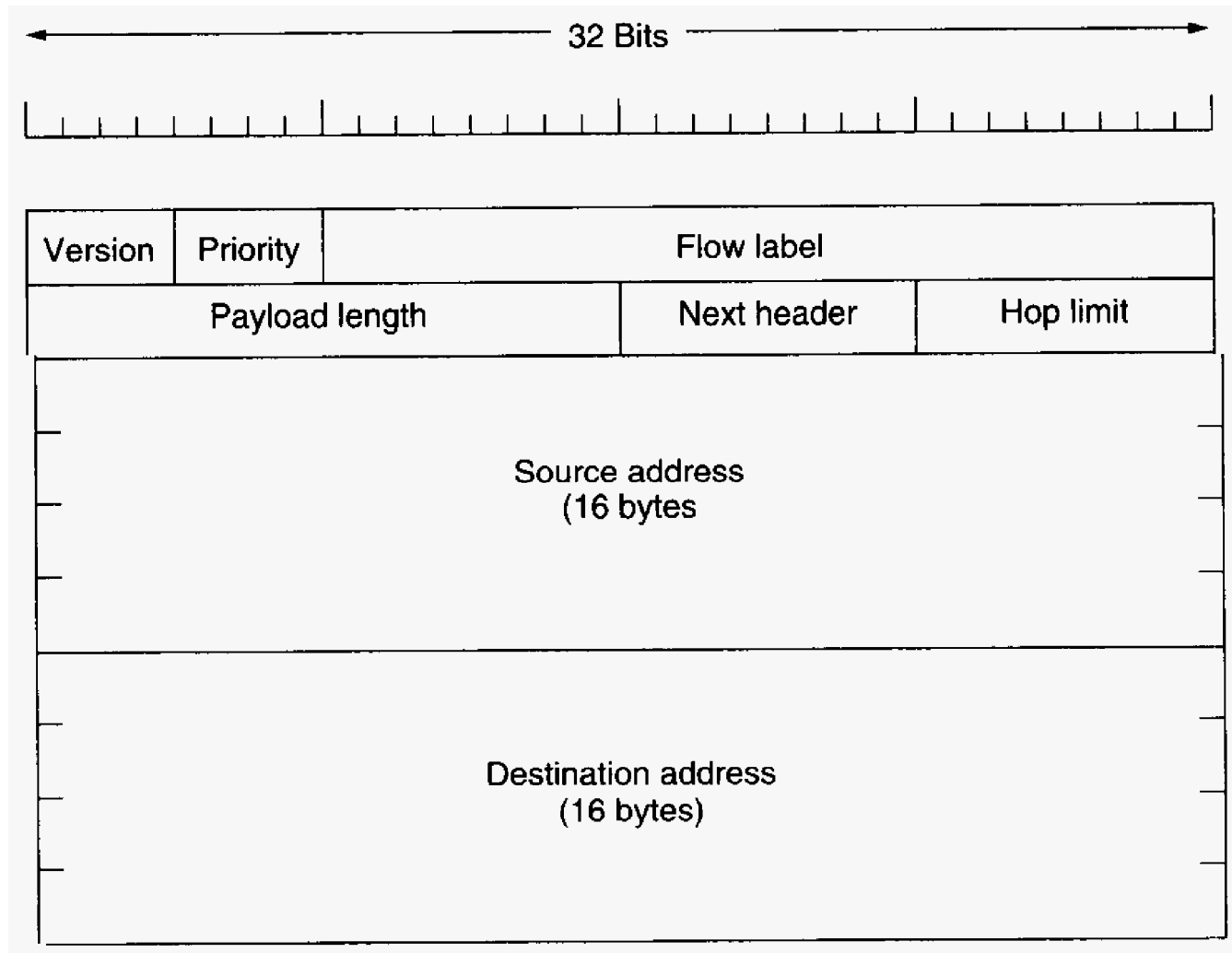
## Major features

---

- 128-bit addresses
- Auto-configuration
- Multicast
- Better QoS Support
- Authentication and security
- End-to-end fragmentation
- Enhanced routing functionality, including support for mobile hosts

## IP Version 6 – Protocol Header

“ The IPv6 header (fixed part):



## IP Version 6 – Header Fields

“ Values are:

- . Version: 6 (vs. 4 in IPv4)
- . Priority: 0-7 for best effort, 8-15 for constant rate (even with loss). Example: 1 for news, 4 for FTP, 6 for telnet
- . Flow Label: Still experimental. Idea: Setting up pseudo connections (virtual circuit) with certain properties.
- . Payload Length: Length of data after header (without header! Different to IPv4).
- . Next Header: Defines if and which of the 6 until now defined extension headers follow after the fixed header; alternatively UDP or TCP.
- . Hop Limit: Same as time-to-live in IPv4

# Structuring IP Addresses...

- 128-bit

- Address notation

- String of eight 16-bit hex values separated by colons

- 5CFA:0002:0000:0000:CF07:1234:5678:FFCD

- Set of contiguous 0's can be elided

- 5CFA:0002::0000:CF07:1234:5678:FFCD

- Address assignment

- Provider-based
- geographic



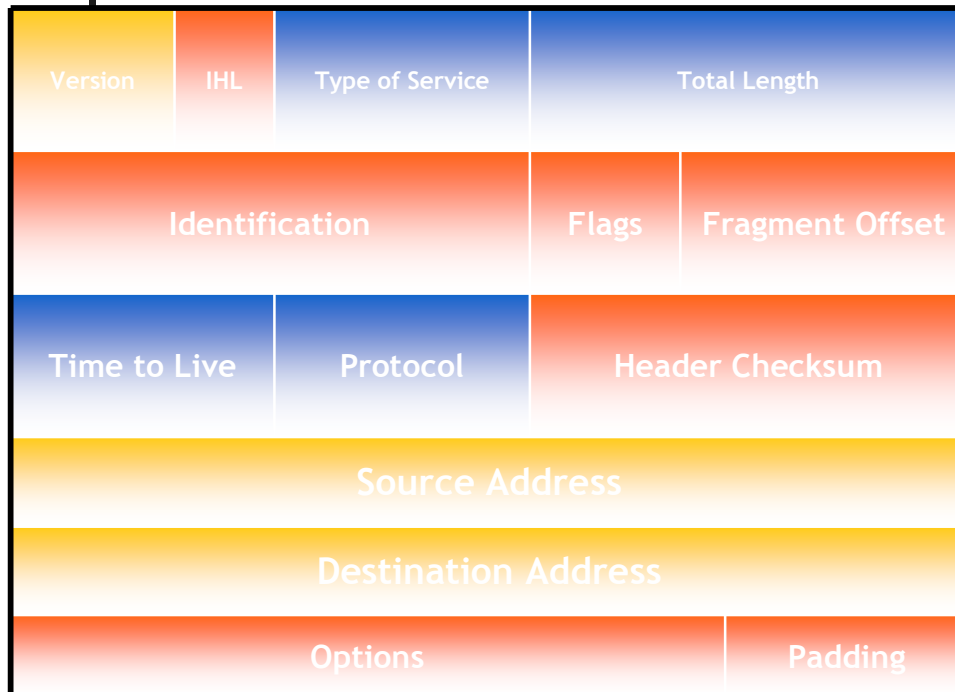






## Auto-configuration

- An IP v6 Address can be derived from the MAC address!
  - A mac address is 48 bits, an IPv6 address is 128 bits. Here's the conversion process step by step:
  - take the mac address: for example 52:74:f2:b1:a8:7f
  - throw ff:fe in the middle: 52:74:f2:**ff:fe**:b1:a8:7f
  - reformat to IPv6 notation 5274:f2ff:feb1:a87f
  - convert the first octet from hexadecimal to binary: **52** -> **01010010**
  - invert the bit at position 6 (counting from 0): **01010010** -> **01010000**
  - convert octet back to hexadecimal: **01010000** -> **50**
  - replace first octet with newly calculated one: **50**74:f2ff:feb1:a87f
  - prepend the link-local prefix: **fe80::**5074:f2ff:feb1:a87f
  - See a calculator under <http://ben.akrin.com/?p=1347>
- more info:  
[http://www.cisco.com/web/about/ac123/ac147/archived\\_issues/ipj\\_7-2/ipv6\\_autoconfig.html](http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_7-2/ipv6_autoconfig.html)
- Useful for massively deployed embedded systems...

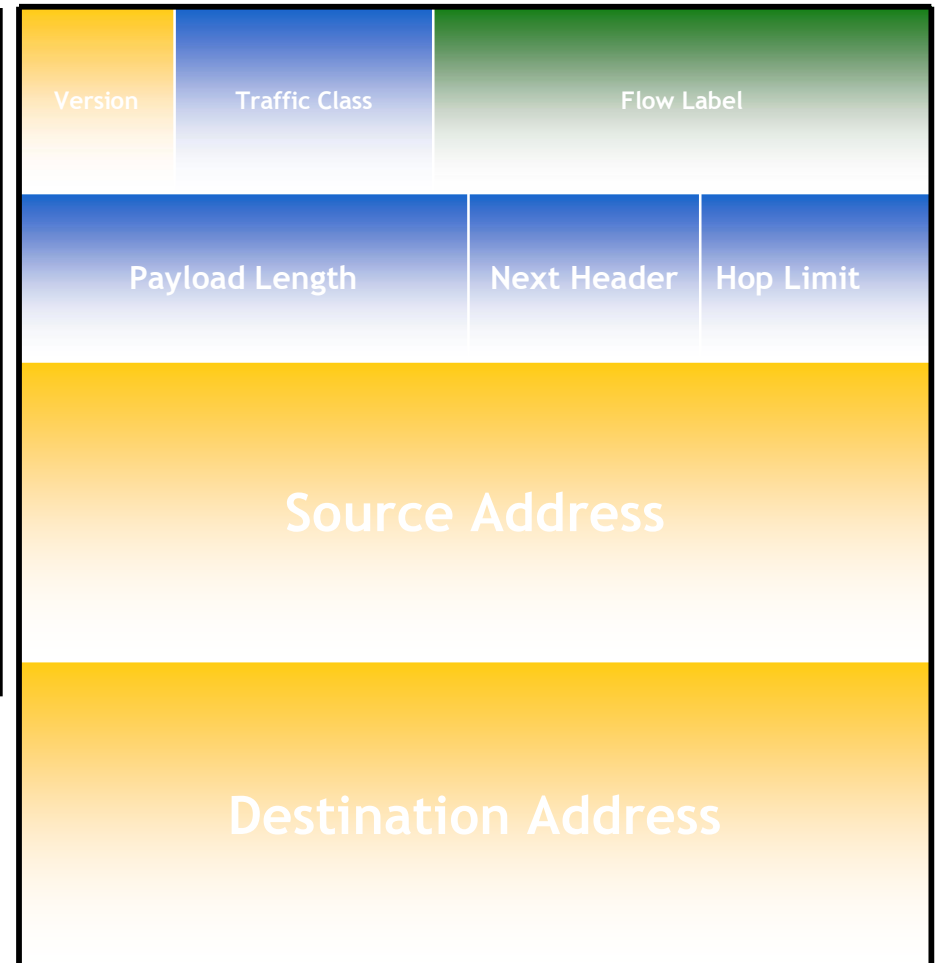
# IPv4 and IPv6 Header Comparison

## IPv4



-  Field name kept from IPv4 to IPv6
-  Fields not kept in IPv6
-  Name & position changed in IPv6
-  New field in IPv6

## IPv6



# Philosophy of Changes

- Don't deal with problems: leave to ends
  - Eliminated fragmentation in routers
  - Eliminated checksum
  - *Why retain TTL?*
- Simplify handling:
  - New options mechanism (uses next header approach)
  - Eliminated header length
    - *Why couldn't IPv4 do this?*
  - Eliminated checksum (*why?*)
- Provide general flow label for packet
  - Not tied to semantics
  - Provides great flexibility

## IP Version 6 – Extension Header Examples

---

- “ Hop-by-hop options: Miscellaneous information for routers
- “ Routing: Full or partial route to follow
- “ Fragmentation: Management of datagram fragments
- “ Authentication: Verification of the senders identity
- “ Encrypted security payload: Information about the encrypted contents
- “ Destination options: Additional information for the destination

# Migration: IPv6 Islands in IPv4 World

