TCP/IP Protocol Stack

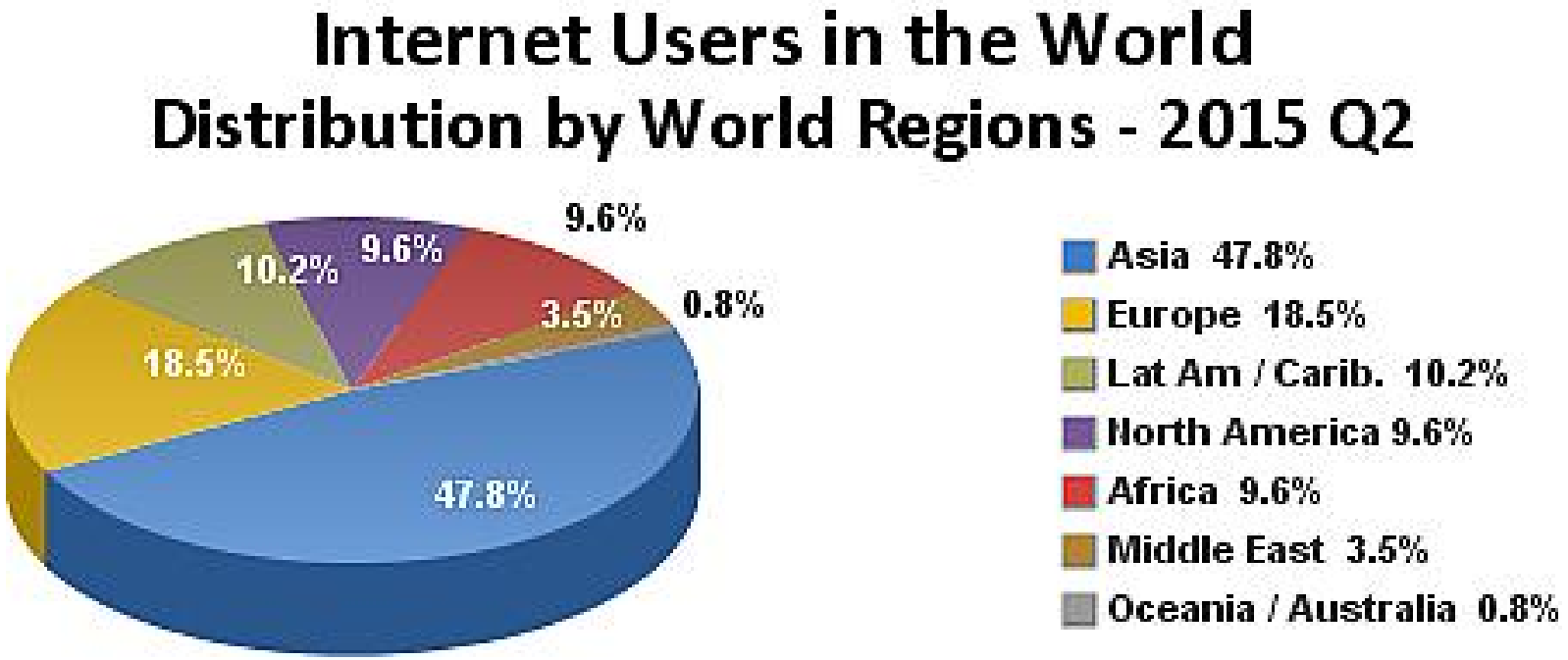
Dr.-Ing. Ali Diab

Outline

- Introduction
- TCP/IP Protocol Suite
- Addressing in TCP/IP World
- TCP/IP Layers
  - Network Interface Hardware
    - ARP
  - Internet Layer
    - IPv4 & v6
  - Transport Layer
    - TCP, UDP
  - Application Layer
  - HTTP, RTP, etc.
- Conclusions
- References

Introduction

Internet Population Growth



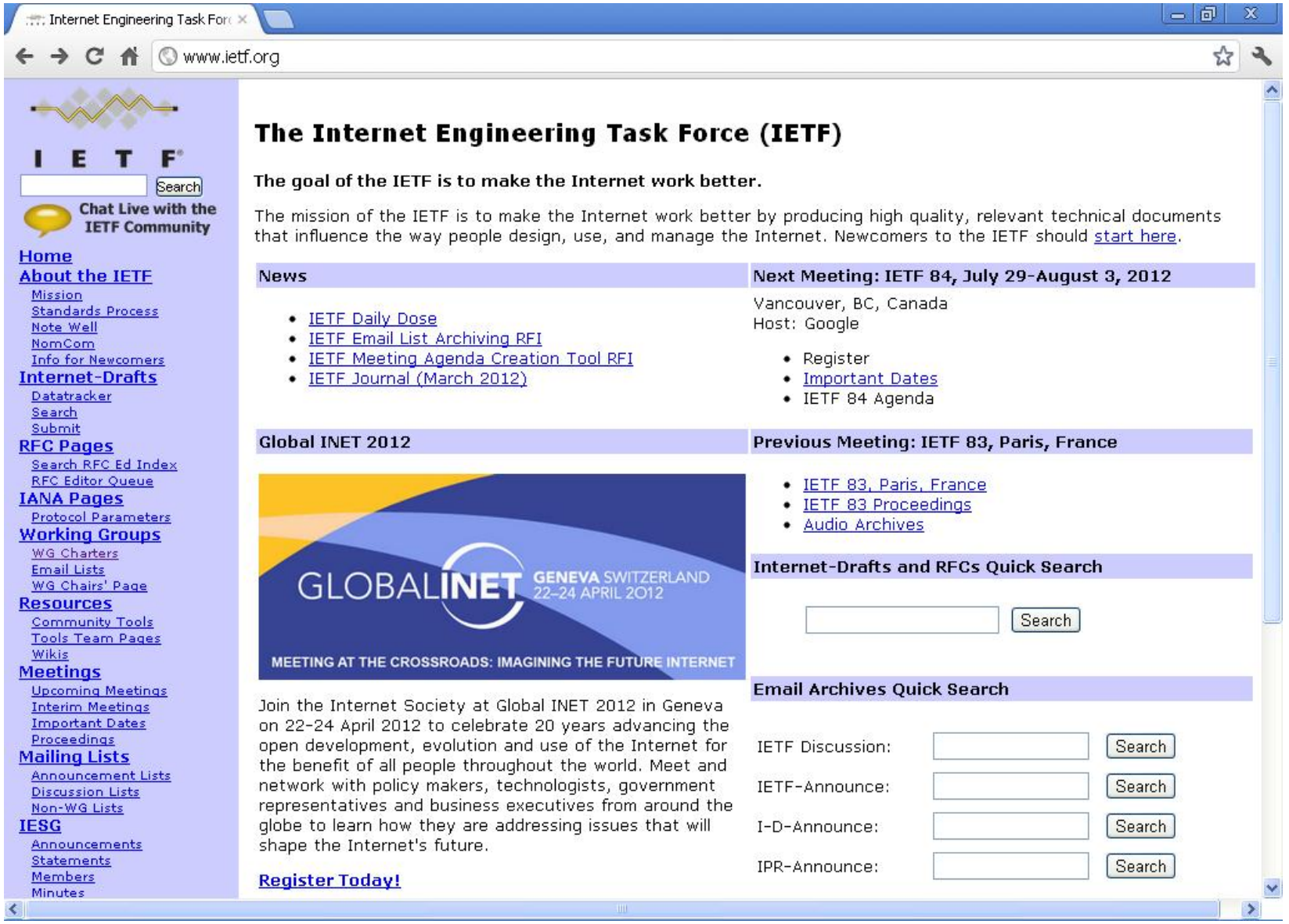
Source: Internet World Stats - www.internetworldstats.com/stats.htm
Basis: 3,270,490,584 Internet users on June 30, 2015

TCP/IP - The Internet Protocol Suite

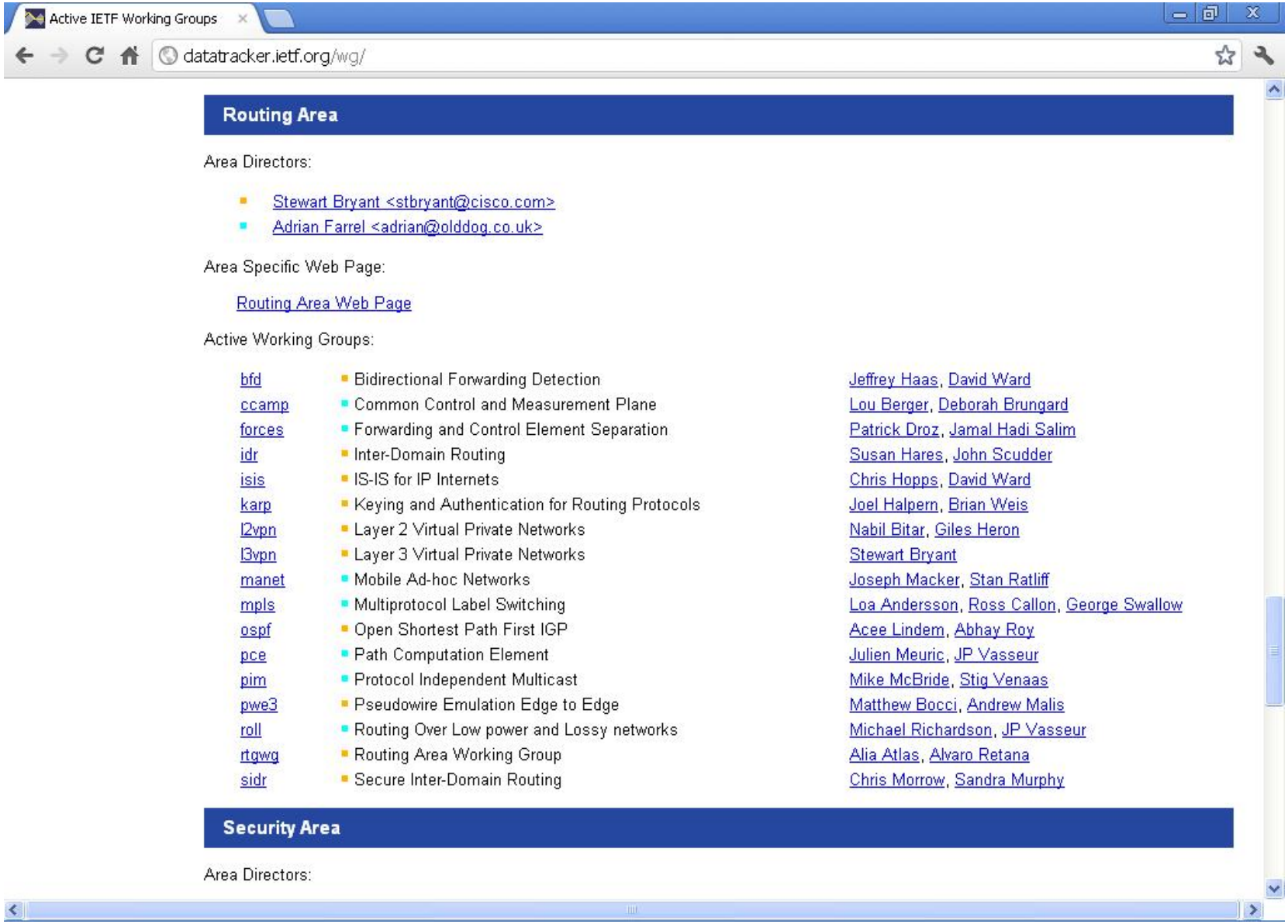
- The **Internet protocol suite**
  - The a set of protocols designed for the **Internet**
  - Allows communication across diverse networks
  - Out of ARPANET
  - Emphasize on robustness in terms of failure handling
  - Emphasize on Flexibility in operating on diverse networks

- Standardisation (ISOC: Internet Society)
  - IAB: Internet Architecture Board
  - IETF: Internet Engineering Task Force: http://www.ietf.org
    - Standards & other information are published as drafts and RFCs (Requests for Comments)
  - IRTF: Internet Research Task Force

IETF


Internet Engineering Task For
www.ietf.org
I E T F
Search
Chat Live with the IETF Community
Home
About the IETF
Mission
Standards Process
Note Well
NomCom
Info for Newcomers
Internet-Drafts
Datatracker
Search
Submit
RFC Pages
Search RFC Ed Index
RFC Editor Queue
IANA Pages
Protocol Parameters
Working Groups
WG Charters
Email Lists
WG Chairs' Page
Resources
Community Tools
Tools Team Pages
Wikis
Meetings
Upcoming Meetings
Interim Meetings
Important Dates
Proceedings
Mailing Lists
Announcement Lists
Discussion Lists
Non-WG Lists
IESG
Announcements
Statements
Members
Minutes
The Internet Engineering Task Force (IETF)
The goal of the IETF is to make the Internet work better.
The mission of the IETF is to make the Internet work better by producing high quality, relevant technical documents that influence the way people design, use, and manage the Internet. Newcomers to the IETF should start here.
News
IETF Daily Dose
IETF Email List Archiving RFI
IETF Meeting Agenda Creation Tool RFI
IETF Journal (March 2012)
Next Meeting: IETF 84, July 29-August 3, 2012
Vancouver, BC, Canada
Host: Google
Register
Important Dates
IETF 84 Agenda
Global INET 2012
GLOBALINET GENEVA SWITZERLAND 22-24 APRIL 2012
MEETING AT THE CROSSROADS: IMAGINING THE FUTURE INTERNET
Join the Internet Society at Global INET 2012 in Geneva on 22-24 April 2012 to celebrate 20 years advancing the open development, evolution and use of the Internet for the benefit of all people throughout the world. Meet and network with policy makers, technologists, government representatives and business executives from around the globe to learn how they are addressing issues that will shape the Internet's future.
Register Today!
Previous Meeting: IETF 83, Paris, France
IETF 83, Paris, France
IETF 83 Proceedings
Audio Archives
Internet-Drafts and RFCs Quick Search
Search
Email Archives Quick Search
IETF Discussion:
IETF-Announce:
I-D-Announce:
IPR-Announce:
Search

IETF



Routing Area

Area Directors:

- Stewart Bryant <stbryant@cisco.com>
- Adrian Farrel <adrian@olddog.co.uk>

Area Specific Web Page:

Routing Area Web Page

Active Working Groups:

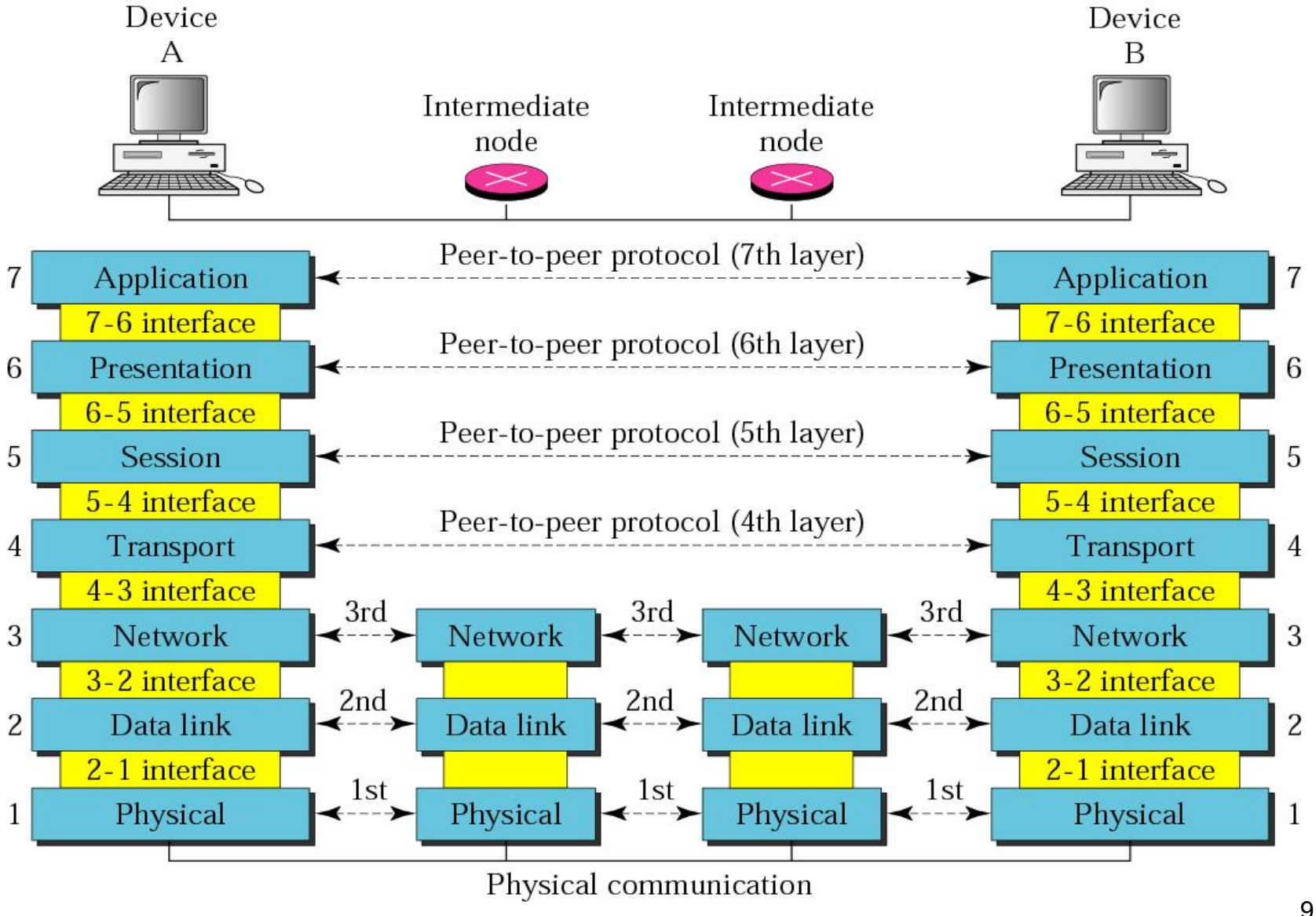
bfd	Bidirectional Forwarding Detection	Jeffrey Haas, David Ward
ccamp	Common Control and Measurement Plane	Lou Berger, Deborah Brungard
forces	Forwarding and Control Element Separation	Patrick Droz, Jamal Hadi Salim
idr	Inter-Domain Routing	Susan Hares, John Scudder
isis	IS-IS for IP Internets	Chris Hopps, David Ward
karp	Keying and Authentication for Routing Protocols	Joel Halpern, Brian Weis
l2vpn	Layer 2 Virtual Private Networks	Nabil Bitar, Giles Heron
l3vpn	Layer 3 Virtual Private Networks	Stewart Bryant
manet	Mobile Ad-hoc Networks	Joseph Macker, Stan Ratliff
mpls	Multiprotocol Label Switching	Loa Andersson, Ross Callon, George Swallow
ospf	Open Shortest Path First IGP	Acee Lindem, Abhay Roy
pce	Path Computation Element	Julien Meuric, JP Vasseur
pim	Protocol Independent Multicast	Mike McBride, Stig Venaas
pwe3	Pseudowire Emulation Edge to Edge	Matthew Bocci, Andrew Malis
roll	Routing Over Low power and Lossy networks	Michael Richardson, JP Vasseur
rtgwg	Routing Area Working Group	Alia Atlas, Alvaro Retana
sidr	Secure Inter-Domain Routing	Chris Morrow, Sandra Murphy

Security Area

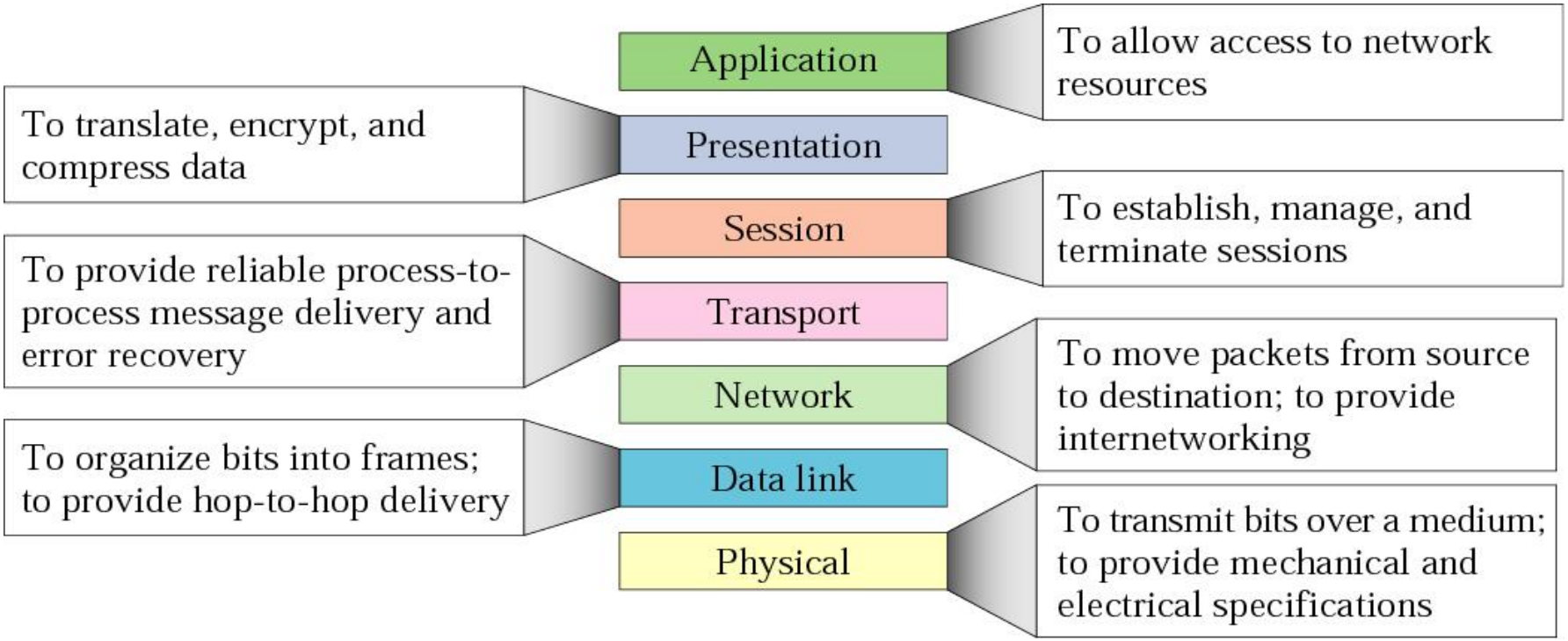
Area Directors:

ISO/OSI &TCP/IP Protocol Suite

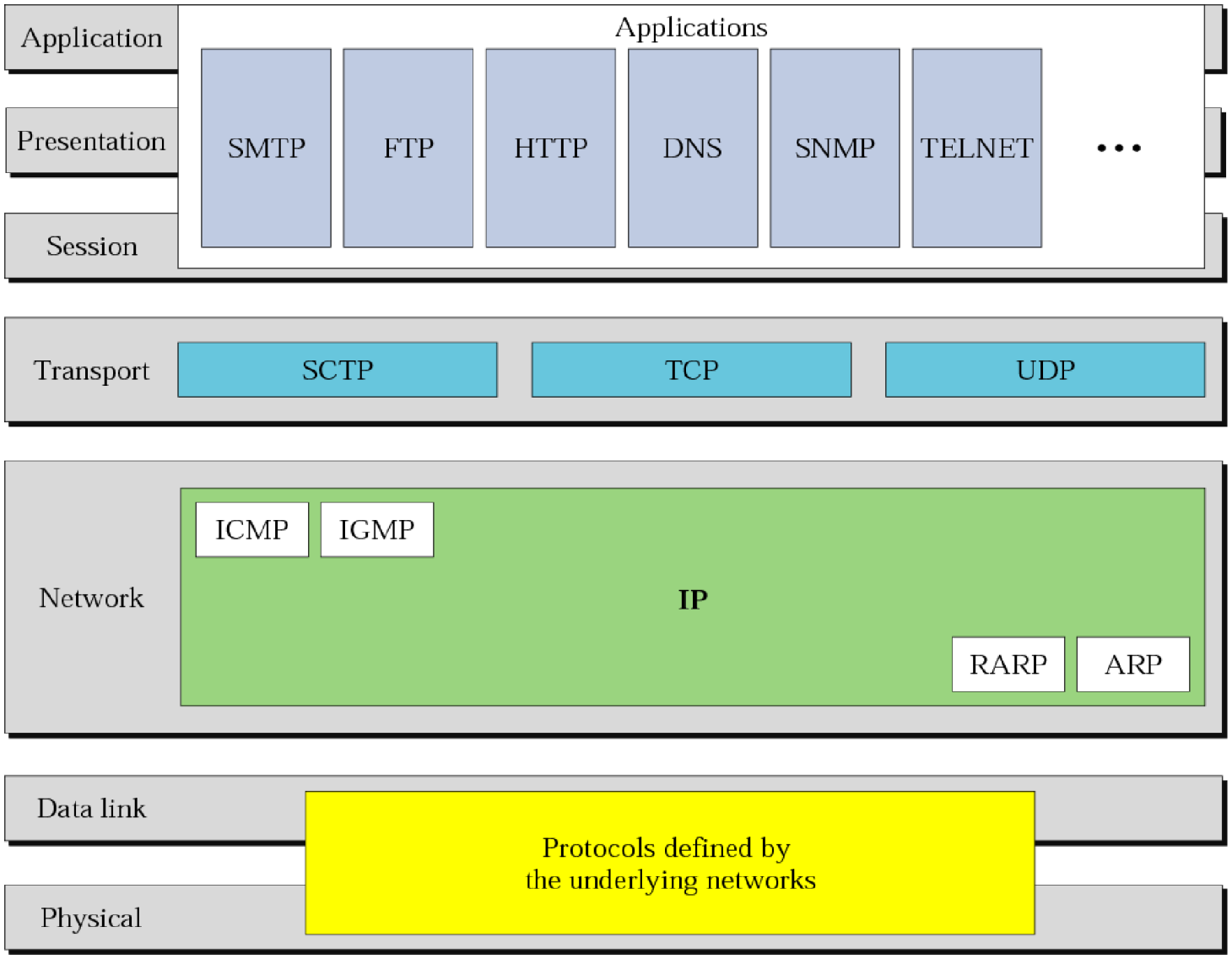
OSI Layers


Device A
Device B
Intermediate node
Intermediate node
Peer-to-peer protocol (7th layer)
Peer-to-peer protocol (6th layer)
Peer-to-peer protocol (5th layer)
Peer-to-peer protocol (4th layer)
7 Application
7-6 interface
6 Presentation
6-5 interface
5 Session
5-4 interface
4 Transport
4-3 interface
3 Network
3-2 interface
2 Data link
2-1 interface
1 Physical
3rd
2nd
1st
Physical communication

OSI Layers


Application
To allow access to network resources
To translate, encrypt, and compress data
Presentation
Session
To establish, manage, and terminate sessions
To provide reliable process-to-process message delivery and error recovery
Transport
Network
To move packets from source to destination; to provide internetworking
To organize bits into frames; to provide hop-to-hop delivery
Data link
Physical
To transmit bits over a medium; to provide mechanical and electrical specifications

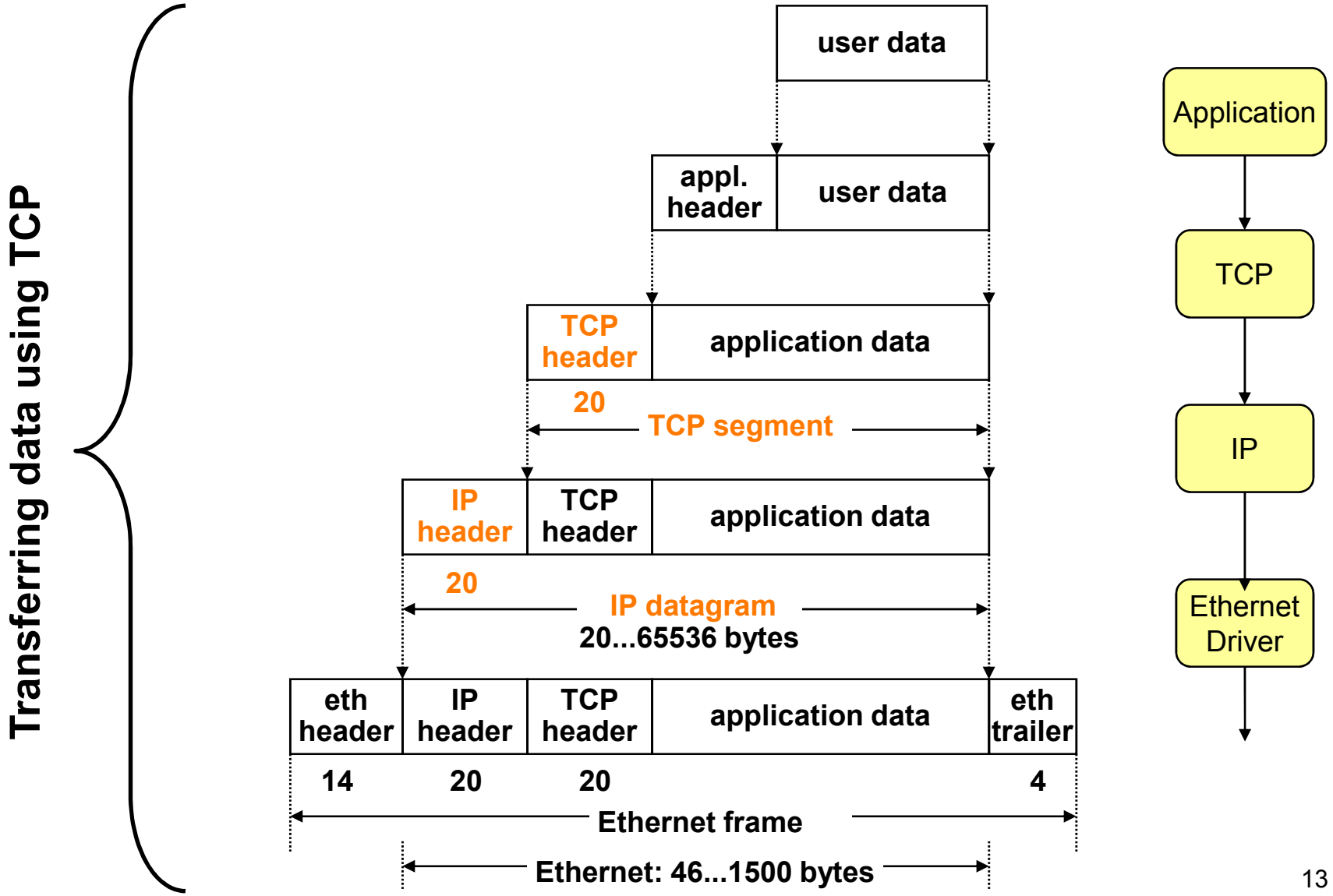
TCP/IP Layers



TCP/IP Layers

TCP/IP Layer	Tasks	Protocol Examples
Application	Application specific	Telnet, rlogin, FTP, SMTP, SNMP, HTTP, etc.
Transport	End-to-end flow of data between application processes	TCP, UDP
Internet	Routing of packets between hosts	IP, ICMP
Network Interface Hardware	Hardware interface (packet transfer be-tween network nodes)	PPP, Ethernet, IEEE 802.x, ARP

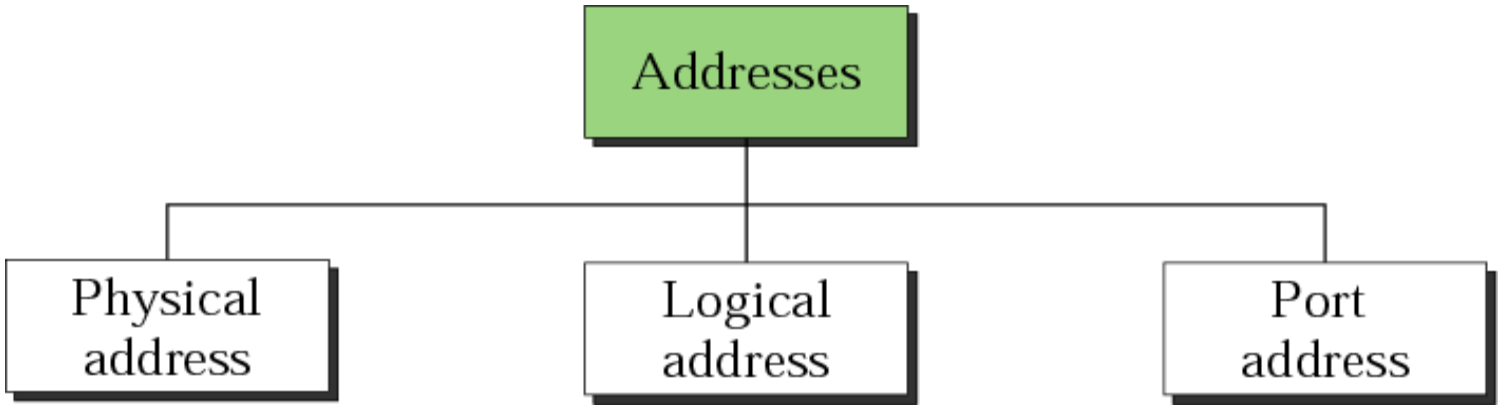
TCP/IP Encapsulation



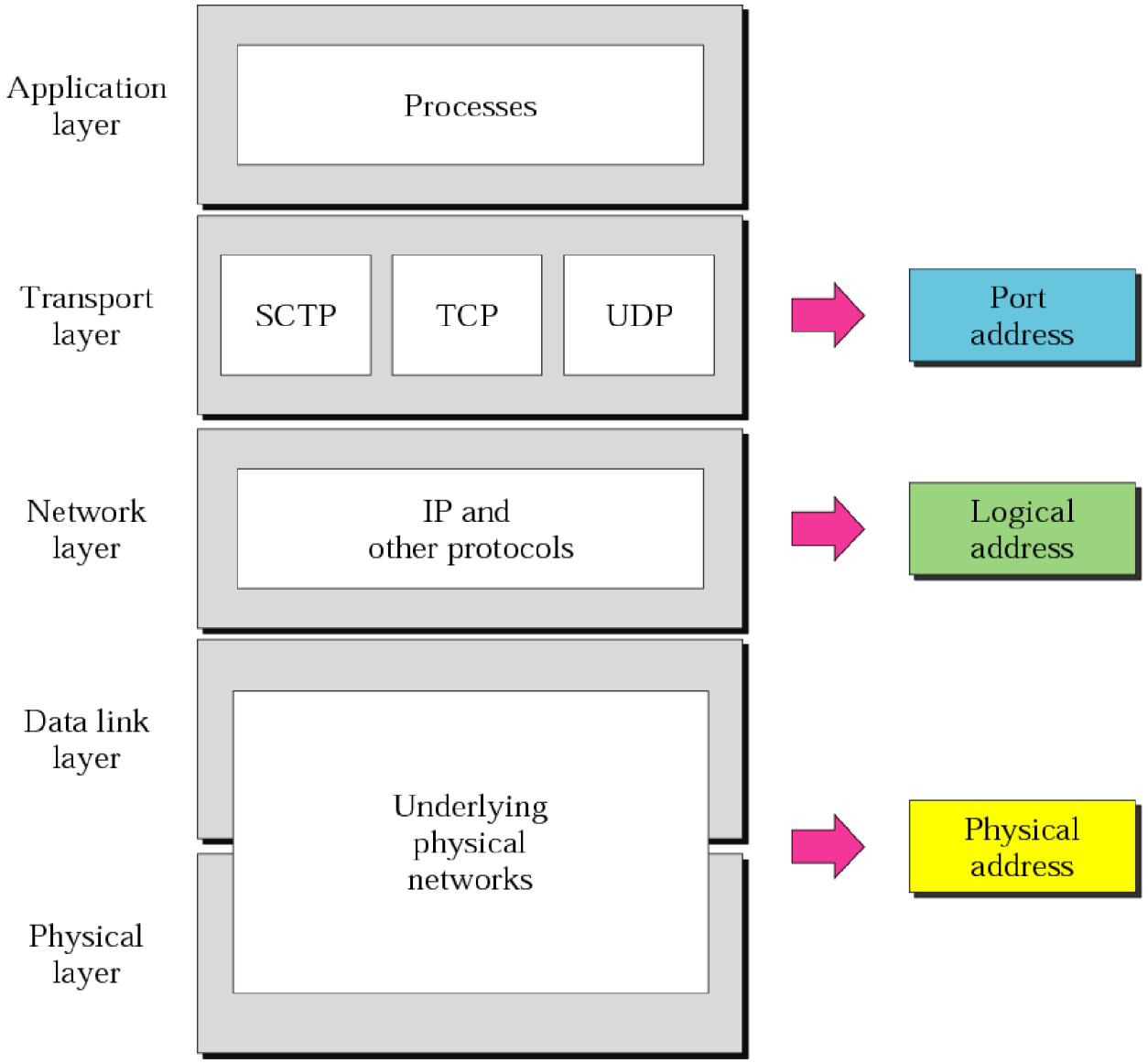
Addressing in TCP/IP World

Types of Addresses

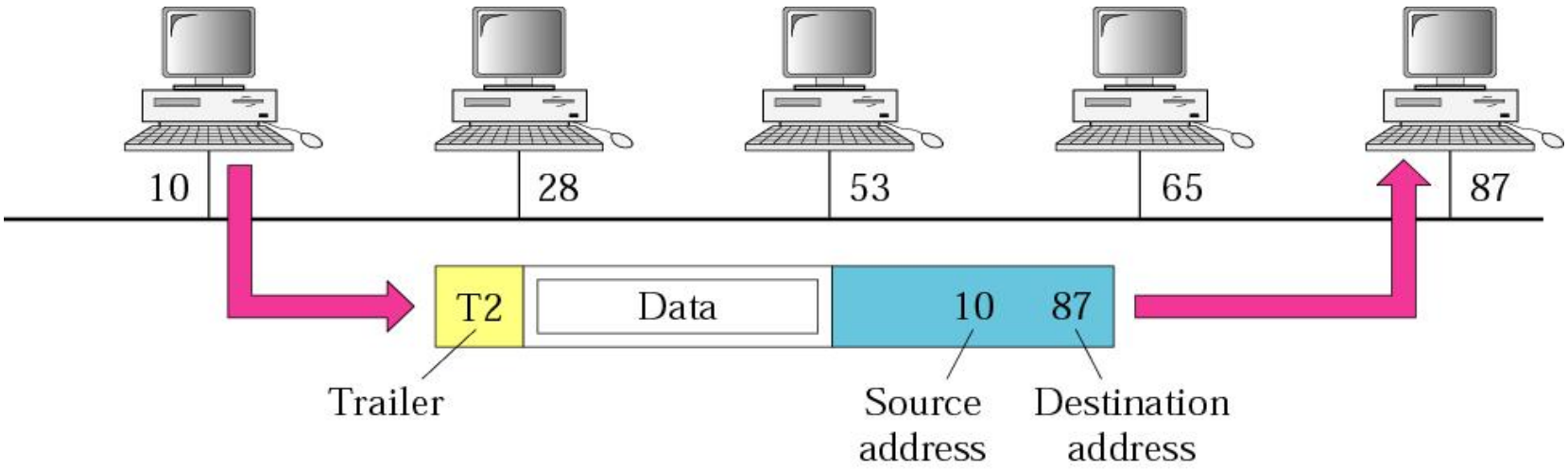
- Three different levels of addresses are used in the internet
  - Physical (link) address
  - Logical (IP) address
  - Port address



Relationship between Layers & Addresses in TCP/IP



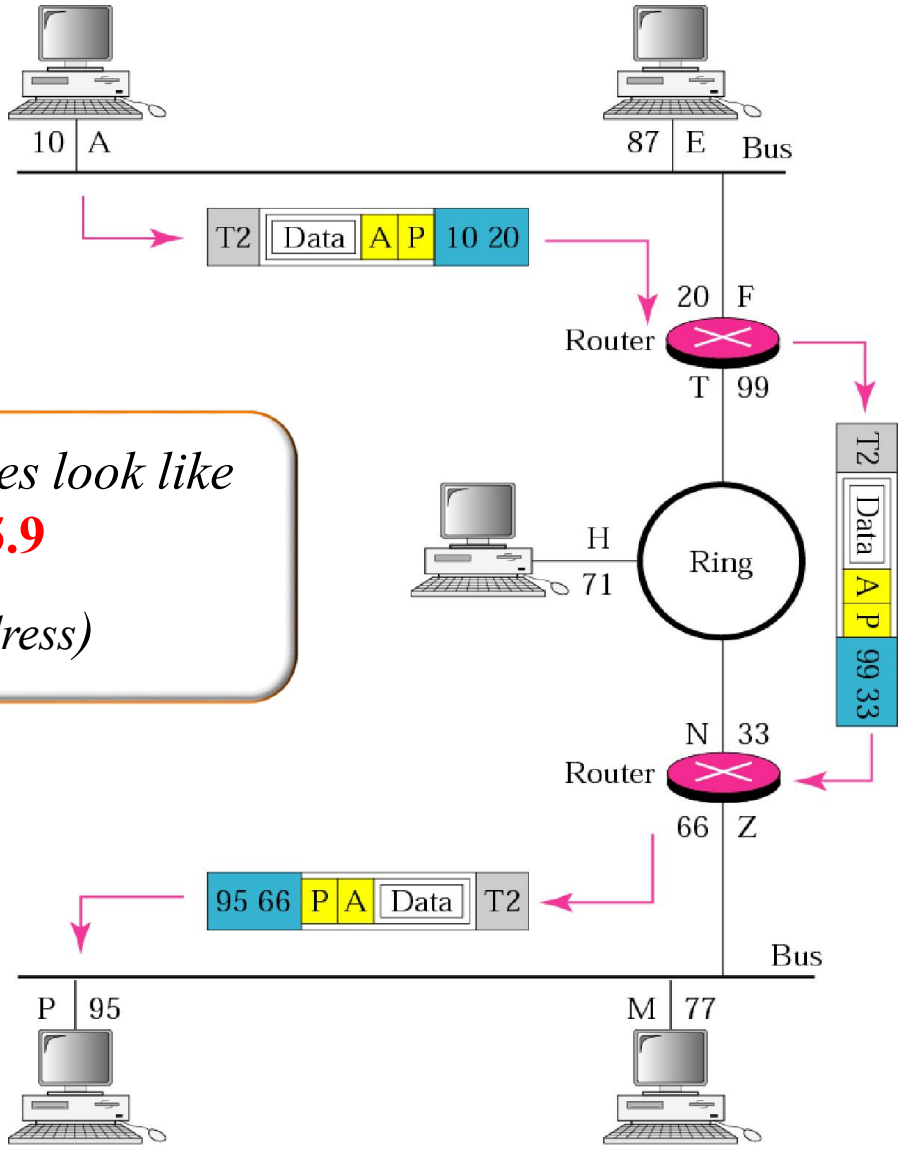
Physical Addresses



In fact, physical addresses look like ***07:01:02:01:2C:4B***

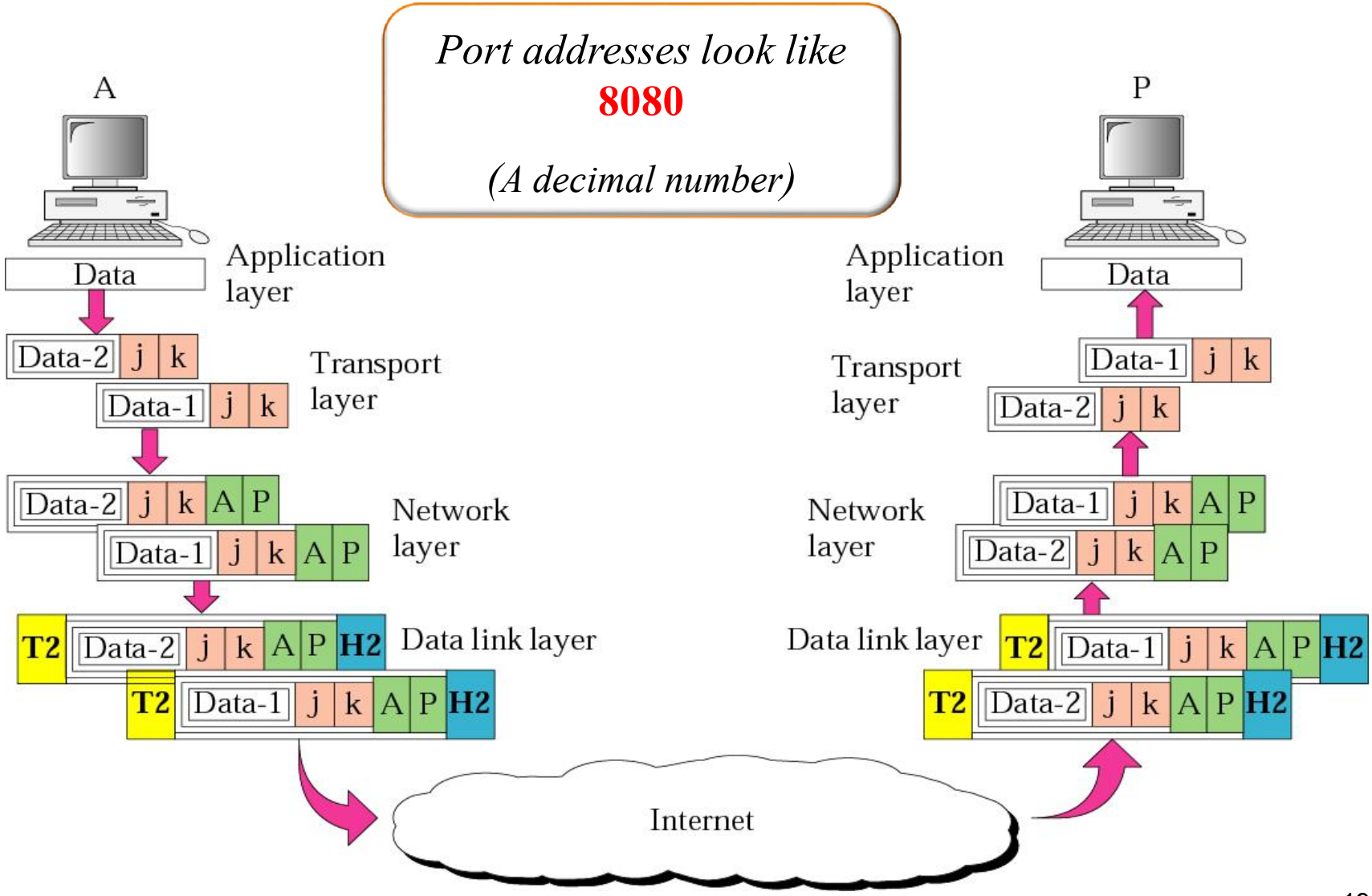
(A 6-byte (12 hexadecimal digits) physical address)

Logical Addresses (IP Addresses)



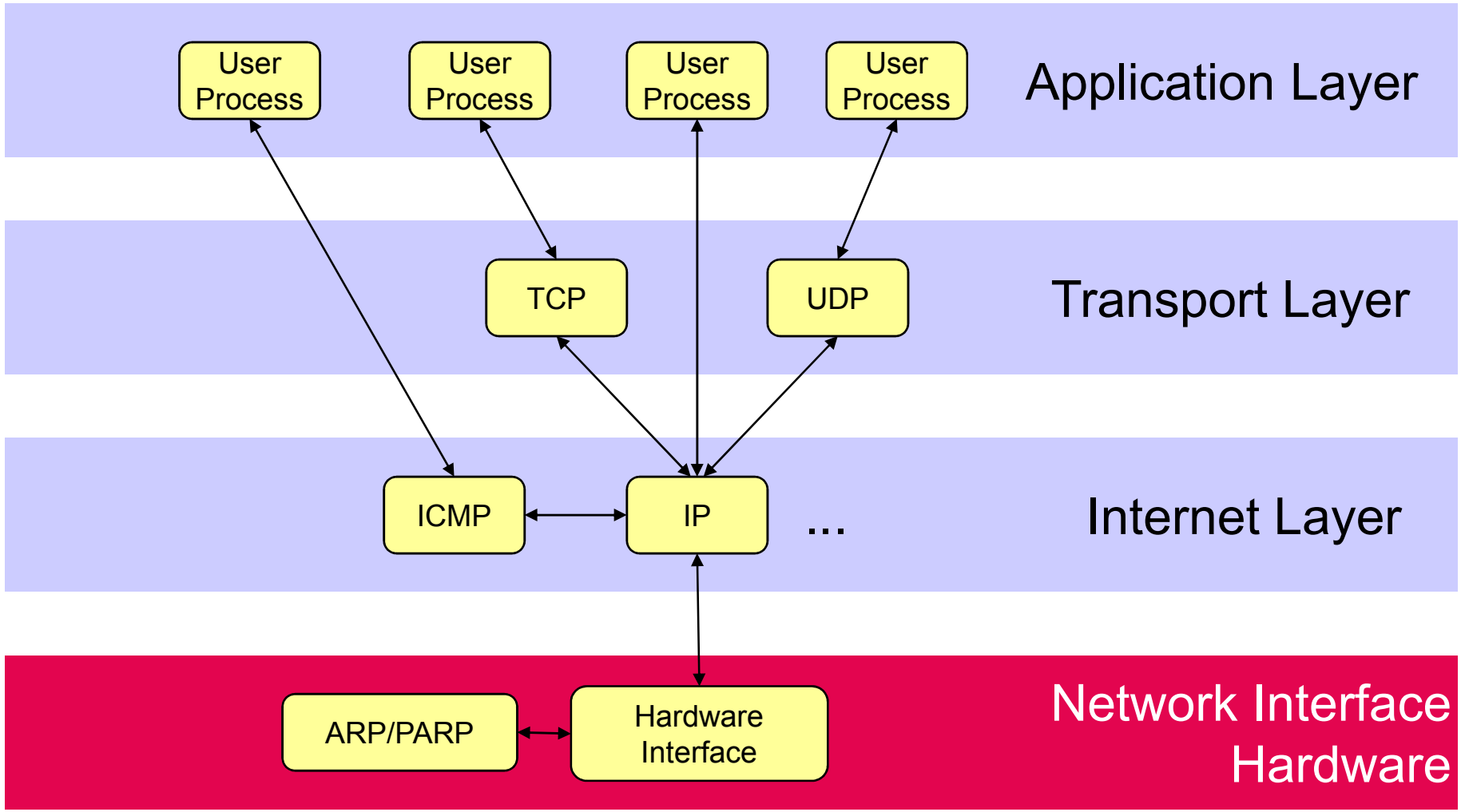
In fact, IP addresses look like
132.24.75.9
(An IPv4 address)

Port Addresses



TCP/IP Layers

Network Interface Hardware



Network Interface Hardware

- Protocol examples
  - **Ethernet**
    - Encapsulation of higher layer packets is defined in RFC 894
  - **PPP**
    - Point-to-Point Protocol for serial lines (defined in RFCs 1332, 1548)
  - **ARP/Proxy ARP (PARP)**
    - Address Resolution Protocol (RFC 826)
    - Address resolution from 32-bit IP addresses to hardware addresses (e.g. 48-bit)

- **MTU**
  - Maximum Transfer Unit
  - Maximum IP packet size in bytes (e.g. for Ethernet: 1500, X.25 Frame Relay: 576)

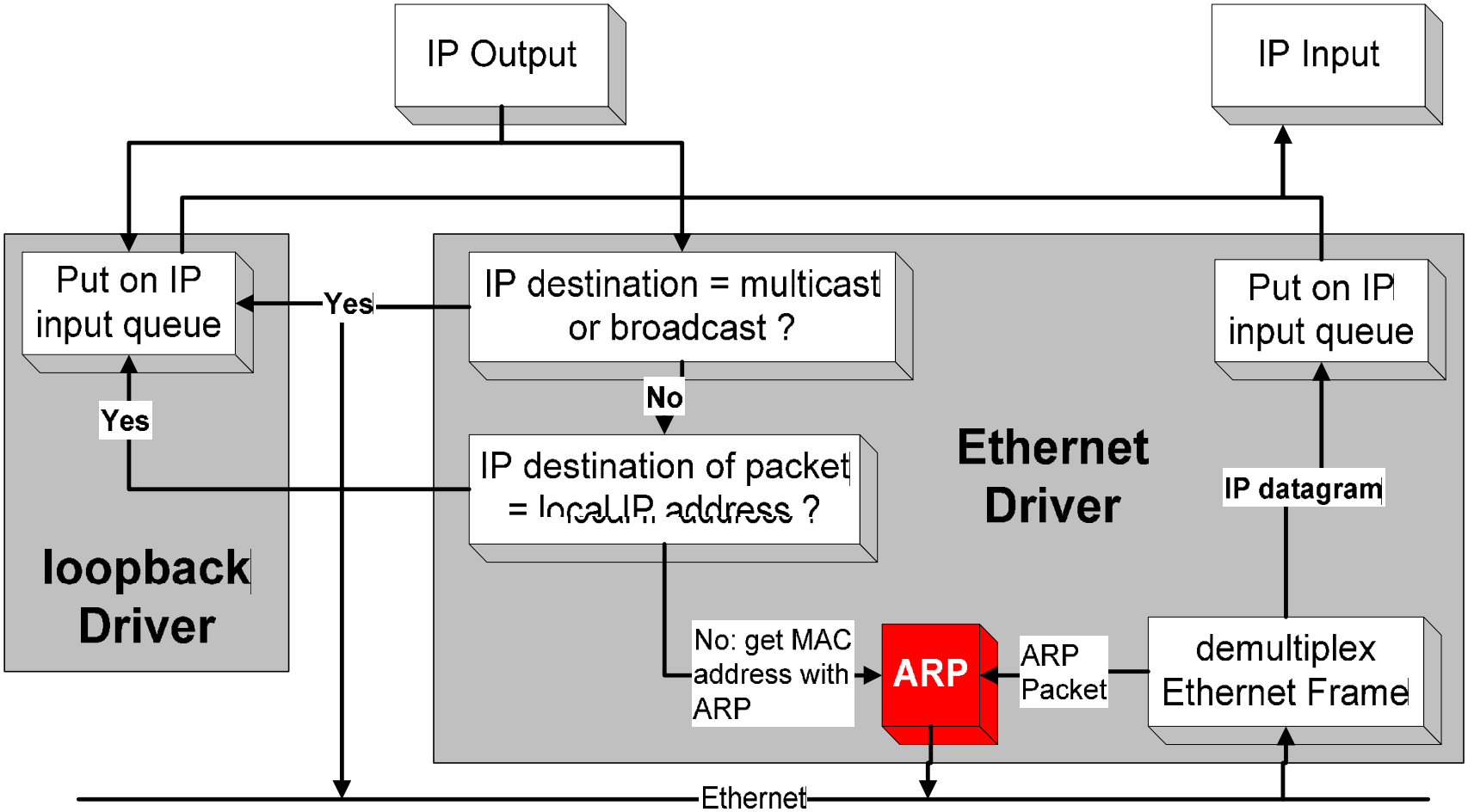
Network Interface Hardware

- Path MTU
  - Smallest MTU of any data link in the path between two hosts
  - Used to avoid IP fragmentation

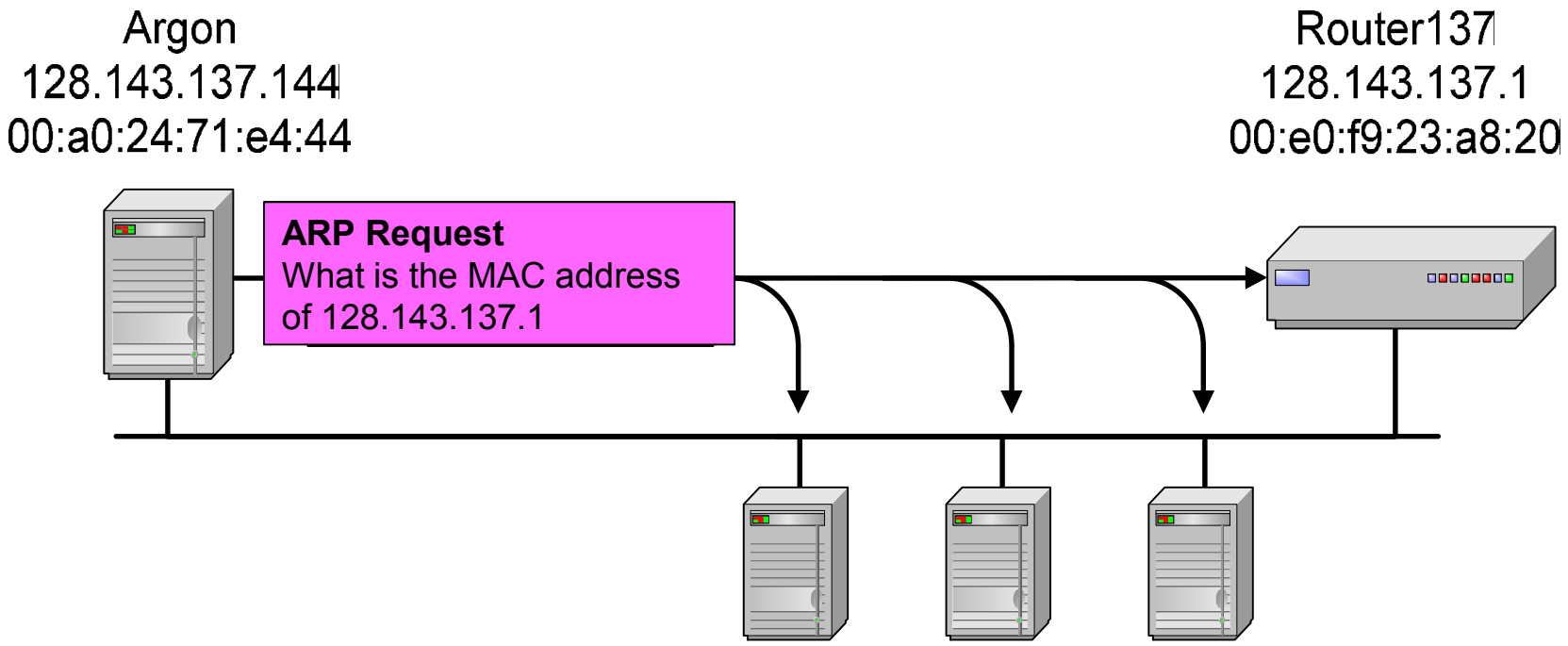
- Loopback Interface
  - Enables a client application to connect to the corresponding server application on the same
    - **localhost** = 127.0.0.1
  - Implemented at the link layer, i.e. full processing of transport and IP layers

Address Resolution Protocol (ARP) / Proxy ARP

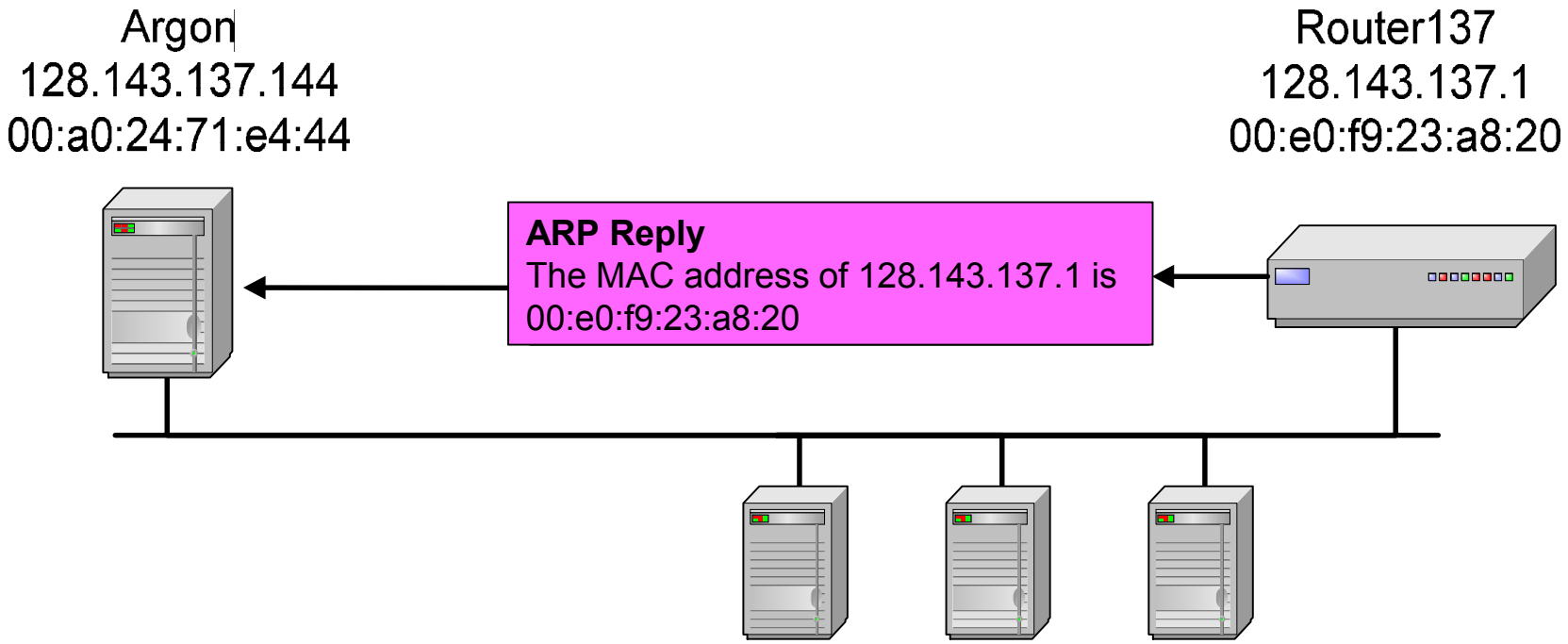
- Translate between IP addresses and MAC layer addresses



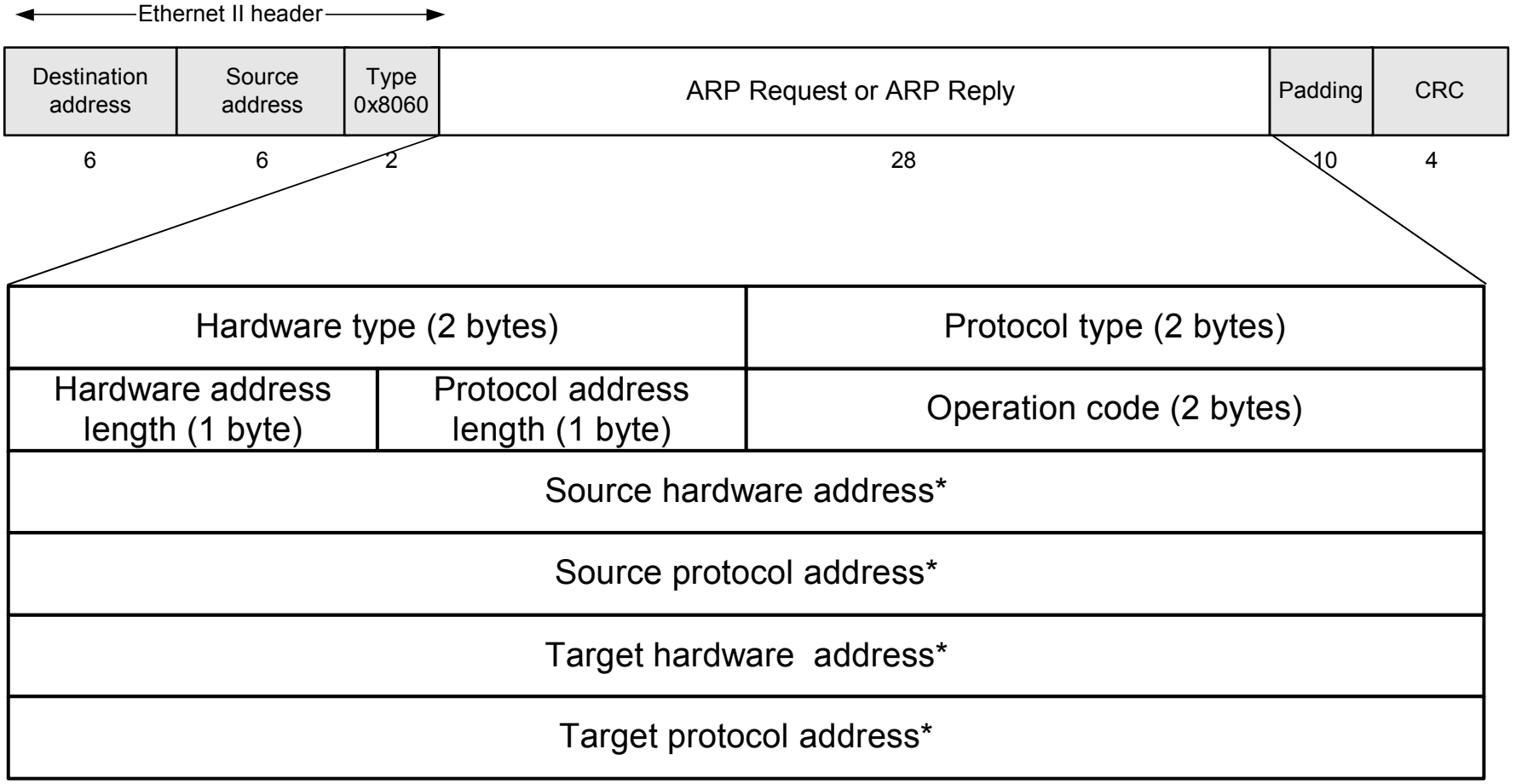
ARP - Operation Overview (Example)



ARP - Operation Overview (Example)



ARP Packet – Header Format



* Note: The length of the address fields is determined by the corresponding address length fields

ARP Packet – Header Format

- Header type (2 byte)
  - Specifies the hardware type used by the local network transmitting the message. Some common used value is:

HRD Value	Hardware Type
1	Ethernet (10 Mb)
6	IEEE 802 Networks
7	ARCNET
15	Frame Relay
16	Asynchronous Transfer Mode (ATM)
17	HDLC
18	Fibre Channel
19	Asynchronous Transfer Mode (ATM)
20	Serial Line

ARP Packet – Header Format

- Protocol type (2 byte)
  - This field is the complement of the Hardware Type field
  - Specifies the type of layer three addresses used in the message
    - For IPv4 addresses, this value is 2048 (0800 hex), which corresponds to the EtherType code for the Internet Protocol

- Hardware Address Length (1 byte)
  - Specifies how long hardware addresses are in this message
    - For Ethernet or other networks using IEEE 802 MAC addresses, the value is 6

- Protocol Address Length (1 byte)
  - The complement of the hardware address length field
  - Specifies how long protocol (layer three) addresses are in this message
    - For IP(v4) addresses this value is of course 4

ARP Packet – Header Format

- Operation code (2 byte)
  - Specifies the type of the message

Opcode	ARP Message Type
1	ARP Request
2	ARP Reply
3	RARP Request
4	RARP Reply
5	DRARP Request
6	DRARP Reply
7	DRARP Error
8	InARP Request
9	InARP Reply

ARP Packet Format (Concerning the Example)

- ARP Request from Argon
  - Source hardware address: 00:a0:24:71:e4:44
  - Source protocol address: 128.143.137.144
  - Target hardware address: 00:00:00:00:00:00
  - Target protocol address: 128.143.137.1

- ARP Reply from Router137
  - Source hardware address: 00:e0:f9:23:a8:20
  - Source protocol address: 128.143.137.1
  - Target hardware address: 00:a0:24:71:e4:44
  - Target protocol address: 128.143.137.144

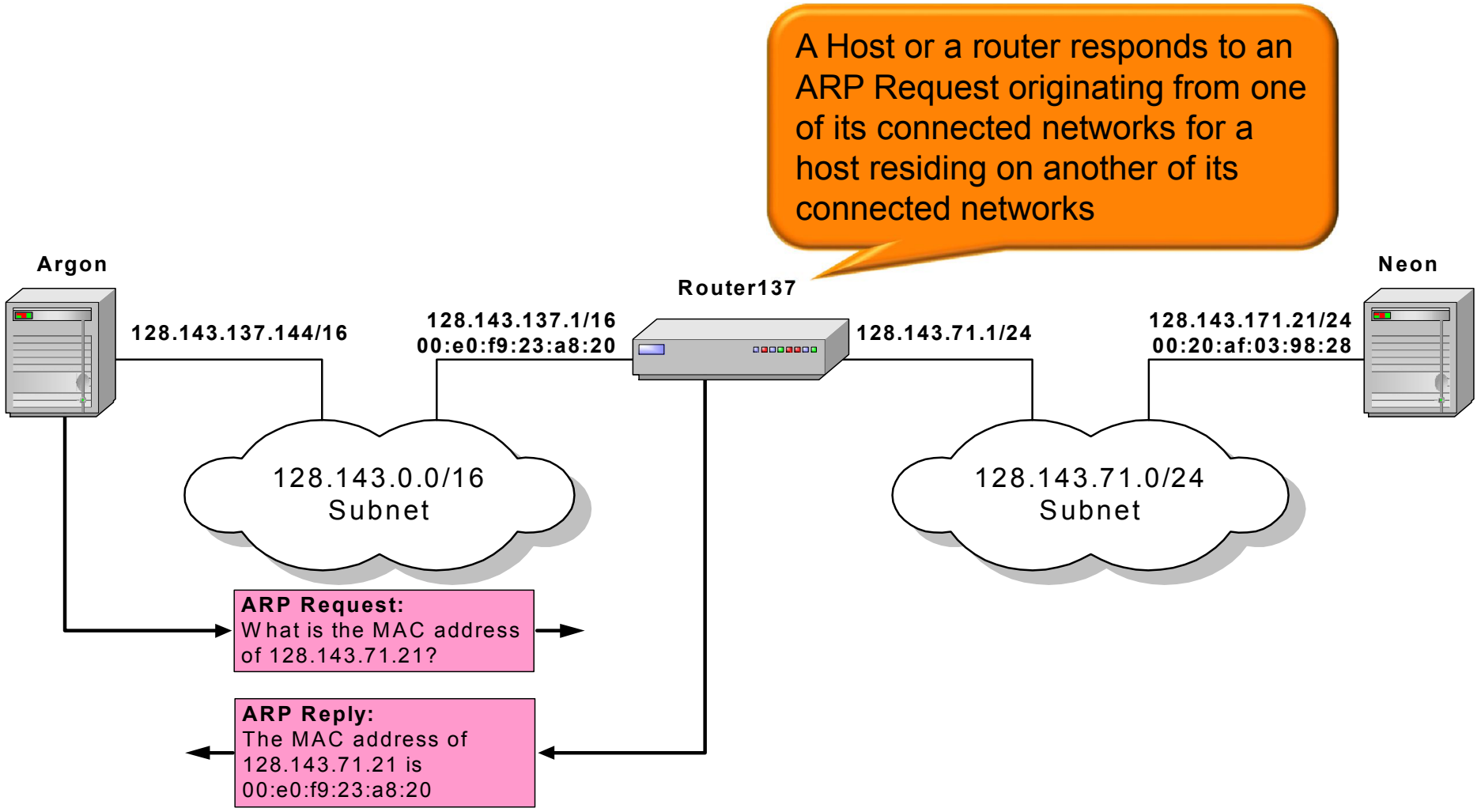
ARP Cache

- Sending an ARP request/reply for each IP datagram is inefficient
  - Hosts maintain a cache (ARP Cache) of current entries
  - ARP Cache expires after 20 minutes, if not refreshed

- Contents of the ARP Cache (an example)

  (128.143.71.37) at 00:10:4B:C5:D1:15 [ether] on eth0
  (128.143.71.36) at 00:B0:D0:E1:17:D5 [ether] on eth0
  (128.143.71.35) at 00:B0:D0:DE:70:E6 [ether] on eth0
  (128.143.136.90) at 00:05:3C:06:27:35 [ether] on eth1
  (128.143.71.34) at 00:B0:D0:E1:17:DB [ether] on eth0
  (128.143.71.33) at 00:B0:D0:E1:17:DF [ether] on eth0

Proxy ARP (PARP)

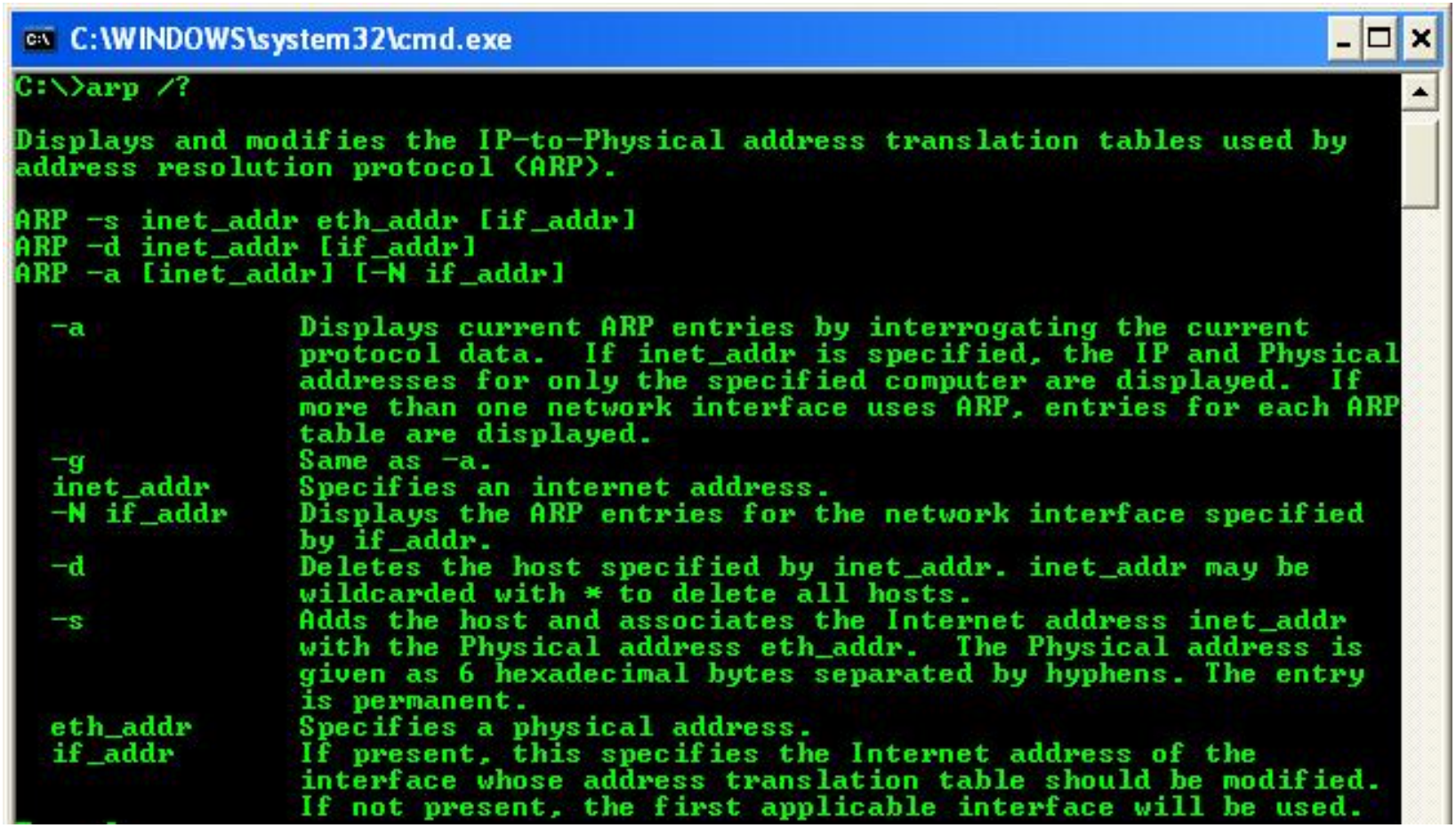


Vulnerabilities of ARP

- ARP does not authenticate requests or replies
  - ARP requests and replies can be forged
- ARP is stateless
  - ARP replies can be sent without a corresponding ARP request
    - The node that receives an ARP reply/request must update its local ARP cache with the information in the source fields, if the receiving node already has an entry for the IP address of the source in its ARP cache

- Typical exploitation of these vulnerabilities
  - A forged ARP request or reply can be used to update the ARP cache of a remote system with a forged entry (ARP poisoning)
  - This can be used to redirect IP traffic to other hosts

Some Praxis


C:\WINDOWS\system32\cmd.exe
C:\>arp /?
Displays and modifies the IP-to-Physical address translation tables used by
address resolution protocol (ARP).
ARP -s inet_addr eth_addr [if_addr]
ARP -d inet_addr [if_addr]
ARP -a [inet_addr] [-N if_addr]
-a Displays current ARP entries by interrogating the current
protocol data. If inet_addr is specified, the IP and Physical
addresses for only the specified computer are displayed. If
more than one network interface uses ARP, entries for each ARP
table are displayed.
-g Same as -a.
inet_addr Specifies an internet address.
-N if_addr Displays the ARP entries for the network interface specified
by if_addr.
-d Deletes the host specified by inet_addr. inet_addr may be
wildcarded with * to delete all hosts.
-s Adds the host and associates the Internet address inet_addr
with the Physical address eth_addr. The Physical address is
given as 6 hexadecimal bytes separated by hyphens. The entry
is permanent.
eth_addr Specifies a physical address.
if_addr If present, this specifies the Internet address of the
interface whose address translation table should be modified.
If not present, the first applicable interface will be used.

Some Praxis

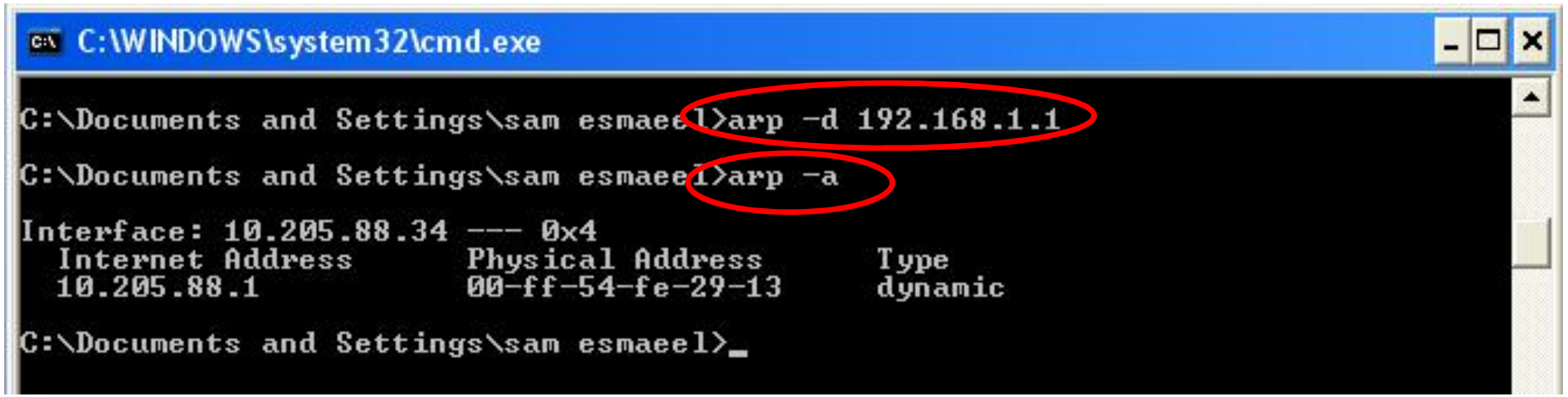
Internet

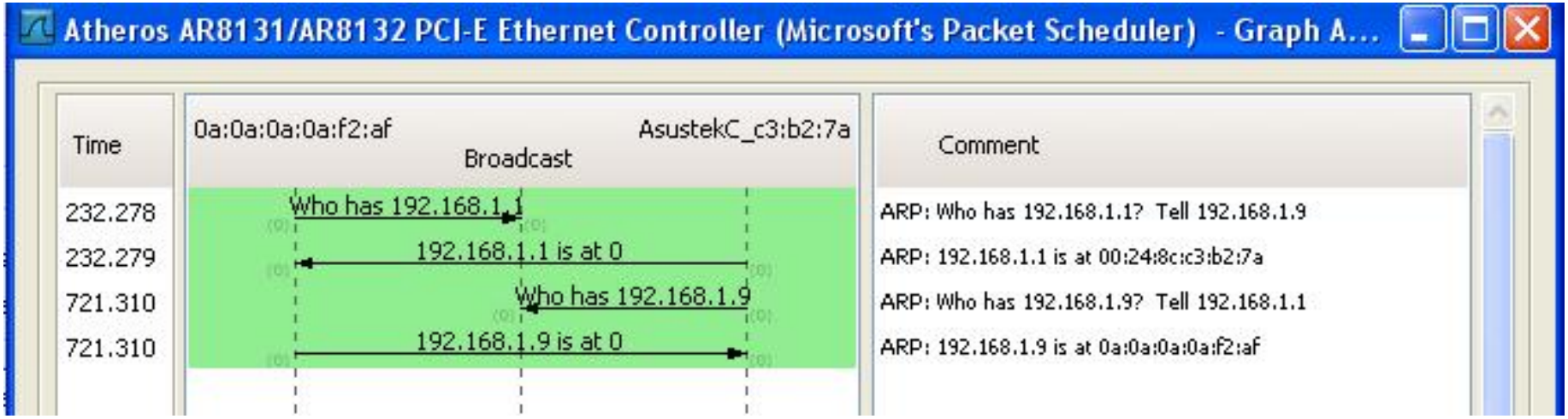
192.168.1.9

192.168.1.1

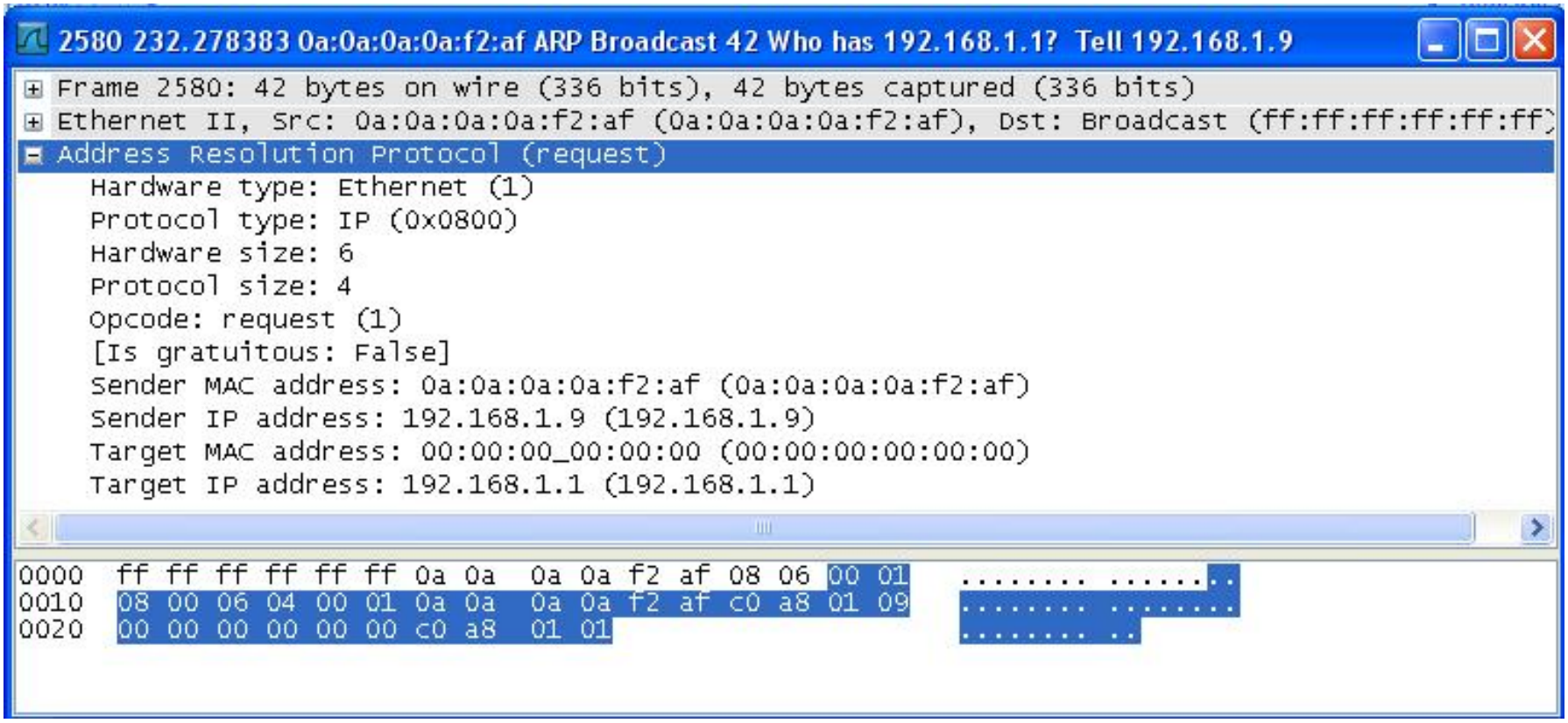
```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\sam esmaeel>arp -a

Interface: 192.168.1.9 --- 0x2
  Internet Address      Physical Address      Type
  192.168.1.1           00-24-8c-c3-b2-7a     dynamic

Interface: 10.205.88.34 --- 0x4
  Internet Address      Physical Address      Type
  10.205.88.1           00-ff-54-fe-29-13     dynamic

C:\Documents and Settings\sam esmaeel>
```

Some Praxis

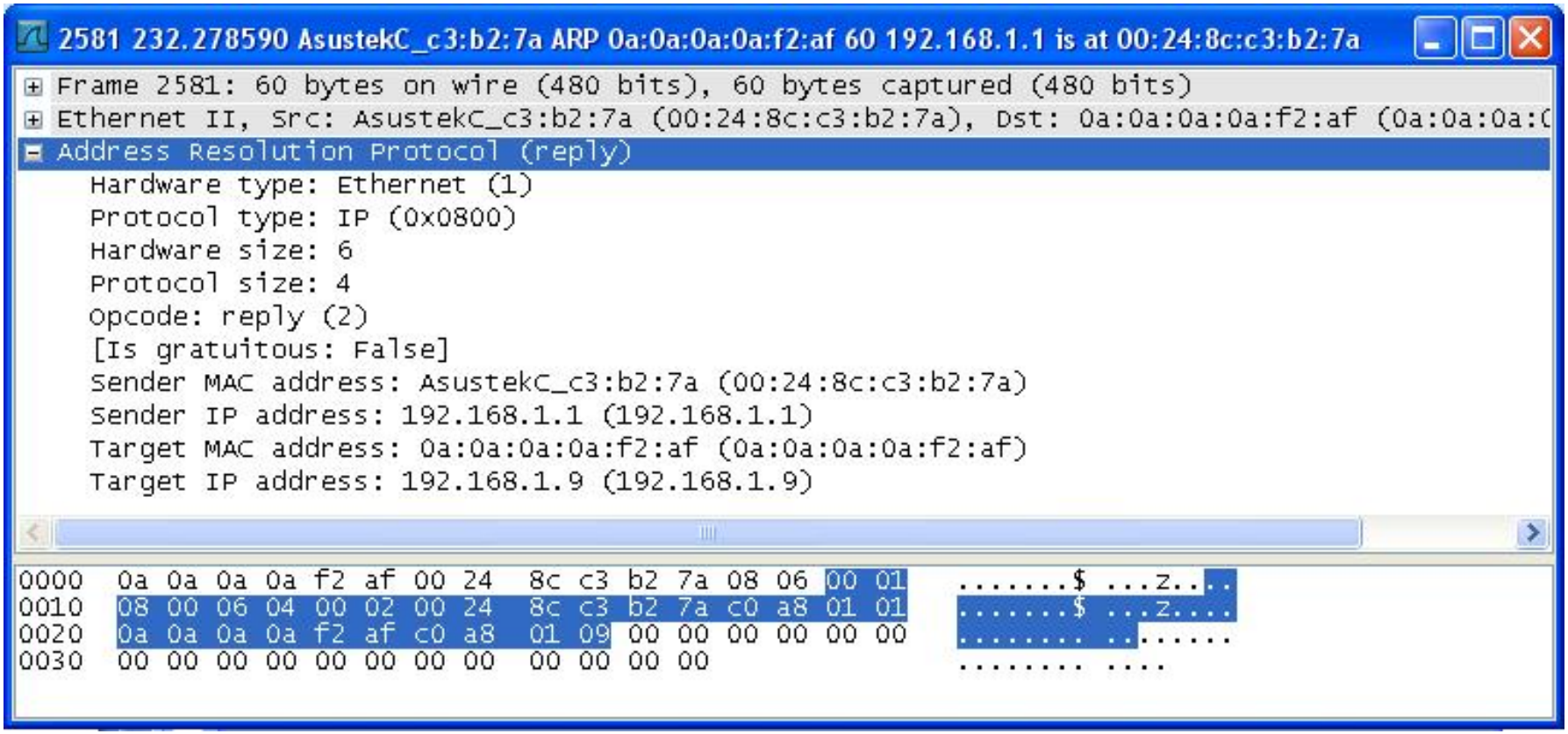

C:\WINDOWS\system32\cmd.exe
C:\Documents and Settings\sam esmaeel>arp -d 192.168.1.1
C:\Documents and Settings\sam esmaeel>arp -a
Interface: 10.205.88.34 --- 0x4
Internet Address Physical Address Type
10.205.88.1 00-ff-54-fe-29-13 dynamic
C:\Documents and Settings\sam esmaeel>


Atheros AR8131/AR8132 PCI-E Ethernet Controller (Microsoft's Packet Scheduler) - Graph A...
Time
0a:0a:0a:0a:f2:af
AsustekC_c3:b2:7a
Broadcast
Comment
232.278
Who has 192.168.1.1
ARP: Who has 192.168.1.1? Tell 192.168.1.9
232.279
192.168.1.1 is at 0
ARP: 192.168.1.1 is at 00:24:8c:c3:b2:7a
721.310
Who has 192.168.1.9
ARP: Who has 192.168.1.9? Tell 192.168.1.1
721.310
192.168.1.9 is at 0
ARP: 192.168.1.9 is at 0a:0a:0a:0a:f2:af

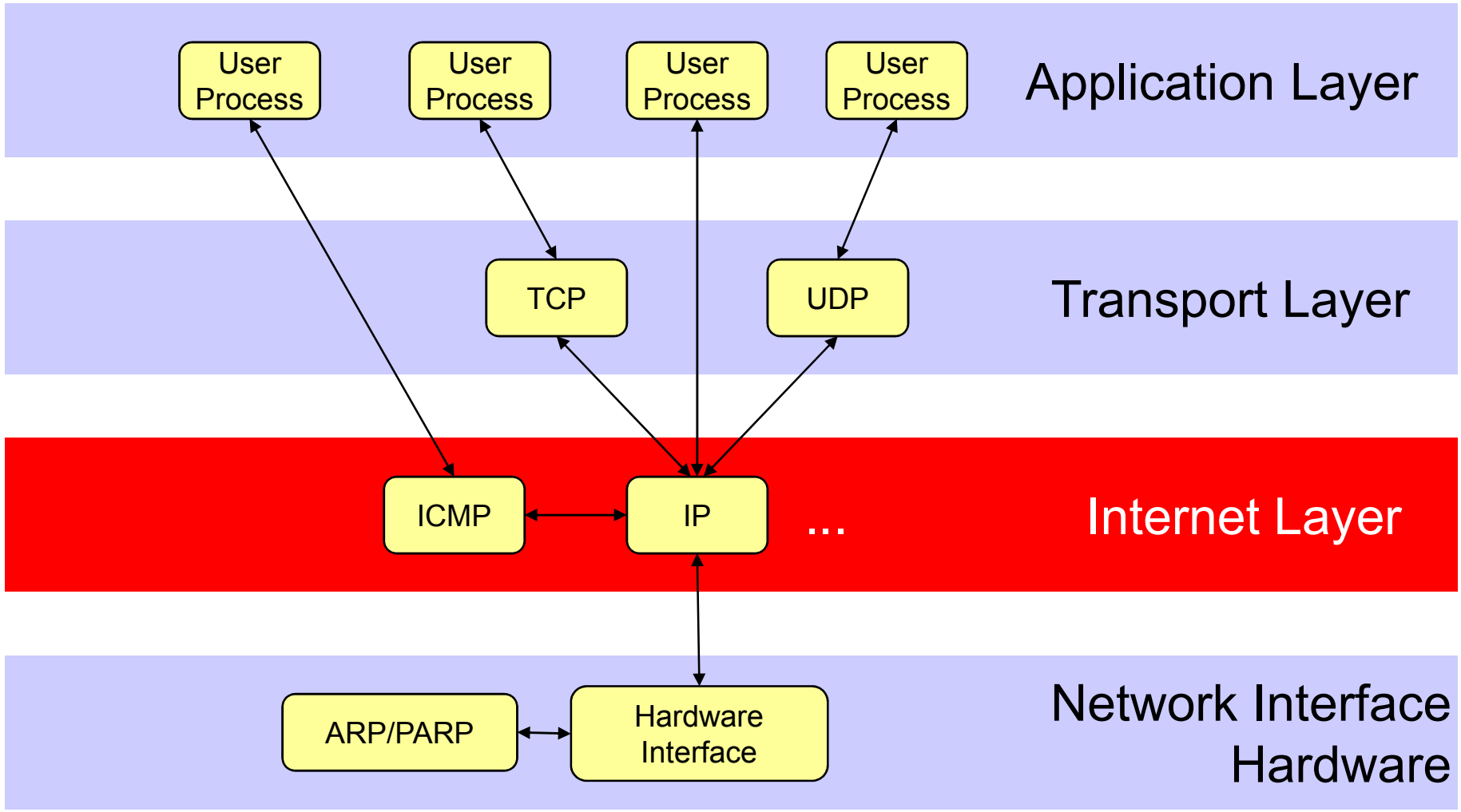
Some Praxis – ARP Request



Some Praxis – ARP Reply



Internet Layer



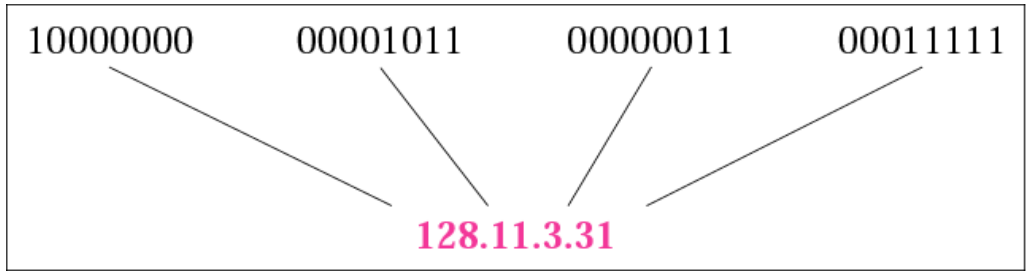
The Internet Protocol version 4 (IPv4) - Overview

- Specified in RFC 791
- Designed for use in interconnected systems of packet-switched computer communication networks, also called **catenet**
- Povides **routing** of data between hosts based on the IP address
  - IPv4 uses 32-bit IP addresses
  - IPv6 uses 128-bit IP addresses
- Provides fragmentation and reassembly of long datagrams, if necessary, for transmission through "small packet" networks
- Unreliable, connectionless datagram delivery service, possible packet loss, possible out-of-order delivery, possible duplication, etc.

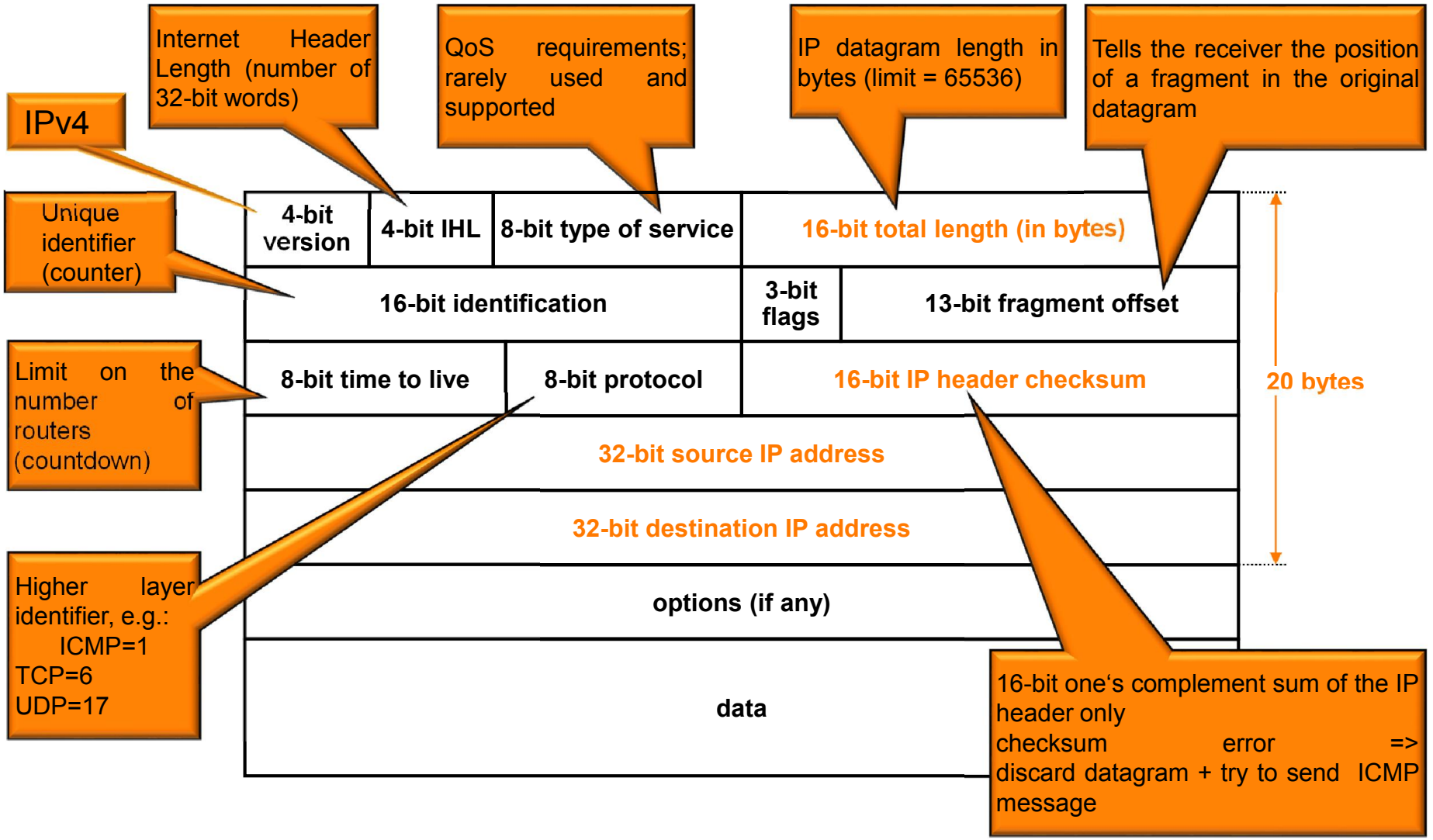
The Internet Protocol version 4 (IPv4) - Overview

- IP fragmentation is used on any link with MTU < original datagram length
  - Duplicates IP header for each fragment and sets flags for re-assembly
  - Re-assembly at the receiving host only, never in the network

- Applications use the Domain Name Service (**DNS**) to convert hostnames to IP addresses and vice-versa
  - www.albaath-uni.sy → 169.254.0.0
  - 169.254.0.0 → www.albaath-uni.sy
  - An arbitrary example



Header Format



Type of Sevice (TOS) Field

- Provides an indication of the abstract parameters of the quality of service desired

- Used to guide the selection of the actual service parameters when transmitting a datagram through a particular network

Precedence	D	T	R	0	0

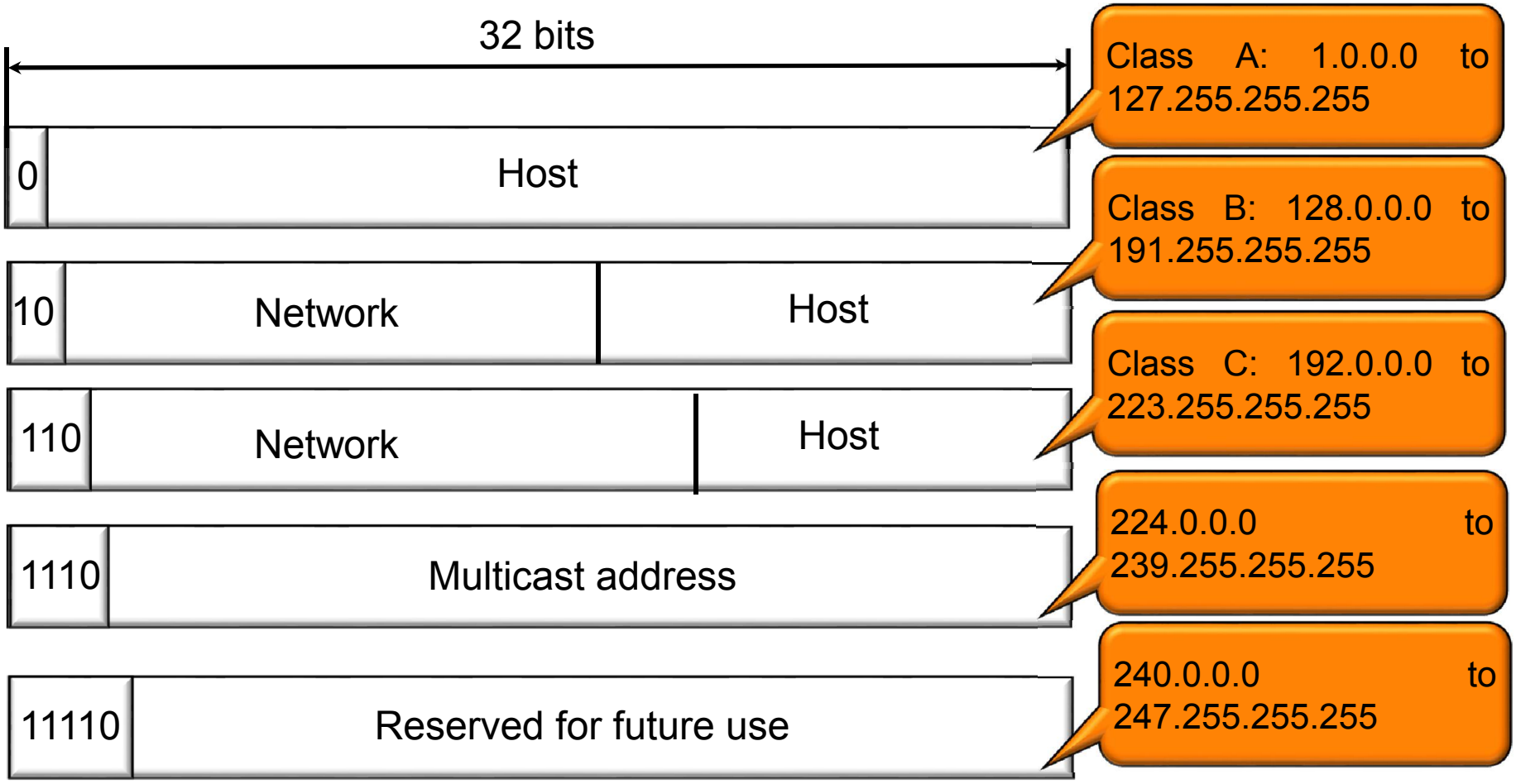
- Precedence: 3 bits determine which kind of traffic the packet transfers (e.g. control, multicast), which priority, etc.
- D: 1 bit stands for the delay bound (0: normal, 1: low)
- T: 1 bit stands for the throughput (0: normal, 1: high)
- R: 1 bit, stands for the reliability (0: normal, 1: high)
- Remaining bits are set to 0 and reserved for future use

Addressing in IPv4 World

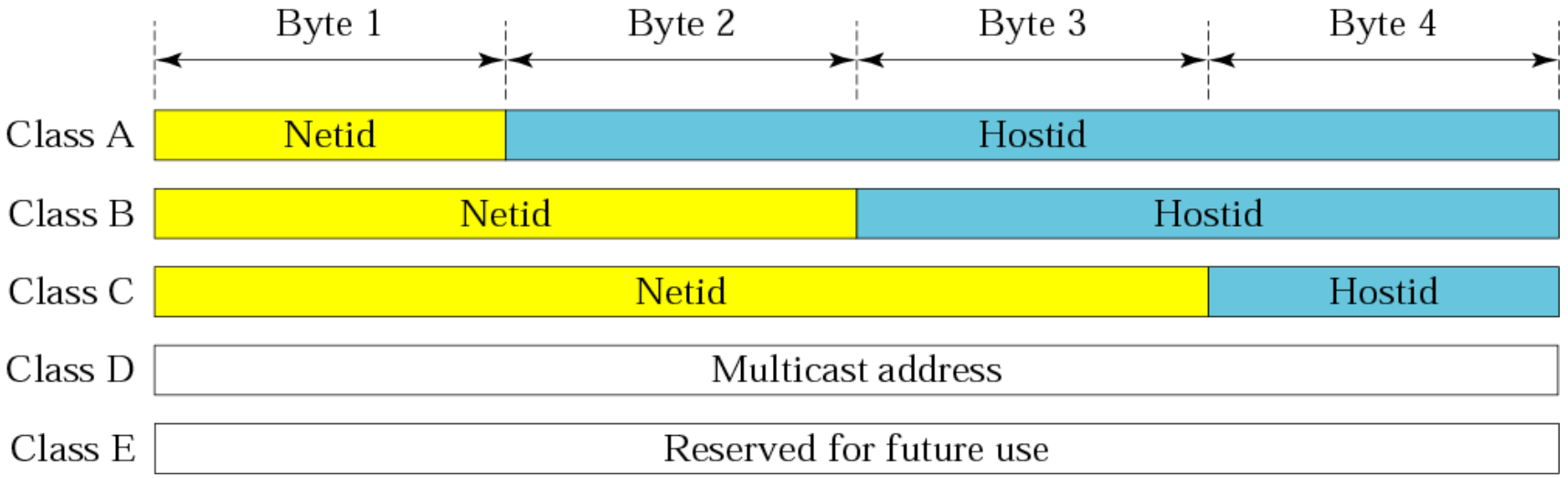
Classfull Adressing

Classless Addressing

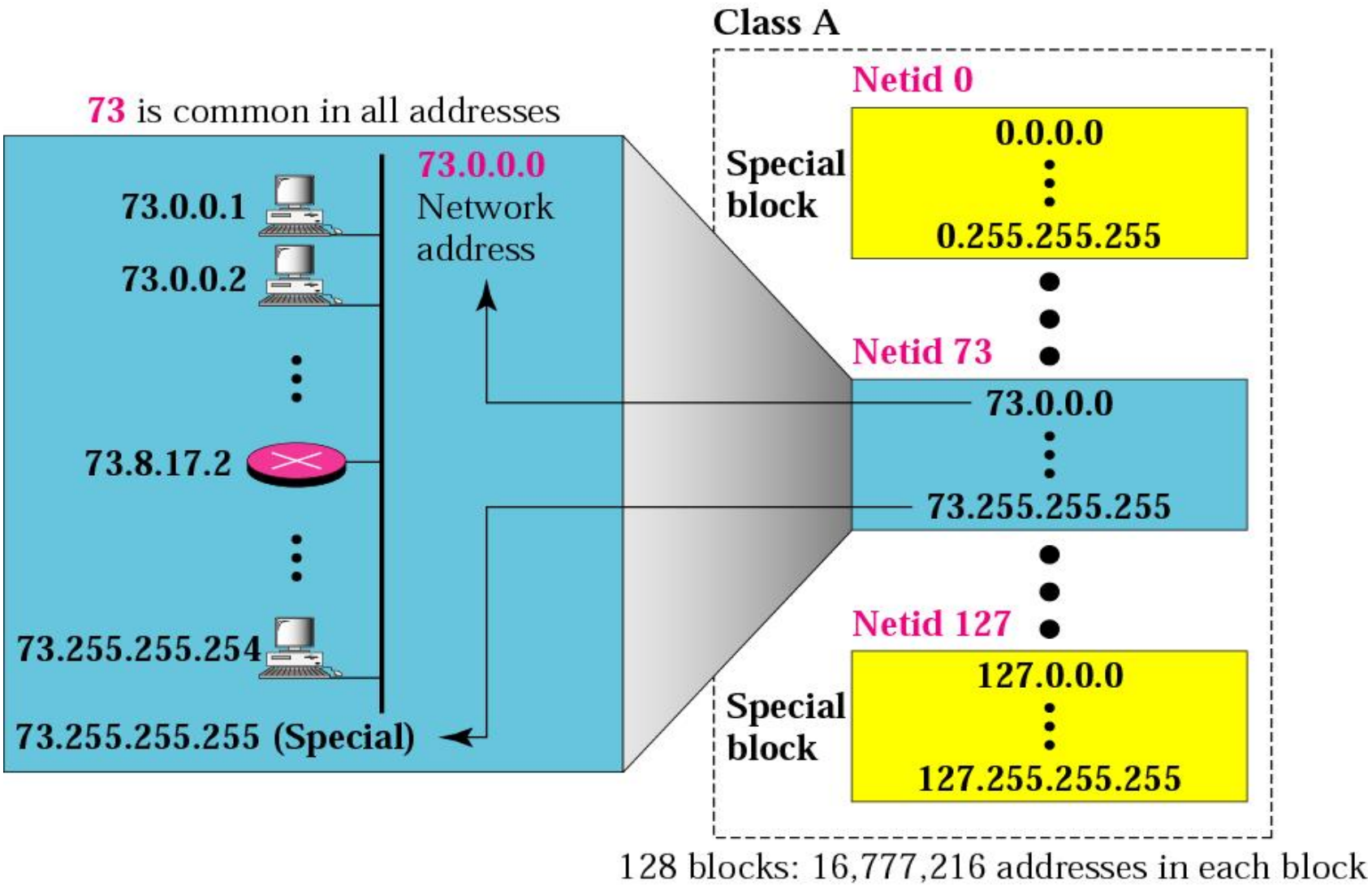
Classful Addressing



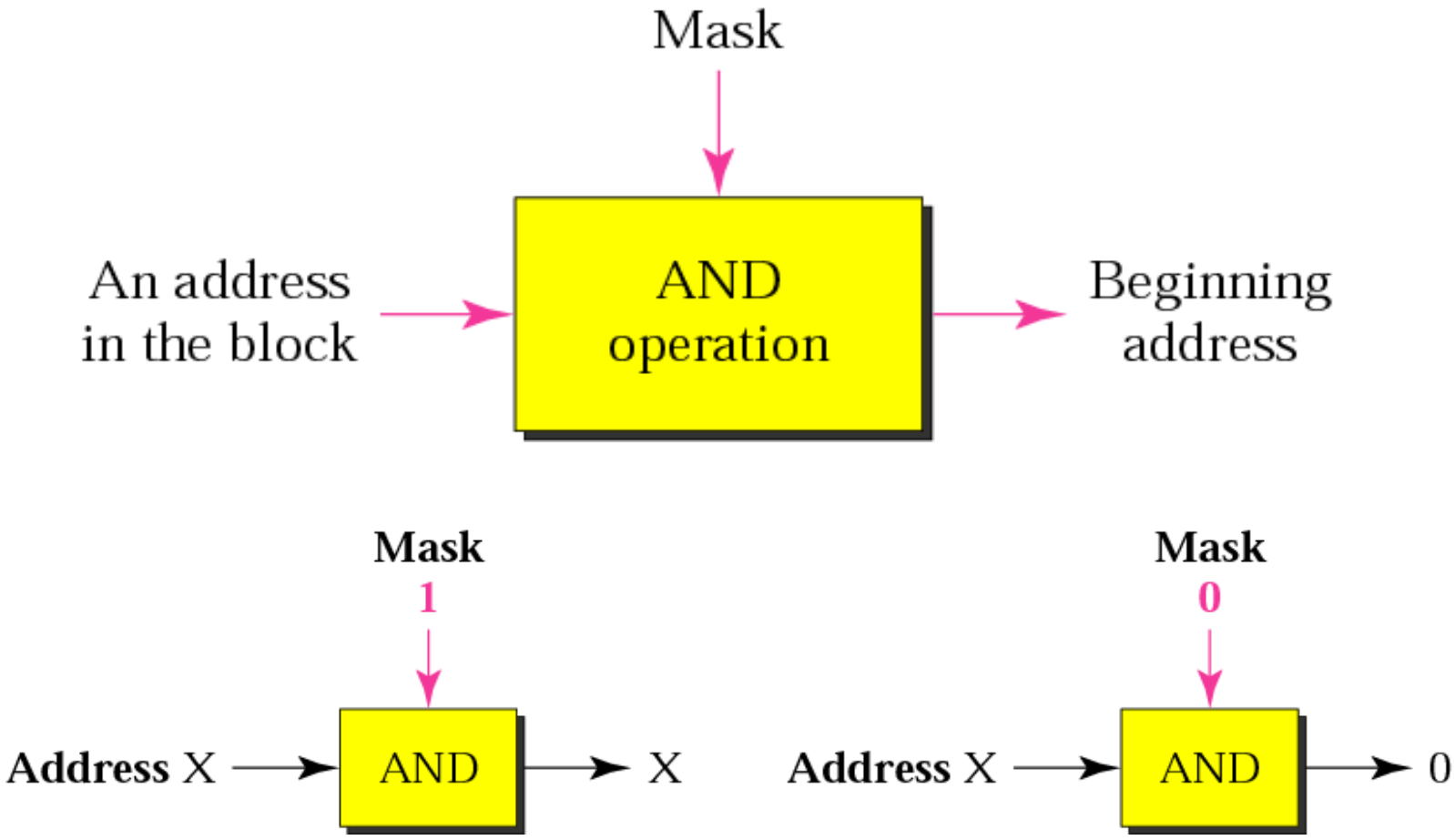
Classful Addressing



Classful Addressing – Class A



Masking Concept



Masking Concept

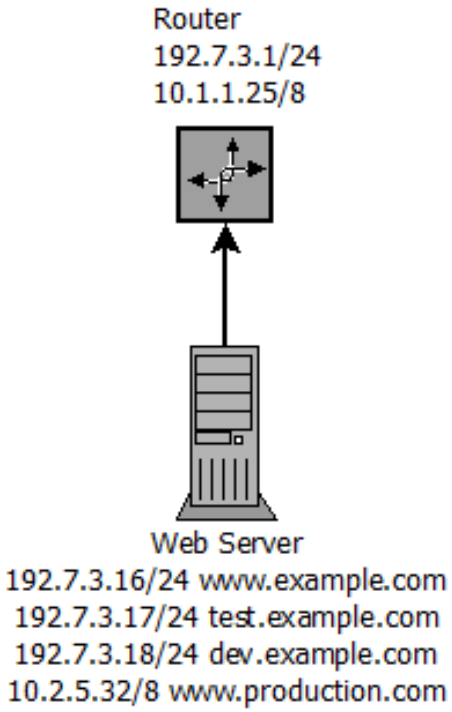
- Default masks

Class	*Mask in binary*	*Mask in dotted-decimal*
A	11111111 00000000 00000000 00000000	255.0.0.0
B	11111111 11111111 00000000 00000000	255.255.0.0
C	11111111 11111111 11111111 00000000	255.255.255.0

- Network address
  - The beginning address of each block
  - Can be found by applying the default mask to any of the addresses in the block (including itself). It retains the network address (**NetId**) of the block and sets the host address (**Hostid**) to zero.

Multihomed Devices

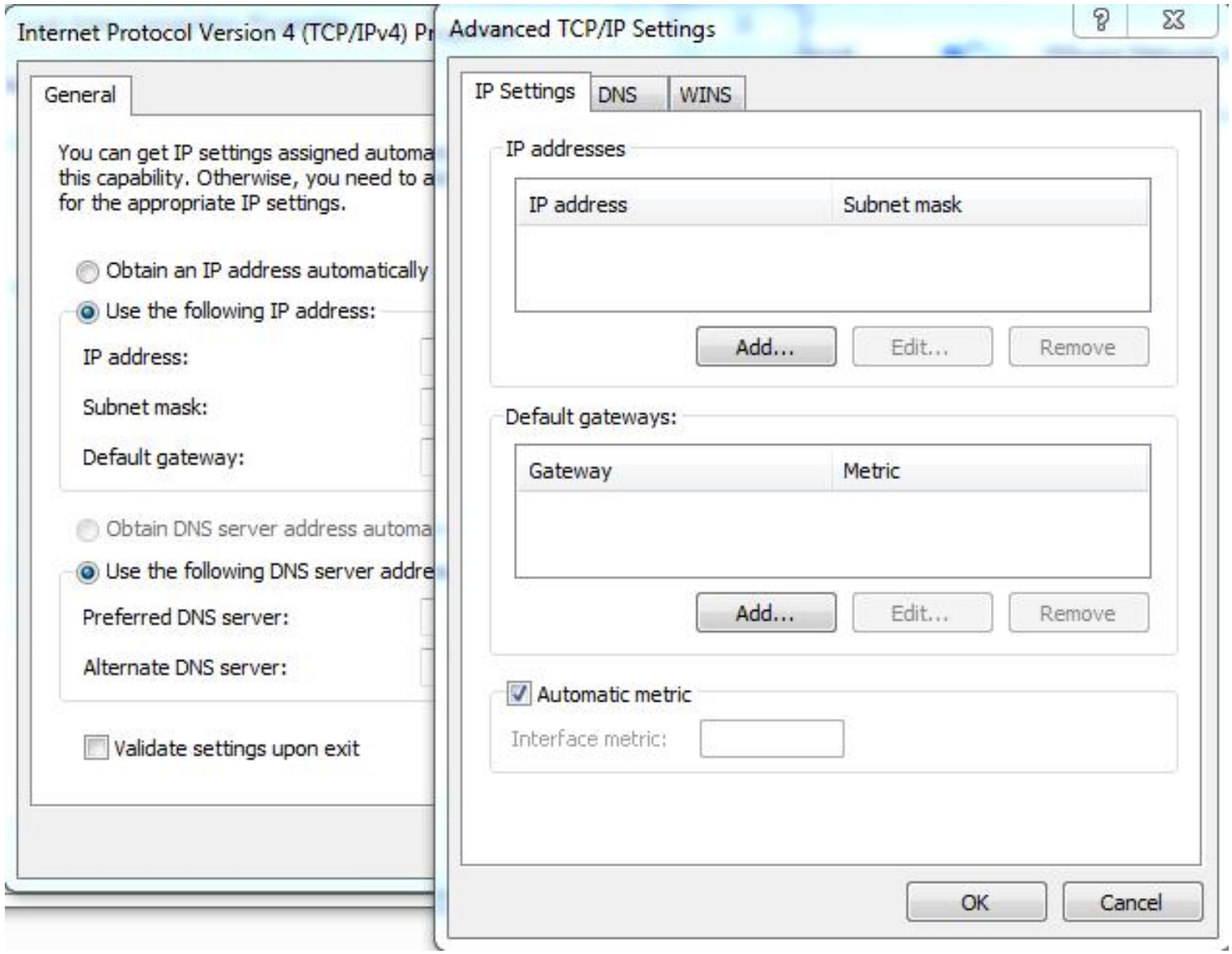
- Multihoming
  - Connecting a single host to multiple networks
    - E.g., a mobile phone might be simultaneously connected to a WiFi network and a 3G network
    - assigning multiple addresses to a single web server with each address representing an individual domain on the server



Multihomed Devices

- Some practice
  - **Network and Sharing Center → Change adapter settings**
  - Right-click on the adapter you want to modify and select **Properties → Internet Protocol Version 4 → Properties**
  - Select the **Use the following IP address** radio button and enter the appropriate primary IP address, subnet mask, and default gateway
    - You should also manually set the DNS server appropriately for this network connection
  - Click on the **Advanced** button
  - Add additional IP addresses and subnet masks in the upper portion and default gateways beneath it
  - Click **Automatic metric**
    - This allows Windows to automatically assign metrics, it will determine the metric for each link based on the link speed with the fastest ones having the lowest metric (higher priority)

Multihomed Devices

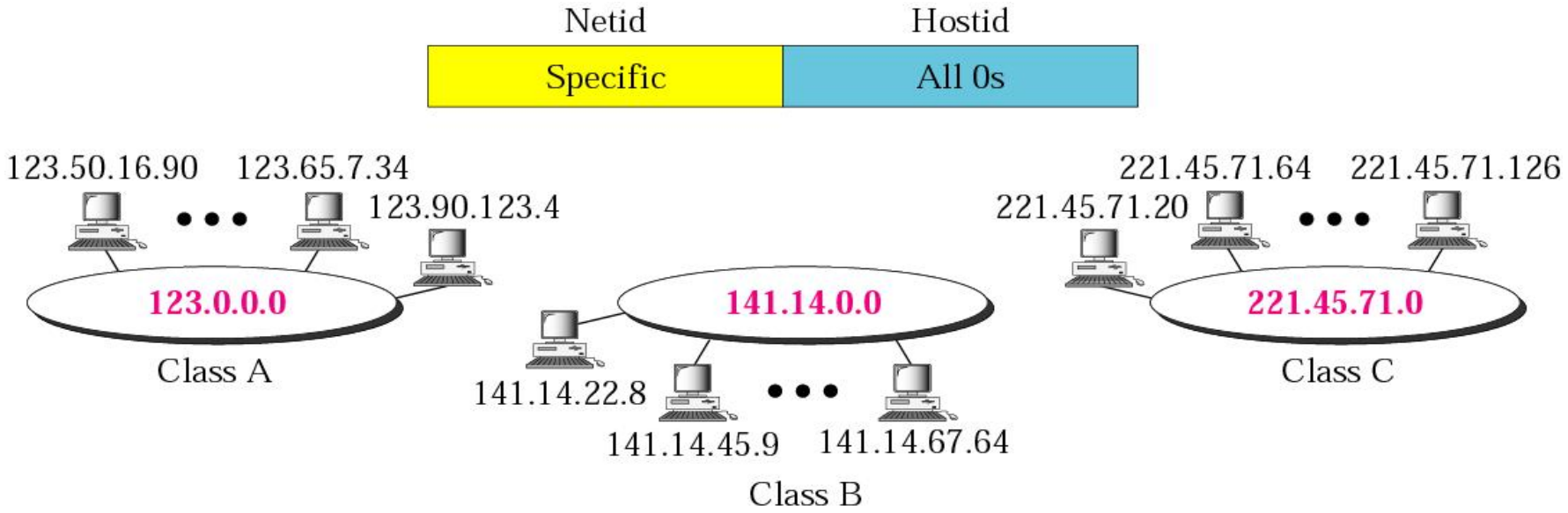


Special Addresses

Special Address	*Netid*	*Hostid*	*Source or Destination*
Network address	Specific	All 0s	None
Direct broadcast address	Specific	All 1s	Destination
Limited broadcast address	All 1s	All 1s	Destination
This host on this network	All 0s	All 0s	Source
Specific host on this network	All 0s	Specific	Destination
Loopback address	127	Any	Destination

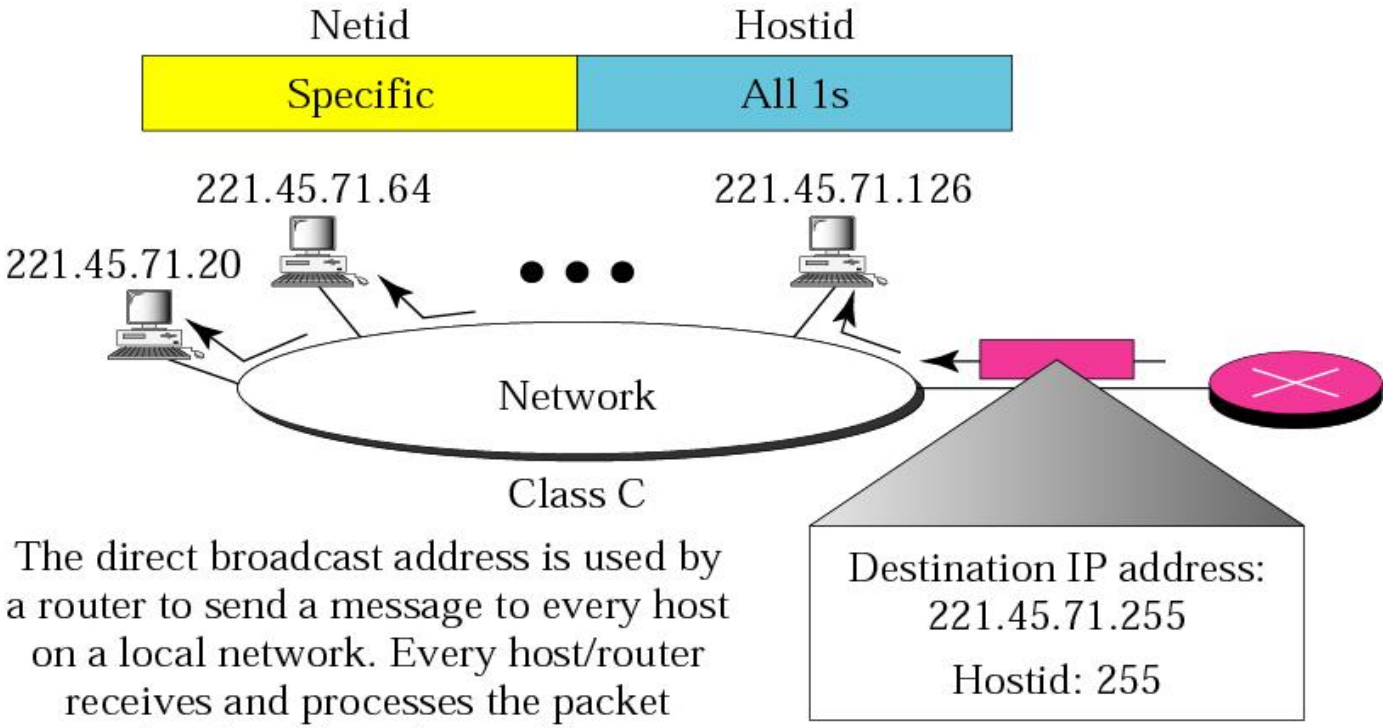
Special Addresses - Examples

- Network address



Special Addresses - Examples

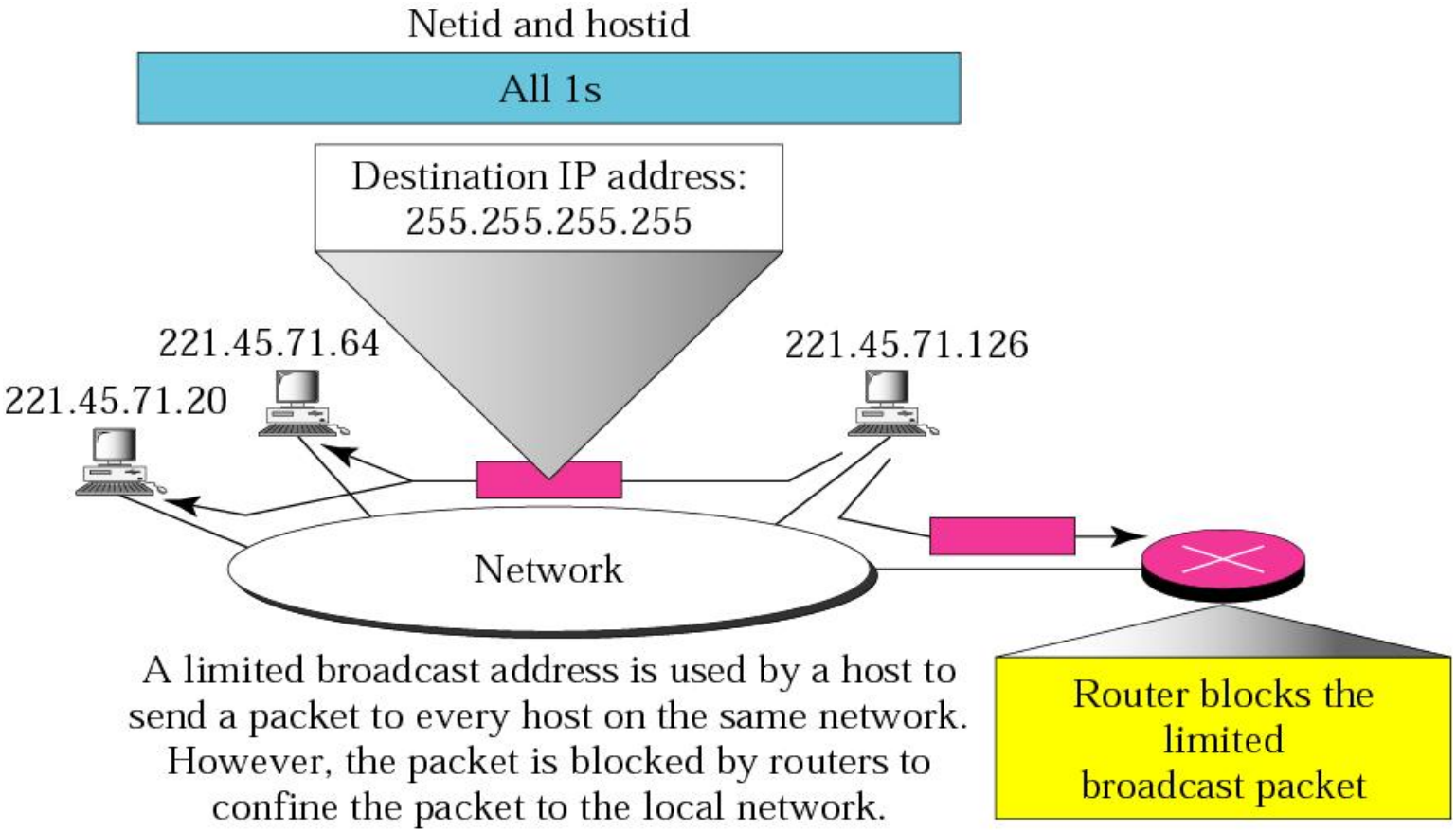
- Direct broadcast address



The direct broadcast address is used by a router to send a message to every host on a local network. Every host/router receives and processes the packet with a direct broadcast address.

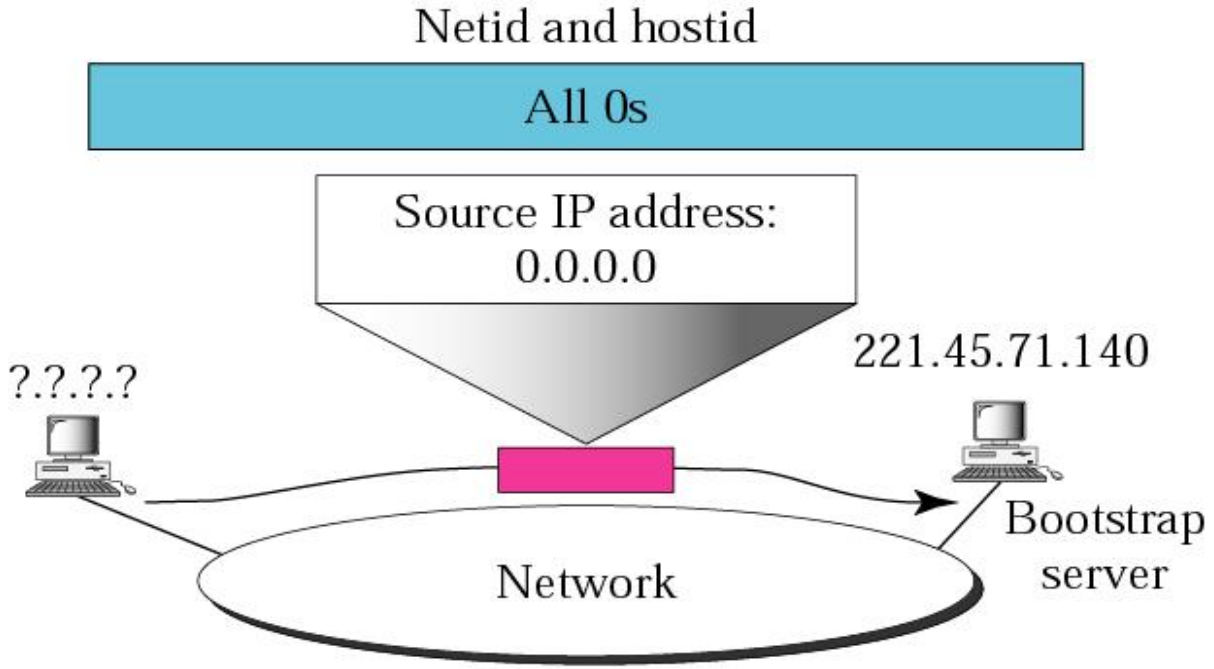
Special Addresses - Examples

- Limited broadcast address



Special Addresses - Examples

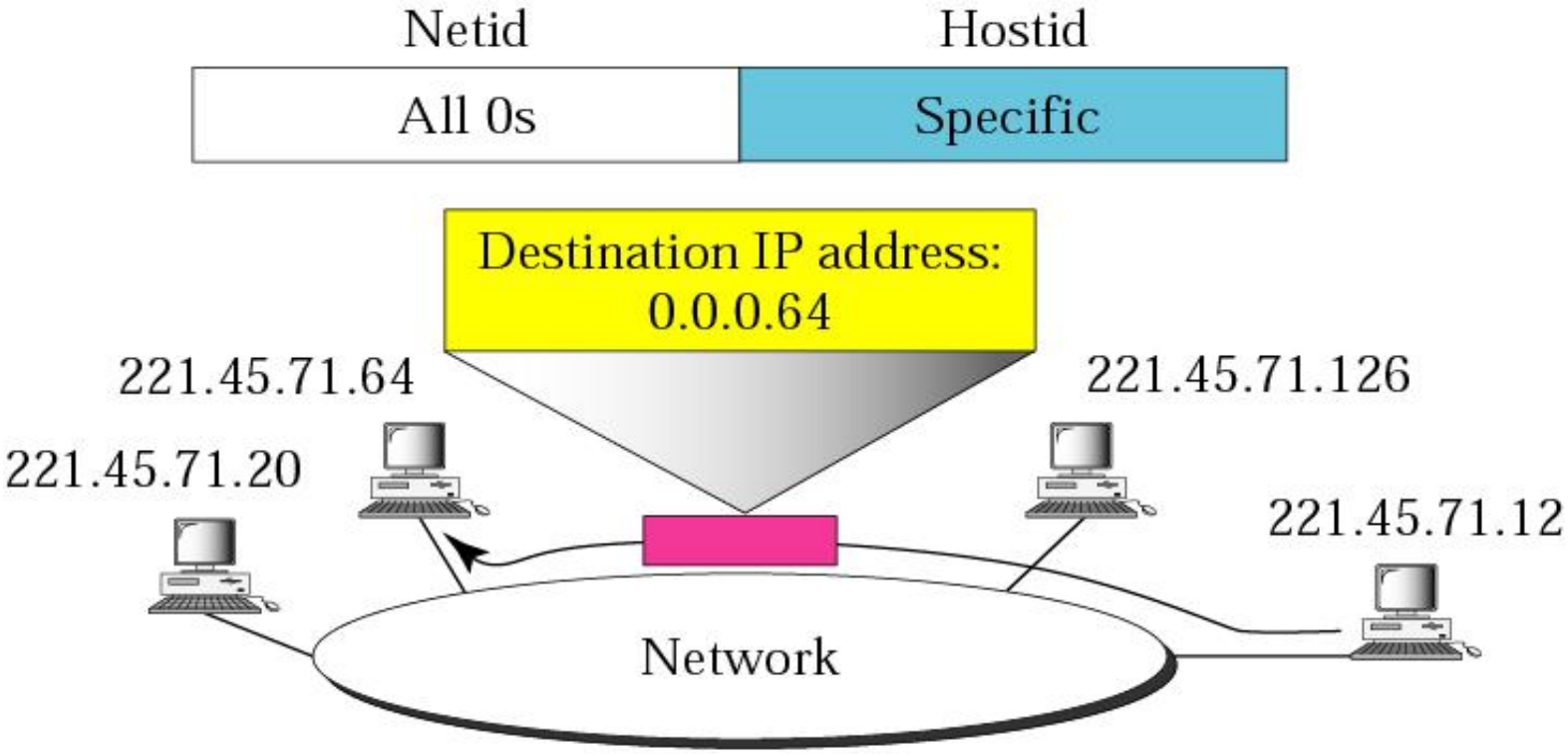
- This host on this network



A host that does not know its IP address uses the IP address 0.0.0.0 as the source address and 255.255.255.255 as the destination address to send a message to a bootstrap server.

Special Addresses - Examples

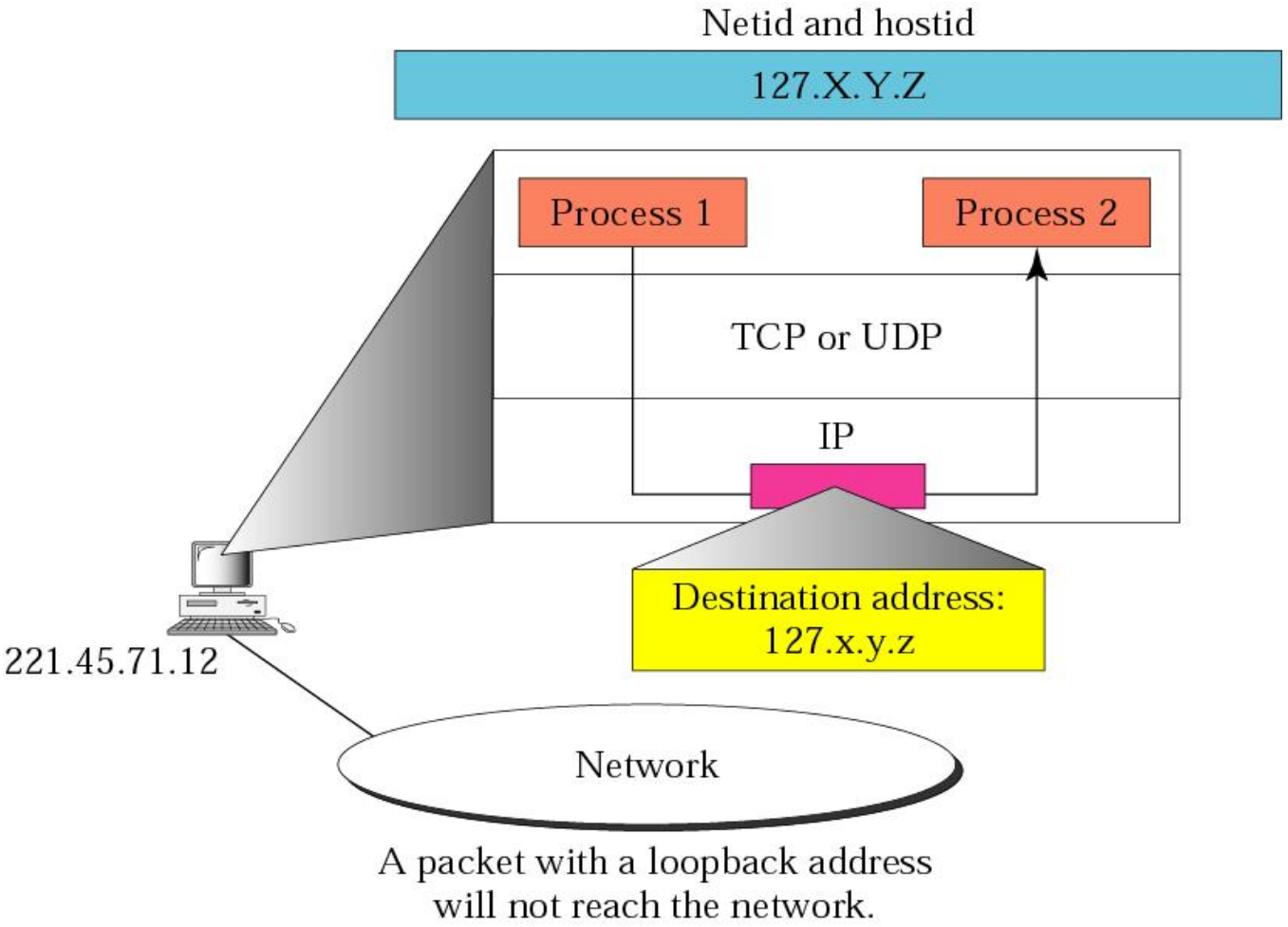
- Specific host on this network



This address is used by a router or host
to send a message to a specific host on the same network.

Special Addresses - Examples

- Loopback address



Addresses of Privat Networks

- Used for research and privat goals
- Not allowed to be visible to outside the network

Class	*Netids*	*Blocks*
A	10.0.0	1
B	172.16 to 172.31	16
C	192.168.0 to 192.168.255	256

Addressing in IPv4 World

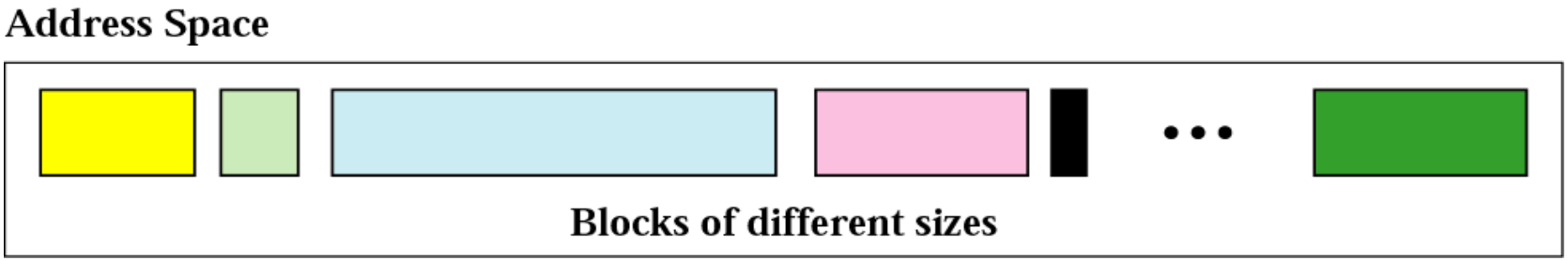
Classfull Adressing

Classless Addressing

Classless Addressing

- In classless addressing variable-length blocks are assigned that belong to no class

- In this architecture, the entire address space (232 addresses) is divided into blocks of different sizes



Classless Addressing

- 205.16.37.32 & 17.17.33.80 are beginning address of a block that contains 16 addresses
  - 32 & 80 are divisible by 16

- 190.16.42.0 & 17.17.32.0 are beginning address of a block that contains 256 addresses
  - 0 is divisible by 16

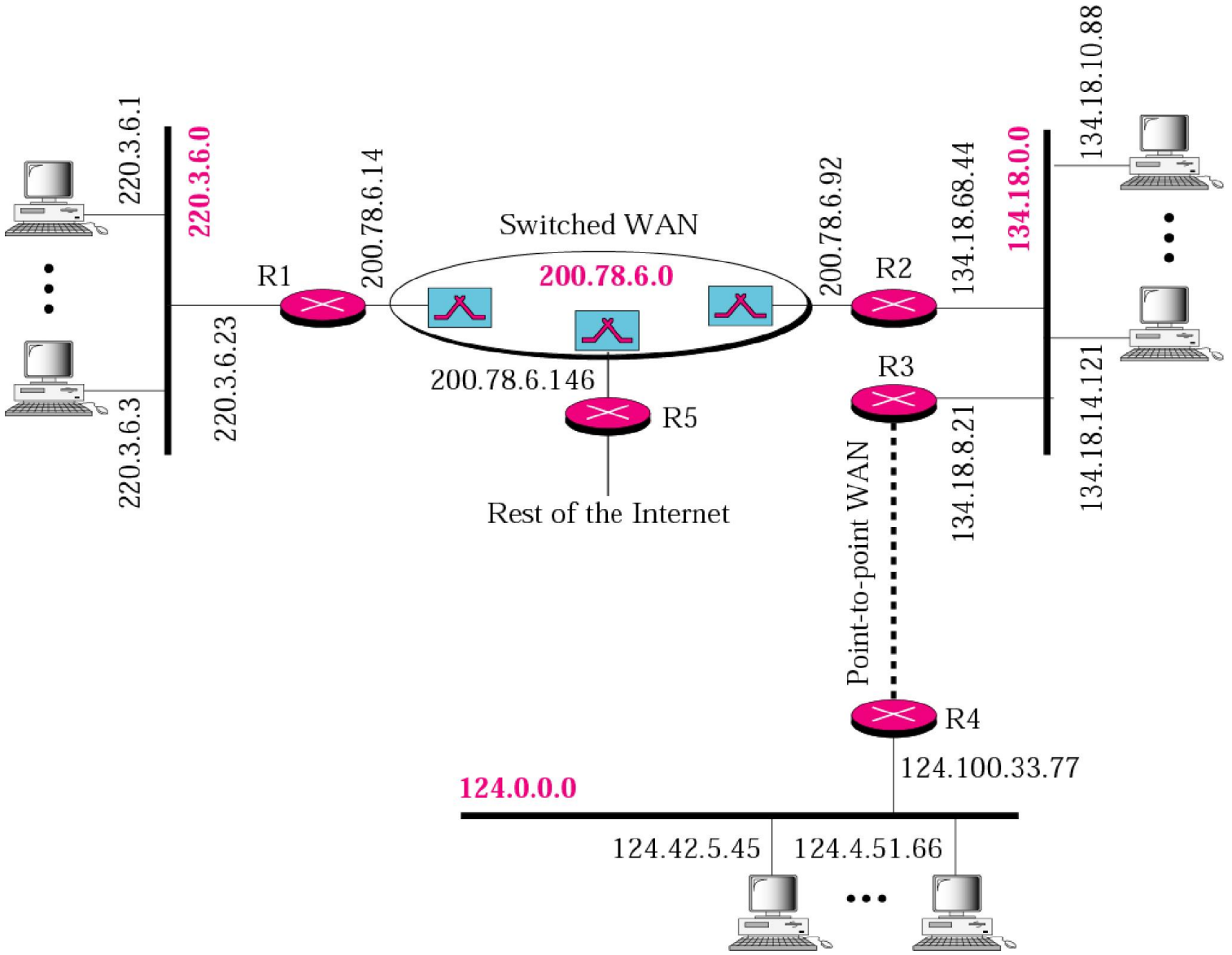
- 17.17.32.0 is a beginning address of a block that contains 1024 addresses
  - 1024 = 4 × 256. The right-most byte must be divisible by 256. The second byte (from the right) must be divisible by 4

Classless Addressing

- Prefix lengths

/n	Mask	/n	Mask	/n	Mask	/n	Mask
/1	128.0.0.0	/9	255.128.0.0	/17	255.255.128.0	/25	255.255.255.128
/2	192.0.0.0	/10	255.192.0.0	/18	255.255.192.0	/26	255.255.255.192
/3	224.0.0.0	/11	255.224.0.0	/19	255.255.224.0	/27	255.255.255.224
/4	240.0.0.0	/12	255.240.0.0	/20	255.255.240.0	/28	255.255.255.240
/5	248.0.0.0	/13	255.248.0.0	/21	255.255.248.0	/29	255.255.255.248
/6	252.0.0.0	/14	255.252.0.0	/22	255.255.252.0	/30	255.255.255.252
/7	254.0.0.0	/15	255.254.0.0	/23	255.255.254.0	/31	255.255.255.254
/8	255.0.0.0	/16	255.255.0.0	/24	255.255.255.0	/32	255.255.255.255

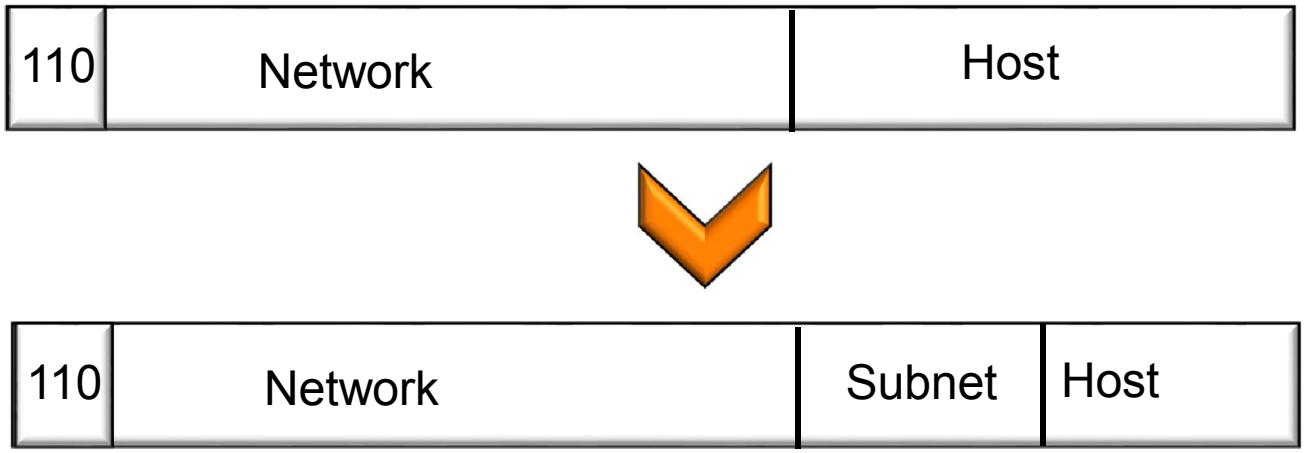
Sample Internet



Subnetting

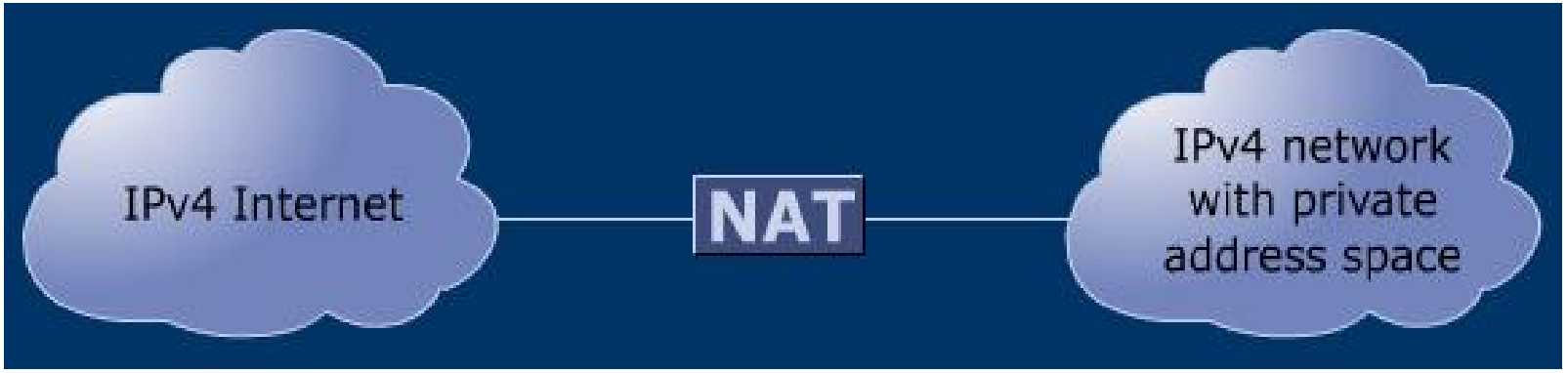
- Addressing quickly leads to a situation where the number of adresses gets insufficient
- Class C is highly preferred over class A or B. However,
  - Only 253 adresses can be assigned

- Solution → **Subnetting**

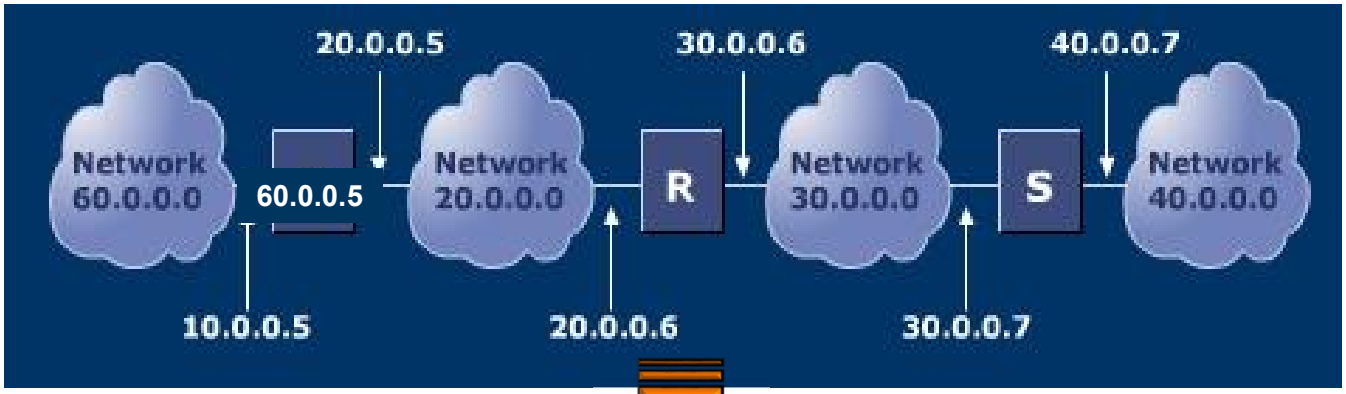


Network Address Translation (NAT)

- Even with the use of subnetting & classless addressing, the space of IPv4 is limited
- To further extend the lifetime of IPv4
  - Use Network Allocation Translation (NAT)
  - Use a public address for the whole network and manage the addresses inside the network privat

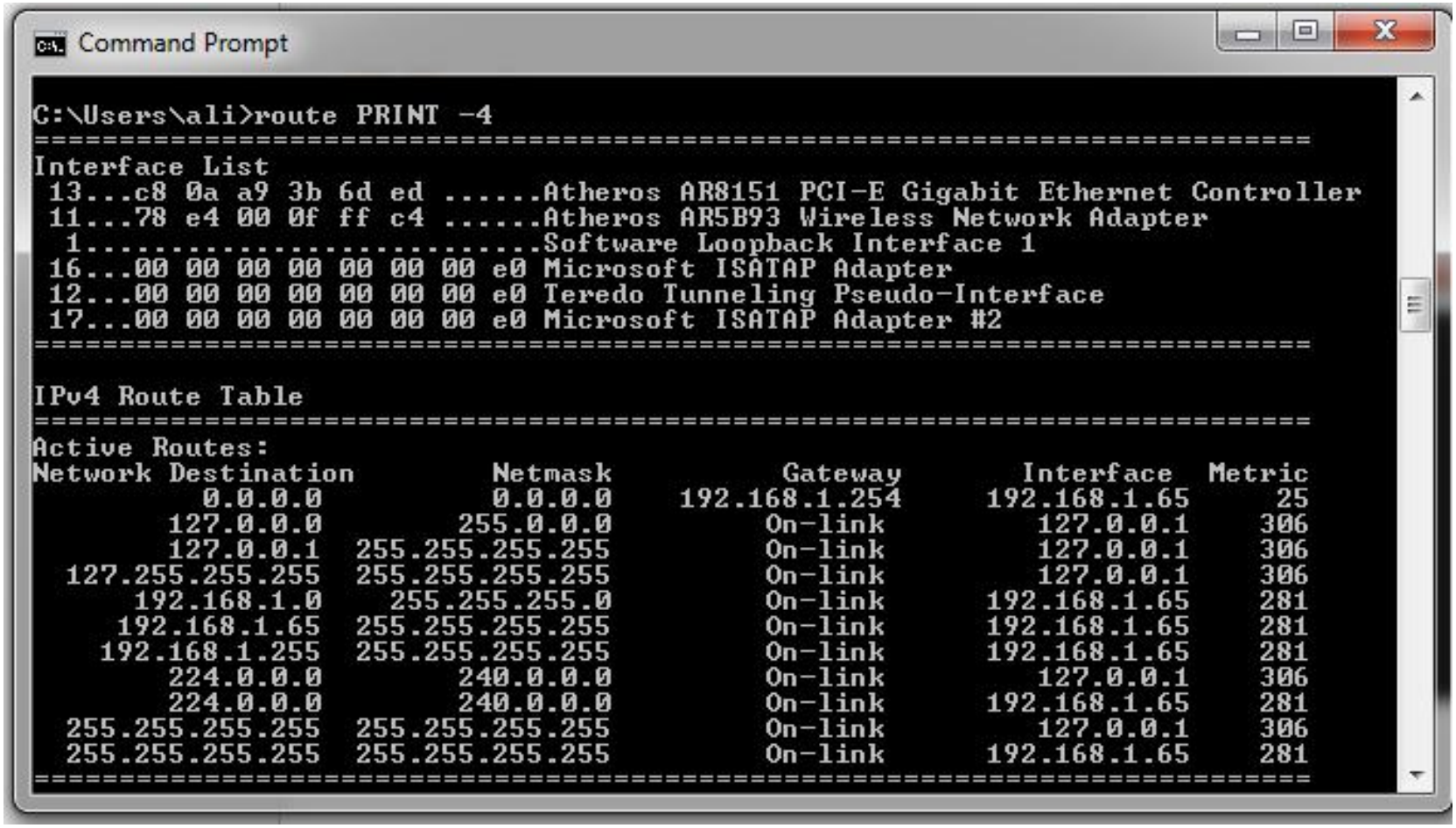


IP Routing

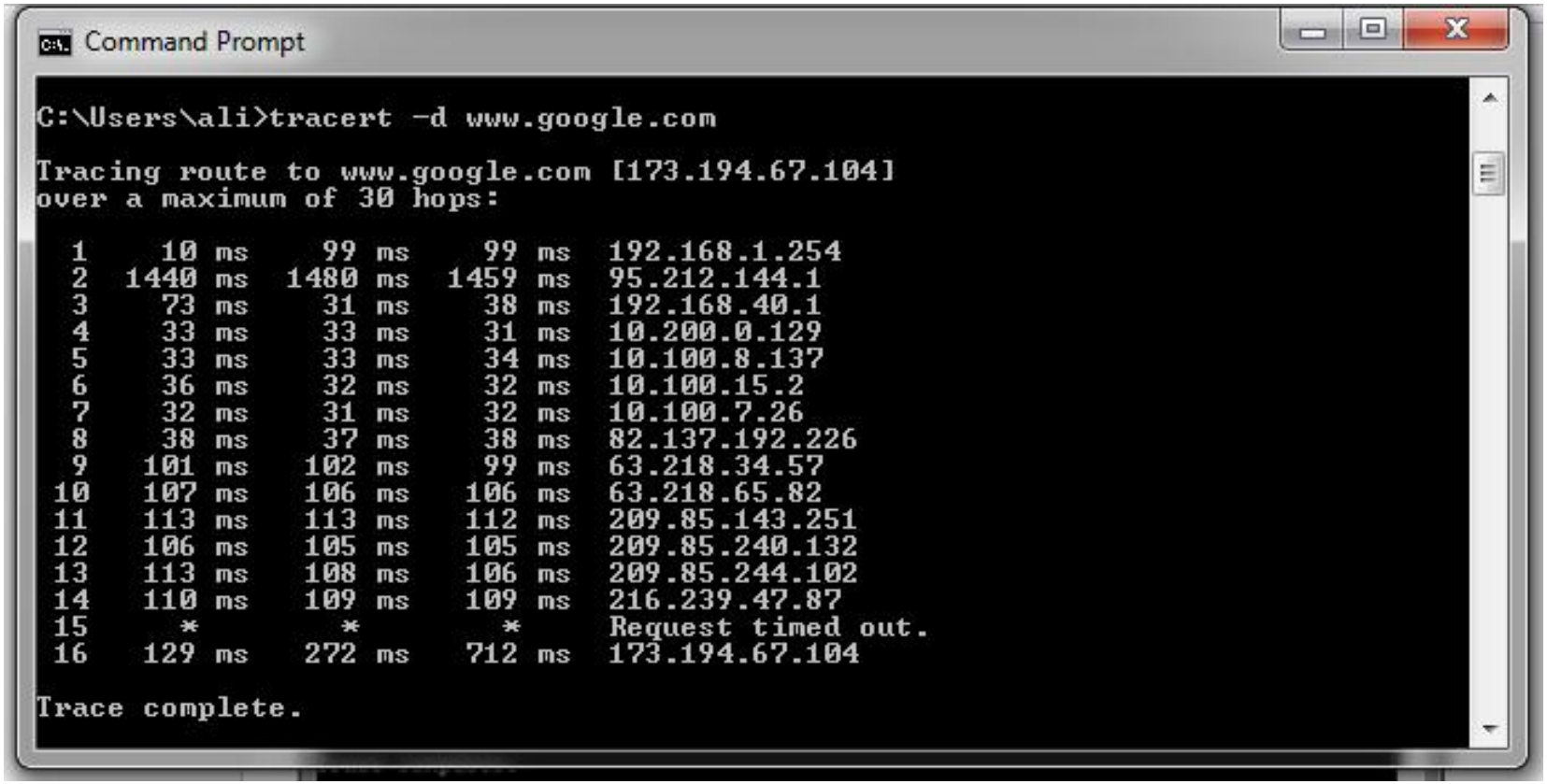


TO REACH HOSTS ON NETWORK	ROUTE TO THIS ADDRESS
20.0.0.0	DELIVER DIRECTLY
30.0.0.0	DELIVER DIRECTLY
60.0.0.0	20.0.0.5
50.0.0.0	30.0.0.7

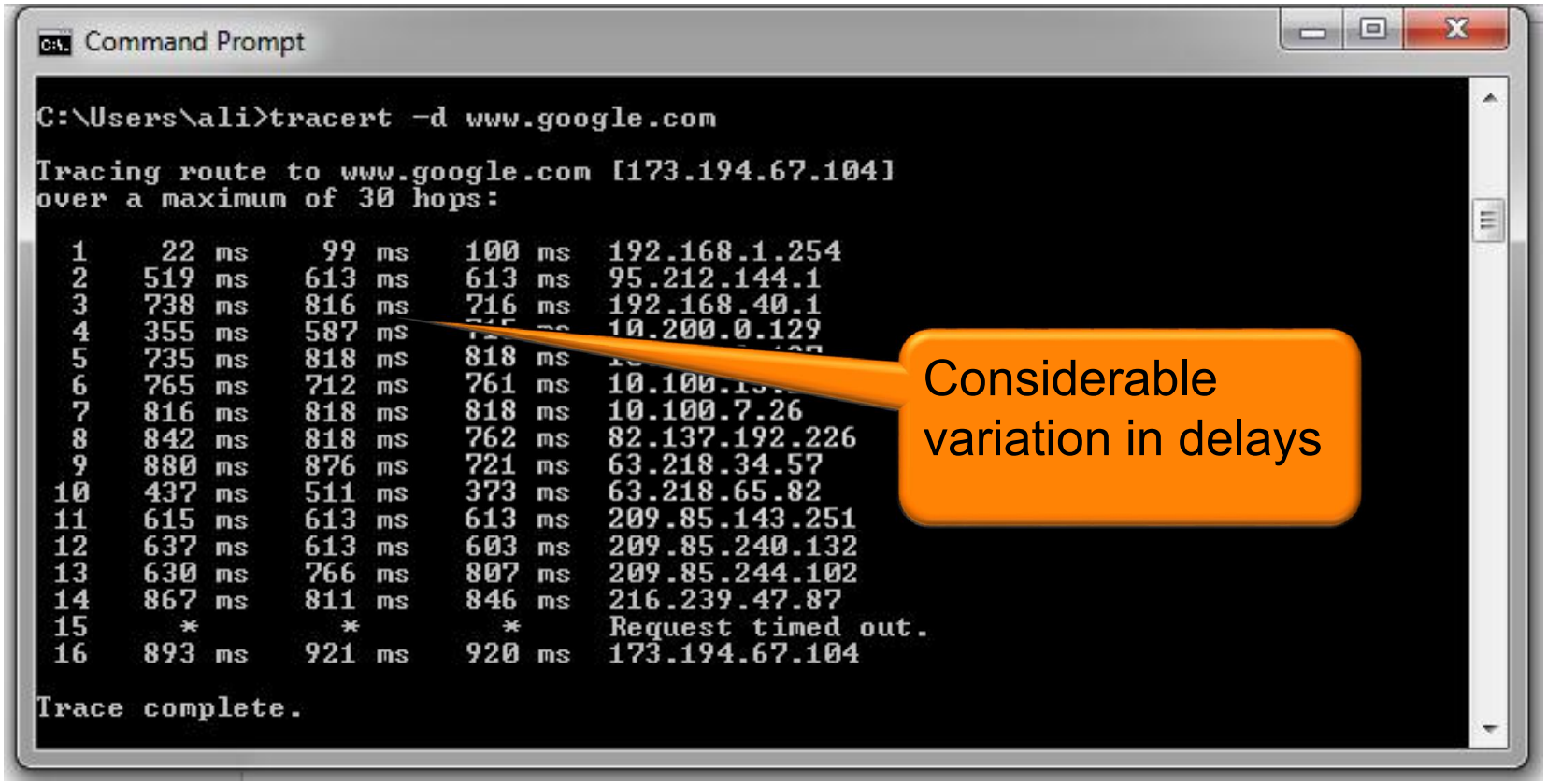
IP Routing – Some Praxis



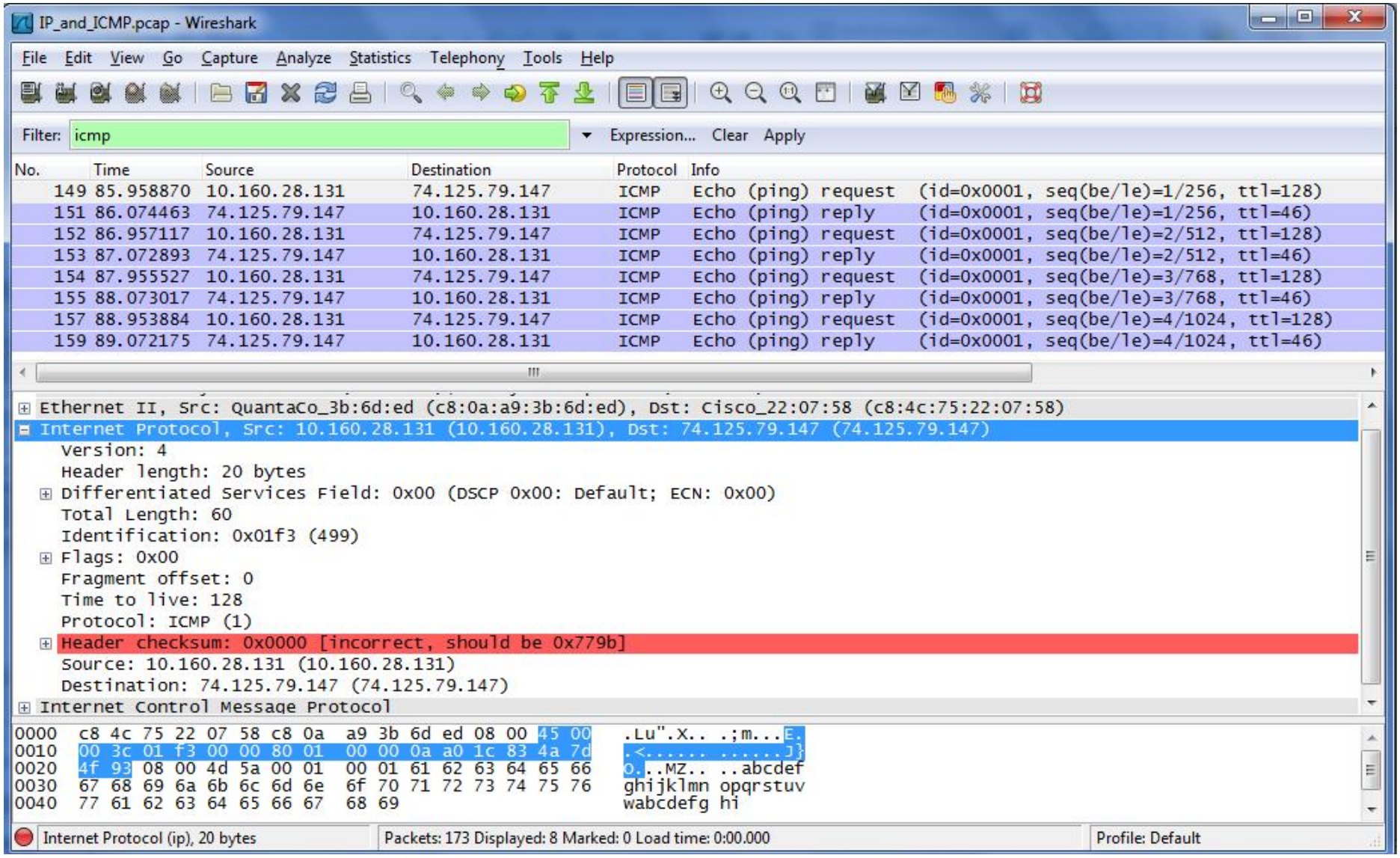
IP Routing – Some Praxis



IP Routing – Some Praxis



IP Routing – Some Praxis



IP Security

- No security considerations in IP
  - No requirements to encrypt datagramms
  - No guard against attacks such as

- Two solutions: IPSec, firwalls

Shortcommings of IPv4

- IPv4 has several shortcommings, especially in the area of
  - Quality of Service (QoS)
  - System management
  - Security
  - Scalability
  - Mobility
  - **Adressing availability**

- IPv6 has been developed to address these shortcommings

Key Features of IPv6

- Extended address space
  - Addresses length is 128 bits instead of 32 bit → addresses space is increased by a factor of 2^{96} compared to that of IPv4 → $6x10^{23}$ unique addresses per square meter of the earth‘s surface

- Increases addressing flexibility
  - The large addresses space of IPv6 allows for multiple levels of subnetting and address allocation from the Internet backbone. Flexibility is further enhanced using
    - Introducing a „Scope“ field to the multicast addresses → improving the scalability of multicast routing
      - There exist 16 types for „Scope“ field
    - Introducing a new type of addresses for anycast

Key Features of IPv6

Number of senders	Number of receivers	Communication association	Example
1	1	Unicast	Lecture
1	1	Dialogue	Telephony
1	< all	Multicast	Pay TV
1	all	Broadcast	Normal TV
>1	1	Concast	Tele-voting
>1	>1	Multipeer	Vidio-conference
1	>1	Anycast	Automatic configuration of network devices (DHCP)

A new type of addresses is introduced in IPv6 for this type of association

Key Features of IPv6

- Reduced routing tables
  - IPv6 addresses length is big enough to facilitate address hierarchies → reduced size of routing tables in routers

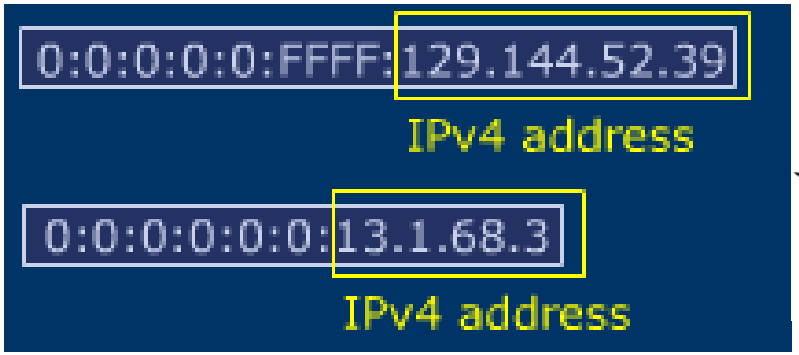
- Header format simplification
  - The header format is desighned to keep header overhead at minimum
    - Not essential fields and option fields are moved to extension header (placed after the formal header)
    - Checksum field is omitted because the data-link and transport layers have their own checksumes, which means that omitting the checksum field will not negatively affect the performance. In fact, it will save the time required to calculate it.

- Statelful (with DHCP server) and stateless (without DHCP server) addrress auto-configuration

Key Features of IPv6

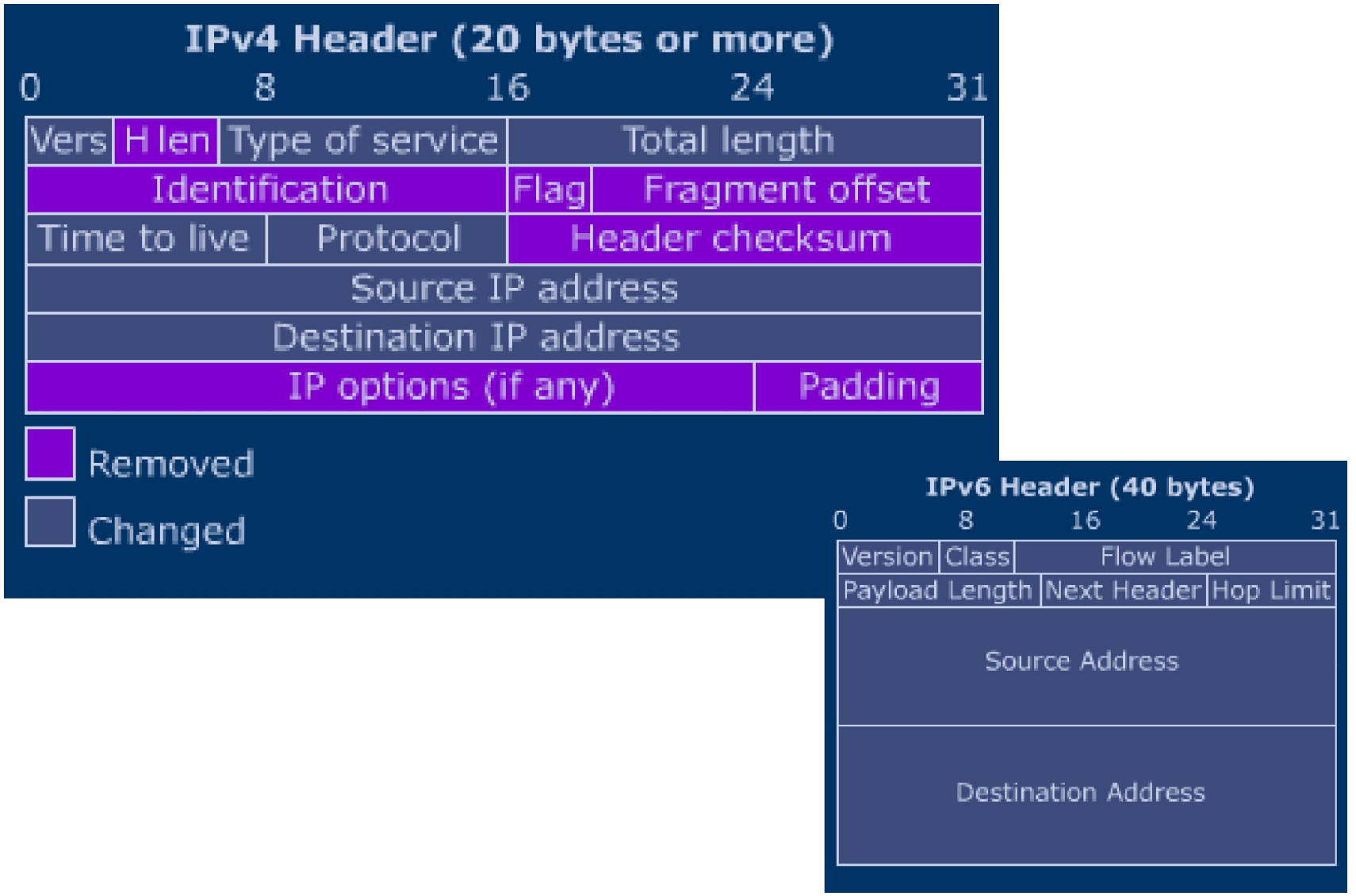
- Security capabilities
  - IPSec is integrated with IPv6
- QoS enhancements
  - Flow lable field allows to provide a specific processing for a given flow (series of packets between a source and a destination) → flow identification
- Improved options mechanisms
  - The option headers (located between the formal IPv6 header and the transport layer protocol header) allow for several tasks to be easily achieved
    - Hop-by-hop extension header allows that a packet receives a special processing from a special router/host on its path towards its destination
    - Further extension headers are possible

IPv6 Addressing Overview



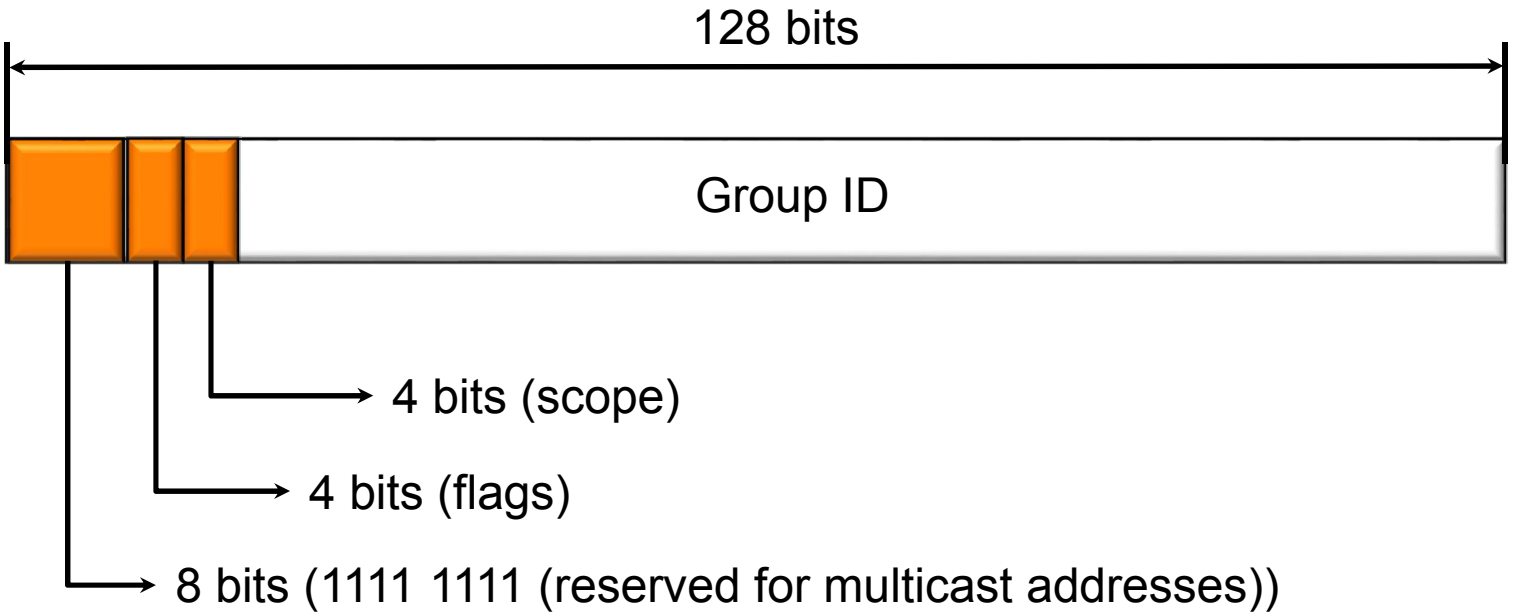
IPv4 addresses can be embedded within IPv6 addresses. This is usefull for interconnecting between IPv4 and IPv6 addresses

IPv6 Header Format



IPv6 Multicast Addresses

- IPv6 multicast communication is similar to IPv4 multicast communication
- IPv6 devices can join and listen for multicast traffic on an IPv6 multicast address
  - An IPv6 multicast address identifies multiple interfaces



IPv6 Multicast Addresses

- The first eight bits are reserved as 1111 1111
  - The prefix of an IPv6 multicast address is ff00::/8
    - It is easy to identify an IPv6 multicast address
- The next four bits are the flags
  - Only 3 of the 4 flags are in use currently (the most significant bit in the 4 bits flags is reserved for future use)
- The next four bits are the Scope bits
  - Scope bits (4 bits) are used to indicate the scope of delivery of IPv6 multicast traffic

IPv6 Multicast Addresses

4 Bits inside flags field	Flag name	When "0" set	When "1" set
0 (Most Significant Bit)	Currently not in use	Currently not in use	Currently not in use
1	R (Rendezvous)	When R flag set to 0, the multicast rendezvous point is not embedded with multicast address	When R flag set to 1, the multicast rendezvous point is embedded with multicast address
2	P (Prefix)	When P flag set to 0, the multicast address is not based on network prefix	When P flag set to 1, the multicast address based on network prefix
3 (Least Significant Bit)	T (Transient)	When T flag set to 0, the multicast address is a permanently assigned (well-known) multicast IPv6 address	When T flag set to 1, the multicast address is a transient (Dynamically assigned) multicast address

IPv6 Multicast Addresses

Value	Scope	Meaning
0	Reserved	Currently not in use
1	Interface-local scope	The Interface-local scope is limited for a local single interface only. Useful only for loopback delivery of multicasts within a node.
2	Link-local scope	Link-local scope is defined for the local link. The traffic with the multicast address of FF02::2 is limited to local link scope. An IPv6 router will never forward the multicast traffic destined to FF02::2 beyond the local link.
3	Subnet-local scope	Subnet-local scope ranges subnets on multiple links.
4	Admin-local scope	Administratively configured scope. Definition states that "the smallest scope that must be administratively configured".

IPv6 Multicast Addresses

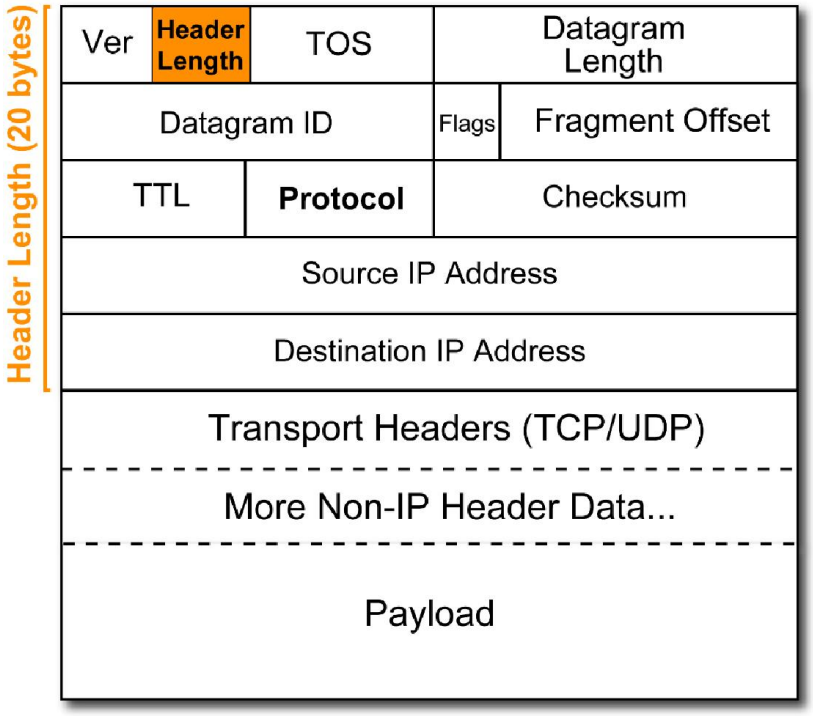
Value	Scope	Meaning
5	Site-local scope	The scope of Site-local multicast addresses are within the local physical network. For example, a small branch office. Site-local Multicast packets will not cross the IPv6 border router at the site.
8	Organization-local scope	The scope of Organization-local addresses are within different sites inside an organization. Organization-Local Multicast packets will not cross the organization's IPv6 border router to reach the IPv6 internet
E	Global scope	Scope is the Global IPv6 internet
F	Reserved	Currently not in use

IPv6 Multicast Addresses

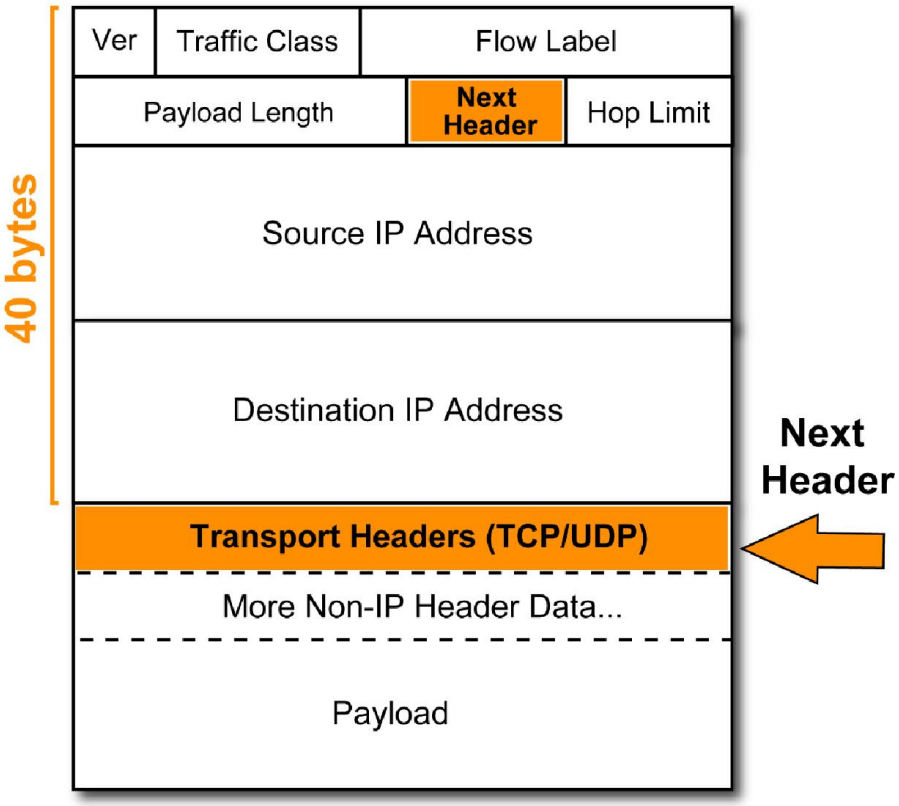
Important IPv6 Link-local scope Multicast Addresses	Description
FF02::1	All Nodes IPv6 multicast address. All IPv6 nodes in same link will listen to FF02::1 IPv6 multicast address.
FF02::2	All Routers IPv6 multicast address. All IPv6 routers on the same link will listen to FF02::2 IPv6 multicast address.
FF02::4	The all-Distance Vector Multicast Routing Protocol (DVMRP) routers address. Used to reach all DVMRP multicast routers on the same link.
FF02::5	OSPF IPv6 multicast address. (Remember, in IPv4, OSPF multicast address was 224.0.0.5)
FF02::6	OSPFv3 Designated Router's and Non-Designated Router's IPv6 multicast address. (Remember, in IPv4, OSPF Designated Router's (DR) and Non-Designated Router's (NDR) multicast address was 224.0.0.6). Only DR and BDR listen to FF02::6 multicast IPv6 address.
FF02::9	RIPng IPv6 multicast address. (Remember, in IPv4, RIPv2 multicast address was 224.0.0.9)
FF02::A	EIGRP IPv6 multicast address. (Remember in IPv4, EIGRP multicast address was 224.0.0.10. A is the hexadecimal equivalant of decimal 10)
FF02::D	All PIM Routers
FF02::16	All MLDv2-capable routers
FF02::1:2	DHCPv6 Servers and DHCPv6 Relay Agents
FF02::1:FF00:0000/104	Solicited-Node IPv6 Multicast address

IPv6 Extension Headers

IPv4 Packet (No Options)

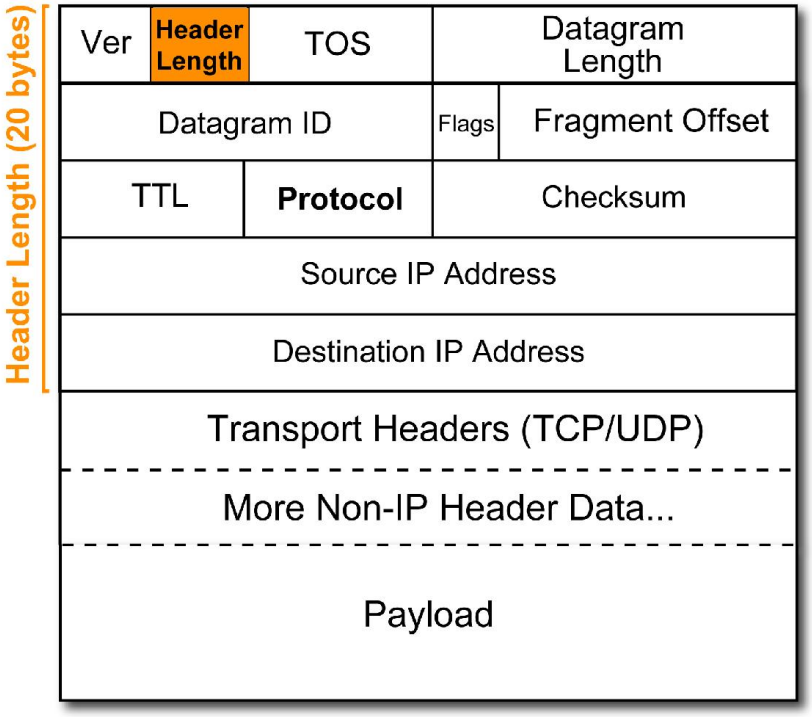


IPv6 Packet (No Extensions)

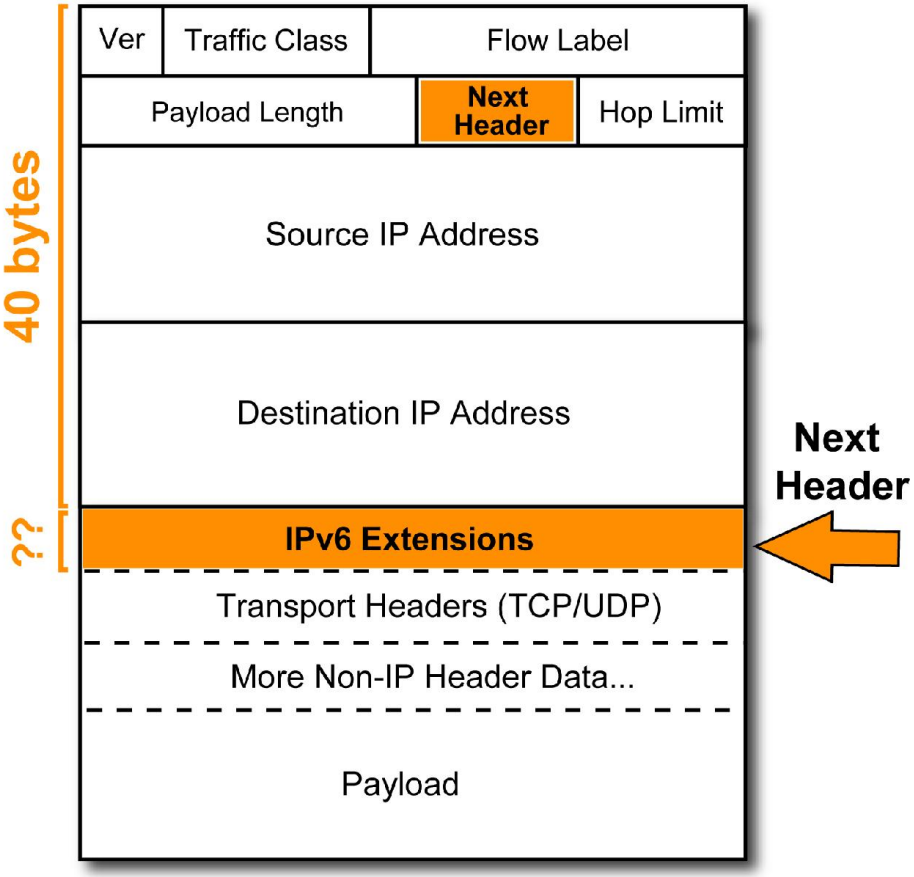


IPv6 Extension Headers

IPv4 Packet (no Options)



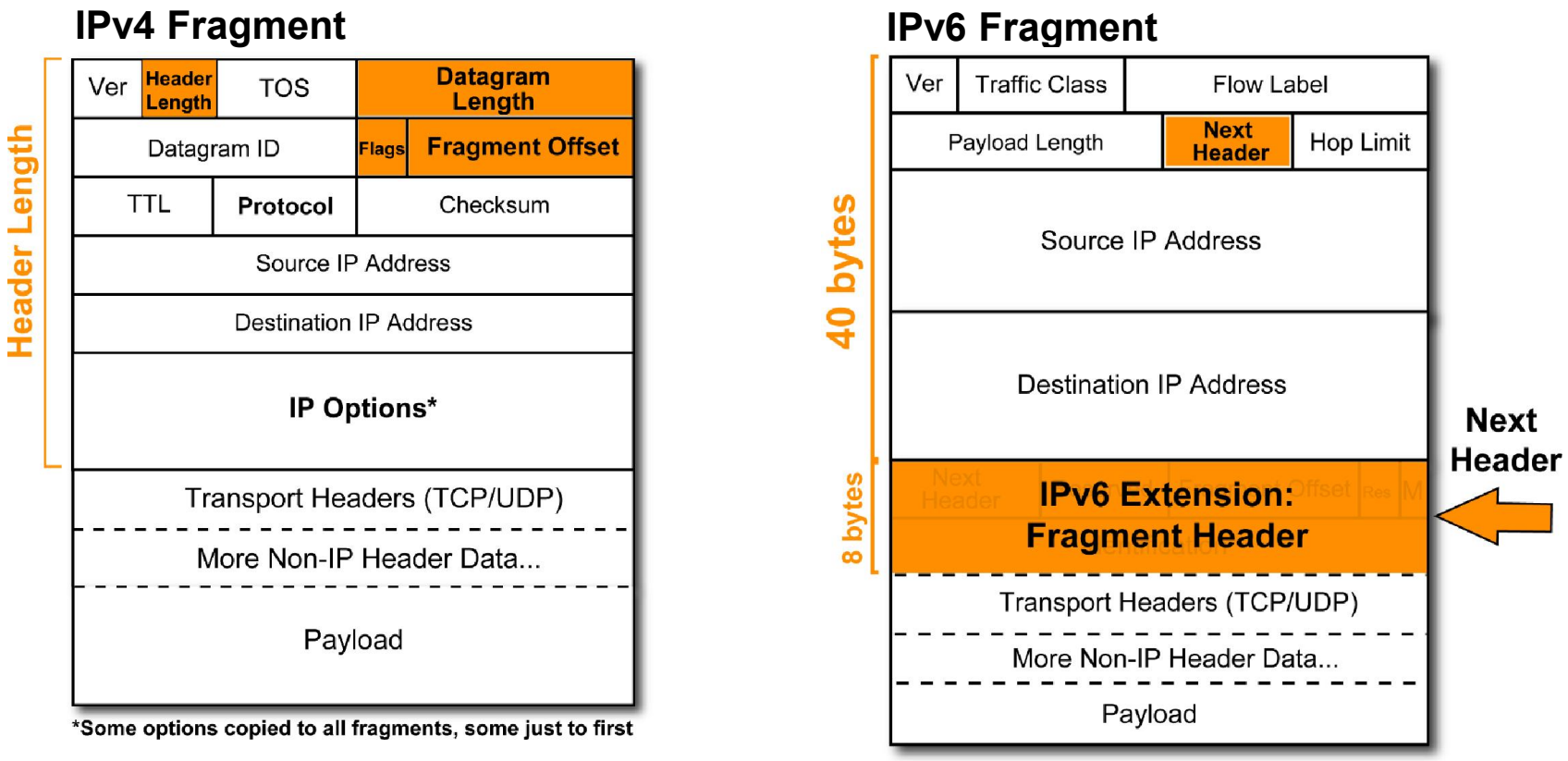
IPv6 Packet (with Extensions)



Values of Next Header Field

Value (in decimal)	Header
0	Hop-by-Hop Options Header
6	TCP
17	UDP
41	Encapsulated IPv6 Header
43	Routing Header
44	Fragment Header
46	Resource ReSerVation Protocol
50	Encapsulating Security Payload
51	Authentication Header
58	ICMPv6
59	No next header
60	Destination Options Header

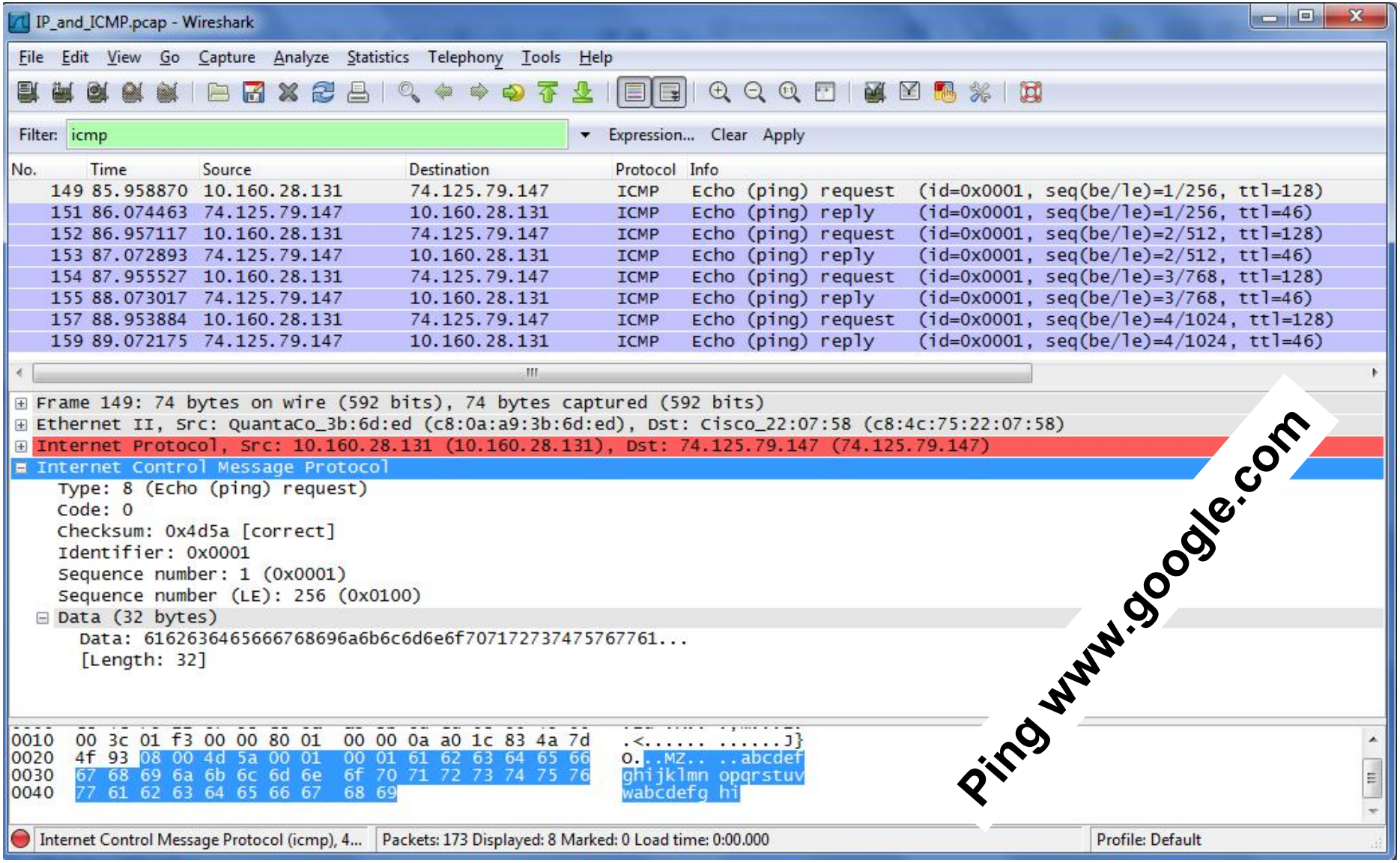
IPv4 vs. IPv6 Fragmentation



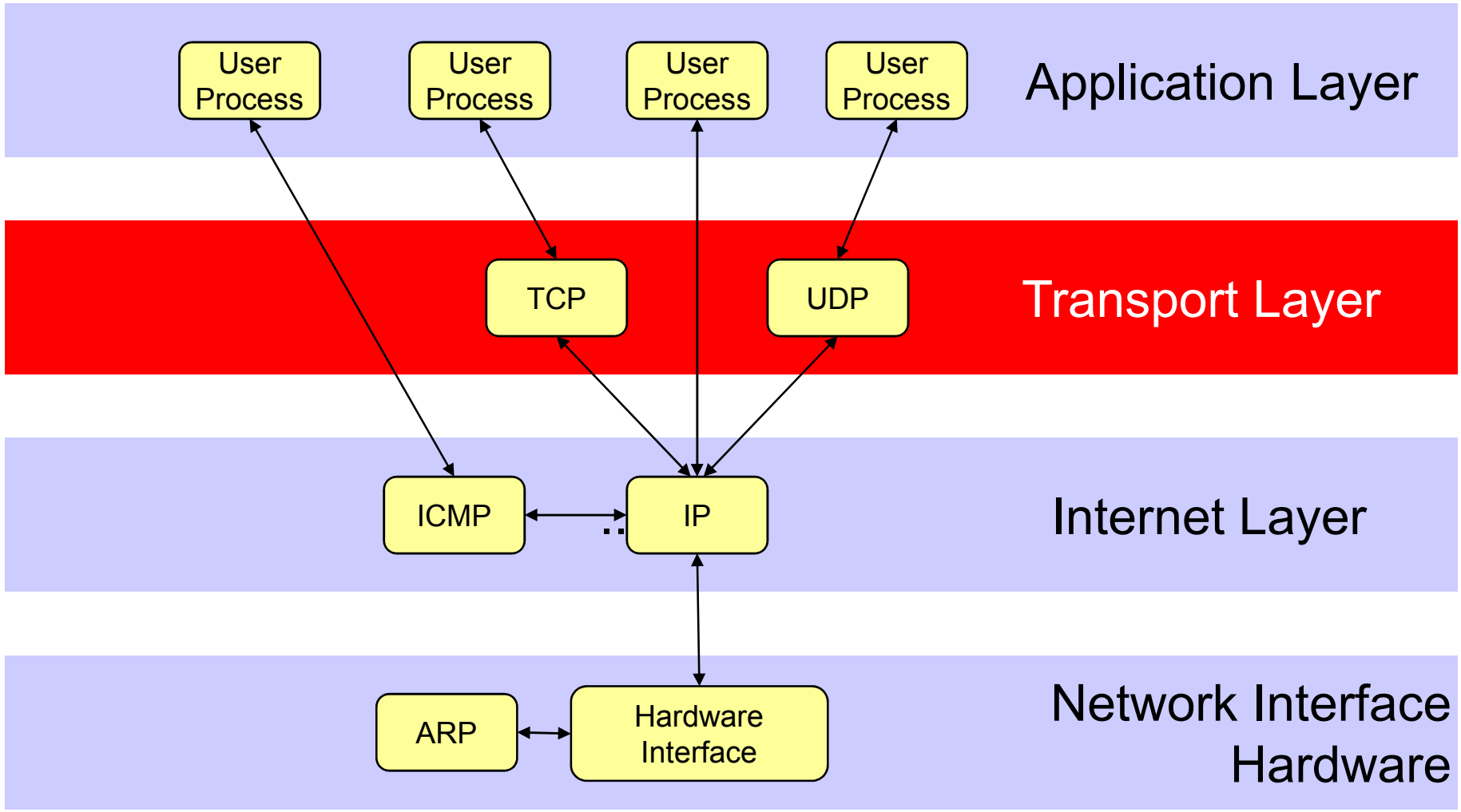
Recommended Order of IPv6 Extension Headers

1. IPv6 header (40 bytes)
2. Hop-by-hop options header (variable)
3. Destination options header (1) (variable)
4. Routing header (variable)
5. Fragment header (8 bytes)
6. Authentication header (variable)
7. Encapsulation Security Payload header (variable)
8. Destination options header (2) (variable)
9. Upper-layer header (for example, TCP or UDP)

Example

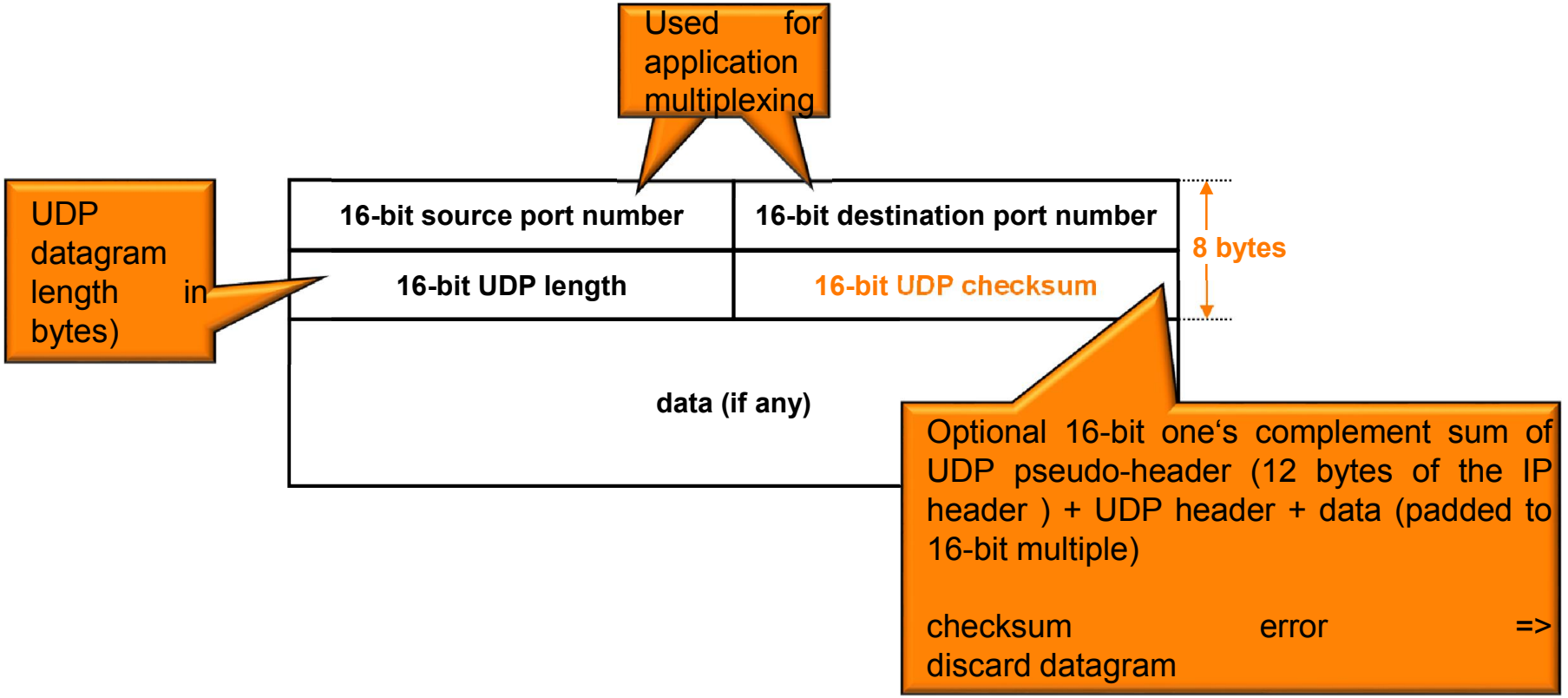

IP_and_ICMP.pcap - Wireshark
File Edit View Go Capture Analyze Statistics Telephony Tools Help
Filter: icmp
Expression... Clear Apply
No. Time Source Destination Protocol Info
149 85.958870 10.160.28.131 74.125.79.147 ICMP Echo (ping) request (id=0x0001, seq(be/le)=1/256, ttl=128)
151 86.074463 74.125.79.147 10.160.28.131 ICMP Echo (ping) reply (id=0x0001, seq(be/le)=1/256, ttl=46)
152 86.957117 10.160.28.131 74.125.79.147 ICMP Echo (ping) request (id=0x0001, seq(be/le)=2/512, ttl=128)
153 87.072893 74.125.79.147 10.160.28.131 ICMP Echo (ping) reply (id=0x0001, seq(be/le)=2/512, ttl=46)
154 87.955527 10.160.28.131 74.125.79.147 ICMP Echo (ping) request (id=0x0001, seq(be/le)=3/768, ttl=128)
155 88.073017 74.125.79.147 10.160.28.131 ICMP Echo (ping) reply (id=0x0001, seq(be/le)=3/768, ttl=46)
157 88.953884 10.160.28.131 74.125.79.147 ICMP Echo (ping) request (id=0x0001, seq(be/le)=4/1024, ttl=128)
159 89.072175 74.125.79.147 10.160.28.131 ICMP Echo (ping) reply (id=0x0001, seq(be/le)=4/1024, ttl=46)
Frame 149: 74 bytes on wire (592 bits), 74 bytes captured (592 bits)
Ethernet II, Src: QuantaCo_3b:6d:ed (c8:0a:a9:3b:6d:ed), Dst: Cisco_22:07:58 (c8:4c:75:22:07:58)
Internet Protocol, Src: 10.160.28.131 (10.160.28.131), Dst: 74.125.79.147 (74.125.79.147)
Internet Control Message Protocol
Type: 8 (Echo (ping) request)
Code: 0
Checksum: 0x4d5a [correct]
Identifier: 0x0001
Sequence number: 1 (0x0001)
Sequence number (LE): 256 (0x0100)
Data (32 bytes)
Data: 6162636465666768696a6b6c6d6e6f707172737475767761...
[Length: 32]
0010 00 3c 01 f3 00 00 80 01 00 00 0a a0 1c 83 4a 7d .<..... ......J}
0020 4f 93 08 00 4d 5a 00 01 00 01 61 62 63 64 65 66 O...MZ.. ..abcdef
0030 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76 ghijklmn opqrstuv
0040 77 61 62 63 64 65 66 67 68 69 wabcdefg hi
Internet Control Message Protocol (icmp), 4...
Packets: 173 Displayed: 8 Marked: 0 Load time: 0:00.000
Profile: Default
Ping www.google.com

Transport Layer



UDP

- **UDP:** User Datagram Protocol
  - Specified in RFC 768
  - Simple
  - Unreliable
  - Datagram-oriented transport of application data blocks



TCP

- **TCP:** Transmission Control Protocol
- Specified in RFC 793 + others
- Connection-oriented
- Reliable byte stream service
- Contains mechanisms for
  - flow control and
  - congestion avoidence

Sockets

- **Socket**: IP address + port number

- **Port numbers** are used for application multiplexing
  - Some applications have well-known ports
    - Port 21 for FTP (RFC 1340)
    - Port 8080 for apache
    - Port 20 for SSH

- **Socket API**: popular API for TCP assosiations and UDP connections
  - **Stream sockets** use TCP
  - **Datagram sockets** use UDP

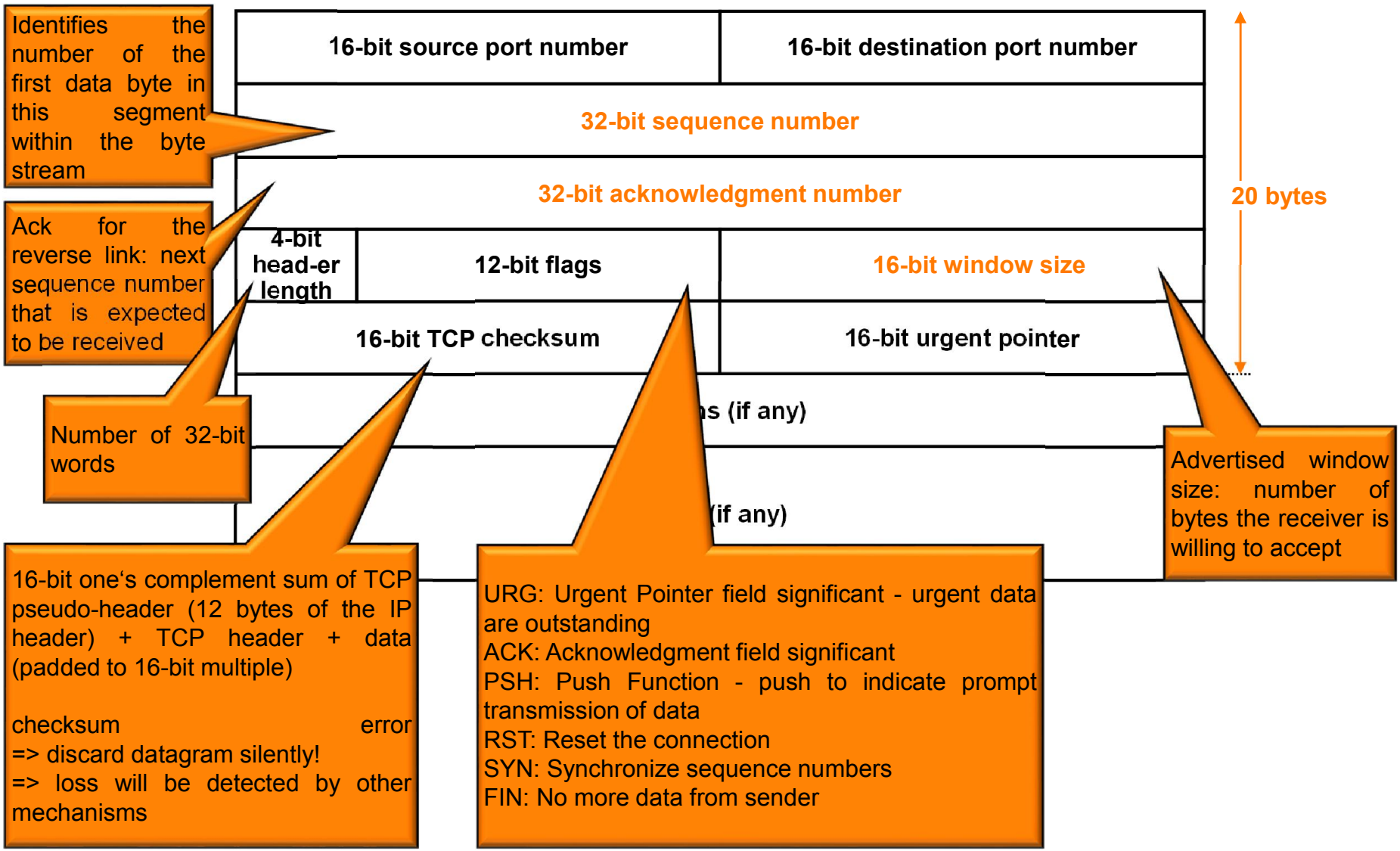
Transmission Control Protocol (TCP)

- Connection-oriented reliable byte-stream service
  - Reliability by ARQ (Automatic Repeat reQuest):
    - TCP receiver acknowledges the receipt of data segments
      - The receiver sends acknowledgements (acks) back to TCP sender to confirm the receipt of data segments
    - The receiver sends cumulative, positive acks for all contiguously received data segments
    - Timeout-based retransmission of segments

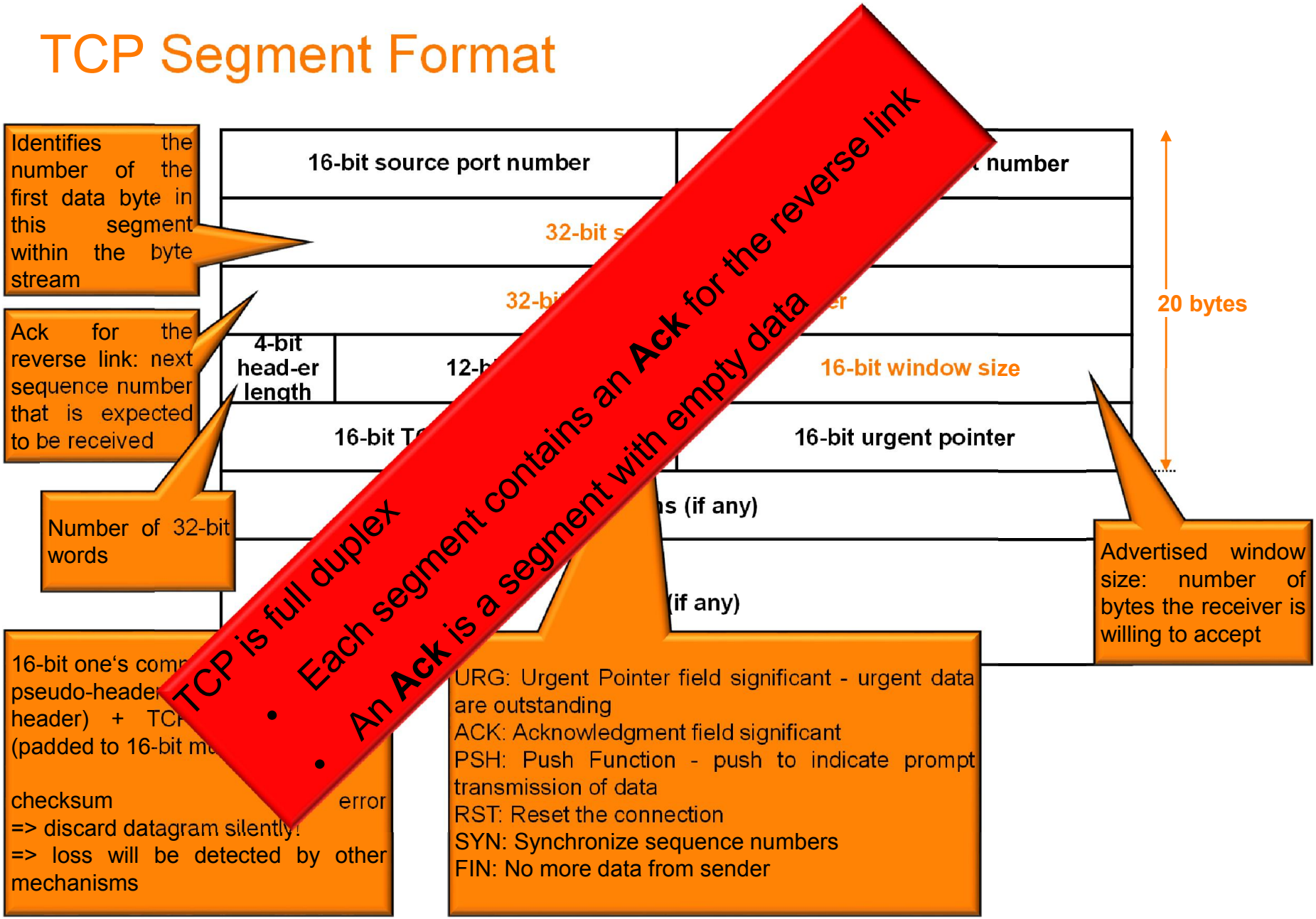
- TCP transfers a byte stream
  - Segmentation into TCP segments
  - Header contains byte sequence numbers

- **Congestion avoidance + flow control mechanism**

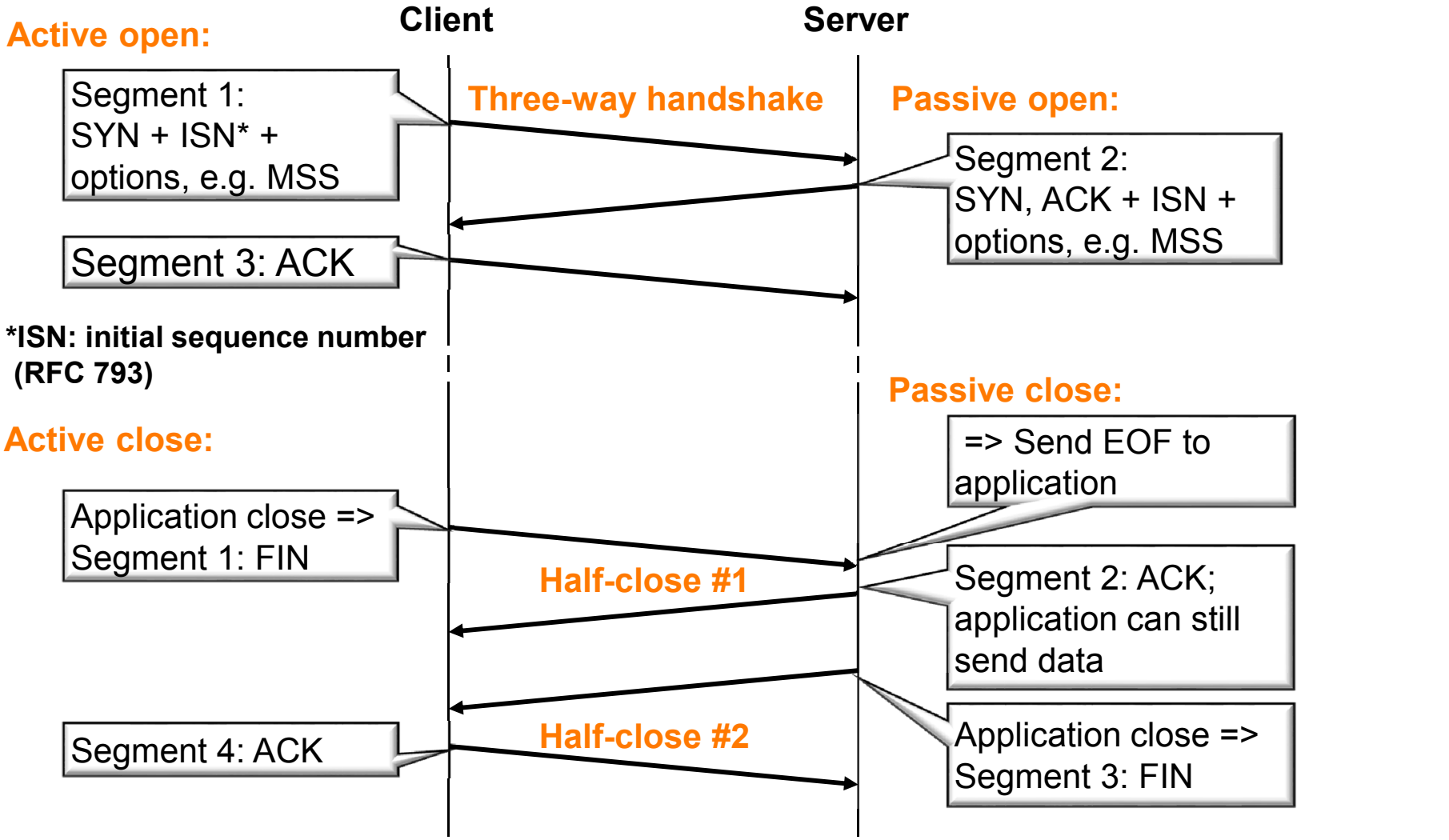
TCP Segment Format



TCP Segment Format

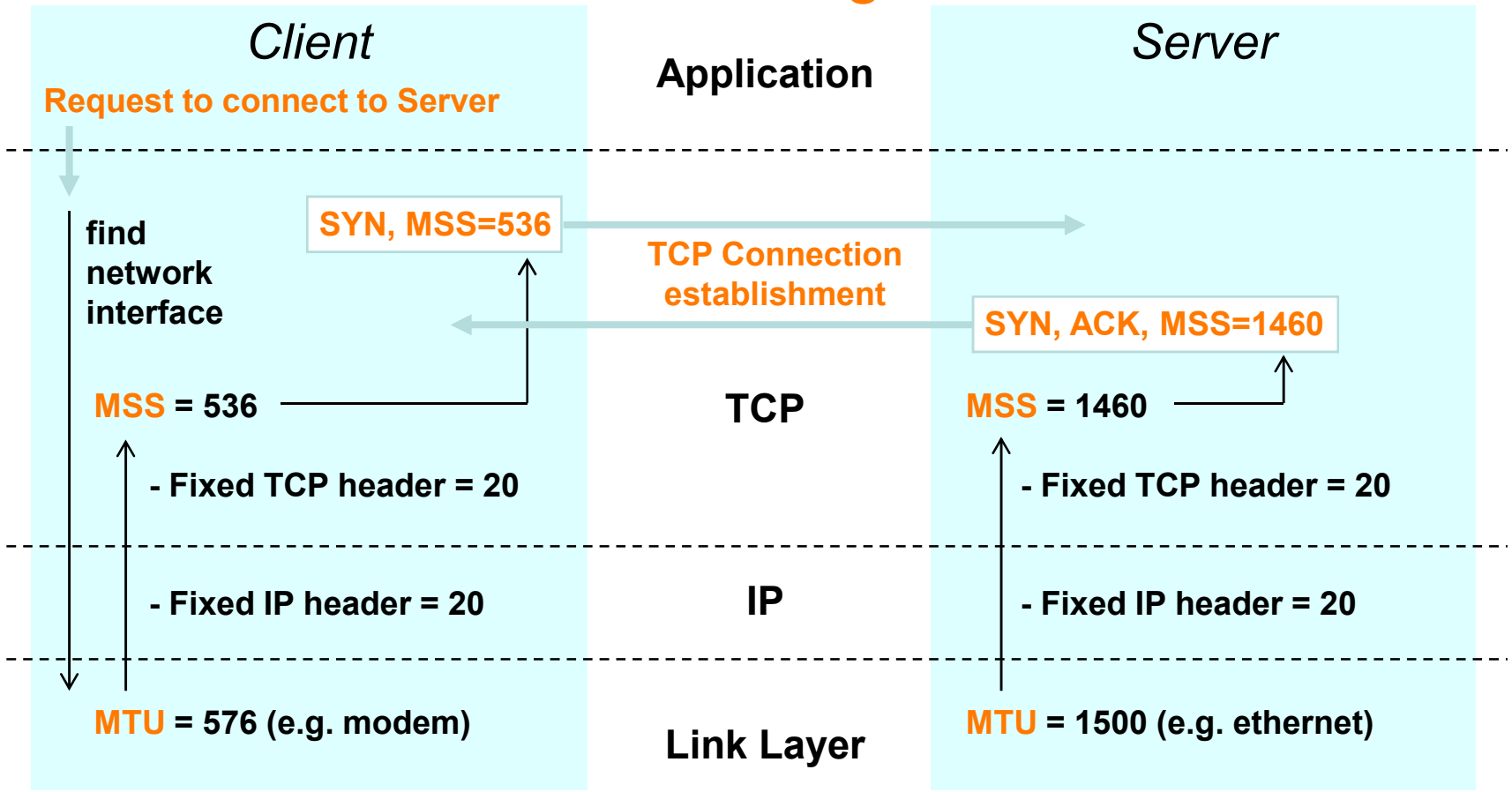


TCP Connection Establishment and Termination



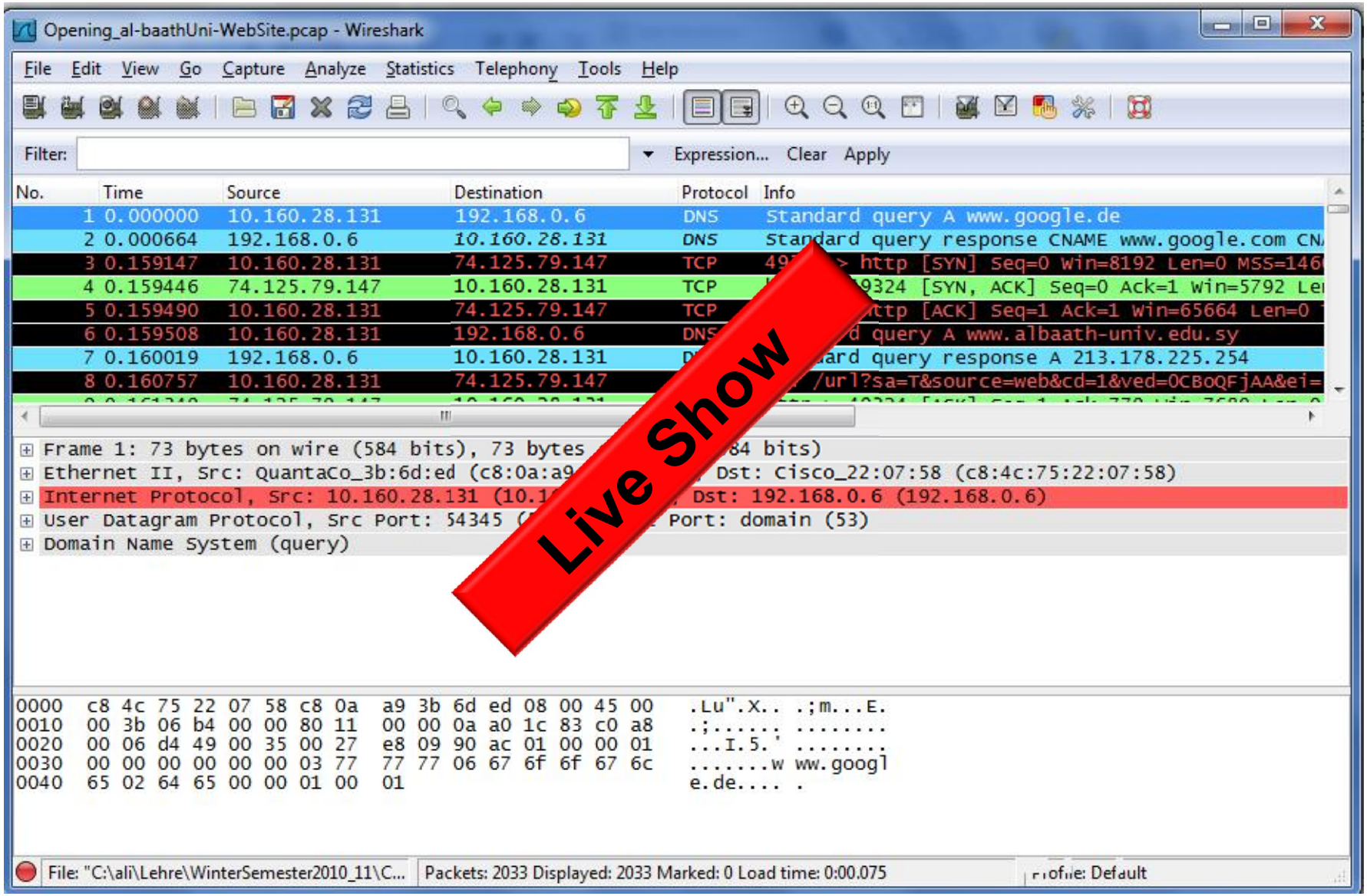
=> Connection establishment and termination take at least 1 RTT

MTU and MSS: Maximum Segment Size



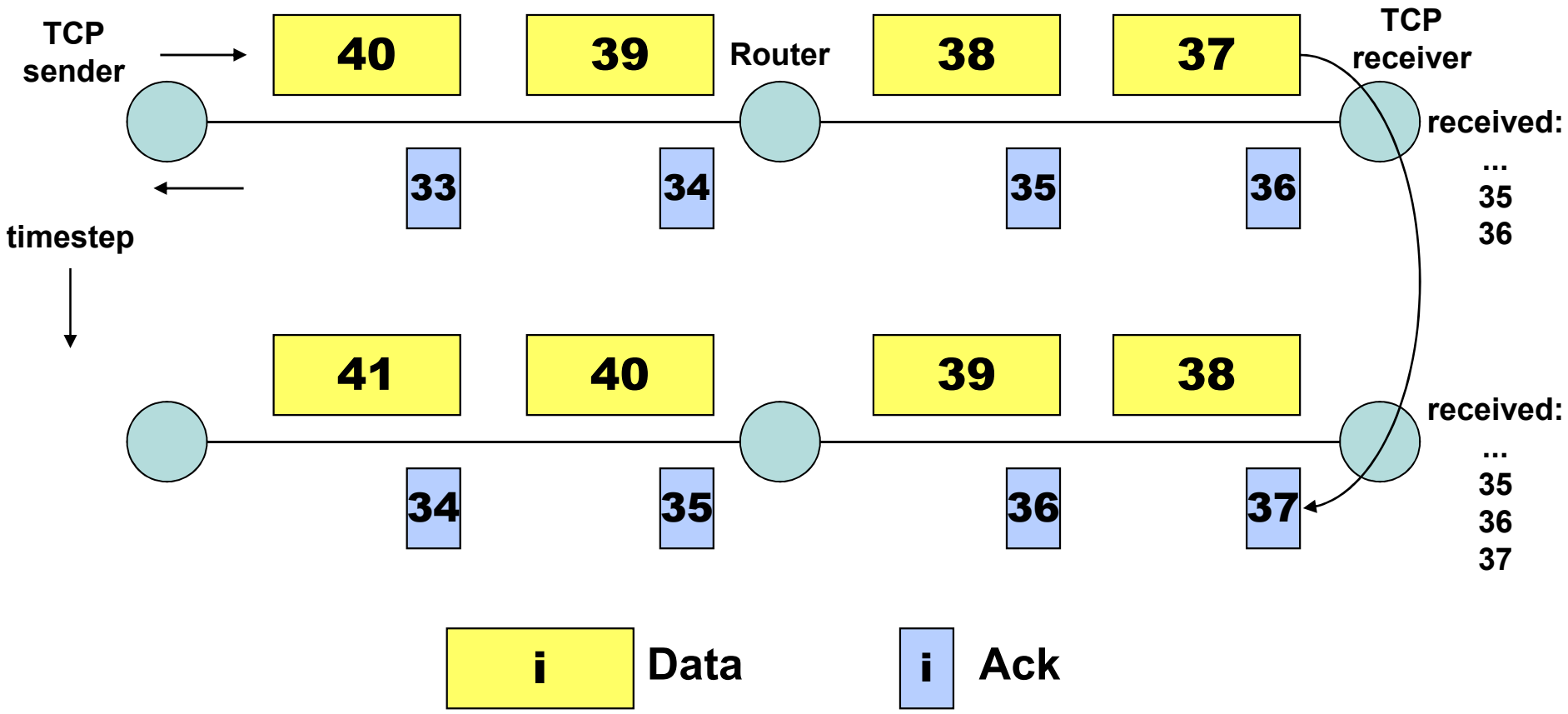
- MSS is optionally announced (not negotiated) by each host at TCP connection establishment (the smaller value is used by both ends, i.e. 536 in the above example).

Example (Access to the Internet)



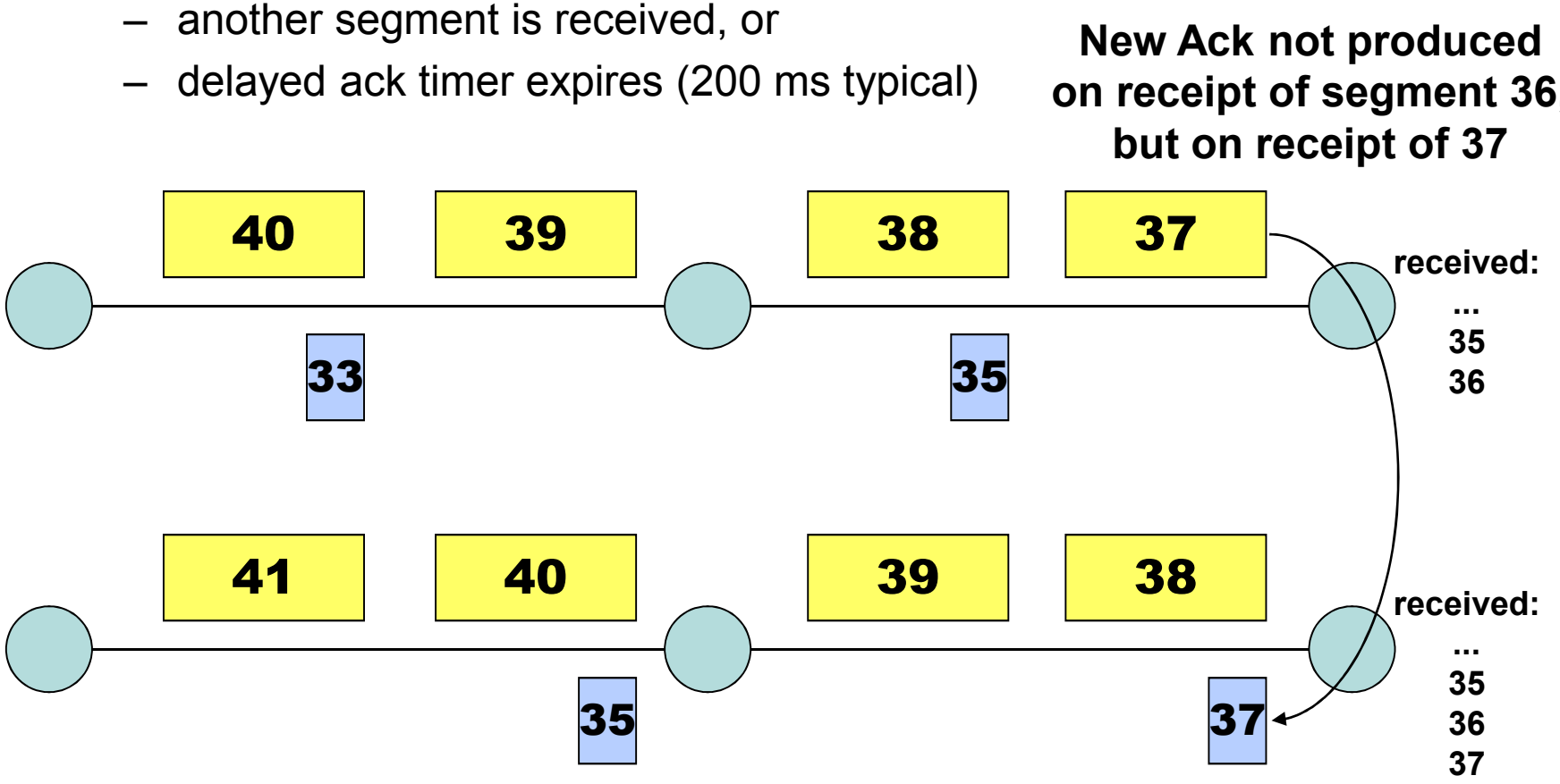
Cumulative Acknowledgements

- A new cumulative Ack is generated only on receipt of a **new in-sequence** segment



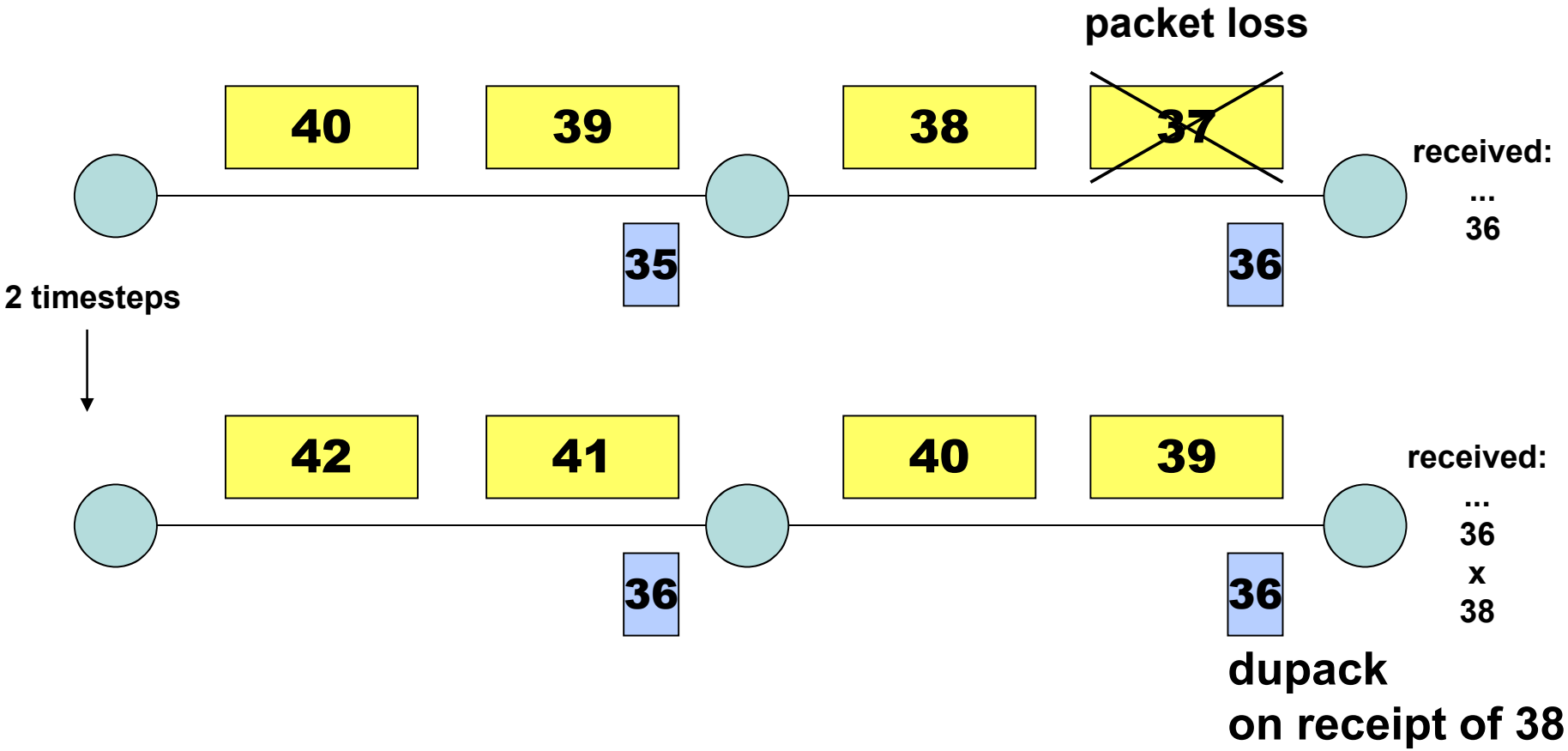
Delayed Acknowledgements

- Delaying Acks reduces Ack traffic
- An Ack is delayed until
  - another segment is received, or
  - delayed ack timer expires (200 ms typical)



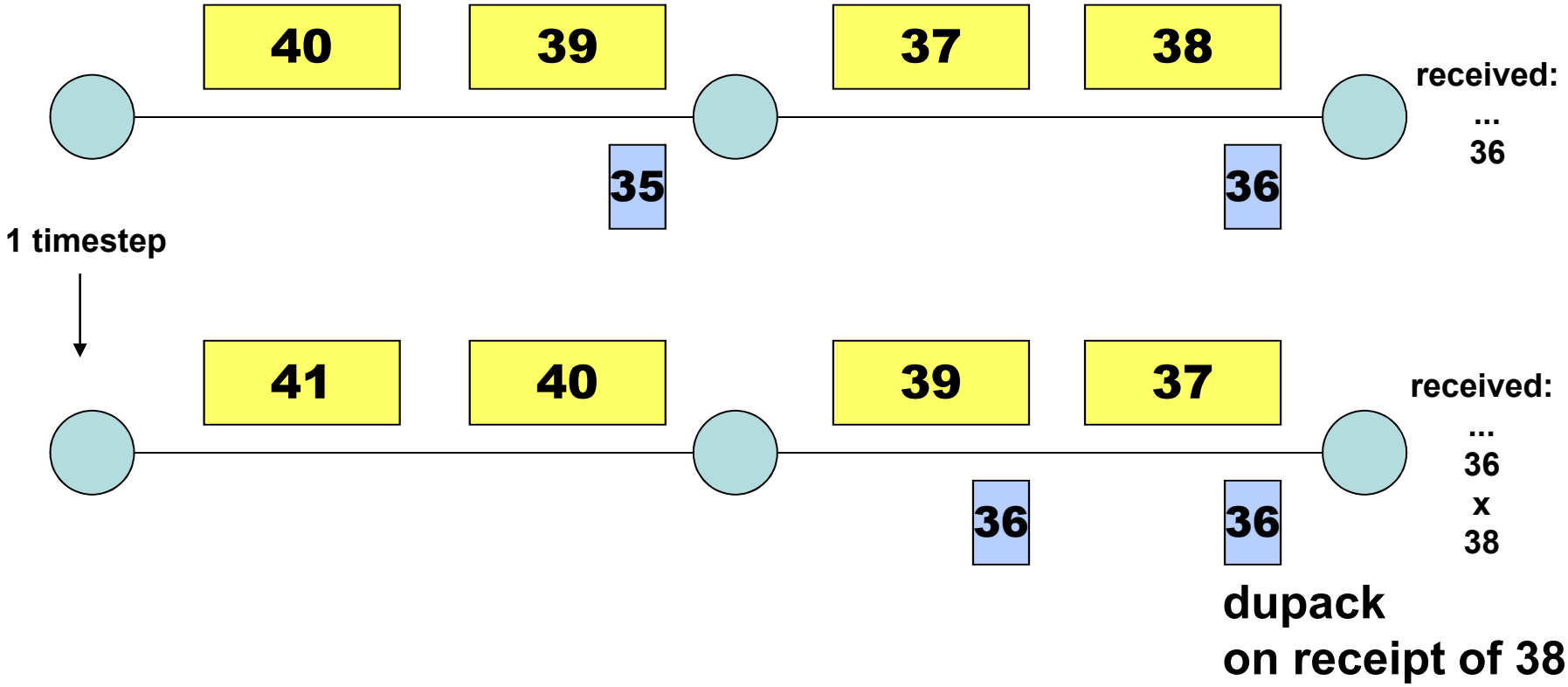
Duplicate Acknowledgements

- A **dupack** is generated whenever an **out-of-order** segment arrives at the receiver (packet 37 gets lost)

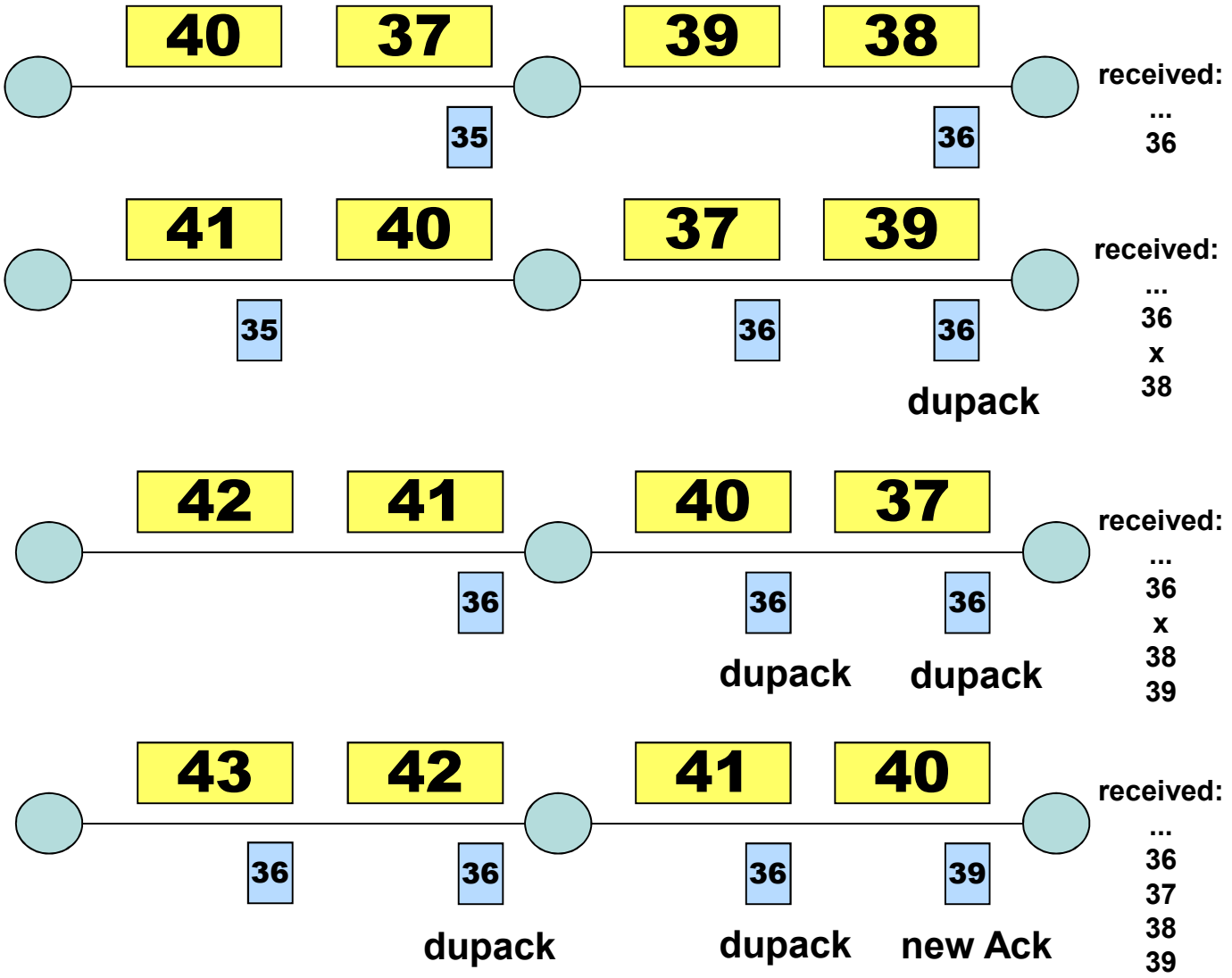


Duplicate Acknowledgements

- Dupacks are not delayed
- Dupacks may be generated when
  - a segment gets lost or
  - a segment is delivered **out-of-order**



Duplicate Acknowledgements

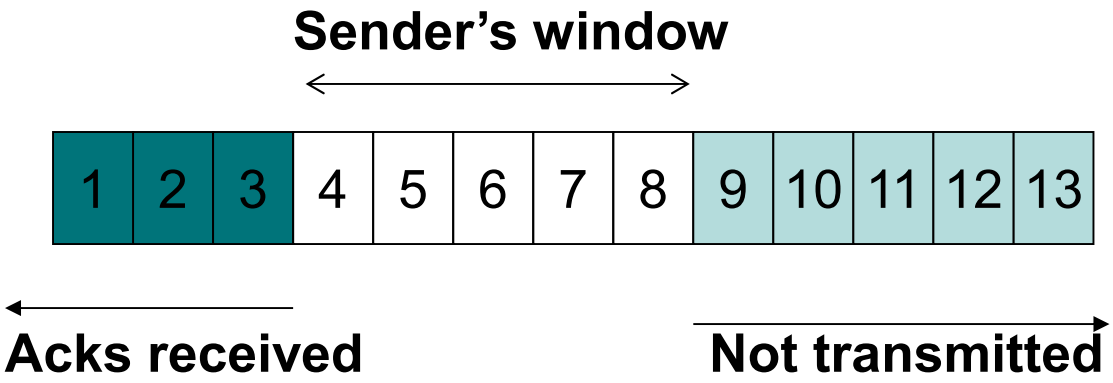


Number of dupacks depends on how much out-of-order a packet is

A series of dupacks allows the sender to guess that a single packet gets lost

Window Based Flow Control

- Achieved using sliding window protocol

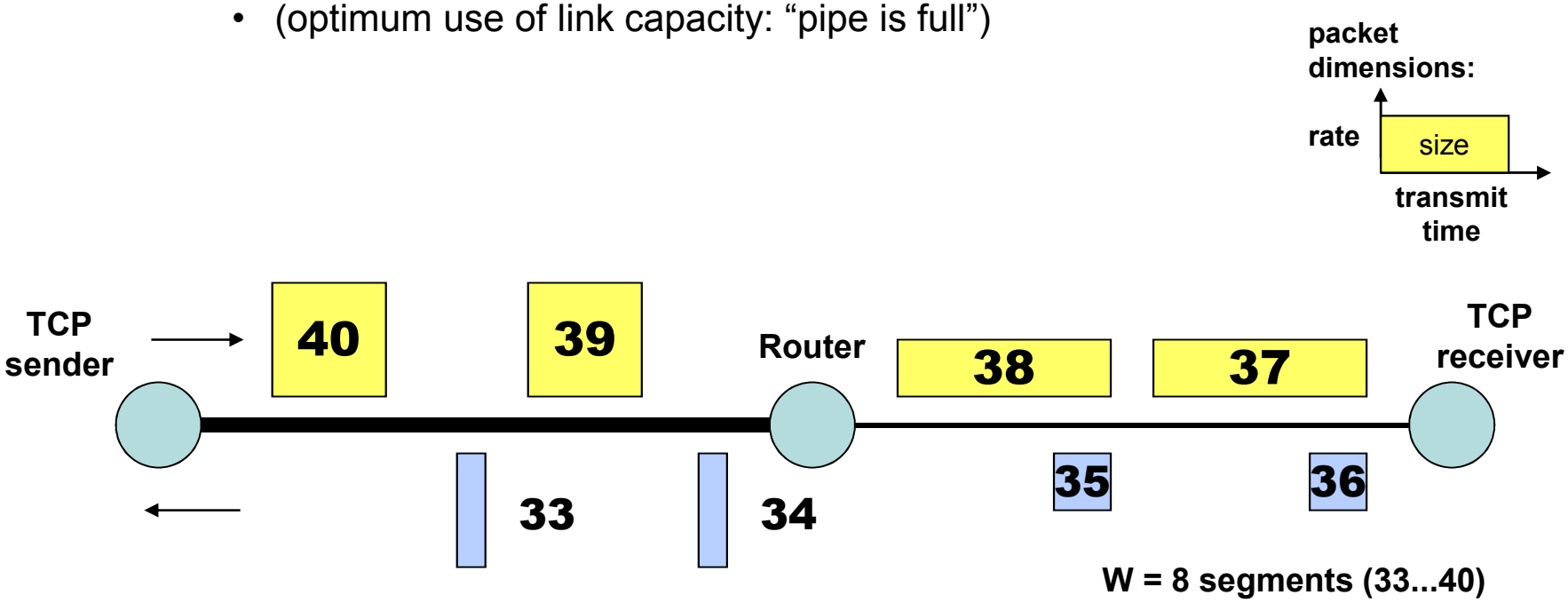


- Window size ***W*** is minimum of
  - **Receiver's advertised window**
    - Determined by available buffer space at the receiver and signaled with each Ack
  - **Congestion window**
    - Determined by the sender based on received Acks

Window Based Flow Control

- TCP's window based flow control is "self-clocking"
  - New segments are sent when outstanding segments are Ack'd

- Optimum window size:
  - W = data rate * RTT = "bandwidth-delay product"
    - (optimum use of link capacity: "pipe is full")



Window Based Flow Control

- What if window size is too large?
  - Queuing at intermediate routers (e.g. at wireless access point)
    - Increased RTT due to queuing delays
    - potential of packet loss

- What if window size is too small?
  - Inefficiency: unused link capacity

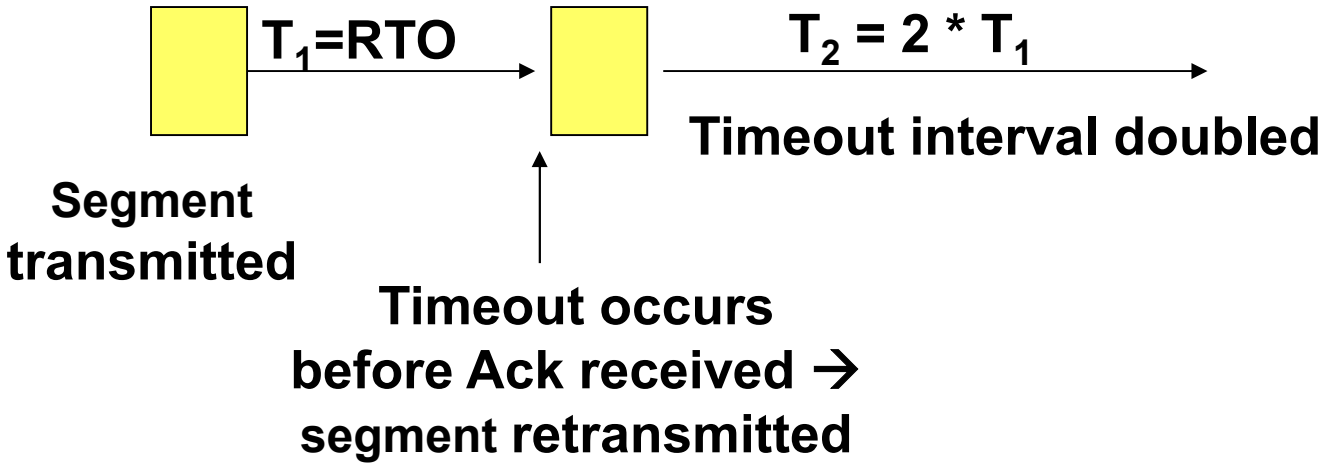
Packet Loss Detection Based on Timeout

- TCP sender starts a timer for a segment (only one segment at a time)
  - If Ack for the timed segment is not received before timer expires → outstanding data are assumed to be lost and retransmitted

- **Retransmission timeout (RTO)** is calculated dynamically
  - Based on the measured RTT

Exponential Backoff

- Double RTO on successive timeouts:

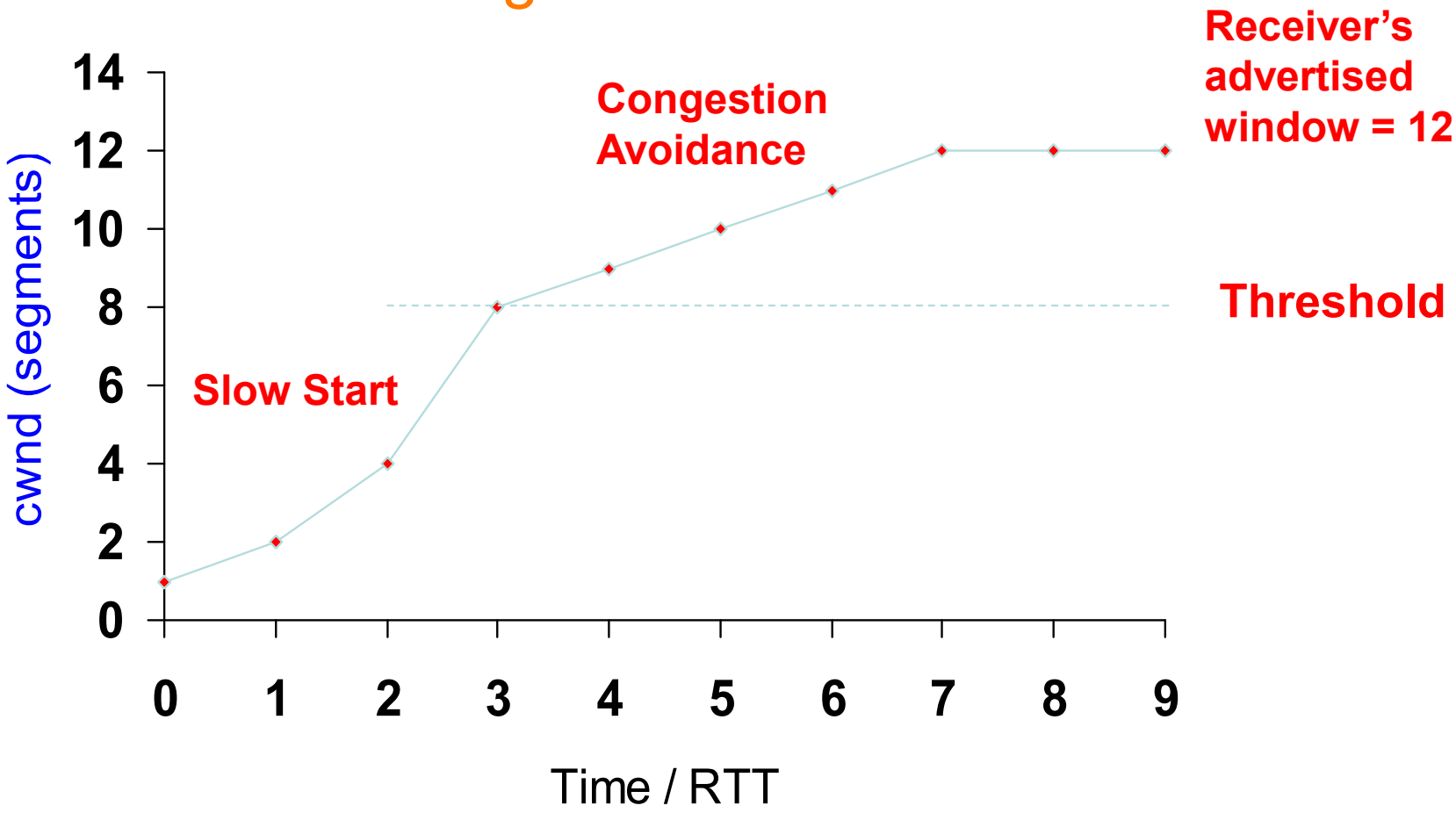


- Total time until TCP gives up is up to 9 min

Packet Loss Detection Based on Dupacks - Fast Retransmit Mechanism

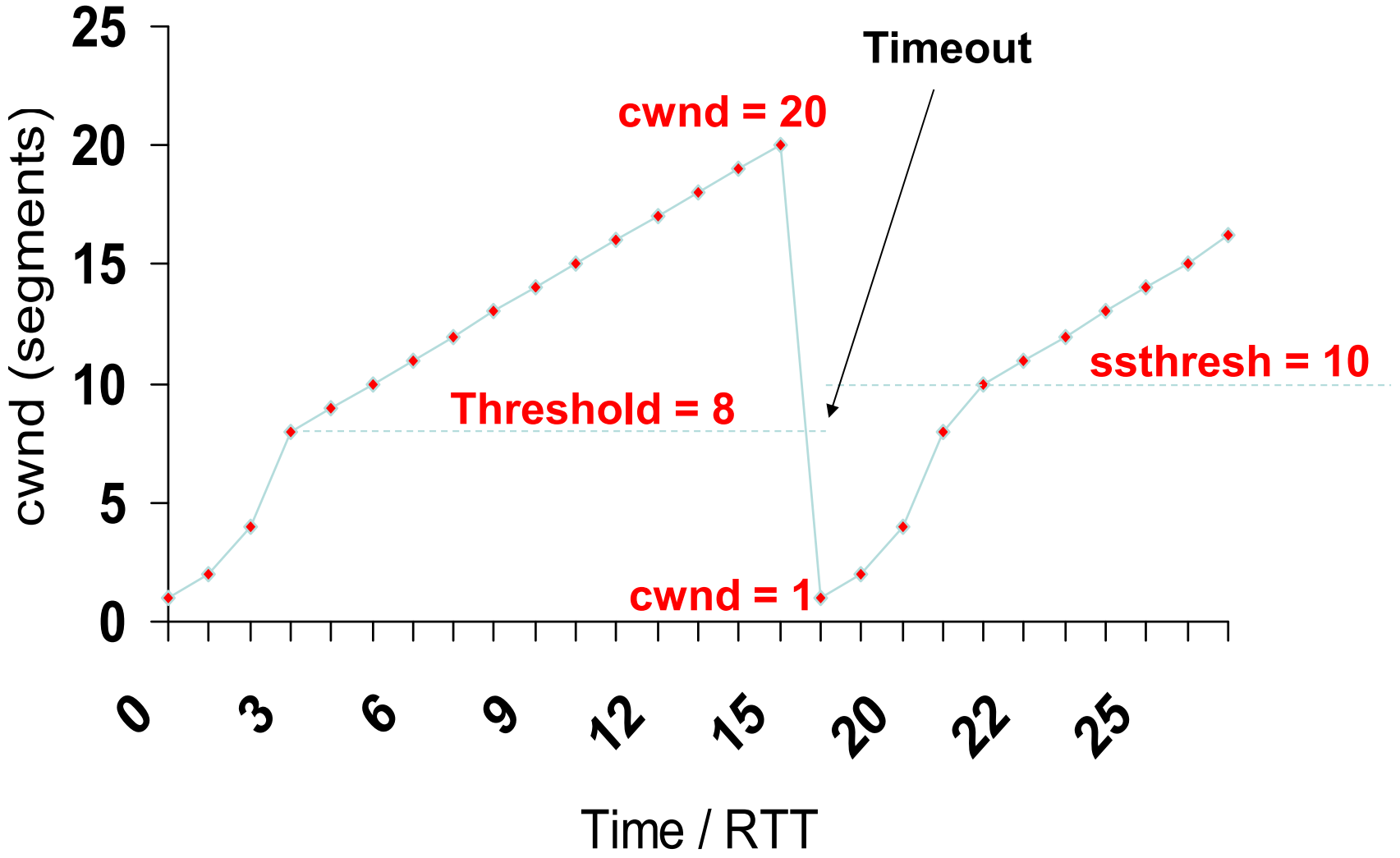
- TCP sender considers timeout as a strong indication that there is a severe link problem
- Continuous reception of Dupacks indicates that following segments are delivered, and the link is ok
- TCP sender assumes that a (single) packet loss has occurred if it receives **three dupacks consecutively**
  - Note: 3 Dupacks are also generated if a segment is delivered at least 3 places out-of-order
- Only the (single) missing segment is retransmitted → **fast retransmission**
- Fast retransmission is useful only if lower layers deliver packets "**almost ordered**"
  - Otherwise, unnecessary fast retransmit

Slow Start and Congestion Avoidance

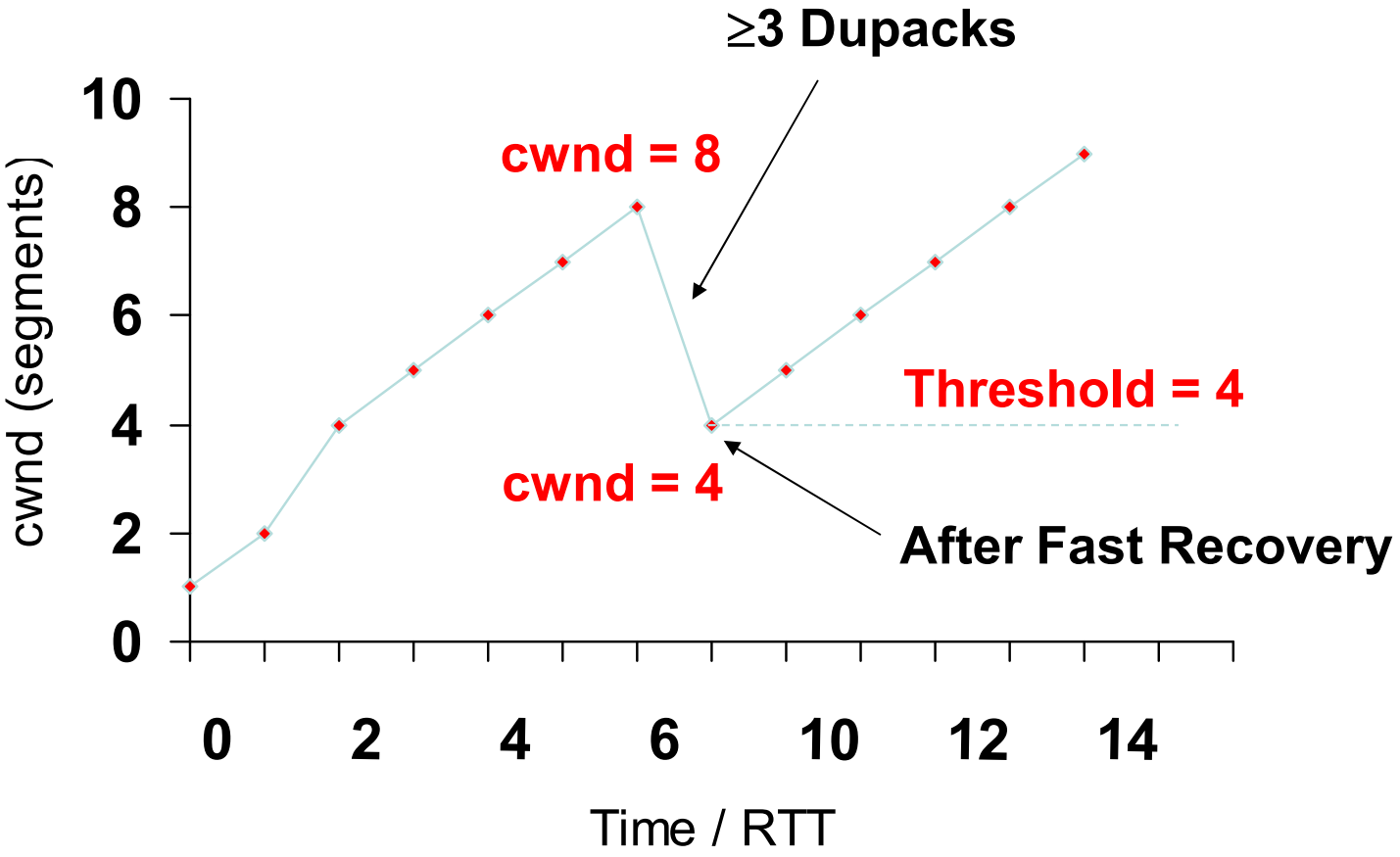


- Theoretical assumption: after sending n segments, n Acks arrive within one RTT
- Note that Slow Start starts slowly, but speeds up quickly

Packet Loss Detected by Timeout



Packet Loss Detected by ≥3 Dupacks

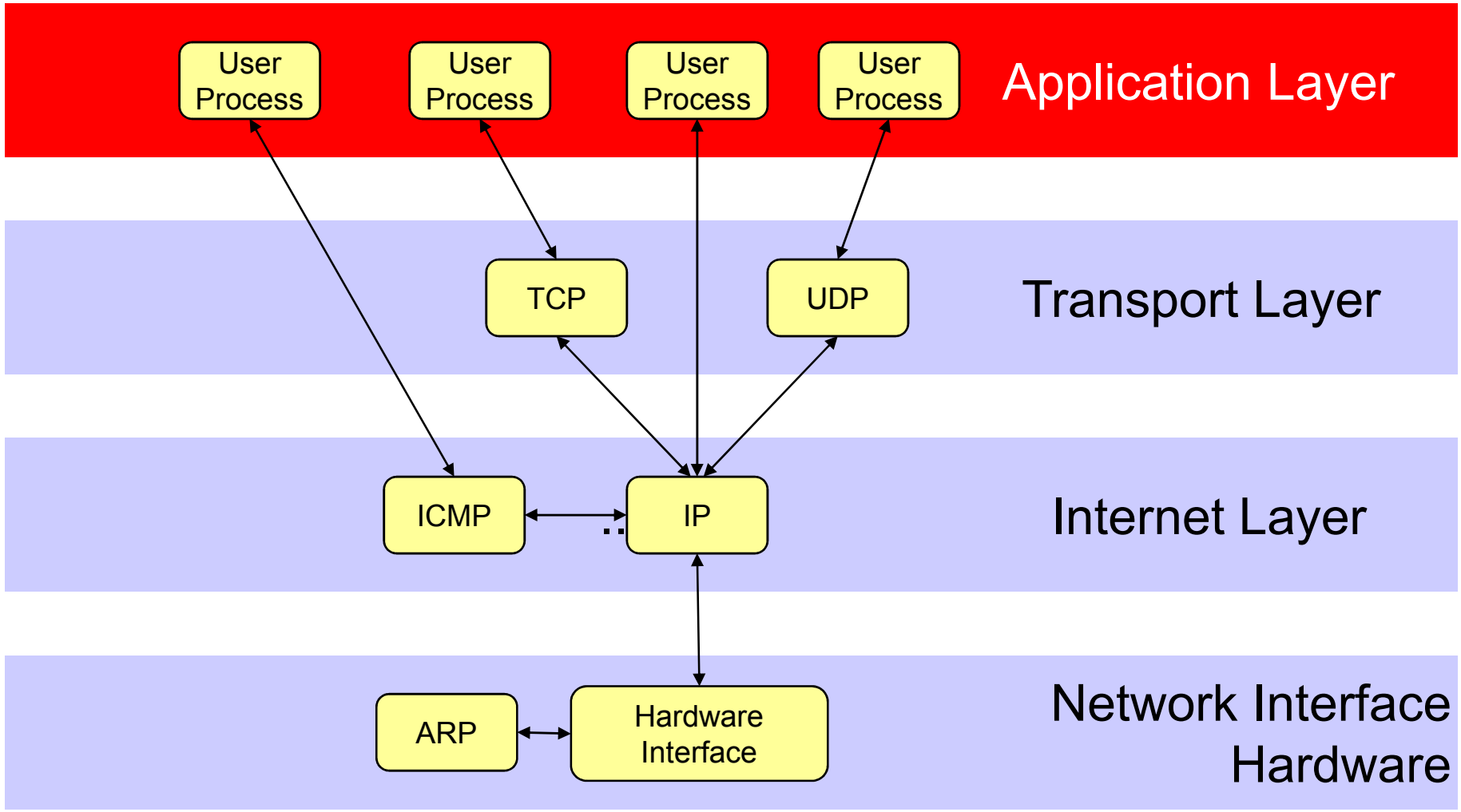


- After fast retransmit and fast recovery window size is reduced in half
- Multiple packet losses within one RTT can result in timeout

Summary

- TCP provides a connection-oriented reliable byte-stream service
- Application data stream is transferred in segments based on lower layer MTU
- TCP employs sliding window mechanism with flow control based on
  - Receiver's advertised window
  - Sender's Slow Start and Congestion Avoidance mechanisms

Application Layer (Selected Applications)



File Transfer Protocol (FTP)

- File Transfer Protocol (FTP)
- File transfer based on **TCP**
- Two connections
  - TCP control connection
    - To well-known server port 21
    - ASCII commands
  - TCP data connection
- QoS requirements:
  - High throughput (optimise TCP bulk data flow)
- Specified in RFC 959

Telnet and Rlogin

- Used for remote login based on **TCP**
  - Rlogin
    - Specified in RFC 1282
    - Simple protocol designed for UNIX hosts
  - Telnet
    - Specified in RFC 854
    - Works with any operating system
    - Option negotiation
    - More flexible and better performance
- QoS requirements
  - Low Round Trip Time (RTT) transport of small packets (optimise TCP interactive data flow)

Hypertext Transfer Protocol (HTTP)

- Hypertext Transfer Protocol
- Transfer of webpages based on **TCP**
  - Webpage typically consists of
    - HTML (HyperText Markup Language) files and
    - Various embedded objects (e.g. pictures, sound files, etc.)
- HTTP/1.0
  - Objects are requested and received serially
  - For each object
    - **New TCP connection** is established, utilized and released
  - **Multiple connections**
    - several TCP connections can be used in parallel

Hypertext Transfer Protocol (HTTP)

- HTTP/1.1
  - The performance is improvements by
    - **Persistent Connections**
      - TCP connections are not released after each object, but used for the next one
        - » avoids TCP connection establishment and termination
        - » avoids slow start for each new connection
    - **Pipelining**
      - Multiple objects can be requested in one request
      - Requested objects are sent sequentially over one TCP connection
  - **Multiple connections** are possible

Real-time Transport Protocol (RTP)

- Transfer of real-time data based on **UDP**

- RTP
  - Real-time Transport Protocol
  - For real-time applications (**audio/video**)
  - **Services**: payload type specification, sequence numbering, timestamping, source identification & synchronization, delivery monitoring
  - No guaranteed quality of service (QoS)

- Network independent
  - Used on top of unreliable and low-delay transport service

- Specified in RFC 1889

Real-time Transport Control Protocol (RTCP)

- RTCP
  - Real-time Transport Control Protocol
  - Contains additional to RTP
    - QoS monitoring and periodic feedback
      - Sender report (synchronization, expected rates, distance)
      - Receiver report (loss ratios, jitter)
- Network independent
- Specified in RFC 1889

Conclusions

- The TCP/IP protocol suite is a set of protocols designed for the **Internet**
- Routing is done via **IP**
- Application data transport using
  - **UDP**: unreliable datagram service
  - **TCP**: reliable byte-stream service
- TCP/IP stack is part of each operating system
- TCP performance is extremely important
  - TCP carries 62% of the flows, 85% of the packets, and 96% of the bytes of Internet traffic
  - TCP error control mechanisms are designed for wired networks
    →Serious problems when used for wireless transport

References

- Course of „**Mobile Communication Networks**“, group of Integrated Communication Systems, Ilmenau University of Technology, Germany (www.tu-ilmenau.de/ics)

- Internet websites:
  - http://www.cs.columbia.edu/~hgs/internet/traffic.html
  - www.ietf.org