

Internet and Intranet

Lecture by:

Jalauddin Mansur

June 2015

Chapter 2: Internet Protocol Overview

Topics:

- TCP/IP and the IP layer Overview
- IPV4 and IPV6 Address Types and Formats
- IPV4 and IPV6 Header Structure
- Internet RFCs

TCP/IP Model

4 layers of the TCP/IP model

- Layer 4: Application
- Layer 3: Transport
- Layer 2: Internet
- Layer 1: Network access (Host to Network)

The network access layer(Host to Network)

- Concerned with all of the issues that an IP packet requires to actually make the physical link.
- All the details in the OSI physical and data link layers.
 - Electrical, mechanical, procedural and functional specifications
 - Data rate, Distances, Physical connector.
 - Frames
 - Synchronization, flow control, error control.

The Internet Layer

- Concerned with Packet Addressing
 - Send source packets from any network and have them arrive at the destination independent of the path and networks they took to get there.
 - May arrive in different order to destination
 - Job of higher layer to arrange them
- Internet Protocol (IP)
 - Deliver packet to exact destination
- Packet Routing
 - Important to avoid congestion

The Transport Layer

- The transport layer deals with the quality-of-service issues of reliability, flow control, and error correction.
- Connection oriented and connectionless.
 - Transmission control protocol (TCP).
 - User datagram protocol (UDP).
- End-to-end flow control.
- Error detection and recovery.
- Allows end-to-end communication
- Connection establishment, error control, flow control

The Transport Layer

Two main protocol at transport layer are

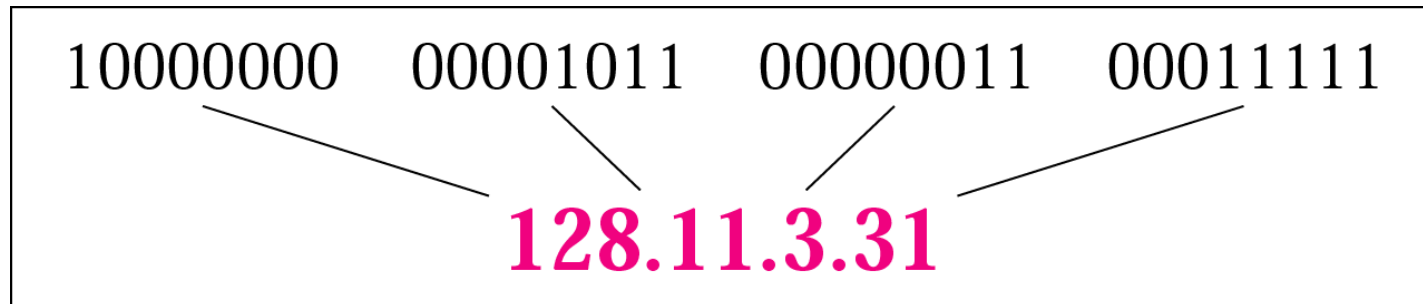
- Transmission control protocol (TCP)
 - Connection oriented
 - Connection established before sending data
 - Reliable
- User datagram protocol (UDP)
 - Connectionless
 - Sending data without establishing connection
 - Fast but unreliable

The Application Layer

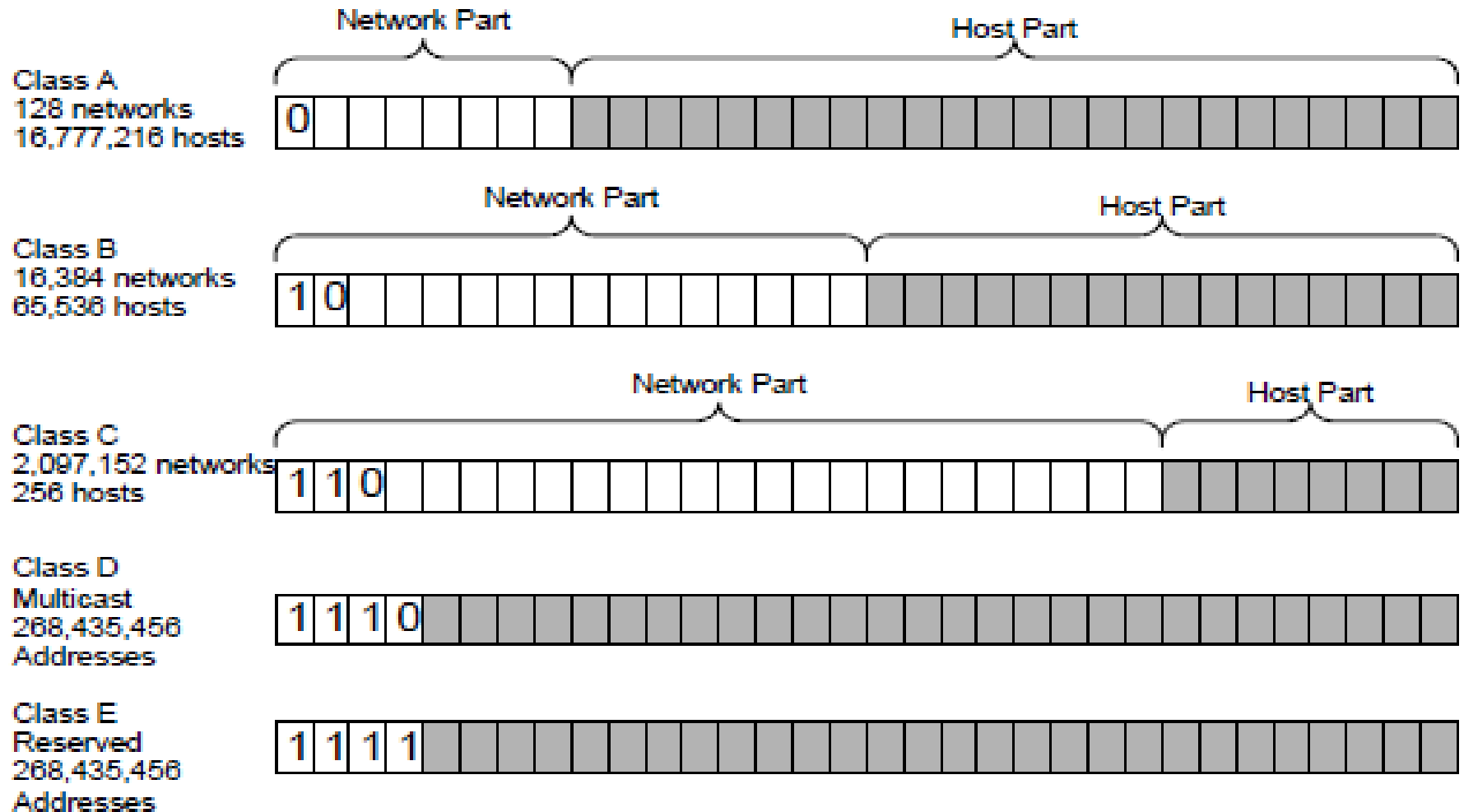
- Handles high-level protocols
- Applications(Software) that works at Application Layer
 - FTP, HTTP, SMTP, DNS
 - Format of data, data structure, syntax and semantics.

IPv4 Address

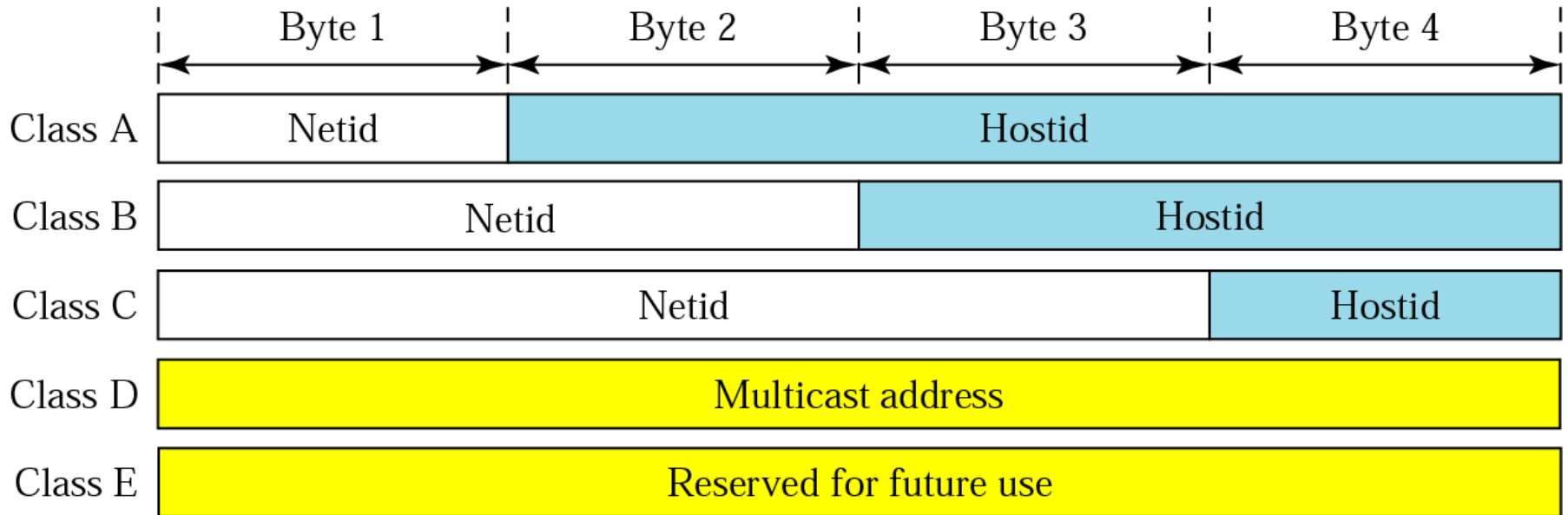
- 32 bit address
- Total unique address equals to 2^{32}
 - Around 4.2 billion address
- Represented in dotted Decimal Format



Classful Addressing



Network Id and Host Id in Classful addressing

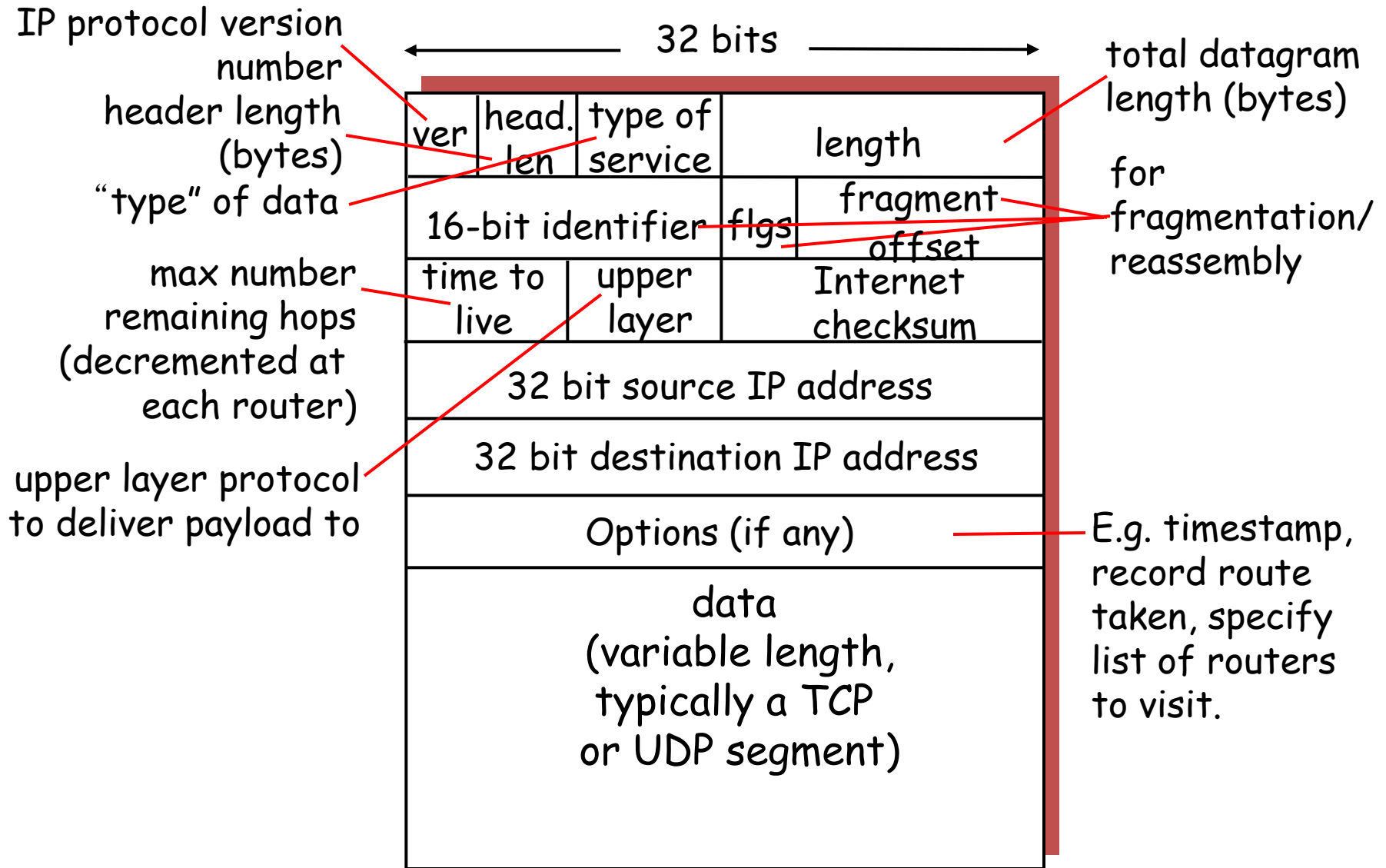


Default subnet mask for Classful address

Class	<i>In Binary</i>	<i>In Dotted-Decimal</i>	<i>Using Slash</i>
A	11111111 00000000 00000000 00000000	255.0.0.0	/8
B	11111111 11111111 00000000 00000000	255.255.0.0	/16
C	11111111 11111111 11111111 00000000	255.255.255.0	/24

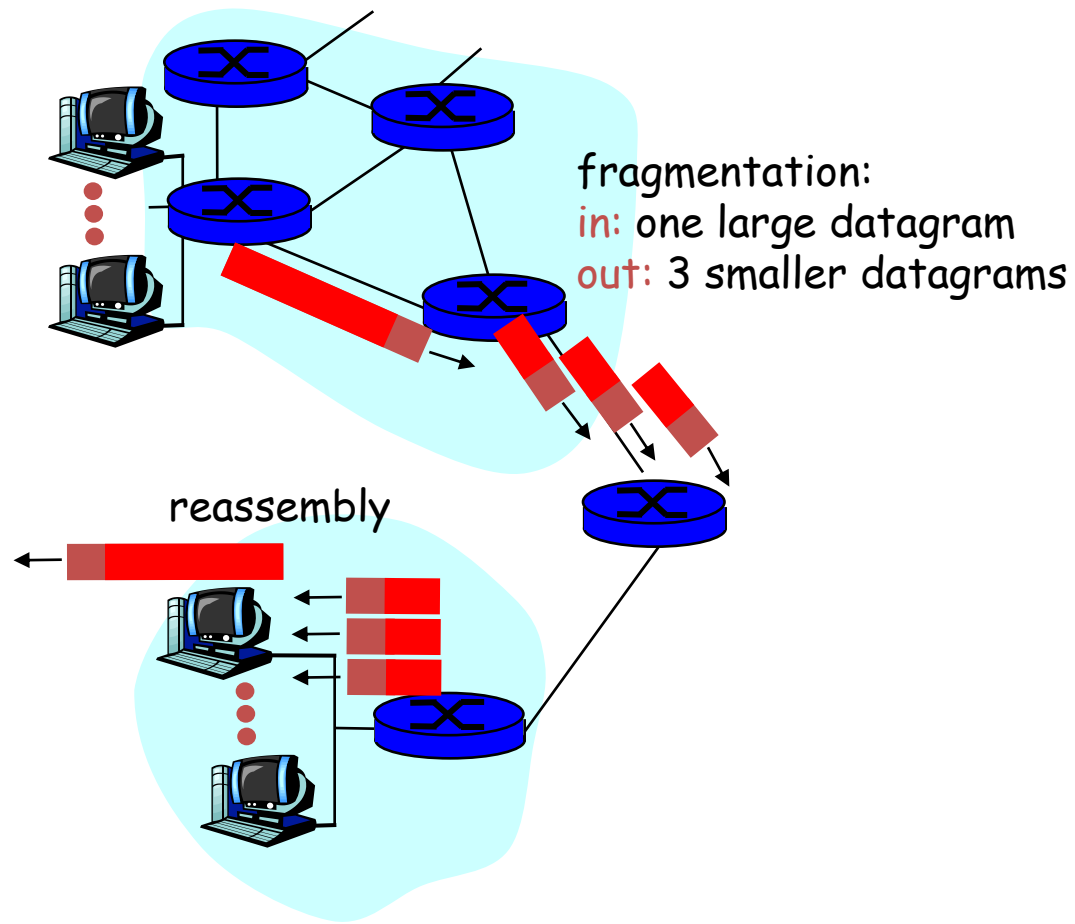
- Logical Address:
 - IP address at the Network Layer
 - Used to Communicate with the different subnets
- Netid: Identify network
- Hostid: Identify End devices
- Mask: Used to find netid and hostid
- CIDR: Classless interdomain routing
 - Used in classless addressing
 - Defined by slash notation /n
 - Example: /8, /16, /24

IP datagram format

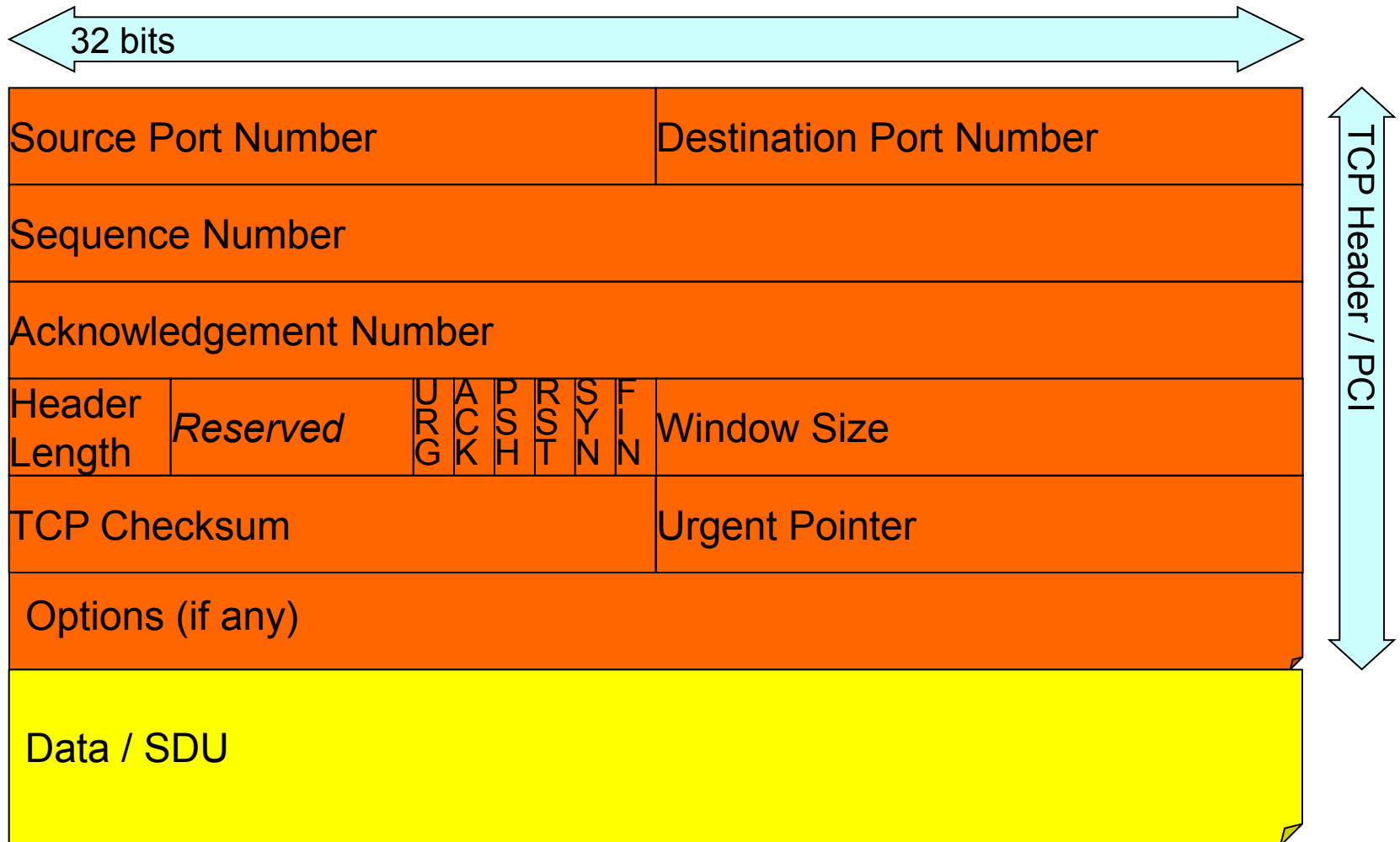


IP Fragmentation and Reassembly

- network links have MTU (max. transfer unit) - largest possible 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



TCP Segment



TCP Segment

- Source Port: The 16-bit source port number, used by the receiver to reply.
- Destination Port: The 16-bit destination port number
- Sequence Number: The sequence number of the first data byte in this segment.
- Acknowledgment Number: If the ACK control bit is set, this field contains the value of the next sequence number that the receiver is expecting to receive.
- Header length: A 4-bit field that represents the header length in multiple of four bytes
- Reserved: Six bits reserved for future use; must be zero.
- URG: Indicates that the urgent pointer field is significant in this segment.

- ACK: Indicates that the acknowledgment field is significant in this segment.
- PSH: Push function.
- RST: Resets the connection
- SYN: Synchronizes the sequence numbers.
- FIN: No more data from sender.
- Window size: It specifies the number of data bytes that the receiver is willing to accept.
- Checksum: A 16-bit field used for error correction.
- Urgent Pointer: Points to the first data octet following the urgent data.

TCP Connections

- TCP identifies connections on the basis of endpoints:
 - IP address + port number
 - Often written as: IP-address : port-number, for instance: 130.89.17.3:80
- Two endpoints define a connection

TCP Connection Management

- **Opening a connection (3-way handshake):**

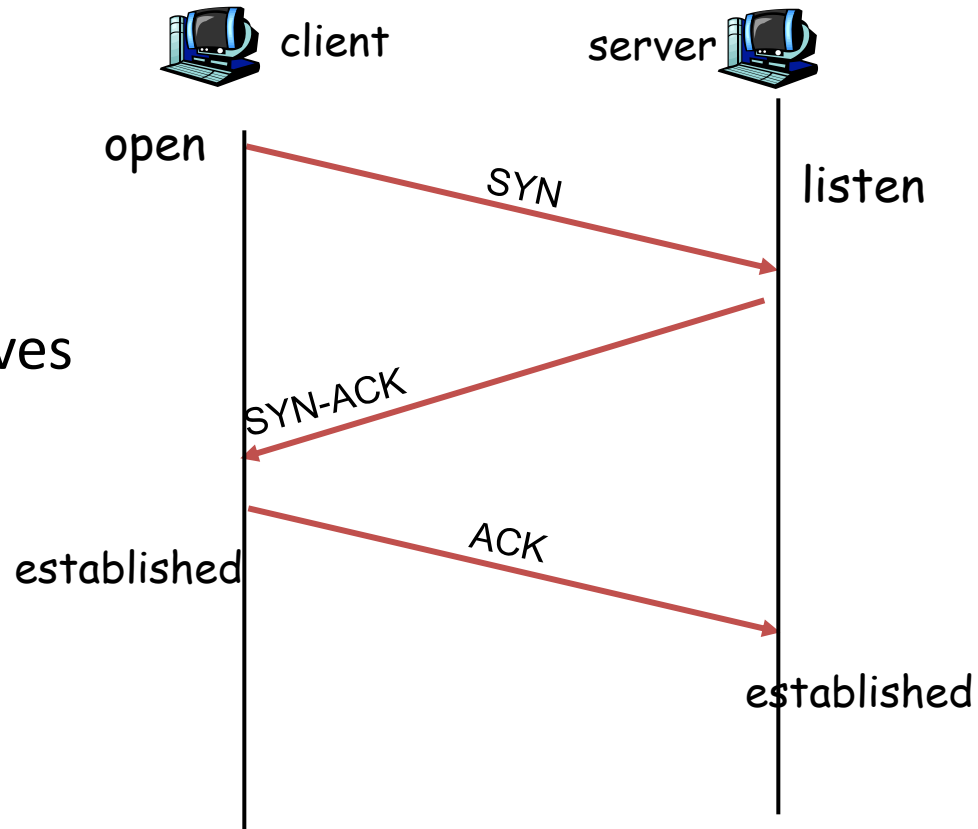
Step 1: client end system sends TCP SYN control segment to server

Step 2: server end system receives SYN, replies with SYN-ACK

- allocates buffers
- ACKs received SYN

Step 3: client receives SYN-ACK

- connection is now set up
- client starts the “real work”

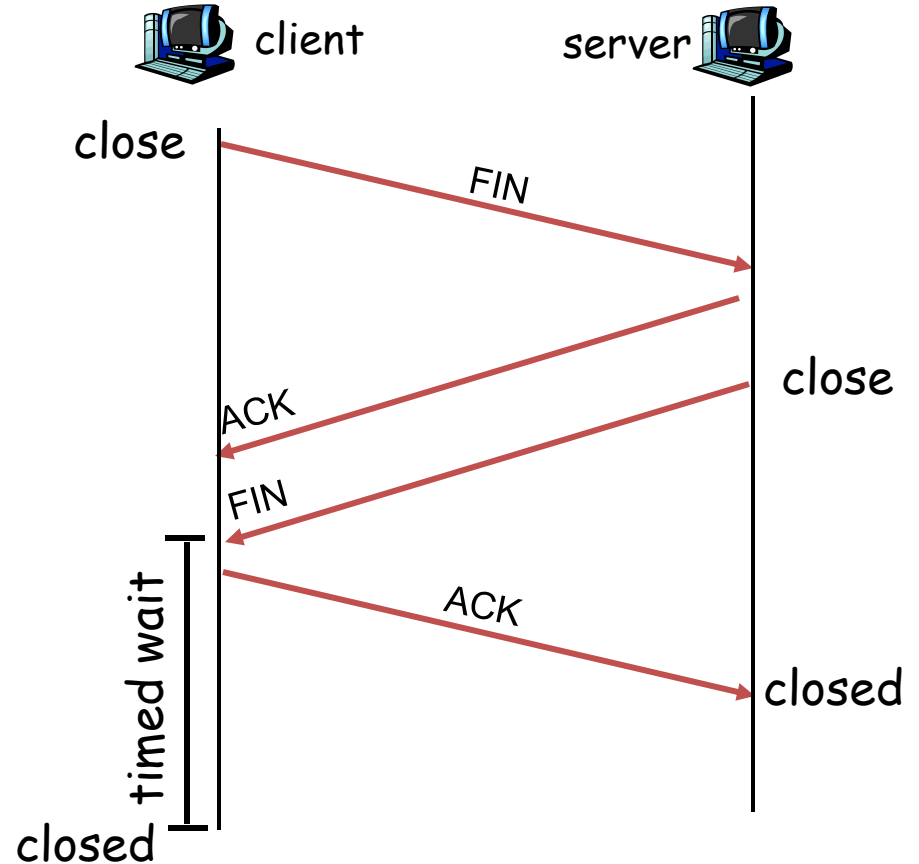


TCP Connection Management contd..

- **Closing a connection:**

Step 1: client end system sends TCP FIN control segment to server

Step 2: server receives FIN, replies with ACK. Closes connection, sends FIN.

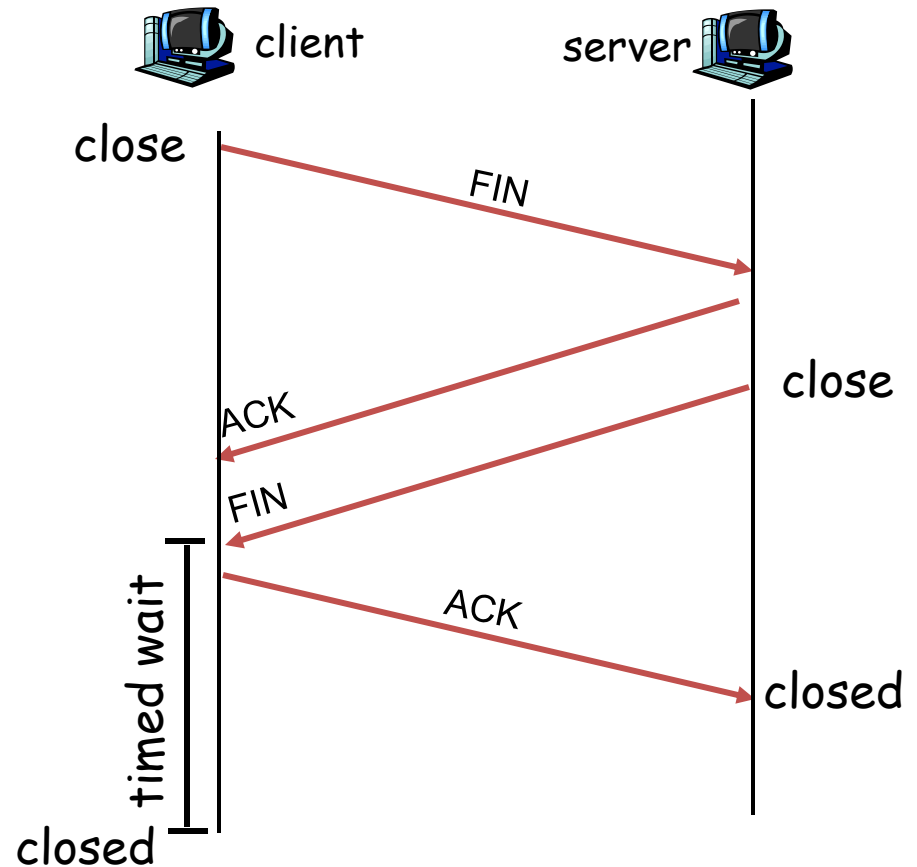


TCP Connection Management contd..

Step 3: client receives FIN,
replies with ACK.

- Enters “timed wait” -
will respond with ACK
to received FINs

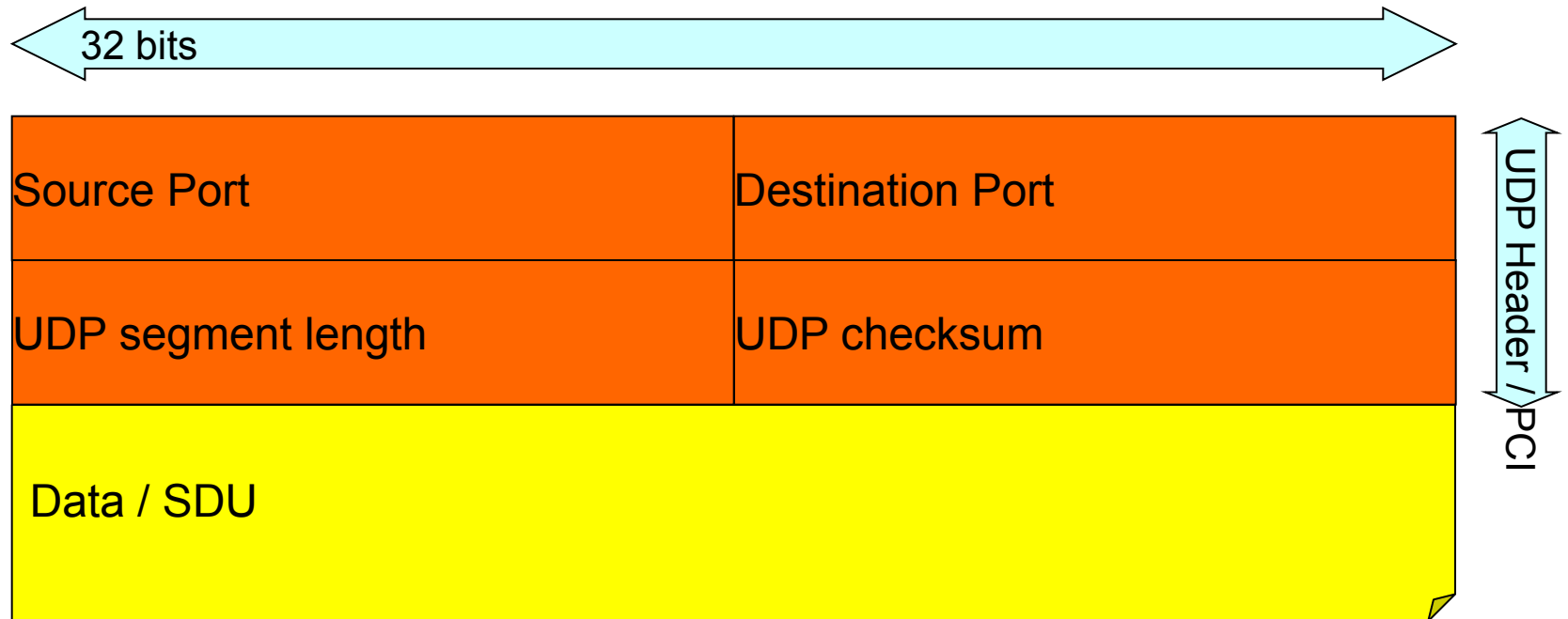
Step 4: server, receives
ACK. Connection closed.



TCP Operation

- TCP provides the following major services to the upper protocol layers:
 - **Connection-oriented data management** to assure the end-to-end transfer of data across the network(s).
 - **Reliable data transfer** to assure that all data is accurately received, in sequence and with no duplicates.
 - **Stream-oriented data transfer** takes place between the sender application and TCP and the receiving application and TCP.
- Prior to data transmission, hosts establish a ***virtual connection*** via a synchronization process. The synch process is a 3-way “handshake”, which ensures both sides are ready to transfer data and determines the initial sequence numbers.
- Sequence numbers are reference numbers between the two devices.
- Sequence numbers give hosts a way to acknowledge what they have received.
- TCP header contain SYN bits, or flags, to achieve this.

UDP Segment



Why Would Anyone Use UDP?

- No delay for connection establishment
 - UDP just blasts away without any formal preliminaries
 - avoids introducing any unnecessary delays
- No connection state
 - No allocation of buffers, parameters, sequence numbers, etc.
 - easier to handle many active clients at once
- Small packet header overhead
 - UDP header is only eight-bytes long

Comparison of TCP and UDP

UDP - User Datagram Protocol

- datagram oriented
- unreliable, connectionless
- simple
- unicast and multicast
- No windows or ACKs
- Smaller header, less overhead
- No Sequencing
- useful only for few applications, e.g., multimedia applications
- network management (SNMP), routing (RIP), naming (DNS), etc.

TCP - Transmission Control Protocol

- stream oriented
- reliable, connection-oriented
- complex
- only unicast
- Uses windows or ACKs
- Full header
- Sequencing
- used for most Internet applications
- web (http), email (smtp), file transfer (ftp), terminal (telnet), etc.

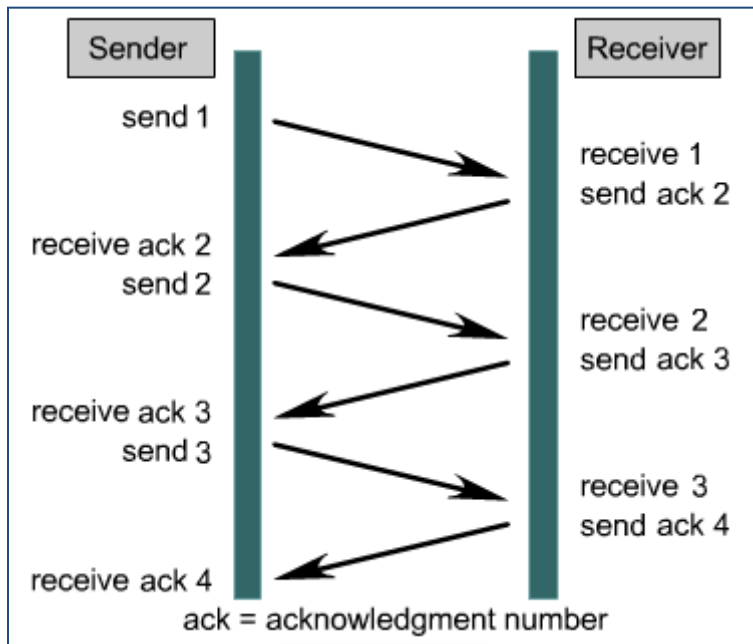
Flow Control

- Flow control at this layer is performed end-to-end rather than across a single link.
- A sliding window is used to make data transmission more efficient as well as to control the flow of data so that the receiver does not become overwhelmed.
- Some points about sliding windows at the transport layer:
 - The sender does not have to send a full window's worth of data.
 - An acknowledgment can expand the size of the window based on the sequence number of the acknowledged data segment.
 - The size of the window can be increased or decreased by the receiver.
 - The receiver can send an acknowledgment at anytime.

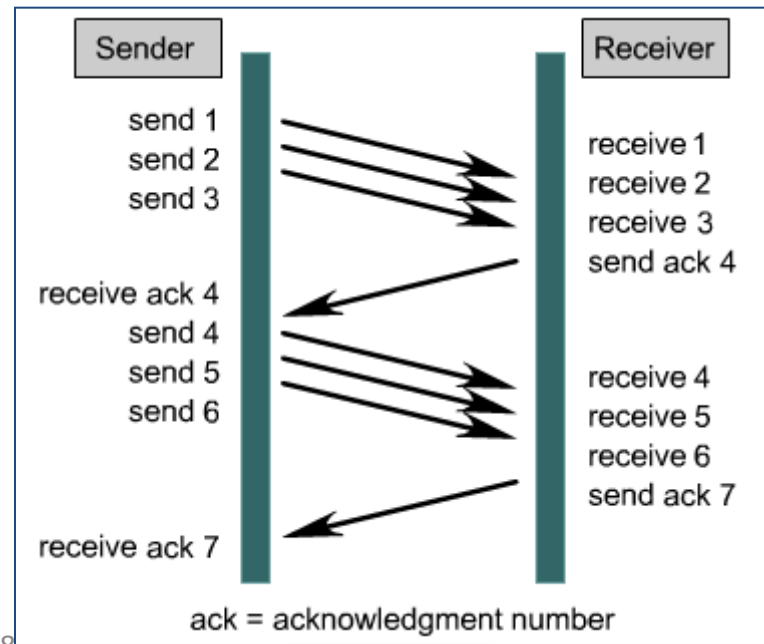
TCP Windows and Flow Control

- Data often is too large to be sent in a single segment.
- TCP splits the data into multiple segments.
- TCP provides flow control through “windowing” to set the pace of how much data is sent at a time
 - how many bytes per window, and how many windows between ACKs.

Window Size = 1



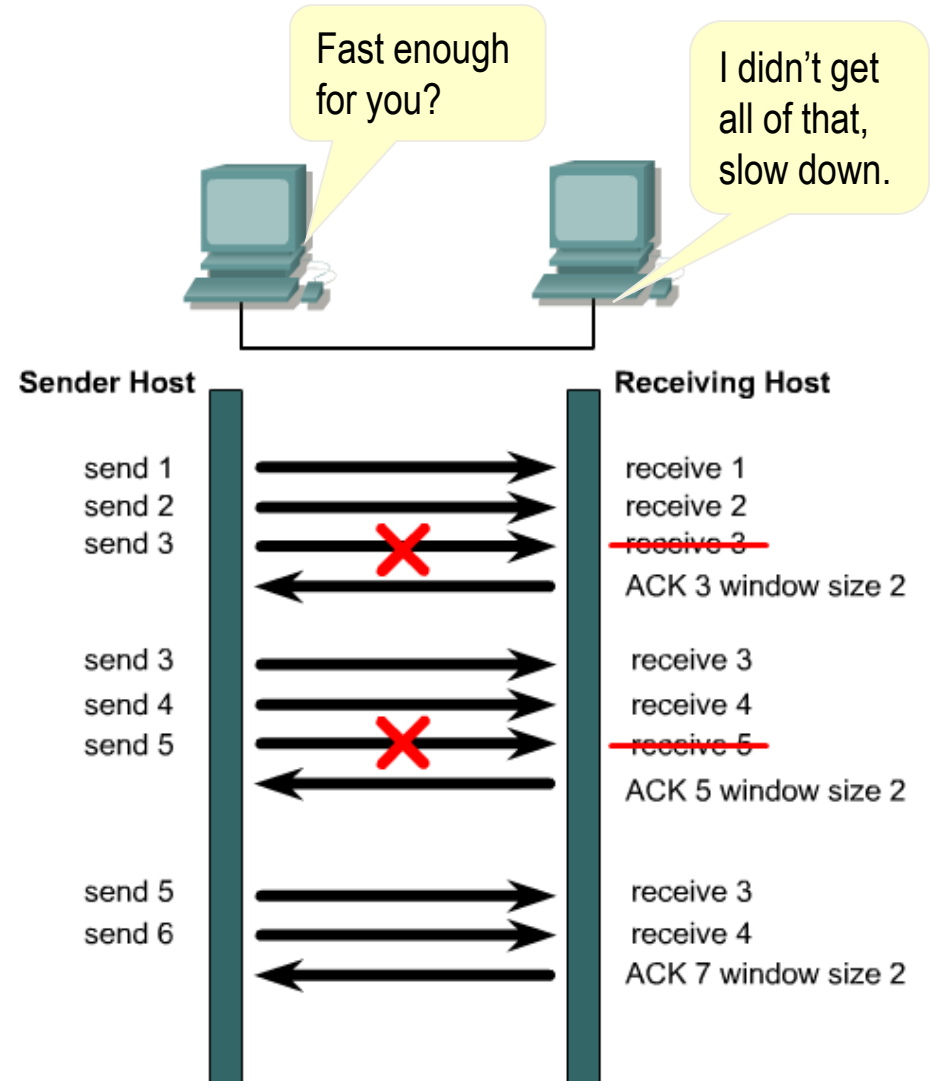
Window Size = 3



Windowing and Window Size

Sliding window refers to the fact that the window size is negotiated dynamically during the TCP session.

If the source receives no acknowledgment, it knows to retransmit at a slower rate.



Reliable Delivery

Sequence and ACK Numbers

- Each TCP segment is numbered before transmission so that the receiver will be able to properly reassemble the bytes in their original order.
- They also identify missing data pieces so the sender can retransmit them.
- Only the missing segments need to be re-transmitted.

Positive Acknowledgement and Retransmission

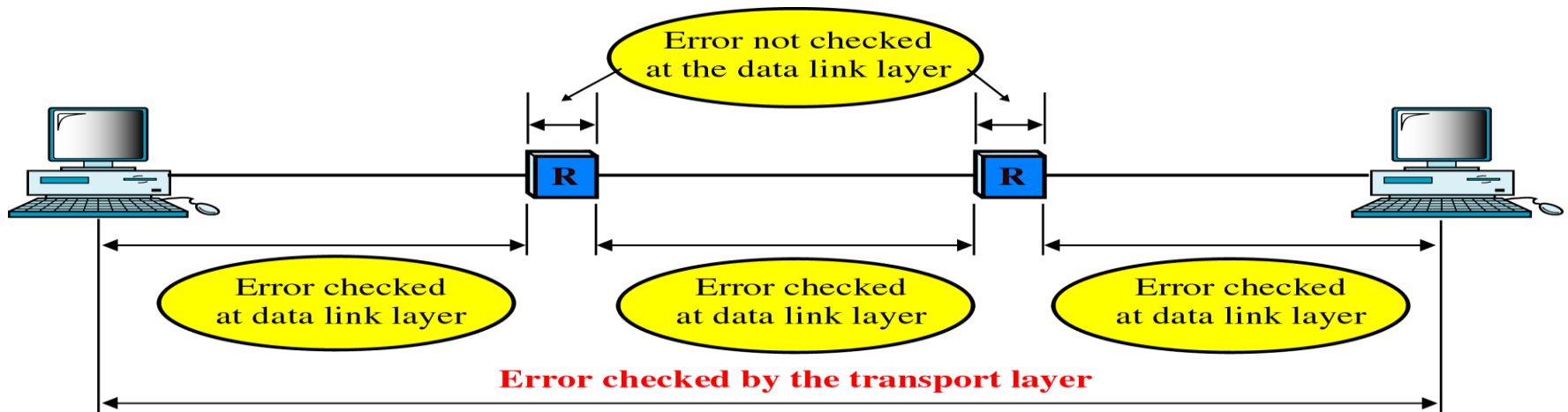
TCP utilizes PAR to control data flow and confirm data delivery.

- Source sends packet, starts timer, and waits for ACK.
- If timer expires before source receives ACK, source retransmits the packet and restarts the timer.

Reliable Delivery

Error Control

- Mechanisms for error control are based on error detection and retransmission.
- Error detections are performed using algorithms implemented in software, such as checksum.
- We already have error handling at the data link layer, why do we need it at the transport layer?



Reliable Delivery contd..

Segmentation and Reassembly

- Transport layer adds a sequence number at each segment.
- This number indicates the order for reassembly.
- Each segment carries a field that indicates whether it is the final segment or middle segment of a transmission.

Concatenation and Separation

- When the size of the data unit belonging to a single session is so small that several units can fit together into a single datagram.
- A sequence number at each unit allows correct separation at the destination.

IPv6 Introduction

- Major points that played a key role in the birth of IPv6:
 - Internet has grown exponentially
 - address space allowed by IPv4 is saturating.
 - There is a requirement to have a protocol that can satisfy the needs of future Internet addresses
 - IPv4 on its own does not provide any security features.
 - Data has to be encrypted with some other security application before being sent on the Internet.

- Data prioritization in IPv4 is not up-to-date.
 - Though IPv4 has a few bits reserved for Type of Service or Quality of Service, but they do not provide much functionality.
- IPv4 enabled clients can be configured manually or they need some address configuration mechanism.
 - It does not have a mechanism to configure a device to have globally unique IP address.

Why IPv6?

- Shortage of IPv4 addresses
 - Internet is expanding very rapidly in developing countries like India, China
 - New devices like phones need IP address
- New Features like Auto configuration, better support for QoS, Mobility and Security, Route Aggregation, Jumbo Frames
- Note: Include all the points of earlier slide (Introduction slide) as well for why IPV6?

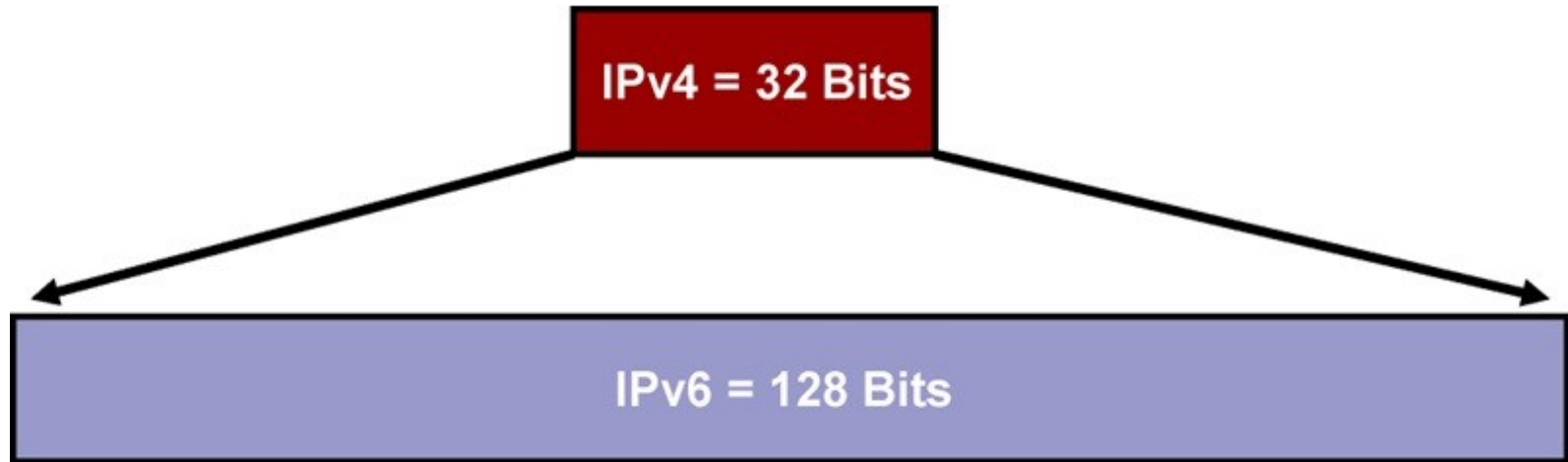
IPv6 Benefits

- Expanded addressing capabilities
- Structured hierarchy to manage routing table growth
- Serverless auto configuration and reconfiguration
- Streamlined header format and flow identification
- Improved support for options / extensions
- Security (IPsec mandatory)

IPv6 Advanced Features

- Security - Built-in, strong IP-layer encryption and authentication
- Mobility - More efficient and robust mechanisms
- Quality of Service
- Privacy Extensions for Stateless Address Auto configuration (RFC 3041)
- Source address selection

IPv6 Address




- IPv4: 32 bits or 4 bytes long
 - 4,200,000,000 possible addressable nodes
- IPv6: 128 bits or 16 bytes
 - $3.4 * 10^{38}$ possible addressable nodes
 - 340,282,366,920,938,463,374,607,432,768,211,456
 - $5 * 10^{28}$ addresses per person

Why Was 128 Bits Chosen as the IPv6 Address Size?

- Proposals for fixed-length, 64-bit addresses
- Proposals for variable-length, up to 160 bits
 - Accommodates auto-configuration using IEEE 802 addresses
 - Sufficient structure for projected number of service providers
- Settled on fixed-length, 128-bit addresses
 - (340,282,366,920,938,463,463,374,607,431,768,211,456)

128-bit IPv6 Address Representation

3FFE:085B:1F1F:0000:0000:0000:00A9:1234



8 groups of 16-bit hexadecimal numbers separated by “:”

Leading zeros can be removed



3FFE:85B:1F1F::A9:1234

:: = all zeros in one or more group of 16-bit hexadecimal numbers

Unabbreviated

FDEC ■ BA98 ■ 0074 ■ 3210 ■ 000F ■ BBFF ■ 0000 ■ FFFF



FDEC ■ BA98 ■ 74 ■ 3210 ■ F ■ BBFF ■ 0 ■ FFFF

Abbreviated

Abbreviated

FDEC ■ 0 ■ 0 ■ 0 ■ 0 ■ BBFF ■ 0 ■ FFFF



FDEC ■ ■ BBFF ■ 0 ■ FFFF

More Abbreviated

IPv6 Address Representation Contd..

- IPv4 compatible (not used any more)
 - 0:0:0:0:0:0:192.168.30.1
 - = ::192.168.30.1
 - = ::C0A8:1E01
- In a URL, it is enclosed in brackets (RFC3986)
 - [http://\[2001:db8:4f3a::206:ae14\]:8080/index.html](http://[2001:db8:4f3a::206:ae14]:8080/index.html)
- Cumbersome for users
- Mostly for diagnostic purposes
- Use fully qualified domain names

IPv6 Address Representation Contd..

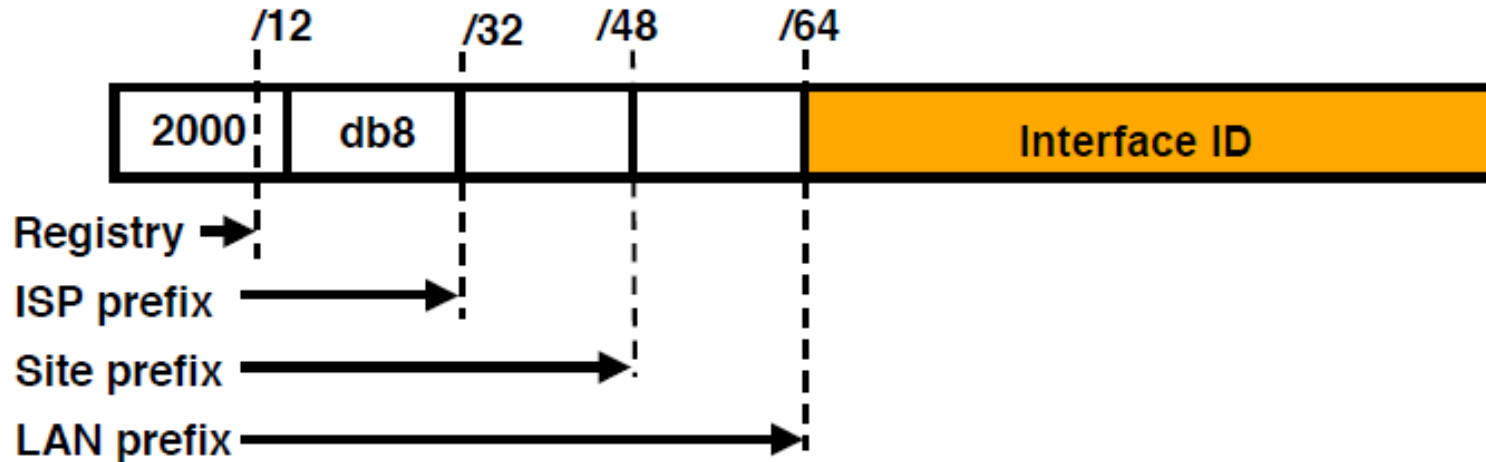
- Prefix Representation
 - Representation of prefix is same as for IPv4 CIDR
 - Address and then prefix length
 - IPv4 address:
 - 198.10.0.0/16
 - IPv6 address:
 - 2001:db8:12::/40

IPv6 ADDRESSING

- There are three types of addresses:
- Unicast
 - An identifier for a single interface (one to one).
- Anycast
 - An identifier for a set of interfaces and is delivered to one of the interfaces identified by that address (one to nearest).
- Multicast
 - An identifier for a set of interfaces and is delivered to all interfaces identified by that address (one to many).
- A single interface may be assigned multiple IPv6 addresses of any type (unicast, anycast, multicast)

Note : There are no broadcast addresses in IPv6, their function being superseded by multicast addresses.

IPv6 Address Allocation



- ❑ The allocation process is:
 - The IANA is allocating out of 2000::/3 for initial IPv6 unicast use
 - Each registry gets a /12 prefix from the IANA
 - Registry allocates a /32 prefix (or larger) to an IPv6 ISP
 - Policy is that an ISP allocates a /48 prefix to each end customer

IPv6 Addressing Scope

- 64 bits reserved for the interface ID
 - Possibility of 2^{64} hosts on one network LAN
 - Arrangement to accommodate MAC addresses within the IPv6 address
- 16 bits reserved for the end site
 - Possibility of 2^{16} networks at each end-site
 - 65536 subnets equivalent to a /12 in IPv4 (assuming 16 hosts per IPv4 subnet)

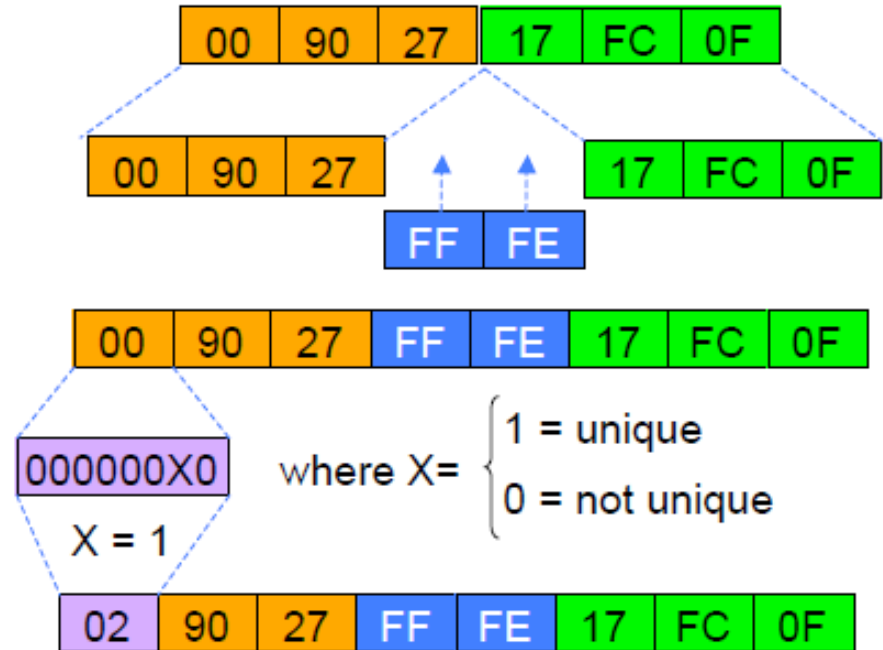
- 16 bits reserved for the service provider
 - Possibility of 2^{16} end-sites per service provider
 - 65536 possible customers: equivalent to each service provider receiving a /8 in IPv4 (assuming a /24 address block per customer)
- 32 bits reserved for service providers
 - Possibility of 2^{32} service providers
 - i.e. 4 billion discrete service provider networks
- Although some service providers already are justifying more than a /32
 - Equivalent to the size of the entire IPv4 address space

Interface IDs

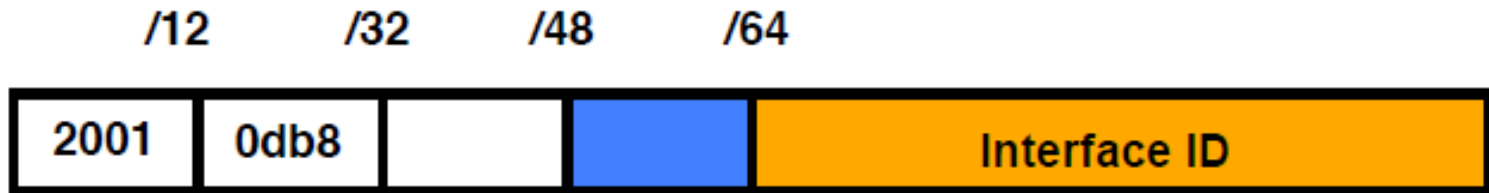
- Lowest-order 64-bit field of unicast address may be assigned in several different ways:
 - auto-configured from a 64-bit EUI-64, or expanded from a 48-bit MAC address (e.g., Ethernet address)
 - auto-generated pseudo-random number (to address privacy concerns)
 - assigned via DHCP
 - manually configured

EUI-64

- Ethernet MAC address(48 bits)
- 64 bits version
- Uniqueness of the MAC
- Eui-64 address
- EUI-64 address is formed by inserting FFFE and OR'ing a bit identifying the uniqueness of the MAC address

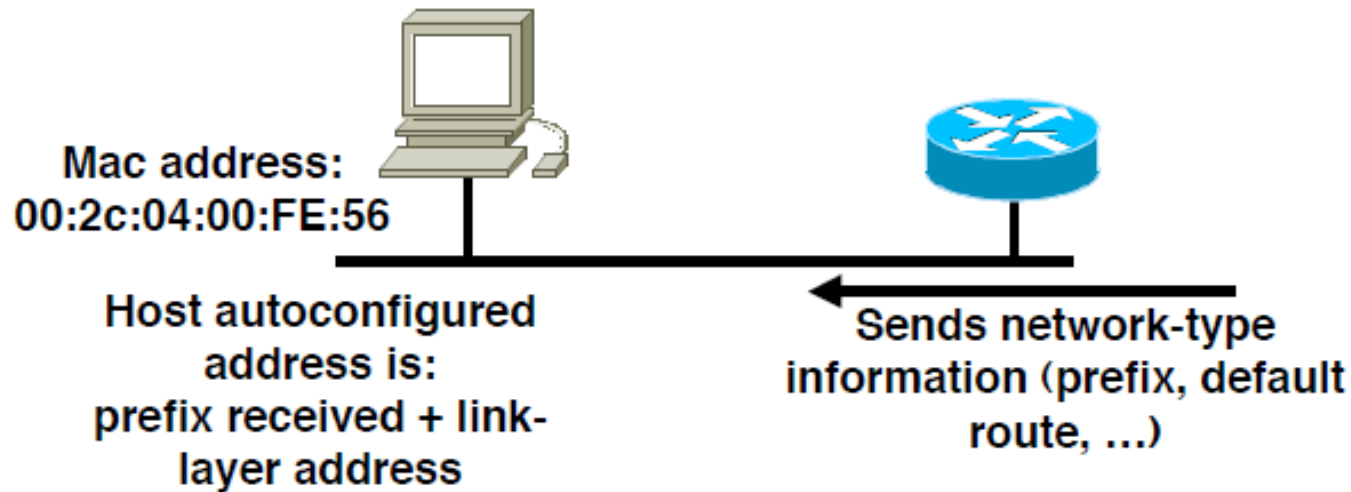


IPv6 Address Privacy (RFC 3041)



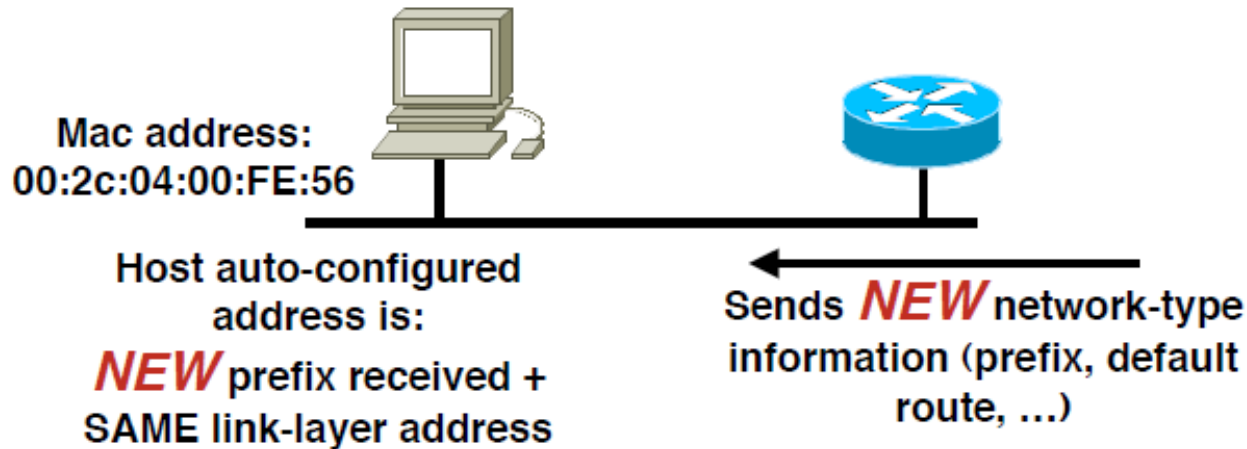
- Temporary addresses for IPv6 host client application, e.g. Web browser
- Intended to inhibit device/user tracking but is also a potential issue
 - More difficult to scan all IP addresses on a subnet
 - But port scan is identical when an address is known
- Random 64 bit interface ID, run DAD before using it
- Rate of change based on local policy
- Implemented on Microsoft Windows XP/vista/7 and apple MacOS 10.7 onwards
- Can be activated on linux with a system call

IPv6 Auto-configuration



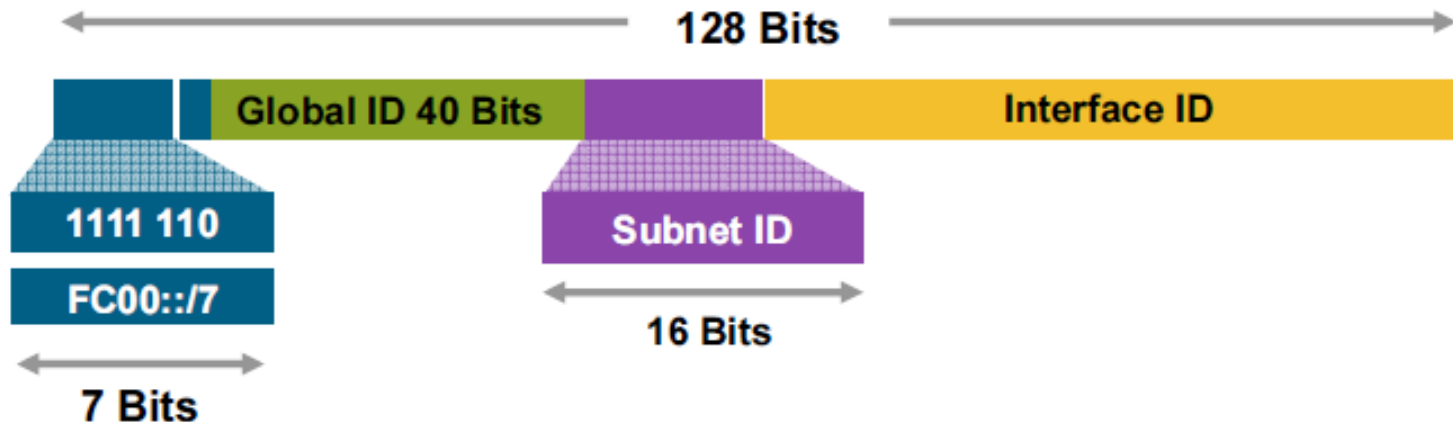
- Client sends router solicitation (RS) messages
- Router responds with router advertisement (RA)
 - This includes prefix and default route
- Client configures its IPv6 address by concatenating prefix received with its EUI-64 address

Renumbering



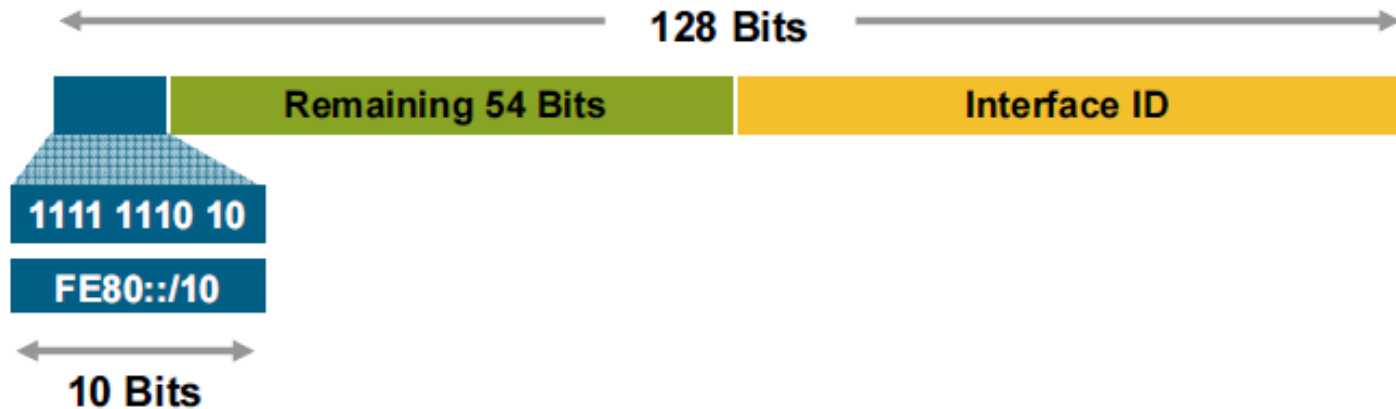
- Router sends router advertisement (RA)
 - This includes the new prefix and default route (and remaining lifetime of the old address)
- Client configures a new IPv6 address by concatenating prefix received with its EUI-64 address
 - Attaches lifetime to old address

Unique- Local



- Unique-Local Addresses Used For:
 - Local communications
 - Inter-site VPNs
 - Local device such as printers, telephone, etc
- Not routable on the Internet

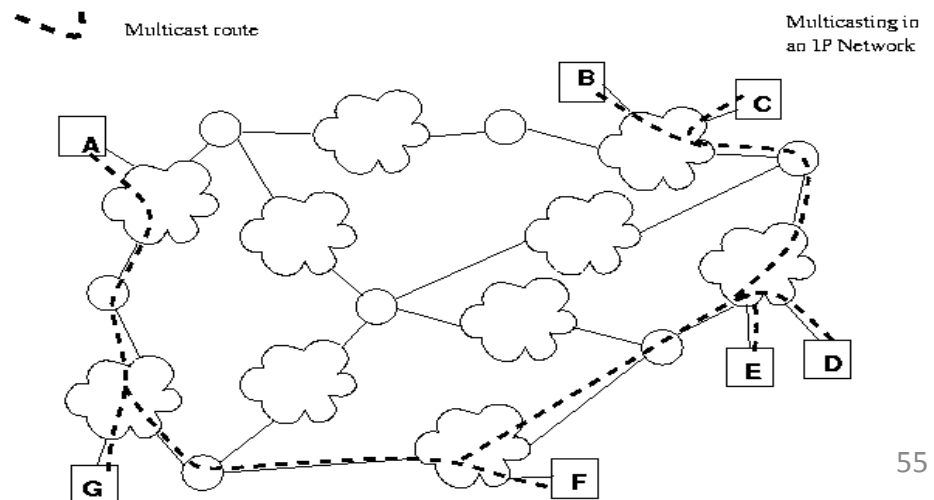
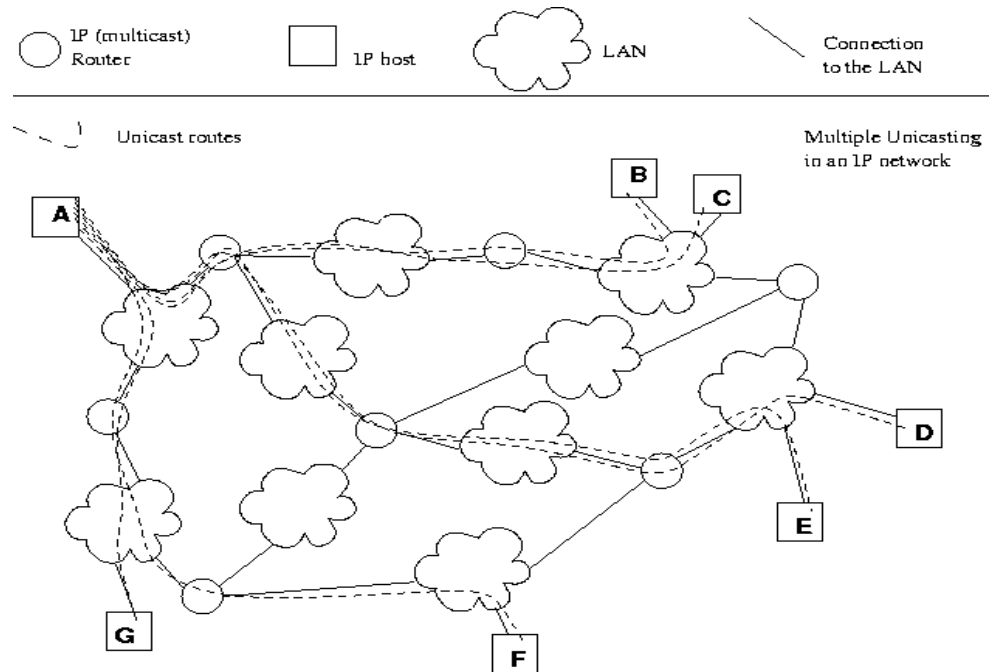
Link-Local



- Link-Local Addresses Used For:
 - Communication between two IPv6 device (like ARP but at Layer 3)
 - Next-Hop calculation in Routing Protocols
- Automatically assigned by Router as soon as IPv6 is enabled
 - Mandatory Address
- Only Link Specific scope
- Remaining 54 bits could be Zero or any manual configured value

Multicasting

- Multicast is communication between a single sender and multiple receivers on a network.



Generic Multicast Group Addresses



- IPv6 multicast addresses are in the range of **FF00::/8**
- Flag field:
 - 000T values
 - T = 0, for permanent addresses defined by IANA
 - T = 1, for transient addresses
- Scope field: Allows limiting the scope of the multicasting
 - 0 – Reserved
 - 1 – Node-local
 - 2 – Link-local
 - 3 – Subnet-local
 - 4 – Admin-local
 - 5 – Site-local
 - 8 – Organization-local
 - E – Global (Internet)

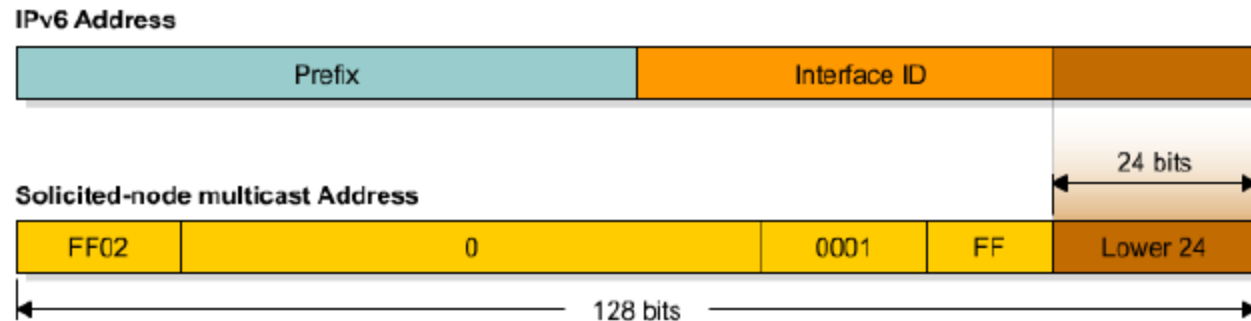
Generic Multicast Group Addresses : Examples

- IANA allocated addresses
 - <http://www.iana.org/assignments/ipv6-multicast-addresses>
- Flags = 0000
- Scope
- Group ID = 101 → NTP servers
 - FF01:0:0:0:0:0:0:101 : All NTP servers on the sender's host
 - FF02:0:0:0:0:0:0:101 : All NTP servers on the sender's link
 - FF05:0:0:0:0:0:0:101 : All NTP servers on the sender's site
 - FF0E:0:0:0:0:0:0:101 : All NTP servers on the Internet

IPv6 Multicast Address Assignments

- Addresses available only for a given scope
 - FF02::1 : All nodes of the link
 - FF02::2 : All routers of the link
 - FF05::1:3 : All DHCP servers of the site
 - FF02::D : All PIM routers of the link
 - ...
- Addresses available for all scopes
 - FF0X :: 101 : Network Time Protocol (NTP)
 - FF0X :: 109 : Multicast Transport Protocol (MTP)
 - ...

Solicited-Node Multicast Address



- For each unicast and anycast address configured there is a corresponding solicited-node multicast
- This is specially used for two purpose, for the replacement of ARP, and duplicate address detection
- Used in neighbor solicitation messages
- Solicited-node multicast consists of prefix + lower 24 bits from unicast or anycast IPV6 address

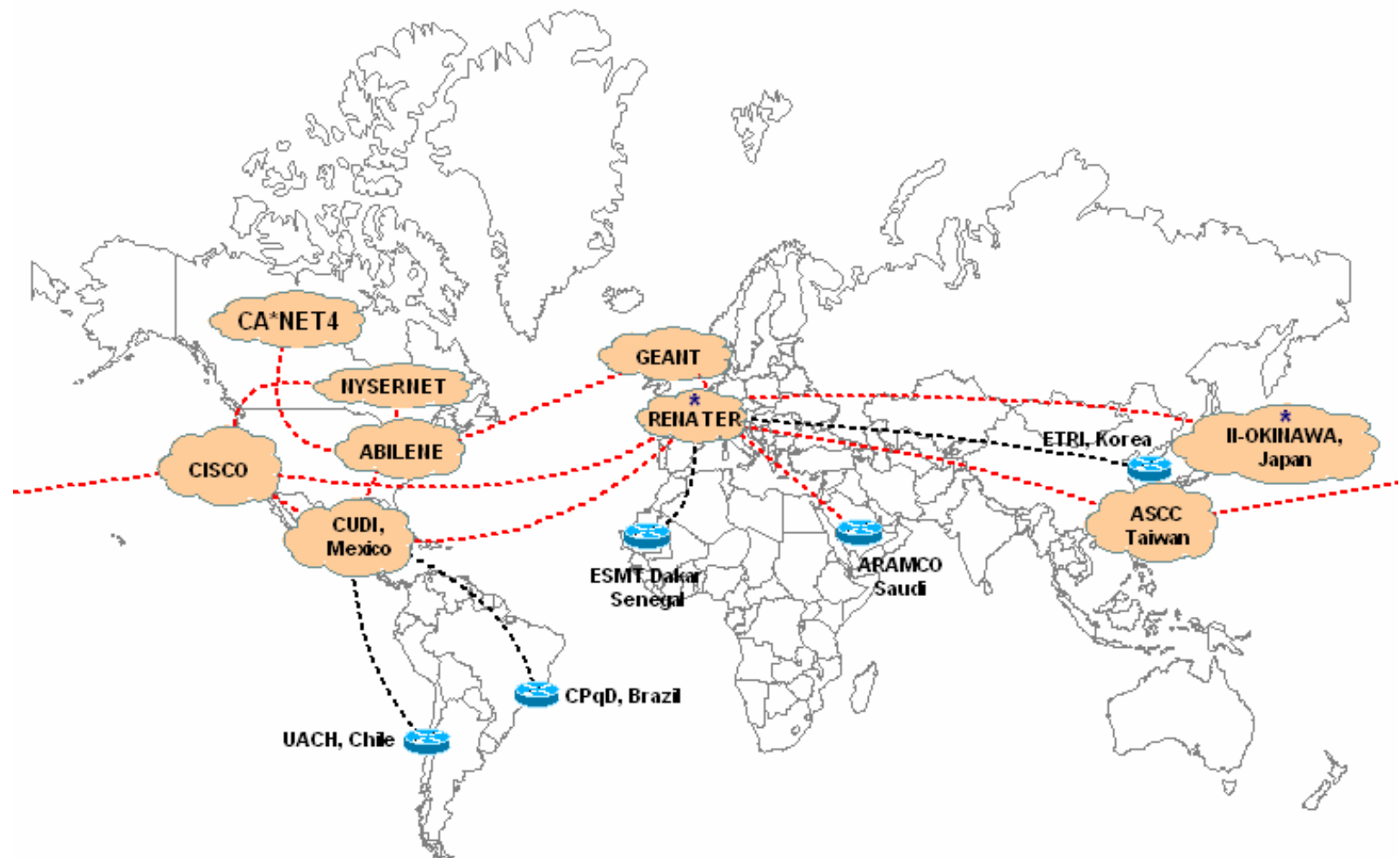
Example of Solicited-Node Multicast Address

- Concatenation of FF02::1:FFXX:XXXX with the last 24 bits of the IPv6 address
- IPv6 address:
 - ***2001:648:1a:4002:4421:21FF:FE24:87c1***
↓
 - ***Sol. Mcast address: FF02::1:FF24:87c1***
↓
 - ***Ethernet address: FF-FF-FF-24-87-c1***

Current IPv6 Multicast Deployment

- M6bone
 - An IPv6 Multicast test network with more than 30 networks & 60 sites connected
 - Started in July 2001 in France
- Aims to
 - offer IPv6 multicast connectivity,
 - test and develop soft and equipments related to IPv6 multicast technologies,
 - be active in IPv6 multicast standardization and
 - provide deployment recommendations

M6Bone



Comparison between IPv4 multicasting & IPv6 multicasting

- In IPv4, multicasting was extension of the basic specification, while specifications of IPv6 require that all IPv6 nodes support multicasting.
- IPv6 explicitly limits the scope of a multicast address by using a fixed address field, whereas the scope was specified using TTL (Time to Live) of a multicast packet in IPv4.
- In IPv4, multicast tunnels were introduced to deploy multicasting .In IPv6,all routers should be multicast-capable, which means that we do not have to use multicast tunnels to deploy IPv6 multicasting.
- IPv4 multicasting use unicast addresses to identify a network interface. However, this is not suitable for IPv6,as an IPv6-capable node may assign multiple addresses on a single interface, which tends to cause a configuration mismatch. In IPv6 ,to identify the interface the user must use specified interface index.

IPv6 Anycast

- An IPv6 anycast address is an identifier for a set of interfaces (typically belonging to different nodes)
 - A packet sent to anycast address is delivered to one of the interfaces identified by that address (the “nearest” one, according to the routing protocol’s measure of distance).
 - RFC4291 describes IPv6 Anycast in more detail
- In reality there is no known implementation of IPv6 Anycast as per the RFC

Anycast on the Internet

- A global unicast address is assigned to all nodes which need to respond to a service being offered
 - This address is routed as part of its parent address block
- The responding node is the one which is closest to the requesting node according to the routing protocol
 - Each anycast node looks identical to the other
- Applicable within an ASN, or globally across the Internet
- Typical (IPv4) examples today include:
 - Root DNS and ccTLD/gTLD nameservers
 - SMTP relays within ISP autonomous systems

MTU Issues

- Minimum link MTU for IPv6 is 1280 octets (versus 68 octets for IPv4)
 - on links with $MTU < 1280$, link-specific fragmentation and reassembly must be used
- Implementations are expected to perform path MTU discovery to send packets bigger than 1280
- Minimal implementation can omit PMTU discovery as long as all packets kept ≥ 1280 octets
- A Hop-by-Hop Option supports transmission of “jumbograms” with up to 2^{32} octets of payload

Fragmentation in IPv6

- IPv6 routers do not support fragmentation or the Don't Fragment option.
- For IPv6, Path MTU Discovery works by initially assuming the path MTU is the same as the MTU on the link layer interface through which the traffic is being sent.
- Then, similar to IPv4, any device along the path whose MTU is smaller than the packet will drop the packet and send back an ICMPv6 Packet Too Big (Type 2) message containing its MTU, allowing the source host to reduce its Path MTU appropriately.
- The process is repeated until the MTU is small enough to traverse the entire path without fragmentation

IPv6 Header Format

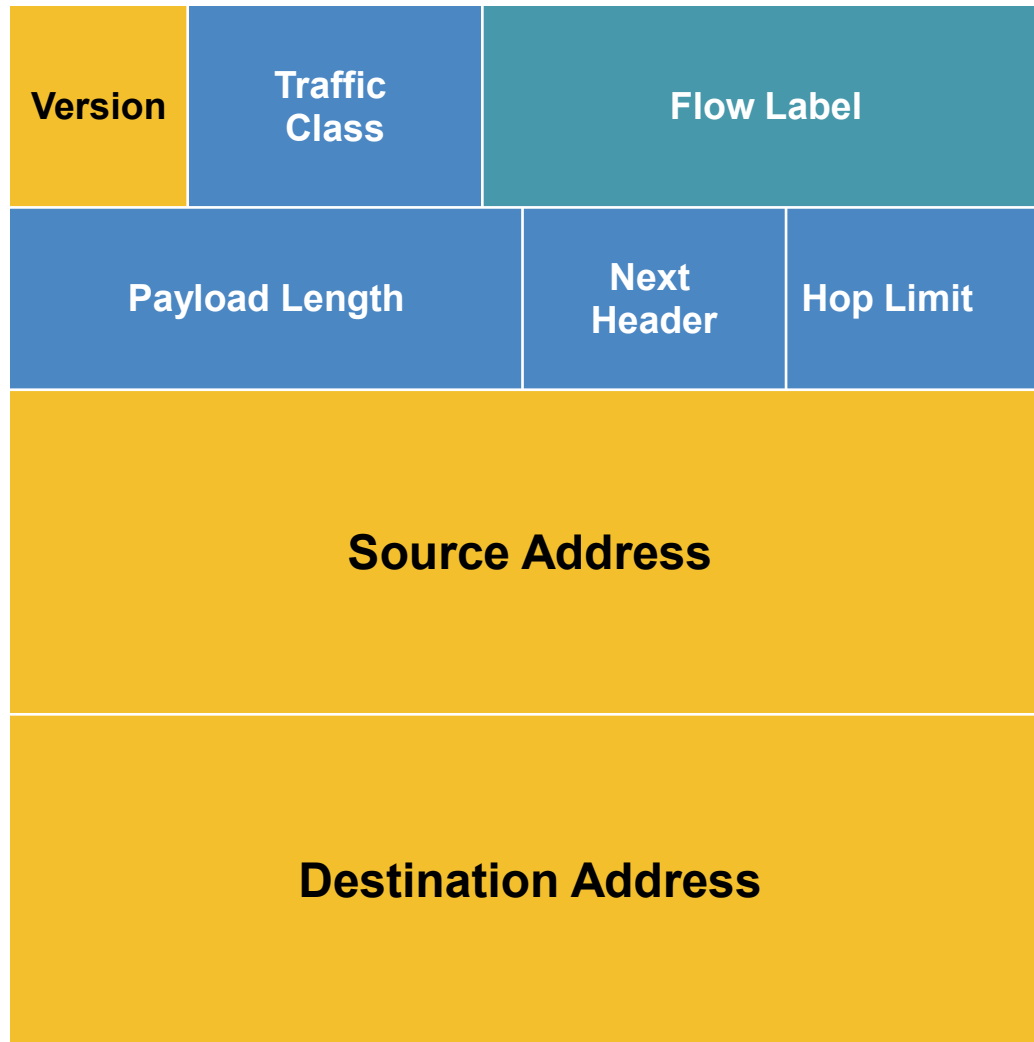


Fig : IPv6 Header

IPv6 Header Format contd..

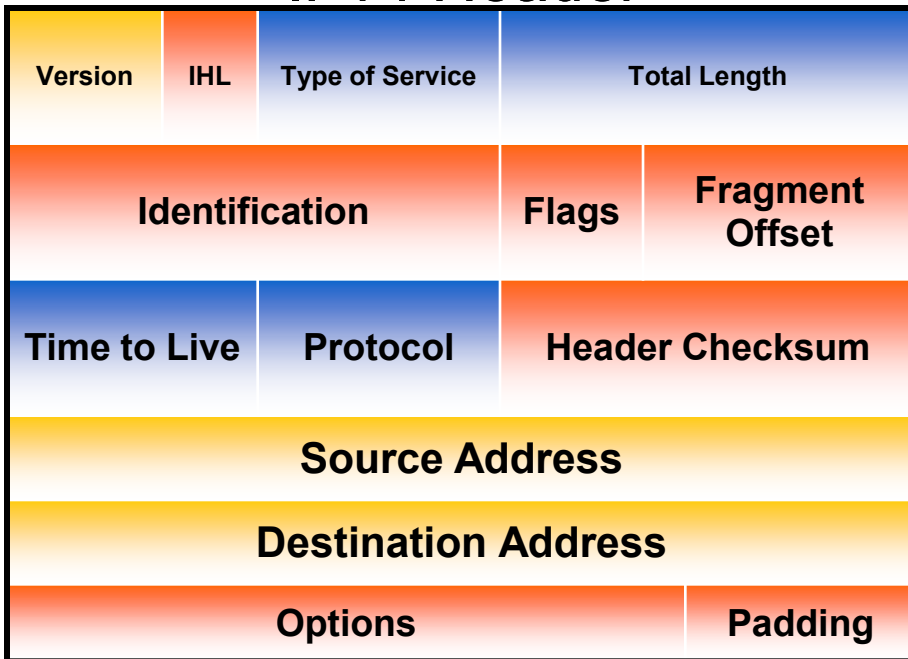
- Version (4 bits)
 - Identifies the version of IP protocol
- Traffic Class (8 bits)
 - Identify and distinguish between different classes or priorities of IPv6 packets.
- Flow Label (20 bits)
 - used to maintain the sequential flow of the packets
 - Used to “label” a flow of traffic.
 - helps avoid re-ordering of data packets.
 - designed for streaming/real-time media.
- Next Header (8 bits)
 - Identifies the “extension” header immediately following
 - Packet may have zero, one, or more extension headers

- Payload Length (16 bits)
 - Length, in octets, of the payload
 - Payload is composed of Extension Headers and Upper Layer data.
- Hop Limit (8 bits)
 - Maximum number of hops an IPv6 packet can be forwarded.
 - Decrement by each node on path
- Source Address (128 bits)
 - This field indicates the address of originator of the packet.
- Destination Address (128 bits)
 - This field provides the address of intended recipient of the packet .

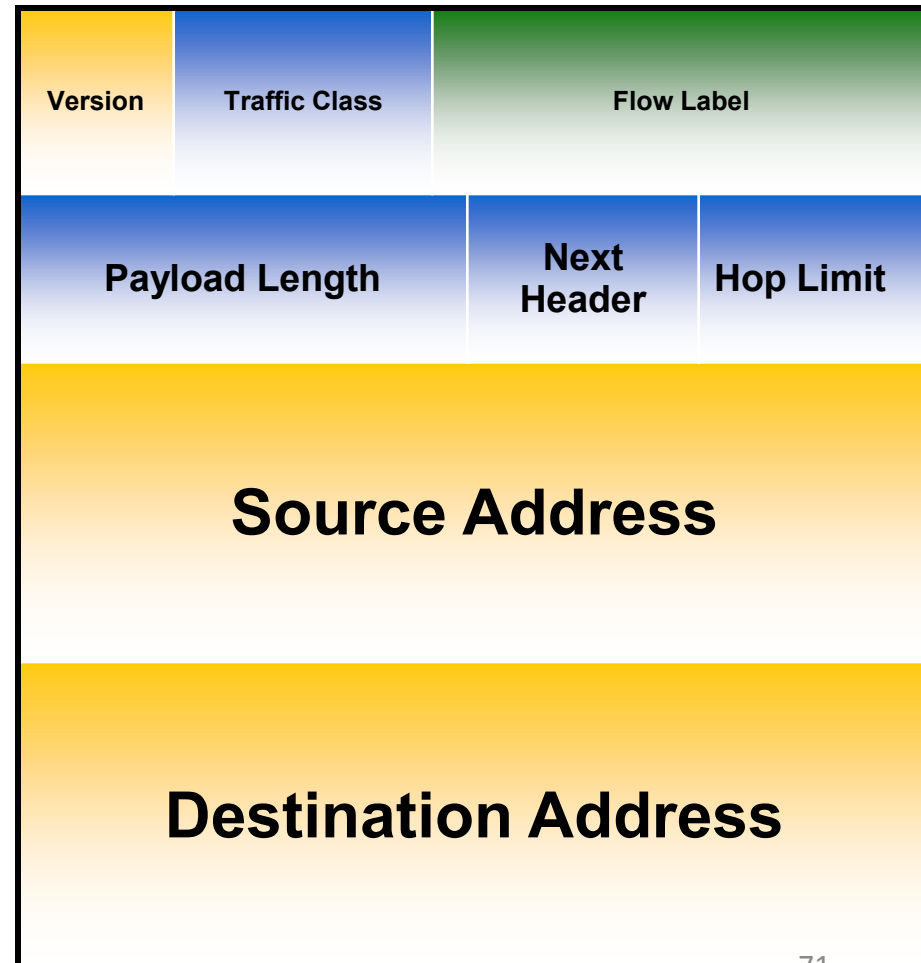
IPv4 & IPv6 Header Comparison





IPv4: 20 Bytes + Options IPv6: 40 Bytes + Extension Header

IPv4 Header



IPv6 Header

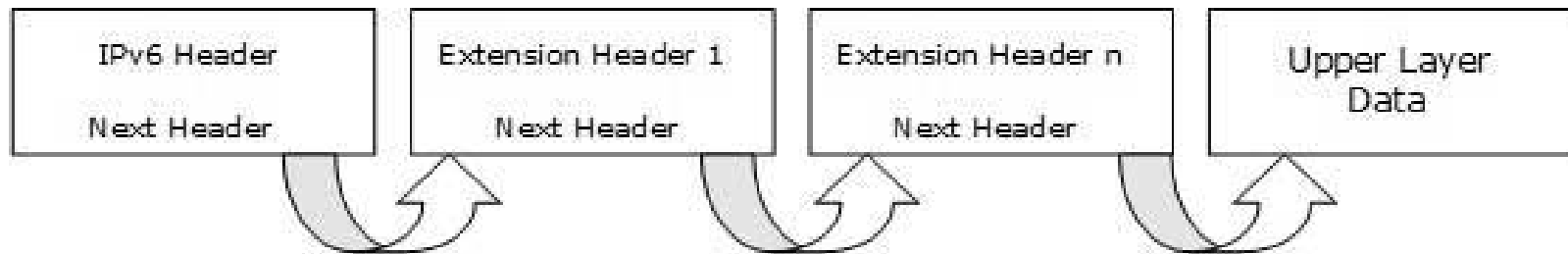


-  - field's name kept from IPv4 to IPv6
-  - fields not kept in IPv6
-  - Name & position changed in IPv6
-  - New field in IPv6

Extension Headers

- Fixed Header contains only necessary information
 - Avoiding information which is either not required or is rarely used.
 - All such information is put between the Fixed Header and the Upper layer header in the form of Extension Headers.
 - Each Extension Header is identified by a distinct value.
- When Extension Headers are used,
 - IPv6 Fixed Header's Next Header field points to the first Extension Header.
 - If there is one more Extension Header, then the first Extension Header's 'Next-Header' field points to the second one, and so on.
 - The last Extension Header's 'Next-Header' field points to the Upper Layer Header.

Extension Headers Contd..



IPv6 Header
Next Header
= TCP

TCP Header
+ Data

IPv6 Header
Next Header
= Routing

Routing Header
Next Header = TCP

TCP Header
+ Data

IPv6 Header
Next Header
= Routing

Routing Header
Next Header =
Fragment

Fragment Header
Next Header = TCP

Fragment of
TCP Header
+ Data

Extension Header Order

Order	Header Type	Next Header Code
1	Basic IPv6 Header	
2	Hop-by-Hop Options	0
4	Routing header	43
5	Fragment header	44
6	Authentication header	51
7	Encapsulation Security Payload header	50
8	Destination Options	60
9	Mobility header	135
	No Next header (Null)	59
	Upper layer: TCP, UDP, ICMP	6, 17, 58

- Hop-by-Hop Options
 - Must be first header extension
 - Examined by every node on a delivery path
- Routing Header
 - List of one or more intermediate nodes to visit
 - Not looked at by each node on path
- Fragment Header
 - IPv6 fragmentation & reassembling is an end-to-end function
 - Only the source node can fragment a packet in IPv6
- Authentication Header
 - for authentication/integrity only

- Encapsulating Security Payload
 - Provides Encryption security
 - Confidentiality
- Destination Options
 - Used to carry optional information for the Destination

Internet RFCs

- Request For Comments (RFC) is a memorandum published by the RFC editor on the behalf of IETF.
- Describes methods, research, or innovations applicable to the working of the internet and internet connected system.
- The “RFC” document series was originally created in 1969 by the research community that developed the ARPAnet and then the Internet.
- Begun by Steve Crocker (RFC 3) and Jon Postel.
- RFC 768 , published in August 1980, remains today the official Internet standard for the User Datagram Protocol, and is the oldest RFC with that status.

- Today, RFCs form the single series for all Internet protocol standards, recommendations, new ideas, procedures, etc.
- Every RFC is submitted in plain ASCII text and published in that form.
- RFCs are published as a numbered series and should be cited as such, much like journal articles.
- The Internet RFC series has an International Standard Serial Number (ISSN), namely 2070-1721.
 - It should be included in the citation whenever appropriate.
 - Most formats require that for on-line series, a URL be provided. The proper form for RFCs is:
 - <http://www.rfc-editor.org/rfc/rfc####.txt>

Sub-Series

- RFCs are numbered (roughly) sequentially.
- To identify significant subsets of RFCs, Postel invented “sub-series”. An RFC may have a subseries designator.
 - e.g., “RFC 2026, BCP 9”
- Sub-series designations:
 - BCP Best Current Practice status
 - STD Standard status
 - FYI User documentation (Informational)

Streams

- There are four streams of RFCs:
 - *IETF stream: Documents reviewed and approved within the Internet Engineering Task Force (IETF).*
 - *IAB stream: Documents reviewed and approved by the Internet Architecture Board (IAB).*
 - *IRTF stream: Documents reviewed and approved within the Internet Research Task Force (IRTF).*
 - *Independent Submissions: Documents submitted directly to, and reviewed under the authority of, the RFC Editor.*
 - The RFC Editor is the collective name for the team that carries out quality control, copyediting and publication of the RFC series.

Status

- Not all RFCs are standard
 - Each RFC is assigned a designation with regard to status
- Each RFC is static
 - If the document is changed or updated it is submitted again and assigned a new RFC number
 - If the RFC becomes an internet standard (STD) it is assigned an STD number but retains its RFC number
- RFC status may be:
 - Status “Informational”
 - Status “Experimental”
 - Status “Best current practice”
 - Status “historic”
 - Status “unknown”

- Status "informational"
 - An informational RFC can be nearly anything from April 1 jokes over proprietary protocols up to widely recognized essential RFCs like Domain Name System Structure and Delegation (RFC 1591).
 - Some informational RFCs formed the FYI sub-series.
- Status "experimental"
 - An experimental RFC can be an IETF document or an individual submission to the 'RFC Editor'.
 - Experimental RFCs may be promoted to standards track if it becomes popular and works well.

- Status "best current practice"
 - The best current practice (BCP) subseries collects administrative documents and other texts which are considered as official rules and not only informational, but which do not affect over the wire data.
 - The border between standards track and BCP is often unclear.
 - If a document only affects the Internet Standards Process, like BCP 9, or IETF administration, it is clearly a BCP.
 - If it only defines rules and regulations for Internet Assigned Numbers Authority (IANA) registries it is less clear; most of these documents are BCPs, but some are on the standards track.

- Status "historic"
 - A historic RFC is one that has been made obsolete by a newer version, documents a protocol that is not considered interesting in the current Internet, or has been removed from the standards track for other reasons.
 - Some obsolete RFCs are not classified as historic, because the Internet standards process generally does not allow normative references from a standards track RFC to another RFC with lower status.
- Status "unknown"
 - Status unknown is used for some very old RFCs, where it is unclear which status the document would get if it were published today.
 - Some of these RFCs would not be published at all today⁸⁴

Thank You

If you have any Queries write to me

@

Jalauddin.mansur@gmail.com

jalawdarling@hotmail.com