EECS, University of Ottawa

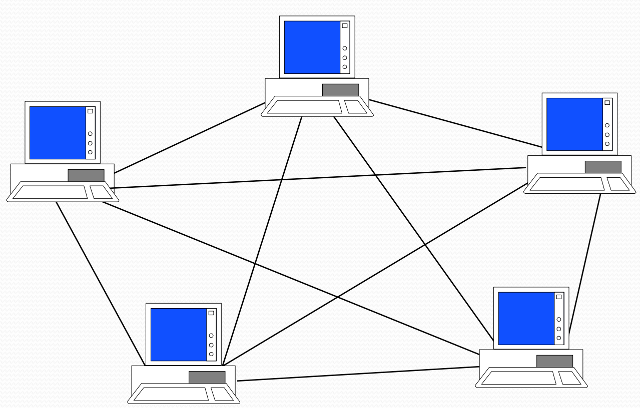
ELG5374 –Fall 2021
Computer Communication Network
Internetwork Protocols

Note: some material in the slides has been taken from various other sources 1-1

1

Computer Networks: Why?

Computing & communication devices need to exchange information



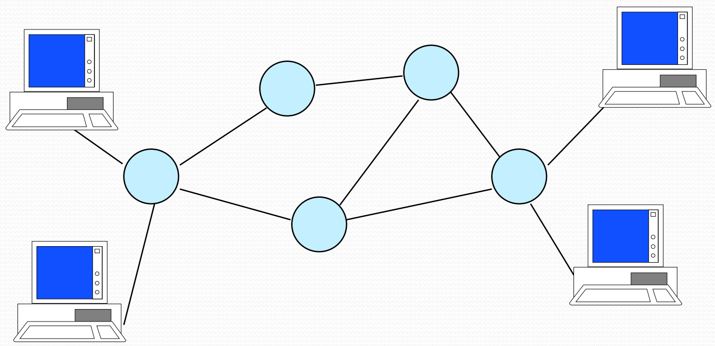
# of links required:
unidirectional links:
n (n-1)
bidirectional links:
n (n-1)/2
n: # of devices

1-2

2

Computer Networks

We need switching nodes



1-3

3

Open System Interconnection (OSI) Reference Model

- Developed by the International Organization for Standardization (ISO).
- Has become the standard model for classifying communication functions.
- Has seven layers.
- It is a "theoretical" system delivered too late!
- It has NOT dominated. TCP/IP is the de facto standard.
- Several reasons:
  - TCP/IP appeared earlier
  - Internet "won" the game
  - OSI has a "complex" structure that could result in "heavy processing"

1-4

4

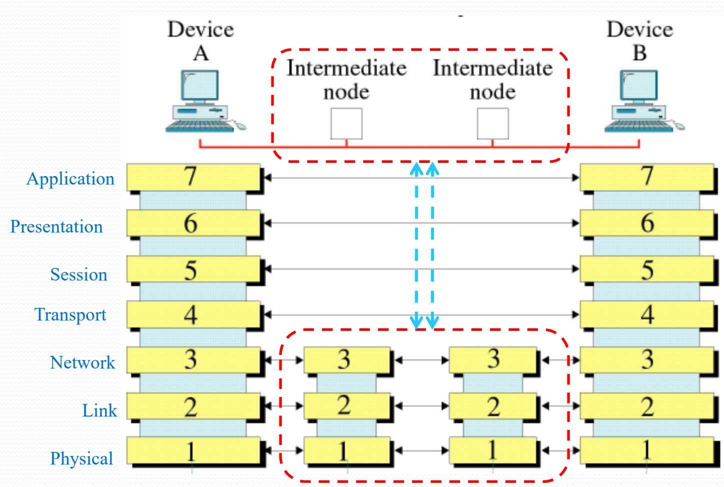
Open System Interconnection (OSI) Reference Model

Application	Access to the users (File transfer, e-mail, r-login, ..
Presentation	Data representation (syntax) (e.g ASCII)
Session	Control structure between applications. Establish/manage connection
Transport	Reliable, transparent transport of data between end-points. End-to-end error recovery and error control
Network	Responsible for establishing, maintaining, terminating connections (routing, addressing, congestion control,….)
Data link	Reliable transfer of information across physical link (sends "frames" of data with proper synchronization., error and flow control)
Physical	How to transmit a signal (access of the transmission medium; Copper, fiber, radio,..). Deals with network hardware, bit encoding)

1-5

5

OSI Reference Model



1-6

6

TCP/IP Protocol Architecture

- ***No official*** model but a working one.
- Has 5 layers (OSI has 7 layers)
- Funded by DARPA (USA).
- Initially developed as a US military research effort funded by the Department of Defense
- It has dominated.
- It is the "heart" of Internet.

1-7

7

TCP/IP Protocol Architecture (2)

Application Layer
Host-to-Host or Transport Layer
Internet Layer
Network Access Layer
Physical Layer

1-8

8

Physical Layer

- concerned with physical interface between computer and network
- concerned with issues like:
  - characteristics of transmission medium
  - signal levels
  - data rates
  - other related matters

1-9

9

Network Access Layer

- exchange of data between an end system and attached network
- concerned with issues like :
  - destination address provision
  - invoking specific services like priority
  - access to & routing data across a network link between two attached systems
- allows layers above to ignore link specifics

1-10

10

Internet Layer (IP)

- routing functions across multiple networks
- for systems attached to different networks
- using IP protocol
- implemented in end systems and routers
- routers connect two networks and relay data between them

1-11

11

Host-to-host / Transport Layer

- common layer shared by all applications
- provides reliable delivery of data
- in same order as sent
- commonly uses TCP

1-12

12

Application Layer

- Provides support for user applications
- Needs a separate module for each type of application

1-13

13

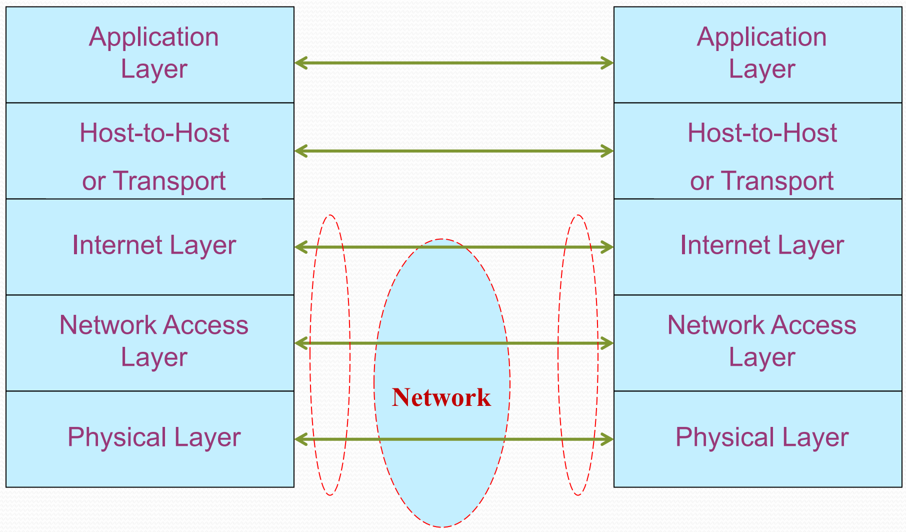
TCP/IP Protocol Architecture

Application Layer	Contains the logic needed to support user applications (ftp, telnet, http etc.) Each application requires different module.
Host-to-Host or Transport	Concerned with the reliability of transmission/reception (error control, sequencing, flow control)
Internet Layer	Provides routing functions across multiple networks. It is implemented in end-systems and routers
Network Access Layer	Concerned with the exchange of data between communicating entities. Depends on network type.
Physical Layer	Covers the physical interface between device (computer and transmission medium or network - medium, signals, data rates..)

1-14

14

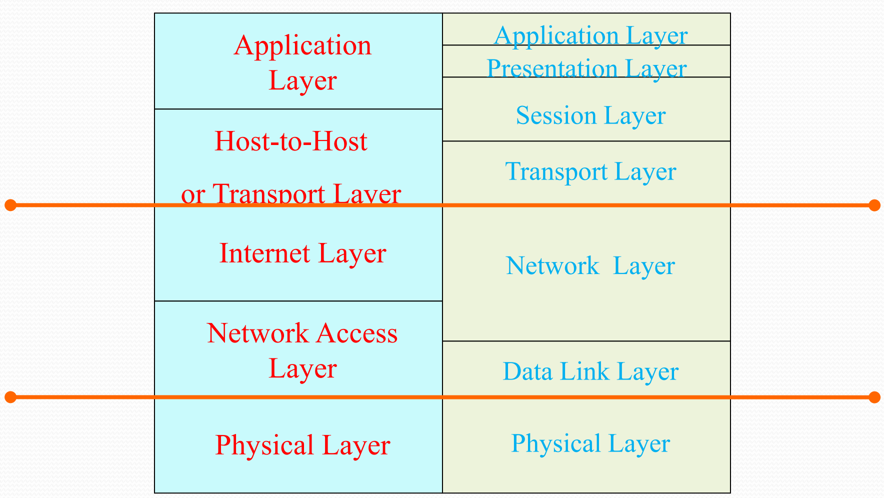
TCP/IP Protocol Architecture



1-15

15

TCP/IP versus OSI



1-16

16

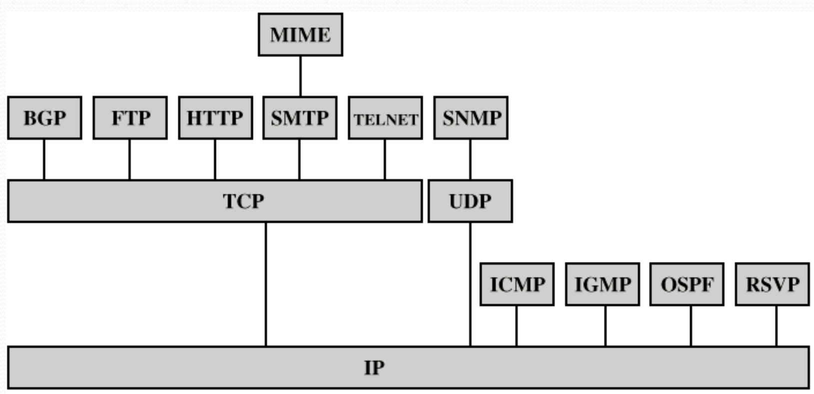
OSI Pros and Cons

- Bad timing (too much detailed concept before actual applications)
  - It tries to design the “perfect world”, which is either difficult or impractical.
  - Technology and human understanding of how things work (or should work) changes.
- More modular but more processing intensive.
- Provides a good architecture for detailed modeling of processes

1-17

17

Some TCP/IP Protocols



BGP = Border Gateway Protocol
FTP = File Transfer Protocol
HTTP = Hypertext Transfer Protocol
ICMP = Internet Control Message Protocol
IGMP = Internet Group Management Protocol
IP = Internet Protocol
MIME = Multi-Purpose Internet Mail Extension
OSPF = Open Shortest Path First
RSVP = Resource ReSerVation Protocol
SMTP = Simple Mail Transfer Protocol
SNMP = Simple Network Management Protocol
TCP = Transmission Control Protocol
UDP = User Datagram Protocol

1-18

18

Internet Protocol (IP) Versions

- IPv1-3 defined and replaced
- IPv4 – 1st widely deployed version (incl. commercial internet)
- IPv5 - stream protocol
  - a **connection-oriented** internet-layer protocol
  - MPLS provides connection-oriented capability.
- IPv6 - replacement for IP v4
  - During development it was called IPng (IP next generation)

1-19

19

IPv4

1-20

20

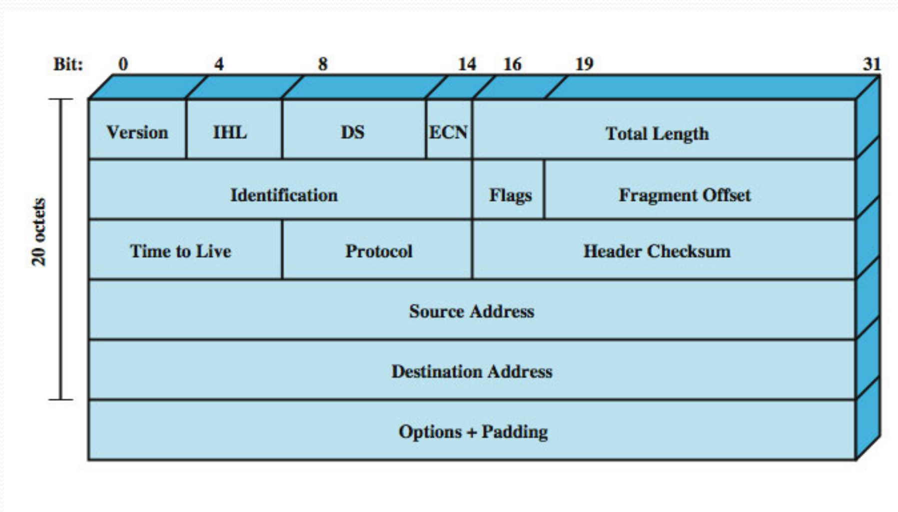
Internet Protocol (IP) v4

- defined in RFC 791
- part of TCP/IP suite
- two parts
  - specification of interface with a higher layer (e.g. TCP)
  - specification of actual protocol format and mechanisms
- is gradually replaced by IPv6

1-21

21

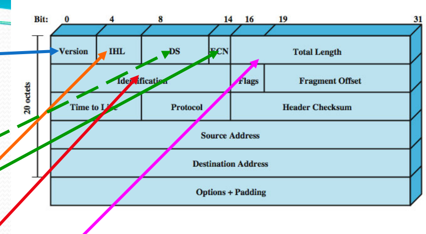
IPv4 Header



1-22

22

Header Fields (1)

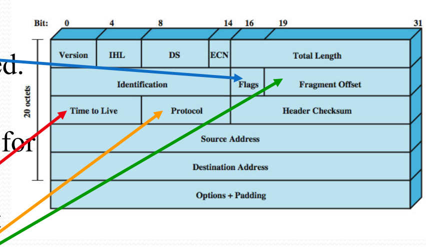


- Version
  - earlier IPv4
  - latter IPv6
- Internet Header Length
  - multiple of 32-bit words (including options)
- DS/ECN (was type of service)
- Total length of datagram (in octets)
- Identification
  - sequence number that together with the source address, destination address, and user protocol, identifies a datagram uniquely

1-23

23

Header Fields (2)

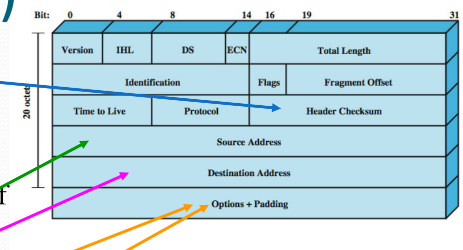


- Flags
  - Only **2** of the bits are currently defined.
  - "More" bit indicates there are more segments of the packet coming (used for fragmentation and reassembly)
  - "Don't fragment" bit prohibits packet fragmentation when set.
- Fragmentation offset
  - Indicates where in the original datagram this fragment belongs.
- Time to live
  - Specifies how long a datagram is allowed to remain in the internet.
- Protocol
  - Next higher layer to receive data field at destination (e.g. TCP)

1-24

24

Header Fields (3)



- Header checksum
  - re-verified and recomputed at each router
  - 16 bit ones complement sum of all 16 bit words in header
- Source address
- Destination address
- Options
  - Encodes the options requested by the sending user
- Padding
  - to fill to multiple of 32 bits

1-25

25

IP Options

- Security
  - Allows a security label to be attached to a datagram
- Source routing
  - A sequenced list of router addresses that specifies the route to be followed.
    - 1) Strict (only identified routers may be visited)
    - 2) Loose (other intermediate routers may be visited).
- Route recording
  - A field is allocated to record the sequence of routers visited by the datagram
- Stream identification
  - Names reserved resources used for stream service.
  - This service provides special handling for volatile periodic traffic (e.g., voice).
- Timestamping
  - The source IP entity and (some or all) intermediate routers add a timestamp (precision to milliseconds) to the data unit as it goes by.

1-26

26

Data Field

- carries user data from next layer up
- integer multiple of 8 bits long (octet)
- max length of datagram (header plus data) is 65,535 octets

1-27

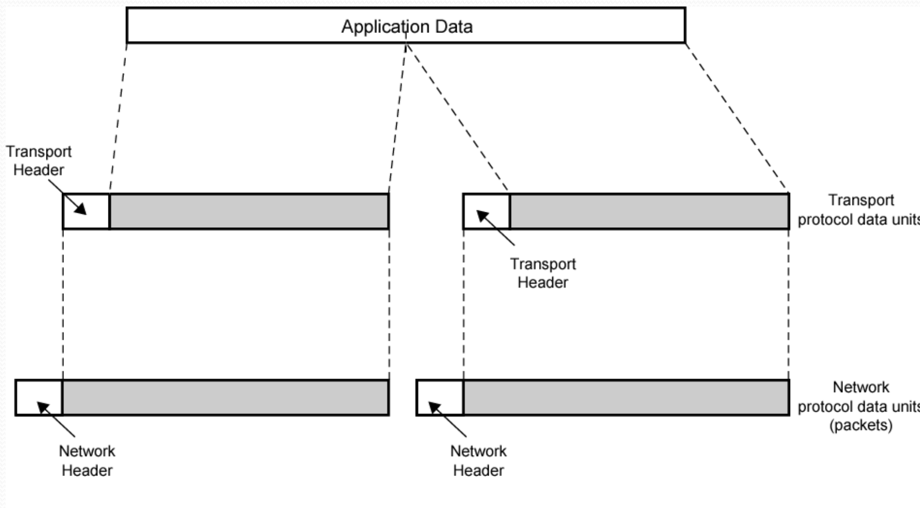
27

Fragmentation and Reassembly (F&R)

1-28

28

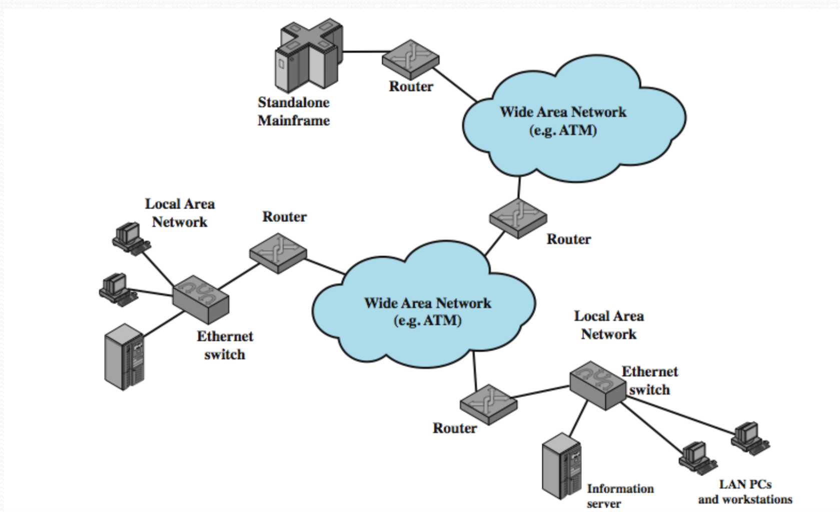
PDUS and Fragmentation



1-29

29

Internet Elements

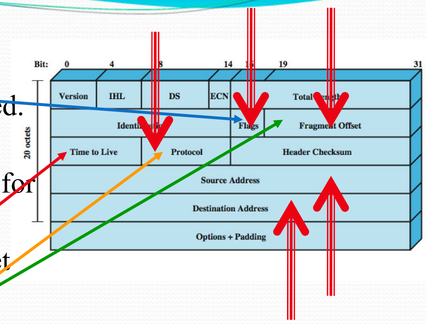


1-30

30

Header Fields (2)

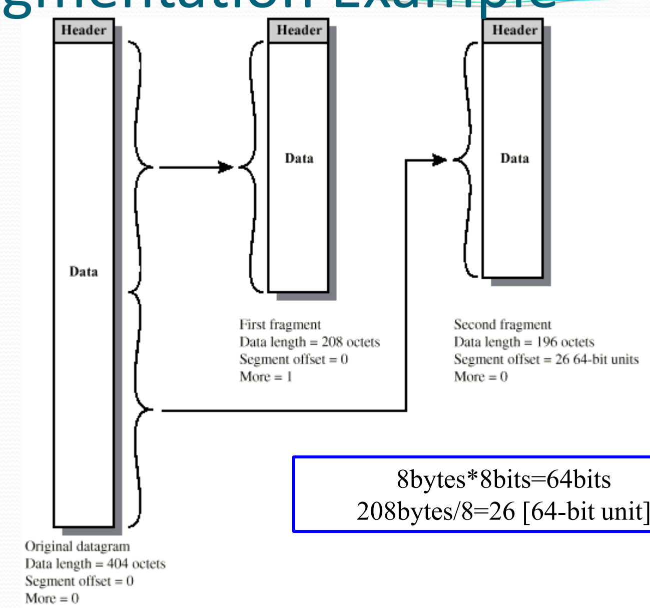
- Flags
  - Only 2 of the bits are currently defined.
  - *"More"* bit indicates there are more segments of the packet coming (used for fragmentation and reassembly)
  - *"Don't fragment"* bit prohibits packet fragmentation when set.
- ***Fragmentation offset***
  - Indicates where in the original datagram this fragment belongs.
- Time to live
  - Specifies how long a datagram can remain in the internet.
- Protocol
  - Next higher layer to receive data field at destination (e.g. TCP)



1-31

31

Fragmentation Example



8bytes*8bits=64bits
208bytes/8=26 [64-bit unit]

1-32

32

Fragmentation and Reassembly: Why?

- Protocol exchanges data between two entities
- Lower-level protocols may need to break data up into smaller blocks, the action called fragmentation
- Why fragmentation? For various reasons
  - network only accepts blocks of a certain size, or it has a minimum and maximum limit for the allowed size of data blocks (e.g.
    - ATM: 53 bytes cell size (48 payload + 5 control)
    - Ethernet frames: minimum size = 72 bytes; maximum size = 1526 bytes
  - more efficient error control & smaller retransmission units
  - fairer access to shared facilities
  - Less waiting times of packets of higher priority in queues
  - smaller buffers
- Disadvantages
  - more bandwidth wasted in overhead related data
  - more interrupts & processing time

1-33

33

F&R: Why and how?

- Problem:
  - Different networks have different Maximum Transmission Unit (MTU) sizes.
  - Within Internet, a data block might pass through several different networks before reaching the destination.
  - What if a packet reaches a network and it exceeds the network's MTU?
- Solution:
  - Use *fragmentation* to split large packets into smaller ones.
  - Use *reassembly* at the destination and/or intermediate nodes to put the fragments together and build the original packet.

1-34

34

Fragmentation and Re-assembly

- Where to re-assemble?
  - At destination
    - Results in packets getting smaller as data traverses internet
  - Intermediate re-assembly
    - Need large buffers at routers
    - Buffers may fill with fragments
    - All fragments must go through same router
      - Inhibits dynamic routing

1-35

35

IP Fragmentation

- IP re-assembles at destination only!
- Uses fields in header
  - Data Unit Identifier (ID)
    - Identifies end system originated datagram using:
      - Source and destination address
      - Protocol layer generating data (e.g. TCP)
      - Identification supplied by that layer
  - Data length
    - Length of user data in octets
  - Offset
    - Position of fragment of user data in original datagram
    - In multiples of 64 bits (8 octets)
  - *More* flag
    - Indicates that this is not the last fragment

1-36

36

IPv6

1-37

37

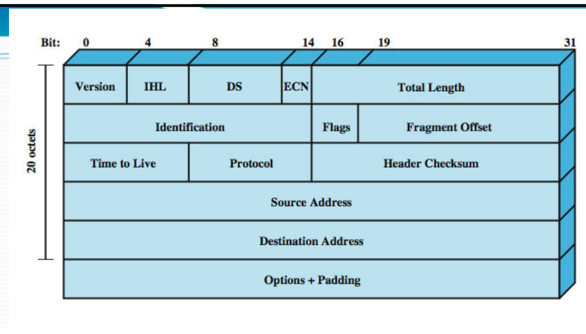
Why Change IP?

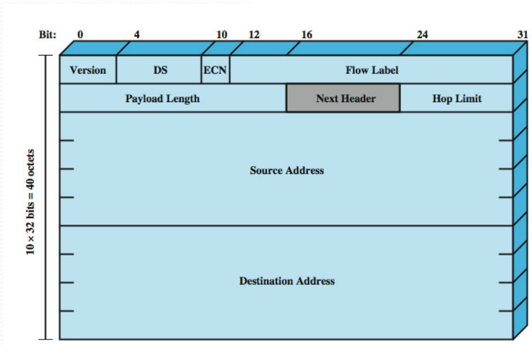
- Address space exhaustion
  - two-level addressing (network and host) wastes space
  - network addresses used even if not connected
  - you have to wait for a while, to get more info, in order to understand these two statements
  - growth of networks and the Internet
  - extended use of TCP/IP
  - single address per host
- requirements for new types of service

1-38

38

Compare IPV6 & IPv4

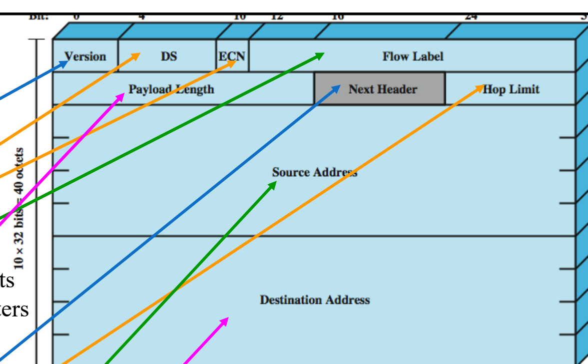




1-39

39

IPv6 Header



Version (4 bits): IP version number (=6).
DS/ECN (8 bits): As in IPv4.
Flow Label (20 bits): Labels packets that require special handling by routers within a network.

Payload Length (16 bits): Length of remainder of packet following header (extension headers + transport-level PDU)
Next Header (8 bits): Identifies type of header immediately following IPv6 header (either an IPv6 extension header or a higher-layer header (e.g. TCP or UDP).
Hop Limit (8 bits): Remaining number of allowable hops for this packet.
Source Address (128 bits): address of originator of the packet.
Destination Address (128 bits): address of intended recipient of the packet.
IPV6 header:
•longer than the mandatory portion of IPv4 header (40 octets versus 20 octets)
•contains fewer fields (8 versus 12).
•Routers have less processing to do per header (speeds up routing).

1-40

40

IPv6 Flow Label

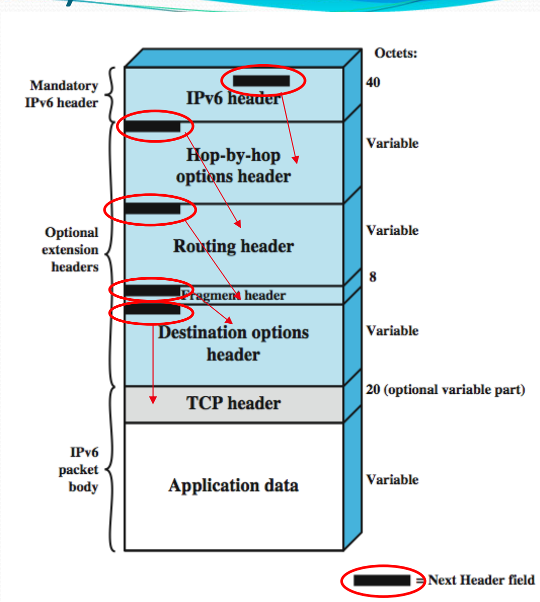
- **Flow:** a sequence of packets sent from a particular source to a particular destination for which the source desires special handling by the intervening routers.
- Flow is identified by src addr & dest addr + flow label
- router treats flow as sharing attributes
  - e.g. path, resource allocation, discard requirements, accounting, security
- may treat flows differently
  - buffer sizes, different forwarding precedence, different quality of service
- alternative to including **<u>all info</u>** in **<u>every header</u>**
- Router has requirements on flow label processing

1-41

41

IPv6 PDU (Packet) Structure

Next header field:
Identifies type of the immediately following header.
--If extension header, field contains header type identifier of that header.
-- Otherwise, field contains protocol identifier of upper-layer protocol (e.g. TCP.

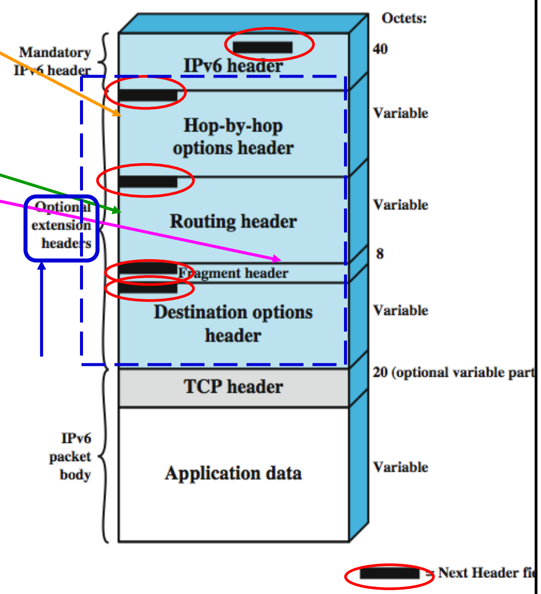


1-42

42

IPv6 PDU Structure: Extension headers

Hop-by-Hop Options header: Defines special options that require hop-by-hop processing.
Routing header: Extended routing, similar to IPv4 source routing.
•**Fragment header:** Fragmentation and reassembly information.
Authentication header: Provides packet integrity and authentication.
Encapsulating Security Payload header: Provides privacy.
Destination Options header: Contains optional information to be examined by the destination node.



IPv6 Enhancements (1)

- Expanded IP address: 128 bit address space
  - increase of address space by a factor of 2^{96}
  - allows (on the order of) 6×10^{23} unique addresses per square meter of the surface of the earth, which seems inexhaustible.
- Improved (flexible) option mechanism
  - options are placed in separate optional headers (between IPv6 header & transport- layer header).
  - most optional headers are not examined/processed by any internet router on the packet's path.
  - simplifies and speeds up IPv6 (vs. IPv4) packet routing processing.
  - Easier to add additional options.

IPv6 Enhancements (2)

- dynamic address assignment (using address auto-configuration)
- Increased addressing flexibility
  - includes anycast & multicast
  - anycast: packet is delivered to just one of a set of nodes.
  - scalability of multicast routing is improved by adding scope field to multicast addresses.
- Support for resource allocation
  - labeled packet flows
  - distinguishes different flows coming from the same (IP address) source (e.g. can identify a Video over IP or Voice over IP session (having real-time constraints) from a file transfer or web browsing session (which are fine with best effort treatment).

1-45

45

IPv6 Addresses

- 128 bits long
- assigned to interface
- single interface may have multiple unicast addresses
- three types of addresses:
  - unicast - single interface address
  - anycast - one of a set of interface addresses
  - multicast - all of a set of interfaces

1-46

46

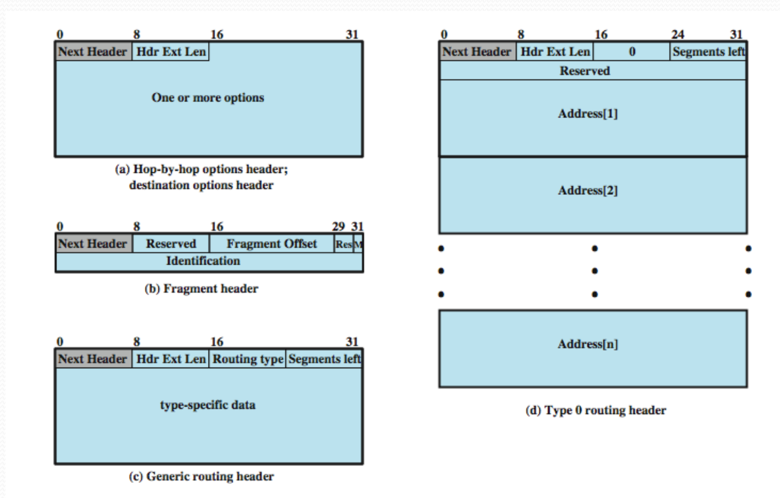
For reference: IPv6 RFCs

- RFC 1752 - Recommendations for the IP Next Generation Protocol
  - requirements
  - PDU formats
  - addressing, routing security issues
- RFC 2460 - overall specification
- RFC 2373 - addressing structure
- Plenty of additional material

1-47

47

IPv6 Extension Headers



(a) Hop-by-hop options header; destination options header

(b) Fragment header

(c) Generic routing header

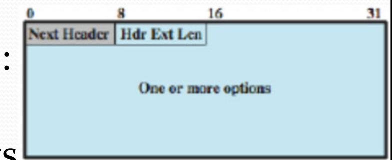
(d) Type 0 routing header

1-48

48

Hop-by-Hop Options/1

- Must be examined by every router
  - if unknown discard/forward handling is specified
- Next header (8 bits)
- Header extension length (8 bits):
  - length = #x64-bit unit
- Options: variable length; consists "definitions"



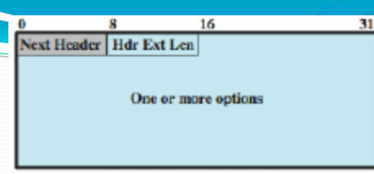
(a) Hop-by-hop options header; destination options header

- Each definition is formed by 3 subfields:
  - option type (8 bits); identifies the option
  - length (8 bits); specifies the length of option's data field in bytes.
  - option data; has variable length

1-49

49

Hop-by-Hop Options/2



(a) Hop-by-hop options header; destination options header

- Must be examined by every router
  - if unknown discard/forward handling is specified
- Next header (8 bis)
- Header extension length (8 bits):
  - length = #x64-bit unit
- Options: variable length; consists of 1 or more "definitions"
- Each definition is formed by 3 subfields:
  - option type (8 bits); identifies the option
    - 5 least significant bits specify type
    - 2 most significant bits specify treatment:
      - "00" – skip this option and continue processing header
      - "01" – discard packet
      - "10" – discard packet and send "ICMP parameter problem" message to source pointing to unrecognized option type
      - "11" – discard packet and proceed sending message to source as indicated above only if the destinations address in not multicast address.
    - 3rd most significant bit:
      - if "0" option data field should not be changed by any node;
      - if "1" can be changed.
  - length (8 bits); specifies the length of option's data field in bytes.
  - option data; has variable length

1-50

50

Hop-by-Hop Options/3

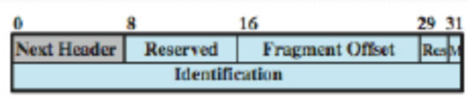
- Four (4) hop-by-hop options have been specified:
  - Pad1: used to insert 1 byte of padding into "options area of header
  - PadN: used to insert N bytes of padding into "options area of header (N>1)
  - Padding options ensure option is multiple of 8 bytes in lenght
  - Jumbo payload: used to indicate sending packet with payload longer than 65,535 bytes.
    - Option data field is 32 bits; specifies packet in octets (excluding IPv6 header).
    - Payload field of IPv6 header is set to zero; no "fragment header" can be placed
  - Router alert: informs router that contents of packet is of interest to router (e.g. passing of control data, e.g. traffic control).
    - Provides efficient support for variety of protocols (e.g. RSVP).
    - If "router alert" absent, packet is routed without further parsing.

1-51

51

Fragmentation Header/1

- fragmentation only allowed at source
- no fragmentation at intermediate routers
- node must perform path discovery to find smallest MTU of intermediate networks
- set source fragments to match MTU
- otherwise limit to 1280 octets
- header includes
  - fragment offset
  - more fragments bit
  - identification



(b) Fragment header

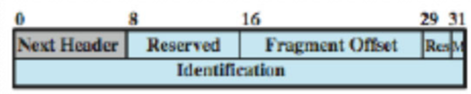
1-52

52

Fragmentation Header/2

Fragment header consists of:

- Next header (8 bits): identifies type of immediately following header.
- Reserved (8 bits): for future use.
- Fragment offset (13 bits): indicates location of packet's data within original payload.
- Res (2 bits): reserved for future use.
- M flag (1 bit): "1"→ more fragments; "0"→ last fragment
- Identification (32 bits): uniquely identifies original packet.

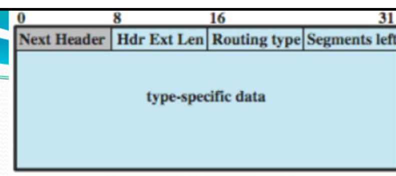


(b) Fragment header

1-53

53

Routing Header/1



(c) Generic routing header

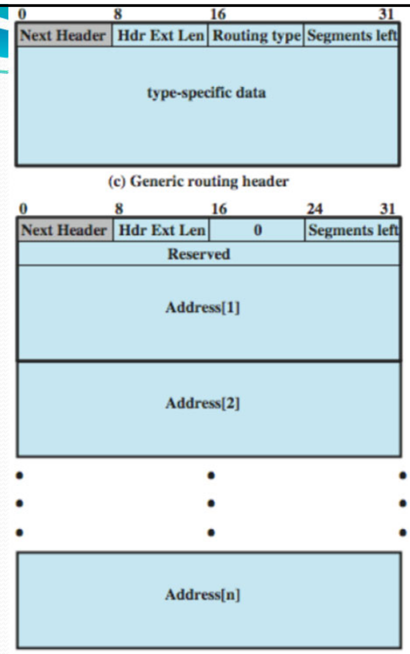
- Contains list of one or more intermediate nodes to visit. It includes:
  - Next Header: identifies type of next header
  - Header extension length: measured in 64-bit units (not including first 64 bits).
  - Routing type: identifies a particular routing header variant.
    - If variant unknown to router, router drops packet.
  - Segments left: # of route segments remaining (# of explicitly listed intermediate nodes still to be visited).

1-54

54

Routing Header/2

- Only defined routing header: “Type 0” routing header (RFC 2460).
  - Source place address of first router to be visited in IPv6 header (not destination’s address).
  - First address on list: IPv6 address of second router to be visited, and goes on….
  - Last address on list: IPv6 address of destination.



(c) Generic routing header

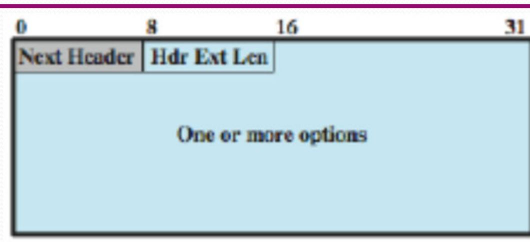
(d) Type 0 routing header

1-55

55

Destination Options Header

- carries optional info for (and examined only by) destination node
- format same as hop-by-hop header

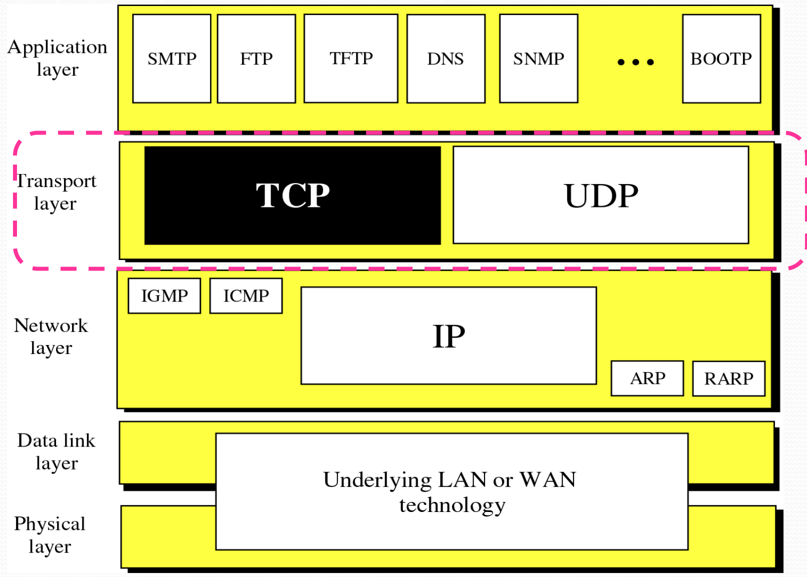


(a) Hop-by-hop options header; destination options header

1-56

56

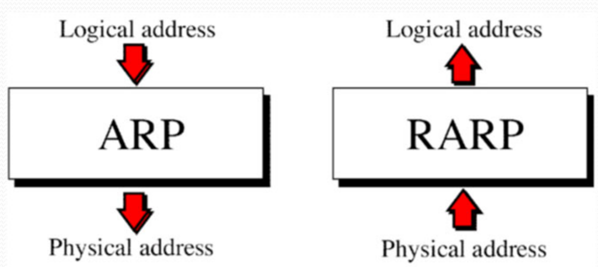
TCP/IP Protocol Suite



1-57

57

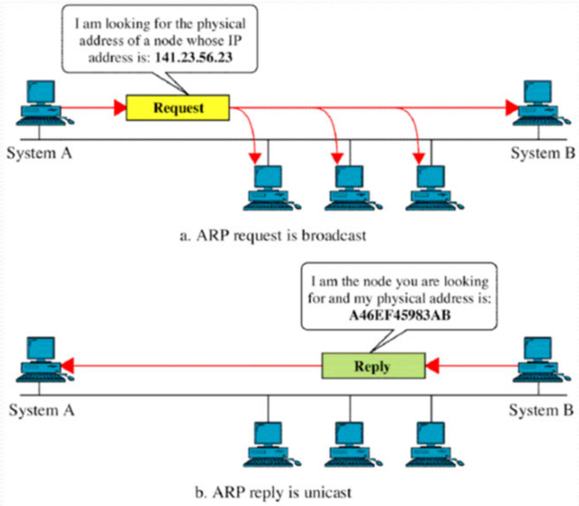
Address Mapping



1-58

58

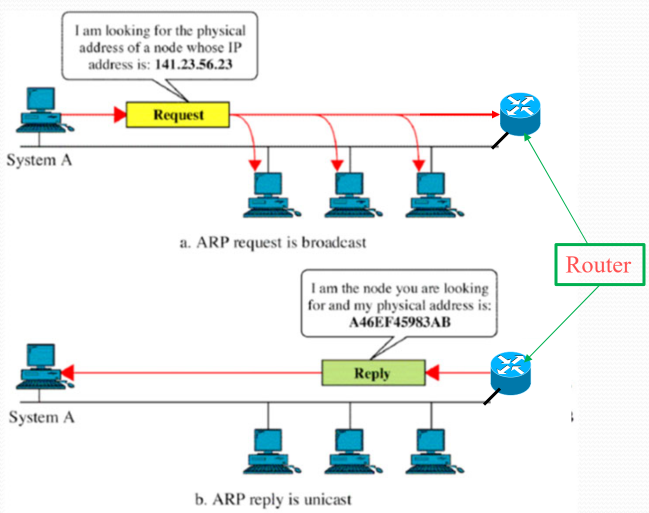
ARP (Address Resolution Protocol)



1-59

59

ARP (Address Resolution Protocol)



1-60

60

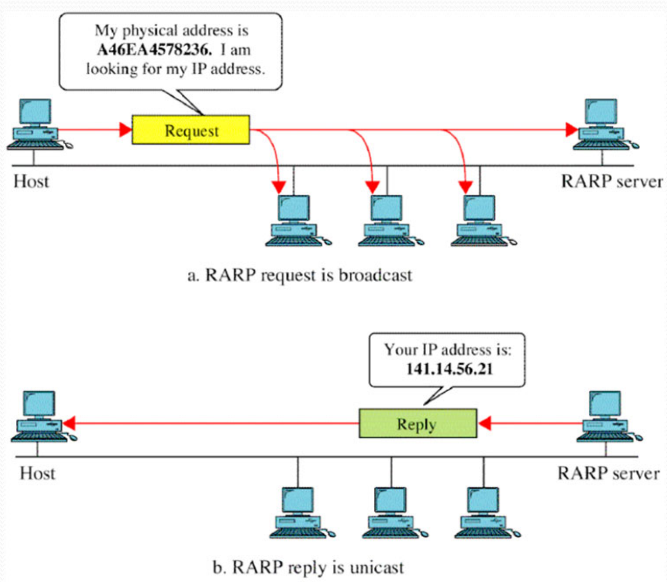
Address Resolution Protocol (ARP)

- Basic idea
  - host-1 wants to send a message to host-2 (in the same Ethernet); host-1 knows about the IP address, IP-a, of host-2
  - host-1 outputs a broadcast packet onto the Ethernet asking "who owns IP-a?"
  - every machine will receive this packet and only host-2 (that owns IP-a) will respond with its Ethernet address
- ARP messages
  - request from source asking for hardware address
  - reply from destination carrying hardware address

1-61

61

Reverse ARP



1-62

62

Internet Control Message Protocol (ICMP)

- RFC 792
- transfer of (control) messages from routers and hosts to hosts
- feedback about problems
  - e.g. time to live expired
- encapsulated in IP datagram
  - hence not reliable

1-63

63

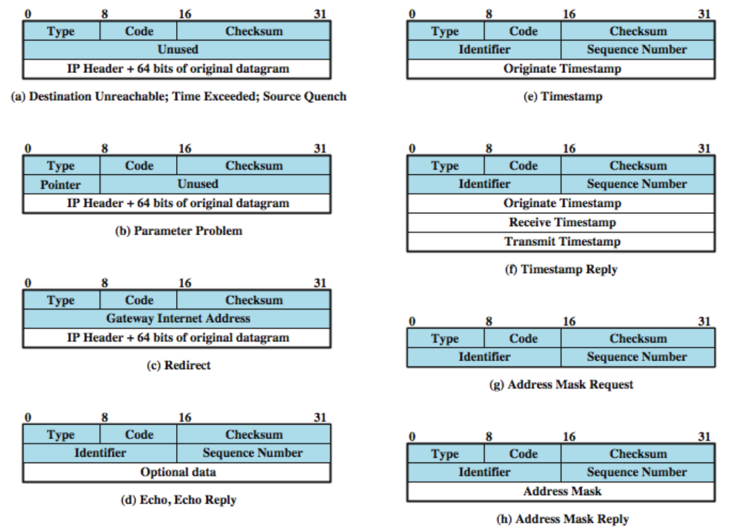
Common ICMP Messages

- destination unreachable
- time exceeded
- parameter problem
- source quench
- redirect
- echo & echo reply
- timestamp & timestamp reply
- address mask request & reply

1-64

64

ICMP Message Formats



1-65

65

Example

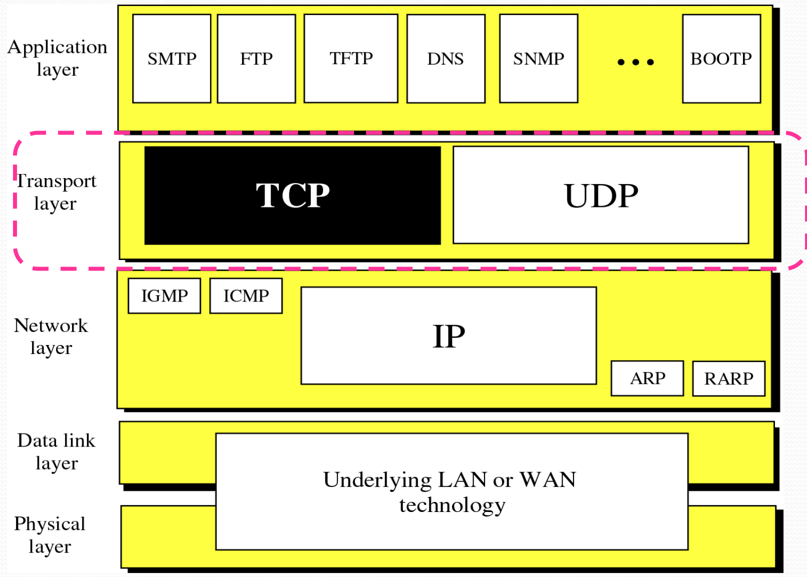
From: System Administrator
Sent: March 23, 2021 3:31 PM
To: Dimitrios Makrakis
Subject: Undeliverable: Midterm Review

```
Your message did not reach some or all of the intended recipients.

      Subject:   RE: Midterm Review
      Sent:      2021-03-23 3:31 PM

The following recipient(s) cannot be reached:

      'Navid Khalili' on 2021-03-23 3:31 PM
            None of your email accounts could send to this recipient.
```

1-66

66

TCP/IP Protocol Suite



1-67

67

Transport Protocols

- Provide end-to-end data transfer service
- Shield upper layers from network details
- Reliable, connection oriented
  - has greater complexity
  - e.g. TCP
- best effort, connectionless
  - datagram
  - e.g. UDP

1-68

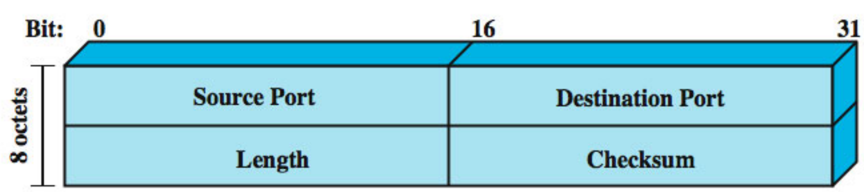
68

User Datagram Protocol (UDP)

1-69

69

UDP Header



1-70

70

User Datagram Protocol (UDP)

- Connectionless service for application level procedures specified in RFC 768
  - unreliable
  - delivery & duplication control not guaranteed
- Reduced overhead
- Weakest service to be expected at higher layers
- Uses:
  - inward data collection (e.g. periodic sampling of data sources such as sensors, automatic self-test reports from security equipment, network components)
  - outward data dissemination (e.g. broadcast messages to network users, announcement of a new node, change of address of a service, distribution of real-time clock values) .
  - request-response (e.g. a transaction service provided by a common server to a number of distributed users, and for which a single request-response sequence is typical).
  - real time application2 (e.g. voice, telemetry involving a degree of redundancy, a real-time transmission requirement).

1-71

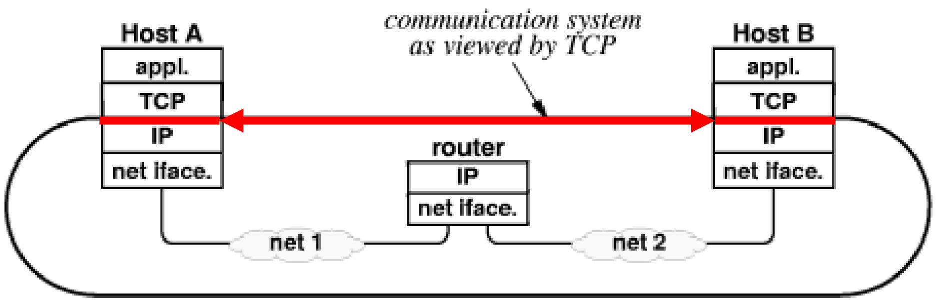
71

Transmission Control Protocol (TCP)

1-72

72

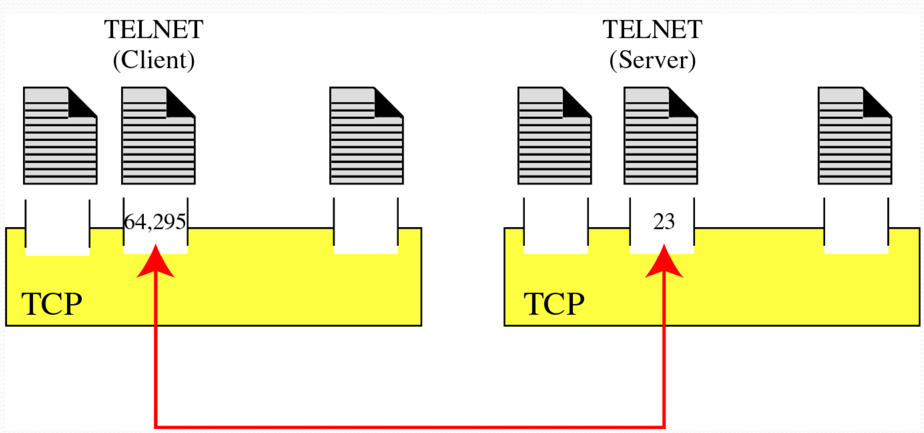
End-to-End Service



1-73

73

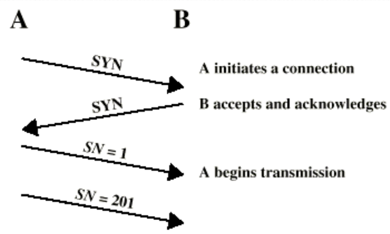
Port Numbers



1-74

74

TCP Connection Establishment:2-way handshake



- A sends SYN
- B replies with SYN

1-75

75

Services TCP Provides

- Connection-oriented service
- Establishes point-to-point communication
- Full-duplex communication
- Addressing, Multiplexing and Demultiplexing
- Complete reliability
- Reliable connection startup and shutdown
- Flow Control
- Congestion Control

1-76

76

Unreliable Network Service

- TCP/IP is assuming unreliable service.
- Consequences:
  - segments may get lost
  - segments may arrive out of order
- Issues:
  - ordered delivery,
  - retransmission strategy,
  - duplication detection,
  - flow control,
  - connection establishment & termination,
  - crash recovery

1-77

77

Ordered Delivery

- Segments may arrive out of order
- Solution: TCP numbers each byte (**NOT segment**) sequentially
- A segment is assigned as sequence number the number that has been assigned to the first octet contained in this segment

1-78

78

Flow Control in TCP

- Needs flow control because receiving side might not be able to keep up, resulting in buffer overflowing
- Issues:
  - longer transmission delay between transport entities compared with actual transmission time (considerable queuing and processing delays)
  - Transmission delay is variable; makes difficult to set timeouts

1-79

79

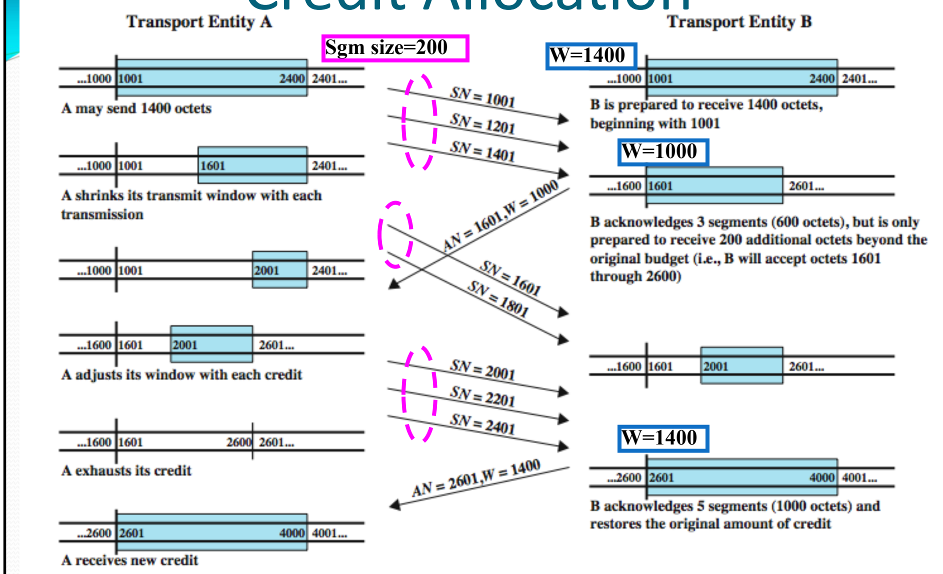
TCP uses "Credit Scheme"

- Decouples flow control from ACK
- Each byte has its own sequence number
- Parameters:
  - sequence number (SN)
  - acknowledgement number (AN)
  - window size (W)
- Segment' s sequence number is the SN of the first octet in the segment
- ACK includes (AN=i, W=j) which means:
  - all bytes through SN=i-**1** are acknowledged, "I" want byte i next
  - (receiving) station can accommodate and is ready to receive (can accommodate) up to W=j new bytes (bytes "**i**" to "**i+j-1**") even without new ACK been sent. Bytes can come in a single segment or in multiple segments.
  - Transmitting size segment size is decoupled from W (determined by congestion avoidance algorithm) but must be < or = to (W=)j.

1-80

80

Credit Allocation



1-81

81

Retransmission Strategy

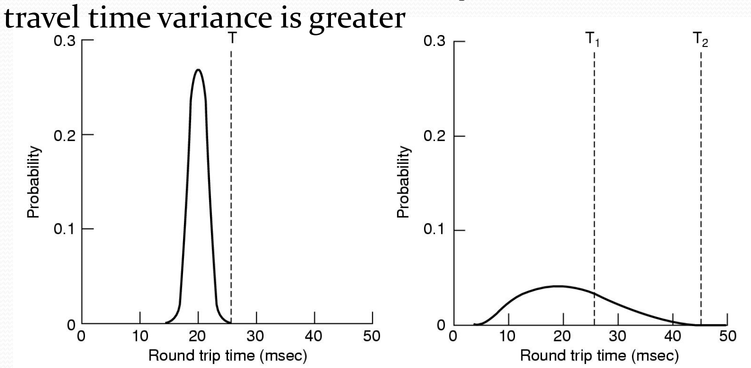
- Retransmission of segment needed because:
  - segment damaged in transit
  - segment fails to arrive
- Transmitter does not know of failure
- Receiver must acknowledge successful receipt (positive acknowledgment)
  - can use cumulative acknowledgement for higher efficiency (less load in return channel)
- Segment should be ACK before the timer (that is starting counting down when the segment is passed down to IP layer) expires.
- If timer expires without the sender receiving ACK, re-transmission is triggered.

1-82

82

TCP Timer Management

- How long should the retransmission timer value be?
- More difficult than in data link layer, since the round-trip travel time variance is greater



- Too short: unnecessary retransmissions will clog network.
- Too long: performance reduction due to long delay before retransmission.
- Value should be a bit longer from Round Trip Time (RTT).

1-83

83

Timer value updating

- Estimates round trip time (RTT) adaptively
- Problems exist: Example:
  - routers at forward and return path lightly loaded→ queuing delay at return path small These are the conditions at To
  - estimated RTT=X+Y (forward end-to-end=X, return end-to-end=Y)
  - At T1, a router of the forward path gets suddenly significantly higher traffic volume → queuing delay increases from X to (W>>X, W>>Y) ; however, router is not yet congested.
  - New RTT =W+Y >> X +Y
  - peer transport entity may not ACK segment immediately
  - counter might time out, triggering re-transmission of segments.
  - this might result in retransmission of many segments that are still on transit.
  - 1) Duplicate ACKs will be occurring. (Sender cannot distinguish between original and retransmitted segment)
  - 2) Waist of network resource, since obsolete segments will be consuming resources.
  - 3) TCP flow might slow down, assuming heavy congestion, which is not the case.

1-84

84

RTT Estimation

- RFC 1122: Requirements for Internet hosts
- Retransmission timer management
- Estimate round trip delay by observing pattern of delay
- Set time to value somewhat greater than estimate
- Estimation Methods:
  - Simple average
  - Exponential average
  - RTT Variance Estimation (Jacobson's algorithm)

1-85

85

Simple Average

- *ARTT(K)*: round trip time observed for the first *K* segments.
- *RTT(i)*: round trip delay observed for the *i*th segment used in the estimation.

$$\boxed{ARTT(K+1) = \frac{1}{K+1}\sum_{i=1}^{K+1} RTT(i)}$$

- Recursive calculation:

$$\boxed{ARTT(K+1) = \frac{K}{K+1} ARTT(K) + \frac{1}{K+1} RTT(K+1)}$$

1-86

86

Exponential Average

- SRTT(K): smoothed RTT estimate
- SRTT(0)=0
- α: constant ($0 < \alpha < 1$).
- Value of α affects speed of algorithm's convergence and its stability behaviour.

$$\boxed{SRTT(K+1) = \alpha \times SRTT(K) + (1-\alpha) \times RTT(K+1)}$$

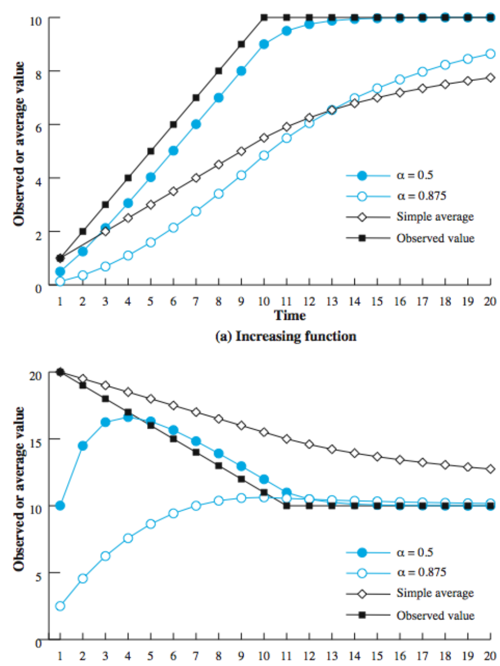
- The more distant a past observation is from the "present", the less weight has on the value of the estimate.

$$\boxed{SRTT(K+1) = (1-\alpha) \times [\sum_{i=1}^{K+1} \alpha^{K+1-i} \times RTT(i)]}$$

1-87

87

Simple versus Exponential Averaging



(a) Increasing function

(b) Decreasing function

1-88

88

Use of exponential algorithm/1

- RFC 793 uses exponential algorithm
- *RTO(K)*: value of retransmission counter after K observations have been used.
- Δ: constant

$$\boxed{RTO(K+1) = SRTT(K+1) + \Delta}$$

- When Δ << *SRTT(K)* , fluctuations in *SRTT(K)* will be controlling value of *RTO(K)*, generating unnecessary retransmissions.
- When Δ >> *SRTT(K)* , Δ will be controlling the value of *RTO(K)*, causing unnecessary delays in retransmitting lost segments.

1-89

89

Use of exponential algorithm/2

- To address these extreme cases, RFC 793 proposes the following approach:

$$\boxed{RTO(K+1) = MINIMUM\{UBOUNT, MAXIMUM[LBOUND, \beta \times SRTT(K+1)]\}}$$

- or equivalently

$$RTO(K+1) = \begin{cases} UBOUND\_\_\_ for \_\_\_ \beta \times SRTT(K+1) \geq UBOUND \\ \beta \times SRTT(K+1)\_\_\_ for \_\_\_ UBOUND \geq \beta \times SRTT(K+1) \geq LBOUND \\ LBOUND\_\_\_ for \_\_\_ \beta \times SRTT(K+1) \leq LBOUND \end{cases}$$

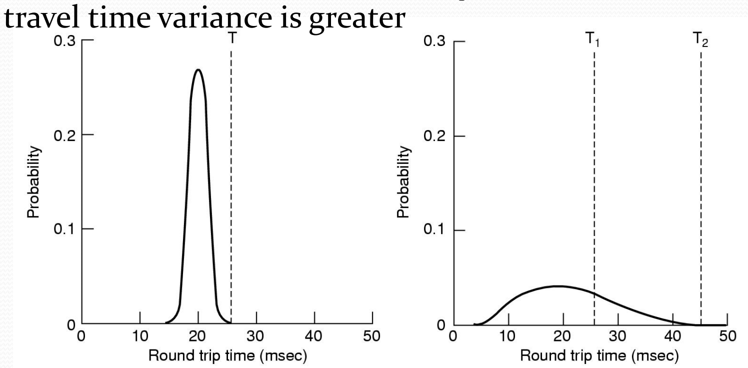
- *UBOUND, LBOUND* are pre-chosen upper and lower bounds (constant) and β is constant
- *RTO(K)* is proportional to *SRTT(K)* within the [*LBOUND, UBOUND*] interval.

1-90

90

TCP Timer Management

- How long should the retransmission timer value be?
- More difficult than in data link layer, since the round-trip travel time variance is greater



- Too short: unnecessary retransmissions will clog network.
- Too long: performance reduction due to long delay before retransmission.
- Value should be a bit longer from Round Trip Time (RTT).

1-91

91

RTT Variance Estimation (Jacobson's Algorithm)

- Previous approach encounters problems when RTT exhibits high variance (large deviations).
- The degree of deviations is taken into consideration.

$$SRTT(K+1) = (1-g) \times SRTT(K) + g \times RTT(K+1)$$

$$SERR(K+1) = RTT(K+1) - SRTT(K)$$

$$SDEV(K+1) = (1-h) \times SDEV(K) + h \times |SERR(K+1)|$$

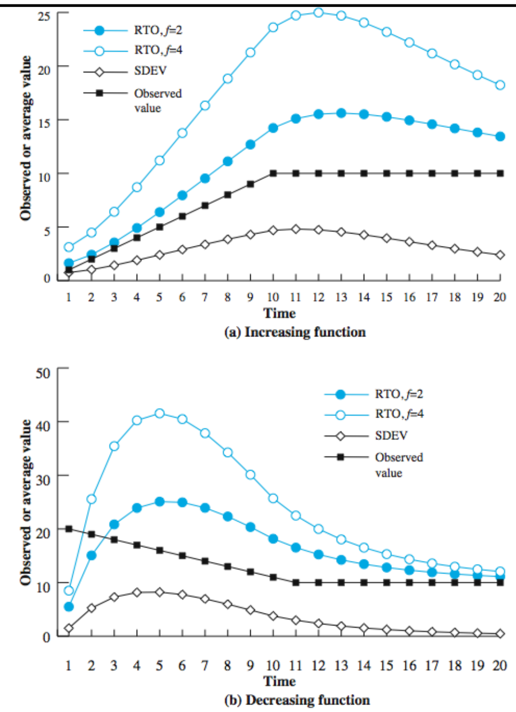
$$RTO(K+1) = SRTT(K+1) + f \times SDEV(K+1)$$

- Suggested values: g=1/8; h=1/4; f=2
- When large deviations occur, RTO provides larger offset from exponential average value (waits longer before expiring)

1-92

92

Jacobson's RTO Calculation



1-93

93

Congestion Control

1-94

94

Congestion

- Caused by too much traffic going through a network; more than the network can handle.
- Results in Routers dropping packets.
- Causes time-outs: equally likely from lost messages due to unreliable transmission media, as from congestion.
- Simply retransmitting a lost message makes congestion worst.
  - Why?
- There needs to be a way to control congestion.

1-95

95

Congestion Control

- Flow control used for congestion control
  - recognize increased transit times & dropped packets
  - react by reducing flow of data
- RFC's 1122 & 2581 detail extensions
  - Tahoe, Reno & New Reno implementations
- Two categories of extensions:
  - retransmission timer management
  - window management

1-96

96

Exponential RTO Backoff

- Timeout probably due to congestion
  - dropped packet or long round-trip time
- Maintaining RTO value is not good idea (else, many retransmissions will be generated, increasing congestion further)
- Better to increase RTO each time a segment is re-transmitted
  - RTO = q*RTO
  - commonly q=2 (binary exponential backoff)
  - as in Ethernet (CSMA/CD)

1-97

97

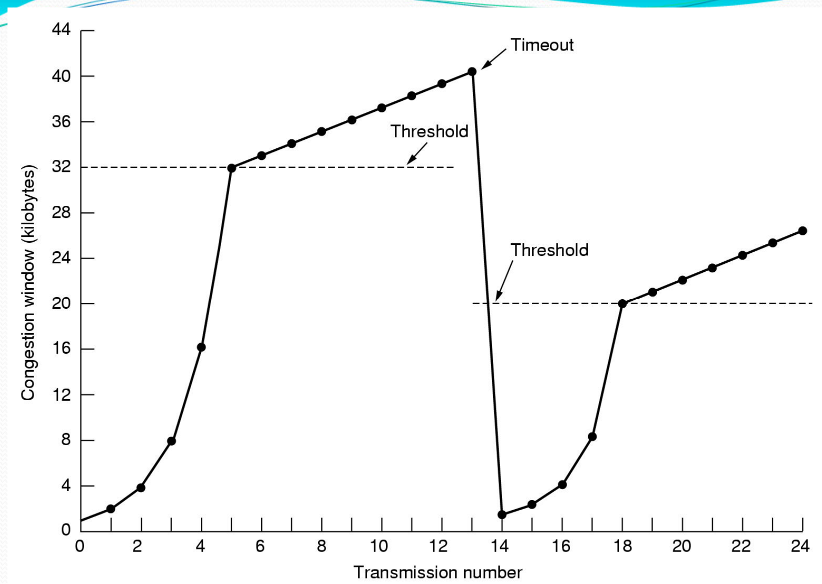
Karn's Algorithm

- If segment is re-transmitted, ACK may be for:
  - first copy of the segment (longer RTT than expected)
  - second copy
- No way to tell
- Don't measure RTT for re-transmitted segments (i.e. don't use them for updating RTO)
- Calculate backoff when re-transmission occurs
- Use backoff RTO until ACK arrives for segment that has not been re-transmitted

1-98

98

TCP Slow Start



99

99

Window Management/1

- Slow start
  - larger windows cause problem on connection created
  - at start limit TCP to 1 segment
  - increase when data ACK, exponential growth
- Dynamic windows sizing on congestion
  - when a timeout occurs perhaps due to congestion
  - set slow start threshold to half current congestion window
  - set window to 1 and slow start until threshold
  - beyond threshold, increase window by 1 for each RTT
- In all cases, the window size should not be exceeding the credit

1-100

100

Window Management/2

- awnd = MIN[credit, cwnd]
- Slow start
  - Start connection with cwnd=1
  - Increment cwnd (double) with ACK, to some threshold
  - Increment cwnd by 1 with ACK after the threshold
- Dynamic windows sizing on congestion
  - When a timeout occurs
  - Set slow start threshold to half current window
    - ssthresh=cwnd/2
  - Set cwnd = 1 and slow start until cwnd=ssthresh
    - Increasing cwnd by 1 for every ACK
  - For cwnd >=ssthresh, increase cwnd by 1 for each RTT

1-101

101

Fast Retransmit - Fast Recovery

- Retransmit timer rather longer than RTT
- If segment lost TCP slow to retransmit
- Fast retransmit
  - if receive 4 ACKs for same segment then immediately retransmit since likely lost
- Fast recovery
  - lost segment means some congestion
  - halve window then increase linearly
  - avoids slow-start

1-102

102

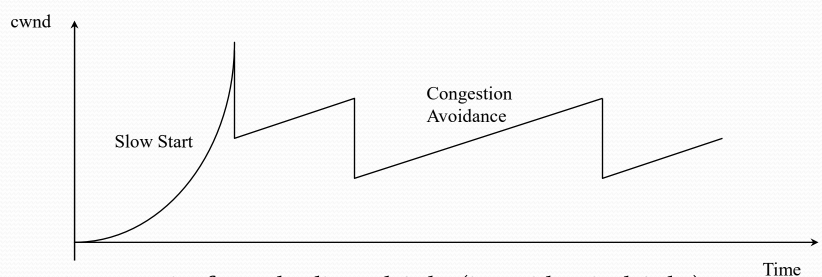
Why wait for 4 identical Acks? Why use fast recovery

- If receiving TS receives a segment out of order, must immediately issue Ack for last in order received segment.
- It will continue repeating this with each incoming segment, until the missing one arrives; then receiving TS sends a cumulative Ack for all in order segments received.
  - 1st Ack: segment following the last confirmed (indicated as the one the receiving TS is waiting for) might have been delayed or lost or a latter sent segment arrived.
  - 2nd Ack: a segment sent latter from the one receiving TS is waiting for arrived. Also traffic still goes through (congestion is not extreme)
  - 3rd and 4th Acks: segments sent after the one receiving TS is waiting for arrived without the one requested having arrived.
  - Chances are segment has been lost; retransmit it.
  - Since congestion appears to be not extreme, go through fast recovery, i.e. reduce your window but do not close to cwnd=1.

1-103

103

Fast Recovery

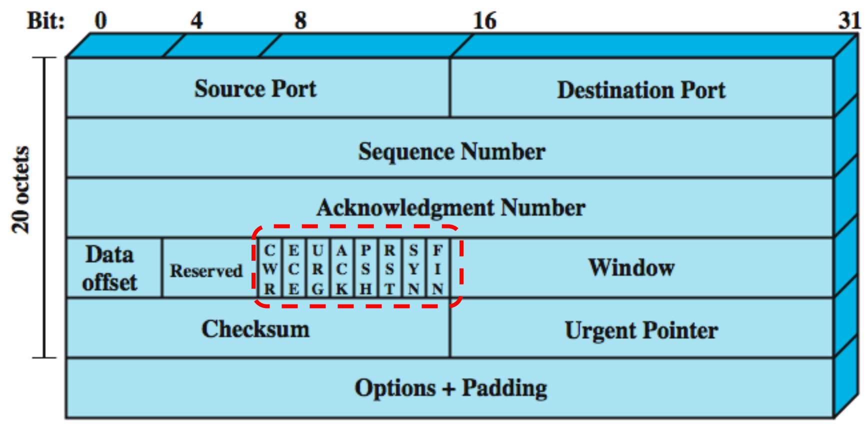


- Retransmit after 3 duplicated Acks (i.e. 4 identical Acks)
  - congestion is not severe
  - prevent expensive timeouts
- No need to slow start again
- Reduce to half the existing window (cwnd=cwnd/2)
- Increase linearly by adding 1 segment each time
- At steady state, *cwnd* oscillates around the optimal window size.

1-104

104

TCP Header



1-105

105

TCP Header Fields

- Source and destination ports: 16 bit address of local port (socket).
- Sequence and acknowledgment numbers:
  - Every byte is numbered in a TCP stream.
  - Acknowledgment number is next byte number expected.
  - 32 bits each.
- Data Offset:
  - Needed because options field can vary in length.
  - Number of 32 bits words in header.
- URG: set to 1 if urgent pointer in use
  - Meaning that the receiving program should be notified of its arrival as soon as possible.
  - Pointer indicates offset from current sequence number at which urgent data ends.

1-106

106

TCP Header Fields (cont'd)

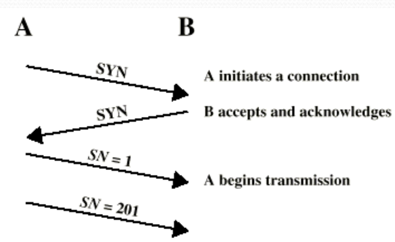
- CWR: Congestion window reduced (used for explicit congestion notification)
- ECE: ECN-Echo: (used for explicit congestion notification)
- ACK: Set to 1 to indicate acknowledgment number is valid
  - If no, no acknowledgment in this segment.
- PSH: Set to 1 to indicate pushed data.
  - Force delivery of bytes currently in the stream without waiting for buffer to fill.
- RST: Set to 1 to indicate reset.
  - Host has become confused due to crash or for other reason.
  - Also used to reject a connection or refuse an invalid segment.
- SYN: used to establish connections.
  - SYN = 1, ACK = 0 in connection request.
  - SYN = 1, ACK = 1 in connection acceptance.
- FIN: set to 1 to indicate end of user data.
  - Used to close connection.
  - May continue to receive data.

1-107

107

TCP Connection Establishment:2-way handshake

- A sends SYN
- B replies with SYN



1-108

108

IP Addressing

1-109

109

IP Address

- Each IP address is divided into a prefix and a suffix
  - prefix identifies the network to which computer is attached
  - suffix identifies the computer within that network
  - we allocate some bits for prefix, some for suffix (total of 32 bits)
    - large prefix, small suffix - many networks, few hosts per network
    - small prefix, large suffix - few networks, many hosts per network
- Network numbers are unique
  - assignment of network numbers must be coordinated globally; assignment of host addresses can be managed locally

1-110

110

Addressing Mechanisms

- Popular mechanisms for addressing networks are:
  - Classful addressing (older style of addressing).
  - Subnetting (A better way to distribute addresses).
  - Variable-length subnetting (Even more refined than subnetting).
  - Supernetting and Classless interdomain routing (CIDR). (An efficient way to advertise addresses)
  - Private addressing and Network Address Translation (NAT). (A way to get more addresses)
  - Dynamic addressing (An easy way to get addresses)

1-111

111

IP Address format-Classful Addressing

		8	32	
0	netid	hostid		Class A

		16	32	
10	netid	hostid		Class B

		24	32	
110	netid	hostid		Class C

	24	32	
1110	multicast		Class D

1-112

112

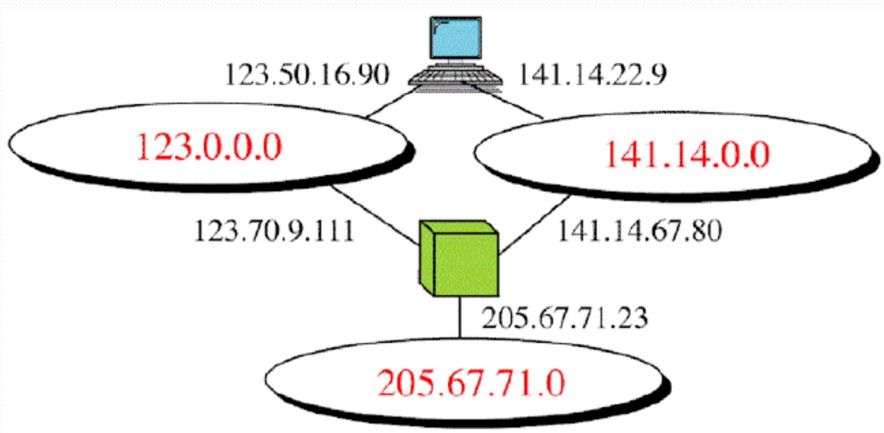
IP Address format

11110	Reserved for future use	Class E

Host id 0 is never assigned to an individual host. It refers to the network itself.

113

Example



114

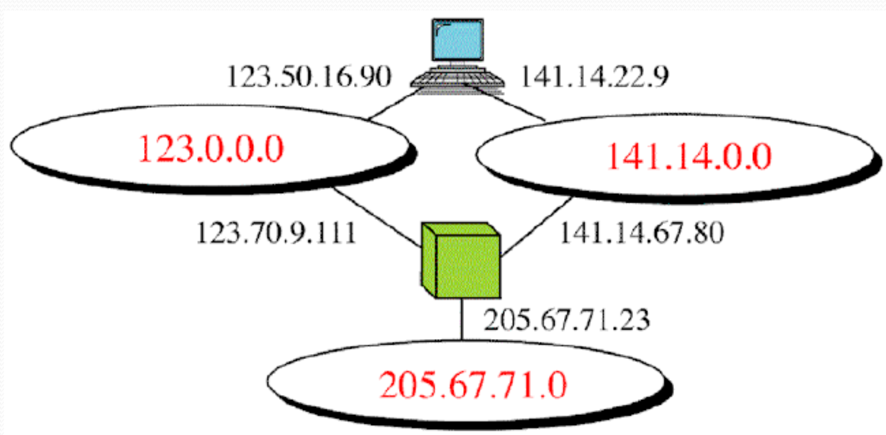
Address Range-Summary

	From		To	
Class A	0.0.0.0	Netid: 0 / Hostid: 0.0.0	127.255.255.255	Netid: 127 / Hostid: 255.255.255
Class B	128.0.0.0	Netid: 128.0 / Hostid: 0.0	191.255.255.255	Netid: 191.255 / Hostid: 255.255
Class C	192.0.0.0	Netid: 192.0.0 / Hostid: 0	223.255.255.255	Netid: 223.255.255 / Hostid: 255
Class D	224.0.0.0	Multicast Address	239.255.255.255	Multicast Address
Class E	240.0.0.0	Reserved	255.255.255.255	Reserved

1-115

115

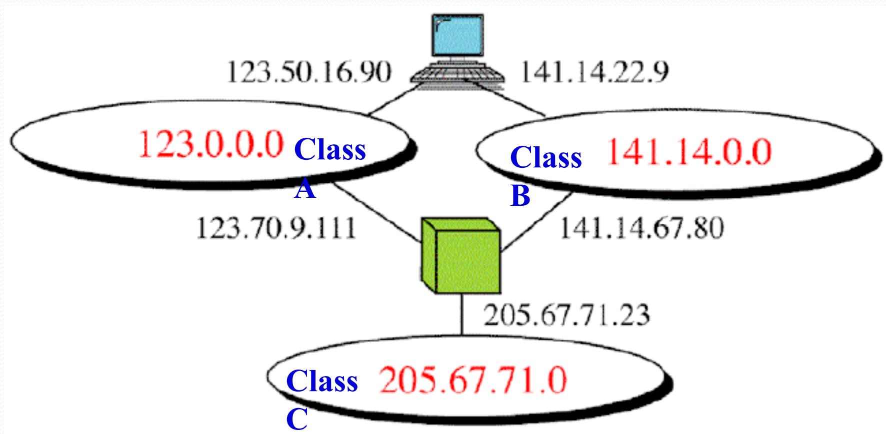
Example



1-116

116

Example



1-117

117

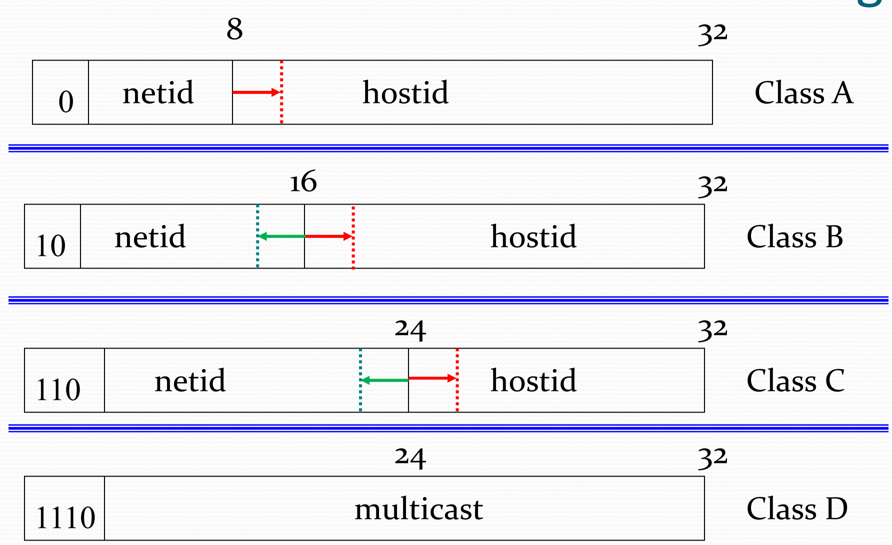
Characteristics of major Classes

Address Class	No. of Networks	No. of Hosts	Comments
A	127	16777214	Very Large Networks
B	16383	65534	Medium Size Netw.
C	2097151	254	Large number of small networks

1-118

118

IP Address format-Classful Addressing



1-119

119

Subnet Mask Calculation

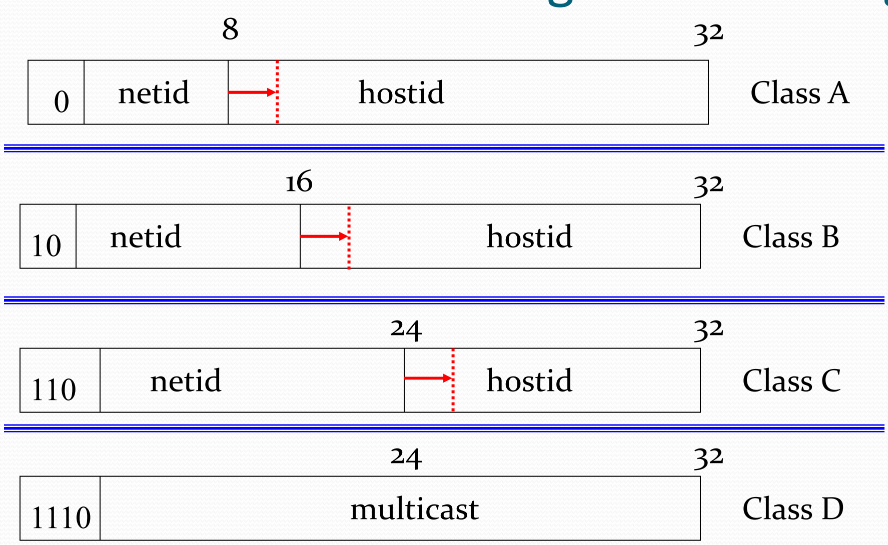
Class C

	Binary Representation	Dotted Decimal
IP address	$\boxed{11}$000000.11100100.00010001 .00111001	192.228.17.57
Subnet mask	11111111.11111111.11111111 .11100000	255.255.255.224
Bitwise AND of address and mask (resultant network/subnet number)	11000000.11100100.00010001 .00100000	192.228.17.32
Subnet number	11000000.11100100.00010001 .001	1
Host number	00000000.00000000.00000000 .00011001	25

1-120

120

From Classful Addressing to Subnetting

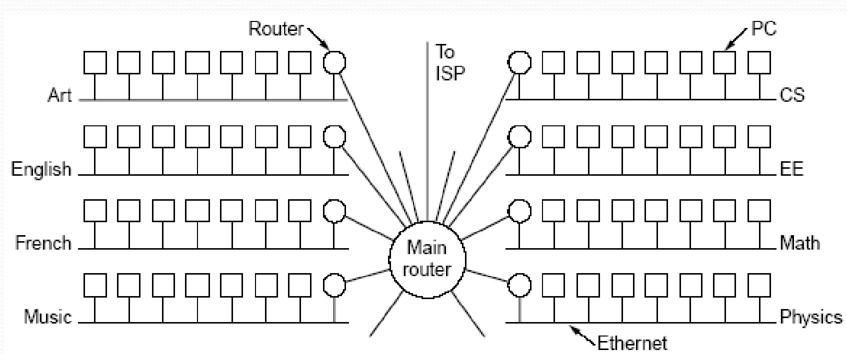


1-121

121

Subnets and Subnet Masks

- Allow arbitrary complexity of internetworked LANs within organization.
- Insulate overall internet from growth of network numbers and routing complexity.
- To rest of internet, site looks like single network.



122

122

Subnet Mask Calculation

Class C

	Binary Representation	Dotted Decimal
IP address	11000000.11100100.00010001 .00111001	192.228.17.57
Subnet mask	11111111.11111111.11111111 .11100000	255.255.255.224
Bitwise AND of address and mask (resultant network/subnet number)	11000000.11100100.00010001 .00100000	192.228.17.32
Subnet number	11000000.11100100.00010001 .001	1
Host number	00000000.00000000.00000000 .00011001	25

1-123

123

Subnets and Subnet Masks

- Each LAN is assigned subnet number.
- Host portion of address partitioned further into subnet number and host number.
- Local routers route within subnetted network.
- Subnet mask indicates which bits are subnet number and which are host number by doing a bitwise AND.

Subnet Mask	255.255.240.000	11111111.11111111.1111 0000 00000000
IP Address	150.215.017.009	10010110.11010111.0001 0001 00001001
Subnet Address	150.215.016.000	10010110.11010111.0001 0000 00000000

124

124

Subnetting

- Allows a classful network address to be segmented into smaller sections by using part of the device address to create another level of hierarchy.
- Basically, it takes address space away from the devices and gives it to the network.
- Improves routing speed.
- Allows establishment of addressing individual networks, using addressing from classes having too many users assigned to the same classful network address.

1-125

125

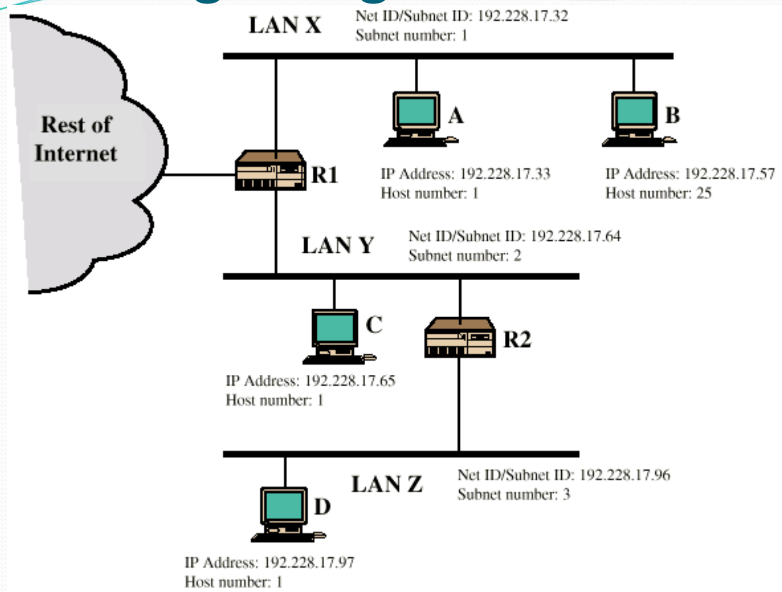
Advantages of Subnetting

- Traffic problems from jabbering devices and broadcast/multicast storms can be reduced by separating the network into subnets connected by routers.
- Routers terminate the link and physical layers thus stopping problems in one subnet from affecting other subnets.
- Subnet masks are only recognized within that network. The hierarchy is not revealed to the outside world.

1-126

126

Routing Using Subnets



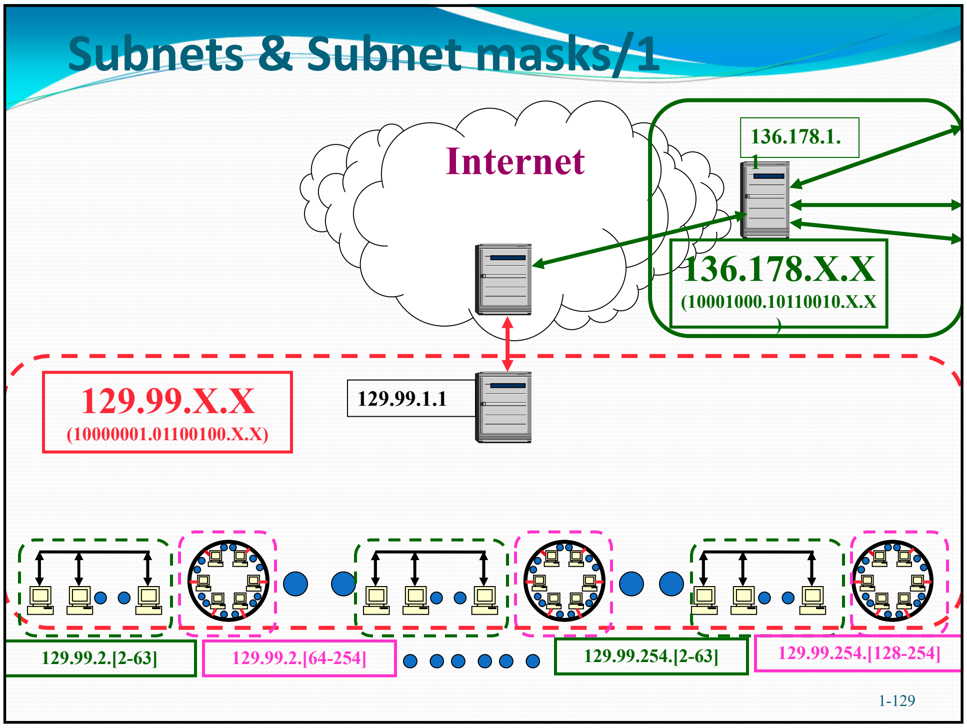
1-127

127

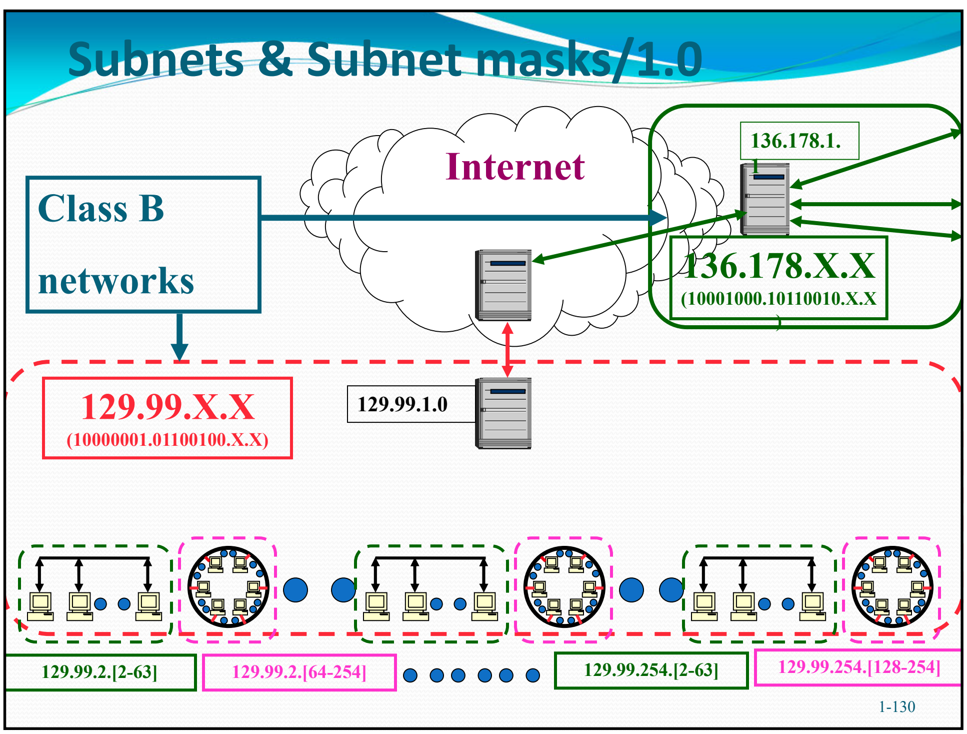
EXAMPLE

1-128

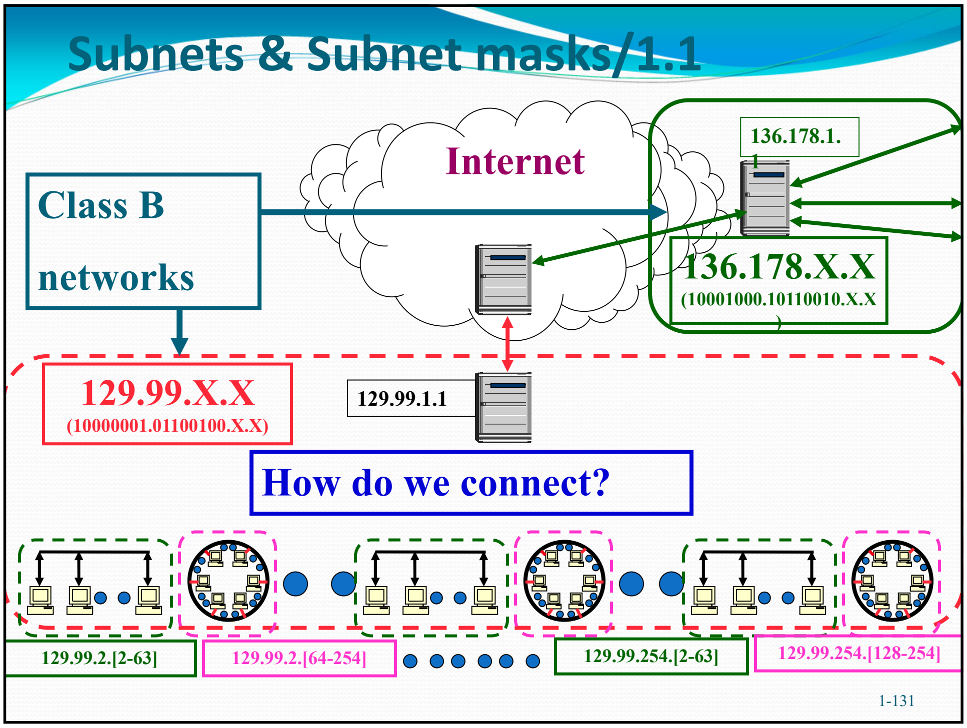
128


Subnets & Subnet masks/1
Internet
136.178.1.1
136.178.X.X
(10001000.10110010.X.X)
129.99.X.X
(10000001.01100100.X.X)
129.99.1.1
129.99.2.[2-63]
129.99.2.[64-254]
129.99.254.[2-63]
129.99.254.[128-254]
1-129

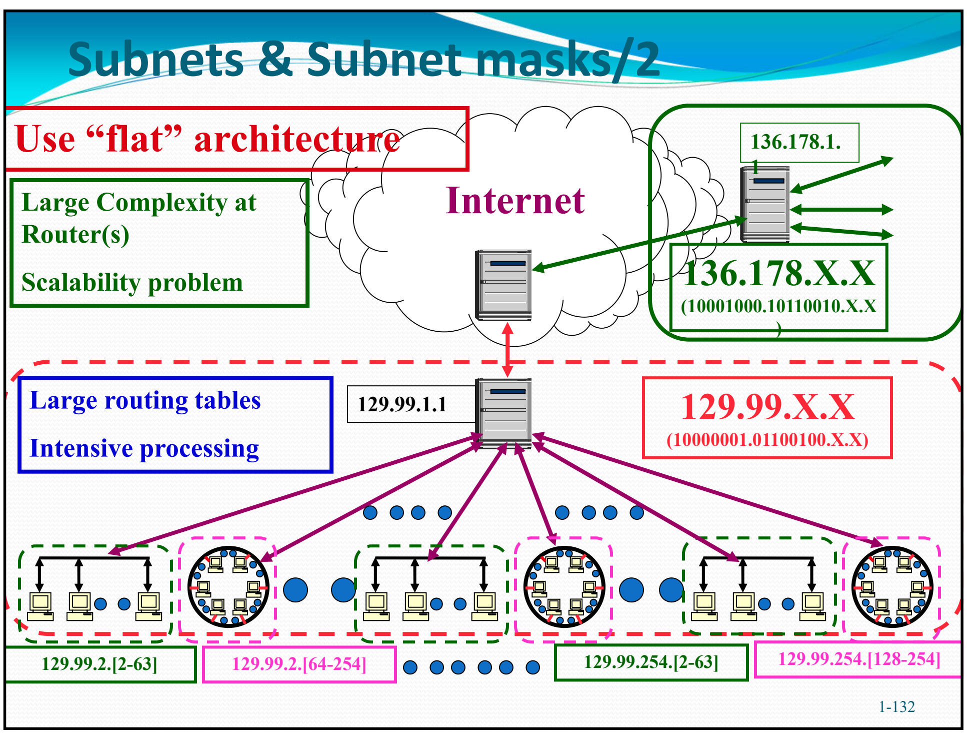
129


Subnets & Subnet masks/1.0
Class B
networks
Internet
136.178.1.1
136.178.X.X
(10001000.10110010.X.X)
129.99.X.X
(10000001.01100100.X.X)
129.99.1.0
129.99.2.[2-63]
129.99.2.[64-254]
129.99.254.[2-63]
129.99.254.[128-254]
1-130

130


Subnets & Subnet masks/1.1
Internet
136.178.1.
Class B
networks
136.178.X.X
(10001000.10110010.X.X)
129.99.X.X
(10000001.01100100.X.X)
129.99.1.1
How do we connect?
129.99.2.[2-63]
129.99.2.[64-254]
129.99.254.[2-63]
129.99.254.[128-254]
1-131

131


Subnets & Subnet masks/2
Use "flat" architecture
Large Complexity at Router(s)
Scalability problem
Internet
136.178.1.
136.178.X.X
(10001000.10110010.X.X)
Large routing tables
Intensive processing
129.99.1.1
129.99.X.X
(10000001.01100100.X.X)
129.99.2.[2-63]
129.99.2.[64-254]
129.99.254.[2-63]
129.99.254.[128-254]
1-132

132

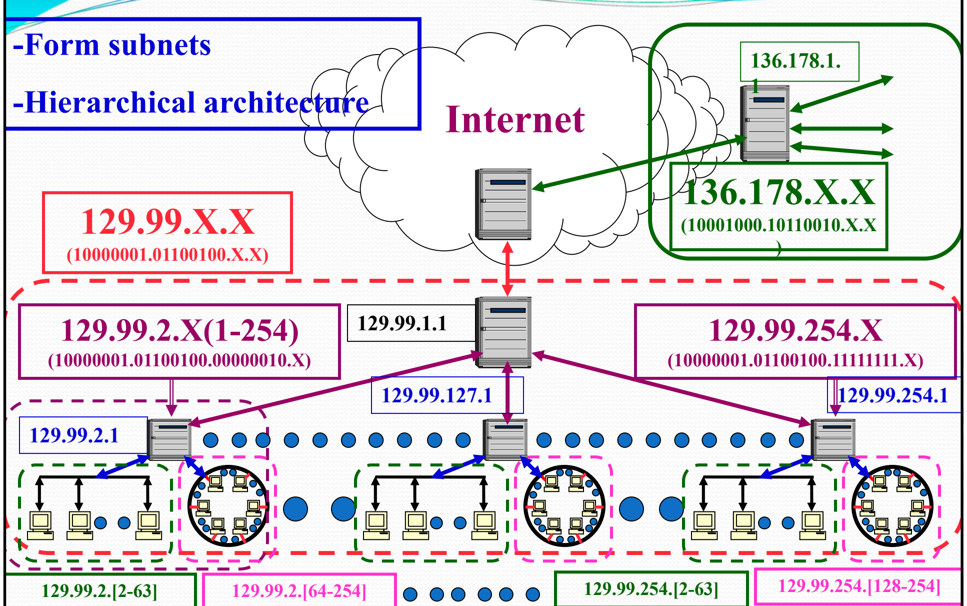
Subnets & Subnet masks/3

- Router <129.99.1.1> has to keep track of [(63-1) + (254-63)]x(254-1)=63504 IP addresses (stored in its routing table; 1 IP address per node).
- Should the class B network was fully utilized, the router would have to keep track of as much as [65534 – 1] =65533 address (stored in its routing table; 1 IP address per node).

1-133

133

Subnets & Subnet masks/4



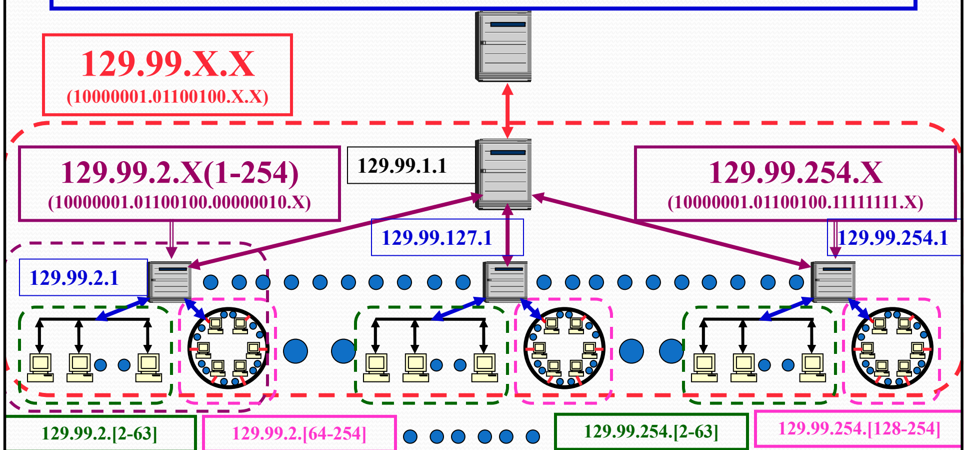
1-134

134

Subnets & Subnet masks/5

Node <129.99.2.20> sends packet to node <X.Y.W.Z>

Router <129.99.2.1> needs to decide if packet should stay within subnet of <129.99.2.20>, move to the neighboring subnet, or be sent to router <129.99.1.1>



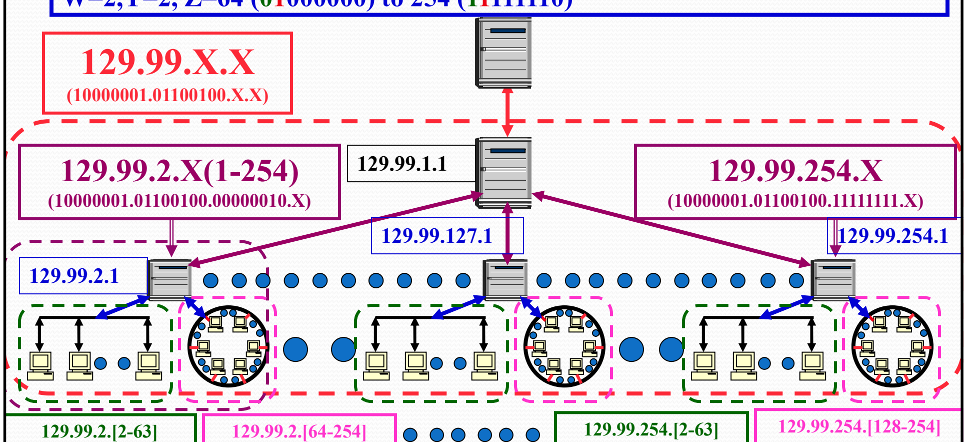
1-135

135

Subnets & Subnet masks/6

If packet is to stay on <129.99.2.20>' s subnet (ethernet), X=129, Y=99, W=2, Z = 2 (00000010) to 127 (00111111)

If packet is destined for the neighboring subnet (token ring), X=129, Y=99, W=2,Y=2, Z=64 (01000000) to 254 (11111110)



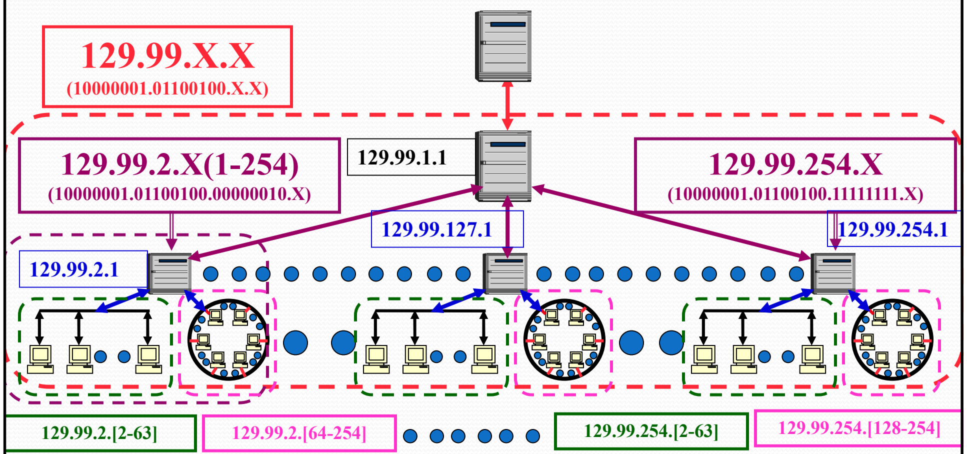
1-136

136

Subnets & Subnet masks/7

If packet is to be sent to Router <129.99.1.1>

then: {X≠129} AND/OR {Y≠99} ABR/OR {W≠ 2}



1-137

137

Subnets & Subnet masks/8

If packet is to be sent to Router <129.99.1.1>

then: {X≠ 129} AND/OR {Y≠99} ABR/OR {W≠ 2}

X.Y.W.Z = xxxxxxx.yyyyyyyy.wwwwwwww.zzzzzzzz

where {x, y, w, z} are 0 or 1.

To find out about the above condition, I need to examine only X,Y, W

We remove Z be applying the following mod2 addition:

xxxxxxxx.yyyyyyyy.wwwwwwww.zzzzzzzz = X.Y.W.Z

<u>11111111.11111111.1 1 1 1 1 1 1 1.00000000=255.255.255.0</u>

xxxxxxxx.yyyyyyyy.wwwwwwww.00000000 = X.Y.W.0

1-138

138

Subnets & Subnet masks/9

If packet is to stay on <129.99.2.20>' s Ethernet, X=129, Y=99, W=2, Z = 2 (00000010) to 63 (00111111)

If packet is destined for the neighboring token ring, X=129, Y=99, W=2,Y=2, Z=64 (01000000) to 254 (11111110)

To decide one or the other of the above, I have to check if X=129, Y=99, W=2, and find out if the 2 most important bits of Z are <00> OR NOT.

To remove the redundant bits, we perform the following mod2 addition:

xxxxxxxx.yyyyyyyy.wwwwwwww.zzzzzzzz = X.Y.W.Z

<u>11111111.11111111.1 1 1 1 1 1 11.11000000=255.255.255.192</u>

xxxxxxxx.yyyyyyyy.wwwwwwww.zz000000

11111111.11111111.11111111.11000000=255.255.255.192

Is the "mask", Router <129.99.2.1> should use.

1-139

139

Subnets & Subnet masks/10

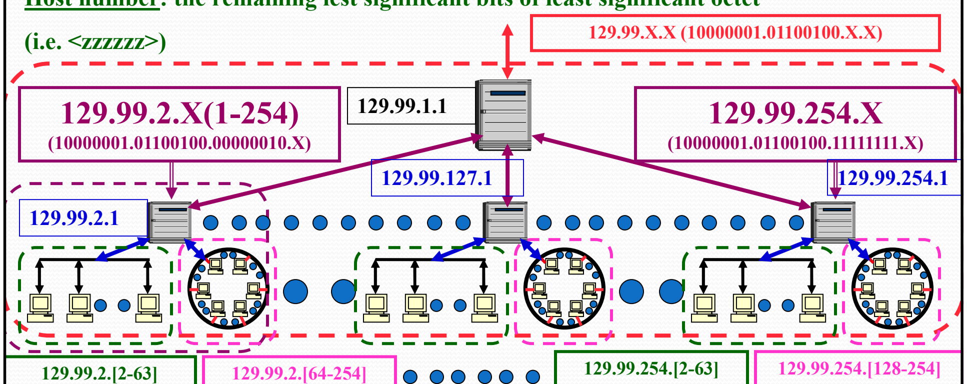
All address of nodes on Ethernet LAN connected to Router <129.99.2.1> are of the form <129.99.2.00zzzzzz> and subnet mask is <255.255.255.192>.

<u>Subnet number</u>:<129.99.2.00zzzzzz> ⊕ <255.255.255.192> = <129.99.2.0>

(10000001.01100100.00000010.00zzzzzz> ⊕ <11111111.11111111.11111111.1100000> = <10000001.01100100.00000010.0000000>

<u>Host number</u>: the remaining lest significant bits of least significant octet

(i.e. <zzzzzz>)



1-140

140

Subnets & Subnet masks/11

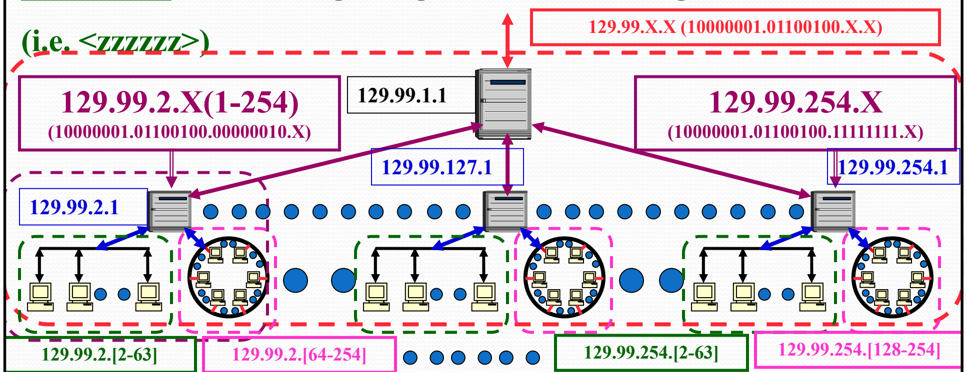
All address of nodes on Token Ring LAN connected to Router <129.99.2.1> are of the form <129.99.2.01zzzzzz> and subnet mask is <255.255.255.192>.

Subnet number:<129.99.2.01zzzzzz> ⊕ <255.255.255.192> = <129.99.2.64>

(10000001.01100100.00000010.01zzzzzz> ⊕ <11111111.11111111.11111111.1100000> = <10000001.01100100.00000010.0100000>

Host number: the remaining lest significant bits of least significant octet

(i.e. <zzzzzz>)



1-141

141

Subnets & Subnet masks/12

- Router identifies the subnet the packet has to be sent to by determining the subnet number (in the example <129.99.2.0> for Ethernet subnet , <192.99.2.64> for Token Ring> subnet.
- Any packet having subnet value different from either of the above is either sent to Router <129.99.1.1> or is dropped (if it arrived from that router).
- Nodes identify if the packet belongs to them by determining the host number part of the packet and comparing it to their own.
- Maintaining list of IP address of all nodes of the connecting subnets is avoided. Scalability is maintained.

1-142

142

Variable-length Subnetting

- Subnetting divides the network into a number of equal-sized subnets which is often inefficient.
- *Variable-length subnetting* is subnetting in which variable length subnet masks are used.

1-143

143

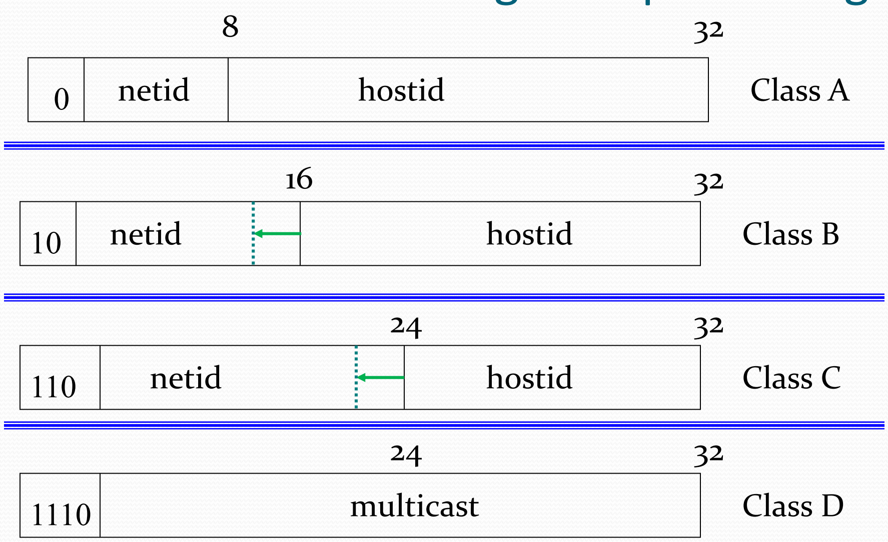
Supernetting

- Millions of Class C addresses can be allocated in lieu of Class A & B.
  - The result is that too many Class C address groups need to be allocated to an organization and advertised among all the Internet routers.
  - The number of routes would grow exponentially such that some experts had predicted that the Internet would collapse by 1995.
  - Obviously this did not happen, because the concept of supernetting was invented.

1-144

144

From Classful Addressing to Supernetting



1-145

145

What if? Supernetting

- It is more common to allocate Class C addresses than A or B. (millions of Class C can be allocated).
- A company needs to support 10,000 devices.
  - A class C supports up to 254 devices, so 40 class C networks are needed.
  - How are we to advertise these 40 class C addresses?
- Supernetting
  - *Supernetting* is the concept of aggregating network addresses by changing the network mask to decrease the number of bits recognized as the network.

1-146

146

Supernetting

- If we take a set of 16 contiguous addresses from a Class C address like 192.92.240.0 we can see that in this case the first 4 digits of the subnet octet do not change. This range of values can be represented as 192.92.240.0 with a subnet mask of 255.255.240.0 where the last 4 bits in the third octet are ignored.
- This then can be used to advertise a group of addresses. i.e. 192.92.240.0/20 = 192.92.240.0 -> 192.92.255.0
- This removed the need for class boundaries and brought into effect the *classless interdomain routing* (CIDR).

1-147

147

CIDR

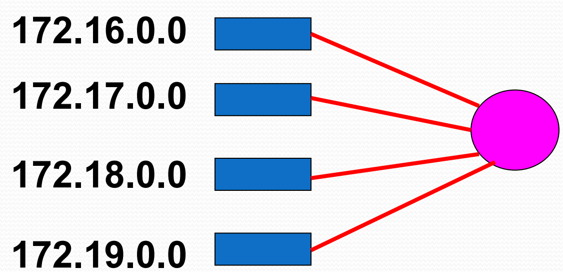
- Addresses must be assigned in contiguous blocks following logical topology.
  - The number of addresses in a CIDR block are powers of 2.
- Used in conjunction with classless routing protocols (e. g. EIGRP, OSPF)
- Network prefix is transmitted along with address
  - Example 192.92.240/22 advertises 4 networks (240, 241, 242, & 243 because the last 2 bits in the subnet can be ignored.
- Use any length prefix, not just 8,16,24
  - For example 200.1.128.0/17 is equivalent to a range of 2^7, or 128 networks from 200.1.128.0 -> 200.1.255.0

1-148

148

Route Summarization

- Classfull addressing automatically summarizes on 8, 16 or 24 bit
- Given:

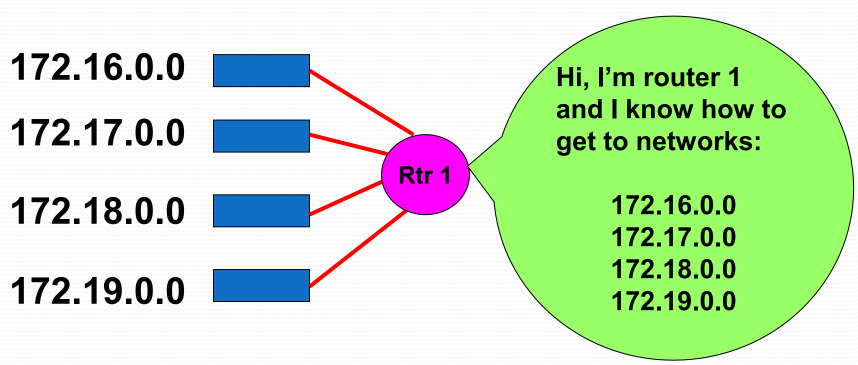


1-149

149

Route Summarization

- Classful router must advertise all 4 nets

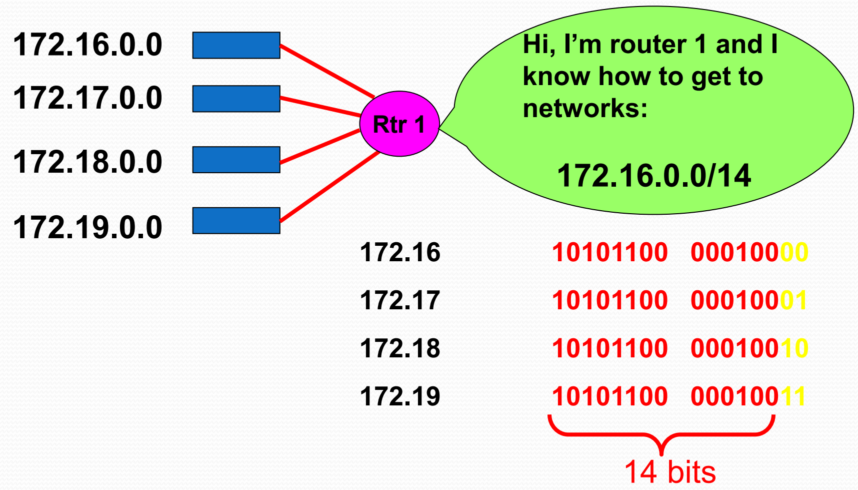


1-150

150

Route Summarization

- Classless router only advertises one



1-151

151

Route Summarization

- How to tell if a block of subnets can be summarized:
  - Number of subnets is a power of 2
  - Relevant octet of <u>first</u> address in block is a multiple of number of subnets.
    - The first address of the block needs to be evenly divisible by the number of subnets.
    - 192.92.240.0 (The 240 is evenly divisible by 1,2,4,8, &16 and therefore can be summarized by /24, /23, /22, /21, & /20

1-152

152

Route Summarization Example

1. Number of networks is 8, which is a power of 2, so first rule is met
2. The third octet is the relevant octet. The first relevant octet is 48, which is a multiple of 8 (the number of networks in rule #1)

10.108.48.0
10.108.49.0
10.108.50.0
10.108.51.0
10.108.52.0
10.108.53.0
10.108.54.0
10.108.55.0

So this block of addresses can be aggregated

1-153

153

Private Addresses

- *Private IP Addresses* are reserved addresses that can't be forwarded to the Internet
- 10.0.0.0 -> 10.255.255.255 (10/8 prefix)
- 172.16.0.0 -> 172.31.255.255 (172.16/12 prefix)
- 192.168.0.0 -> 192.168.255.255 (192.168/16 prefix)

1-154

154

Network Address Translation (NAT)

- Translates private <-> public addresses
  - A binding is created between the addresses that lasts a period of time.
- Can be implemented in:
  - Router
  - Firewall
  - Specialized device
- Assigned pool of public address
  - 1 to 1 private-public binding known as *static NAT* generally used for a server
  - 1 to many bindings are *dynamic NAT* generally used for devices.
  - Port address translation known as *Network Address Port Translation* (NAPT). Commonly used in corporations or for HTTP.

1-155

155

NAT Servers

- May use static assignments for Web servers, etc.
- NAT gateway must handle all internal/external traffic, may have to:
  - Translate addresses within packet (for example Session Initiation Protocol SIP packets)
  - Recalculate header checksums
- Should have fast processor to achieve low delay and high throughput

1-156

156

Dynamic Addressing

- Domain assigns to a user an IP address temporarily.
- Makes use of Dynamic Addressing
- Use of Dynamic Name Server (DNS)
- Use of address auto-configuration (check IPv6)

1-157

157

END

1-158

158