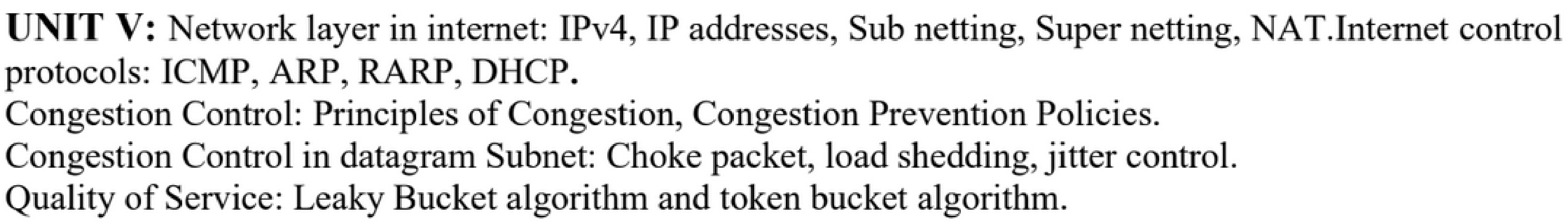
Computer Networks

UNIT – V



UNIT V: Network layer in internet: IPv4, IP addresses, Sub netting, Super netting, NAT.Internet control protocols: ICMP, ARP, RARP, DHCP**.**
Congestion Control: Principles of Congestion, Congestion Prevention Policies.
Congestion Control in datagram Subnet: Choke packet, load shedding, jitter control.
Quality of Service: Leaky Bucket algorithm and token bucket algorithm.

The Network Layer Principles (1)

1. Make sure it works
2. Keep it simple
3. Make clear choices
4. Exploit modularity
5. Expect heterogeneity

. . .

The Network Layer Principles (2)

. . .

6. Avoid static options and parameters
7. Look for good design (not perfect)
8. Strict sending, tolerant receiving
9. Think about scalability
10. Consider performance and cost

The Network Layer in the Internet

1. **Make sure it works:** Do not finalize the design or standard until multiple prototypes have successfully communicated with each other.
2. **Keep it simple:** When in doubt, use the simplest solution. If a feature is not absolutely essential, leave it out, especially if the same effect can be achieved by combining other features.

3. **Make clear choices:** If there are several ways of doing the same thing, choose one.

4 . **Exploit modularity:** This principle leads directly to the idea of having protocol stacks, each of whose layers is independent of all the other ones. In this way, if circumstances that require one module or layer to be changed, the other ones will not be affected.

5. **Expect heterogeneity:** Different types of hardware, transmission facilities, and applications will occur on any large network. To handle them, the network design must be simple, general, and flexible.

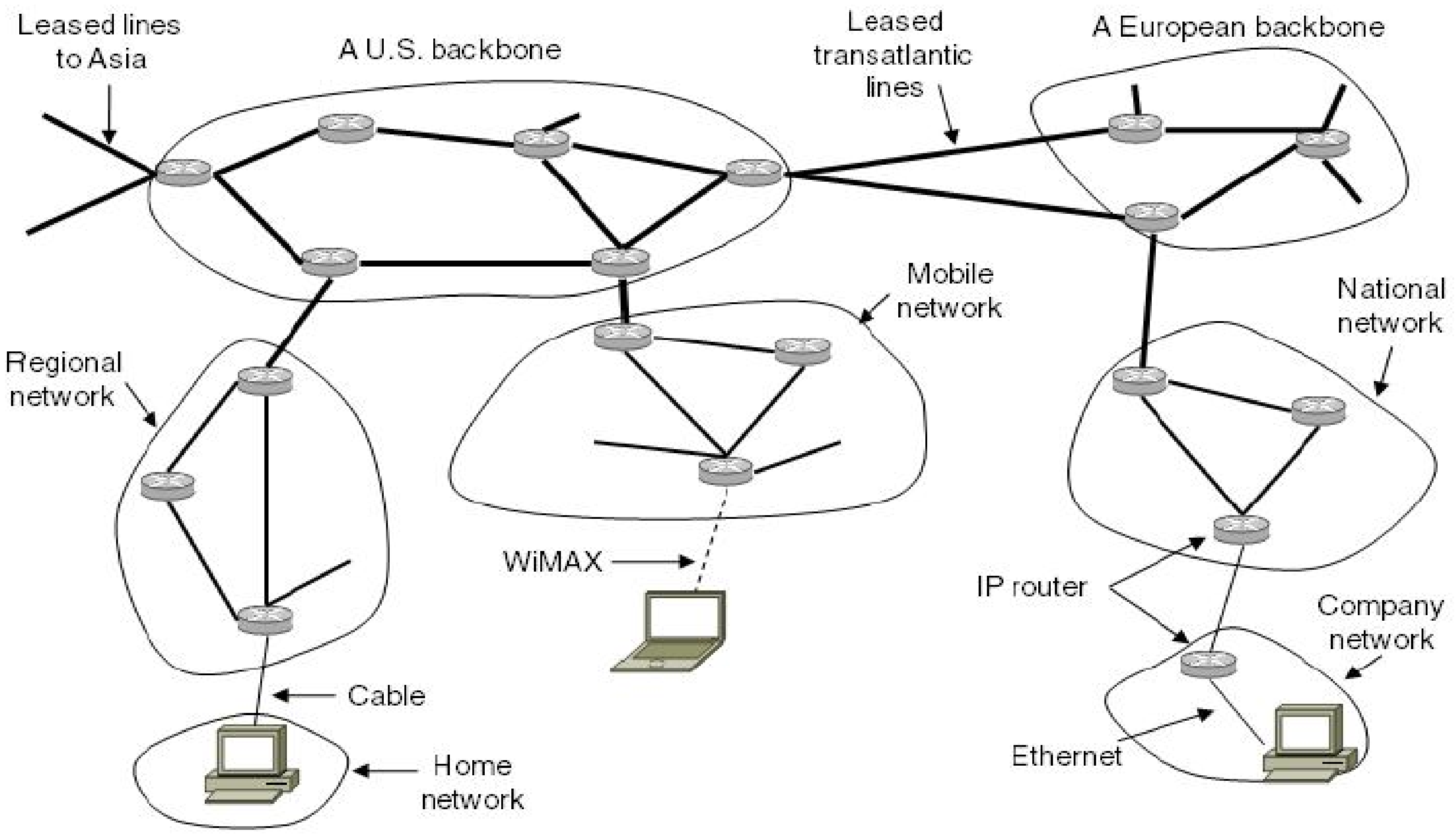
6. **Avoid static options and parameters:** If parameters are unavoidable (e.g., maximum packet size), it is best to have the sender and receiver negotiate a value than defining fixed choices.
7. **Look for a good design; it need not be perfect:** Often the designers have a good design but it cannot handle some weird special case. Rather than messing up the design, the designers should go with the good design and put the burden of working around it on the people with the strange requirements.
8. **Be strict when sending and tolerant when receiving:** In other words, only send packets that rigorously comply with the standards, but expect incoming packets that may not be fully conformant and try to deal with them.

9. **Think about scalability:** If the system is to handle millions of hosts and billions of users effectively, no centralized databases of any kind are tolerable and load must be spread as evenly as possible over the available resources.

10. **Consider performance and cost:** If a network has poor performance or outrageous costs, nobody will use it.

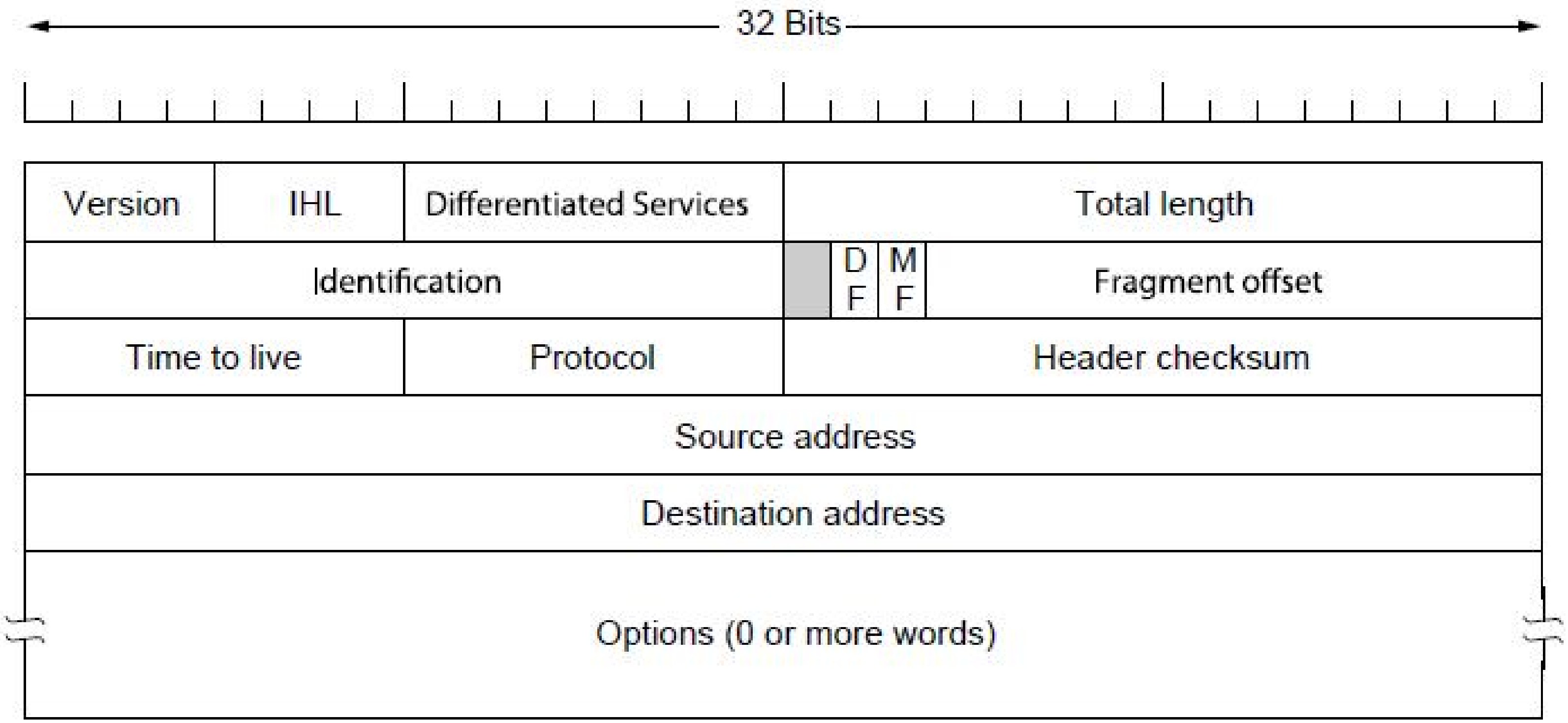
- Let us now leave the general principles and start looking at the details of the Internet's network layer. At the network layer, the Internet can be viewed as a collection of sub networks or Autonomous Systems (ASes) that are interconnected.
- There is no real structure, but several major backbones exist. These are constructed from high-bandwidth lines and fast routers. Attached to the backbones are regional (midlevel) networks, and attached to these regional networks are the LANs at many universities, companies, and Internet service providers.

The Network Layer in the Internet (2)



The Internet is an interconnected collection of many networks.

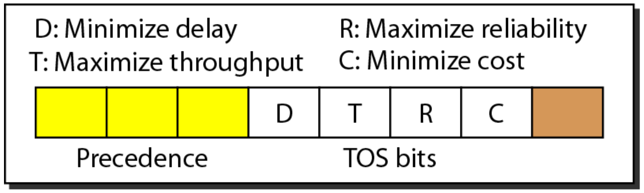
The IP Version 4 Protocol (1)



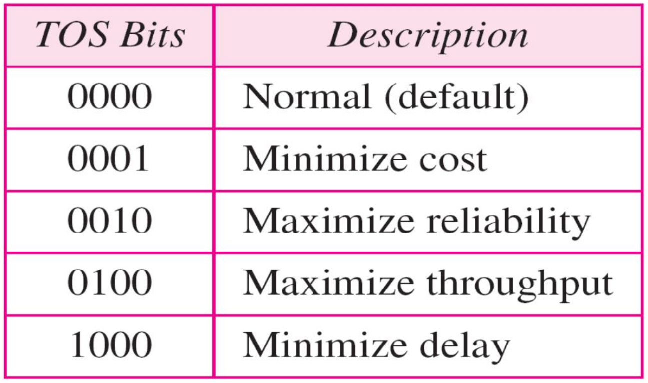
The IPv4 (Internet Protocol) header.

- **Version:(4 bits):** The Version field keeps track of which version of the protocol the datagram belongs to.
- **Internet Header Length(IHL 4 bits):** IHL, is provided to tell how long the header is, in 32-bit words. The minimum value is 5, which applies when no options are present. The maximum value of this 4-bit field is 15, which limits the header to 60 bytes, and thus the Options field to 40 bytes.
- **Type-of-Service (8 bits):** The Type of service field is one of the few fields that has changed its meaning (slightly) over the years. It was and is still intended to distinguish between different classes of service. Various combinations of reliability and speed are possible. For digitized voice, fast delivery beats accurate delivery. For file transfer, error-free transmission is more important than fast transmission.

- Originally, the 6-bit field contained (from left to right), a three-bit Precedence field and three flags, D, T, and R. The Precedence field was a priority, from 0 (normal) to 7 (network control packet). The three flag bits allowed the host to specify what it cared most about from the set {Delay, Throughput, Reliability}.



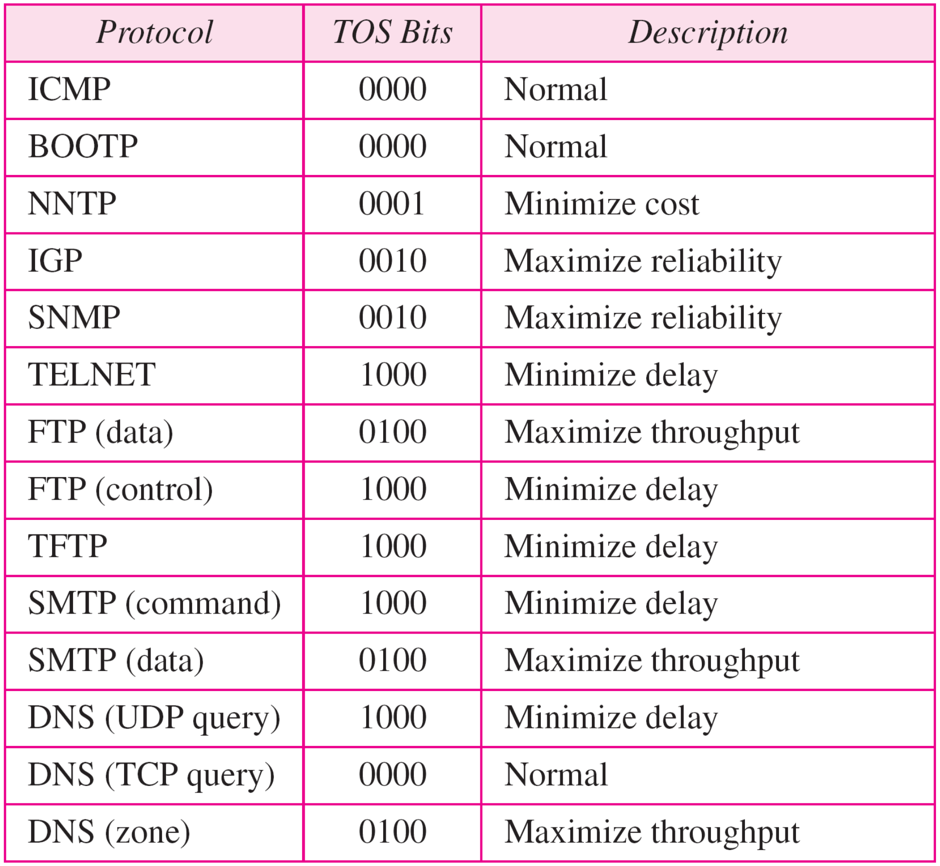
Service type



TOS Bits	*Description*
0000	Normal (default)
0001	Minimize cost
0010	Maximize reliability
0100	Maximize throughput
1000	Minimize delay

- **Total length(16 bits):** The Total length includes everything in the datagram—both header and data. The maximum length is 65,535 bytes. At present, this upper limit is tolerable, but with future gigabit networks, larger data grams may be needed.

Default TOS for Applications



Protocol	*TOS Bits*	*Description*
ICMP	0000	Normal
BOOTP	0000	Normal
NNTP	0001	Minimize cost
IGP	0010	Maximize reliability
SNMP	0010	Maximize reliability
TELNET	1000	Minimize delay
FTP (data)	0100	Maximize throughput
FTP (control)	1000	Minimize delay
TFTP	1000	Minimize delay
SMTP (command)	1000	Minimize delay
SMTP (data)	0100	Maximize throughput
DNS (UDP query)	1000	Minimize delay
DNS (TCP query)	0000	Normal
DNS (zone)	0100	Maximize throughput

- **Identifier (16 bits):** To have a proper reassembling of fragments , The Identification field is needed to allow the destination host to determine which datagram a newly arrived fragment belongs to. All the fragments of a datagram contain the same Identification value.
- **Flags(3 bits):** First one an unused bit Only two of the bits are currently defined: MF(More Fragments) ,DF(Don't Fragment)

  MF(More Fragments) :When a receiving host sees a packet arrive with the MF = 1, it examines the Fragment Offset to see where this fragment is to be placed in the reconstructed packet.

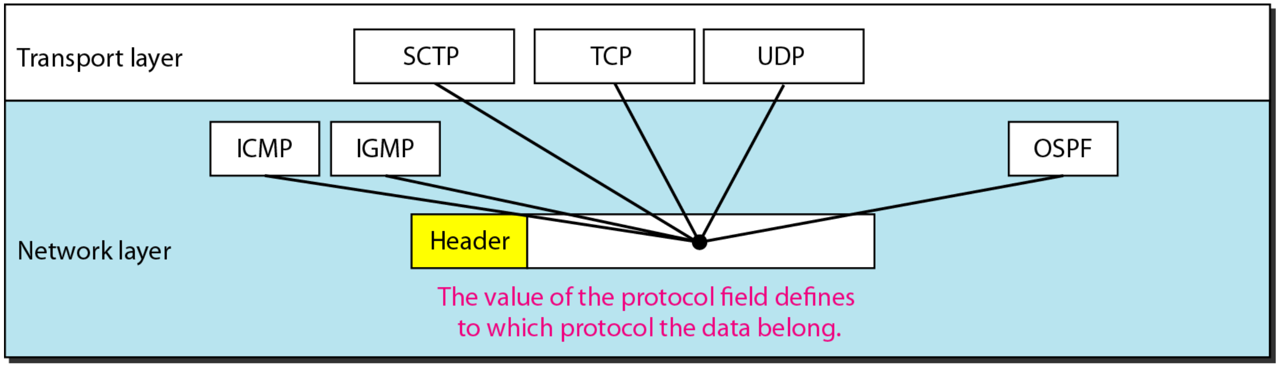
  Don't Fragment flag (DF):The Don't Fragment (DF) flag is a single bit in the Flag field that indicates that fragmentation of the packet is not allowed.

- **Fragment offset:** The Fragment offset tells where in the current datagram this fragment belongs.
- **Time-to-Live (TTL) (8 bits):** The Time to live field is a counter used to limit packet lifetimes. It is supposed to count time in seconds, allowing a maximum lifetime of 255 sec. It must be decremented on each hop and is supposed to be decremented multiple times when queued for a long time in a router. In practice, it just counts hops. When it hits zero, the packet is discarded and a warning packet is sent back to the source host..

- **Protocol (8 bits):** The Protocol field tells it which transport process to give it to. TCP is one possibility, but so are UDP and some others.
- **Header checksum (16 bits):** The Header checksum verifies the header only. Such a checksum is useful for detecting errors generated by bad memory words inside a router.

IPv4 Header

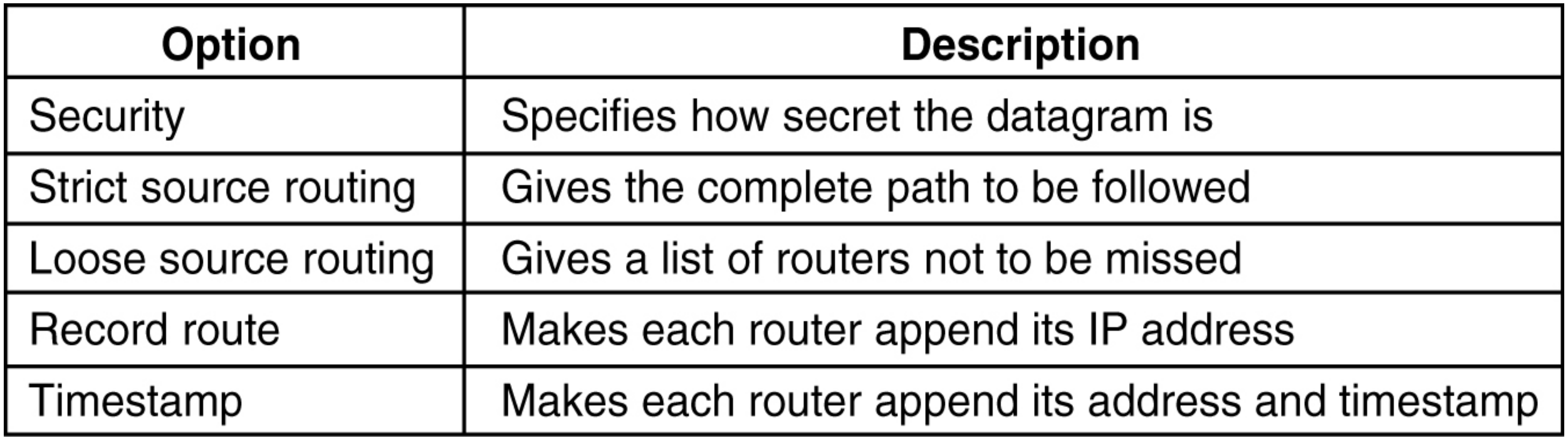
- Protocol field for higher-level protocol



The value of the protocol field defines to which protocol the data belong.

Value	*Protocol*
1	ICMP
2	IGMP
6	TCP
17	UDP
89	OSPF

- **IP Destination Address (32 bits):** The IP Destination Address field contains a 32-bit binary value that represents the packet destination Network layer host address.
- **IP Source Address (32 bits):** The IP Source Address field contains a 32-bit binary value that represents the packet source Network layer host address.
- **Options (variable):** The Options field is padded out to a multiple of four bytes. Originally, five options were defined The current complete list is now maintained on-line at www.iana.org/assignments/ip-parameters.



Option	Description
Security	Specifies how secret the datagram is
Strict source routing	Gives the complete path to be followed
Loose source routing	Gives a list of routers not to be missed
Record route	Makes each router append its IP address
Timestamp	Makes each router append its address and timestamp

- The Security option tells how secret the information is.
- The Strict source routing option gives the complete path from source to destination as a sequence of IP addresses. The datagram is required to follow that exact route.

- The Loose source routing option requires the packet to traverse the list of routers specified, and in the order specified, but it is allowed to pass through other routers on the way.
- The Record route option tells the routers along the path to append their IP address to the option field. This allows system managers to track down bugs in the routing algorithms.
- Finally, the Timestamp option is like the Record route option, except that in addition to recording its 32-bit IP address, each router also records a 32-bit timestamp. This option, too, is mostly for debugging routing algorithms.

What is an IP Address?

- An IP address is a unique global address for a network interface
- It is a **32 bit long** identifier
- An IP address contains two parts:

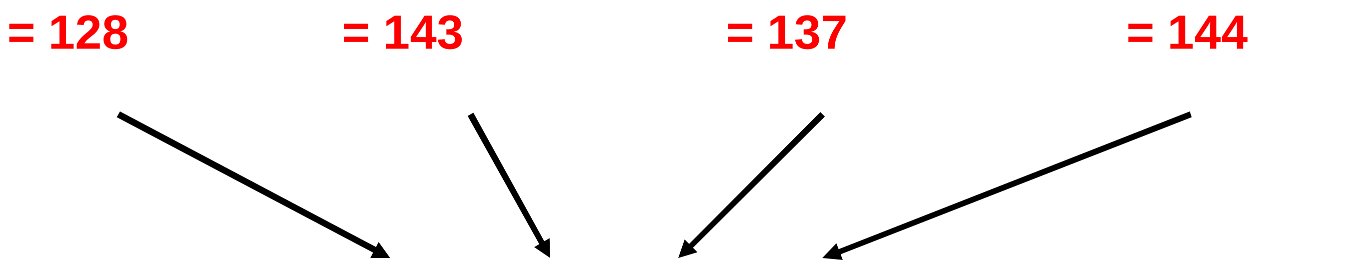
  - network number (**network prefix**)

  - **host number**

Dotted Decimal Notation

- IP addresses are written in a so-called ***dotted decimal*** **notation**
- Each byte is identified by a decimal number in the range [0..255]:
- **Example:**

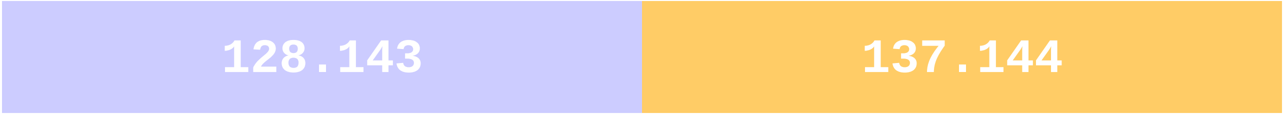
10000000	10001111	10001001	10010000



= 128 **= 143** **= 137** **= 144**

Example

- **Example**:



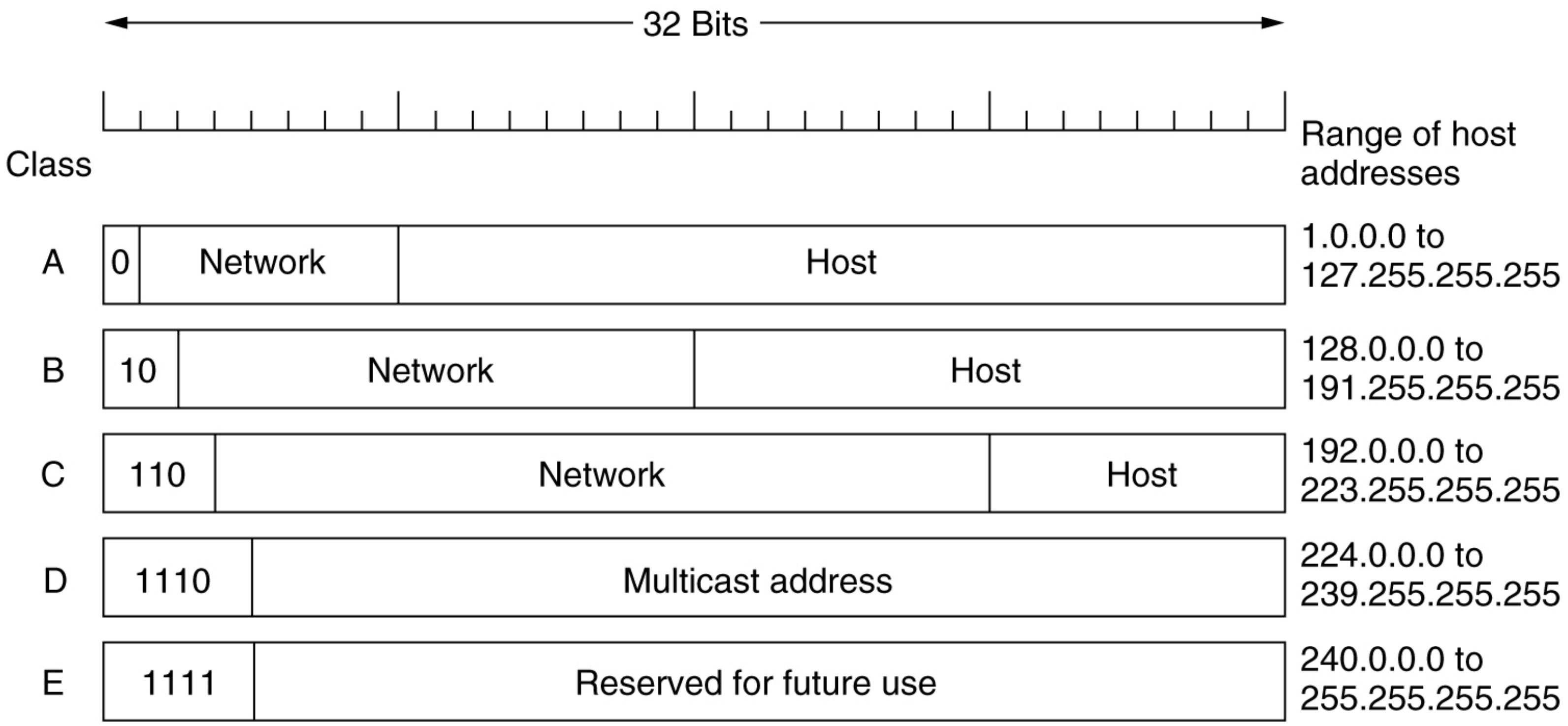
128.143 137.144

- Network id is: **128.143**
- Host number is: **137.144**
- Prefix notation: **128.143.137.144**
  - » Network prefix is 16 bits long

Class full IP Addresses

- The Internet address space was divided up into classes:
- **Class A addressing –** Allow 128 networks and 16 millions hosts .
- **Class B addressing –** Allows 16,384 networks with 65,534 hosts.
- **Class C addressing** – 2 million networks with 254 hosts.
- **Class D addressing-** number of groups are 2^28 million groups
- **Class E addressing:** Future purpose

IP Addresses

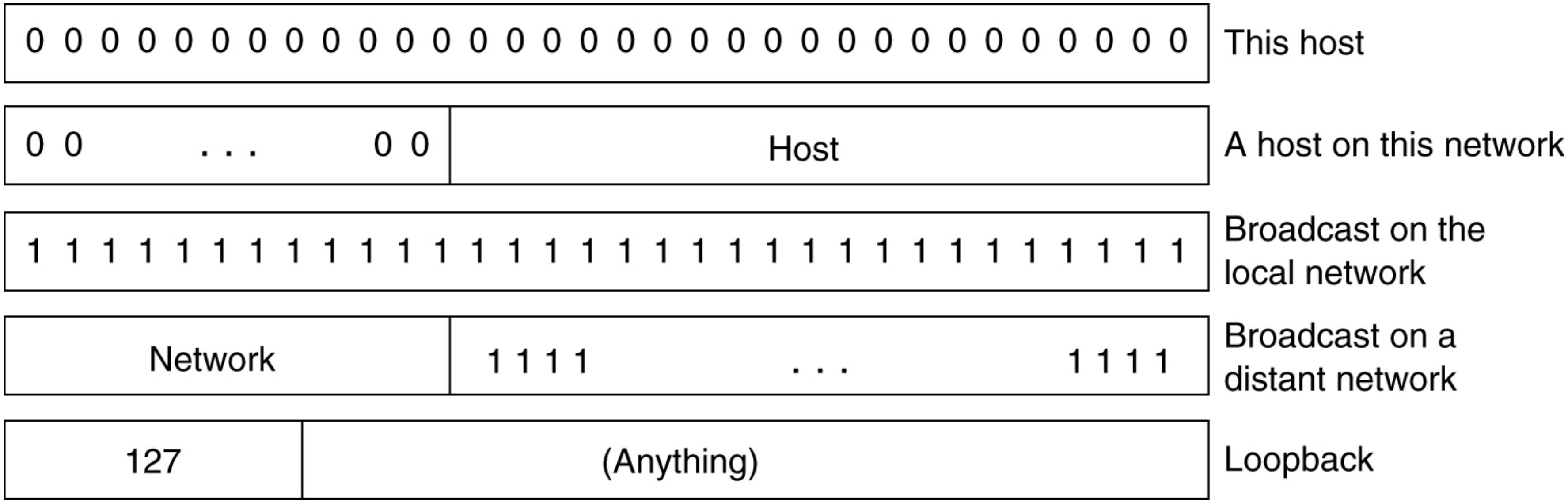


32 Bits

Class				Range of host addresses
A	0	Network	Host	1.0.0.0 to 127.255.255.255
B	10	Network	Host	128.0.0.0 to 191.255.255.255
C	110	Network	Host	192.0.0.0 to 223.255.255.255
D	1110	Multicast address		224.0.0.0 to 239.255.255.255
E	1111	Reserved for future use		240.0.0.0 to 255.255.255.255

IP address formats.

IP Addresses (2)

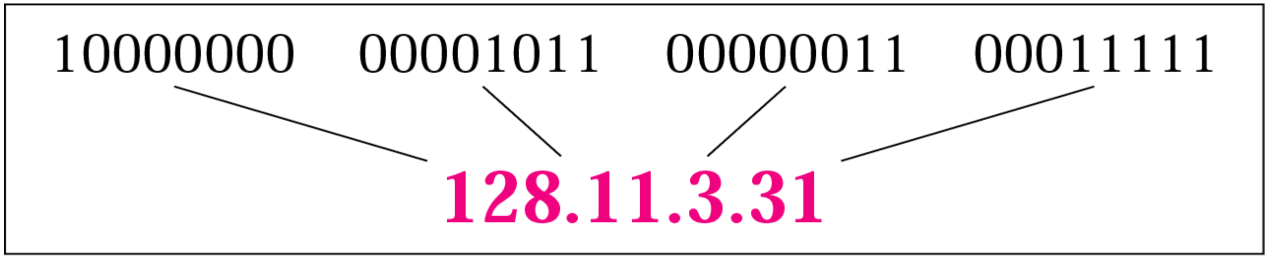


0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	This host
0 0 . . . 0 0 | Host	A host on this network
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	Broadcast on the local network
Network | 1 1 1 1 . . . 1 1 1 1	Broadcast on a distant network
127 | (Anything)	Loopback

Special IP addresses.

IPv4 Addresses

- An IP address is a 32-bits long
- The IP addresses are unique and universal
- The address space of IPv4 is 2^{32} or 4,294,967,296
- Binary notation: 01110101 10010101 00011101 00000010
- Dotted-decimal notation: 117.149.29.2



10000000 00001011 00000011 00011111

128.11.3.31

Example

- Change the following IP addresses from binary notation to dotted-decimal notation.

  a. 10000001 00001011 00001011 11101111

  b. 11111001 10011011 11111011 00001111

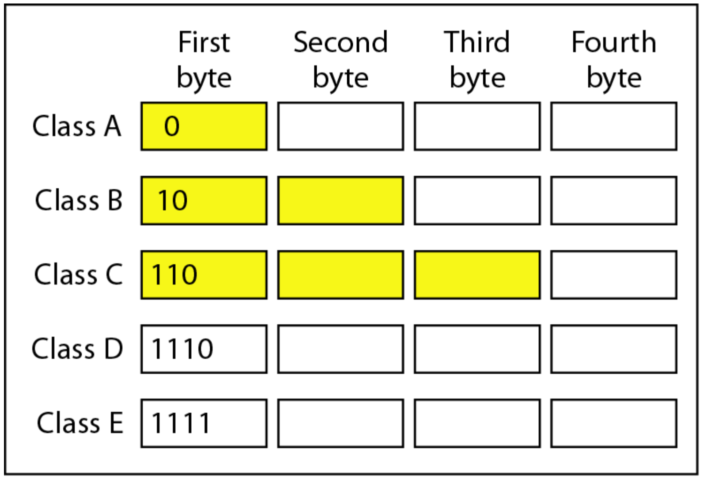
We replace each group of 8 bits with its equivalent decimal number and add dots for separation:

a. 129.11.11.239

b. 249.155.251.15

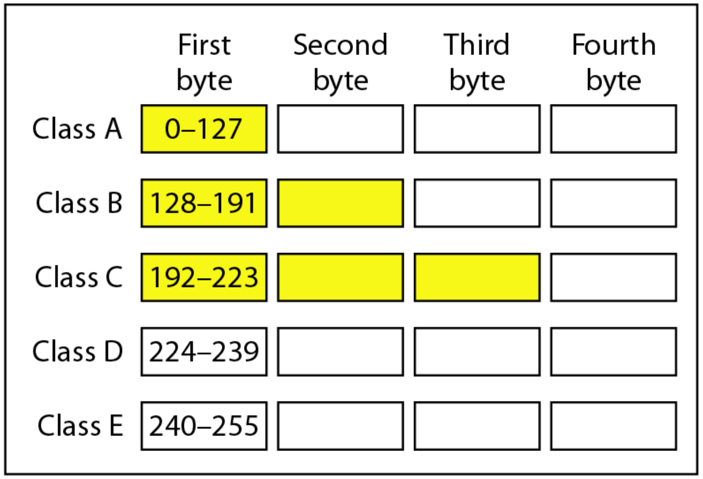
Classful addressing

- In classful addressing, the address space is divided into five classes: A, B, C, D, E
- A new architecture, called classless addressing was introduced in the mid-1990s



	First byte	Second byte	Third byte	Fourth byte
Class A	0			
Class B	10	🟨		
Class C	110	🟨	🟨	
Class D	1110			
Class E	1111			

a. Binary notation



	First byte	Second byte	Third byte	Fourth byte
Class A	0–127			
Class B	128–191	🟨		
Class C	192–223	🟨	🟨	
Class D	224–239			
Class E	240–255			

b. Dotted-decimal notation

Classful Addressing: Example

- Find the class of each address.
  a. 00000001 00001011 00001011 11101111
  b. 11000001 10000011 00011011 11111111
  c. 14.23.120.8
  d. 252.5.15.111

- Solution
  a. The first bit is 0. This is a class A address.
  b. The first 2 bits are 1; the third bit is 0. This is a class C address.
  c. The first byte is 14; the class is A.
  d. The first byte is 252; the class is E.

Network & Host Identification

Circle the network portion of these addresses:

177.100.18.4

119.18.45.0

209.240.80.78

199.155.77.56

Circle the host portion of these addresses:

10.15.123.50

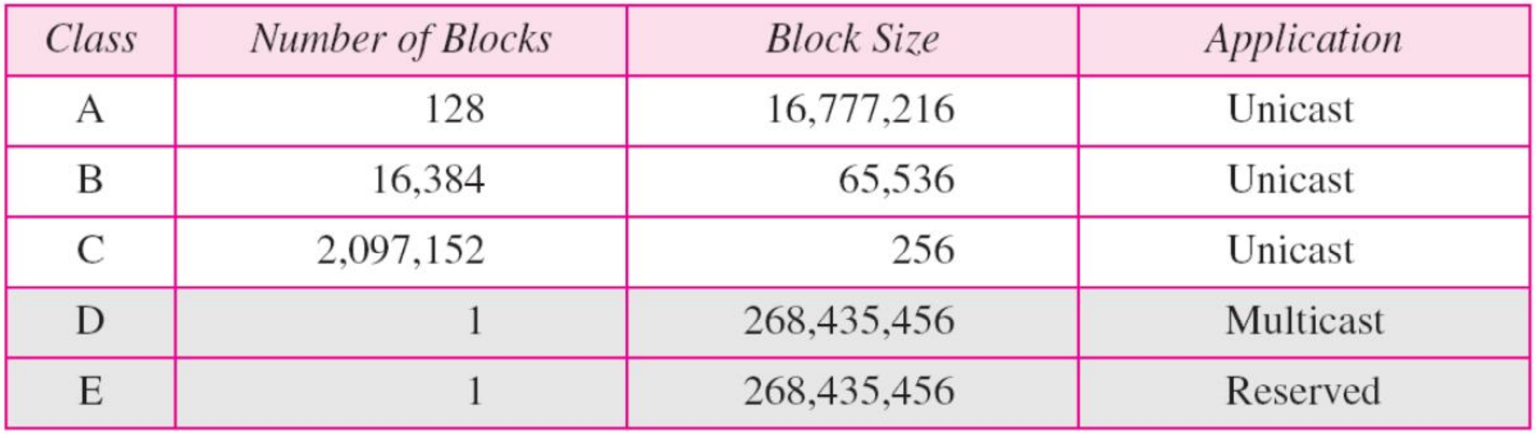
171.2.199.31

198.125.87.177

223.250.200.222

Classes and Blocks

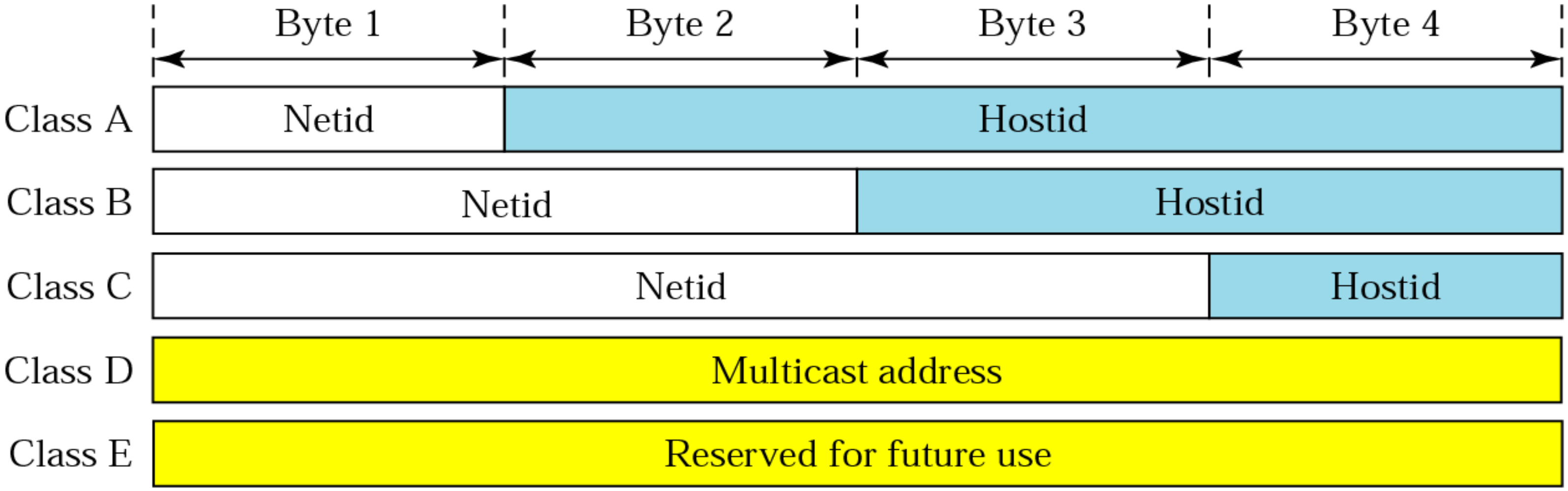
- In classful addressing, a large part of the available addresses were wasted



Class	*Number of Blocks*	*Block Size*	*Application*
A	128	16,777,216	Unicast
B	16,384	65,536	Unicast
C	2,097,152	256	Unicast
D	1	268,435,456	Multicast
E	1	268,435,456	Reserved

Netid and Hostid

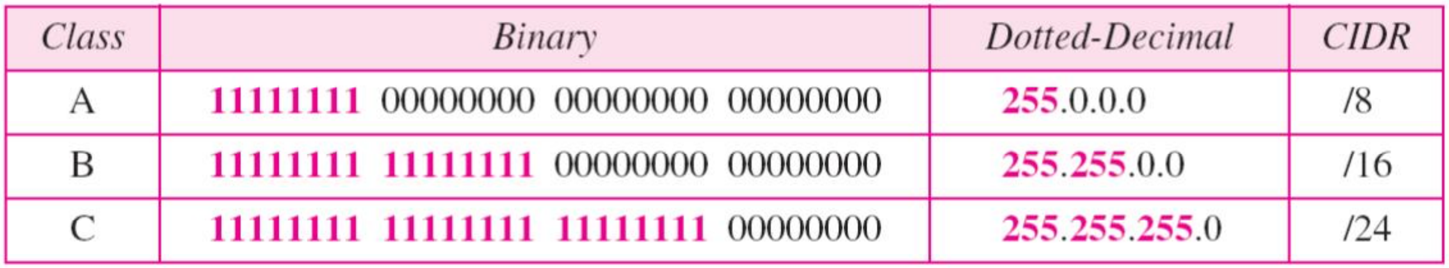
- IP address in classes A, B, and C is divided into **netid** and **hostid**



	Byte 1	Byte 2	Byte 3	Byte 4
Class A	Netid	Hostid		
Class B	Netid		Hostid	
Class C	Netid			Hostid
Class D	Multicast address			
Class E	Reserved for future use			

Mask: Default Mask

- The length of the netid and hostid is predetermined in classful addressing
- **Default masking**
- CIDR (Classless Interdomain Routing) notation



Class	*Binary*	*Dotted-Decimal*	*CIDR*
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

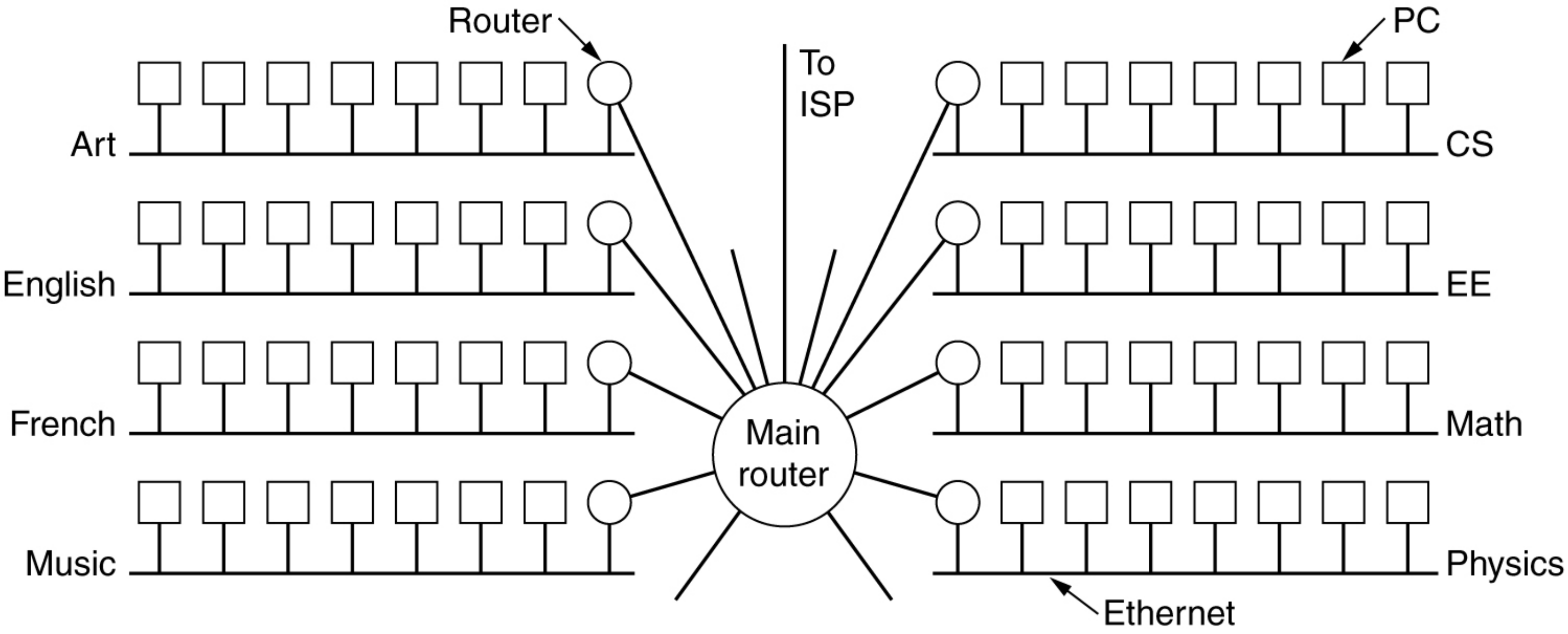
Subnetting

- Divide a large block of addresses into several contiguous groups and assign each group to smaller networks called subnets
- Increase the number of 1s in the mask

Supernetting

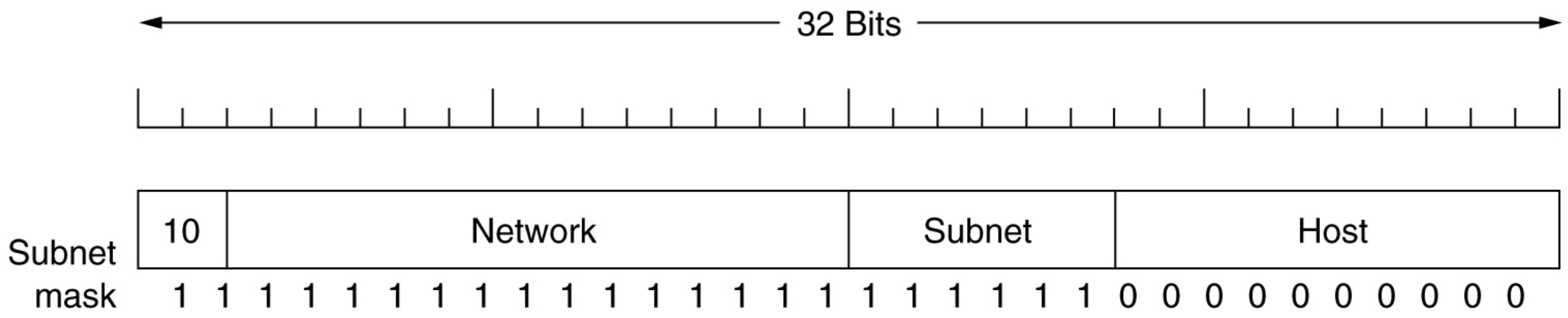
- Combine several class C blocks to create a larger range of addresses
- Decrease the number of 1s in the mask (/24 → /22 for C addresses)

Subnets

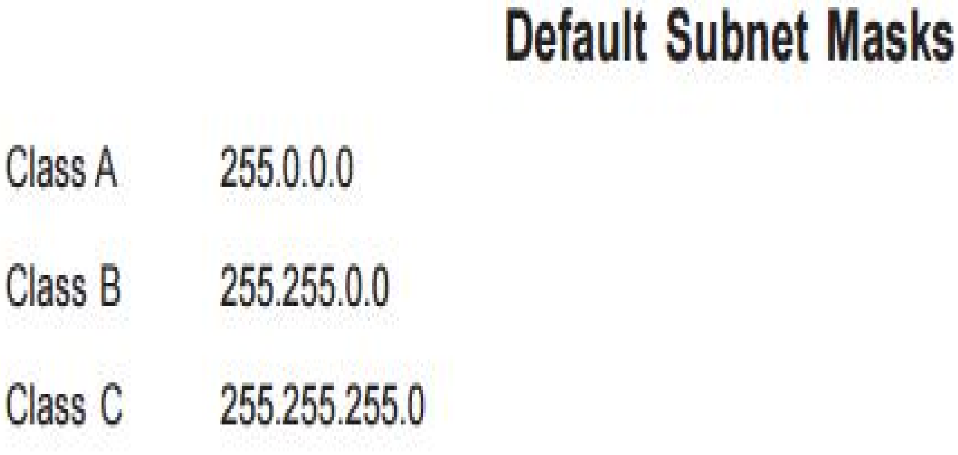


A campus network consisting of LANs for various departments.

Subnets (2)



A class B network subnetted into 64 subnets.

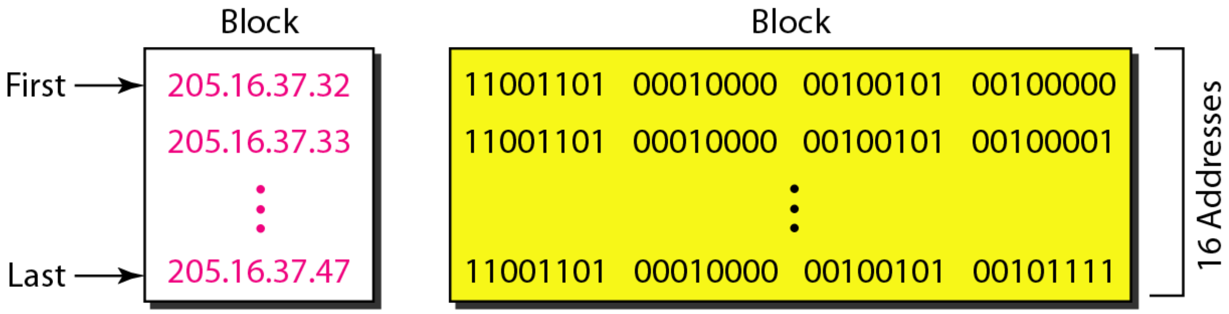


Default Subnet Masks

Class A	255.0.0.0
Class B	255.255.0.0
Class C	255.255.255.0

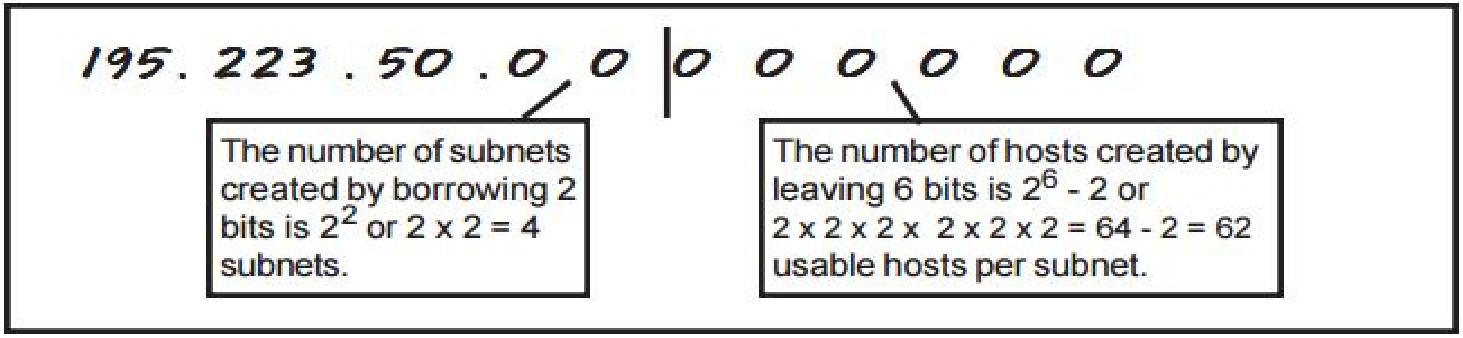
Classless addressing: CIDR

- Classful addressing has created many problems
- Many ISPs and service users need more addresses
- Idea is to have variable-length blocks that belong to no class
- Three restrictions on classless address blocks;
  - The addresses in a block must be contiguous, one after another
  - The number of addresses in a block must be a power of 2
  - The first address must be evenly divisible by the number of addresses



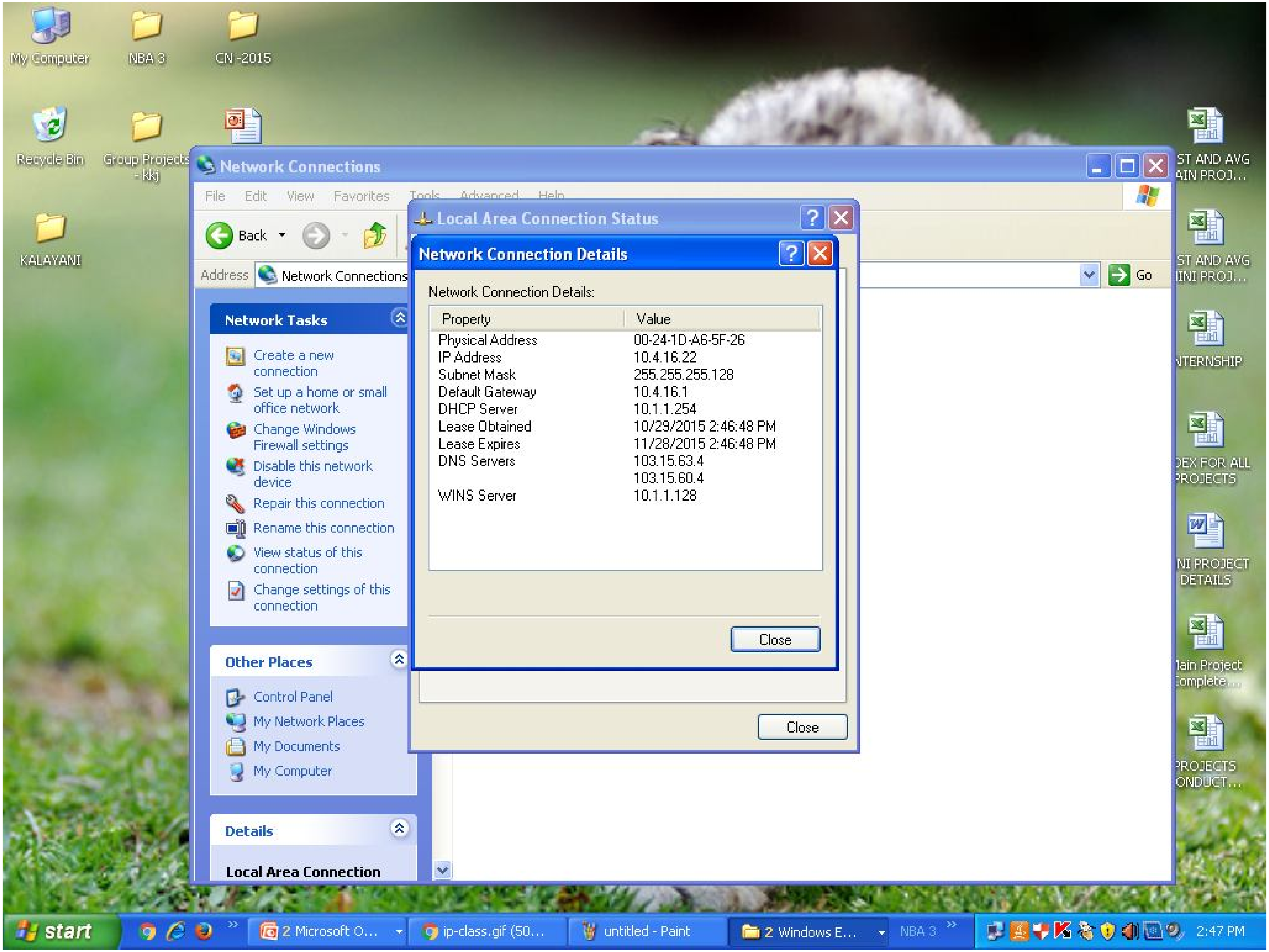
a. Decimal

b. Binary



The number of subnets created by borrowing 2 bits is 2^2 or 2 x 2 = 4 subnets.

The number of hosts created by leaving 6 bits is 2^6 - 2 or 2 x 2 x 2 x 2 x 2 x 2 = 64 - 2 = 62 usable hosts per subnet.



Network Connection Details

Network Connection Details:

Property	Value
Physical Address	00-24-1D-A6-5F-26
IP Address	10.4.16.22
Subnet Mask	255.255.255.128
Default Gateway	10.4.16.1
DHCP Server	10.1.1.254
Lease Obtained	10/29/2015 2:46:48 PM
Lease Expires	11/28/2015 2:46:48 PM
DNS Servers	103.15.63.4
	103.15.60.4
WINS Server	10.1.1.128

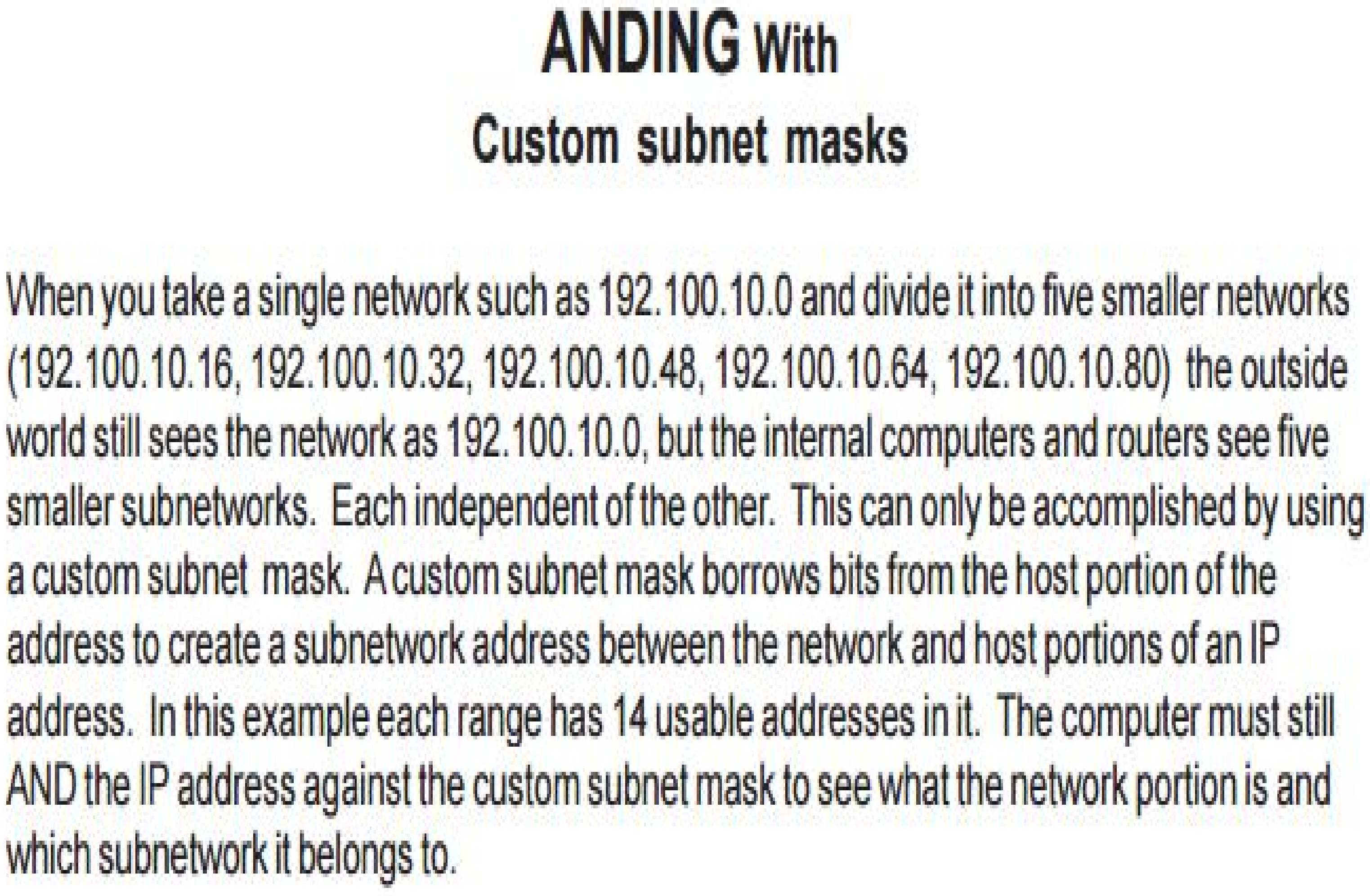
Mask and Address Blocks

- In IPv4 addressing, a block of addresses can be defined as x.y.z.t /*n* in which x.y.z.t defines one of the addresses and the /*n* defines the mask.
- The first address in the block can be found by setting the rightmost $32 - n$ bits to 0s
- The last address in the block can be found by setting the rightmost $32 - n$ bits to 1s
- The number of addresses in the block can be found by using the formula 2^{32-n}

- Example: 205.16.37.39/28
  - The binary representation is 1100110 00010000 00100101 00100111
  - If we set $32 - 28$ rightmost bits to 0, we get 11001101 00010000 00100101 00100000

    → 205.16.37.32 (First address)
  - If we set $32 - 28$ rightmost bits to 1, we get 11001101 00010000 00100101 00101111

    → 205.16.37.47 (Last address)
  - The value of n is 28, which means that number of addresses is 2^{32-28} or 16



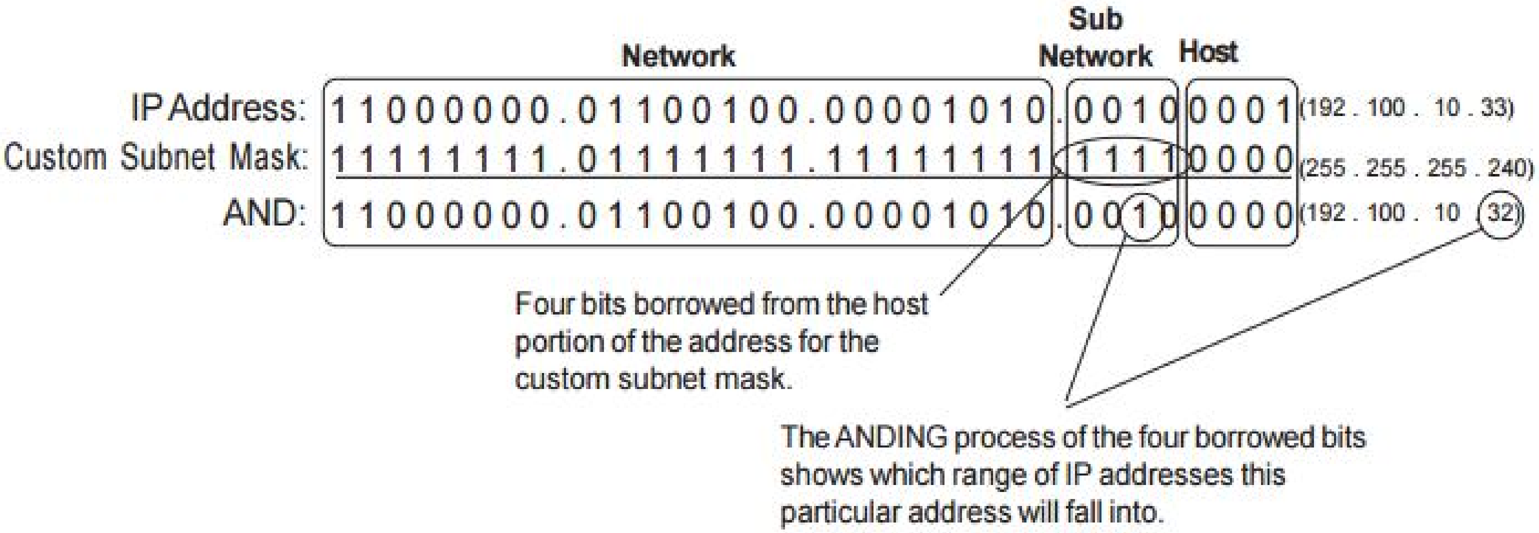
ANDING With

Custom subnet masks

When you take a single network such as 192.100.10.0 and divide it into five smaller networks (192.100.10.16, 192.100.10.32, 192.100.10.48, 192.100.10.64, 192.100.10.80) the outside world still sees the network as 192.100.10.0, but the internal computers and routers see five smaller subnetworks. Each independent of the other. This can only be accomplished by using a custom subnet mask. A custom subnet mask borrows bits from the host portion of the address to create a subnetwork address between the network and host portions of an IP address. In this example each range has 14 usable addresses in it. The computer must still AND the IP address against the custom subnet mask to see what the network portion is and which subnetwork it belongs to.

IP Address: 192 . 100 . 10 . 0
Custom Subnet Mask: 255.255.255.240

Address Ranges:
192.10.10.0 to 192.100.10.15
192.100.10.16 to 192.100.10.31
192.100.10.32 to 192.100.10.47 (Range in the sample below)
192.100.10.48 to 192.100.10.63
192.100.10.64 to 192.100.10.79
192.100.10.80 to 192.100.10.95
192.100.10.96 to 192.100.10.111
192.100.10.112 to 192.100.10.127
192.100.10.128 to 192.100.10.143
192.100.10.144 to 192.100.10.159
192.100.10.160 to 192.100.10.175
192.100.10.176 to 192.100.10.191
192.100.10.192 to 192.100.10.207
192.100.10.208 to 192.100.10.223
192.100.10.224 to 192.100.10.239
192.100.10.240 to 192.100.10.255



IP Address: 1 1 0 0 0 0 0 0 . 0 1 1 0 0 1 0 0 . 0 0 0 0 1 0 1 0 . 0 0 1 0 0 0 0 1 (192 . 100 . 10 . 33)
Custom Subnet Mask: 1 1 1 1 1 1 1 1 . 0 1 1 1 1 1 1 1 . 1 1 1 1 1 1 1 1 . 1 1 1 1 0 0 0 0 (255 . 255 . 255 . 240)
AND: 1 1 0 0 0 0 0 0 . 0 1 1 0 0 1 0 0 . 0 0 0 0 1 0 1 0 . 0 0 1 0 0 0 0 0 (192 . 100 . 10 32)

Four bits borrowed from the host portion of the address for the custom subnet mask.

The ANDING process of the four borrowed bits shows which range of IP addresses this particular address will fall into.

How to determine the number of subnets and the number of hosts per subnet?

- From 8 bits if 3 is allocated for subnet and 5 for host.

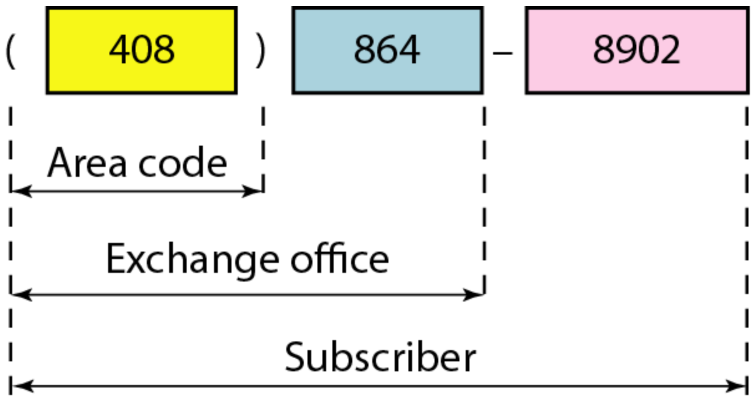
For example if you borrow three bits from the host portion of the address use the number of subnets formula to determine the total number of subnets gained by borrowing the three bits. This would be 2^3 or 2 x 2 x 2 = 8 subnets.

To determine the number of hosts per subnet you would take the number of binary bits used in the host portion and apply this to the number of hosts per subnet formula If five bits are in the host portion of the address this would be 2^5 or 2 x 2 x 2 x 2 x 2 = 32 hosts. Usable host are 32-2=30.

Network Address

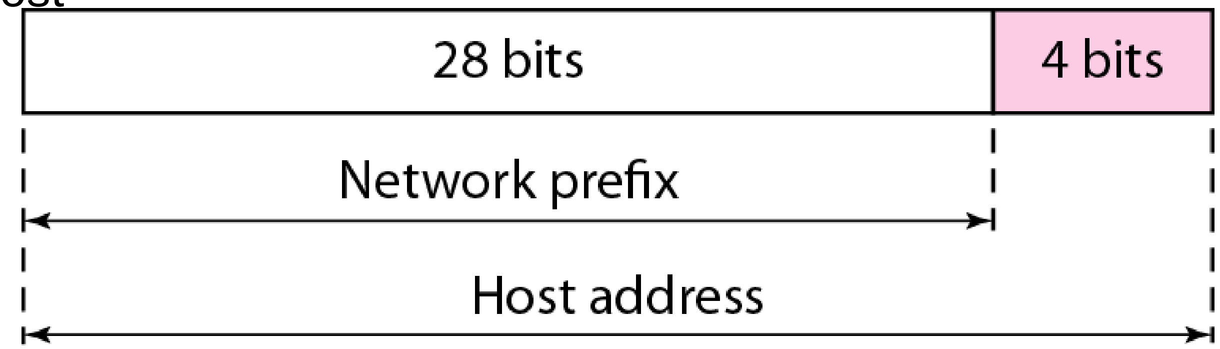
- The first address in a block is normally not assigned to any device; it is used as the network address that represents the organization to the rest of the world

Hierarchy

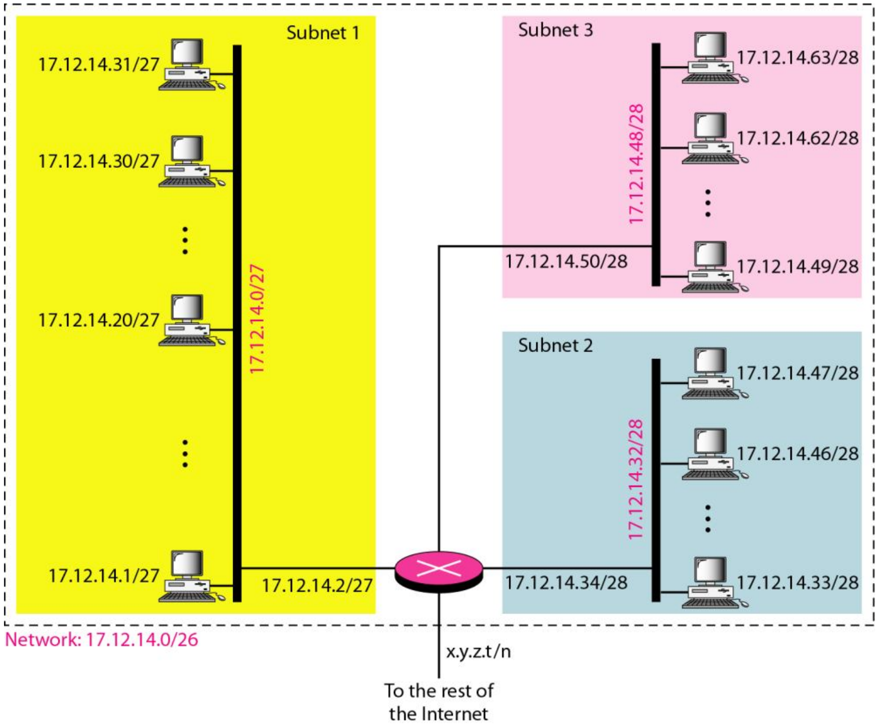


Two-Level Hierarchy: No Subnetting

- Each address in the block can be considered as a two-level hierarchical structure:
  the leftmost *n* bits (prefix) define the network; the rightmost 32 − n bits define
  the host



Three-Levels of Hierarchy: Subnetting



IP Address Classes

Class A	1 – 127	(Network 127 is reserved for loopback and internal testing) Leading bit pattern	0	00000000.00000000.00000000.00000000 Network . Host . Host . Host
Class B	128 – 191	Leading bit pattern	10	10000000.00000000.00000000.00000000 Network . Network . Host . Host
Class C	192 – 223	Leading bit pattern	110	11000000.00000000.00000000.00000000 Network . Network . Network . Host
Class D	224 – 239	(Reserved for multicast)		
Class E	240 – 255	(Reserved for experimental, used for research)		

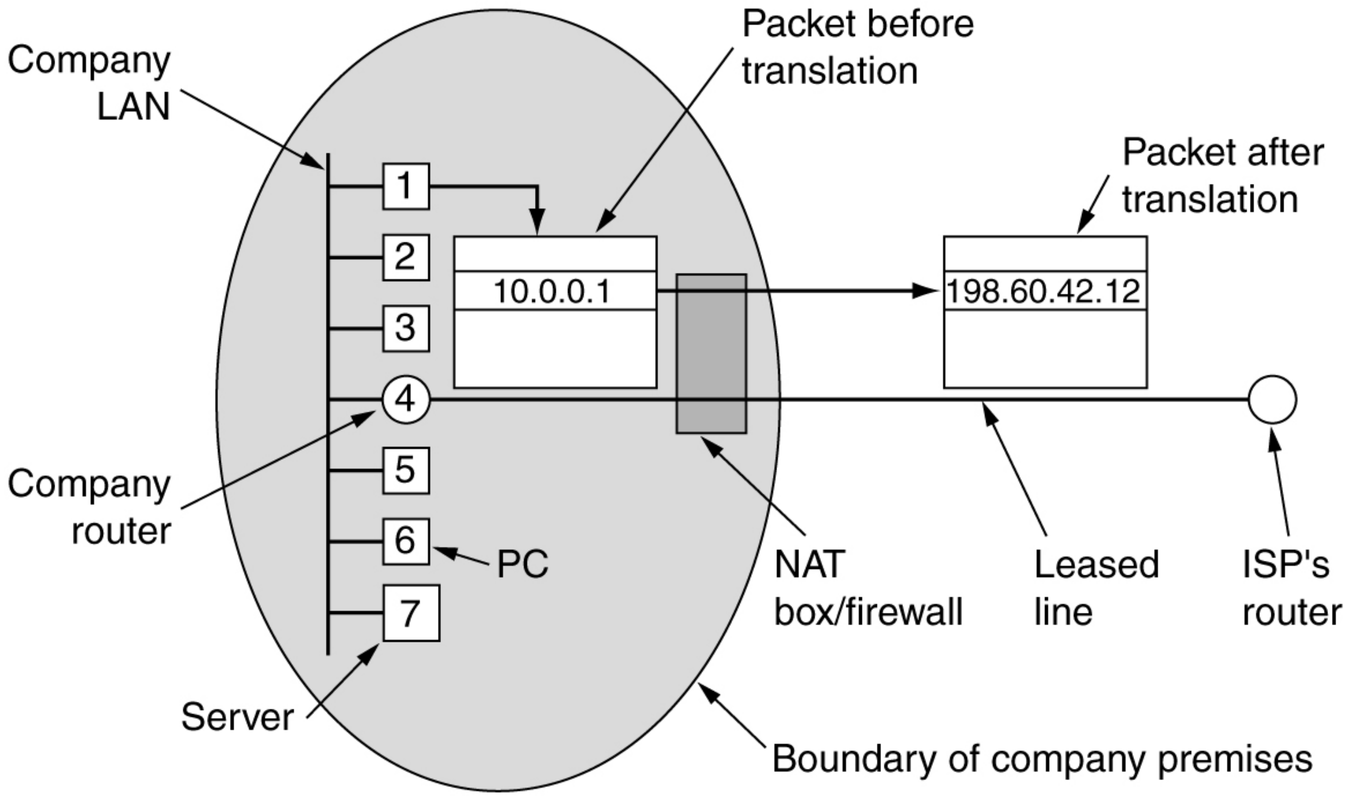
Private Address Space

Class A	10.0.0.0 to 10.255.255.255
Class B	172.16.0.0 to 172.31.255.255
Class C	192.168.0.0 to 192.168.255.255

Network Address Translation

- RFC-1631
- A short term solution to the problem of the depletion of IP addresses
  - Long term solution is IP v6 (or whatever is finally agreed on)
  - CIDR (Classless InterDomain Routing) is a possible short term solution
  - NAT is another
- NAT is a way to conserve IP addresses
  - Hide a number of hosts behind a single IP address
  - Use:
    - 10.0.0.0-10.255.255.255,
    - 172.16.0.0-172.32.255.255 or
    - 192.168.0.0-192.168.255.255 for local networks

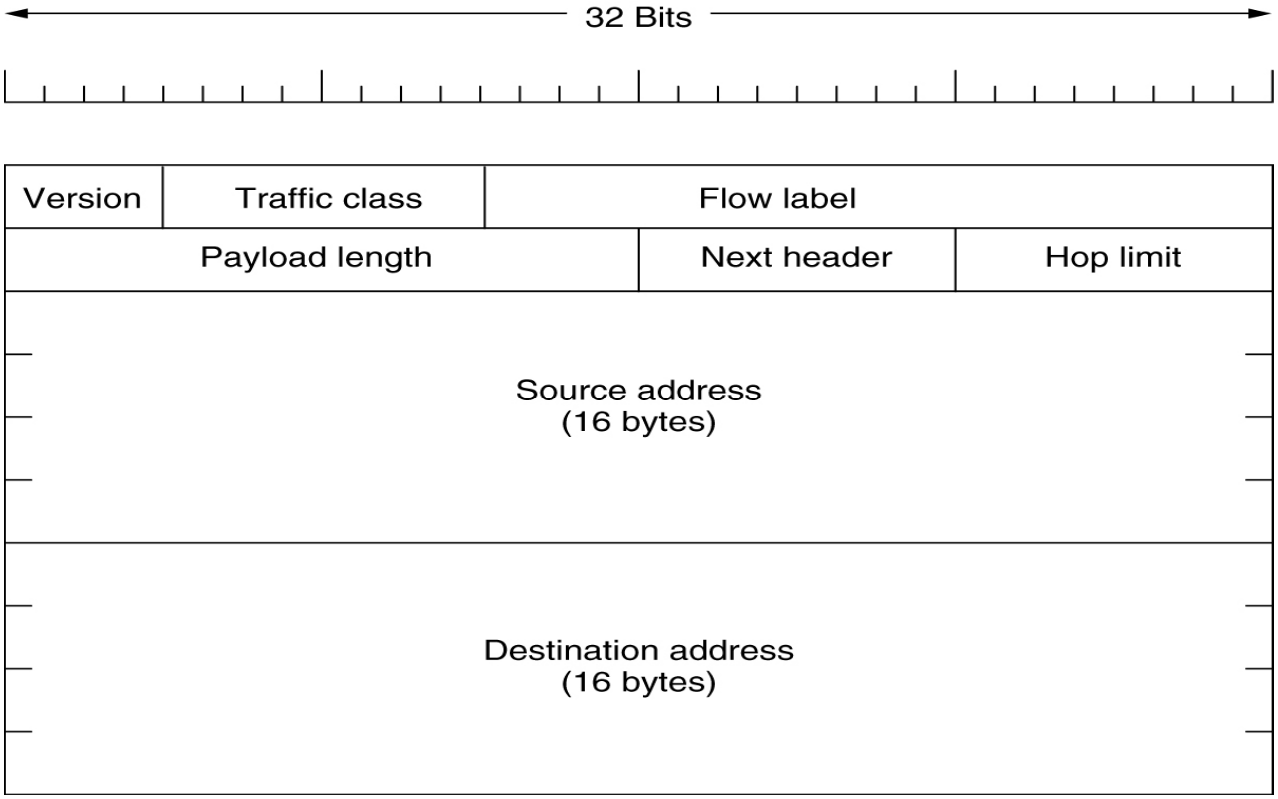
NAT – Network Address Translation



Placement and operation of a NAT box.

IPv6

The Main IPv6 Header



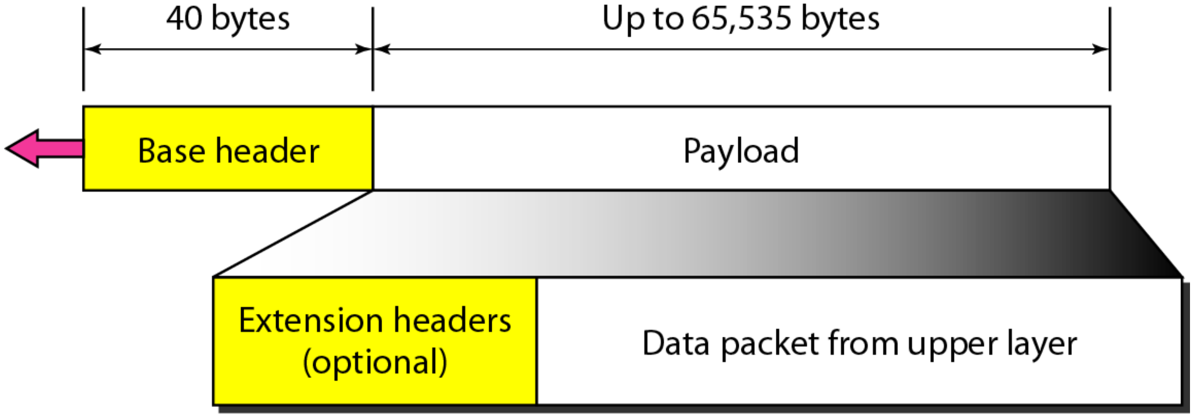
The IPv6 fixed header (required).

IPv6 address

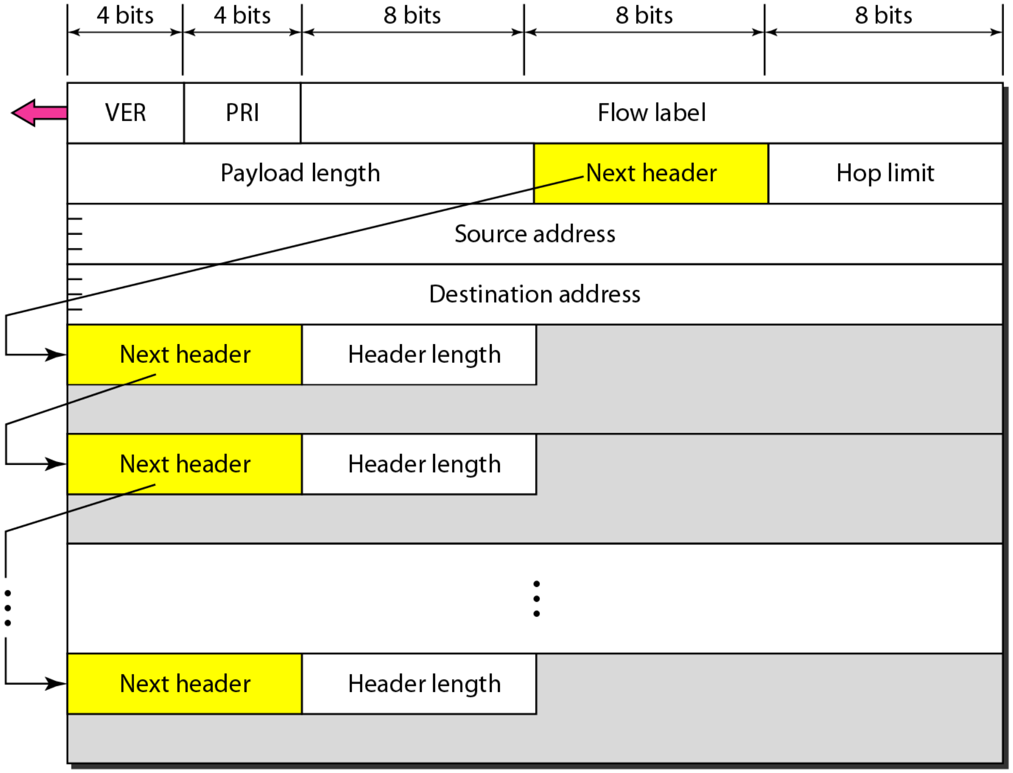
- The use of address space is inefficient
- Minimum delay strategies and reservation of resources are required to accommodate real-time audio and video transmission
- No security mechanism (encryption and authentication) is provided
- IPv6 (IPng: Internetworking Protocol, next generation)
  - Larger address space (128 bits)
  - Better header format
  - New options
  - Allowance for extention
  - Support for resource allocation: flow label to enable the source to request special handling of the packet
  - Support for more security

IPv6 Datagram

- IPv6 defines three types of addresses: unicast, anycast (a group of computers with the same prefix address), and multicast
- IPv6 datagram header and payload



IPv6 Datagram Format

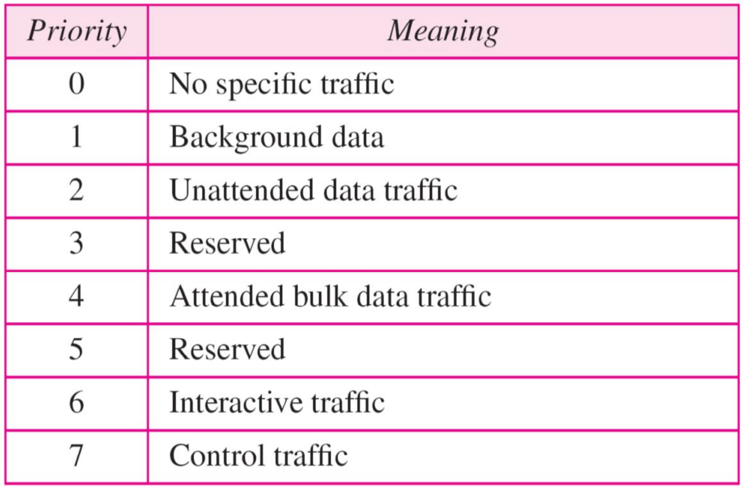


IPv6 Header

- Version: IPv6
- Priority (4 bits): the priority of the packet with respect to traffic congestion
- Flow label (3 bytes): to provide special handling for a particular flow of data
- Payload length
- Next header (8 bits): to define the header that follows the base header in the datagram
- Hop limit: TTL in IPv4
- Source address (16 bytes) and destination address (16 bytes): if source routing is used, the destination address field contains the address of the next router

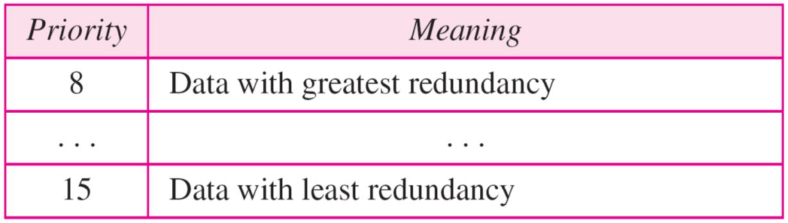
Priority

- IPv6 divides traffic into two broad categories: congestion-controlled and noncongestion-controlled
- Congestion-controlled traffic



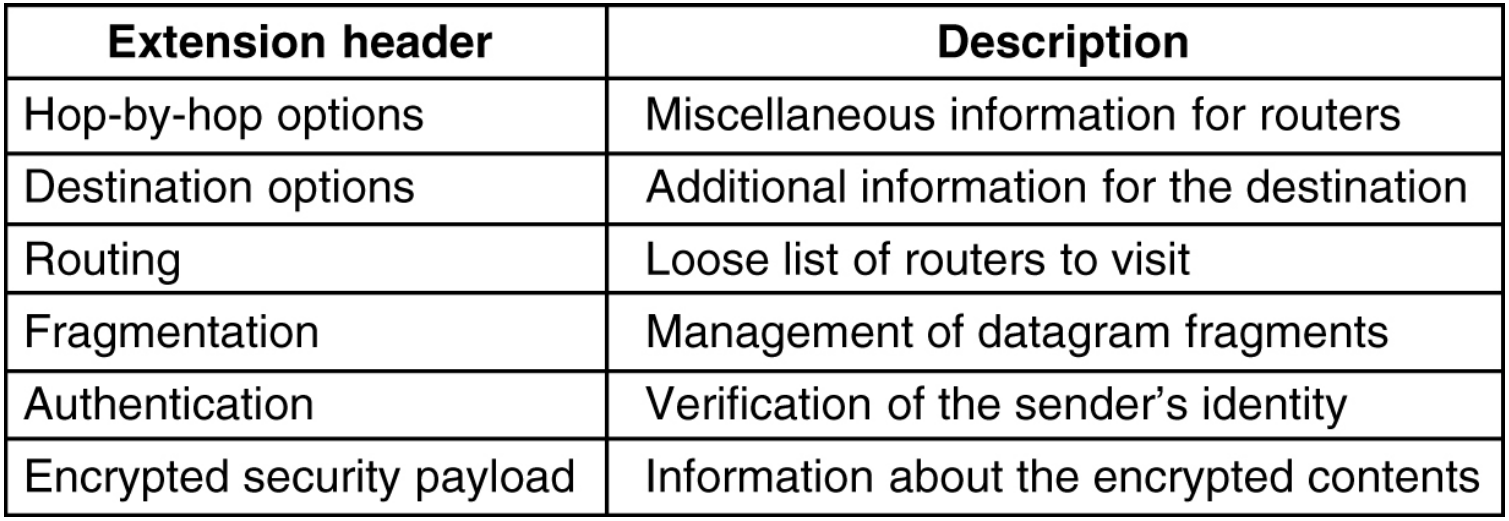
Priority	*Meaning*
0	No specific traffic
1	Background data
2	Unattended data traffic
3	Reserved
4	Attended bulk data traffic
5	Reserved
6	Interactive traffic
7	Control traffic

- Noncongestion-controlled traffic



Priority	*Meaning*
8	Data with greatest redundancy
. . .	. . .
15	Data with least redundancy

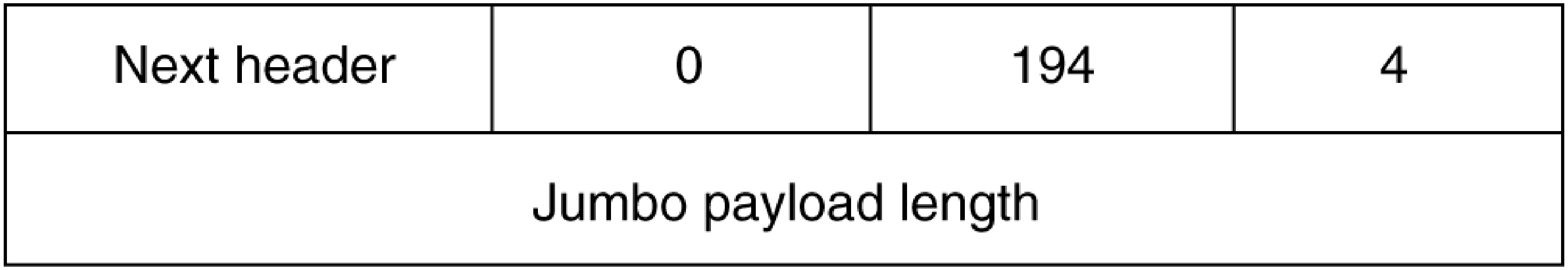
Extension Headers



Extension header	Description
Hop-by-hop options	Miscellaneous information for routers
Destination options	Additional information for the destination
Routing	Loose list of routers to visit
Fragmentation	Management of datagram fragments
Authentication	Verification of the sender’s identity
Encrypted security payload	Information about the encrypted contents

IPv6 extension headers.

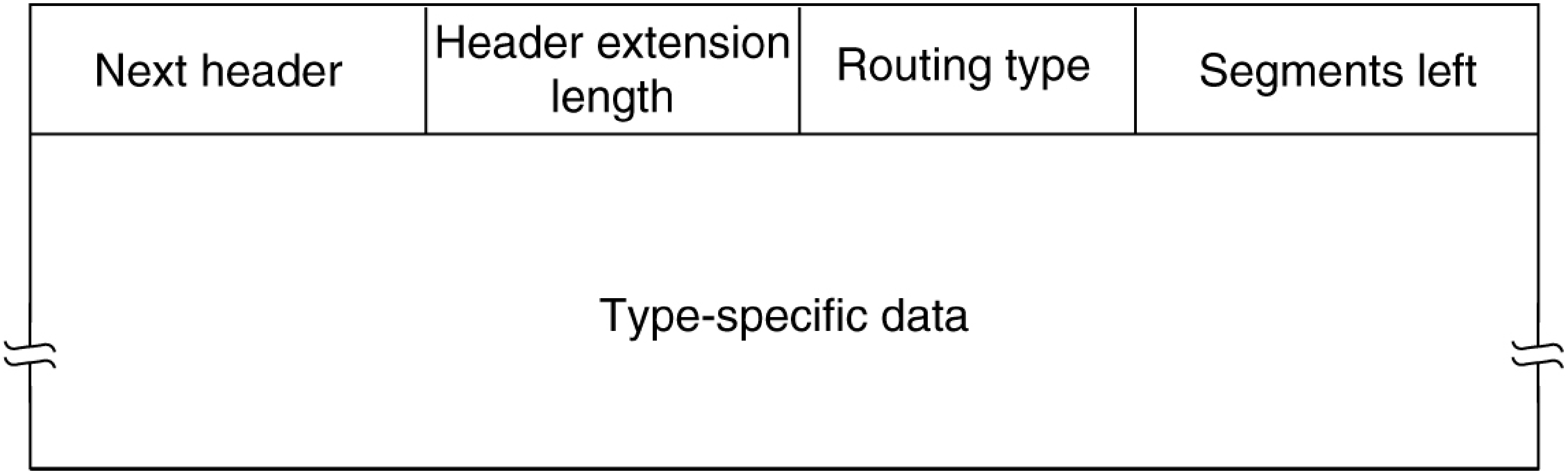
Extension Headers (2)



Next header	0	194	4
Jumbo payload length			

The hop-by-hop extension header for large datagrams (jumbo grams).

Extension Headers (3)



Next header

Header extension length

Routing type

Segments left

Type-specific data

The extension header for routing.

IPV4/IPV6

IPv4	IPv6
IPv4 addresses are 32 bit length.	IPv6 addresses are 128 bit length.
IPv4 addresses are binary numbers represented in decimals.	IPv6 addresses are binary numbers represented in hexadecimals.
IPSec support is only optional.	Inbuilt IPSec support.
Fragmentation is done by sender and forwarding routers.	Fragmentation is done only by sender.
No packet flow identification.	Packet flow identification is available within the IPv6 header using the Flow Label field.
Checksum field is available in IPv4 header.	No checksum field in IPv6 header .
Options fields are available in IPv4 header .	No option fields, but IPv6 Extension headers are available

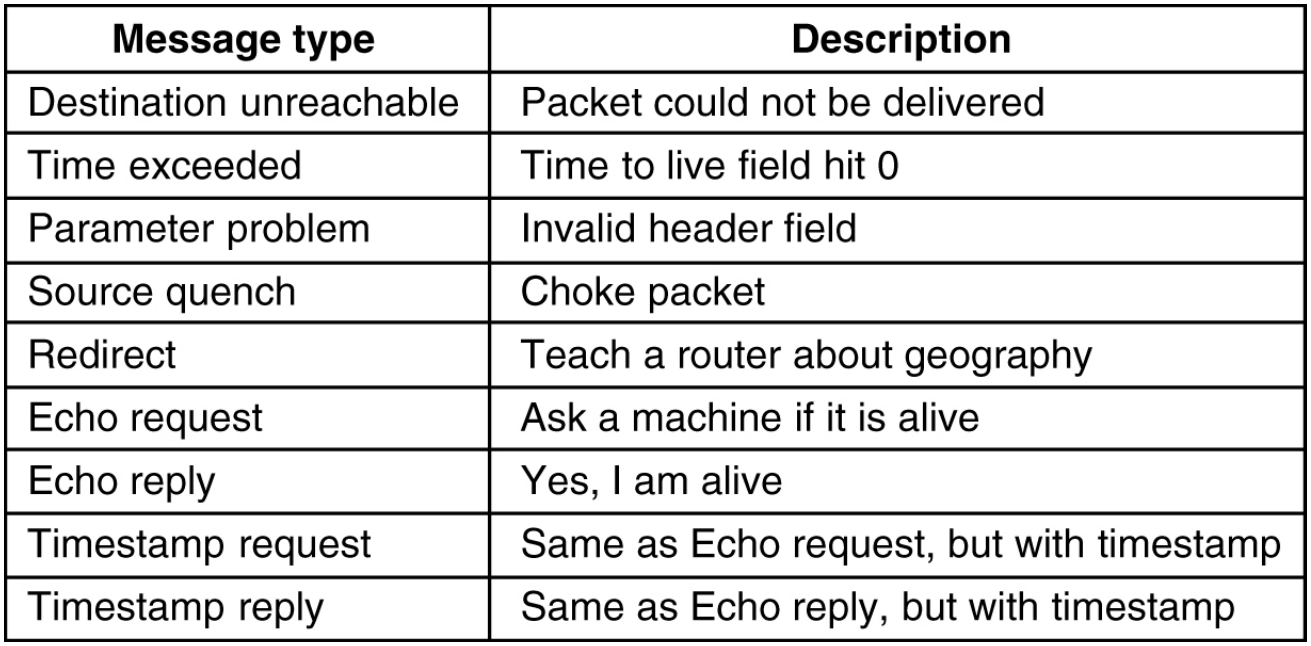
Internet Protocols

1. ICMP
2. ARP
3. RARP
4. BOOTP
5. DHCP

Internet Control Message Protocol

- The operation of the internet is monitored by the routers.
- When something unexpected occurs, the event is reported by ICMP, which is also used to test the internet.

Internet Control Message Protocol



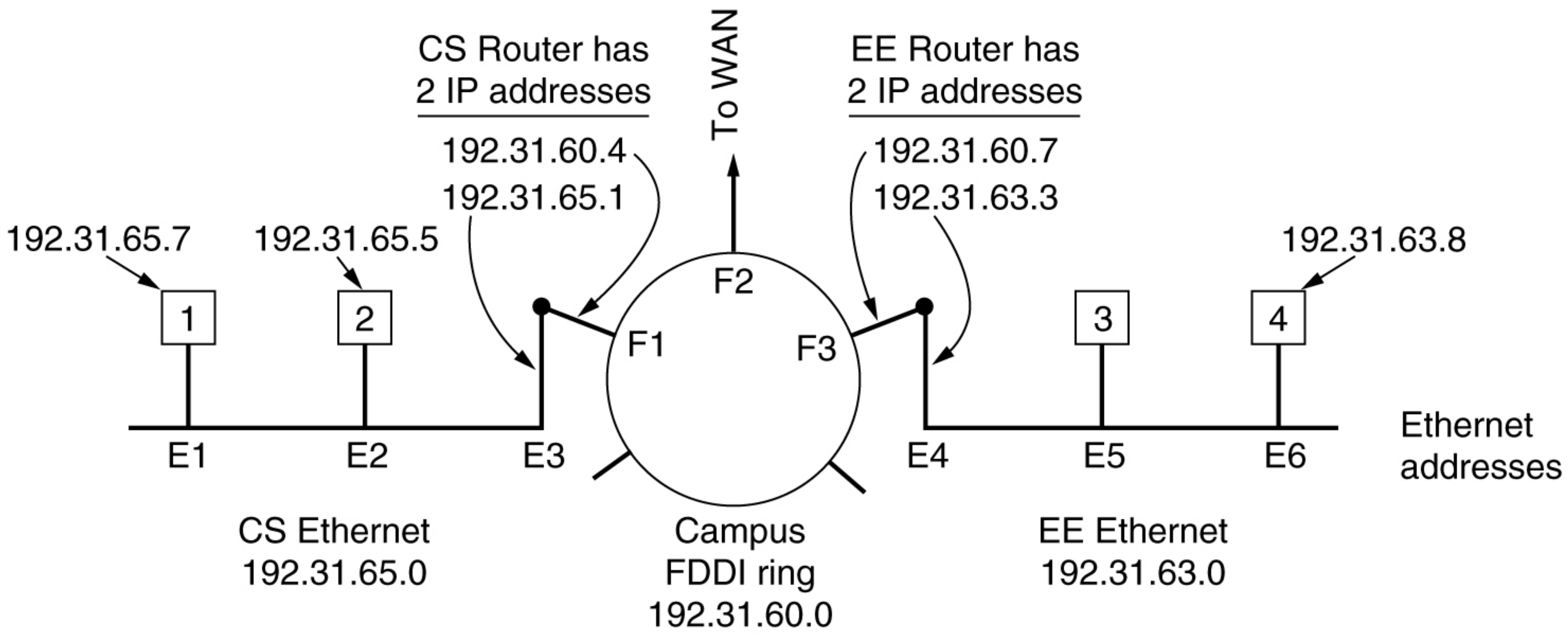
Message type	Description
Destination unreachable	Packet could not be delivered
Time exceeded	Time to live field hit 0
Parameter problem	Invalid header field
Source quench	Choke packet
Redirect	Teach a router about geography
Echo request	Ask a machine if it is alive
Echo reply	Yes, I am alive
Timestamp request	Same as Echo request, but with timestamp
Timestamp reply	Same as Echo reply, but with timestamp

The principal ICMP message types.

ARP– The Address Resolution Protocol

- Although every machine on the Internet has IP address,This cannot actually be used for sending packets because DL layer hardware does not understand Internet address.
- How do IP address get mapped onto data-link layer address, such as Ethernet?
- A host 1 sends a packet to a user on host 2.
- Let us assume sender knows name of the receiver.
- The first step is to find out IP address of H2, which is done by DNS.
- Now H1 builts a packet with destination address H2 IP address.
- But now it need a technique to find out Destination's Ethernet address.
- One solution is to have a configuration file that could do the mapping, but it is error-prone & time-consuming job.
- The better solution is to use a protocol called ARP(Address Resolution Protocol), which finds out Ethernet address corresponding to a given IP address.

ARP– The Address Resolution Protocol



Three interconnected /24 networks: two Ethernets and an FDDI ring.

ARP contd..

- The advantage is its simplicity.
- Optimization made to ARP:
- once a machine runs ARP, it caches the results in case it needs to contact the same machine shortly.
- Next time it will find mapping in its own cache, thus eliminates second broadcast
- Another is to have every machine broadcast its mapping when it boots
- In the above, Suppose H1 wants to send packet to H4.
- Using ARP will fail, bcoz H4 will not see this broadcast msg.
- There are two solutions..
- 1. CS router could be configured to respond to ARP request for the network 192.31.63.0.
- H1 will make an ARP cache entry of (192.31.63.8, E3) and sends all traffic for H4 to local router. This solution is called **proxy ARP.**
- 2. H1 immediately see that the destination is on a remote network and just send all such traffic to a default Ethernet address that handles all remote traffic, in this case E3

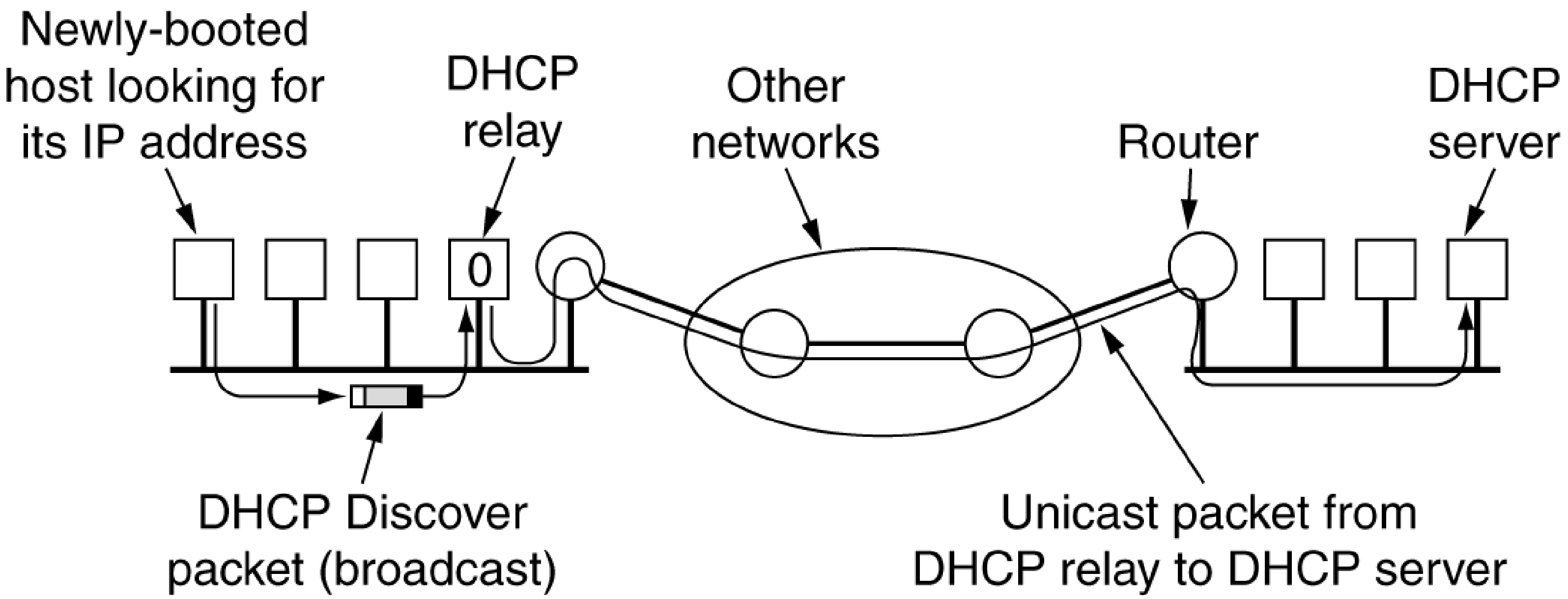
RARP and BOOTP

- Reverse Address Resolution protocol(RARP) allows a newly-booted workstation to broadcast its Ethernet address, and RARP server sees this request, looks up the Ethernet address in its configuration files, and sends back the corresponding IP address.
- A disadvantage of this is it uses a destination address of all 1's to reach the RARP server. However such broadcast are not done forwarded by routers, so RARP server is needed on each network.
- An alternative bootstrap protocol called BOOTP was used
- BOOTP uses UDP messages, which are forwarded over routers.
- It also provides a diskless workstation, including IP address of the file server holding memory image, IP address of the default router, and the subnet mask to use
- BOOTP requires manual configuration of tables mapping IP address to Ethernet address

Dynamic Host Configuration Protocol

- DHCP allows both manual IP address assignment and automatic assignment.
- Uses special server that assign IP address, it is need not be on same LAN as the requesting host.
- DHCP server cannot be reachable by broadcasting, a DHCP relay agent is needed on each LAN.
- To find IP address, a new machine broadcasts a DHCP DISCOVER packet.
- The relay agent intercepts the broadcast messages. When it finds DHCP DISCOVER packet, it sends the packet as a uni-cast packet to DHCP server(Relay agent needs to know IP address of DHCP server).
- An issue is how long an IP address should be allocated.
- Uses a technique called Leasing.

Dynamic Host Configuration Protocol


Newly-booted host looking for its IP address
DHCP relay
Other networks
Router
DHCP server
0
DHCP Discover packet (broadcast)
Unicast packet from DHCP relay to DHCP server

Operation of DHCP.

Congestion Control

- When one part of the subnet (e.g. one or more routers in an area) becomes overloaded, congestion results.
- Because routers are receiving packets faster than they can forward them, one of two things must happen:
  - The subnet must prevent additional packets from entering the congested region until those already present can be processed.
  - The congested routers can discard queued packets to make room for those that are arriving.

Factors that Cause Congestion

- Packet arrival rate exceeds the outgoing link capacity.
- Insufficient memory to store arriving packets
- Bursty traffic
- Slow processor

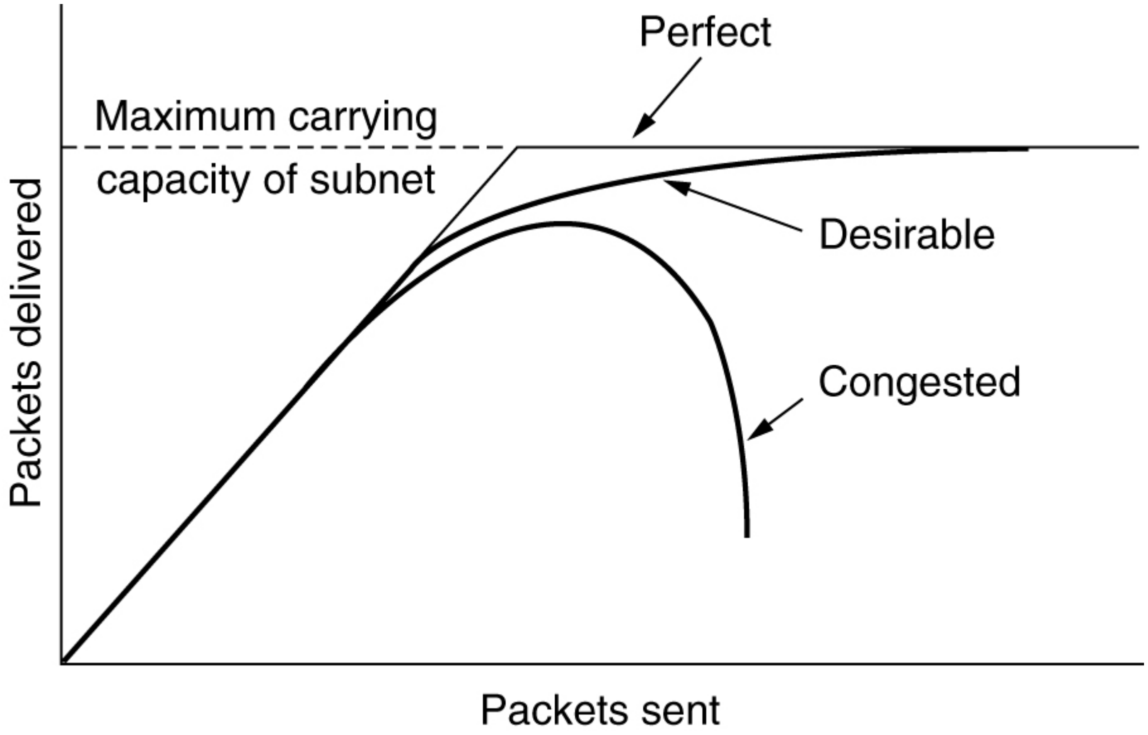
Congestion Control vs Flow Control

- Congestion control is a global issue – involves every router and host within the subnet
- Flow control – scope is point-to-point; involves just sender and receiver.

Congestion Control Algorithms

- General Principles of Congestion Control
- Congestion Prevention Policies
- Congestion Control in Virtual-Circuit Subnets
- Congestion Control in Datagram Subnets

Congestion



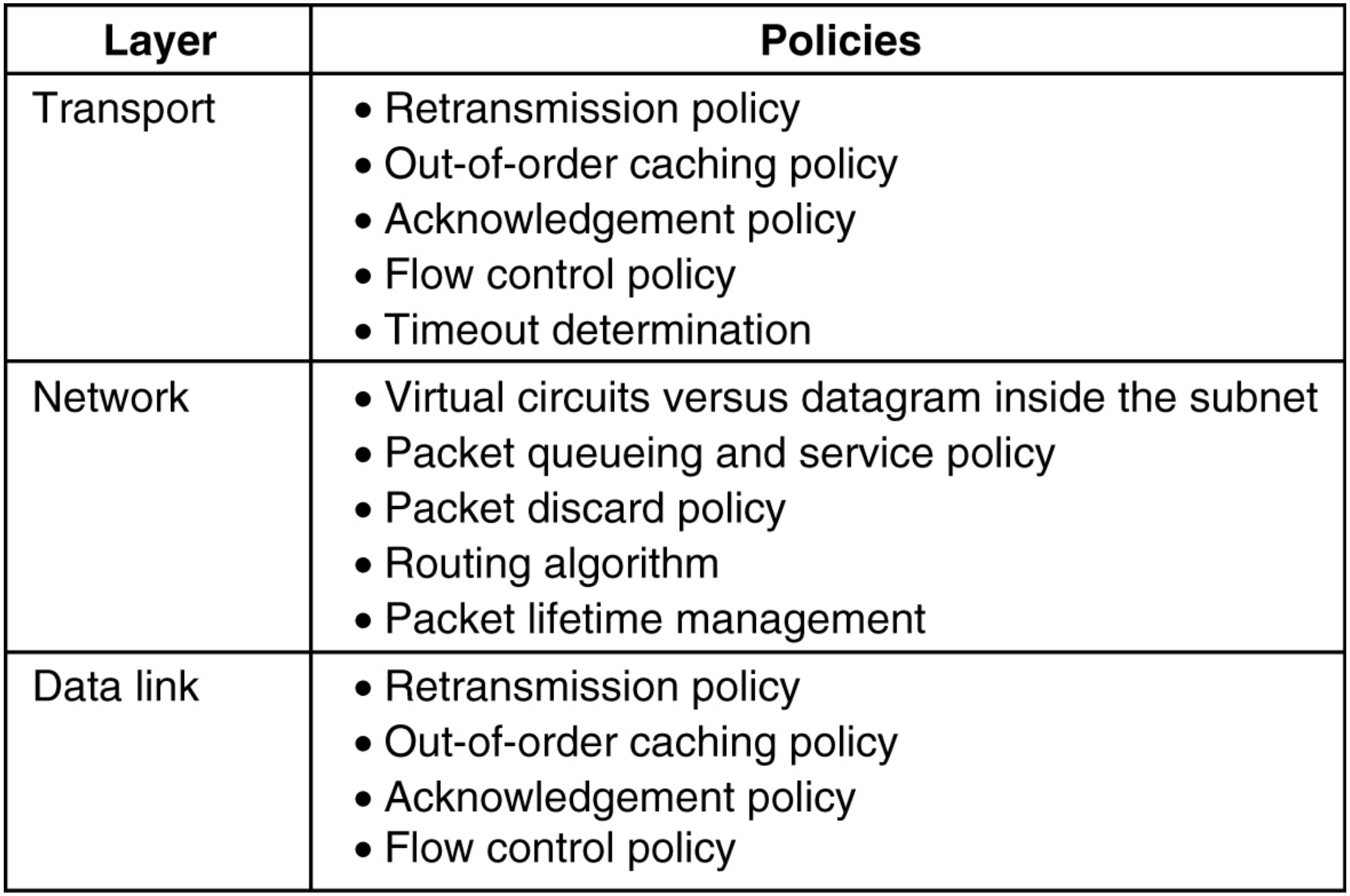
When too much traffic is offered, congestion sets in and performance degrades sharply.

General Principles of Congestion Control

- Open loop & closed loop

1. Open loop include when to accept & when to discard packets, which ones, and making scheduling decisions at various points in the network.
2. Closed loop based on feedback loop
   - Monitor the system detect when and where congestion occurs.
   - Pass information to where action can be taken.
   - Adjust system operation to correct the problem.

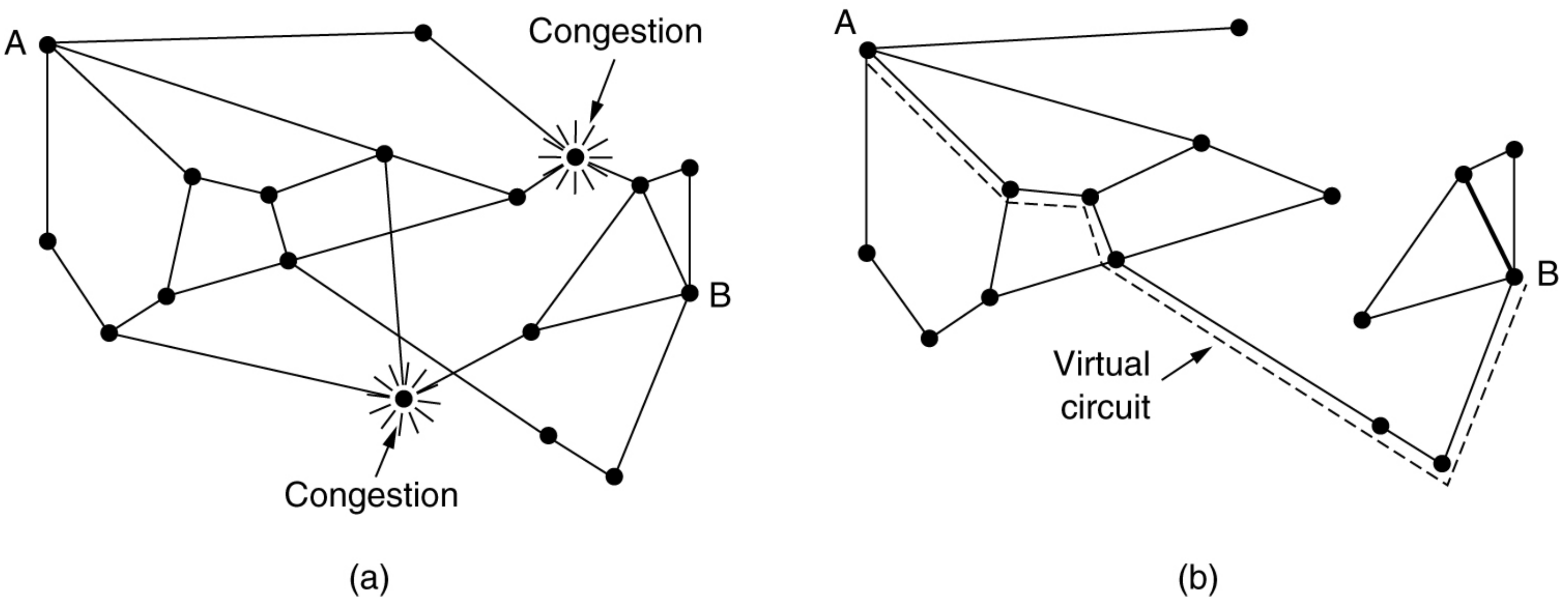
Congestion Prevention Policies



Layer	Policies
Transport	• Retransmission policy • Out-of-order caching policy • Acknowledgement policy • Flow control policy • Timeout determination
Network	• Virtual circuits versus datagram inside the subnet • Packet queueing and service policy • Packet discard policy • Routing algorithm • Packet lifetime management
Data link	• Retransmission policy • Out-of-order caching policy • Acknowledgement policy • Flow control policy

Policies that affect congestion.

Congestion Control in Virtual-Circuit Subnets



(a) A congested subnet. (b) A redrawn subnet, eliminates congestion and a virtual circuit from A to B.

Congestion Control in Datagram Subnets

- Warning Bit
- Choke packet
- Hop-by-Hop Choke packet
- Load Shedding
- Jitter Control

Warning bit

- The old DECNET architecture signaled the warning state by setting a special bit in the packet’s header.
- When the packet arrived at its destination, the transport entity copied the bit into the next acknowledgement sent back to the source. The source then cut back on the traffic.
- As long as the router was in the warning state, it continued to set the warning bit, which meant that source continued to get acknowledgements with it set.
- The source monitored the fraction of acknowledgements with the bit set and adjusted its transmission rate accordingly.
- As long as warning bits continued to flow in, the source continued to decrease its transmission rate.

Warning Bit

- A special bit in the packet header is set by the router to warn the source when congestion is detected.
- The bit is copied and piggy-backed on the ACK and sent to the sender.
- The sender monitors the number of ACK packets it receives with the warning bit set and adjusts its transmission rate accordingly.

Source based approach

- Warning bit
  - Output line in warning state
    - Warning bit set in header
    - Destination copies bit into next ack
    - Source cuts back traffic
  - Algorithm at source
    - As long as warning bits arrive: reduce traffic
    - Less warning bits: increase traffic
  - Problems
    - voluntary action of host!
    - correct source selected?
  - Used in
    - DecNet
    - Frame relay

Choke Packet

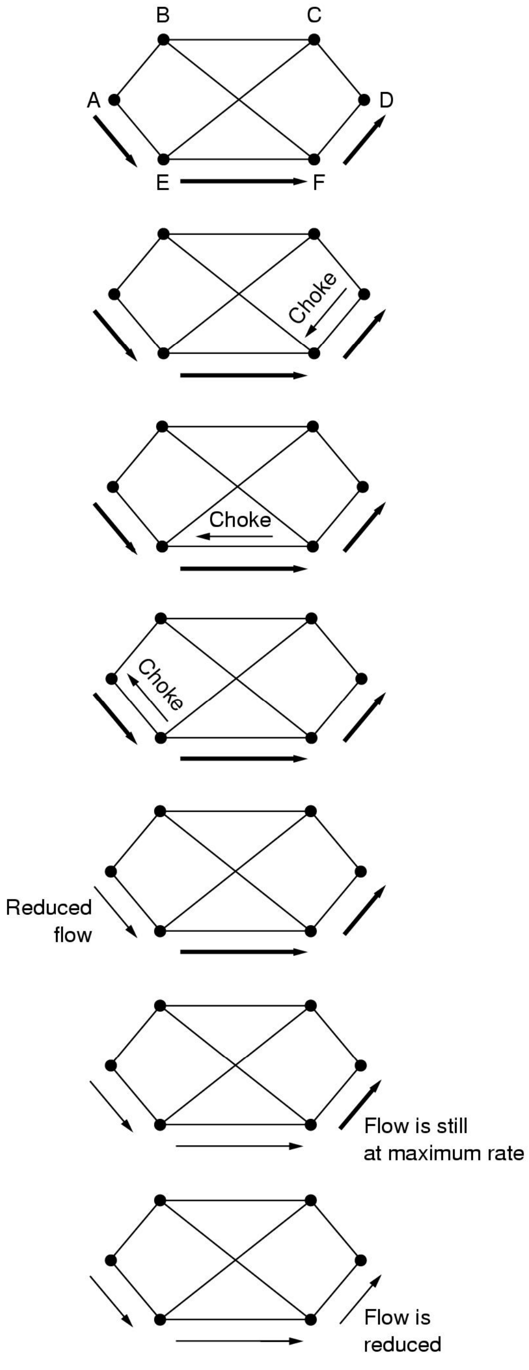
- A more direct way of telling the source to slow down.
- A choke packet is a control packet generated at a congested node and transmitted to restrict traffic flow.
- The source, on receiving the choke packet must reduce its transmission rate by a certain percentage.
- An example of a choke packet is the ICMP Source Quench Packet.

Source based approach

- Choke packet
  - In case of overload: router sends choke packet to host causing the overload
  - Host receiving choke packet
    - reduces traffic to the specified destination
    - ignores choke packets for a fixed interval
    - new choke packets during next listening interval?
      - Yes: reduce traffic
      - No: increase traffic
  - Problems:
    - voluntary action of host!
    - correct host selected?

Source based approa

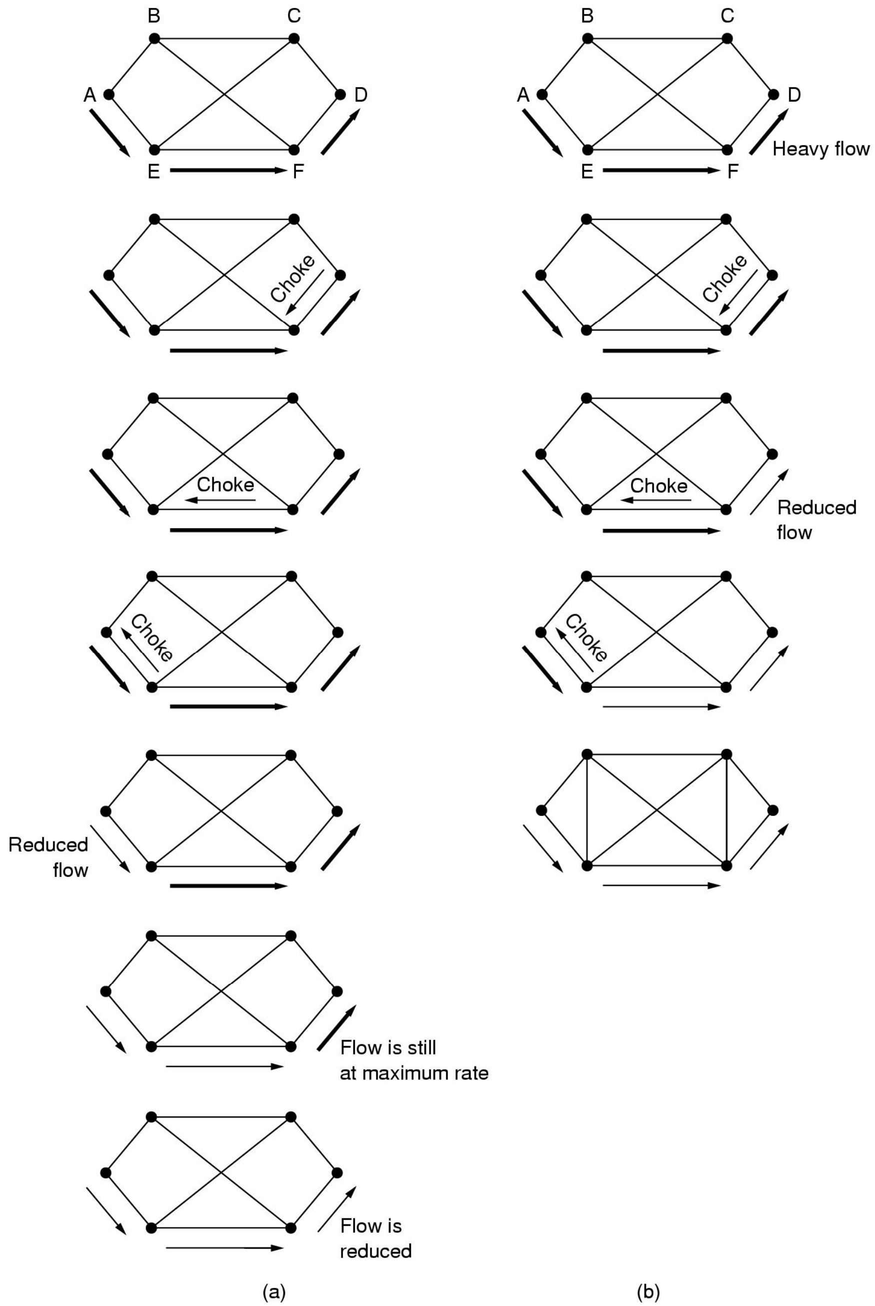
- Choke packets:
  - Example showing slow reaction
  - Solution: Hop-by-Hop choke packets



(a)

Source based approach

- Hop-by-Hop choke packets
  - Have choke packet take effect at every hop
  - Problem: more buffers needed in routers



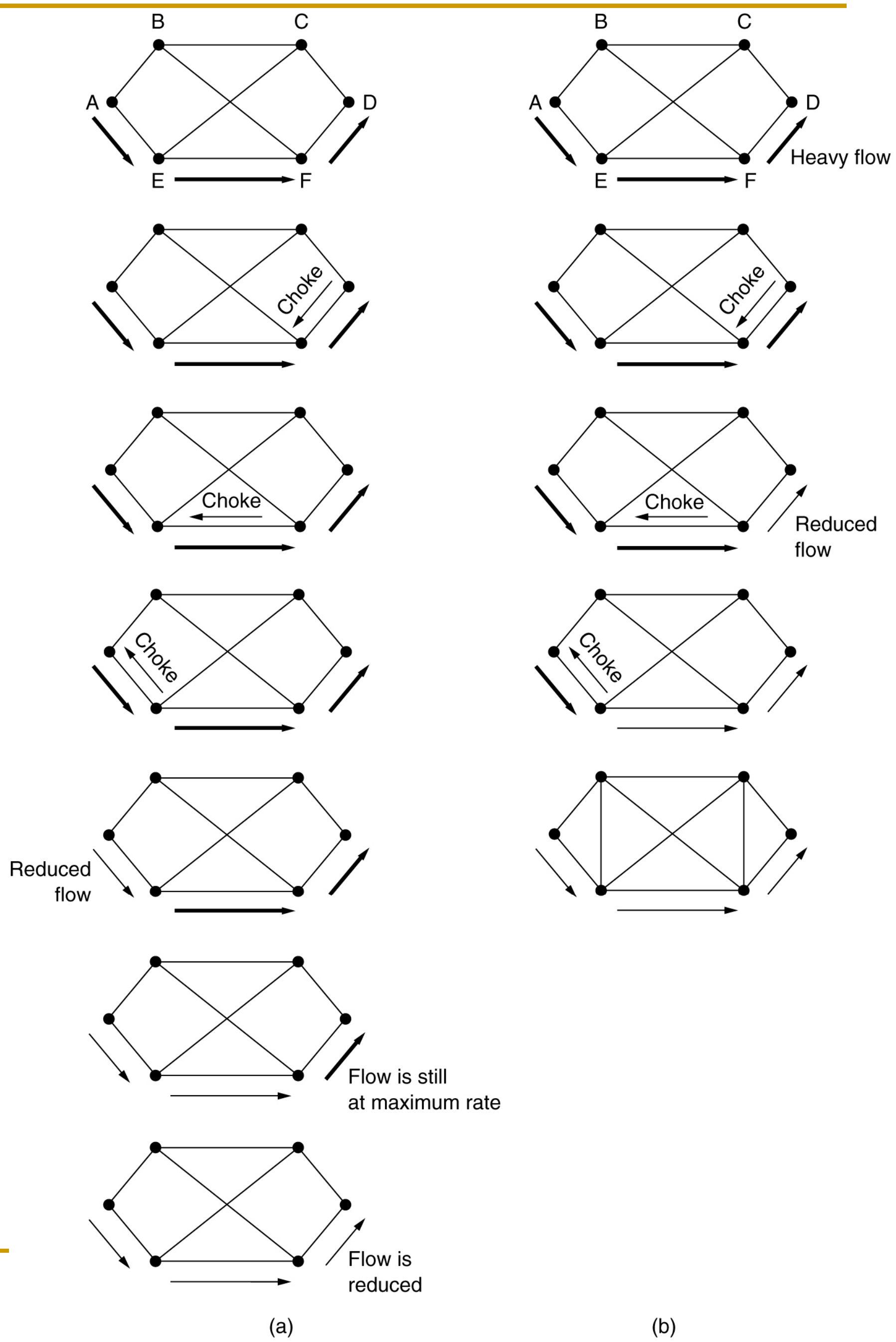
Hop-by-Hop Choke Packets

- Over long distances or at high speeds choke packets are not very effective.
- A more efficient method is to send to choke packets hop-by-hop.
- This requires each hop to reduce its transmission even before the choke packet arrive at the source.

Hop-by-Hop Choke Packets

(a) A choke packet that affects only the source.

(b) A choke packet that affects each hop it passes through.



Load Shedding

- When buffers become full, routers simply discard packets.
- Which packet is chosen to be the victim depends on the application and on the error strategy used in the data link layer.
- For a file transfer, for, e.g. cannot discard older packets since this will cause a gap in the received data.
    - For real-time voice or video it is probably better to throw away old data and keep new packets.
- Get the application to mark packets with discard priority.

Load shedding

- Throw away packets that cannot be handled!!
- Packet selection?
  - Random
  - Based on application
    - File transfer: discard new packet
    - Multimedia: discard old packet
  - Let sender indicate importance of packets
    - Low, high priority
    - Incentive to mark a packet with low priority
      - Price
      - Allow hosts to exceed agreed upon limits
- *Random early detection …*

Load shedding

- Throw away packets that cannot be handled!!
- *Packet selection?*
- Random early detection
  - Discard packets before all buffer space is exhausted
  - Routers maintain running average of queue lengths
  - Select at random a packet
  - Inform source?
    - Send choke packet? ➔ more load!!
    - No reporting
  - When does it work?
    - Source slows down when packets are lost

Random Early Discard (RED)

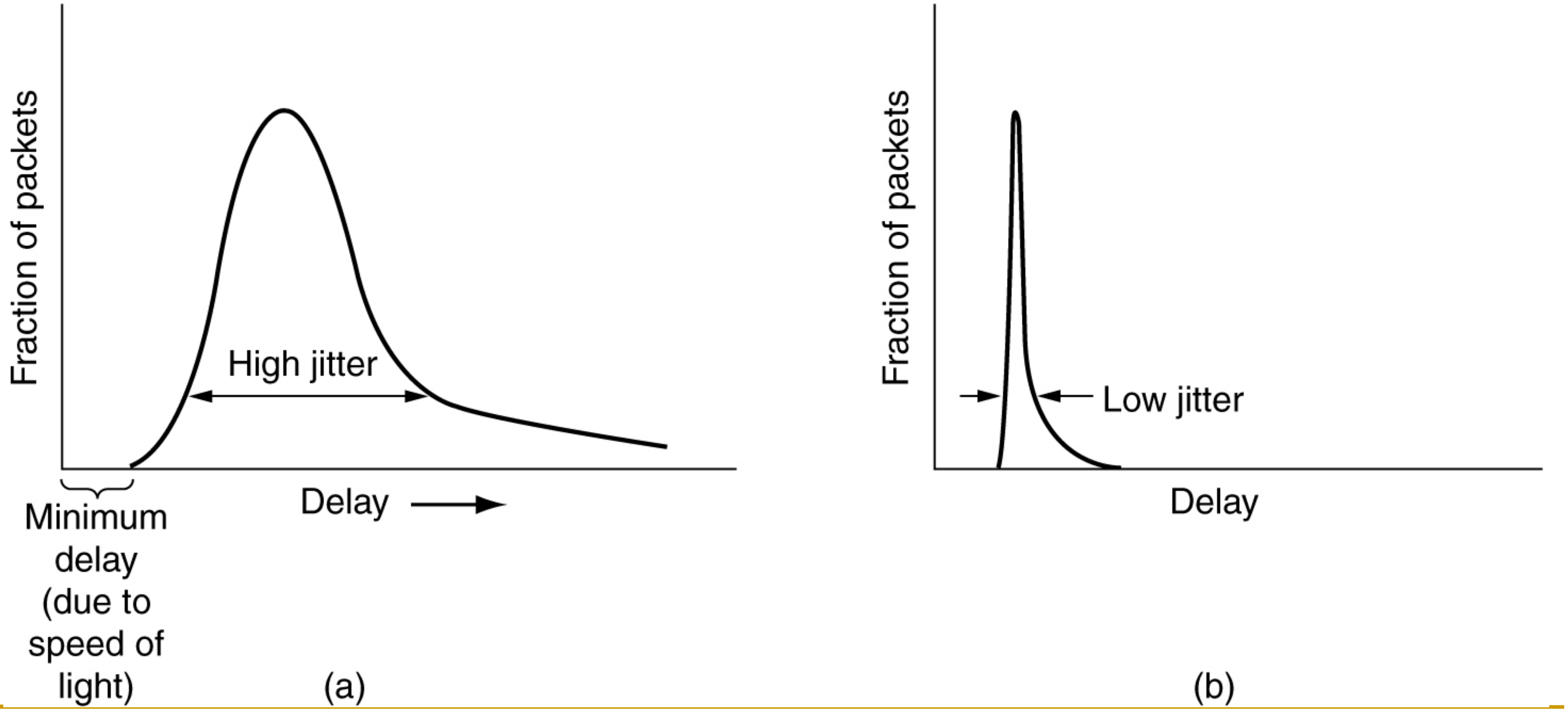
- This is a proactive approach in which the router discards one or more packets *before* the buffer becomes completely full.
- Each time a packet arrives, the RED algorithm computes the average queue length, ***avg***.
- If *avg* is lower than some lower threshold, congestion is assumed to be minimal or non-existent and the packet is queued.

RED, cont.

- If *avg* is greater than some upper threshold, congestion is assumed to be serious and the packet is discarded.
- If *avg* is between the two thresholds, this might indicate the onset of congestion. The probability of congestion is then calculated.

Congestion: jitter control

- Important for audio and video applications?
  - not delay



Congestion: jitter control

- Jitter = variation in packet delay
- Compute feasible mean value for delay
  - compute expected transit time for each hop
  - router checks to see if packet is
    - behind
    - ahead   schedule
  - behind: forward packet asap
  - ahead: hold back packet to get it on schedule again
- Buffering? Depends on characteristics:
  - Video on demand: ok
  - Videoconferencing: nok

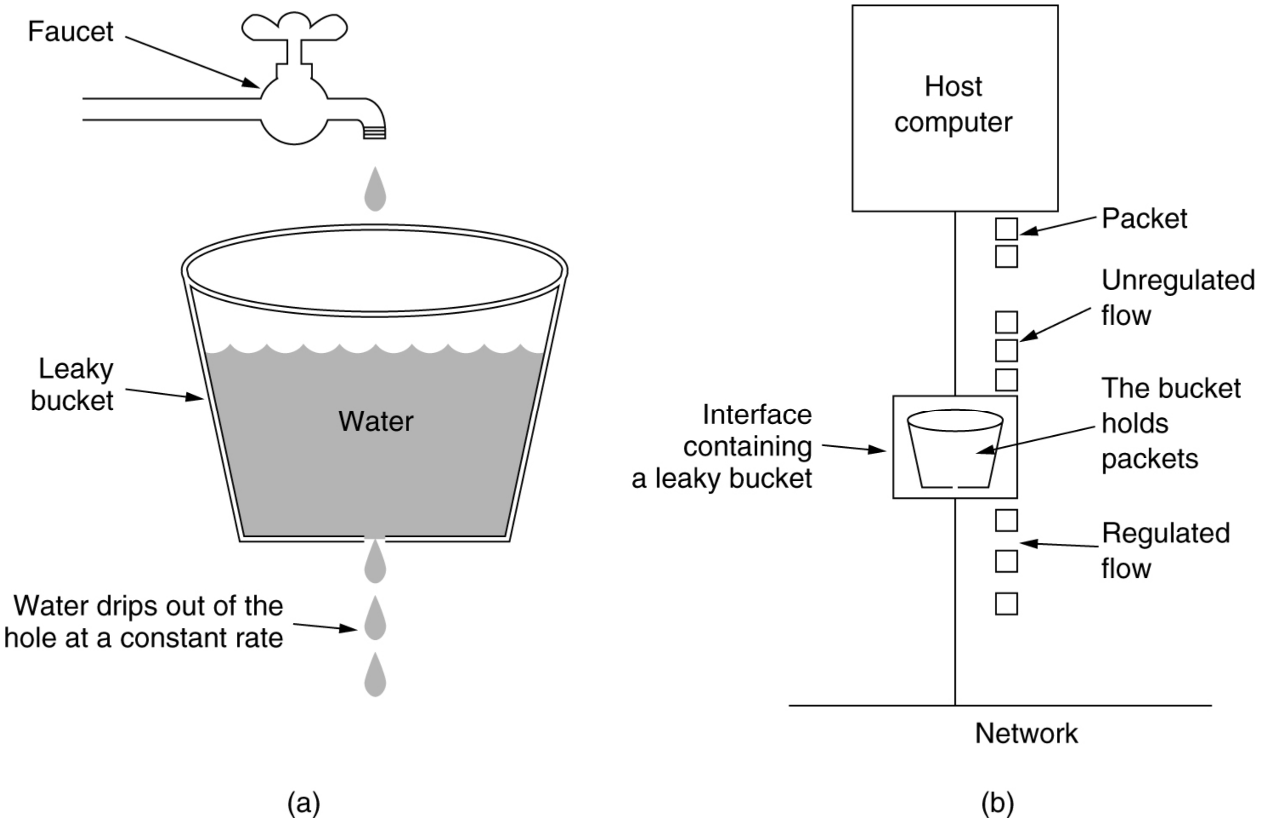
Traffic Shaping

- Another method of congestion control is to “shape” the traffic before it enters the network.
- Traffic shaping controls the *rate* at which packets are sent (not just how many). Used in ATM and Integrated Services networks.
- At connection set-up time, the sender and carrier negotiate a traffic pattern (shape).
- Two traffic shaping algorithms are:
  - ➢Leaky Bucket
  - ➢Token Bucket

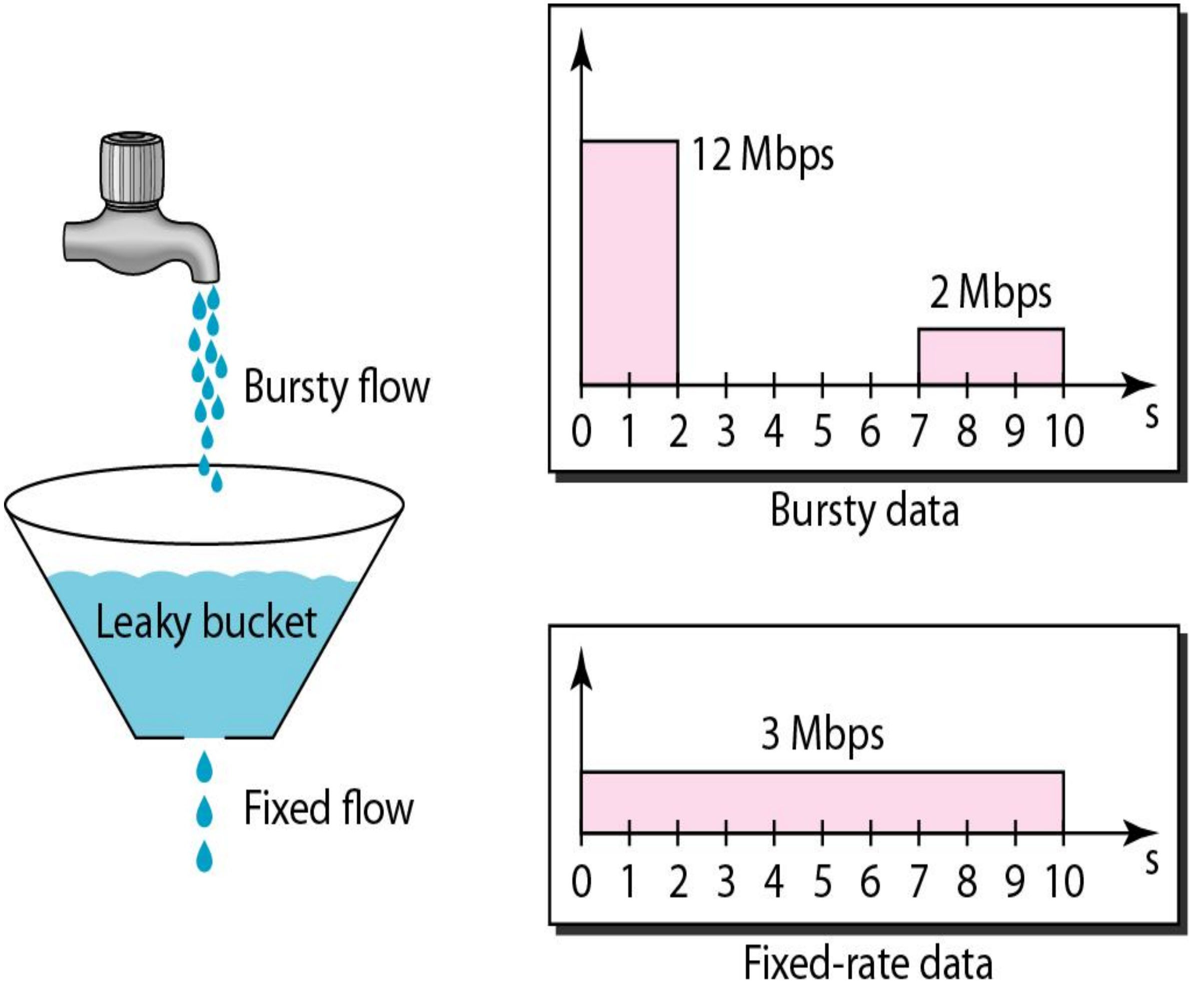
The Leaky Bucket Algorithm

- The **Leaky Bucket Algorithm** used to control rate in a network. It is implemented as a single-server queue with constant service time. If the bucket (buffer) overflows then packets are discarded.

The Leaky Bucket Algorithm



(a) A leaky bucket with water. (b) a leaky bucket with packets.



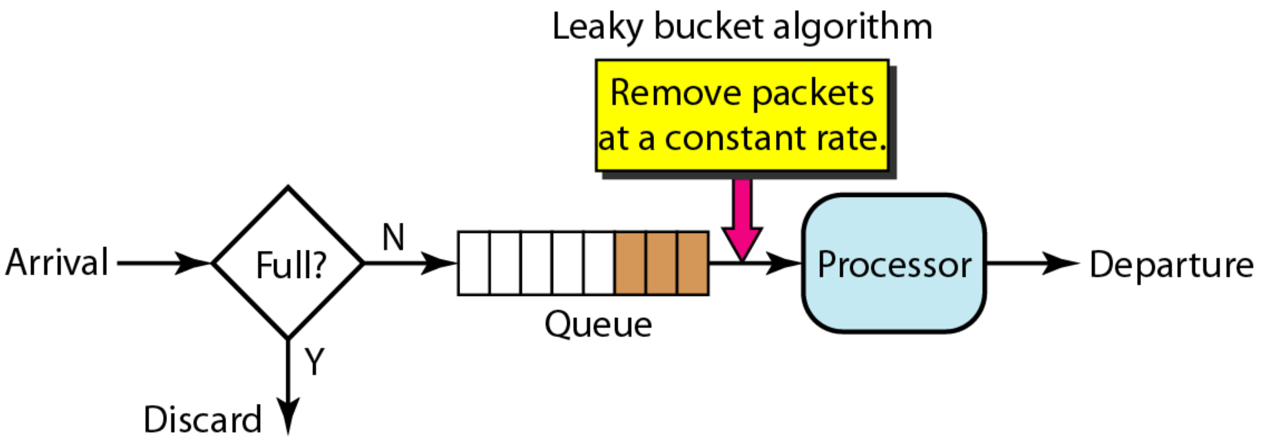
Bursty data

Fixed-rate data

Leaky Bucket Algorithm, cont.

- The leaky bucket enforces a constant output rate (average rate) regardless of the burstiness of the input. Does nothing when input is idle.
- The host injects one packet per clock tick onto the network. This results in a uniform flow of packets, smoothing out bursts and reducing congestion.
- When packets are the same size (as in ATM cells), the one packet per tick is okay. For variable length packets though, it is better to allow a fixed number of bytes per tick. E.g. 1024 bytes per tick will allow one 1024-byte packet or two 512-byte packets or four 256-byte packets on 1 tick.

Leaky Bucket Implementation



- Algorithm for variable-length packets:
  1) Initialize a counter to n at the tick of the clock
  2) If n is greater than the size of the packet, send packet and decrement the counter by the packet size. Repeat this step until n is smaller than the packet size
  3) Reset the counter and go to step 1

Example

200	**700**	**500**	**450**	**400**	**200**

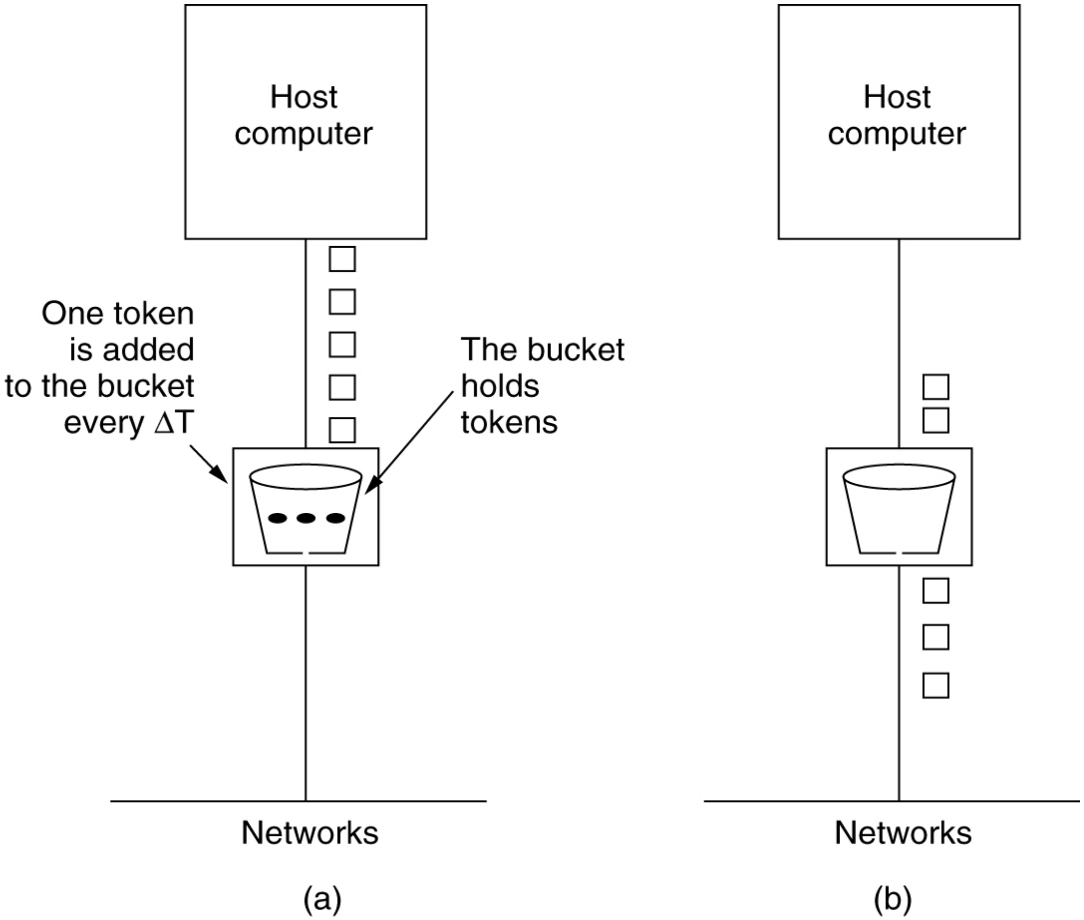
Let n=1000 and assume that the queue contains packets of variable size.

1. n>front of queue
1000>200(true)
Therefore n=1000-200=800
Packet size of 200 is sent to the network.
2. Now the Queue is **200 700 500 450 400**
800>400(front of queue) so 800-400=400
Packet size of 400 is sent
3. Now the queue becomes 200 700 500 450
400< 450 i.e the front of queue is greater than current ‘n’ value
Therefore packet size 450 can not sent. The procedure is stopped and we initialize n=1000 on another tick of clock.
This procedure is repeated until all the packets are sent to the network.

Token Bucket Algorithm

- In contrast to the LB, the Token Bucket Algorithm, allows the output rate to vary, depending on the size of the burst.
- In the TB algorithm, the bucket holds tokens. To transmit a packet, the host must capture and destroy one token.
- Tokens are generated by a clock at the rate of one token every Δt sec.
- Idle hosts can capture and save up tokens (up to the max. size of the bucket) in order to send larger bursts later.

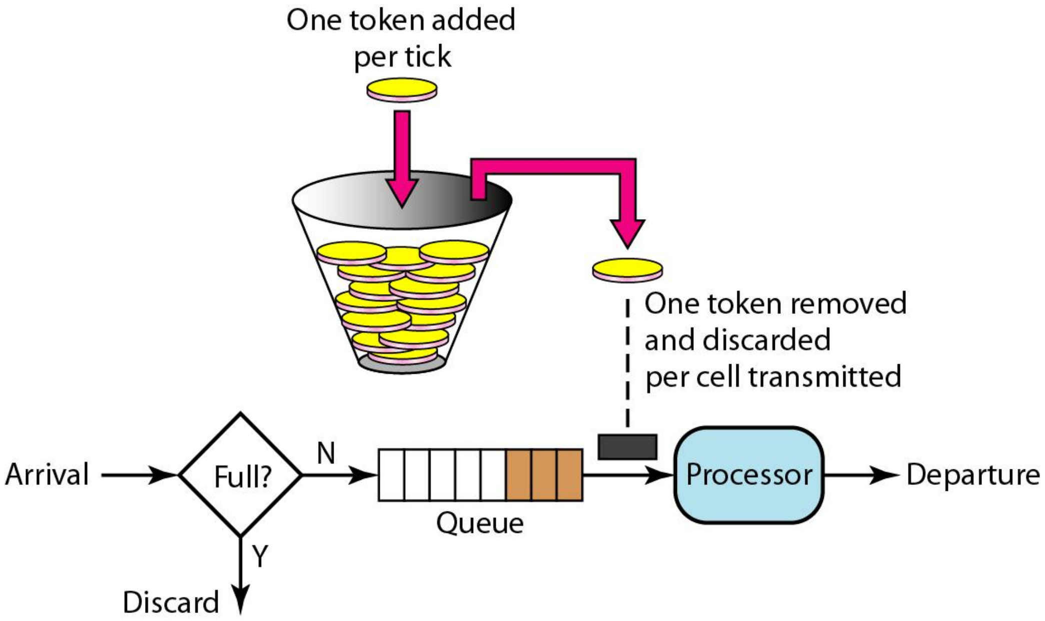
The Token Bucket Algorithm



(a) Before. (b) After.

Token Bucket

- The token bucket allows bursty traffic at a regulated maximum rate.



-

Token Bucket Algorithm:

- A token is added at every Δt time.
- The bucket can hold atmost b-tokens. If a token arrive when the bucket is full it is discarded.
- When a packet has ‘m’ bytes arrived ‘m’ tokens are removed from the bucket and the packet is sent to the network
- If less than ‘n’ tokens are available no tokens are removed from the bucket and the packet is considered to be non-conformant.

  The non-conformant packet may be enqueued for subsequent transmission when sufficient tokens have been accumulated in bucket.

Leaky Bucket vs Token Bucket

- LB discards packets; TB does not. TB discards tokens.
- With TB, a packet can only be transmitted if there are enough tokens to cover its length in bytes.
- LB sends packets at an average rate. TB allows for large bursts to be sent faster by speeding up the output.
- TB allows saving up tokens (permissions) to send large bursts. LB does not allow saving.