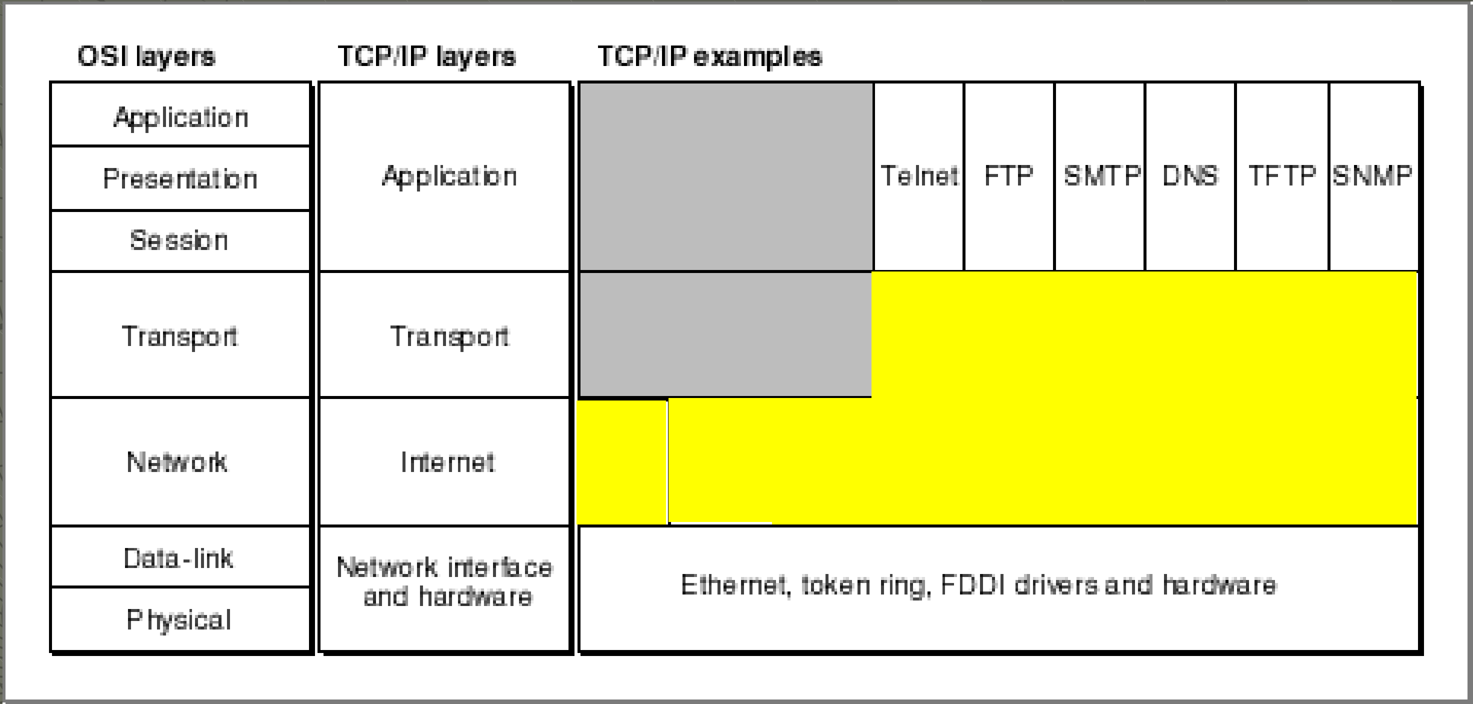
TCP/IP Protocol Suite

Internet Protocol Suite

- TCP
- IP
- ICMP
- UDP
- ARP



OSI layers	TCP/IP layers	TCP/IP examples						
Application Presentation Session	Application		Telnet	FTP	SMTP	DNS	TFTP	SNMP
Transport	Transport							
Network	Internet							
Data-link Physical	Network interface and hardware	Ethernet, token ring, FDDI drivers and hardware						

TCP/IP services:

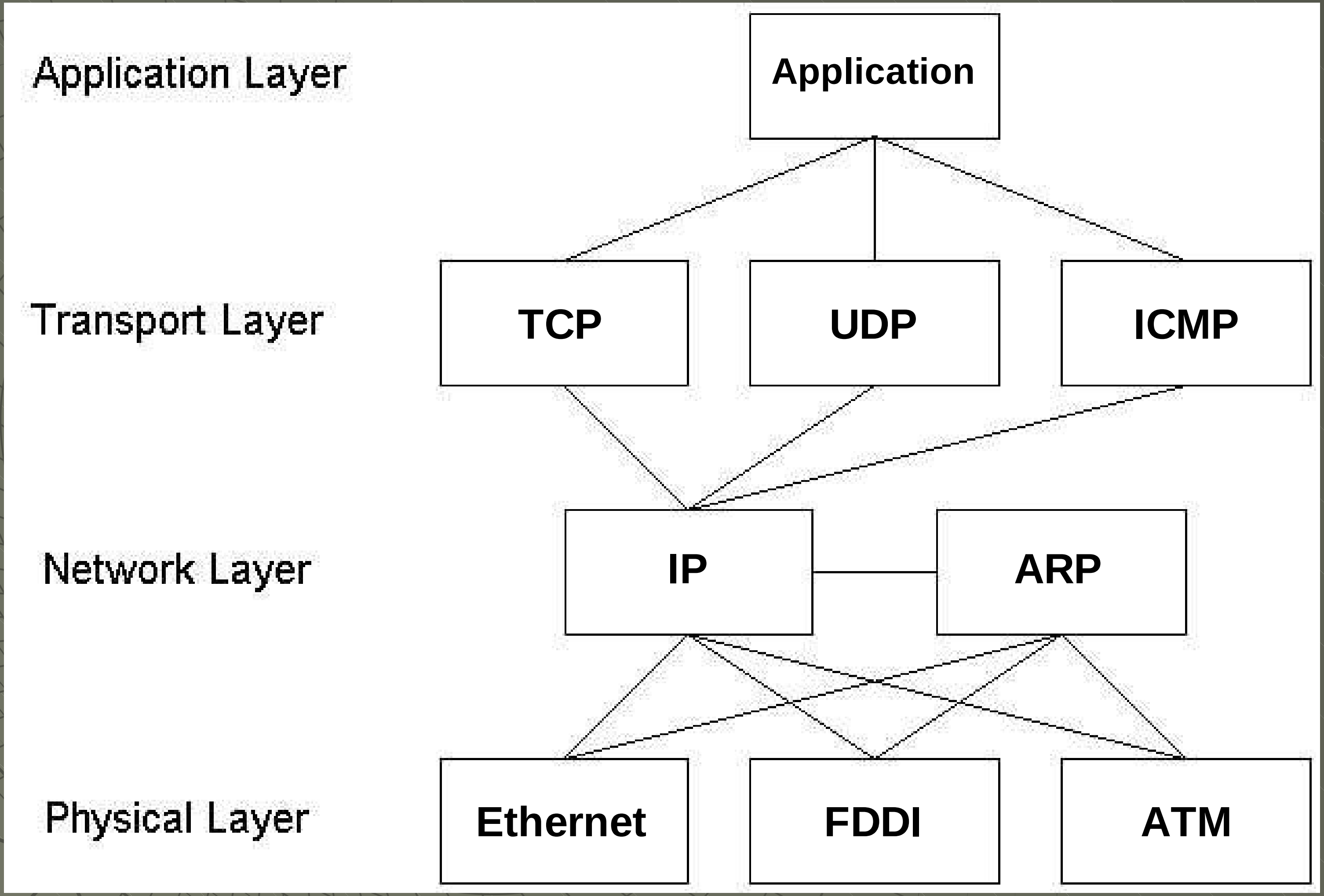
(Traditional) FTP, Remote log-in, Email

(Complex) NFS, Web browsers, Video and audio on the Internet, Networked games, Entertainment

TCP/IP Protocol Suite

- TCP/IP:
  - Built on "**connectionless**" technology. Information is transferred as a sequence of "**datagrams**"
  - Based on the "**catenet model**".
- TCP is a connection-oriented, "**reliable**" protocol.
  - Retransmission
  - Packetizing
- IP is responsible for **routing** and **delivering** the individual datagrams.

TCP/IP Protocol Suite



Internet Protocol (IP)

- Network layer protocol
- Connectionless
- Routing and delivering the individual datagrams from source to destination hosts
- Fragmentation & Sequencing datagrams
- TCP module calls on the IP module to take a TCP packet as the data portion of the IP datagram.
- IP module call on the local network interface to send the IP datagram.

IP Header

<table>
<tr><td>Version
(4 bit)</td><td>IHL
(4 bit)</td><td>Type of Service
(8 bit)</td><td colspan="2">Total Length (16 bit)</td></tr>
<tr><td colspan="3">Identification (16 bit)</td><td>Flags
(3 bit)</td><td>Fragment Offset (13 bit)</td></tr>
<tr><td colspan="2">Time to Live (8 bit)</td><td>Protocol (8 bit)</td><td colspan="2">Header Checksum (16 bit)</td></tr>
<tr><td colspan="5">Source Address (32 bit)</td></tr>
<tr><td colspan="5">Destination Address (32 bit)</td></tr>
<tr><td colspan="5">Options</td></tr>
</table>

- Version
  - The current version is 4 (Ipv4)
- IHL
  - Internet Header Length in 32-bit words
- Type of Service
  - Abstract parameters of the quality of service desired

IP Header

<table>
<tr><td>Version
(4 bit)</td><td>IHL
(4 bit)</td><td>Type of Service
(8 bit)</td><td colspan="2">Total Length (16 bit)</td></tr>
<tr><td colspan="3">Identification (16 bit)</td><td>Flags
(3 bit)</td><td>Fragment Offset (13 bit)</td></tr>
<tr><td colspan="2">Time to Live (8 bit)</td><td>Protocol (8 bit)</td><td colspan="2">Header Checksum (16 bit)</td></tr>
<tr><td colspan="5">Source Address (32 bit)</td></tr>
<tr><td colspan="5">Destination Address (32 bit)</td></tr>
<tr><td colspan="5">Options</td></tr>
</table>

- Total Length
  - Total length of the datagram including the IP header and data, measured in octets.
- Identification
  - To determine which datagram a fragment belongs to.
- Flags
  - Bit 0: Reserved
  - Bit 1 (Don’t Fragment - DF):
    0 = May Fragment, 1 = Don’t Fragment
  - Bit 2 (More Fragments - MF):
    0 = last Fragment, 1 = More Fragments

IP Header

<table>
<tr><td>Version
(4 bit)</td><td>IHL
(4 bit)</td><td>Type of Service
(8 bit)</td><td colspan="2">Total Length (16 bit)</td></tr>
<tr><td colspan="3">Identification (16 bit)</td><td>Flags
(3 bit)</td><td>Fragment Offset (13 bit)</td></tr>
<tr><td colspan="2">Time to Live (8 bit)</td><td>Protocol (8 bit)</td><td colspan="2">Header Checksum (16 bit)</td></tr>
<tr><td colspan="5">Source Address (32 bit)</td></tr>
<tr><td colspan="5">Destination Address (32 bit)</td></tr>
<tr><td colspan="5">Options</td></tr>
</table>

- Fragment Offset
  - Indicates where in the current datagram this fragment belongs.
- Time to Live
  - The maximum time the datagram is allowed to remain in Internet.
- Protocol

  1 = ICMP 6 = TCP 17 = UDP

IP Header (cont.)

- Header Checksum
  - A checksum on the header only
- Source/Destination Address
- Options

Option	Description
Security	Specifies the level of security for the
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 it’s address and timestamp

IP Addressing

- "**Internet address**" or **IP Address**
  - Eg. 142.232.90.114
    - 142.232 = network number assigned by a central authority to BCIT
    - 142.232.90 = a network in the lab
    - last octet aloows for up to 254 hosts on each subnet (excludes 0 and 255)
  - 32 bit number in binary
- **Network ID**: Identifies the systems that are located on the same physical network.
- **Host ID**: Identifies TCP/IP device within a network.

IP Address Classes

IP Address Class	First Octet Minimum	First Octet Maximum	Leading Bit Pattern	Number of Networks	Number of Hosts
Class A	1	126	0	126	16,777,214
Class B	128	191	10	16,384	65,534
Class C	192	223	110	2,097,152	254
Class D	224	239	1110		
Class E	240	247	11110		

- Class D addresses are reserved for multicast groups.
- Class E addresses are an experimental class of IP addresses.

Private IP Address

Class	Address Range	Default Mask
A	10.xxx.xxx.xxx	255.0.0.0
B	172.16.xxx.xxx	255.255.0.0
C	192.168.xxx.xxx	255.255.255.0

- Address ranges:
  - 10.0.0.0/8
  - 172.16.0.0/12
  - 192.168.0.0/16
- Routers on the Internet will not forward packets coming from these addresses.

Subnetting

- The Classic rules wastes large numbers of addresses, especially class A and B addresses
- A newer set of rules called **Classless Inter-Domain Routing (CIDR)** uses available IP addresses more efficiently.
- Default subnet mask
  - Class A = 255.0.0.0
  - Class B = 255.255.0.0
  - Class C = 255.255.255.0

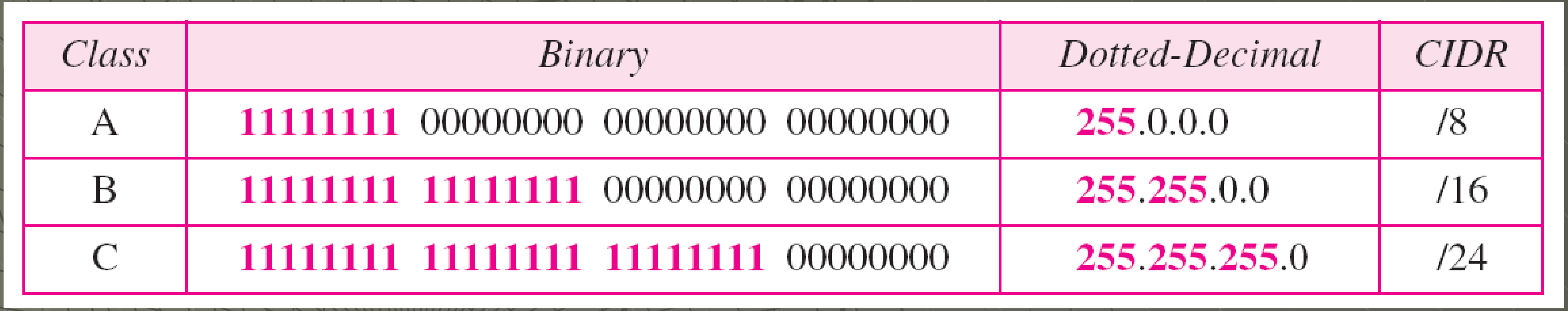
Eg. Class B default subnet mask:

11111111 1111111 0000000 0000000

1s portion = network address

0s portion = host address

Table 19.2 *Default masks for classful addressing*



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 Example

- Class B network address: **142.232.0.0**
- Default subnet mask: **255.255.0.0**
- Borrowing 2 bits from the Host ID to create a two-subnet network using a single class B address (192 = 1100 0000)

Network Address		**142.232.0.0**		
Subnet Mask		**255.255.192.0**		
All Possible Subnets				
142.232	**00**000000.00000000	142.232.	0.0	Not Available All zeros in the borrowed bits
	00111111.11111111		63.255	
	01000000.00000000		**64.0**	1st Valid Subnet Range **142.232.64.0 ~ 142.232.127.255**
	01111111.11111111		**127.255**	
	10000000.00000000		**128.0**	2nd Valid Subnet Range **142.232.128.0 ~ 142.232.191.255**
	10111111.11111111		**191.255**	
	11000000.00000000		192.0	Not Available All ones in the borrowed bits
	11111111.11111111		255.255	

Example 19.10

An ISP is granted a block of addresses starting with 190.100.0.0/16 (65,536 addresses). The ISP needs to distribute these addresses to three groups of customers as follows:

a. The first group has 64 customers; each needs 256 addresses.

b. The second group has 128 customers; each needs 128 addresses.

c. The third group has 128 customers; each needs 64 addresses.

Design the subblocks and find out how many addresses are still available after these allocations.

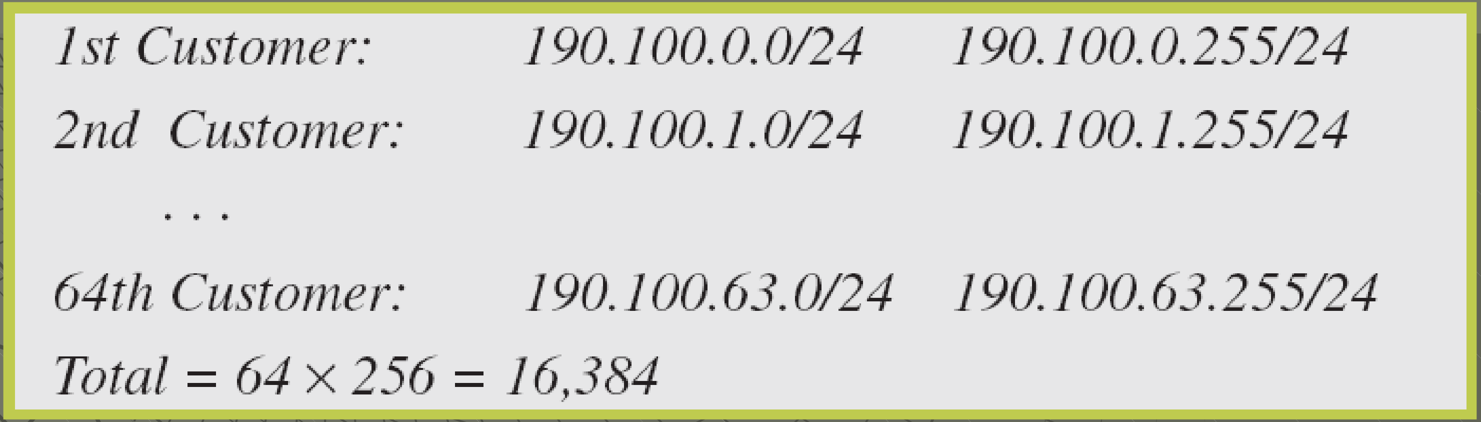
Example 19.10 (continued)

Solution

Figure 19.9 shows the situation.

Group 1

For this group, each customer needs 256 addresses. This means that 8 (log2 256) bits are needed to define each host. The prefix length is then 32 − 8 = 24. The addresses are

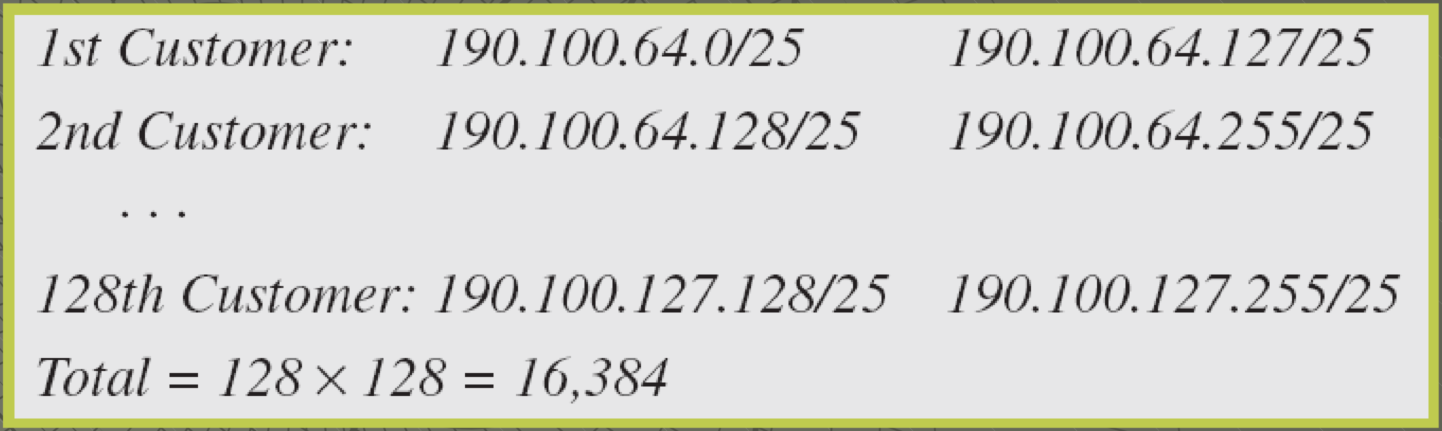


1st Customer:	*190.100.0.0/24*	*190.100.0.255/24*
2nd Customer:	*190.100.1.0/24*	*190.100.1.255/24*
. . .		
64th Customer:	*190.100.63.0/24*	*190.100.63.255/24*
Total = 64 × 256 = 16,384		

Example 19.10 (continued)

Group 2

For this group, each customer needs 128 addresses. This means that 7 (log2 128) bits are needed to define each host. The prefix length is then 32 − 7 = 25. The addresses are



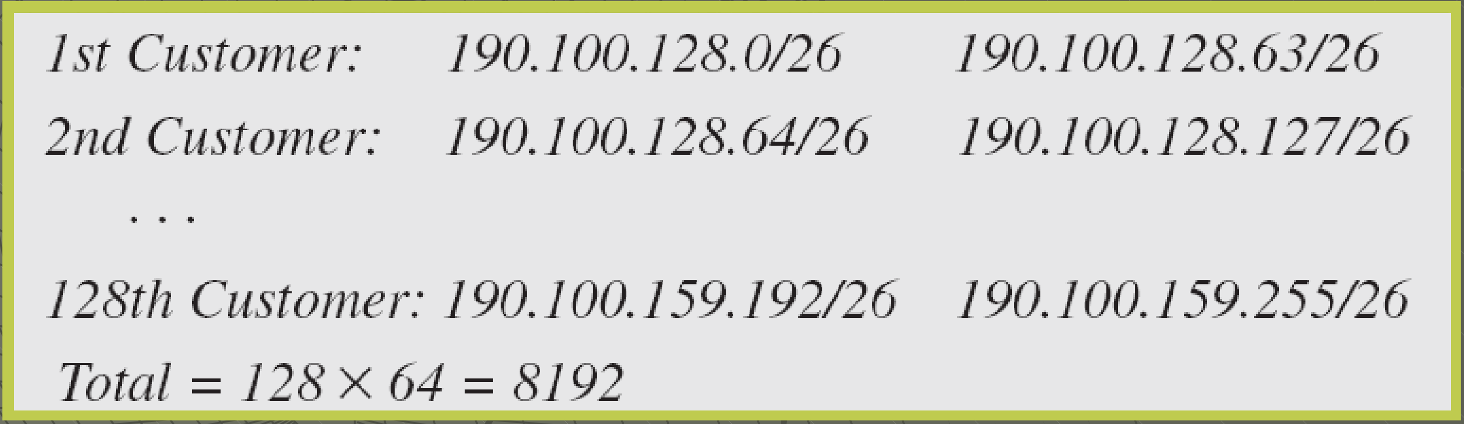
1st Customer:	190.100.64.0/25	190.100.64.127/25
2nd Customer:	190.100.64.128/25	190.100.64.255/25
. . .		
128th Customer:	190.100.127.128/25	190.100.127.255/25

Total = 128 × 128 = 16,384

Example 19.10 (continued)

Group 3

For this group, each customer needs 64 addresses. This means that 6 ($\log_2 64$) bits are needed to each host. The prefix length is then 32 − 6 = 26. The addresses are



1st Customer:	190.100.128.0/26	190.100.128.63/26
2nd Customer:	190.100.128.64/26	190.100.128.127/26
. . .		
128th Customer:	190.100.159.192/26	190.100.159.255/26

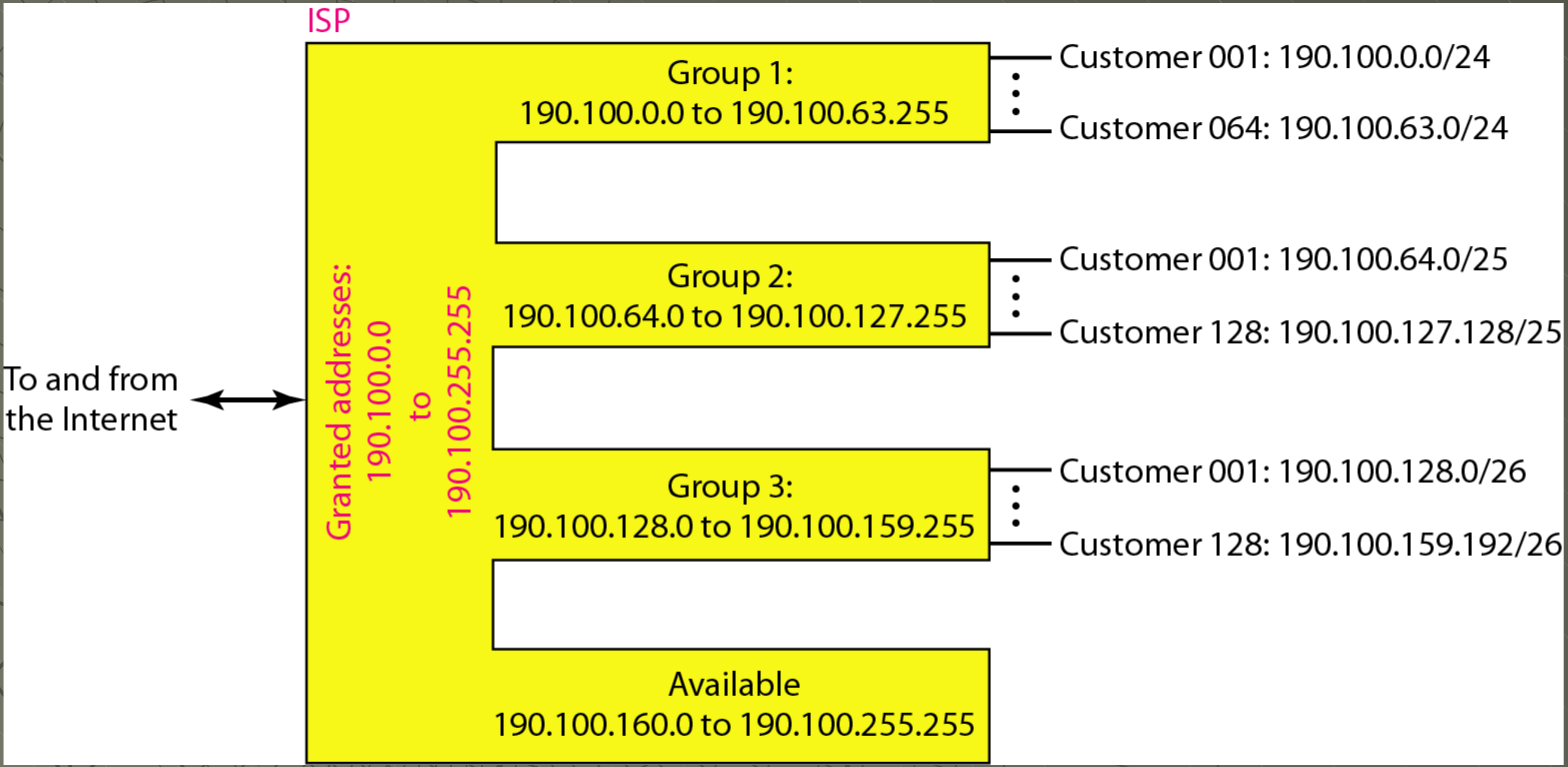
Total = 128 × 64 = 8192

Number of granted addresses to the ISP: 65,536

Number of allocated addresses by the ISP: 40,960

Number of available addresses: 24,576

Figure 19.9 *An example of address allocation and distribution by an ISP*


ISP
Group 1:
190.100.0.0 to 190.100.63.255
Customer 001: 190.100.0.0/24
Customer 064: 190.100.63.0/24
Group 2:
190.100.64.0 to 190.100.127.255
Customer 001: 190.100.64.0/25
Customer 128: 190.100.127.128/25
To and from
the Internet
Granted addresses:
190.100.0.0
to
190.100.255.255
Group 3:
190.100.128.0 to 190.100.159.255
Customer 001: 190.100.128.0/26
Customer 128: 190.100.159.192/26
Available
190.100.160.0 to 190.100.255.255

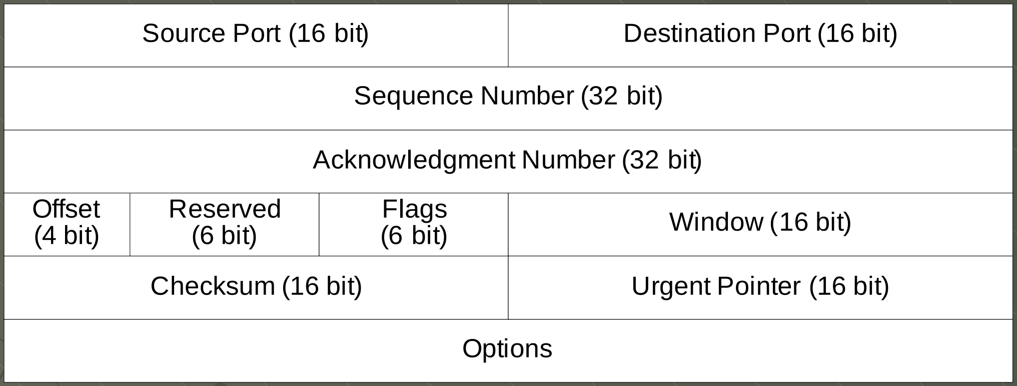
Transmission Control Protocol (TCP)

- Connection-oriented, reliable service.
- Full-duplex flow control service
- Packetizing

<table>
<tr><td colspan="3">Source Port (16 bit)</td><td>Destination Port (16 bit)</td></tr>
<tr><td colspan="4">Sequence Number (32 bit)</td></tr>
<tr><td colspan="4">Acknowledgment Number (32 bit)</td></tr>
<tr><td>Offset
(4 bit)</td><td>Reserved
(6 bit)</td><td>Flags
(6 bit)</td><td>Window (16 bit)</td></tr>
<tr><td colspan="3">Checksum (16 bit)</td><td>Urgent Pointer (16 bit)</td></tr>
<tr><td colspan="4">Options</td></tr>
</table>

TCP Header

TCP Header



Source Port (16 bit)			Destination Port (16 bit)
Sequence Number (32 bit)			
Acknowledgment Number (32 bit)			
Offset (4 bit)	Reserved (6 bit)	Flags (6 bit)	Window (16 bit)
Checksum (16 bit)			Urgent Pointer (16 bit)
Options			

TCP Header

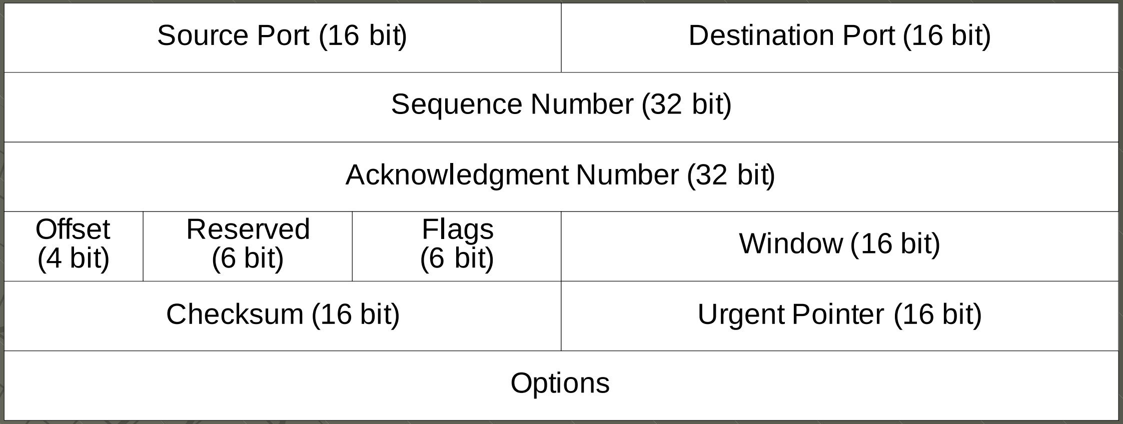
- Source and Destination Ports
  - TCP ports are entry points into services.
  - Well-known ports: 21(FTP), 25(SMTP), 80(HTTP)
- Sequence Number
- Acknowledge Number

  = Sequence number of next packet expected by the receiver
- TCP Header Length

  Number of 32-bit words

TCP Header

- Control Bits
  - URG
  - PSH
  - SYN
  - ACK
  - RST
  - FIN
- Window Size
  - Size of the sliding window for low control
- Checksum
  - Includes the header and the data
- Urgent Pointer
- Options

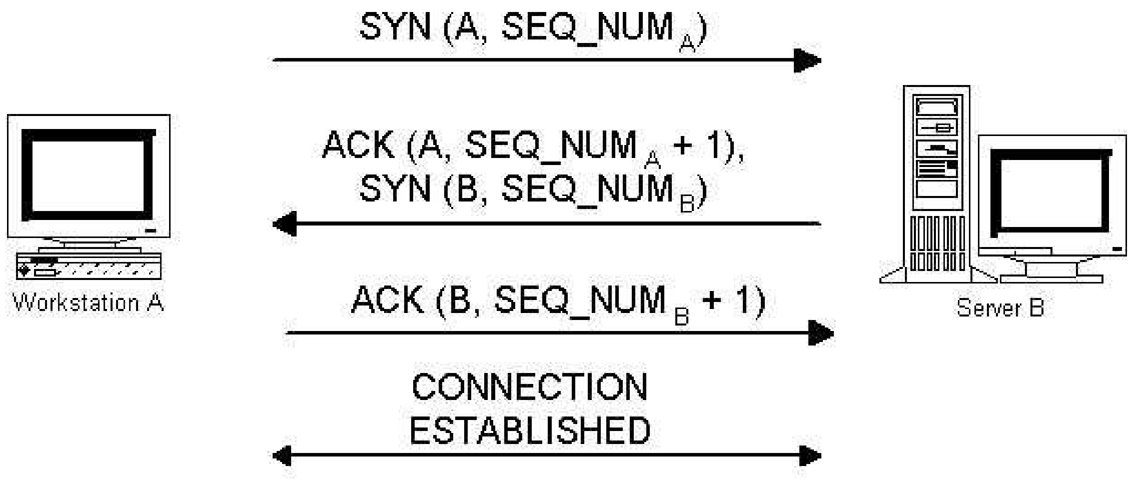


Source Port (16 bit)			Destination Port (16 bit)
Sequence Number (32 bit)			
Acknowledgment Number (32 bit)			
Offset (4 bit)	Reserved (6 bit)	Flags (6 bit)	Window (16 bit)
Checksum (16 bit)			Urgent Pointer (16 bit)
Options			

TCP Header

TCP - Three Way Handshake

TCP Connection Establishment



TCP 3-Way Handshake

User Datagram Protocol (UDP)

- Unreliable service
  - No guarantees for delivery
  - No protection from duplicate datagrams (no sequence numbers).
- Simple => small overhead
- No connection established

Source Port (16 bit)	Destination Port (16 bit)
Length (16 bit)	Checksum (16 bit)

UDP Header

UDP Pseudo Header

- Added to the beginning of the datagram to calculate the checksum
- Used to ensure that the datagram has been delivered to the correct destination.

Source IP Address		
Destination IP Address		
00000000	Protocol = 0x11	UDP Length

Address Resolution Protocol (ARP)

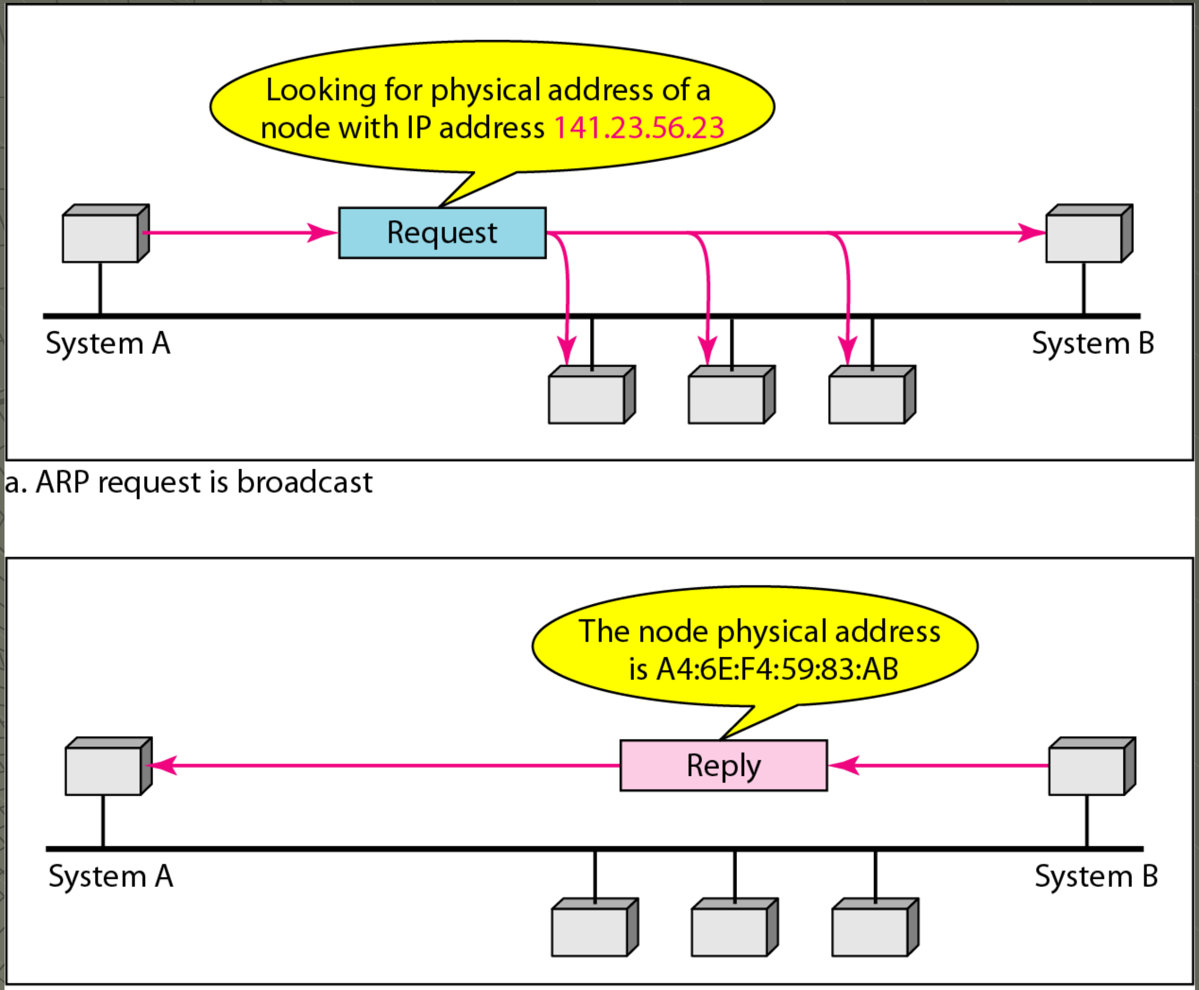
- Maps an IP address to a physical machine (MAC) address
- **ARP cache**: used to maintain a correlation between each MAC address and its corresponding IP address
- How does it work?

1. When incoming packet arrives at a gateway or router, the device asks the ARP program to find a physical host for the IP address.

2a. If the address is found in the cache, it provides.

2b. If the address is not found, ARP broadcasts the request and updates its cache.

Figure 21.1 *ARP operation*


Looking for physical address of a node with IP address 141.23.56.23
Request
System A
System B
The node physical address is A4:6E:F4:59:83:AB
Reply
System A
System B

a. ARP request is broadcast

b. ARP reply is unicast

Reverse Address Resolution Protocol (RARP)

- Request to learn its IP address from a router’s ARP table or cache.
- A network administrator creates a table in a local area network's gateway.
- How does it work?

1. A new machine that is connected to the network requests the RARP server on the router for its IP address.
2. Assuming that an entry has been set up in the router table, the RARP server will return the IP address to the machine.

Internet Control Message Protocol (ICMP)

- Error reporting and diagnostic utility.
- Required part of any IP implementation.
- ICMP messages are used by routers, intermediary devices, or hosts.

Type (8 bit)	Code (8 bit)	Checksum (16 bit)
[Unused] (32 bit)		
Internet Header + 64 bits of Original Data Datagram (32 bit)		

ICMP Header

ICMP Header

Type

- Echo Request & Echo Reply
  - Used by "**ping**"
- Source Quench
  - Sent when the destination is unable to process traffic as fast as the source is sending it.
- Redirect Message
  - Generated by an intermediary device when a route being requested can be reached either locally or through a better path

Type	Description
0	Echo Reply
3	Destination
4	Source Quench
5	Redirect Message
8	Echo Request
11	Time Exceeded
12	Parameter Problem
13	Timestamp Request
14	Timestamp Reply
15	Information Request
16	Information Reply
17	Address Mask Request
18	Address Mask Reply

Destination Unreachable

Type 3 Code	Description
0	Network Unreachable
1	Host Unreachable
2	Protocol Unreachable
3	Port Unreachable
4	Fragmentation needed and DF (Don't Fragment) set
5	Source route failed
6	Destination Network unknown
7	Destination Host unknown
8	Source Host isolated
9	Communication with Destination Network Administratively
10	Communication with Destination Host Administratively
11	Network Unreachable for Type Of Service
12	Host Unreachable for Type Of Service
13	Communication Administratively Prohibited by Filtering
14	Host Precedence Violation
15	Precedence Cutoff in Effect

ICMP Header (cont.)

Type

- Time Exceeded
  - Generated when a router or host **discards a packet due to a time-out**.
- Parameter Problem
  - Generated when an intermediary device or host **discards a datagram due to inability to process**.

  Eg. Corrupt header, missing options

Type	Description
0	Echo Reply
3	Destination
4	Source Quench
5	Redirect Message
8	Echo Request
11	Time Exceeded
12	Parameter Problem
13	Timestamp Request
14	Timestamp Reply
15	Information Request
16	Information Reply
17	Address Mask Request
18	Address Mask Reply

ICMP Header (cont.)

Type

- Timestamp Request & Timestamp Reply
  - Rudimentary method for **synchronizing the time** maintained on different devices.
- Information Request & Information Reply
  - **Obsolete** and no longer used.

Type	Description
0	Echo Reply
3	Destination
4	Source Quench
5	Redirect Message
8	Echo Request
11	Time Exceeded
12	Parameter Problem
13	Timestamp Request
14	Timestamp Reply
15	Information Request
16	Information Reply
17	Address Mask Request
18	Address Mask Reply