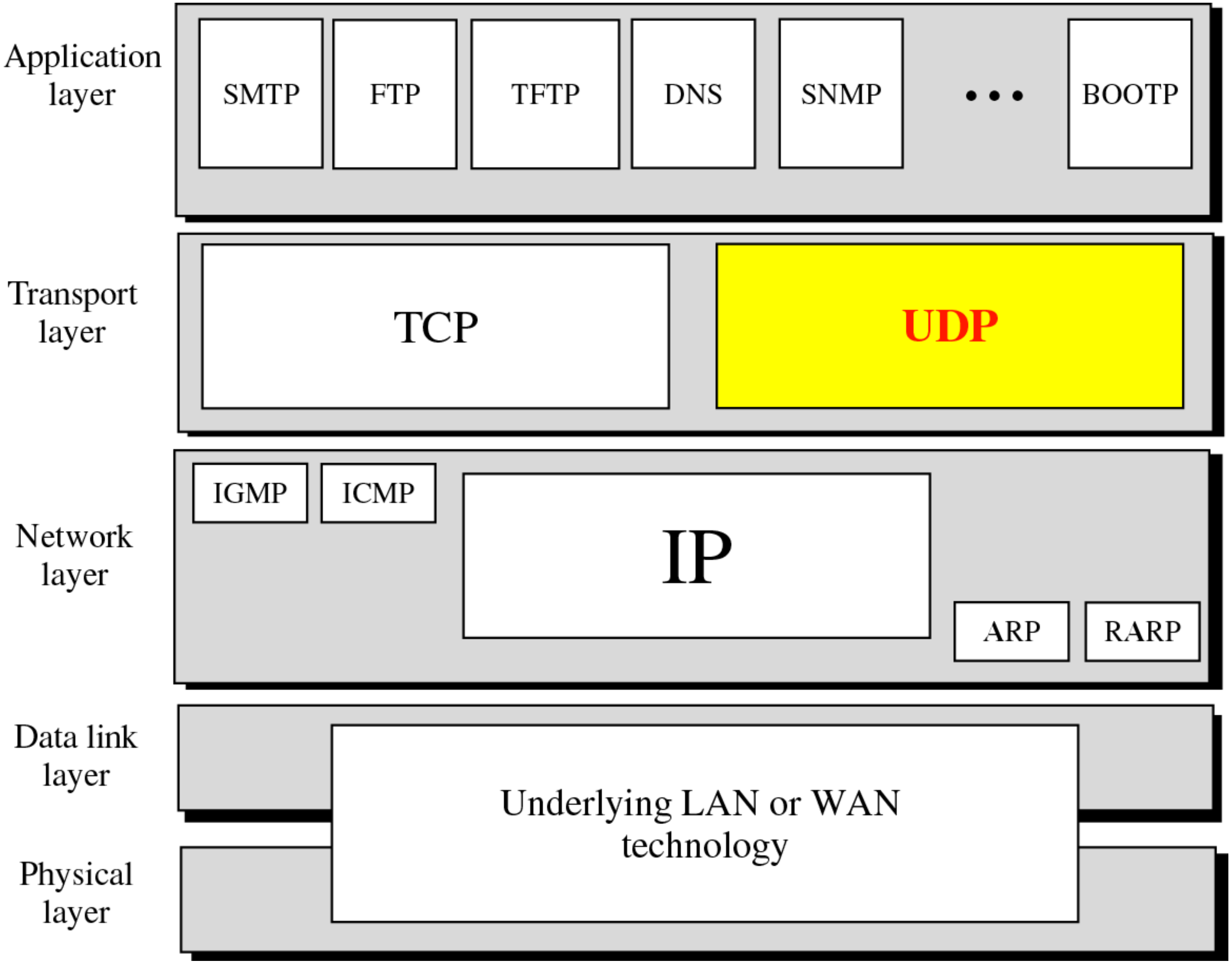
Chapter 14

User Datagram Protocol (UDP)

CONTENTS

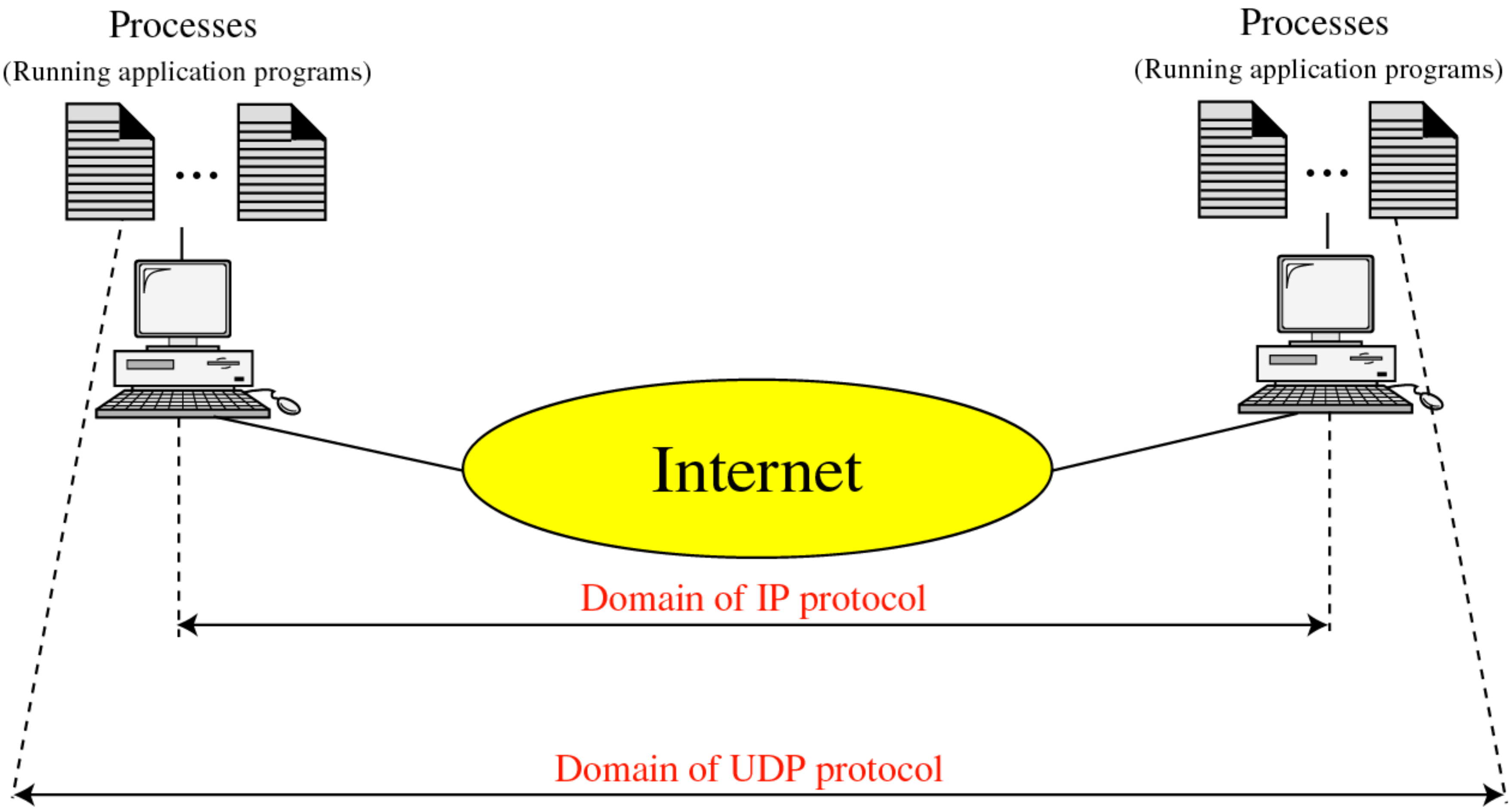
- PROCESS-TO-PROCESS COMMUNICATION
- USER DATAGRAM
- CHECKSUM
- UDP OPERATION
- USE OF UDP
- UDP PACKAGE

Position of UDP in the TCP/IP protocol suite

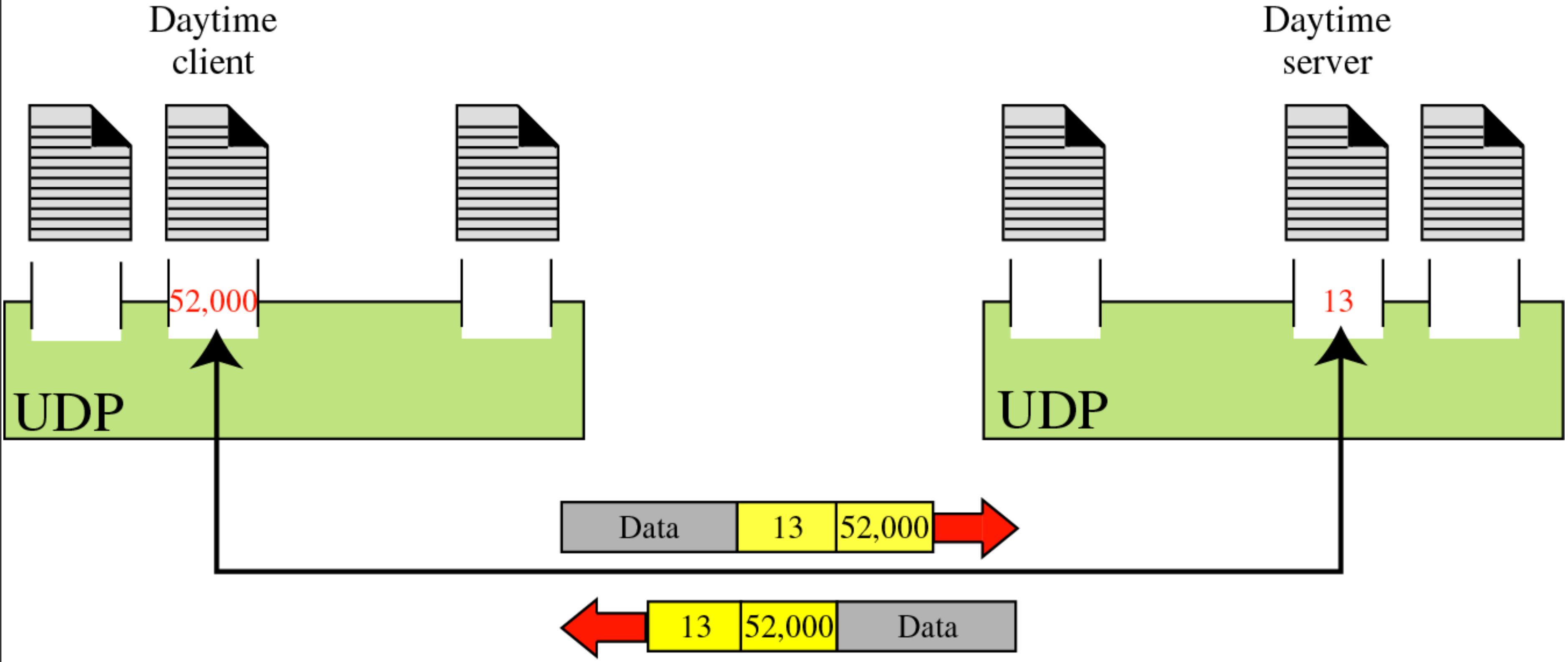


PROCESS TO PROCESS COMMUNICATION

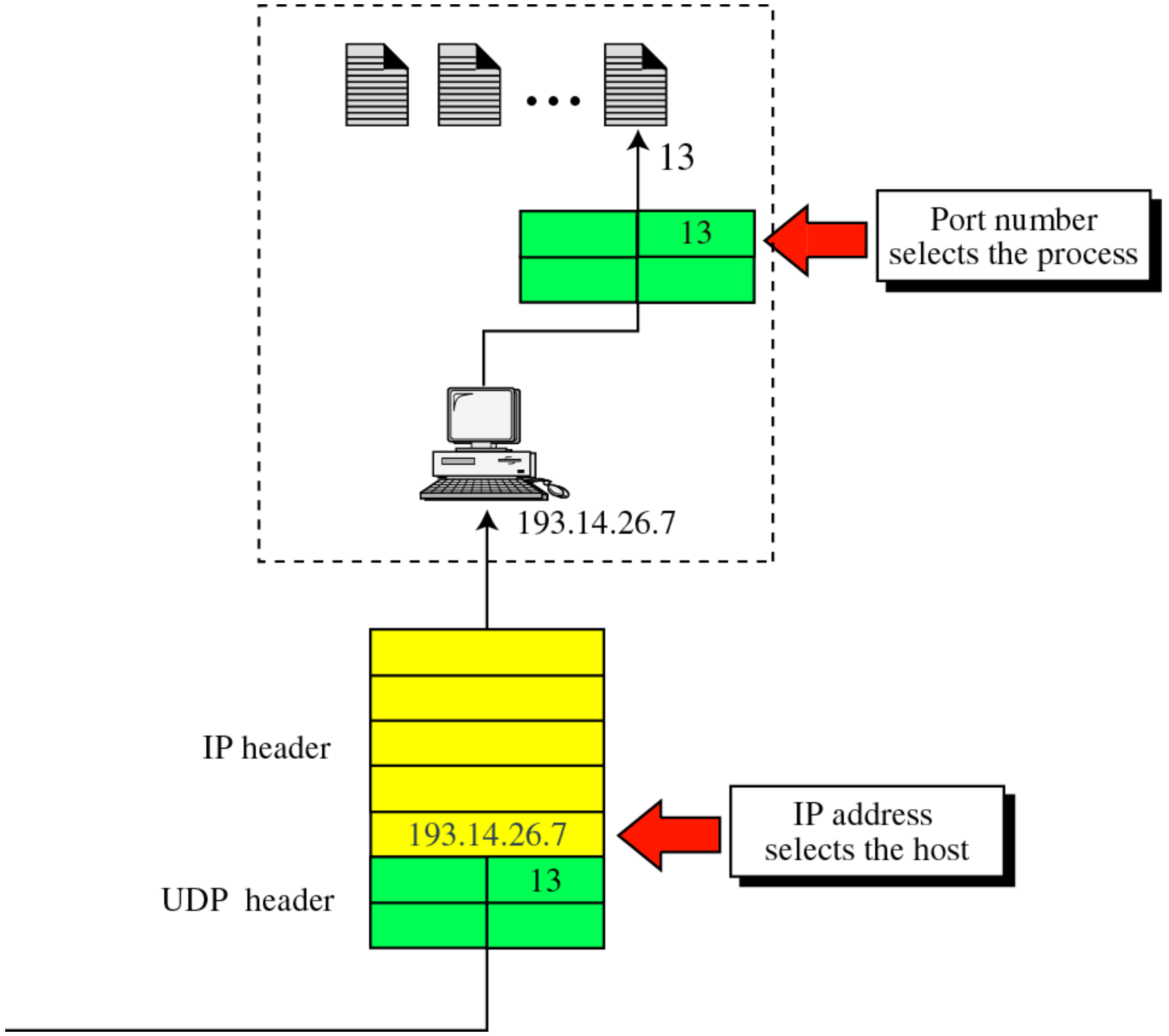
UDP versus IP



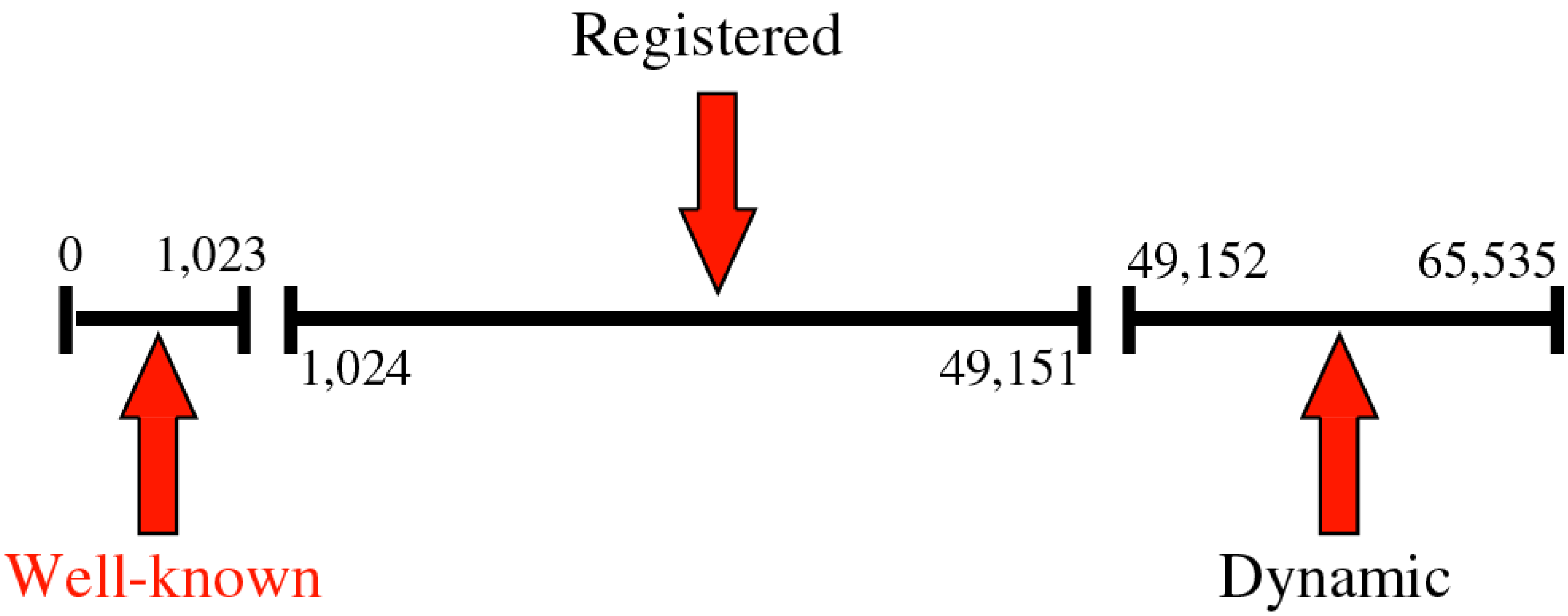
Port numbers



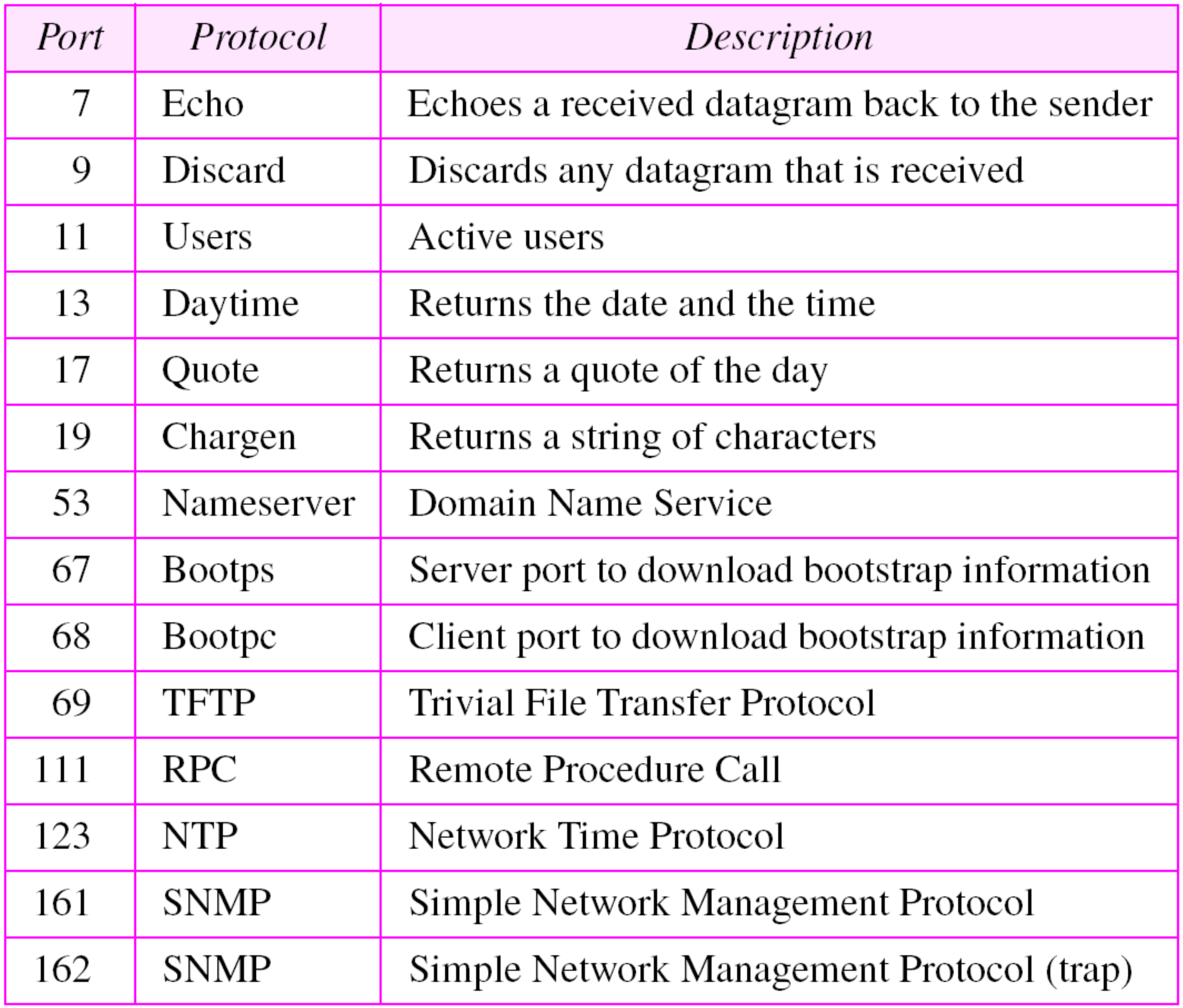
IP addresses versus port numbers



IANA ranges


Registered
0
1,023
1,024
49,151
49,152
65,535
Well-known
Dynamic

Well-known ports used with UDP



Port	*Protocol*	*Description*
7	Echo	Echoes a received datagram back to the sender
9	Discard	Discards any datagram that is received
11	Users	Active users
13	Daytime	Returns the date and the time
17	Quote	Returns a quote of the day
19	Chargen	Returns a string of characters
53	Nameserver	Domain Name Service
67	Bootps	Server port to download bootstrap information
68	Bootpc	Client port to download bootstrap information
69	TFTP	Trivial File Transfer Protocol
111	RPC	Remote Procedure Call
123	NTP	Network Time Protocol
161	SNMP	Simple Network Management Protocol
162	SNMP	Simple Network Management Protocol (trap)

EXAMPLE 1

In UNIX, the well-known ports are stored in a file called /etc/services. Each line in this file gives the name of the server and the well-known port number. We can use the grep utility to extract the line corresponding to the desired application. The following shows the port for TFTP. Note TFTP can use port 69 on either UDP or TCP.

```
$ grep tftp /etc/services
  tftp                69/tcp
  tftp                69/udp
```

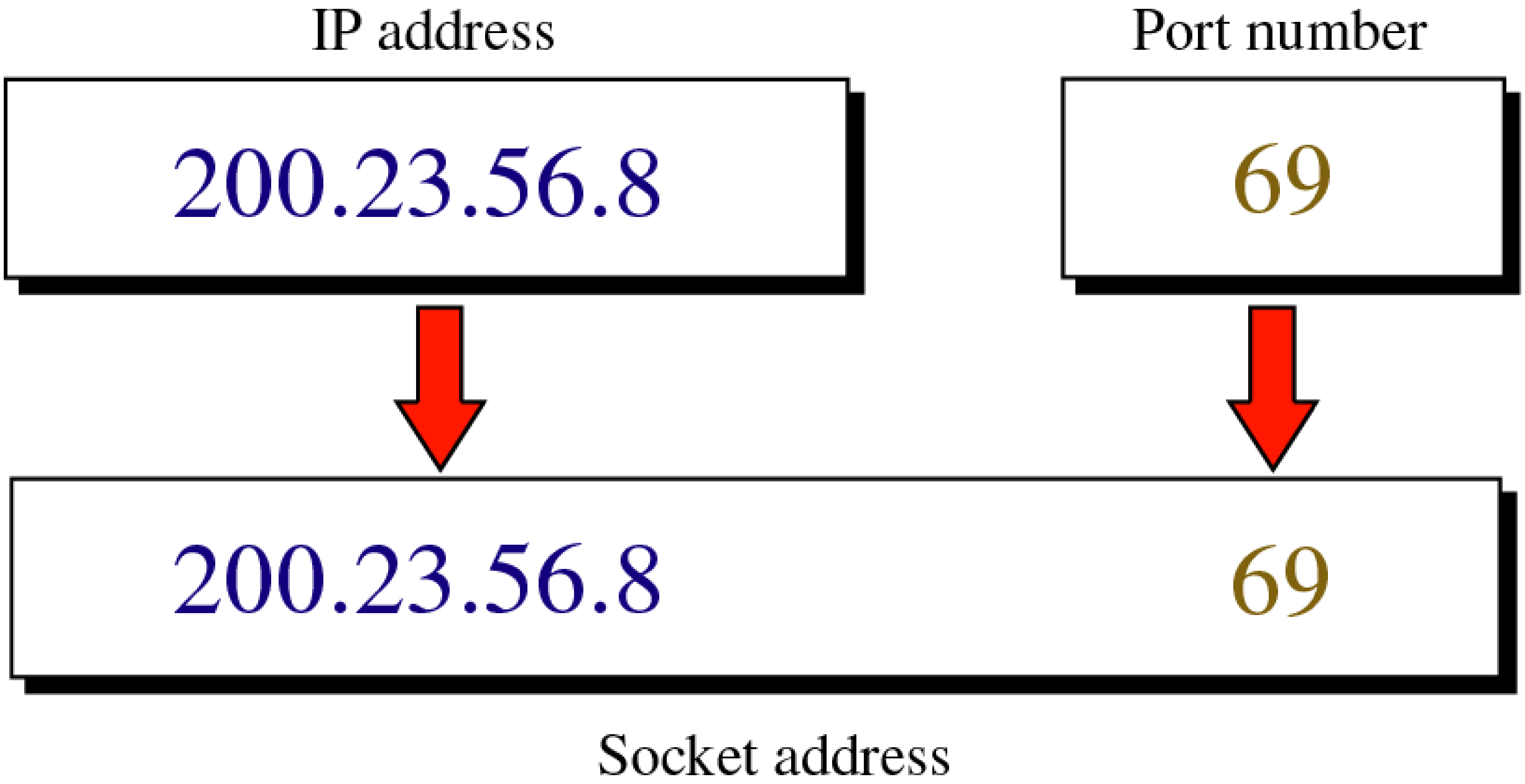
See Next Slide

EXAMPLE 1 (CONTINUED)

SNMP uses two port numbers (161 and 162), each for a different purpose, as we will see in Chapter 21.

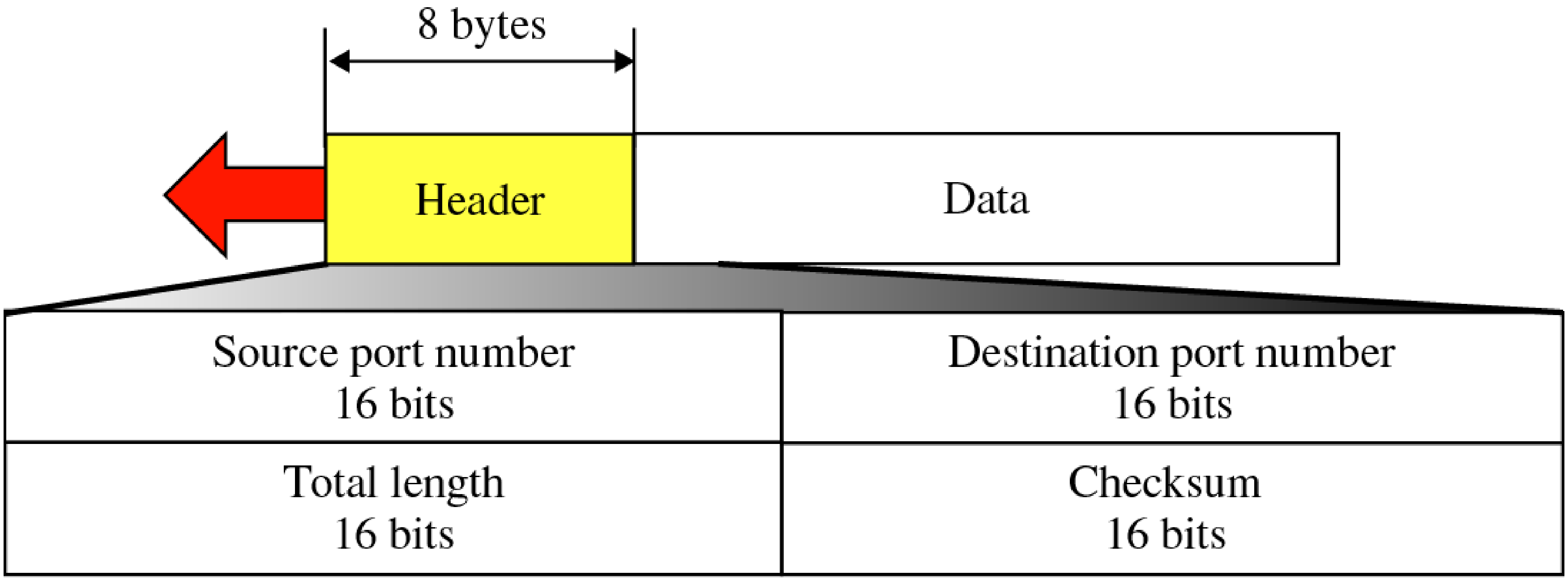
```
$ grep snmp /etc/services
  snmp       161/tcp    #Simple Net Mgmt Proto
  snmp       161/udp    #Simple Net Mgmt Proto
  snmptrap              162/udp    #Traps for SNMP
```

Socket addresses



USER DATAGRAM

User datagram format



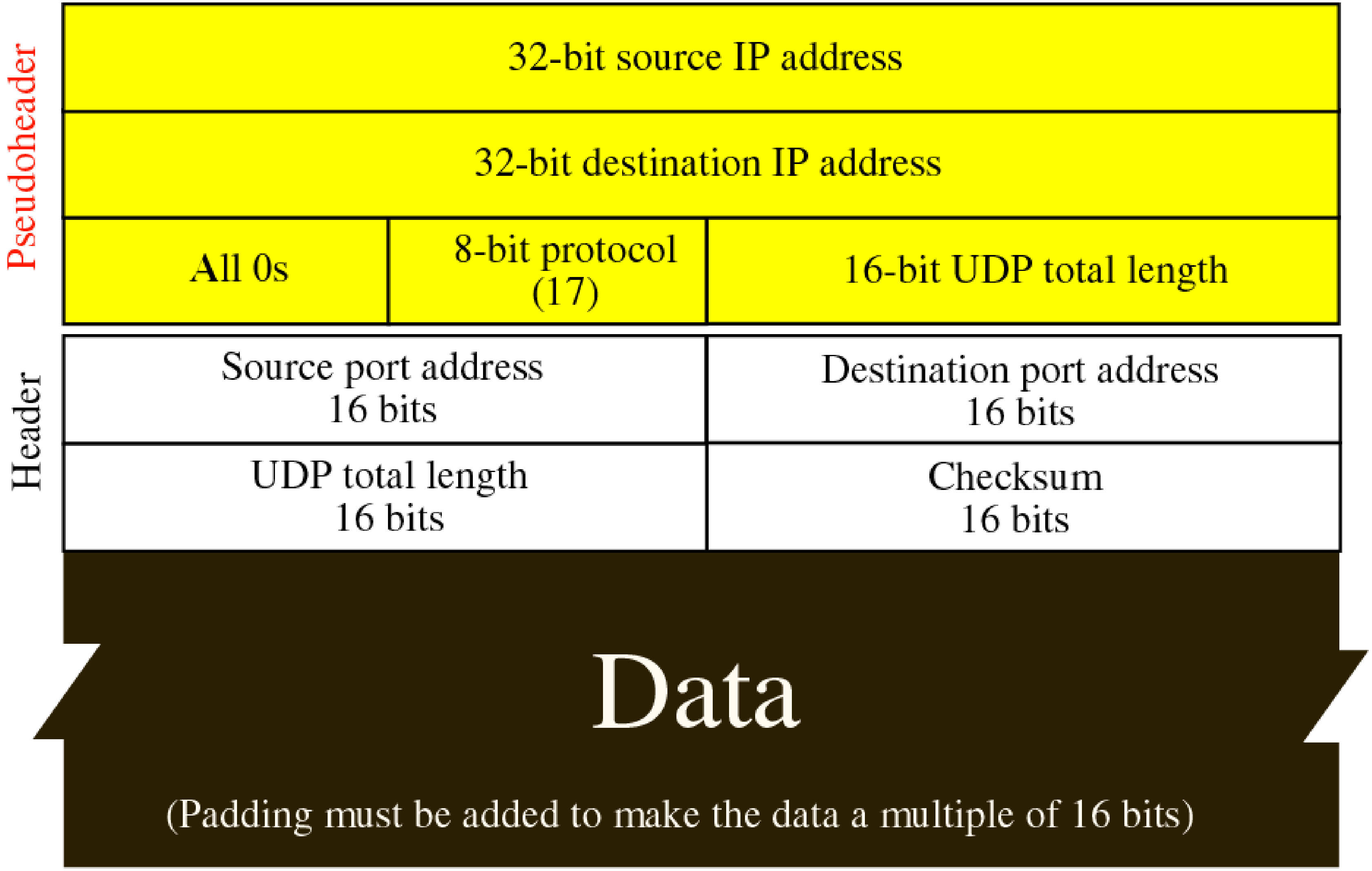
Source port number 16 bits	Destination port number 16 bits
Total length 16 bits	Checksum 16 bits

Note

UDP length =
IP length – IP header's length

CHECKSUM

Pseudoheader added to the UDP datagram



Pseudoheader

32-bit source IP address

32-bit destination IP address

All 0s

8-bit protocol (17)

16-bit UDP total length

Header

Source port address 16 bits

Destination port address 16 bits

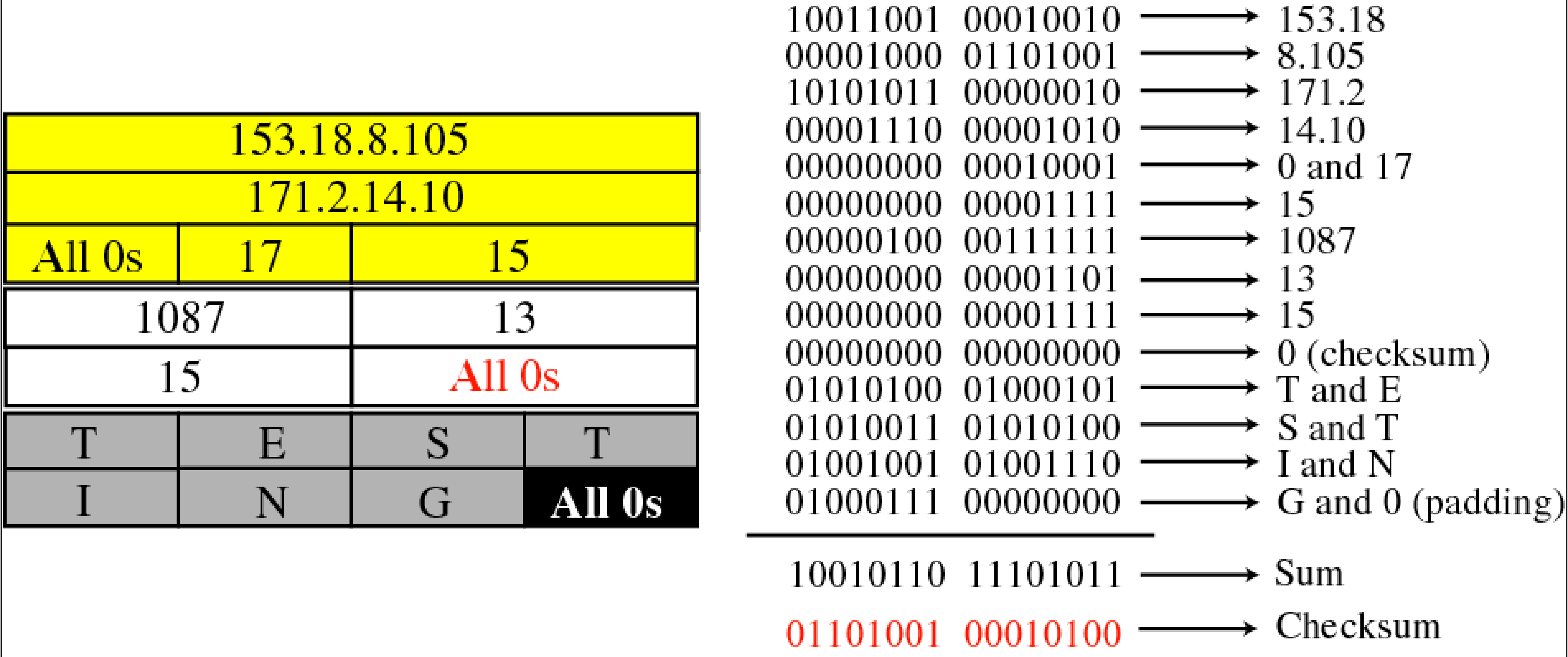
UDP total length 16 bits

Checksum 16 bits

Data

(Padding must be added to make the data a multiple of 16 bits)

Checksum calculation of a simple UDP user datagram

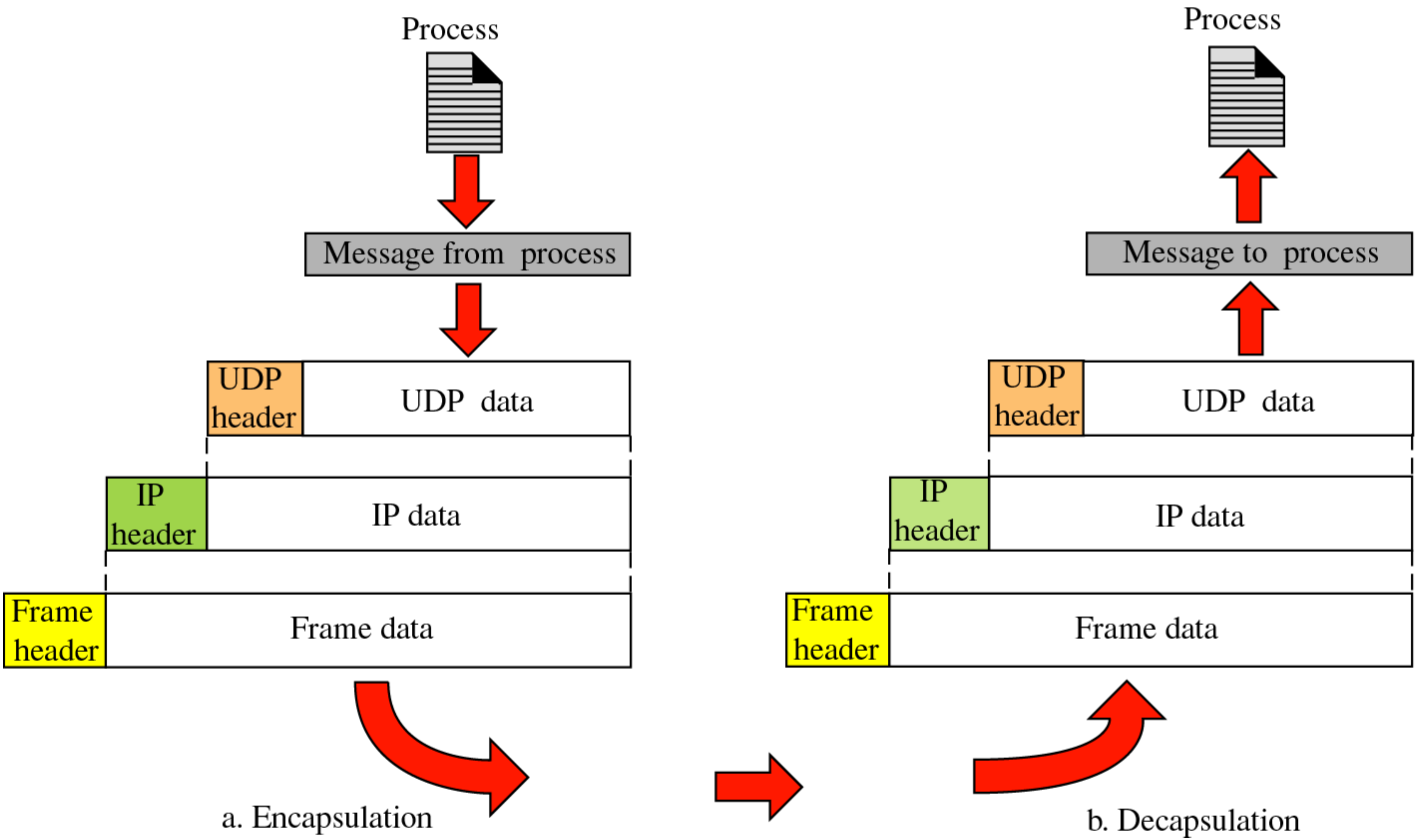


153.18.8.105			
171.2.14.10			
All 0s	17	15	
1087		13	
15		All 0s	
T	E	S	T
I	N	G	All 0s

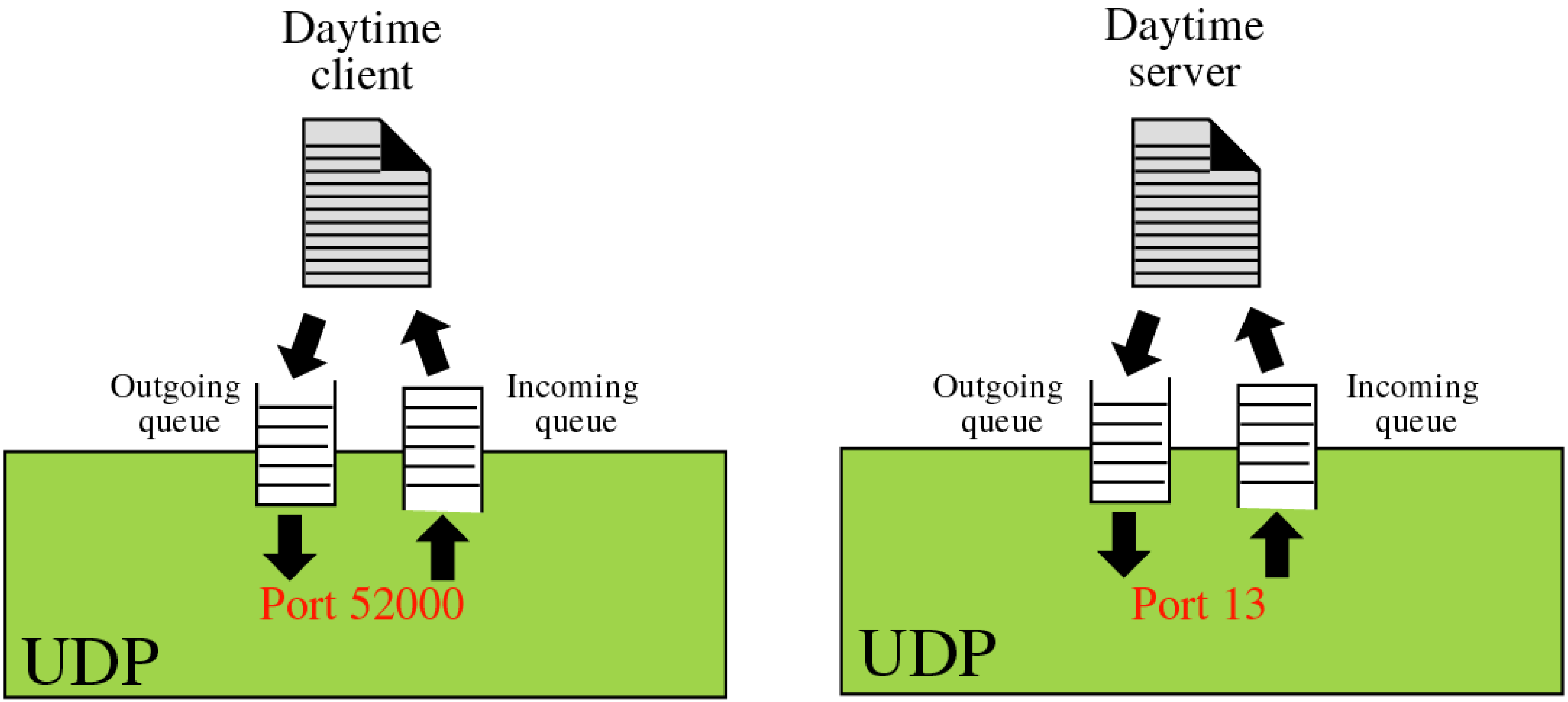
10011001 00010010 →	153.18
00001000 01101001 →	8.105
10101011 00000010 →	171.2
00001110 00001010 →	14.10
00000000 00010001 →	0 and 17
00000000 00001111 →	15
00000100 00111111 →	1087
00000000 00001101 →	13
00000000 00001111 →	15
00000000 00000000 →	0 (checksum)
01010100 01000101 →	T and E
01010011 01010100 →	S and T
01001001 01001110 →	I and N
01000111 00000000 →	G and 0 (padding)
10010110 11101011 →	Sum
01101001 00010100 →	Checksum

UDP OPERATION

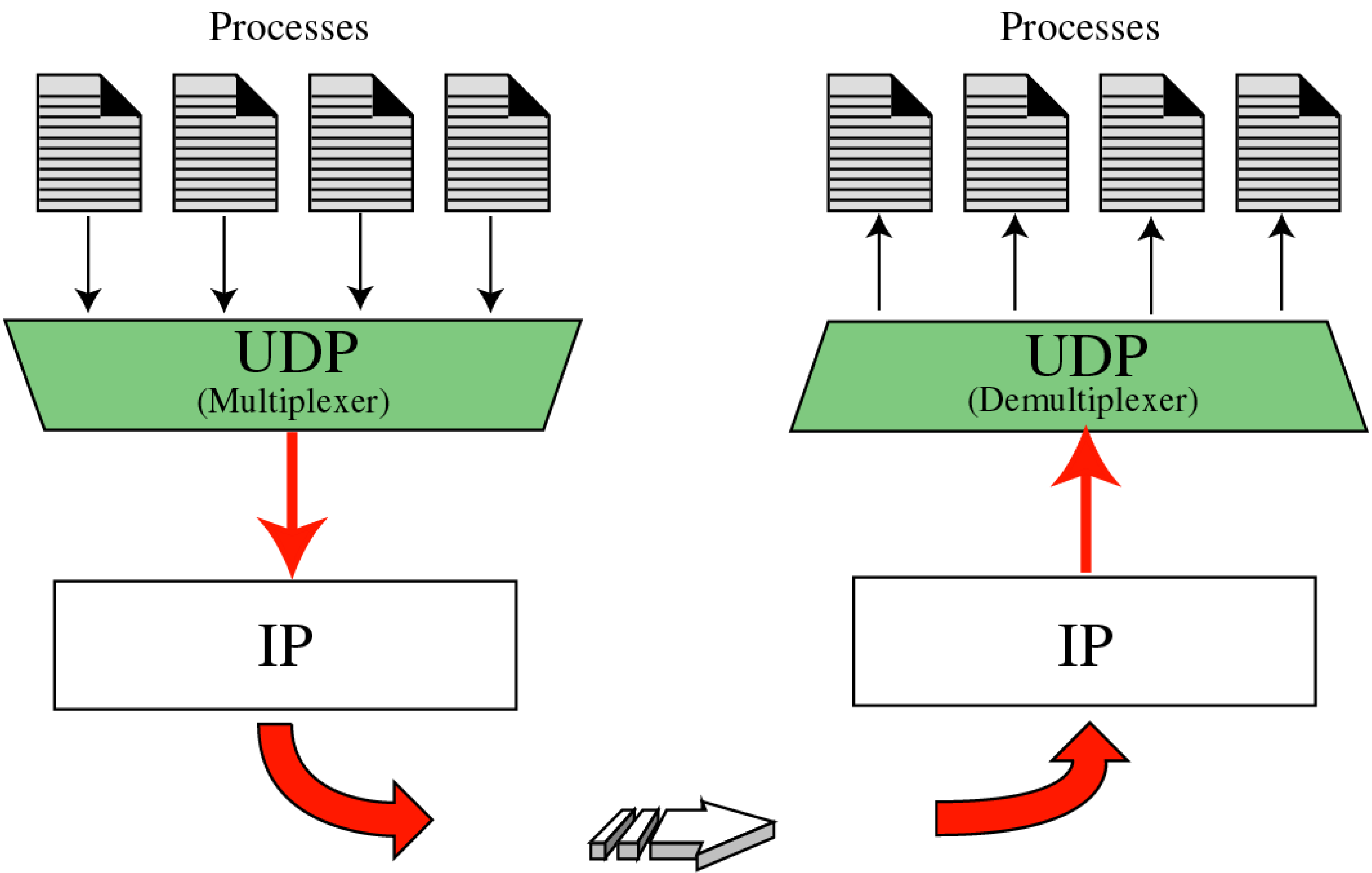
Encapsulation and decapsulation



Queues in UDP



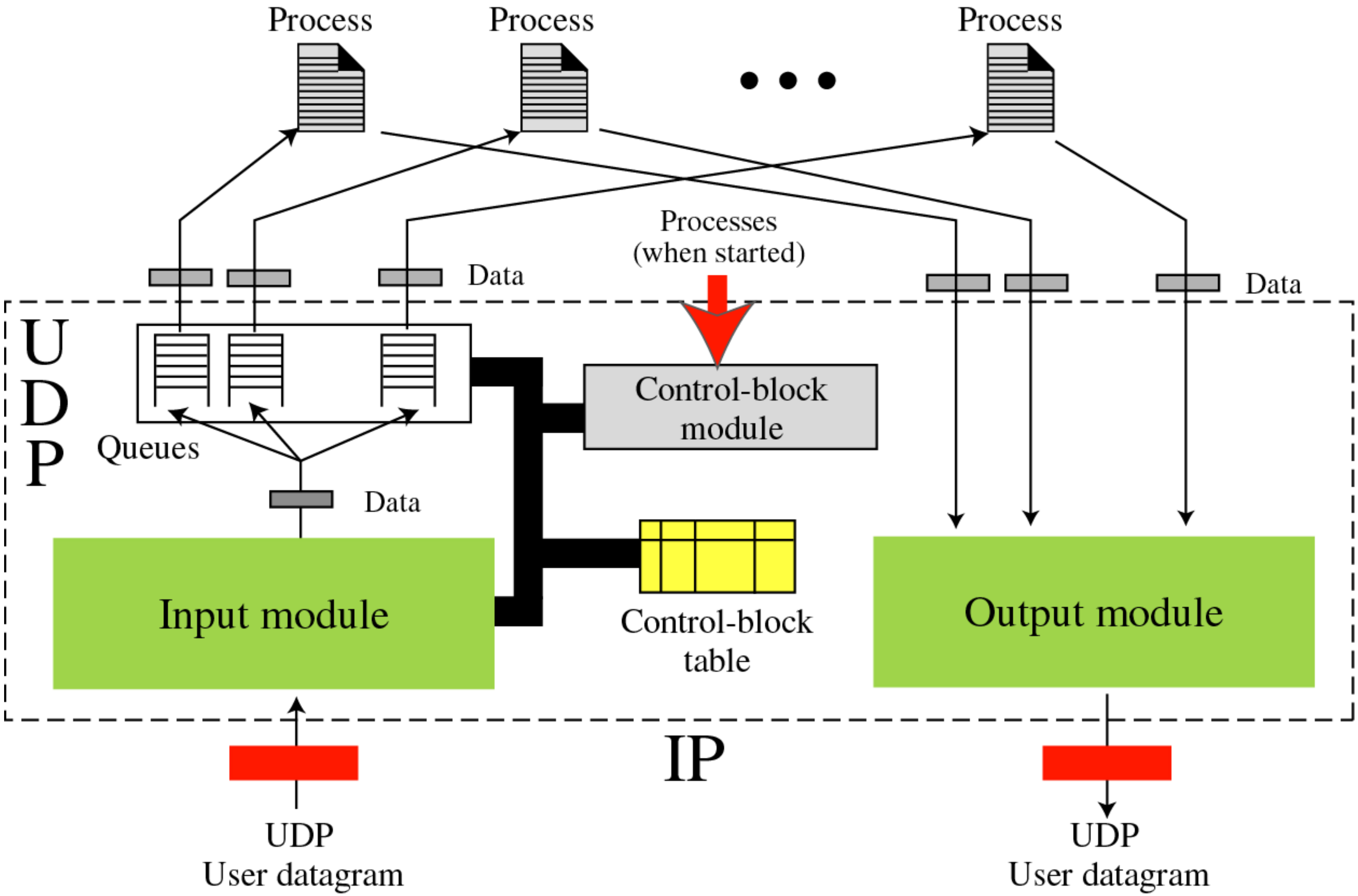
Multiplexing and demultiplexing



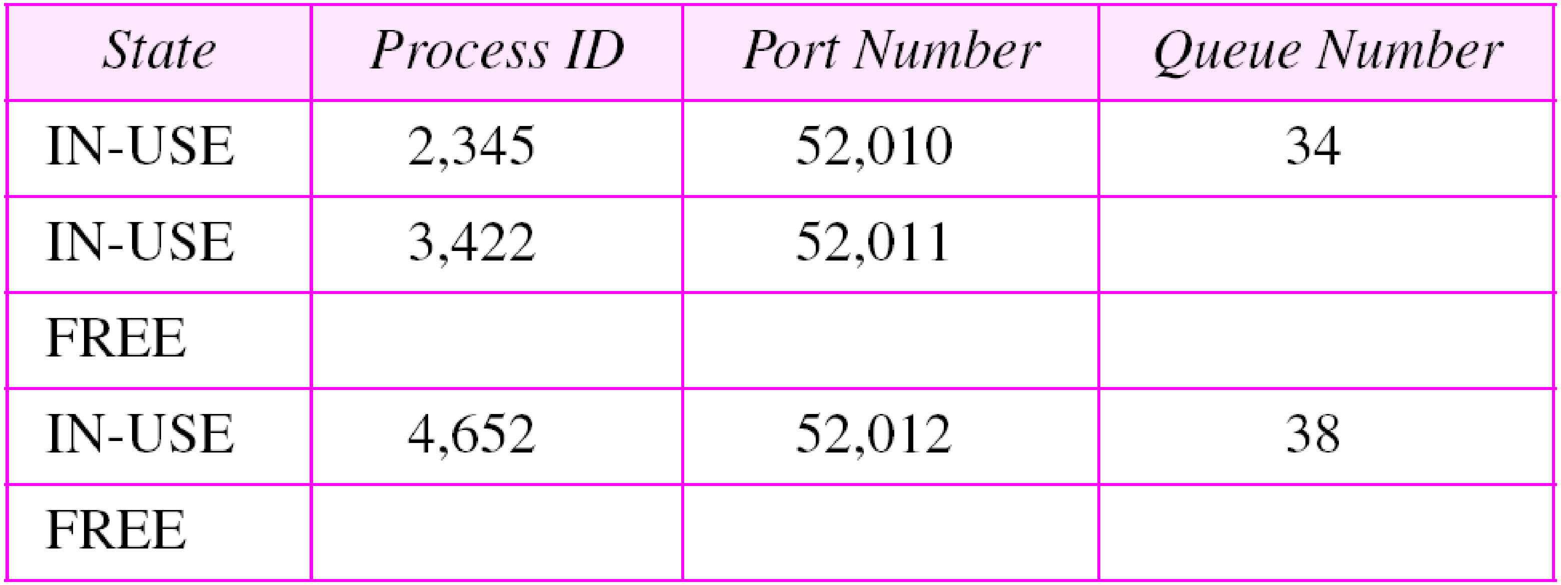
USE OF UDP

UDP PACKAGE

UDP package


Process
Process
Process
Processes
(when started)
Data
Data
U
D
P
Queues
Data
Control-block
module
Control-block
table
Input module
Output module
IP
UDP
User datagram
UDP
User datagram

The control-block table at the beginning of examples



State	*Process ID*	*Port Number*	*Queue Number*
IN-USE	2,345	52,010	34
IN-USE	3,422	52,011	
FREE			
IN-USE	4,652	52,012	38
FREE			

EXAMPLE 2

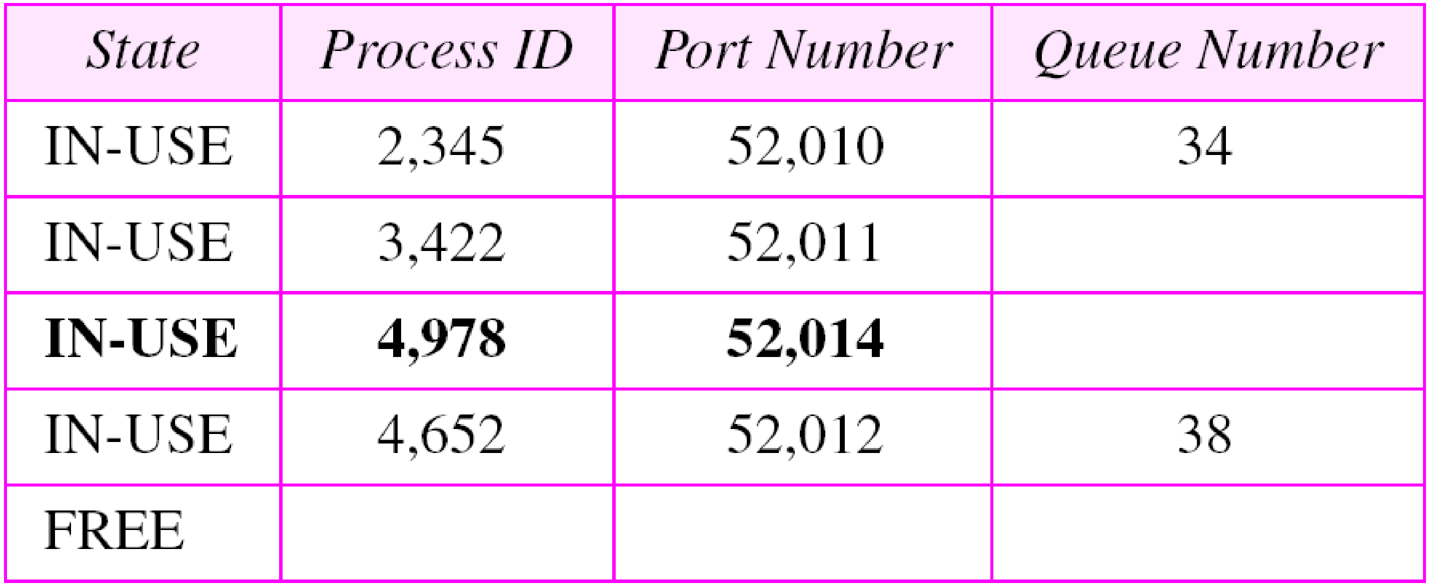
The first activity is the arrival of a user datagram with destination port number 52,012. The input module searches for this port number and finds it. Queue number 38 has been assigned to this port, which means that the port has been previously used. The input module sends the data to queue 38.The control-block table does not change.

EXAMPLE 3

After a few seconds, a process starts. It asks the operating system for a port number and is granted port number 52,014. Now the process sends its ID (4,978) and the port number to the control-block module to create an entry in the table. The module takes the first FREE entry and inserts the information received. The module does not allocate a queue at this moment because no user datagrams have arrived for this destination (see Table 11.3).

See Next Slide

Control-block table after Example 3



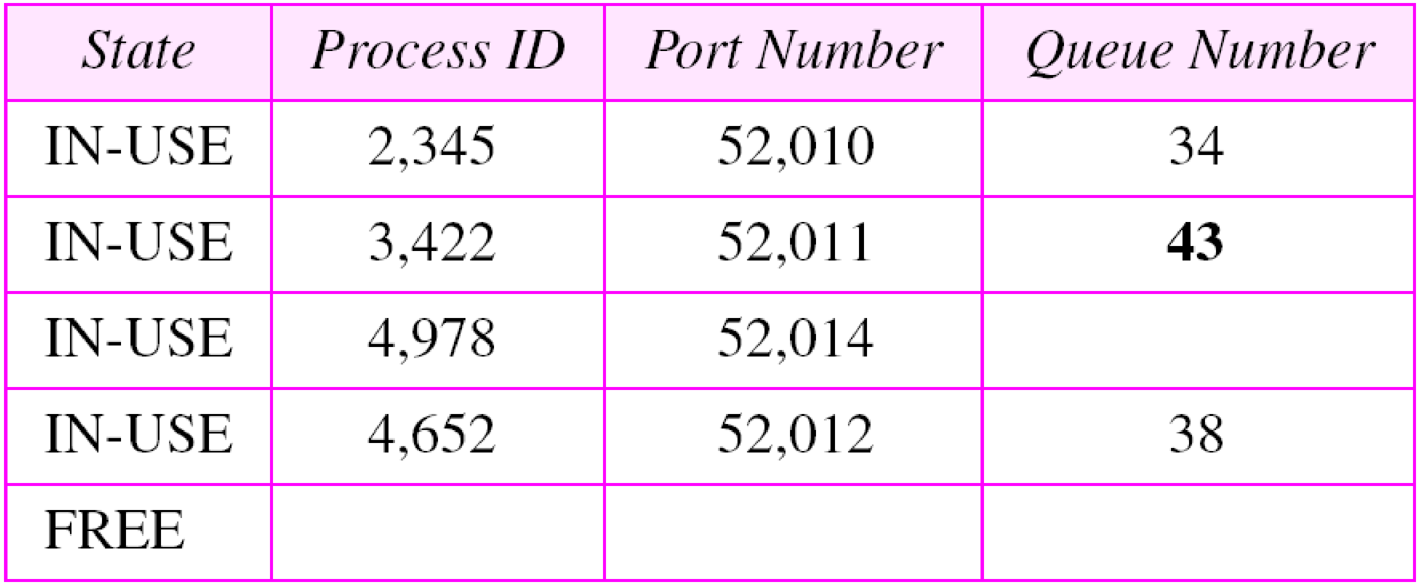
State	*Process ID*	*Port Number*	*Queue Number*
IN-USE	2,345	52,010	34
IN-USE	3,422	52,011	
IN-USE	**4,978**	**52,014**	
IN-USE	4,652	52,012	38
FREE			

EXAMPLE 4

A user datagram now arrives for port 52,011. The input module checks the table and finds that no queue has been allocated for this destination since this is the first time a user datagram has arrived for this destination. The module creates a queue and gives it a number (43). See Table 11.4.

See Next Slide

Control-block after Example 4



State	*Process ID*	*Port Number*	*Queue Number*
IN-USE	2,345	52,010	34
IN-USE	3,422	52,011	**43**
IN-USE	4,978	52,014	
IN-USE	4,652	52,012	38
FREE			

EXAMPLE 5

After a few seconds, a user datagram arrives for port 52,222. The input module checks the table and cannot find an entry for this destination. The user datagram is dropped and a request is made to ICMP to send an "unreachable port" message to the source.