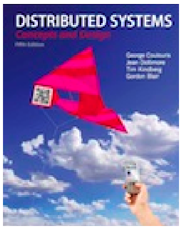

Distributed Systems: Concepts and Design

Edition 5

By George Coulouris, Jean Dollimore, Tim Kindberg and Gordon Blair
Addison-Wesley ©Pearson Education 2012

Chapter 3 Exercise Solutions

3.1 A client sends a 200 byte request message to a service, which produces a response containing 5000 bytes. Estimate the total time to complete the request in each of the following cases, with the performance assumptions listed below:

i) Using connectionless (datagram) communication (for example, UDP);

ii) Using connection-oriented communication (for example, TCP);

iii) The server process is in the same machine as the client.

[Latency per packet (local or remote,
incurred on both send and receive):5 milliseconds
Connection setup time (TCP only):5 milliseconds
Data transfer rate:10 megabits per second
MTU:1000 bytes
Server request processing time:2 milliseconds
Assume that the network is lightly loaded.]

3.1 Ans.

The send and receive latencies include (operating system) software overheads as well as network delays. Assuming that the former dominate, then the estimates are as below. If network overheads dominate, then the times may be reduced because the multiple response packets can be transmitted and received right after each other.

i) UDP: 5 + 2000/10000 + 2 + 5(5 + 10000/10000) = 37.2 milliseconds

ii) TCP: 5 + 5 + 2000/10000 + 2 + 5(5 + 10000/10000) = 42.2 milliseconds

iii)same machine: the messages can be sent by a single in memory copy; estimate interprocess data transfer rate at 40 megabits/second. Latency/message ~5 milliseconds. Time for server call:

5 + 2000/40000 + 5 + 50000/40000 = 11.3 milliseconds

3.2 Describe the network type that is used in place of wired LANs to provide connectivity for mobile devices.

3.2 Ans.

WLANs are designed for use in place of wired LANs to provide connectivity for mobile devices or simply to remove the need for a wired infrastructure to connect computers within homes and office buildings to each other and the Internet. They are in widespread use in several variants of the IEEE 802.11 standard (WiFi) offering bandwidths of 10–100 Mbps over ranges up to 1.5 kilometres.

3.3 What is the use of a switching system? What are the different types of switching used in computer networks?

3.3 Ans.

A network consists of a set of nodes connected together by circuits. To transmit information between two arbitrary nodes, a switching system is required.

There are four types of switching that are used in computer networking –

a. Broadcasting

b. circuit-switching

c. packet-switching

d. frame relay.

3.4 Make a table similar to Figure 3.5 describing the work done by the software in each protocol layer when Internet applications and the TCP/IP suite are implemented over an Ethernet.

3.4 Ans.

Layer	Description	Examples
Application	Protocols that are designed to meet the communication requirements of specific applications, often defining the interface to a service. network representation that is independent of the representations used in individual computers. Encryption is performed in this layer.	HTTP, FTP, SMTP, CORBA IIOP, Secure Sockets Layer, CORBA Data Rep
Transport	UDP: checksum validation, delivery to process ports. TCP: segmentation, flow control, acknowledgement and reliable delivery.	TCP, UDP
Network	IP addresses translated to Ethernet addresses (using ARP). IP packets segmented into Ether packets.	IP
Data link	Ethernet CSMA CD mechanism.	Ethernet MAC layer
Physical	Various Ethernet transmission standards.	Ethernet base-band signalling

3.5 What does the term "protocol" mean? What is the role of the protocol layers in a network system? Describe the role of the protocols used in the presentation layer, with examples.

3.5 Ans.

The term protocol is used to refer to a well-known set of rules and formats to be used for communication between processes in order to perform a given task.

Network software is arranged in a hierarchy of layers. Each layer presents an interface to the layers above it that extends the properties of the underlying communication system. Each module appears to communicate directly with a module at the same level in another computer in the network, but in reality data is not transmitted directly between the protocol modules at each level. Instead, each layer of network software communicates by local procedure calls with the layers above and below it.

Protocols at the presentation layer transmit data in a network representation that is independent of the representations used in individual computers, which may differ. Encryption is also performed in this layer, if required. Examples of protocols used in the presentation layer – TLS Security, CORBA Data Rep.

3.6 How does adaptive routing ensure the best route of communication between two points in the network?

3.6 Ans.

Adaptive routing determines the best route for communication between two points in the network is re-evaluated periodically, taking into account the current traffic in the network and any faults such as broken connections or routers.

3.7 Compare connectionless (UDP) and connection-oriented (TCP) communication for the implementation of each of the following application-level or presentation-level protocols:

i) virtual terminal access (for example, Telnet);

ii) file transfer (for example, FTP);

iii) user location (for example, rwho, finger);

iv) information browsing (for example, HTTP);

v) remote procedure call.

3.7 Ans.

i) The long duration of sessions, the need for reliability and the unstructured sequences of characters transmitted make connection-oriented communication most suitable for this application. Performance is not critical in this application, so the overheads are of little consequence.

ii) File calls for the transmission of large volumes of data. Connectionless would be ok if error rates are low and the messages can be large, but in the Internet, these requirements aren't met, so TCP is used.

iii) Connectionless is preferable, since messages are short, and a single message is sufficient for each transaction.

iv) Either mode could be used. The volume of data transferred on each transaction can be quite large, so TCP is used in practice.

v) RPC achieves reliability by means of timeouts and re-trys. so connectionless (UDP) communication is often preferred.

3.8 Explain how TCP ensures the reliable delivery of long sequences of bytes via stream-based programming abstraction. Why can't this happen in a UDP?

3.8 Ans.

TCP ensures the reliable delivery of long sequence of bytes via stream-based programming abstraction as TCP is connection-oriented. In the TCP mechanism before any data is transferred, the sending and receiving process must cooperate in the establishment of a bi-directional communication channel. The connection is simply an end-to-end agreement to perform reliable data transmission. UDP is connectionless and therefore does not ensure reliable delivery.

3.9 A specific problem that must be solved in remote terminal access protocols such as Telnet is the need to transmit exceptional events such as 'kill signals' from the 'terminal' to the host in advance of previously-transmitted data. Kill signals should reach their destination ahead of any other ongoing transmissions. Discuss the solution of this problem with connection-oriented and connectionless protocols.

3.9 Ans.

The problem is that a kill signal should reach the receiving process quickly even when there is buffer overflow (e.g. caused by an infinite loop in the sender) or other exceptional conditions at the receiving host.

With a connection-oriented, reliable protocol such as TCP, all packets must be received and acknowledged by the sender, in the order in which they are transmitted. Thus a kill signal cannot overtake other data already in the stream. To overcome this, an out-of-band signalling mechanism must be provided. In TCP this is called the URGENT mechanism. Packets containing data that is flagged as URGENT bypass the flow-control mechanisms at the receiver and are read immediately.

With connectionless protocols, the process at the sender simply recognizes the event and sends a message containing a kill signal in the next outgoing packet. The message must be resent until the receiving process acknowledges it.

3.10 Explain the following issues for a network with Ethernet transmission technology:

i) packet layout

ii) packet collision resolution

iii) efficiency.

3.10 Ans.

i) The packets (or more correctly, frames) transmitted by stations on the Ethernet have the following layout:

bytes: 7	*1*	*6*	*6*	*2*	*46 < length < 1500*	*4*
Preamble	S	Destination address	Source address	Length of data	Data for transmission	Checksum

ii) Ethernet uses *Carrier sensing for* packet collision resolution. Carrier sensing; the interface hardware in each station listens for the presence of a signal (known as the *carrier* by analogy with radio broadcasting) in the medium. When a station wishes to transmit a packet, it waits until no signal is present in the medium and then begins to transmit.

iii) The efficiency of an Ethernet is the ratio of the number of packets transmitted successfully as a proportion of the theoretical maximum number that could be transmitted without collisions.

3.11 For an organization that has Internet connectivity, suggest a scheme that would improve its security and enable it to provide services to its customers and other external users, while also allowing internal users to access information and services.

3.11 Ans.

A firewall security policy may be used to monitor and control all communication into and out of an intranet. A firewall is implemented by a set of processes that act as a gateway to an intranet applying a security policy determined by the organization.

3.12 Show the sequence of changes to the routing tables in Figure 3.8 that would occur (according to the RIP algorithm given in Figure 3.9) after the link labelled 3 in Figure 3.7 is broken.

3.12 Ans.

Routing tables with changes shown in red (grey in monochrome printouts):

Step 1: costs for routes that use Link 3 have been set to ∞ at A, D

Routings from A		
To	*Link*	*Cost*
A	local	0
B	1	1
C	1	2
D	3	∞
E	1	2

Routings from B		
To	*Link*	*Cost*
A	1	1
B	local	0
C	2	1
D	1	2
E	4	1

Routings from C		
To	*Link*	*Cost*
A	2	2
B	2	1
C	local	0
D	5	2
E	5	1

Routings from D		
To	*Link*	*Cost*
A	3	∞
B	3	∞
C	6	2
D	local	0
E	6	1

Routings from E		
To	*Link*	*Cost*
A	4	2
B	4	1
C	5	1
D	6	1
E	local	0

Step 2: after first exchange of routing tables

Routings from A		
To	*Link*	*Cost*
A	local	0
B	1	1
C	1	2
D	3	∞
E	1	2

Routings from B		
To	*Link*	*Cost*
A	1	1
B	local	0
C	2	1
D	1	∞
E	4	1

Routings from C		
To	*Link*	*Cost*
A	2	2
B	2	1
C	local	0
D	5	2
E	5	1

Routings from D		
To	*Link*	*Cost*
A	3	∞
B	3	∞
C	6	2
D	local	0
E	6	1

Routings from E		
To	*Link*	*Cost*
A	4	2
B	4	1
C	5	1
D	6	1
E	local	0

Step 3: after second exchange of routing tables

Routings from A		
To	*Link*	*Cost*
A	local	0
B	1	1
C	1	2
D	3	∞
E	1	2

Routings from B		
To	*Link*	*Cost*
A	1	1
B	local	0
C	2	1
D	4	2
E	4	1

Routings from C		
To	*Link*	*Cost*
A	2	2
B	2	1
C	local	0
D	5	2
E	5	1

Routings from D		
To	*Link*	*Cost*
A	6	3
B	6	2
C	6	2
D	local	0
E	6	1

Routings from E		
To	*Link*	*Cost*
A	4	2
B	4	1
C	5	1
D	6	1
E	local	0

Step 4: after third exchange of routing tables.

Routings from A		
To	*Link*	*Cost*
A	local	0
B	1	1
C	1	2
D	1	3
E	1	2

Routings from B		
To	*Link*	*Cost*
A	1	1
B	local	0
C	2	1
D	4	2
E	4	1

Routings from C		
To	*Link*	*Cost*
A	2	2
B	2	1
C	local	0
D	5	2
E	5	1

Routings from D		
To	*Link*	*Cost*
A	6	3
B	6	2
C	6	2
D	local	0
E	6	1

Routings from E		
To	*Link*	*Cost*
A	4	2
B	4	1
C	5	1
D	6	1
E	local	0

3.13 Use the diagram in Figure 3.13 as a basis for an illustration showing the segmentation and encapsulation of an HTTP request to a server and the resulting reply. Assume that request is a short HTTP message, but the reply includes at least 2000 bytes of HTML.

3.13 Ans.

Left to the reader.

3.14 Consider the use of TCP in a Telnet remote terminal client. How should the keyboard input be buffered at the client? Investigate Nagle's and Clark's algorithms [Nagle 1984, Clark 1982] for flow control and compare them with the simple algorithm described on page 103 when TCP is used by

(a) a web server,

(b) a Telnet application,

(c) a remote graphical application with continuous mouse input.

3.14 Ans.

The basic TCP buffering algorithm described on p. 105 is not very efficient for interactive input. Nagle's algorithm is designed to address this. It requires the sending machine to send any bytes found in the output buffer, then wait for an acknowledgement. Whenever an acknowledgement is received, any additional characters in the buffer are sent. The effects of this are:

a) For a web server: the server will normally write a whole page of HTML into the buffer in a single *write*. When the *write* is completed, Nagle's algorithm will send the data immediately, whereas the basic algorithm would wait 0.5 seconds. While the Nagle's algorithm is waiting for an acknowledgement, the server process can write additional data (e.g. image files) into the buffer. They will be sent as soon as the acknowledgement is received.

b) For a remote shell (Telnet) application: the application will write individual key strokes into the buffer (and in the normal case of full duplex terminal interaction they are echoed by the remote host to the Telnet client for display). With the basic algorithm, full duplex operation would result in a delay of 0.5 seconds before any of the characters typed are displayed on the screen. With Nagle's algorithm, the first character typed is sent immediately and the remote host echoes it with an acknowledgement piggy-backed in the same packet. The acknowledgement triggers the sending of any further characters that have been typed in the intervening period. So if the remote host responds sufficiently rapidly, the display of typed characters appears to be instantaneous. But note that a badly-written remote application that reads data from the TCP buffer one character at a time can still cause problems - each read will result in an acknowledgement indicating that one further character should be sent - resulting in the transmission of an entire IP frame for each character. Clarke [1982] called this the *silly window syndrome*. His solution is to defer the sending of acknowledgements until there is a substantial amount of free space available.

c) For a continuous mouse input (e.g. sending mouse positions to an X-Windows application running on a compute server): this is a difficult form of input to handle remotely. The problem is that the user should see a smooth feedbvack of the path traced by the mouse, with minimal lag. Neither the basic TCP algorithm nor Nagle's nor Clarke's algorithm achieves this very well. A version of the basic algorithm with a short timeout (0.1 seconds) is the best that can be done, and this is effective when the network is lightly loaded and has low end-to-end latency - conditions that can be guaranteed only on local networks with controlled loads.
See Tanenbaum [1996] pp. 534-5 for further discussion of this.

3.15 Construct a network diagram similar to Figure 3.10 for the local network at a shopping mall or for your college campus.

3.15 Ans.

Left to the reader.

3.16 Describe how you would configure a firewall to protect the local network at your institution or company. What incoming and outgoing requests should it intercept?

3.16 Ans.

Left to the reader.

3.17 Consider a system where the underlying network is an Ethernet. How does the address resolution module convert the 32-bit Internet address to a 48-bit Ethernet address?

3.17 Ans.

Translation of 32-bit Internet addresses to 48-bit Ethernet addresses (This translation is network technology-dependent)-

a. Some hosts are connected directly to Internet packet switches; IP packets can be routed to them without address translation.

b. Some local area networks allow network addresses to be assigned to hosts dynamically, and the addresses can be conveniently chosen to match the host identifier portion of the Internet address – translation is simply a matter of extracting the host identifier from the IP address.

c. For Ethernets and some other local networks the network address of each computer is hard-wired into its network interface hardware and bears no direct relation to its Internet address – translation depends upon knowledge of the correspondence between IP addresses and addresses for the hosts on the local network and is done using an address resolution protocol (ARP).

3.18 Do all the computers and devices that access the Internet in a LAN need to be assigned globally unique IP addresses? What mechanism is available to deal with such issues?

3.18 Ans.

Computers that are attached to a local network and access the Internet through a NAT-enabled router can rely upon the router to redirect incoming UDP and TCP packets for them. As a result in such network not all of the computers and devices that access the Internet need to be assigned globally-unique IP addresses.