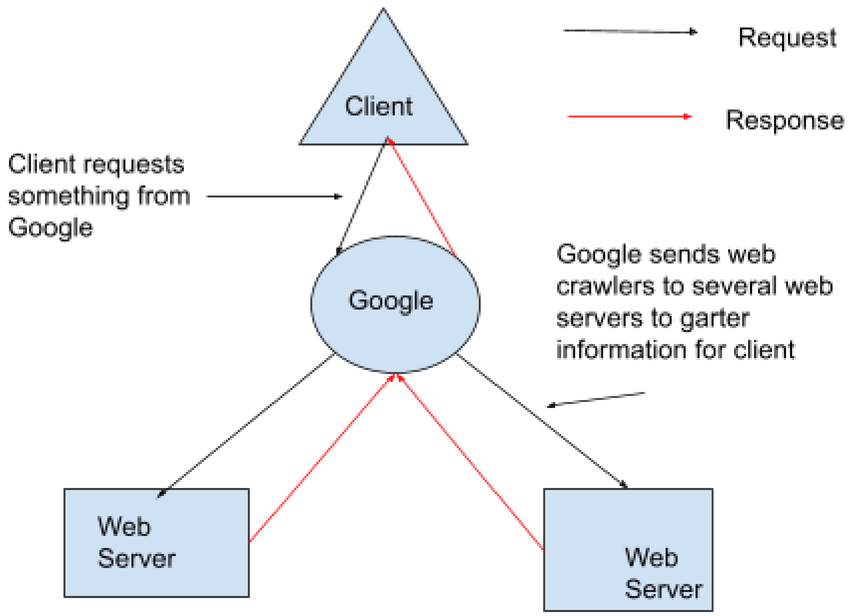
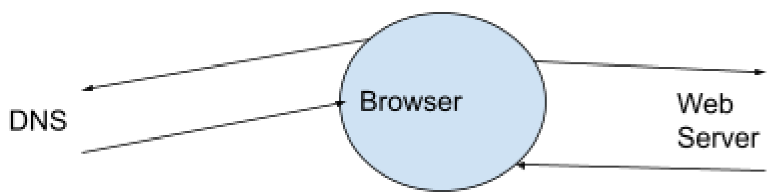
2.3) Describe and illustrate the client-server architecture of one or more major Internet applications (for example, the Web, email or netnews). page 46

- A web application (client), such as a search engine (e.g. Google), works as being both a client and a server. A user will request a search form Google (client browser request), Google will then have web crawlers request information from several web servers for said information. The information is aggregated and returned to the client. Browsers are clients of both DNS and Web Servers as they request information from the DNS for their domain name, and the web server (HTTP) for the information about their domain.
- 

- 

2.9) Consider a hypothetical car hire company and sketch out a three-tier solution to the provision of their underlying distributed car hire service. Use this to illustrate the benefits and drawbacks of a three-tier solution considering issues such as performance, scalability, dealing with failure and also maintaining the software over time. page 52

- Let's start with what a three-tier solution may consist of:
    - Web-based front-end where the user can interact and request a car
    - Middle-tier support that helps operations associated with core issues such as locating a car, pricing, etc.
    - Backend/Database: Stores all the data associated with the current stock.
- This system would lacks optimal performance due to the latency caused by the three layers (web requests middle which requests the database). However, processing information should be pretty high as the system is spread out via three systems allowing for better scalability than a 2-layer system.
- Dealing with failure: due to the three layers there is a higher chance of failure occurring (more moving parts). They would also be harder to deal with because if the front-end or back-end go down such as obvious issues such as the user not being able to interact

with the inventory/go shopping, or the lack of any information being displayed on the front-end.

- Future: This system would be better for future usage in the company because it would be easier to modify parts of the system as they are modular. Issues relating to interface are one end, actual application issues are one layer, and the database storage is another.

2.11) Consider a simple server that carries out client requests without accessing other servers. Explain why it is generally not possible to set a limit on the time taken by such a server to respond to a client request. What would need to be done to make the server able to execute requests within a bounded time? Is this a practical option? page 62

- Let's consider a multithreaded simple server that gets multiple requests. It can execute these requests concurrently on its threads, however, it may be unable to allocate enough time to a particular request within a given time limit as there are other threads that need to be taken care of. If the server finishes them one at a time instead of concurrently, the other requests may wait in the queue for an unknown amount of time until that one request is completed. If it is needed that requests are executed within a bounded time then there should be a limit on the number of clients the server will accept at a time to suit its capacity to be able to handle all the requests within a time. If more clients are needed to be handled then increase processing with a processor with more threads. The more powerful processor (or multiple servers needed) to handle more clients could be costly so it isn't the most practical option but it works.

2.14) Consider two communication services for use in asynchronous distributed systems. In service A, messages may be lost, duplicated or delayed and checksums apply only to headers. In service B, messages may be lost, delayed or delivered too fast for the recipient to handle them, but those that are delivered arrive with the correct contents. Describe the classes of failure exhibited by each service. Classify their failures according to their effects on the properties of validity and integrity. Can service B be described as a reliable communication service? page 67, page 71

- A service:
    - Arbitrary failures are possible asas checksums don't apply to messages bodies possibly leading to their corruption. Also because there will be duplicated messages. Omission failures are also possible because a message may be lost. Because of lost messages and possibility of corrupted messages and duplicated messages, it lacks both validity and integrity.
- B Service:
    - Omission failures due to the lost and/or dropped messages due to the inability to process them. Because of this it passes the integrity test but not validity because it loses messages, and therefore not a reliable form of a communication service.

3.1) A client sends a 200 byte request message to a service, which produces a response containing 5000 bytes. Estimate the total time required 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) when the server process is in the same machine as the client. [Latency per packet (local or remote, incurred on both send and receive): 5 ms Connection setup time (TCP only): 5 ms Data transfer rate: 10 Mbps MTU: 1000 bytes Server request processing time: 2 ms Assume that the network is lightly loaded.]

- i) UDP: 37.2 milliseconds
- ii)TCP: 42.2 milliseconds
- iii) IT depends on the interprocess data transfer rate which isn't given because it's a local machine, but everything else would stay the same. But assuming an interprocess data transfer rate of 40 megabits/second. 11.3 milliseconds

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

- ii) file transfer (for example, FTP);
- File transfer via the internet needs to have low error rates, and handle large message sizes (I send large group chat messages). UDP has a substantial error rate because it is not guaranteed all the data will get to the other end, in addition it can't handle very large packets of data (64000 bytes only). TCP can meet these criteria as it is a stable connection with guaranteed message delivery, and can send large messages.
-

3.8) Explain how it is possible for a sequence of packets transmitted through a wide area network to arrive at their destination in an order that differs from that in which they were sent. Why can't this happen in a local network?

- Packets transmitted via a wide area network paths are determined dynamically. Certain routes will have to go through slower hardware or more hardware which can cause the packets to take a longer time to get to their destination (i.e. a packet has to go through a 1987 switch vs a 2021 switch). This is because UDP sends each packet independent of a connection and thus can cause the packets to arrive in differing orders. TCP connections can arrive in different orders too but because they add sequence numbers to their packets they are able to be re-ordered by the host in a proper fashion.
- This can't happen on a local network because there is only one channel connecting the hosts together. Packets are sent and received in the order they are requested because of this.

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

Routing from A

To	Link	Cost
A	local	0
B	1	1
C	1	2
D	3	infinity
E	1	2

Routing from B

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

Routing from C

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

Routing from D

To	Link	Cost
A	3	infinity
B	3	infinity
C	6	2
D	local	0
E	6	1

Routing from E

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

STEP 2: exchange of routing tables

Routing from A

To	Link	Cost
A	local	0
B	1	1
C	1	2
D	3	infinity
E	1	2

Routing from B

To	Link	Cost
A	1	1
B	local	0
C	2	1
D	1	infinity
E	4	1

Routing from C

To	Link	Cost
A	2	2
B	2	1
C	local	0

D	5	2
E	5	1

Routing from D

To	Link	Cost
A	3	infinity
B	3	infinity
C	6	2
D	local	0
E	6	1

Routing from E

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