CHAPTER 19

Network Structure

Practice Exercises

19.1 Why would it be a bad idea for routers to pass broadcast packets between networks? What would be the advantages of doing so?

Answer:
All broadcasts would be propagated to all networks, causing a *lot* of network traffic. If broadcast traffic were limited to important data (such as network routing information), then broadcast propagation would save routers from having to run special software to watch for the information and rebroadcast it.

19.2 Discuss the advantages and disadvantages of caching name translations for computers located in remote domains.

Answer:
There is a performance advantage to caching name translations for computers located in remote domains: repeated resolution of the same name from different computers located in the local domain can be performed locally without requiring a remote name-lookup operation. The disadvantage is that there may be inconsistencies in the name translations when updates are made in the mapping of names to IP addresses. These consistency problems can be solved by invalidating translations (by the server telling the caching clients that a change has occurred), which require state to be managed regarding which computers are caching a certain translation and also require a number of invalidation messages. Alternatively, the problems can be solved by using leases, whereby the caching entity invalidates a translation after a certain period of time. The latter approach requires less state and no invalidation messages but might suffer from temporary inconsistencies.

19.3 What are two formidable problems that designers must solve to implement a network system that has the quality of transparency?

Answer:

One issue is making all the processors and storage devices seem transparent across the network. In other words, the distributed system should appear as a centralized system to users. The Andrew file system and NFS provide this feature: the distributed file system appears to the user as a single file system, but in reality it may be distributed across a network.

Another issue concerns the mobility of users. We want to allow users to connect to the "system" rather than to a specific machine (although in reality they may be logging in to a specific machine somewhere in the distributed system).

19.4 To build a robust distributed system, you must know what kinds of failures can occur.

a. List three possible types of failure in a distributed system.

b. Specify which of the entries in your list also are applicable to a centralized system.

Answer:

Three common failures in a distributed system are: (1) network link failure, (2) host failure, (3) site failure failure. Both (2) and (3) are failures that could also occur in a centralized system, whereas a network link failure can occur only in a networked distributed system.

19.5 Is it always crucial to know that the message you have sent has arrived at its destination safely? If your answer is "yes," explain why. If your answer is "no," give appropriate examples.

Answer:

No. Many status-gathering programs work from the assumption that packets may not be received by the destination system. These programs generally *broadcast* a packet and assume that at least some other systems on their network will receive the information. For instance, a daemon on each system might broadcast the system's load average and number of users. This information might be used for process migration target selection. Another example is a program that determines if a remote site is both running and accessible over the network. If the program sends a query and gets no reply, it knows the system cannot currently be reached.

19.6 A distributed system has two sites, A and B. Consider whether site A can distinguish among the following:

a. B goes down.

b. The link between A and B goes down.

c. B is extremely overloaded, and its response time is 100 times longer than normal.

What implications does your answer have for recovery in distributed systems?

Answer:

One technique would be for B to periodically send an ***I-am-up*** message to A indicating that it is still alive. If A does not receive an ***I-am-up*** message, it can assume that either B—or the network link—is down. Note that an ***I-am-up*** message does not allow A to distinguish between the types of failure. One technique that allows A to better determine if the network is down is to send an ***Are-you-up*** message to B using an alternate route. If A receives a reply, it can determine that, indeed, the network link is down and B is up.

If we assume that A knows B is up and is reachable (via the ***I-am-up*** mechanism) and that A has some value *N* that indicates a normal response time, A could monitor the response time from B and compare the values, allowing A to determine if B is overloaded or not.

The implications of both of these techniques are that A could choose another host—say C—in the system if B is either down, unreachable, or overloaded.

Exercises

19.7 Even though the OSI model of networking specifies seven layers of functionality, most computer systems use fewer layers to implement a network. Why do they use fewer layers? What problems could the use of fewer layers cause?

Answer:

A particular protocol may achieve the same functionality as the OSI model in fewer layers by using one layer to implement functionality provided in two (or more) layers in the OSI model. Other models may decide there is no need for certain OSI layers. For example, the presentation and session layers are absent in the TCP/IP protocol. Another reason may be that certain OSI layers do not apply to a certain implementation. Again using TCP/IP as an example, no data link or physical layer is specified by the protocol. The thinking behind TCP/IP is that the functionalities in the data link and physical layers are not pertinent to TCP/IP—it merely assumes some network connection is provided, whether Ethernet, wireless, or some other connection.

A potential problem with implementing fewer layers is that certain functionalities may not be provided.

19.8 What are the advantages of using dedicated hardware devices for routers? What are the disadvantages of using these devices compared with using general-purpose computers?

Answer:

The advantages are that dedicated hardware devices for routers and gateways are very fast, as all their logic is provided in hardware (firmware). Using a general-purpose computer for a router or gateway means that routing functionality is provided in software—which is not as fast as providing the functionality directly in hardware.

A disadvantage is that routers or gateways as dedicated devices may be more costly than the off-the-shelf components that comprise a modern general purpose computer.

19.9 In what ways is using a name server better than using static host tables? What problems or complications are associated with name servers? What methods could you use to decrease the amount of traffic name servers generate to satisfy translation requests?

Answer:
Name servers require their own protocol, so they add a complication to the system. Also, if a name server is down, host information may become unavailable. Backup name servers are required to avoid this problem. Caches can be used to store frequently requested host information to cut down on network traffic.

19.10 The lower layers of the OSI network model provide datagram service, with no delivery guarantees for messages. A transport-layer protocol such as TCP is used to provide reliability. Discuss the advantages and disadvantages of supporting reliable message delivery at the lowest possible layer.

Answer:
Many applications might not require reliable message delivery. For instance, a coded video stream could recover from packet losses by performing interpolations to derive lost data. In fact, in such applications, retransmitted data would be of little use, since they would arrive much later than the optimal time and not conform to real-time guarantees. For such applications, reliable message delivery at the lowest level is an unnecessary feature and might result in increased message traffic, most of which would be useless, thereby resulting in performance degradation. In general, the lowest levels of the networking stack need to support the minimal amount of functionality required by all applications and leave extra functionality to be implemented at the upper layers.

19.11 A DNS name can map to multiple servers, such as www.google.com. However, if we run the program shown in Figure 19.4, we get only one IP address. Modify the program to display all the server IP addresses instead of just one.

Answer:
A sample piece of Java code that displays all server IP addresses associated with a host through the use of the getAllByName() method is shown in Figure 19.101.

When run on www.google.com in June 2017, the code yielded these IP addresses:

- `74.125.70.104`
- `74.125.70.99`
- `74.125.70.105`
- `74.125.70.106`

```
/**
* Usage: java DNSLookUpAll <name>
* i.e. java DNSLookUpAll www.wiley.com
*/

public class DNSLookUpAll {
   public static void main(String[] args) {
      InetAddress hostAddress;
      try {
         InetAddress[] inetAddressList = InetAddress.getAllByName(args[0]);
         for (int i=0; i<inetAddressList.length; i++) {
         System.out.println(inetAddressList[i].getHostAddress());
         }
      }
      catch (UnknownHostException uhe) {
         System.err.println("Unknown host: " + args[0]);
      }
   }
}
```

Figure 19.101 A possible Java solution.

- `74.125.70.103`
- `74.125.70.147`

19.12 The original HTTP protocol used TCP/IP as the underlying network protocol. For each page, graphic, or applet, a separate TCP session was constructed, used, and torn down. Because of the overhead of building and destroying TCP/IP connections, performance problems resulted from this implementation method. Would using UDP rather than TCP be a good alternative? What other changes could you make to improve HTTP performance?

Answer:

Despite the connectionless nature of UDP, it is not a serious alternative to TCP for use with the HTTP protocol. The problem with UDP is that it is unreliable, and items delivered via the web must be delivered reliably. (This is easy to illustrate—a single packet missing from an image downloaded from the web makes the image unreadable.)

One possibility is to modify how TCP connections are used. Rather than setting up—and breaking down—a TCP connection for every web resource, allow ***persistent*** connections whereby a single connection stays open and is used to deliver multiple web resources.

19.13 For each of the following workloads, identify whether a cluster-based or a client–server DFS model would handle the workload best. Explain your answers.

- Hosting student files in a university lab.
- Processing data sent by the Hubble telescope.
- Sharing data with multiple devices from a home server.

Answer:

Users in a university lab would likely appreciate having a remote file system mounted transparently into their home directories. In addition, the workload would likely involve random writes on small- and medium-sized files. This points to a client–server model DFS like NFS.

Hubble telescope files would likely be large and not modified once written. In addition, it would be nice to process these large files in parallel. Thus, a cluster-based DFS like HDFS would be appropriate.

Data stored in a home environment would not likely be spread and replicated among many data servers. Instead, a single server with a RAID or offline backups would be sufficient. A client–server model DFS like NFS or CIFS would probably be most appropriate.

19.14 Under what circumstances would a client prefer a location-transparent DFS? Under what circumstances would she prefer a location-independent DFS? Discuss the reasons for these preferences.

Answer:

A location-transparent DFS is adequate in systems in which files are not replicated. A location-independent DFS is necessary when any replication is done.

19.15 Compare and contrast the techniques of caching disk blocks locally, on a client system, and remotely, on a server.

Answer:

Caching locally can reduce network traffic substantially, as the local cache may be able to handle a significant number what would be remote accesses. This can reduce the amount of network traffic and lessen the load on the server. However, to maintain consistency, local updates to disk blocks must be updated on the remote server using either a write-through or delayed-write policy. A strategy must also be provided that allows the client to determine if its cached data are stale and need to be updated.

Caching locally is obviously more complicated than having a client request all data from the server. But if access patterns indicate heavy writes to the data, the mechanisms for handling inconsistent data may increase network traffic and server load if the data are cached on the server.