Questions (5 points)

1. (1 point)
   (a) Define what is meant by a scalable system.
   (b) Discuss why message delays in the Internet grow only very slowly with the number of networks in the Internet. How are the networks in the Internet configured to achieve this type of scalability?
   (c) State three techniques commonly used to achieve scalability.

2. (1 point)
   (a) Distinguish between buffering and caching.
   (b) 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 effect on the properties of validity and integrity. Can service B be described as a reliable communication service?

3. (1 point)
   (a) If a system does not provide a blocking receive, can you simulate it by a non-blocking receive? If so, show a pseudo code. If not, explain why. Discuss your answer for the case of a single thread and for the case where you can use multiple threads.
   (b) If a system does not provide a non-blocking send, can you simulate it by a blocking send? If so, show the pseudo code. If not, explain why. Discuss your answer for the case of a single thread and for the case where you can use multiple threads.

4. (1 point) Use the program in Figure 4.3 to make a client program that repeatedly reads a line of input from the user, sends it to the server in a UDP datagram message, then receives a message from the server. The client sets a timeout on its socket so that it can inform the user when the server does not reply. Test this client program with the server in Figure 4.4.
5. (1 point) To get you started in Java programming, this question asks you to understand a simple Java program, run it, and explain how it produces the output.

Compile and run the Time-of-Day Server (ToDServer.java) with the class Connection (Connection.java) downloadable from the course web page. Type

```
telnet 127.0.0.1 5555
```

and observe the output.

(a) Explain the sequence of events in the server and client that led to the observed output.
(b) Next, use `telnet` from a remote machine. In this case, you need to replace the first argument (127.0.0.1) by something else. Give two possible replacements.