University of Rochester **Mobile Computing (CSC290e)**

Midterm Exam

Amy Murphy
5 November 2001

Read before beginning: Please write clearly. Illegible answers cannot be graded. Be sure to identify all of your answers in your blue book (e.g., 1a, 2b, etc). All questions are worth 10 points, numbers in parenthesis indicate relative point values. Answer all questions.

While I have tried to make the questions as unambiguous as possible, if you need to make any assumptions in order to continue, state these as clearly as possible as part of your answer. In the name of fairness, I would like to avoid answering questions during the exam.

1. **Distributed and Mobile Computing**: your responses should be around 2 lines of text:

   (a) (4) Name and briefly explain two advantages to distributed computing.

   increase fault tolerance (redundancy), exploit physical distribution of machines, increase processing capabilities by increasing the number of hosts assigned to a task.

   (b) (3) Name and briefly explain one disadvantage in designing and building distributed computing systems.

   Once data is distributed, it has to be managed. If you replicate, you have to keep them up to data and consistent. If part of the system crashes, you need to be able to keep going. In general, there are more complexities to deal with.

   (c) (3) Mobile computing applications are also distributed applications. Name one more complication that mobility adds that is not typically present in distributed computing.

   resources are not always available. bandwidth is usually limited. battery power limits the lifetime of the system.

2. **Networking.**

   (a) What is the difference between a point to point TCP socket and a point to point UDP socket?

   TCP is a reliable protocol where data is delivered in order, and without gaps. UDP does not make either of these guarantees.

   (b) Why does video use UDP?

   There is little point in trying to recover a lost packet (as would be done with TCP). By the time that it is recovered, the video would have passed the time when that data was useful.

   (c) Why does FTP use TCP?

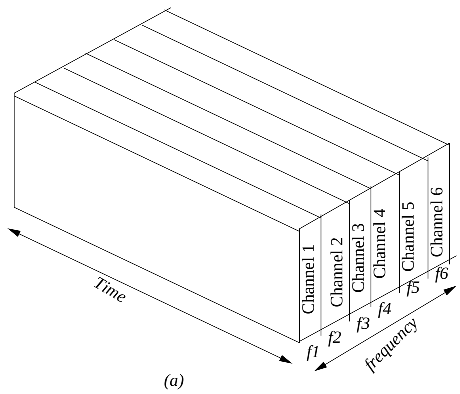
   files are no good if they are not complete. TCP gives this guarantee.

   (d) Give one reason why multicast on the mbone is closer to UDP than to TCP.

   ack implosion. If you used the straightforward TCP acknowledgment scheme for data, the sender would receive an ack from each receiver. In general, multicast is used for streaming media (e.g., video, audio) where best effort transmission is good enough.

3. **Medium Multiplexing.**

   Figure 1 shows two different multiplexing schemes: FDM and TDM. For each scheme, identify which figure illustrates it, spell out the acronym, and briefly describe each scheme. *Briefly* means give the intuition about how multiple stations share the medium over time, describe a station's communication behavior through time.



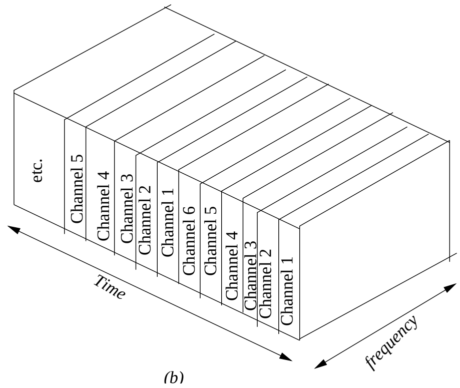


Figure 1: For question 3.

Figure (a): FDM = frequency division multiplexing: each station is assigned a frequency which it is always allowed to transmit on.

Figure (b): TDM = Time division multiplexing: each station is assigned a time slice. During that time slice, they can exploit all of the available frequencies. This is tricky because we need tight time synchronization among all senders and receivers to avoid interference.

4. **Medium Access Control.**

   Assume all stations are within range of all other stations (i.e., everyone can hear everyone else). One station wants to transmit and senses the carrier idle. Why can a collision still occur after the start of transmission? In class, we mentioned the hidden terminal problem as one reason not to use ethernet-like mechanisms such as CSMA-CD. When all stations can hear all others, does this the hidden terminal still exist and does this reasoning still hold? Briefly explain.

   A collision can occur if, despite waiting, two hosts decide to start transmitting at exactly the same time. This is unlikely, which is why it works most of the time, but it can still happen. In this scenario, because the hidden terminal problem does not exist, the wireless medium has the basically the same characteristics as a shared ethernet wired, and therefore a simple scheme such as CSMA-CD should work as well as it does in an ethernet.

5. **Cellular Communication.**

   What are the main problems when transmitting data using wireless systems that were made for voice transmission?

Voice is continuous. Data is bursty. While the same overall data rate may be needed for both, this fundamental difference means that the medium will not be used as efficiently. Also, data rates are lower and communication may not be as reliable through a wire.

6. **Ad hoc Networks.**

   In ad hoc networks, there is no infrastructure to support unicast routing. Similarly, there is no infrastructure to support a controlled broadcast (sending a messages to all connected hosts). Explain the problem of broadcast storms, and give one simple solution to help alleviate them. Remember, we talked about this issue when discussing the spreading of route requests in DSR.

   A broadcast storm occurs when each terminal, upon hearing a broadcast message, blindly repeats this message. Among other problems, it can cause messages to propagate endlessly, it can overwhelm the medium when multiple stations close to one another try to simultaneously retransmit the same broadcast packet, etc. One way to moderate these is to give each station a memory. It transmits each packet once, remembers it, and does not send it again. Other, fancier mechanisms, allow a station to choose not to even transmit the packet once if it has heard it multiple times with strong signal strength. This works because in this case, the rebroadcast of the packet by the station is not likely to cover any stations which have not already seen it from some other station.

7. **Lime.**

   The code on the last page of this exam shows two LIME agents A0 and A1. All exception handling has been removed for readability, so you should assume that it will compile and there are no syntax errors. Assume you have two shells that you are typing in. What is the output of the programs if you start them with the following command lines (specify which shell has the output):

   - (shell 1) java lime.util.Launcher -load A0
     (wait a couple of seconds)
     (shell 2) java lime.util.Launcher -quit -load A1

     shell 1:
     A0: wrote the tuple: $\langle 42 \rangle$
     A0: read the tuple: null
     A1: readP the tuple: null
     A1: read the tuple: $\langle 42 \rangle$
     shell 1:
     empty

   - (shell 1) java lime.util.Launcher -load A0
     (wait a couple of seconds)
     (shell 2) java lime.util.Launcher -port 1900 -load A1

     shell 1:
     A0: wrote the tuple: $\langle 42 \rangle$
     A0: read the tuple: null
     shell 2:
     A1: readP the tuple: null