CS455 - Lab 02

Thilina Buddhika

February 2, 2015
Agenda

- Course Logistics
- Quiz 1 review
- ASG 01 - Makefile and setup script
- ASG 01 - Threads
- ASG 01 - Dealing with concurrency problems
- Wrap Up
Course Logistics

- **Quiz 1**
  - Collect your graded quizzes.
  - Check the short answers.
  - Check the grade on RamCT.

- **Term Paper**
  - Groups and topic assignments are posted on Wiki.
  - Start working on it in parallel to programming assignments.

- Communicate your problems effectively. Use the Wiki.

- Follow the milestone plan.
1 Manchester encoding fixes the following problem(s) in NRZ:
   a. Baseline wander
   b. Clock recovery
   c. Both (a) and (b)
   d. None of the above

2 The 4B/5B encoding scheme ensures that the translated codes (4-bit to 5-bit encodings) do not result in more than 1 leading 0 and no more than 2 trailing 0s. This is because:
   a. It allows the transmissions to be more energy efficient
   b. The 5-bit codes are transmitted using NRZI that performs transitions for 1s
   c. The receiver averages for signal strengths would be much closer to the media signal strength between 1 and 0.
   d. None of the above
3. What is the efficiency of the 4B/5B encoding scheme? **80%**

4. Circuit switched networks require circuits and their accompanying state to be established before data can be transferred. [True/False]

5. When multiplexing data onto a physical link, setting aside time quanta is not feasible because:
   a. Number of flows must be known ahead of time
   b. The link may be idle if a particular flow is not active
   c. Typical network usage patterns are bursty
   d. All of the above
The Delay x Bandwidth product tells us how many bits fit in a network pipe. What is the maximum number of pipes that a sender can fill before it receives an acknowledgement from the receiver?

2

The latency observed in a channel cannot be attributed to:

a. Bandwidth
b. Propagation Delay
c. Queuing delay
d. Time required to transmit a unit of data
8. In statistical multiplexing, packets from a particular sender are placed in a predetermined slot in the routing schedule. **[True/False]**

9. In computer networks, end hosts are kept simple while the complex communication semantics are pushed to the center of the network. **[True/False]**

10. IP attempts to recover from reassembly failures at the receiver side by sending a retransmission request to the source. **[True/False]**
Available on the wiki
Not ideal for Java programs like Maven or Ant.
Doesn’t need extra tools to be installed and configured.
Modify according to the structure of your program

```
1  all: compile
2  @echo -e 'INFO] Done!'
3  clean:
4  @echo -e 'INFO] Cleaning Up..
5  @rm -rf cs455/**/*.class cs455/**/**/*.class
6
7  compile:
8  @echo -e 'INFO] Compiling the Source..
9  @javac -d . cs455/**/*.java cs455/**/**/**.java
```
Setup script

- Point test_home to the root directory of your project
- Create a file called machine_list at the same level with a list of machines.
- One messaging node per each machine
- Registry node should be started separately and its hostname and port should be included in the script.

```
1  test_home=/path/to/the/compiled/source
2
3  for i in 'cat machine_list'
4    do
5      echo 'logging into '${i}
6      gnome-terminal -x bash -c "ssh -t ${i} 'cd $test_home; java cs455.overlay.node.MessagingNode
7        registry_host registry_port;bash;''" &
```

CS455 - Lab 02
An example file with 4 machines
Each machine name should appear in a new line.
Use only CS 120 lab machines.
CS Department Machine list:
http://www.cs.colostate.edu/~info/machines

1. annapolis
2. atlanta
3. augusta
4. boston
Threads

- Implement Runnable interface
  - Most commonly used and the recommended
  - The definition of the job is decoupled from the choice of the concurrency
  - Still allows extending some other class

- Extend Thread class
  - Easier to implement: one less class to worry about.
Things that need to happen in parallel in a node:
  - Accepting new connections (serverSocket.accept() is a blocking call.)
  - Listening to keyboard input
  - Listening for inbound data (on multiple connections)

TCPReceiverThread for each connection should run on a separate thread.

Listening to keyboard input or accepting new connections can be handled using the main thread.
TCPConnection class

- Represents the connection between two nodes.
- Encapsulates the corresponding TCPReceiverThread and TCPSender.
- Store and reuse the connection.
  - ConnectionCache
  - Hashtable is a candidate data structure
  - key - a unique identifier for the destination node
  - value - the connection
Accepting an incoming socket connection

```java
while(true){
    Socket socket = serverSocket.accept();
    TCPConnection conn = new TCPConnection(node, socket);
    // cache the connection
    ....
}
```
public TCPConnection(Node node, Socket socket){
    this.receiver = new TCPReceiver(socket, node);
    this.sender = new TCPSender(socket);
    ....
}

Dealing with concurrent threads

- **Synchronized keyword.**
  - If specified in a method signature, it will ensure that only a single thread will execute that method (Or any synchronized method in that class) at any given time
  - Serializes the invocation of the method

- More sophisticated and efficient schemes for dealing with concurrency
  - Next section in the class
  - Use of Synchronized methods is sufficient for this assignment
ASG 01 - Possible concurrency issues

- Caching connections
- Concurrent writes to the same socket by multiple threads, especially when relaying messages.
- Concurrent updates to the counters used to keep track of statistics
- Encapsulate the logic in a synchronized method
Wrap Up

Questions?