CS3210 Computer Networks Lab
Even Sem. 2015, Prof. Krishna Sivalingam
Lab 7: Slotted Aloha Protocol
Individual or Group of 2 Students
Due date: April 20, 2015, 6PM, On Moodle

Announced on April 20, 2015

1 Description

The objective of this project is to implement the Slotted Aloha protocol, as discussed in class.

The project is done individually or by a Group of 2 Students. Those who not attend Lab session on April 20 will not be allowed to submit the assignment.

Input: The command-line input parameters are:

1. "-N integer": number of users sharing the channel
2. "-b double": specifies initial value of probability of retransmission from backoff state for all nodes ($b(i)$; default: 0.3).
3. "-p double": specifies PACKET_GEN_RATE (p): probability of packet generation per unit time per node
4. "-M integer": MAX_PACKETS

Protocol Operation:

Time is slotted. All packets are of fixed length, equal to one slot. This can be simulated using a while loop and a variable called *SimTime*. The variable is initialized to 0, and incremented by one in each loop instance.

Initially, all nodes are non-backlogged, i.e. do not have a packet to transmit. Each node's buffer can hold at most one packet (including the packet for which transmission is being attempted). The packet is removed from the buffer after it has been successfully transmitted.

The main steps of the program are:

1. In each time slot, every non-backlogged node will generate a packet with probability p.

   Note: This packet generation step precedes the packet transmission step.

2. A new packet will be transmitted in the slot it was generated. Every backlogged node will attempt retransmission with probability $b(i)$.

3. A packet transmission is successful if only one node attempts transmission during a slot.

4. If node i's transmission attempt is successful, then $b(i) = min(0.75, b(i) * 1.15)$. Remove the packet from the buffer; the node becomes non-backlogged at the end of the slot.

5. If node i's transmission attempt is not successful, then $b(i) = max(0.1, b(i)/1.15)$. The node enters backoff state and is considered backlogged.

6. Calculate: (i) mean delay per packet and (ii) average utilization per slot (throughput), i.e. number packets sent per slot.

The program terminates after MAX_PACKETS (a command-line parameter) have been successfully transmitted combined across all nodes (OR) if the maximum retransmission attempt for any packet exceeds 20.

Output: On termination, the program will print the following information, in a single line to the screen:

1. Number of nodes, b and p

2. Utilization: Average number of packets successfully sent per slot

3. Average Packet Delay: Time between packet generation and successful transmission.

2 What to Submit

The platform for this project will be Linux and C/C++/Java. Create a tar-gz file with name: Lab7-RollNo1-RollNo2.tgz (e.g., Lab7-CS12B110-CS11B099.tgz). or Lab7-RollNo1.tgz (e.g., Lab7-CS12B110.tgz).

The directory should contain the following files:

- Source File(s)
- Makefile and Script File
  Typing command 'make' or your script program, at the UNIX command prompt, should generate all the required executables.
- A Script file obtained by running UNIX command *script* which will record the way you have finally tested your program.
- a README file containing instructions to compile, run and test your program.

Optional: Write a shell script (bash, perl, etc.) that will generate throughput values for varying N, p and b values. Verify if the maximum throughput matches the theoretical estimate of 0.36 for $p = 1$.

3 Help

1. Ask questions EARLY and start your work NOW (really, no choice). Take advantage of the help of the TAs and the instructor.
2. Use random() or similar (divide random() by RAND_MAX if needed). Make sure that you seed the random number generator before used.
3. Submissions PAST the extended deadline SHOULD NOT be mailed to the TAs. Only submissions approved by the instructor or uploaded to Moodle within the deadline will be graded.
4. Demonstration of code execution to the TAs MUST be done using the student's code uploaded on Moodle.
5. NO sharing of code between students, submission of downloaded code (from the Internet, Campus LAN, or anywhere else) is allowed. The first instance of code copying will result in ZERO marks for the Lab component of the Course Grade. The second instance of code copying will result in a 'U' Course Grade. Students may also be reported to the Campus Disciplinary Committee, which can impose additional penalties.
6. Please protect your Moodle account password. Do not share it with ANYONE, including your team member. Do not share your academic disk drive space on the Campus LAN.
7. There is no need to use threads; no need to use per-node buffers. Each node can be in three states: IDLE, TRANSMITTING, BACKLOGGED.

4 Grading

- Aloha Code working correctly: 50 points
- Shell script and output for varying parameter values: 10 points (Bonus); Bonus will be added only if you obtain at least 45/50 points for Aloha implementation