HKUSTx: ELEC1200.3x A System View of Communications: From Signals to Packets (Part 3)

Bookmarks

- Pre-course Materials
- Topic 1: Course Overview
- **Topic 2: The Link Layer**

2.1 Link Layer
Week 1 Quiz due Jan 25, 2016 at 15:30 UTC

2.2 Multiple Access Protocols
Week 1 Quiz due Jan 25, 2016 at 15:30 UTC

2.3 Aloha Protocol
Week 1 Quiz due Jan 25, 2016 at 15:30 UTC

2.4 Efficiency of Slotted Aloha
Week 1 Quiz due Jan 25, 2016 at 15:30 UTC

Topic 2: The Link Layer > 2.5 Lab 1: Link Layer > LAB 1 - TASK 1

Bookmark

LAB 1 - TASK 1 (EXTERNAL RESOURCE) (1.0 points possible)

2.5 Lab 1: Link Layer
Lab due Jan 25, 2016 at
15:30 UTC

▸ MATLAB download
and tutorials

LAB 1 - TASK 1

This task provides an overview of the slotted ALOHA system. Your task is to compute the efficiency
system, which is defined as the ratio between the number of successful transmissions and the num
slots.

INSTRUCTIONS

In our simulatino of the slotted ALOHA protocol we study here, we assume a population of **n_users**
nodes (nodes with data to send) access a shared channel by transmitting a frame in each slot with
probability 0 < **p** < 1. There is a nonzero probability that that only one of the nodes will use the cha
avoiding collision and enabling a successful transmission. Since access to the channel is controlled
random transmission by the nodes, we call this a random access scheme. This is a slight simplificat
original ALOHA protocol, since we do not consider retransmission, but rather that frames are lost if
collisions, and we do not explicity detect collisions.

The MATLAB code in the below window simulates a slotted ALOHA system with **n_users**=4 nodes.
identified by an index **id** ranging from 1 to **n_users**, and transmits a data frame in each slot with pr
p=0.1. The code simulates the transmission for **n_slots**=1000 time slots. At the beginning of each
code initializes the variable **slot**, which is the state of the shared channel in the current slot, to zero
code, the length of each slot/frame is 16 bits (see the next task for details). Then, each user check
frame is empty. Its frame will be empty if the user transmited its frame in the previous time slot. If it
gets a new datagram (function **getNewDatagram**) and then creates and stores the frame (function
createFrame and **updateFrame**). Then, the user determines whether to transmit the frame or no
generating a Bernoulli random variable with parameter **p**=0.1. If the user transmits its frame, the co
the previously stored frame (function **getFrame**), update the variable **slot**, and then resets (empti
(function **resetFrame**) so that it will get a new frame in the next iteration. If the user does not trans
frame is retained for future transmission. In this simulation, we assume a binary channel where the
channel, **slot**, is obtained by taking the logical **or** operation among all frames transmitted. In a prac
this would be implement if each node was connected to a common wire with a pull down resistor to
state at zero if no nodes are transmitting, and each node is connected to the wire with a pull up tran

The correctness of the received frame is checked using the checksum (function **checkReceivedF** will be explained in details in Task 3. In this code, errors in the transmission are only due to multiple transmissions in the same time slot (collision), not to noise. If the checksum is correct, we incremen of successful transmissions (**n_succ**). If the checksum is incorrect, we increment the number of col (**n_coll**). If the slot is empty (zero), we increment the counter of empty frames (**n_empty**).

At the end of the simulation, we display the total number of time slots, empty slots, collisions and su transmitted frames. We also plot in Fig. 1 the transmission traffic for the first 50 slots to illustrate ho protocol works in controling the access of multiple users to the same shared channel. For each use shows a circle at the time it transmits a frame. When we observe more than one circle in a time slot 4, 14, ...), there is a collision. When there are no circles (e.g., slots 1, 5, ...) there is an empty fram

Your task is to compute the efficiency of the system and store the result inside the variable **efficie** revise the code between the lines

```
% % % % Revise the following code   % % % %
```

and

```
% % % % Do not change the code below % % % %
```

Do not change other parts of the code.

Your Solution

Reset MATLAB Documentation (https://www.mathworks.

```
% % % % Revise the following code   % % % %

efficiency = n_succ / n_slots;
display(['Efficiency: (n_succ / n_slots): ' num2str(efficiency) ]);
% % % % Do not change the code below % % % %

display(['Total number of slots: ' num2str(n_slots) ]);
display(['Empty slots: ' num2str(n_empty) ]);
```

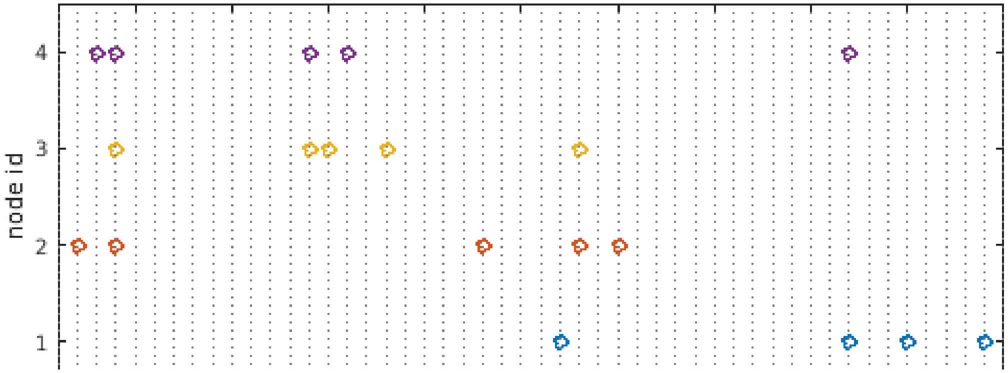
Assessment Tests: Passed

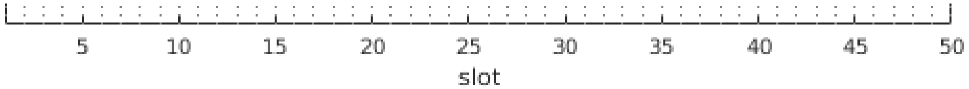
✓ Is problem setup unmodified?

✓ Is the efficiency correct?

Output

```
Efficiency: (n_succ / n_slots): 0.316
Total number of slots: 1000
Empty slots: 636
Collisions: 48
Frame transmitted successfully: 316
```




5
10
15
20
25
30
35
40
45
50
slot

POWERED BY OPENedX