Lab 3: Hands-On MAC Layer

Introduction

In this lab, you will continue with the groups you formed in the last lab to implement a Media Access Control (MAC) network layer on top of the PHY layer of the previous lab, again using audio systems. This MAC layer is needed to ensure that access to the medium is regulated to avoid corruption of data in the presence of multiple parties attempting to transmit (possibly simultaneously).

Plagiarism Warning

There will be **strict** plagiarism checks for this assignment. You are not allowed to copy code verbatim from other groups. An elaborate MOSS check would be used for this plagiarism check.
If your code is taken from a public website or created using generative AI, then you must clearly mention this in your code (as a comment). Moreover, such usage of web/AI must be limited to at most 50% of your code.

Experimental Setup

The system you will build should consist of three nodes attempting to transmit data simultaneously. The device(s) comprising each of the nodes is/are your choice. You can make use of any reasonable MAC protocol (such as Token Ring, CSMA, etc.).

Input to the system will be in the form of messages made of an arbitrary number of bits, which you should reliably be able to transfer to the other end. The number of bits in a message would be at most **15**.
During experimental evaluation (i.e. the demo of your system), we will provide input messages to each of the 3 nodes along with the message destinations. The destination could either be a single node or the message could be a broadcast message which means that all the nodes will have to receive the message. You will be given 2 chances at 2 different points in time. Each chance will comprise the act of transferring 4 messages (at most 2 messages per node) in total. There is no bifurcation into short and long messages as such, unlike the previous lab. All the messages carry equal marks. Finally, we will take the best (accuracy) over the 2 demo chances to calculate your score. The whole demo is expected to take under 5 minutes.
Note that there would be **no errors** introduced in the messages in this lab, hence your codeword can just be the raw data. This time, you are allowed to make any other non-codeword transmissions that you might feel are relevant to your design. There are essentially no restrictions on what transmissions you want to make.
Three students out of four (operators) would be randomly chosen to operate at each of the three nodes. The nodes would be placed in close proximity. Wifi, Mobile Broadband (4G), etc. must be strictly turned off on both the sender and the receiver laptops. Bluetooth is allowed but only as long as it is used for connecting Bluetooth speakers/microphones for aiding in transmission. If at any point it is found that Bluetooth is being used for

direct communication between the nodes, you would be awarded 0 for the assignment. Sometimes the audio reception can be poor (completely possible, does happen at times). In such a case, we would allow you to redo the experiment from the *beginning* **once**.

Program Description

Everything in this lab needs to be **automated**. There should be programs running on all the nodes.
You are required to decide a numbering of the nodes (each number has to be either 1, 2 or 3) and tell us the numbers at the start of the demo. No two nodes should have the same numbers. These numbers would act as the "MAC addresses".

Input format (for each node):

Line 1: `<message> <destination>`
Line 2: `<message> <destination>`

At the start of the demo, we would provide the messages for each node along with the destinations in the format `<message> <destination>`, one space-separated pair on each line, where `<message>` is the bit-string to transmit and `<destination>` would be one of the five numbers: -1, 0, 1, 2, 3. It will be ensured that the destination is not the same as the sender.

- -1 as the destination means that the message is not destined for anyone. Such inputs can be ignored by the program. (This would be applicable for those nodes that only have to transmit one message)
- 0 as the destination means that the message is a broadcast message and must be received by all the nodes.
- Any other number as the destination means that the message is destined to the node having that number as the MAC address.

Note that we would be giving out all the 4 messages in the start itself, but this doesn't mean that all 4 messages are to be attempted to transfer simultaneously just after the start of the experiment. You are also supposed to provide a provision for the TAs to trigger a message-sending event on the nodes which the TA can use at any point during the demo (e.g.: This trigger could be in the form of hitting he enter key on the keyboard). This means that as soon as such a trigger is sent, the message will be attempted to transfer, but the transmission must be regulated by the MAC layer i.e.. it must ensure that when such a trigger is sent, either the MAC layer completely avoids collisions or makes sure that if a transmission is immediately done and results in a collision, then it can recover from collisions. Note that once a message is sent to the MAC layer, the next trigger must enqueue the next message at the MAC layer and not the same message i.e. If messages a, b are given to one of the nodes at the start of the demo, the first trigger by the TA should result in message a enqueueing at the MAC layer and the next trigger must enqueue message b at the MAC layer.
On the screens, we require you to print the messages sent and messages received by the nodes along with their timestamps in the following format:

- For messages that get sent : `[SENT]: <message> <destination> <timestamp>`

- For messages that are received : `[RECVD]: <message> <sender> <timestamp>`

Note that `<message>` must just be the data bits and not contain any auxiliary bits you might have used in your transmissions. `<sender>` must be the number of the node ("MAC address") that made the transmission. Please make sure that the clocks on all the nodes are synchronized using NTP before the experiment starts so as for the timestamps to make sense.

Program Requirements

Communication should occur purely via **audio mechanisms only**. Absolutely, no communication by any other methods, including verbal speech, would be permitted during the duration of the experiment. Please try to keep the physical layer as close as possible to your physical layer implementation from Lab-2 in terms of the protocol used for transmission (your encodings might be different as there are no error bits this time).

Design Document

You are again required to submit a design document outlining the implementation idea i.e. information in headers, MAC protocol used, auxiliary transmissions made, etc. Concise does not mean incomplete. Append it to the design document of lab 2, under a separate heading for the MAC layer. Please make sure that your description would allow an intelligent independent reader to be able to replicate your design.

Submission Instructions

By **Friday, 6th September, 23:59 PM**, you should submit a design document pdf named

`<rollno_1>_<rollno_2>_<rollno_3>_<rollno_4>_dd_CS378_lab3.pdf`

By **Tuesday, 10th September, 23:59 PM**, you are supposed to submit a tar file containing all the code, documentation you've written for the code along with running instructions for the nodes, and the final design document (if you've made any significant changes to the first design document submitted a week earlier). The file must be named

`<rollno_1>_<rollno_2>_<rollno_3>_<rollno_4>_CS378_lab3.tgz`

Note that all the letters in your roll number, if applicable, must be in lower case.

There are going to be **individual** vivas after the demo, so make sure that the entire team knows all aspects of the implementation.

Grading

The score distribution for this lab is as follows:

- Design Document - 25%
- Successful Demo - 40%

- Well-Documented Code - 25%
- Viva - 10%

Note that unlike the last lab, we wouldn't be measuring the time taken for the demo because it can be arbitrarily affected due to the inputs given by the TAs. A more sophisticated MAC protocol implementation would fetch more marks than a less sophisticated MAC protocol, **given** that the demo is **successful**. In other words, perfect accuracy is more important than sophistication. E.g.: A perfectly working CSMA MAC layer would fetch more marks than a perfectly working Token Ring MAC layer.
The viva would be taken individually.
Similar to the last lab, the viva would consist of two components. The first component would check that you actually know how your model works and can explain its working satisfactorily. Success in this component of this viva is crucial to ensure that you are awarded marks for this lab. In particular, if you are not able to satisfactorily explain the model, you shall receive a 0 in this lab.
The second component would comprise of questions about the MAC layer to check your understanding of it. This is the component that's worth 10% of the grades.

Hacking Ideas

We very much appreciate hacks and loopholes in the problem statement and encourage you to post them on Piazza for everyone's amusement. However, do note that such hacks (irrespective of whether they have been mentioned on Piazza or not) which go against the pedagogical spirit of the assignment won't attract marks if applied in practice. If in doubt, please clarify on Piazza (you could choose to send a private message too).