National University of Sciences & Technology
School of Electrical Engineering and Computer Science
Department of Computing
EE353: Computer Networks, BSCS5 Fall 2017

Project Specifications (V1)	
CLO 4: Design and implement solutions to contemporary networking issues (through hands on programming)	
Maximum Marks: 10	Instructor: Dr Nadeem Ahmed / Dr Arsalan Ahmed
Date: 20th Nov 2017	Due Date: mid night 7th Jan 2018

1.0 General:

This is a group assignment. Maximum size of group is restricted to 2 members.

Please avoid plagiarism; any such case would result in award of zero marks both to the "sharer" and the "acquirer". Maximum score is 20 points that would be scaled back to 10 marks.

2.0 Learning Objectives

This is a major hands on assignment. By completing this project, the students will develop skills for

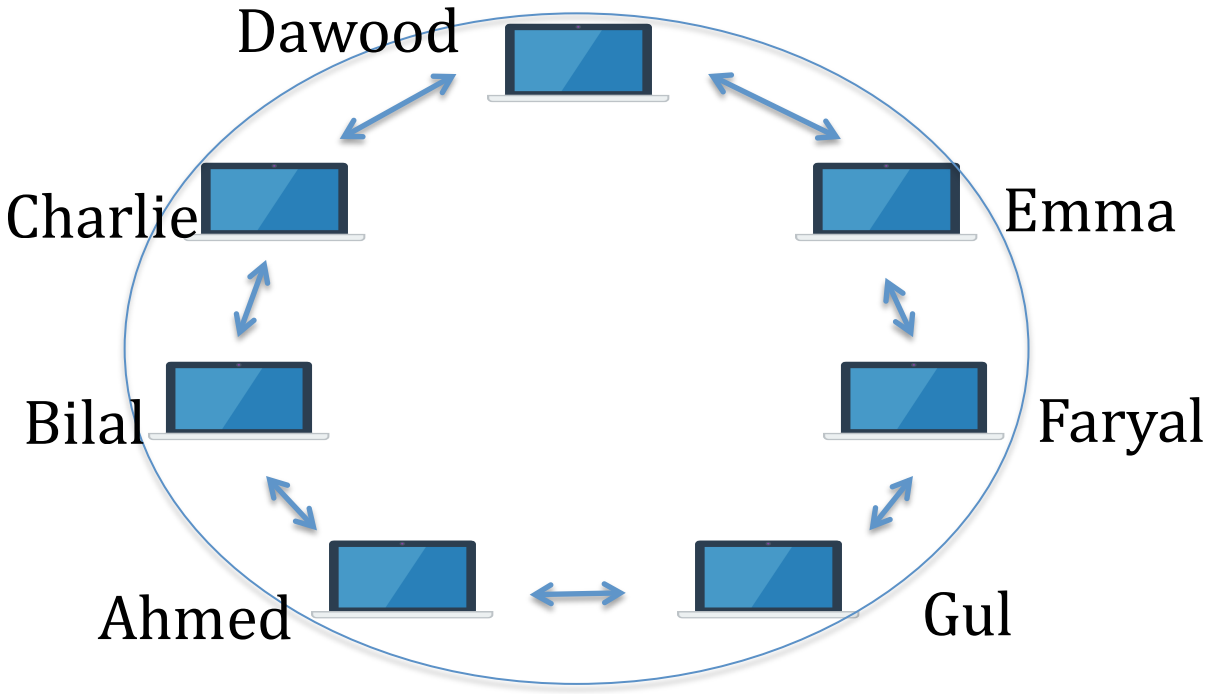
- Eliciting detailed implementation requirements from a high level description (as often provided in Internet Drafts or RFCs) of a routing protocol.
- Developing and testing routing software
- Handling routing dynamics
- Proposing extensions or improvements to existing routing protocols.
- Socket programming in Python
- Multi-threading in Python

3.0 Specifications:

For this project, you are asked to implement a number of components (as opposite to the entire) of the Ad-hoc On demand Distance Vector (AODV) routing protocol. The main components of AODV are path discovery using flooding, distance vector routing, route table management and path repair. You will be able to learn how these components are put together to get a working protocol.

Seven SEECS students - Ahmed, Bilal, Charlie, Dawood, Emma, Fatima and Gul are sitting around a table working on a project together. They each have their laptop with them and their sitting positions are given in figure 1. Each laptop is equipped with wireless connectivity and is only within the radio range of its two neighboring laptops. For example, Ahmed's laptop can hear and talk

to both Bilal's and Gul's directly, but not with the other laptops. These laptops form an ad-hoc network running the AODV routing protocol. You can assume that the routing tables at all the laptops are empty in the beginning.



For this assignment, you are asked to implement the AODV protocol and demonstrate that it is working correctly. Your implementation part should include at-least:

- Basic protocol functionality
- Route discovery and path setup using RREQ and RREP messages
- Route table management (including forward and reverse routes)
- Fault tolerance and repair
- Maintenance of active route and precursor list
- Path repair using RERR messages
- Performance improvement
- Implementation of your proposed solution for improving performance of the AODV protocol (detailed in Task 2 and Event 5).

The assignment tasks are listed below. Note that you do not have to implement all the header fields of the AODV protocol listed in the RFC 3561. A protocol field is required only if it is necessary. You are also required to document the packet format that you have used in file Project.doc. See the submission instructions.

Task 1: Assign addresses to the laptops wireless interfaces.

- Document your address assignment in a file named project.doc and submit it together with your written programs.
- IPv4 classless addressing is assumed.

- Assume any sufficient block of IPv4 addresses and assign IP addresses out of this block only.
- You should give your address allocation using the notation: IP address/ prefix length.

Event 1: Communication between Gul (initiator) and Dawood (responder).

- You need to demonstrate the process of route discovery by showing the messages exchanged, the actions applied to these messages, the routing table and the states maintained by each hosts.
- After the path setup is complete, initiator Gul sends data packets to the responder. Show how the data packets are routed from Gul's laptop to Dawood's.
- Dawood sends replies (data packets) to Gul. Show how the data packets get routed.

Event 2: Communication between Ahmed (initiator) and Dawood (responder).

- You need to demonstrate the process of route discovery by showing the messages exchanged, the actions applied to these messages, the routing table and the states maintained by each host.
- After the path setup is complete, initiator Ahmed sends data packets to the responder. Show how the data packets are routed from Ahmed's laptop to Dawood's.
- Dawood sends replies (data packets) to Ahmed. Show how the data packets get routed.

Event 3: Emma leaves the Group

- Demonstrate how the connectivity loss is handled by AODV. Show the appropriate actions taken by each host.
- Show the routing tables of Dawood, Fatima and Gul's laptop.

Event 4: Communication between Gul (initiator) and Dawood (responder).

- You need to demonstrate the process of route discovery by showing the messages exchanged, the actions applied to these messages, the routing table and the states maintained by each hosts.
- After the path setup is complete, initiator Gul sends data packets to the responder. Show how the data packets are routed from Gul's laptop to Dawood's.
- Dawood sends replies (data packets) to Gul. Show how the data packets get routed.

Task 2: Propose changes to AODV protocol.

- The loss of connectivity is handled by RERR messages and local repair in AODV. For the given network topology, local repair is not very useful. This results in certain delay in alternate path discovery. Propose changes in the basic AODV routing protocol such that alternate route discovery delays, for the given network topology, are minimized with low additional complexity. You can suggest changes to existing AODV messages or design new message types.

- You should detail your proposed changes in the file *project.doc* and include how the proposed changes will be handled by AODV nodes and how your proposed solution is better than the existing path repair strategy of AODV.

Event 5: Implement the proposed changes in the AODV protocol

- You need to demonstrate that the modified AODV handles loss of connectivity efficiently than the original AODV for the given network topology.
- Show all the relevant control messages exchanged.

3.1 Miscellaneous specifications

- For route discovery process, you need to show that RREQ, RREP (including gratuitous, if needed) messages are correctly created, correctly handled by the hosts and correctly propagated in the network. In addition you must also show the creation of appropriate entries in the routing table. In case the hosts need to maintain intermediate states to handle route discovery, you will need to show theses states are maintained too.
- You do not need to implement the multicast functionality of AODV.
- Assume that the G flag is ALWAYS set in RREQ messages (D flag is never used).
- Flag A is never used in RREP messages. (All links are bi-directional and no RREP-ACK messages are used).
- You need to implement the HELLO messages.
- LifeTime of routing table entries can be taken as infinite.
- TTL value is taken as 7.
- Assume that each device knows the IP address of every other device in the ad-hoc network (can hard code this information).
- You are free to configure any timer required in a protocol for your implementation. Record this time interval in your design.

3.2 Program Design, Packet format, and Data structures

Students are required to use Python as the programming language to do this project. The good news is that the programming will be done entirely in the user (application) space; no kernel programming is required. There is no ``stub'' program supplied. You will demonstrate your creativity by designing your own data structures that are efficient and easy to understand. Use simple interfaces.

You are free to design your own format and data structure for the routing and data messages. Just make sure your program handles these messages appropriately. Also, you are not required to implement all the packet header fields; you only have to implement the header fields that are sufficient in demonstrating the above specifications. For implementation purposes, you will have to workout many of the details that are necessary for the true representation of the protocol and for your program to work. Document your worked-out details at the start of your source code as COMMENTS.

The most important element in this assignment is to understand the logic behind the handling of routing messages so that you can implement it correctly. For example, you can write one generic client program for the routers. Specific router instances can be created on the fly by running these generic processes with command line parameters that provide specific details such as ID. For router initialization, you may create one input (configuration) file per router and pass the name of the input file as a command line parameter. Input files can contain configuration information such as, neighbour routers, their IP addresses etc.

For communication between hosts, use sockets. Note that broadcast and unicast both are used in this project. You need to ensure that a broadcast message reaches all the neighbours, while the unicast message reaches only its next hop.

3.3 Support Provided

If you have problems doing this assignment, you can post your question on the discussion forum. We strongly encourage you to do this as asking questions and trying to answer them is a great way to learn. Do not be afraid that you question may appear to be silly, the other students may very well have the same question!

We generally DO NOT provide any explicit support for the programming aspects for this project. However, should you require further clarifications on the given specifications, please do not hesitate to send queries via the discussion board provided.

It is important to make sure that your code compiles and runs correctly with Python 2.7.x UNIX-based machines. If your code does not compile, you will receive zero mark for your assignment and this has happened to some students in the past.

Your submitted code would be checked for similarities and any instances of plagiarism would result in severe consequences including award of ZERO marks for all such students, irrespective of who has shared with whom. No exceptions!!

3.5 Submission & Report:

You are required to submit your source code and a short report in a file named project.doc to an upload link that would be made available on the LMS. Zip your source code files and your report document in a file named Project_Surname_Surname (Surname of both Group members). Nominate one of the group members to submit on behalf of the group. Please note that you must upload your submission BEFORE the deadline. The LMS would continue accepting submissions after the due date. Late submissions would carry penalty per day with maximum of 2 days late submission allowed (see Section 3.6). Students who fail to submit would not be allowed to appear in viva and those who miss the viva would not be allowed to retake the viva.

You must write down group members' Registration No's and Names at the beginning of the report **!!** All reports will be read for marking purposes.

The size of your report **MUST be under 3 pages**. Your report should briefly document your techniques and methodology used for implementation and how you combat the relevant problems in development. It should act a reference for your instructor to quickly figure out what you have and haven't completed, how you did it, and it should mention anything you think is special about your system. You will be asked to demonstrate your program during your viva.

3.6 Late Submission Penalty:

Late penalty (on your received marks) will be applied as follows:

1 day after deadline: 25% reduction

2 days after deadline: 50% reduction

3 days after deadline: Not accepted