Lab c2: 802.11 Fairness and Comparison of DSR and AODV Routing Protocols

PART I - Comparison of DSR and AODV Routing Protocols

Introduction

This assignment aims at making the students familiar with the reactive (on-demand) routing protocols used in ad hoc networks and compare the performance of DSR and AODV protocols.

Performance Metrics used

We will evaluate the routing protocols in terms of two metrics-
1. Packet Delivery Fraction/Ratio : It is the ratio of packets delivered to that transmitted by the source.
2. Routing Load : It is the number of routing packets sent per data packet delivered.

NS2 Instructions

- 1. Get the tcl script from here.
- 2. This script, "compare.tcl" takes 4 command line arguments - scenario file, traffic file, output trace file and routing protocol(1 = DSR and 2 = AODV). Script usage is :

```
$ ns compare.tcl -scen {scen} -tfc {tfc} -tr {tr} -rpr {rpr}
```

  This script requires, a (i) Scenario file and a (ii) Traffic file. These files can be generated using third party tools which are now part of the NS2 installation

- 3. To simplify your work in creating these scenario files, we have a script which can do this for you automatically. The script named "make-scenario.sh" is available from here
  NB: This script is tested on EWS, and is tuned to work with the EWS NS2 installation. However, it only requires trivial modifications to work with your own NS2 installation.
  To generate the scenario files, just run this script as follows.

```
$ chmod +x make-scenario.sh
$ ./make-scenario.sh
```

  This script will generate 5 mobility scenarios for you, and are placed in a directory named "movement".

- 4. Next, is to generate the required traffic patterns. This is also done using a script named "make-traffic.sh" available from here. You are to run it in a similar fashion with the following command.

```
$ chmod +x make-traffic.sh
$ ./make-traffic.sh
```

  This script generates four traffic patterns and places them in a directory named "traffic".

- 5. After generating the required scenarios and traffic patters, we are to run them with a specific routing protocol. As expected, we will be using the DSR and AODV protocols on these scenarios. Again, we have two scripts which will run the script "compare.tcl" with different options and also extract information of interest. The scripts are named "run-aodv.sh", available here and "run-dsr.sh", available here. You can run these scripts as follows.

```
$ chmod +x run-aodv.sh
$ chmod +x run-dsr.sh
```

```
$ ./run-aodv.sh
$ ./run-dsr.sh
```

- 6. The above scripts will create 3 files each. The files have a suffix recv, sent, route_pkts, to mean received, sent and routing packets. Each line in the file is "Pause Time", "CBR Load", "Extracted Number".

Hand In (Part I)

1. Plot each performance metric for both DSR and AODV versus pause-times, for each CBR Load.
2. State if any peculiar behavior is observed. Give a brief report as to the interpretation of the graphs.
3. Write briefly about what you understand by the random node movement files (generated by ./setdest command in the make-scenario.sh script file) and traffic pattern files (generated by ns cbrgen.tcl command in the make-traffic.sh file) and their usage in this assignment.

Recommendations

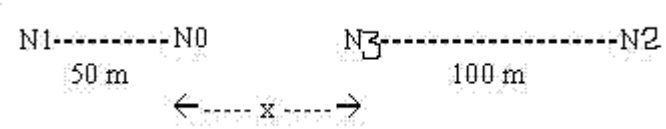
1. You will need to use the trace files calculation in many ns2 simulations. Hence, it is good idea to get familiar with the trace file formats. Go through the sections 16.1.6 and 16.1.7 of the ns2 manual. Can you now make out the use of grep commands provided in this assignment?
2. If time permits, try running the whole experiment with seed = 2 in the run-aodv.sh and run-dsr.sh, in the "ns cbrgen.tcl..." command. And make the plots again. Does the behavior change markedly?
3. Study about the cmu-gen utilities we have used in this assignment (used in the make-scenario.sh and make-traffic.sh files).

PART 2 - 802.11 Fairness

Introduction

The goal of this exercise is to understand the concept of starvation and unfairness in IEEE 802.11 standard.

We consider the following topology:



There are 2 CBR flows: f1 (N0->N1) and f2 (N2->N3) that start at the same time.

NS2 Instructions

- 1. Get the tcl script from here.
- 2. A script "fairsim.tcl" is provided. This script takes a single command-line argument "**dist**" and creates a toplogy like that shown in the figure with x varying as the argument. Script usage is :

```
ns fairsim.tcl -dist {x}
```

- 3. There are 2 CBR flows: f1 (N0->N1) and f2 (N2->N3) that start at the same time.
- 4. Run simulations for x=100m, 200m, 300m, 450m, 500m, 600m.
- 5.Determine the throughput of flows f1 and f2. Plot the throughput of f0 and f1 on the same graph with the distance x on the x-axis. To calculate throughput, you need to calculate [num of received packets]. Following command will give the number of received packets where {num} is 1 for flow f1 and 3 for flow f2.

```
grep "AGT" fairsim.tr | grep "^r" | grep "Ni {num}" | wc -l
```

To understand more about the trace file and what it contains, refer to sections 16.1.6 and 16.1.7 of the ns2 manual.

Hand In (Part II)

Turn in plots from the previous section. Expected plots are the per flow throughput vs distance x for different x values. Also, turn in your answers to the questions in the analysis section.

Analysis

This assignment lets you understand how unfairness can play its role in 802.11. Looking at the results, you will realize that there is lot happening behind this simple looking topology. First try to recall the following about 802.11 and ns2 , then try to answer latter questions with respect to the graph obtained above-

1. If a node correctly decoded the last packet detected, it must sense the channel to be idle for DIFS prior to resuming backoff. If it detected a packet but couldn't correctly decode it, it must backoff for EIFS. EIFS > DIFS.
2. If a node sends an RTS and does not receive a CTS, it doubles its contention window and then retries.
3. As per default ns2 settings, TX Range is approx. 250m and CS Range approx. 550m. This simply means that nodes within TX Range can detect as well as decode the packet correctly but nodes outside TX Range but within CS Range are only able to detect the packet. This further implies that such nodes(outside TX Range) would back off for EIFS and not DIFS.

Keep these in mind and answer the following:

Question-

1.What do you observe from the graph obtained? Can you explain the observations on the basis of the 802.11 contention resolution/backoff mechanism?

2.There are different types of "fairness", please explain the difference between short-term and long-term fairness. Which type is harder to achieve?

References

1. Overview of IEEE 802.11
2. Know ns2