SAN JOSÉ STATE

UNIVERSITY

EE-289 PROJECT#2

SIMULATION & STUDY OF ADHOC ROUTING PROTOCOLS

Submitted By:

Arpit Singh (010810514)

Ankita Chandrachud (010804833)

Aakash Shah (010819926)

Harsh Raijiwala(010827765)

TASK-3:

The trace file was generated by considering values:
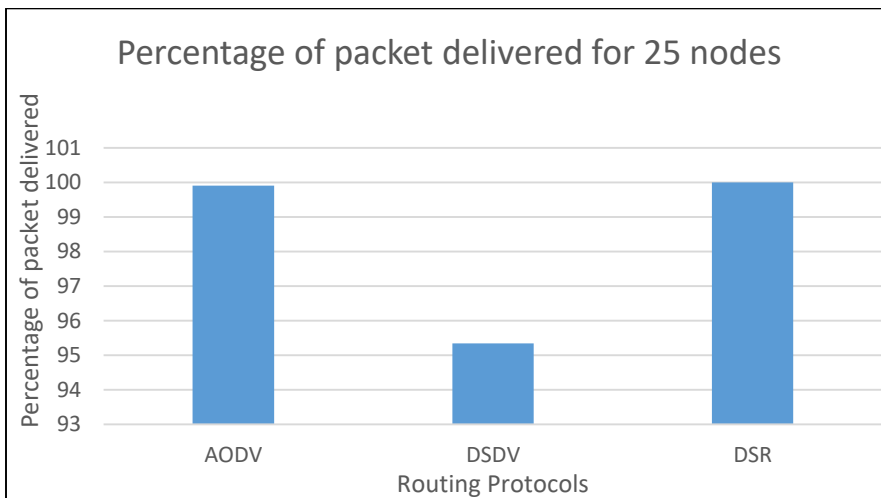
Number of nodes: 25
_ Pause time: 10
_ Max speed: 1
_ Connections: 5

The results obtained are tabulated below:

routing protocol	Percentage of packet delivered	Number of packet sent	Number of packet received
AODV	99.91	2399	2397
DSDV	95.34	2406	2294
DSR	100	2408	2408

Graph of results obtained:

Percentage of packet delivered for 25 nodes

(Bar chart: y-axis "Percentage of packet delivered" from 93 to 101; x-axis "Routing Protocols" with AODV ≈ 99.91, DSDV ≈ 95.34, DSR = 100)

Observation:

In case of less mobility (speed =1) and pause time = 10. In the above graph it can be seen that the AODV and DSR performs particularly well, DSR gives 100% packet delivery.

Screenshots of Output Obtained:

DSR Protocol:

```
ubuntu@ubuntu:~/Desktop/Finalproject2/25nodes/DSR$ python parser.py simple.tr
Number of packets sent:-  2408
Number of packets received:-  2408
Percentage of packet delivered:-  100.0
Average delay:-  0.0154281290569
```

DSDV Protocol

```
ubuntu@ubuntu:~/Desktop/Finalproject2/25nodes/DSDV$ python parser.py simple.tr
Number of packets sent:-  2406
Number of packets received:-  2294
Percentage of packet delivered:-  95.3449709061
Average delay:-  0.0171658726935
```

AODV Protocol

```
ubuntu@ubuntu:~/Desktop/Finalproject2/25nodes/AODV$ python parser.py simple.tr
Number of packets sent:-  2399
Number of packets received:-  2397
Percentage of packet delivered:-  99.91663193
Average delay:-  0.0218568869174
```

Task 4:

Value of parameters were taken as:
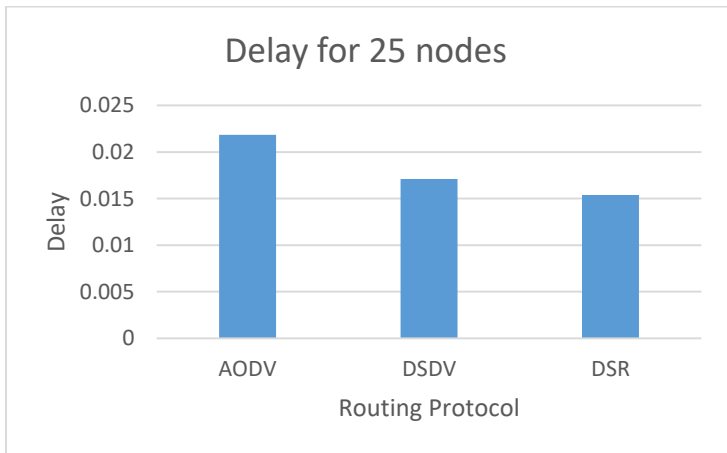
Number of nodes: 25
_ Pause time: 10
_ Max speed: 1
_ Connections: 5

The delay observed for routing:

Routing Protocol	Average Delay
AODV	0.02185
DSDV	0.0171
DSR	0.0154

Graph for Delay of 25 nodes:

Observation:

For high pause time DSR has the lowest delay, in this case pause time (10) is high compared to the speed (1). AODV in this scenario gives the highest delay because it is a reactive algorithm and exchanges tables.

(2). DSDV has pre-configured routes which facilitates faster delivery.

(3). DSR directly sends the packet which allows it in the fastest delivery in case of high pause and low speed.
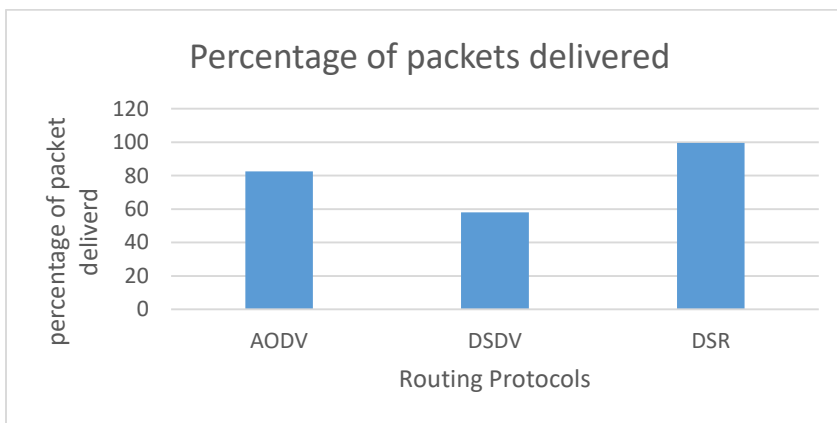
TASK 5:

Values of 10 nodes:

Routing Protocol	Percentage of packets delivered	Number of packet sent	Number of packet Received	Average Delay
AODV	82.6	1194	987	0.0749
DSDV	58.1	1203	699	0.0058
DSR	99.498	399	397	0.6019

Number of nodes: 10
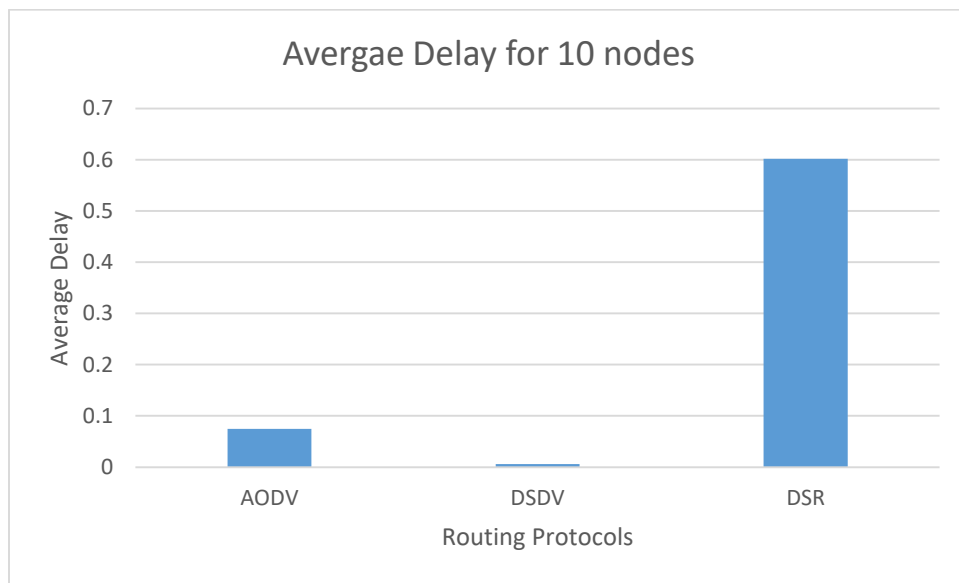_ Pause time: 10
_ Max speed: 1
_ Connections: 2

OBSERVATION:

In the above scenario there are 10 nodes with a maximum of 2 connections, max speed is 1 and pause time is 10.

DSR gives the maximum packet delivery ratio due to less mobility

Delay:

Avergae Delay for 10 nodes

Observation:

In the above scenario the number of connections is 2 and total nodes are 10, with a pause time of 10 and speed of 1.

DSR experiences higher delay in case of less number of nodes because of the cache is DSR and lack of mechanisms to expire stale routes.

DSDV experiences the least delay because of its proactive nature and less number of connections.

Screenshots of Output Obtained:

DSDV Protocol:

```
ubuntu@ubuntu:~/Desktop/Finalproject2/10nodes/AODV$ cd ../DSDV
ubuntu@ubuntu:~/Desktop/Finalproject2/10nodes/DSDV$ python parser.py simple.tr
Number of packets sent:-  1203
Number of packets received:-  699
Percentage of packet delivered:-  58.1047381546
Average delay:-  0.0058154335608
```

AODV PROTOCOL:

```
AODV.tcl~  h10p10s1_mov.tcg  parser.py            simple.nam
ubuntu@ubuntu:~/Desktop/Finalproject2/10nodes/AODV$ python parser.py simple.tr
Number of packets sent:-  1194
Number of packets received:-  987
Percentage of packet delivered:-  82.6633165829
Average delay:-  0.0749831118419
```
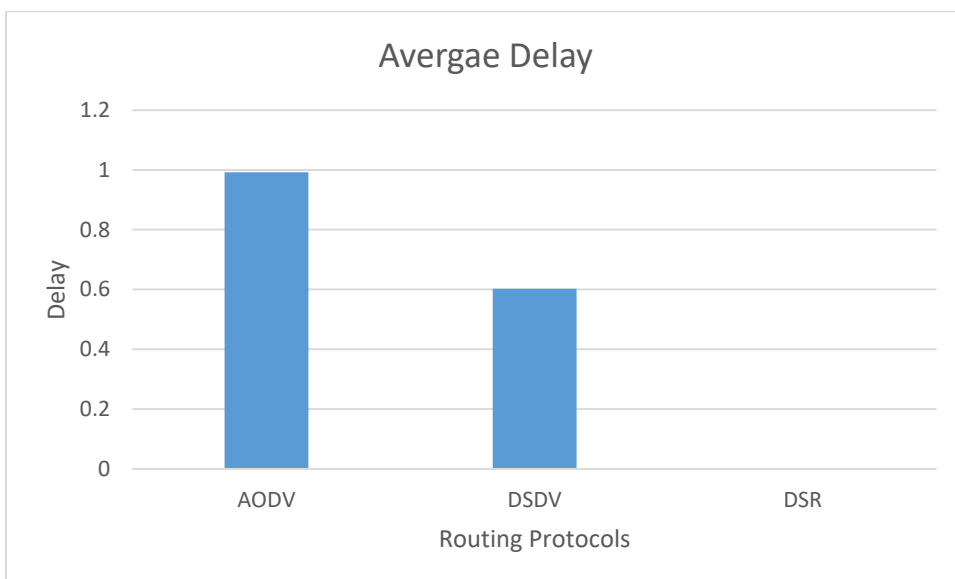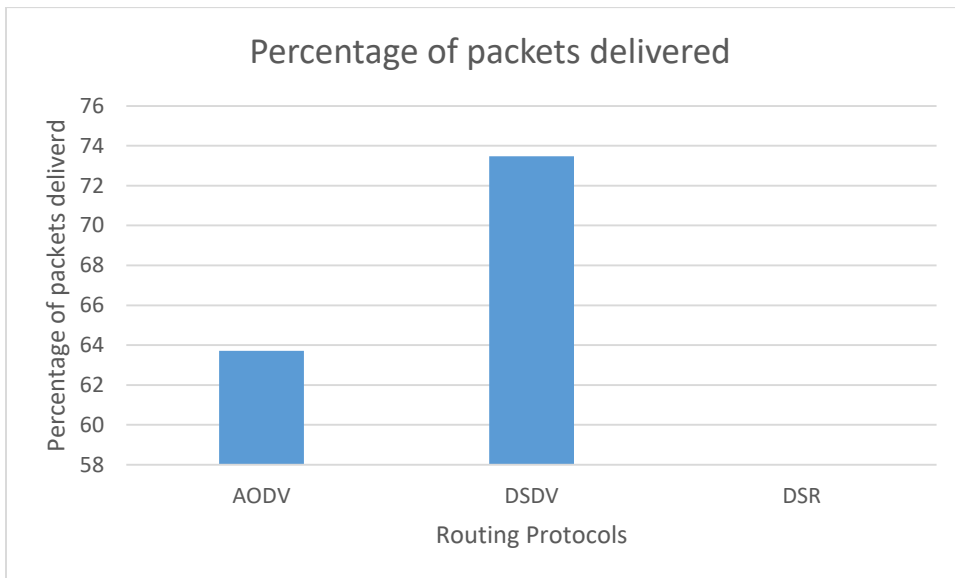
DSR PROTOCOL:

```
ubuntu@ubuntu:~/Desktop/Finalproject2/10nodes/DSR$ python parser.py simple.tr
Number of packets sent:-  399
Number of packets received:-  397
Percentage of packet delivered:-  99.4987468672
Average delay:-  0.601984310746
```

Values in case of 100 nodes:

Routing Protocols	Percentage of Packets Delivered	Numbers of packet Sent	Number of packets Received	Average Delay
AODV	63.72	8909	5677	0.9922
DSDV	73.48	8897	6538	0.6024
DSR	NA	NA	NA	NA

Note: Unable to calculate DSR in this case due to the segmentation fault error. Tried increasing the RAM but got the same error.

OBSERVATION:

In the above case DSR is not available due to the segmentation fault error, and it can be observed that the DSDV gives least delay in case of change in nodes.

Screenshots of Output Obtained:

DSR Protocol:

```
ubuntu@ubuntu:~/Desktop/Finalproject2/100nodes/DSR$ python parser.py simple.tr
Number of packets sent:-  1
Number of packets received:-  0
Percentage of packet delivered:-  0.0
Average delay:-
Traceback (most recent call last):
  File "parser.py", line 80, in <module>
    print "Average delay:- ",delay1/i
ZeroDivisionError: integer division or modulo by zero
```

DSDV PROTOCOL:

```
ubuntu@ubuntu:~/Desktop/Finalproject2/100nodes/DSDV$ python parser.py simple.tr
Number of packets sent:-  8897
Number of packets received:-  6538
Percentage of packet delivered:-  73.4854445319
Average delay:-  0.602418317188
```
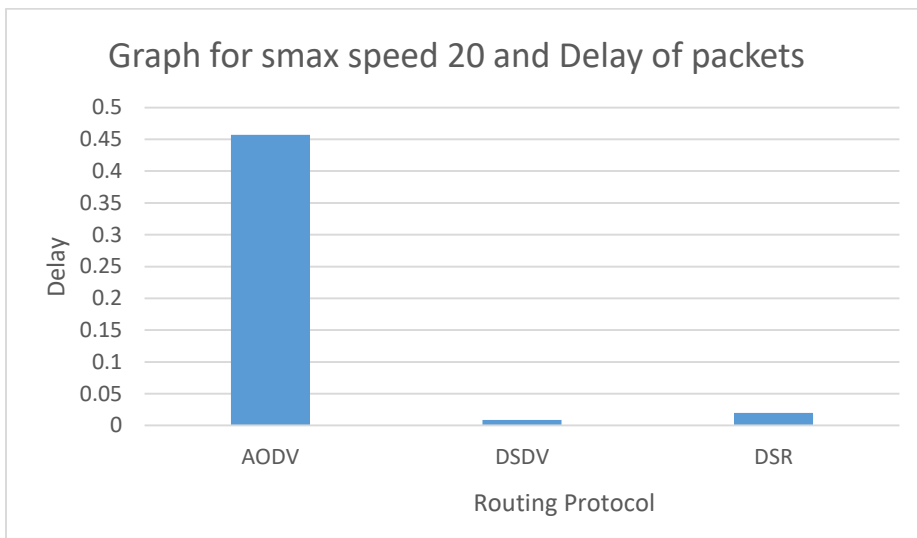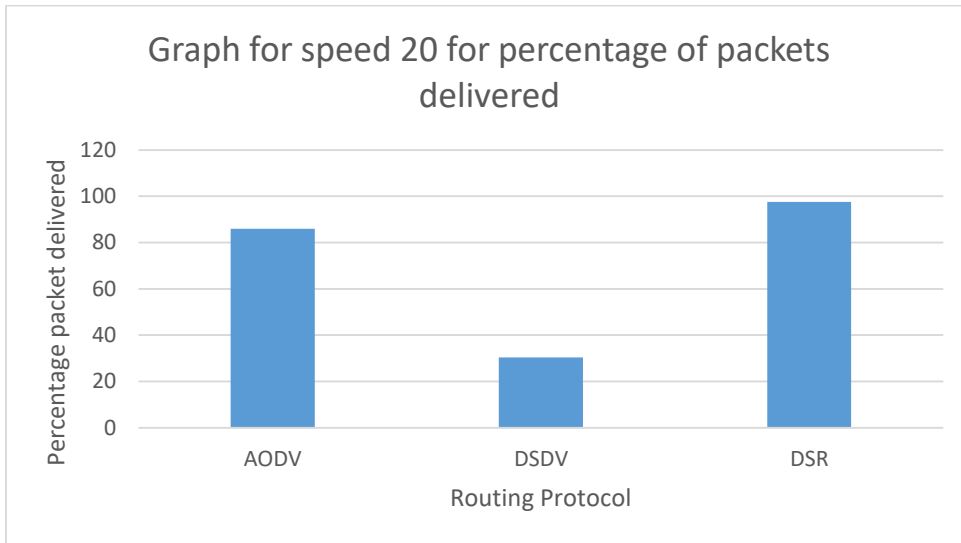
AODV PROTOCOL:

```
ubuntu@ubuntu:~/Desktop/Finalproject2/100nodes/AODV$ python parser.py simple.tr
Number of packets sent:-  8909
Number of packets received:-  5677
Percentage of packet delivered:-  63.7220787967
Average delay:-  0.992217214672
```

Task 6:

PDR AND DELAY FOR SPEED = 20

Routing Protocol	Average Delay	Percentage packet delivered	Number of Packets sent	Received
AODV	0.457	85.91	2399	2061
DSDV	0.009	30.37	1264	384
DSR	0.02	97.56	41	40

Observation:

In the above scenario maximum speed is 20, which leads to change in the delay of packets. In this case AODV experiences highest delay among all the three protocols. In case of throughput DSR gives the maximum throughput.

DSR proves to be the best protocol in case of high mobility because it experiences higher throughput and minimal delay.