Performance Evaluation of Migration Strategies in Mobile AdHoc Networks

G13-2017 Mode of Project : A

Anuraadhaa Kandadai
Final Year Student, SNU
1310110064
ak411@snu.edu.in

Apurva Bansal
Final Year Student, SNU
1310110069
ab461@snu.edu.in

Sushrut Khandelwal
Final Year Student, SNU
1310110329
sk467@snu.edu.in

Supervisor:
Dr. Anuradha Ravi
Assistant Professor, SNU

Project Overview

- The aim of the project is to analyse different migration strategies for seamless service provisioning as there is high risk of service discontinuation in Mobile AdHoc Networks.
- List down evaluation criteria for performance analysis in different existing migration strategies in the ad hoc Cloud.
- Analysing the migration strategies with different migration techniques based on the evaluation criteria.
- We intend to check the feasibility of migration in SDN based ad hoc networks and compare the migration in SDN based network to that of a non-SDN based network.

System Architecture

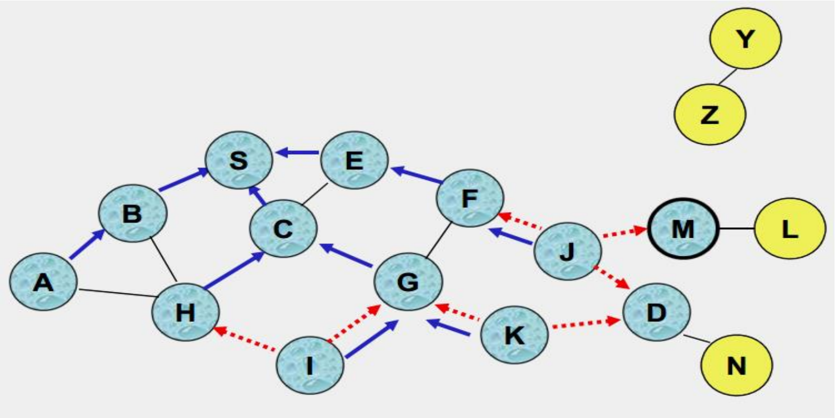
MANET

- A collection of mobile nodes which can move freely are connected by wireless links
- Node have reduced processing, storage, communication, and power resources. Each node has a communication range. Each node connects with all other nodes within its vicinity.
- IEEE 802.11n network using AODV has been implemented.

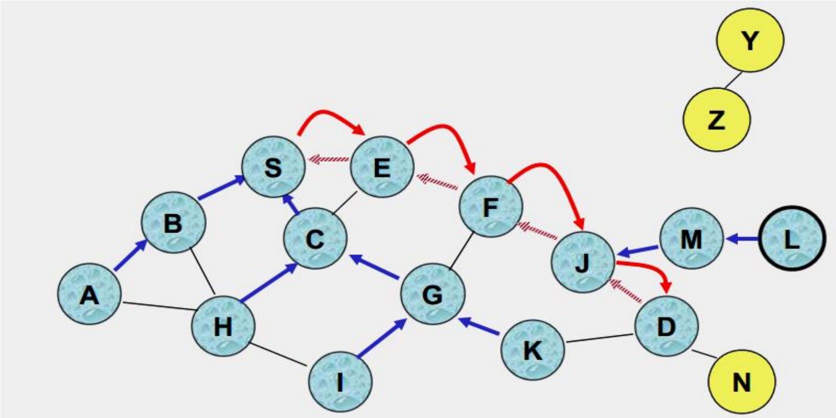
SDN

- SDN Controller
- SDN Forwarding Devices
- SDN Interfaces
- SDN Application
- Data plane routing table constructed by SDN controller. Updated whenever there is a loss of connectivity of nodes or any route is found with high traffic.
- Data plane routing -Dijkstra's Shortest Path Routing Algorithm.
- Control plane routing- AODV

Algorithms-AODV



Reverse Path Setup in AODV



Forward Path setup in AODV

- Source node initiates path discovery by broadcasting a route request RREQ packet to its neighbors.
- Neighbor either satisfies the RREQ by sending a route reply RREP back to the source or re-broadcasts the RREQ to its neighbors after increasing the hop count.
- When either the destination or some intermediate node moves away from the network or gets disconnected due to node failure, RRER is sent to the nodes.

Algorithms-SDN Data Plane

```
#Node[1]'s Routing Table at time:  56.00
Src_Node[1]:1.0.2.1 -> Dst_Node[4]:1.0.2.4, Next_Hop[4]:1.0.2.4
Total routing rules: 1
#Node[1]'s Routing Table at time:  61.00
Src_Node[1]:1.0.2.1 -> Dst_Node[4]:1.0.2.4, Next_Hop[4]:1.0.2.4
Total routing rules: 1
#Node[1]'s Routing Table at time:  66.00
Src_Node[1]:1.0.2.1 -> Dst_Node[4]:1.0.2.4, Next_Hop[4]:1.0.2.4
Total routing rules: 1
#Node[1]'s Routing Table at time:  71.00
Src_Node[1]:1.0.2.1 -> Dst_Node[4]:1.0.2.4, Next_Hop[4]:1.0.2.4
Total routing rules: 1
```

Data plane routing table

- Dijkstra's Shortest Path Routing Algorithm
- A graph G= (V, E) which is weighted, directed and connected, derived from the SDN topology.
- For a given source node and a destination node, the algorithm can find the shortest path between them. The algorithm considers the node weights and edge weights to make its decisions.

To create the data plane routing table, NODE[1] broadcasts the hello messages for neighbour discovery.

Neighbour Update Message is sent to the controller of the network with neighbour node's id.

The controller replies with the UpdateDataPlaneRouting message with the destination node's id and next hop node's id

Algorithms-SDN

```
Time: 2.000000 sec
Packet_Type                Count Packet_total_lengths
Hello                      1           80
===========================================
Time: 3.000000 sec
Packet_Type                Count Packet_total_lengths
Hello                      17          1360
NeighborUpdate             48          26112
UpdateDataPlaneRouting     88          70048
===========================================
Time: 4.000000 sec
Packet_Type                Count Packet_total_lengths
Hello                      2           160
NeighborUpdate             8           4352
UpdateDataPlaneRouting     14          11144
===========================================
Time: 5.000000 sec
Packet_Type                Count Packet_total_lengths
Hello                      10          800
NeighborUpdate             26          14144
UpdateDataPlaneRouting     40          31840
===========================================
Time: 6.000000 sec
Packet_Type                Count Packet_total_lengths
NeighborUpdate             6           3264
```

SDN control plane routing

As in traditional AODV, each node broadcasts Hello packets in its neighbourhood for identifying its neighbours.

Neighbor Update Messages are sent from nodes to controller for further monitoring of the network and formation of DPR table.

Data Plane Routing Rule update messages are sent from controller to all nodes.

Experimental Setup

AODV

- Each mobile device in the simulator models a wireless host with 1 wireless (802.11) card.
- Hosts also contain routing, mobility and battery components.
- UDP used as transport protocol.
- These mobile nodes participate in ad-hoc routing and have UDP applications installed.

SDN

- Each SDN mobile device is equipped with 2 Wi-Fi interface cards – one is for the control plane and the other for the data plane.
- An SDN mobile device can play the role of data forwarder (an agent) or routing-path arbitrator (a controller).
- No OpenFlow; Instead, a suite of control messages are defined to support the required SDN operations.

Experimental Setup

Parameters	Value
Constraint X coordinates	0-600m
Constraint Y coordinates	0-500m
Carrier Frequency	2GHz
Bandwidth	2MHz
Transmitter Power	1.4mW
Preamble Duration	10us
Header Bit Length	0 b
Sensitivity	-80 dbm
Energy Detection	85dBm
SNIR Threshold	4dB
Maximum Queue Size	10 packets

WITH AODV ROUTING

Source	Destination
host A	host B
host[0]	host[5]
host[2]	host[7]

Source	Destination
host A	host B
host[0]	host[5]
host[1]	host[8]

WITH SDN (AODV as Control Plane and Dijkstra as Data Plan)

Source	Destination
host1	host4
host2	host5
host6	host9

Source	Destination
host1	host3
host9	host4
host11	host7

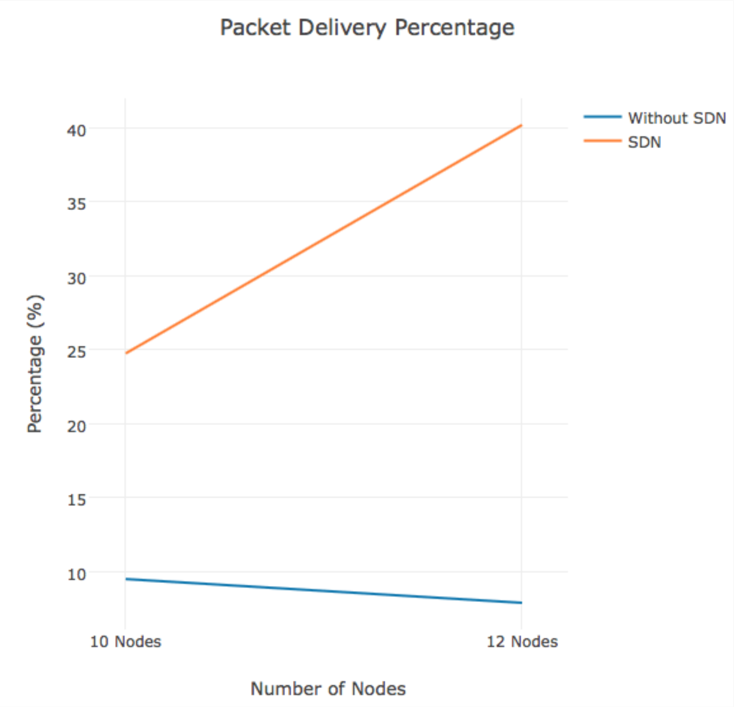
Results

		WITH AODV ROUTING						
Packet Length : 1028 bytes **Hosts mobility speed** : 5-7 mps **Sim_Time** : 100sec **Area** : 600x500m2								
Number of nodes	**Host A: No of Packets Sent**	**Host B: No of Packets Received**	**Downtime (s)**	**Migration Time(s)**	**Control Overhead(Bytes)**	**Packet Delivery Ratio**	**Throughput (KBps)**	**Last Event (s)**
10(A-B)	2115	213	16.07558226	16.41497773	127152	0.1007092199	9.254626325	23.65995042
10(0-5)	2118	277	13.45721804	13.74210957		0.1307837583	11.61684842	24.51232811
10(2-7)	2023	108	12.16464616	12.60575083		0.05338606031	6.439192162	17.24191439
12(A-B)	2123	267	7.751717334	8.412770404	88220	0.1257654263	11.01238169	24.92430864
12(0-5)	2038	74	5.796170405	5.830004029		0.03631010795	9.518593274	7.991937234
12(1-8)	2117	158	7.22337484	7.23183307		0.07463391592	6.49718769	24.99912389

Results

		WITH SDN (AODV as Control Plane and Dijkstra as Data Plan)							
Packet Length : 1024 byte **Hosts mobility speed** : 5-7 mps **Sim_Time** : 100sec **Area** : 600x500m^2									
Number of nodes	**Host A: No of Packets Sent**	**Host B: No of Packets Received**	**Downtime(s)**	**Migration Time(s)**	**Control Overhead (Bytes)**	**Packet Delivery Ratio**	**Throughput(KBps)**	**Avg Tx_Pkt size (Bytes)**	**Avg Rx_Pkt size (Bytes)**
10(1-4)	14428	1120	10.272	30.232		0.07762683671	68.6	2210	853.43
10(2-5)	33970	16697	0	2.32	43840	0.4915219311	327.7	2210	611.66
10(6-9)	28233	18174	16.536	29.605		0.6437148018	537.6	2210	2204.36
12(1-3)	31189	22693	16.44	26.961		0.7275962679	620.5	2210	2176.53
12(9-4)	22901	7924	1.143	2.799	49920	0.3460110912	439.3	2210	2168.36
12(11-7)	14298	1885	4.321	11.099		0.1318366205	596.9	2210	1184.94

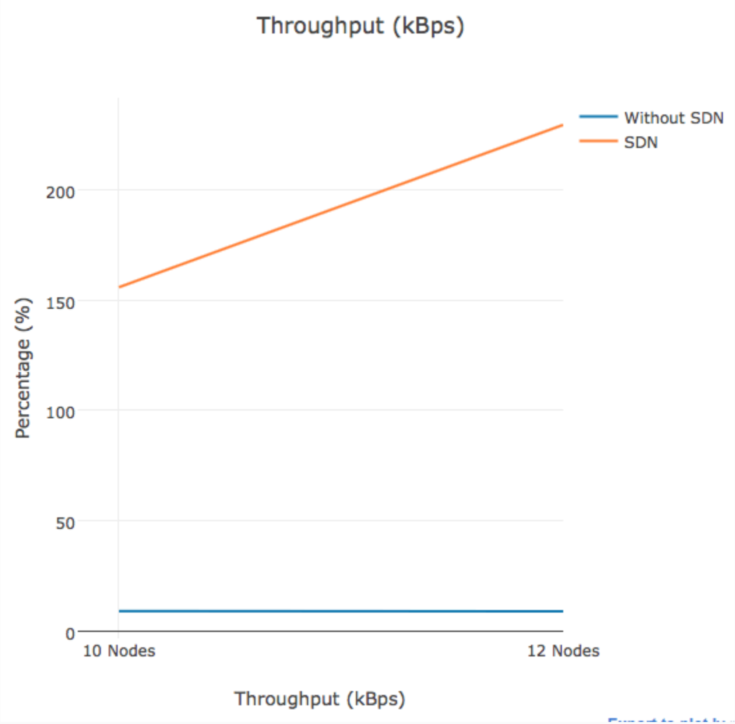
Packet Delivery Percentage



Packet Delivery Percentage for AODV and SDN Framework

- It can be observed that packet delivery ratio increases significantly for SDN even when the number of nodes increases from 10 to 12.
- In case of traditional AODV, the PDR is almost constant.
- This is found to be due to the involvement of SDN controller, which programmatically defines the route for the network.

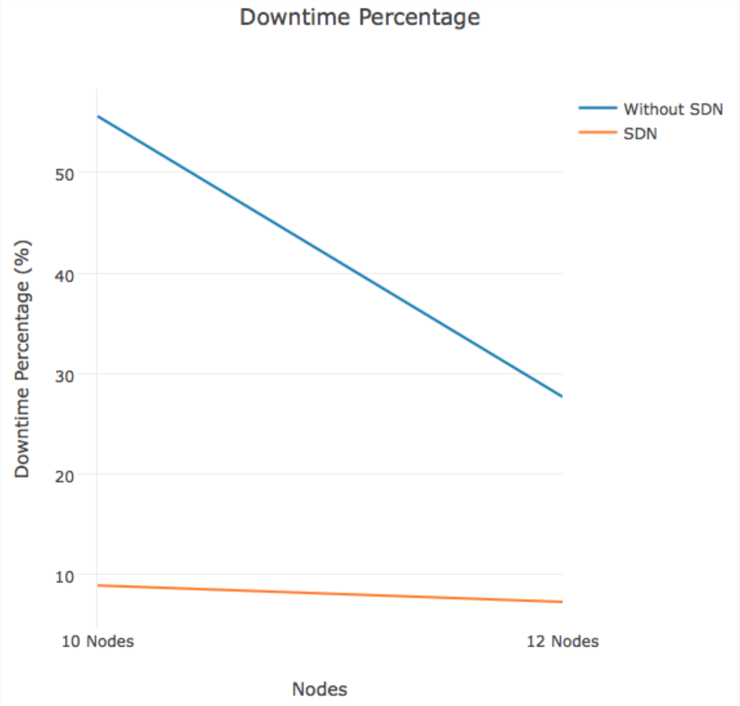
Throughput



Throughput in KBps for SDN and AODV Framework

- **SDN performs better than AODV in terms of Throughput.**
- **The controller monitors the entire network and sends DataPlaneRoutingUpdate to all the nodes.**
- **This reduces the number of updates to be communicated in distributed manner, thus reducing the downtime and migration time which gives raise to increased throughput.**

Downtime and Migration Time



Downtime Comparison for SDN and AODV

- **The figure shows a decreasing trend on increasing number of nodes.**
- **This is due to the random mobility of nodes.**
- **When compared to AODV, the downtime is far less due to the SDN controller monitoring the entire network.**

Downtime and Migration Time

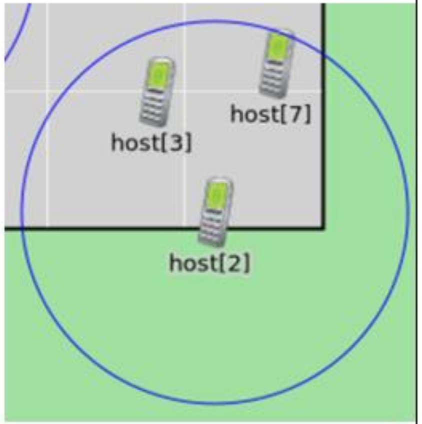


Fig 5.12a

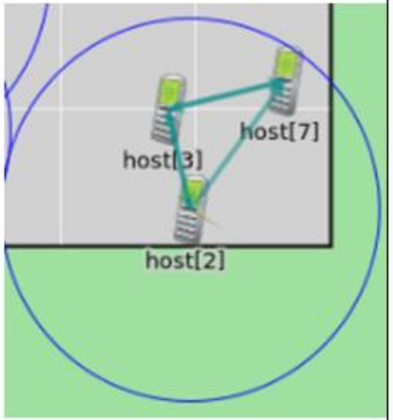


Fig 5.12b

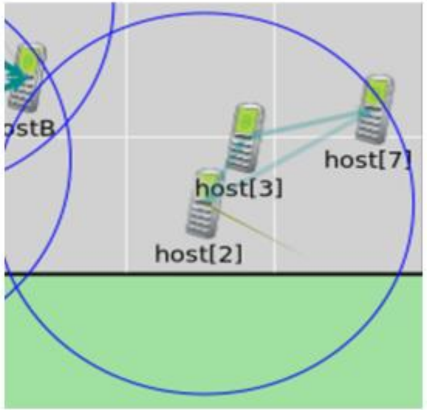


Fig 5.12c

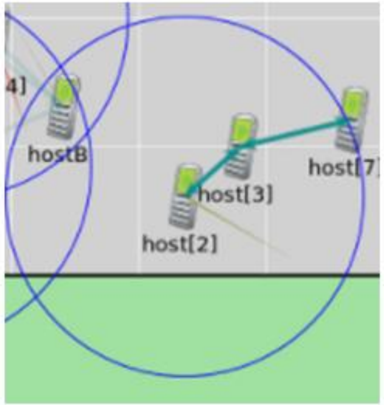
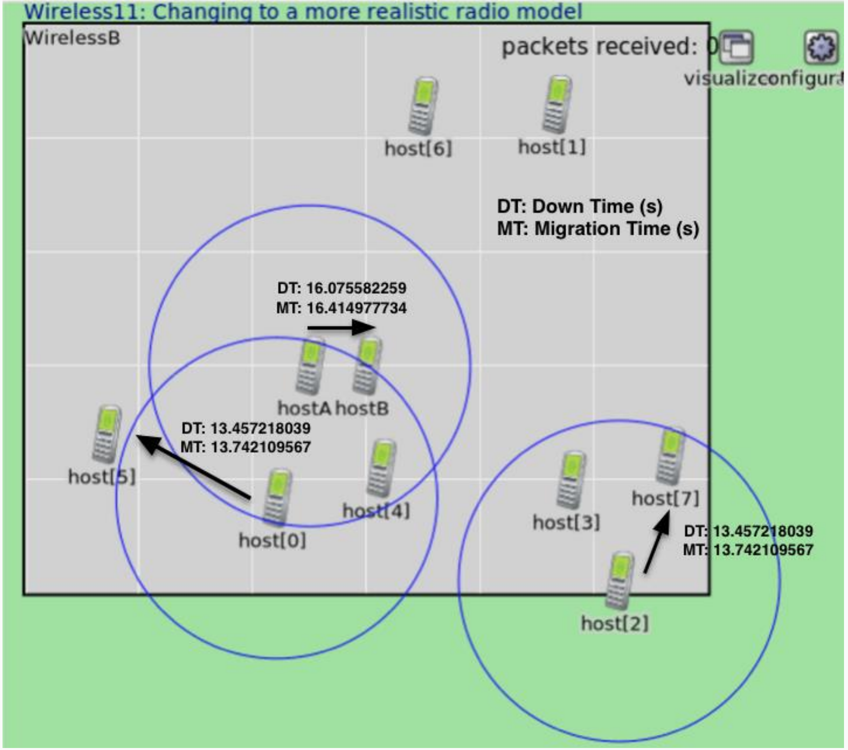


Fig 5.12d

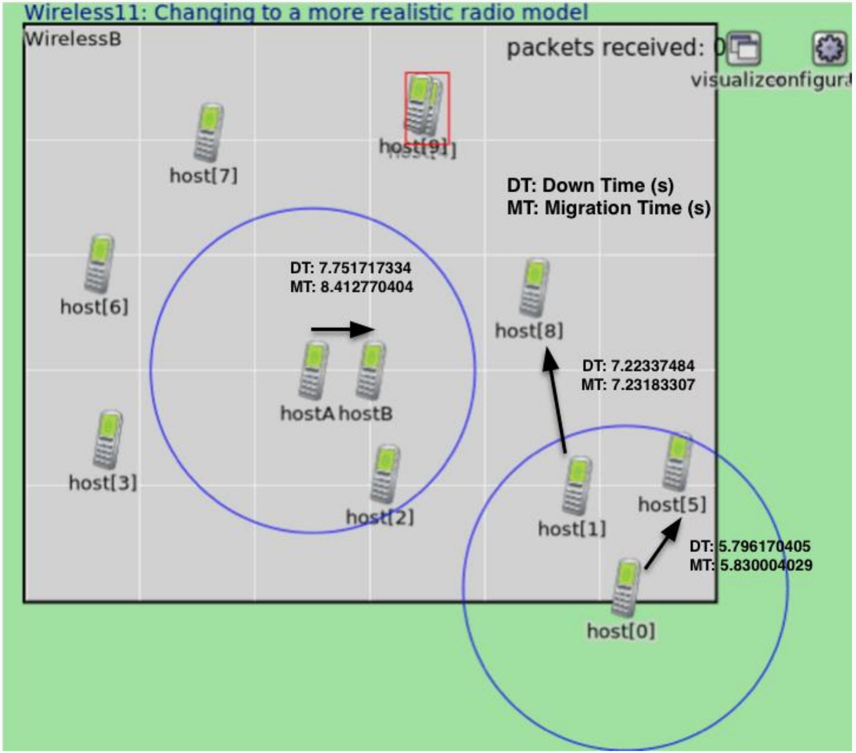
Displays migration between source host[2] and destination host[7] with AODV routing.

Downtime and Migration Time

WITH AODV ROUTING



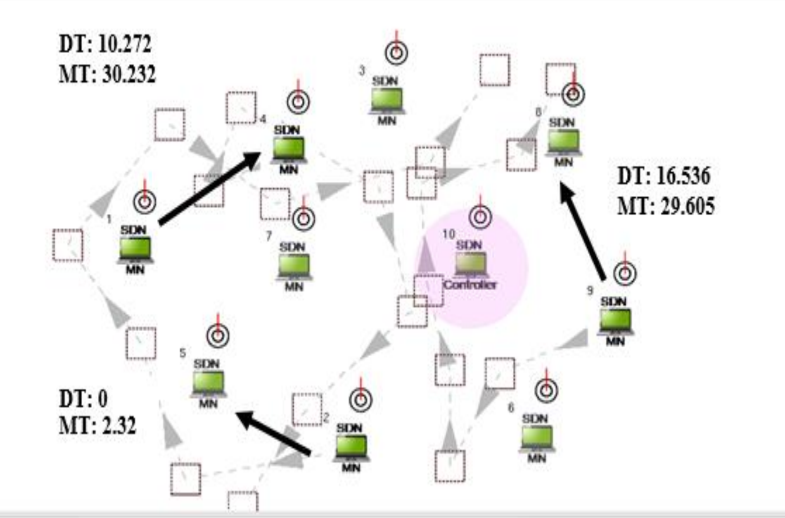
Downtime and Migration time for 10 nodes



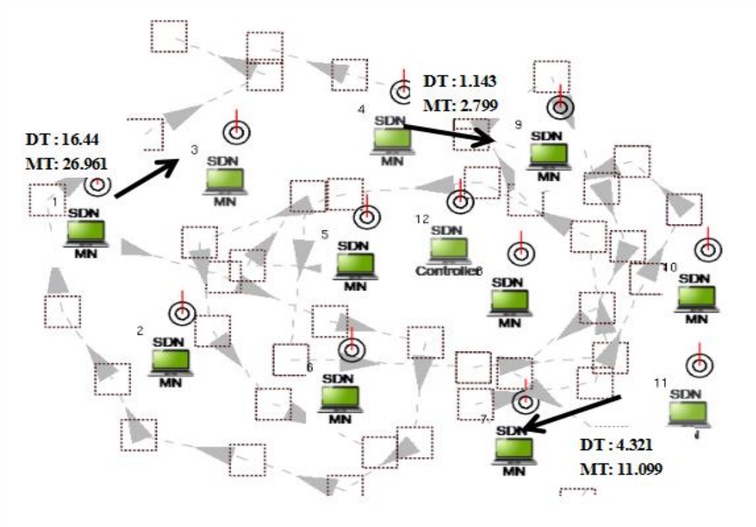
Downtime and Migration time for 12 nodes

Downtime and Migration Time

WITH SDN (AODV as Control Plane and Dijkstra as Data Plan)



Downtime and Migration time for 10 nodes



Downtime and Migration time for 12 nodes

Conclusion and Future Scope

- Migration performed in SDN yields better throughput, lower downtime, lower control packets, lower migration time and better packet delivery ratio compared to AODV routing.

- SDN controller reduces the number of updates to be communicated in distributed manner, thus reducing the downtime and migration time. Reduced the control overhead when compared to AODV.

- In case of failure of link between the controller and the mobile node, there has to be an alternate controller chosen for the network which is a disadvantage.

- Efficient algorithms are yet to be proposed for selection of controller and routing algorithms for data plane that can improve the throughput of the network.

Extra Slides

Migration Strategies

- Offloading to Cloud
- Offloading to Coudlet
- Offloading to MAC

Migration Techniques

- Based on Workload
- Based on Network Characteristics
- Based on Interaction between Nodes