Distributed Cache Updating for Dynamic Source Routing Protocol

Shivaprasad S Shivakalimath[1], Mrs. Pratibha Kantanavar[2]

[1]Student, Dept. of Electronics and Communication Engineering, R V College of Engineering, Bangalore
[2]Assistant Professor, Dept. of Electronics and Communication Engineering, RV College of Engineering, Bangalore – 59

Abstract - *Routing is one of the challenging tasks in wireless ad hoc networks using on-demand routing protocols which relay on route caches to make routing decisions. These route caches become stale due to node mobility and the changing network topology. The prior work used structures called the cache table to implement cache updating. This was implemented using DSR protocol that uses route cache in forwarding the packets as it involves less overhead and latency than other on-demand routing protocol. It proposed proactively removing the broken link information from the node cache. This mechanism was less effective in efficiently removing all the stale routes. In this work, we propose enhanced queue processing mechanism for distributed cache updating. This mechanism unlike other queue processing techniques processes the queue depending on the packet's priority. When a link failure is detected the entire queue is deleted and new routes are discovered and processed again on the basis of priority. Further we compare the performance of cache updating on existing DSR and with the proposed enhanced queue processing mechanism to achieve cache updating for DSR. The proposed work is simulated using NS-2 tool for five network scenarios with 10, 20, 30, 40 and 50 nodes. The proposed work outperforms the cache updating for existing DSR with overall decrease in end-to-end delay of 60.10 percent, and 26.13 percent of reduction in average energy consumption, having negligible increase in packet loss of 0.5 percent and PDR almost same. 0.56 percent throughput reduction and 6.99 percent increase in overhead.*

Key Words: **Dynamic Source routing, Queue processing, Cache updating, NS-2, Ad Hoc, MANETS, Priority queue, Network density**

1. INTRODUCTION

A wireless ad hoc network consists of different number of wireless nodes forming a wireless network which has no infrastructure or centralized administration. Consider an example of a military application where sensors are placed remotely for observing movements of the enemy troops in the battlefield. The sensors are nothing but wireless nodes, these nodes form an infra structureless network to communicate with each other. Under normal conditions the nodes function properly by communicating with each other under the desired conditions. Consider that the nodes are mobile, in such case it is challenging to maintain the communication with the nodes along the desired route. This causes breaking of the link and the nodes lose communication until they again set up the link. The function of node in such a temporary network is to provide aid to other nodes in forwarding/transmitting and in receiving the packets to the destination and intermediate nodes. This is achieved by caching the routes discovered in a routing table such that each node will maintain and use it for routing. But, as mentioned earlier mobility causes the network topology is continuously change and new routes are discovered and cached in the table. Every node in the network follows the same and the process is known as cache updating. This process comes with the problem of stacking up of stale routes that are not needed and the use of such stale routes will only cause the data packets to be lost. Hence, a cache updating mechanism for handling or deleting such stale routes from the routing table and updating the cache is important. The on-demand routing protocols make use of route caches for making routing decisions. Due to changing network topology and other link failures such cached routes become stale. To address the cache staleness, issue the work proposes cache updating for DSR based on enhanced queue processing technique which uses priority-based mechanism to route the packets based on priority. This model is exclusively used for DSR protocol, unlike other works that used a separate table called the routing table to route the packets.

1.1Dynamic Source Routing protocol

The Dynamic Source Routing (DSR) [2] proposed in the work is an on-demand routing protocol that is based on the concept of source routing. Mobile nodes maintain the route caches containing possible routes to some nodes it is transmitting the data or the one that is already cached. The route cache is updated regularly as and when new routes are discovered. The DSR protocol consists route discovery and route maintenance. When a mobile node has a packet to send, it checks the route cache to see a route to the destination is already exists or not. If a route is available then it will use this route to send the packet and If the node does not have such a route else the route in the table is expired, then route discovery is initiates by broadcasting a route request Route Request (RREQ) packet which contains source ID, destination ID and a unique sequence number. After every reception the nodes checks by comparing the destination ID to determine if the packet was for itself or not. If the destination ID do not match it refers that packet is for some other node. The intermediate node will then check its cache to see if a route is available to the destination specified in the packet's header, if yes, it adds its own address to the packet header and then forwards the packet along its path, if not then the node only transmits route request packet to reduce the network load. How will a node notify the sender of the packet whether the RREQ packet has reached the destination node or an intermediate node is by receiving route reply Route Reply (RREP) packet. This is sent by the receiver of the RREQ packet. The RREP packet received from the destination will contain the information related to the number of hops taken to reach the destination. The sender node may get different number of RREP packets to a set destination. It then chooses the path with a smaller number of hop counts and less cost. Fig 1 illustrates the route discovery process. The Fig 2 shows an example of route reply to the source. Route maintenance is one of the important operations for on-demand protocols to maintain the communication path free from

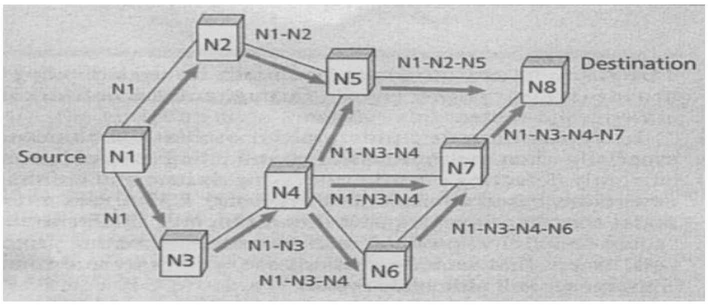

Fig -1: Building of route record during route discovery

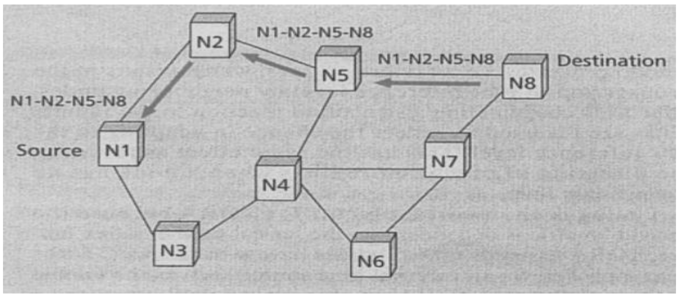

Fig -2: Propagation of the route reply with route records

causing errors or link failures. This includes route error packets that are generated at a node when the data link layer faces an error in the transmission. This problem may involve dropping of packets due to flooding at the receiver or due to link breakage. Upon reception of the route error packet, the path which caused the error is removed from the node's route cache and all routes containing that path in its cache are deleted. Acknowledgments are also used to verify the operation of the routing links. These are passive acknowledgments where a mobile is able to hear the next hop to where the packet will be forwarded along the path.

1.2Proposed approach

This paper proposes a mechanism to update the node cache efficiently in the events of link failure and network errors caused by nodes or due to geographical complications. This mechanism is implemented for the on-demand routing protocol i.e. the dynamic source routing protocol. A priority-based queue processing mechanism for updating the cache for DSR is proposed in this paper. The section 2 discusses the process of queue processing for cache updating and the performance metrics parameters used in the work. The section 3 briefs the methodology with flowcharts over the entire process. The section 4 discusses the results and chapter 5 briefs the overall conclusions derived from the work.

2. DISTRIBUTED CACHE UPDATING

Continuous changing topology results in link errors which are common for ad hoc networks. It is the routing protocol's function to re-establish the broken link. The time take to set up this broken link is termed re-route interval, collectively over a duration it is termed as re-route time. This duration is in particular the house for stale routes to exist which the routing protocol takes care of by initiating re-routing. In such case clearing the cache is important to avoid the use of stale routes that will cause packet loss and reduce the efficiency. For this the NS-2 provides priority-based queuing procedure package called the "CMUPriQueue" that depends the queuing procedures to be implemented specifically for DSR protocol. This module has certain blocked state functions to keep a check on the node status and automatically deque the queue in the event of link breakage. This prevents the stale routes again being used. This module requires setting up of certain thresholds to set priorities for the nodes. When thresholds like the queue length and the queue processing rates are given, the priorities are set on the queue is processed depending upon the processing rates initialized for nodes. In real world, wireless network nodes have different queuing mechanism and various queue length and queue processing rates, this becomes a problem if the nodes serve the packets on FIFO based or other queue management process as the nodes with faster processing rates would flood the nodes' queue receiving the packets and would result in packet. In this work all the nodes in the network are configured to have same queue length and queue processing rates i.e. it uses IEEE 802.11g's 54Mbps and 2.4GHz bandwidth. While that of the existing DSR that makes use of IEEE 802.11 2.4GHz bandwidth and 4.5Mbps processing per speed rates.

2.1Performance metrics parameters

For performance evaluation of the work implemented, a set of performance metrics are analyzed in terms of throughput, delay, overhead, packet loss and PDR. To implement each of these performance metrics parameters they have a set of basic design equations which are as follows:

- Throughput: Throughput is nothing but the efficiency. It specifies the data rate achieved in unit time by successfully receiving a certain number of packets. Throughput is usually measured is Kbps. The formula used to calculate the throughput is: Throughput= (received packet size (stop time- start time)) *(8/1000)
- End-to-end delay: The average time taken for a packet to reach the destination considering the additional delays caused along the line due to buffering, re transmissions, queuing and that of the packet itself is called as end-to-end delay. The end-to-end delay is calculated using a set of formulas that are run in loops to cover the entire simulation duration:
  Delay[k] = Received packet delay[k] - sending packet delay[k]
  Total Delay = Total Delay + Delay[k]
  Average Delay = Total Delay / Total number of packets
  Where, k= sequence number of the packet.
- Overhead: The routing packets sent per data packet specifies the overhead. The overhead is calculated using the formula:
  Overhead= (Total number of routing packets received/Total number of data packets received)
- Packet loss: As the name suggests, it is the total number of packets lost in a given transmission. Packet loss is obtained by using the formula:
  Packet loss = (Total number of packets sent by the sender node - Total number of packets sent by the receiver node)
- PDR (Packet Delivery Ratio): The calculation of PDR is the ratio of total packets received to that of transmitted between the destination and the source respectively. The PDR formula is:
  PDR= (Total number of packets received by the receiver node / Total number of packets sent by the sender node)
- Energy consumption: As already the parameters related to energy model mentioned in the above chapter let's see the energy consumption as an individual node and as the whole network. Input is the initial energy of the node and output is the energy consumed. For every transmission a node uses energy that may be in sending or accepting the packets. This sets up new value to the initial energy parameter every time the node communicates and gradually decreases. Hence, it is validated to say it's the difference in the start and current energy levels. An initial energy of zero only means that the node is incapable of communication. If an energy level of a node reaches zero, it cannot receive or transmit anymore packets.

3. METHODOLOGY

The flow diagram in the Fig 3 below shows the implementation of distributed cache updating for DSR protocol. The steps in achieving cache updating at the nodes end can be briefly explained as follows.

- The topology module script describes the functionality in building a topology.
- The wireless mobile nodes are set in the area using the Node deployment algorithm.
- Next the routing table algorithm is implemented, in which a neighbor is tracked by the node whenever the routing is initiated.
- Whenever a data packet needs to be sent to destination DSR protocol is invoked.
- The route for the delivery of packet to the destination is checked in the dynamic cache.
- If route is available then send the packet else update this information to the nodes and initiate DSR protocol again by root discovery and by updating the route cache and send the data.

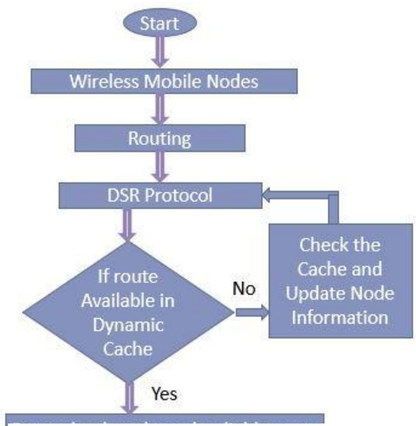

Fig -3: Flow diagram showing Cache updating for DSR protocol

3.1Detailed Overview

A detailed overview over the points discussed in the above sectopn are explained here.

- The Topology description: Scripts to build topology, its description and functionality that involves constructing the wireless network topology and setting up nodes to work with more than one channel. This module flow is as follows:
  - Laying the Network Topology: Tasks like domain settings, node cofiguration, and creating topology.
  - Fixing the bandwidth and threshold: Bandwidth and geographic co-ordinates are set for each and every node in the network.
  - Neighbor identification: Euclidean distance a mathematical concept is used to locate the neighbors w.r.t every node is used.
  - Single or multi-hop data transmission: specifies the sender node and the receiver node for a particular data packet. The time interval and total number of data packets to be sent are also specified.
  - The simulation duration: The simulation begin and end time specified in NS allows to view the network's transactions through NAM which would take fraction of time to complete otherwise.
- The Wireless Mobile Nodes: The node deployment algorithm in the Fig 4 gives the flow of how the nodes are set in the network in a given area. The total nodes and the nodes euclidean distance are the inputs. List of node ID's and their positions (w.r.t neighbor nodes) are the outputs.
- Routing: A prominent task for reliable communication in dynamic networks facing challenges like mobility, unpredictable topology and link breakage. Also power being an important aspect as nodes are battery operated, a disrupt or a drained battery will cause the network to divide creating partitions. The choice of an efficient routing plays an important role in forwarding packets over a network. Any routing protocol's task is to find paths and maintain those paths and continuously adapt and find solutions to changing topology and link failure. The routing is controlled by the routing table algorithm as show in the flow diagram Fig 5 below. Routing tables are created for every node which carry the network topology information . Every node in the network follow the flow to set up the routing table generating the ID's and node position in the form of a map. This information is updated in the routing table in the same manner.

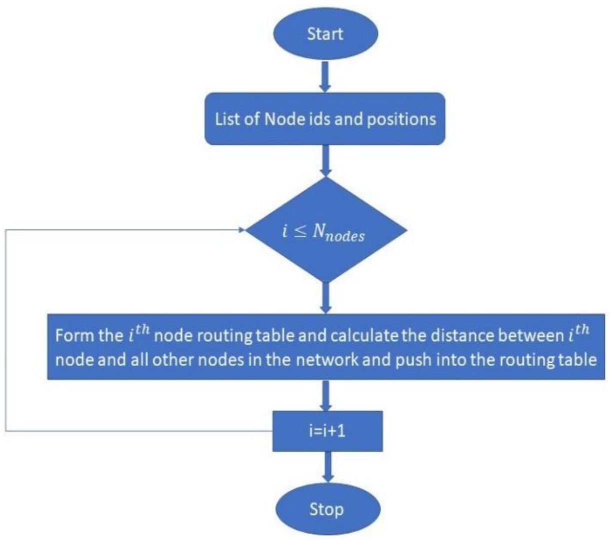

Fig 4: Flow diagram for node deployment algorithm

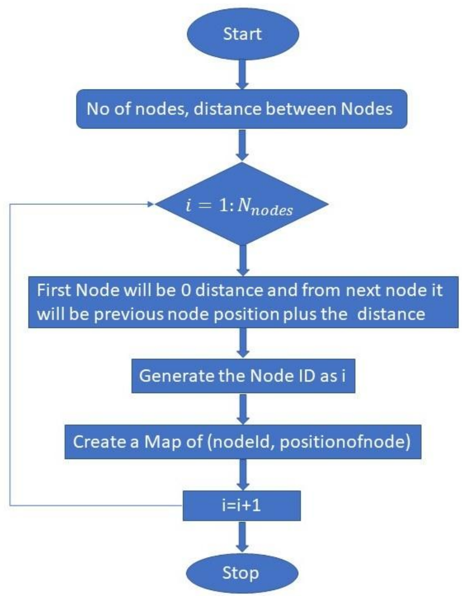

Fig 5: Flow diagram for routing table formation

3.2Implementing the Distributed Cache Updating for Dynamic Source Routing Protocol

Table -1: General simulation parameters

Parameters	Values
Simulation tool	Network Simulator 2
Simulation area	800*800sq mts
Channel type	Wireless channel
Radio propagation model	TwoRayGround

MAC layer protocol	802.11
Traffic type	CBR
Queue type (IFQ)	CMUPriQueue (Priority queue)
Queue size	50
Antenna type	Omnidirectional
Traffic interval	1ms
Packet size	512 bytes/packet
Routing protocol	DSR and E-DSR
Simulation duration	100sec
Node maximum	20 m/sec
Mobility model	Random model
Number of nodes	10, 20, 30, 40 and 50
Transmission range	250 meters
Sensing range	250 meters
Initial power	100J
Transmission power	10mW
Reception power	1mW
Sleep power	0.1mW
Transition power	0.2mW

The performance comparison among cache updating mechanism for existing DSR protocol and between the proposed cache updating for DSR protocol is conducted using NS-2 tool. The enhanced queue processing mechanism for updating cache is implemented for both the protocols. The simulation is carried in the same environment for both existing DSR and enhanced DSR to see the comparison of performance on throughput, end-to-end delay, overhead, PDR, packet loss and energy consumption of the network. The Table 1 below gives the details of the simulation environment and the node parameters. The NS-2 tool is used to implement the given network topology in an area of 800*800 square meters, TwoRayGround propagation model in a wireless channel with CBR type of traffic with an interval of 1ms with the size of each packet being 512 bytes, the random mobility model is used to configure node's mobility behavior having a constant velocity of 20 m/sec. the interface queue type being the PriQueue model and other node parameters like omnidirectional antenna type and transmission range of 250 meters is mentioned. The simulation duration is 100 seconds. It also mentions the energy model specifications for the network nodes. The comparison of performance over throughput, end-to-end delay, overhead, packet delivery ratio, packet loss and energy consumption of the network are carried in a wireless network for different node numbers and constant mobility. Although many previous works used analysis on constant network density by varying mobility, which proved that DSR performs best in networks with high mobility and greater densities. Hence, analyzing the performance of DSR protocol and efficient cache updating in different network densities is proposed in this work. The figures below Fig 6, Fig 7, Fig 8, Fig 9 and Fig 10 show the snaps of network animator window displaying scenario creation with 10, 20, 30, 40 and 50 nodes respectively.

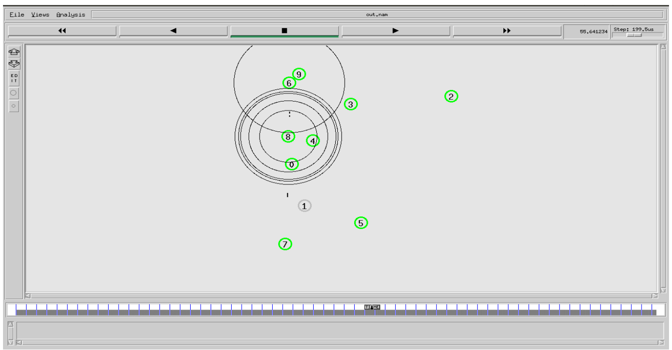

Fig -6: Animator showing scenario creation having 10 nodes

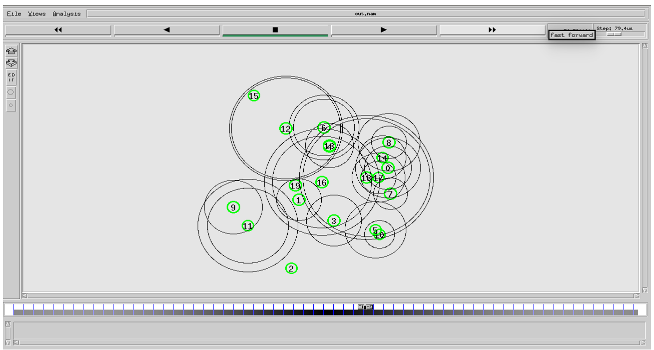

Fig -7: Animator showing scenario creation having 20 nodes

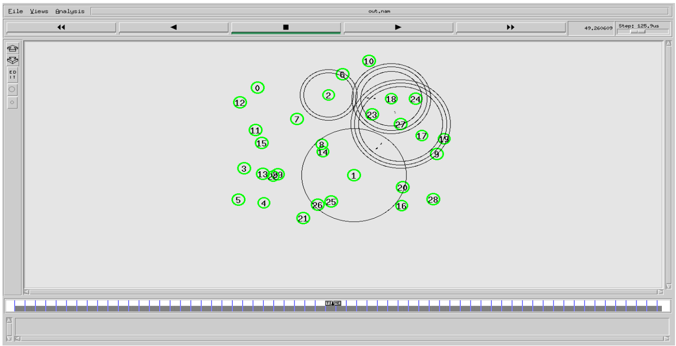

Fig -8: Animator showing scenario creation having 30 nodes

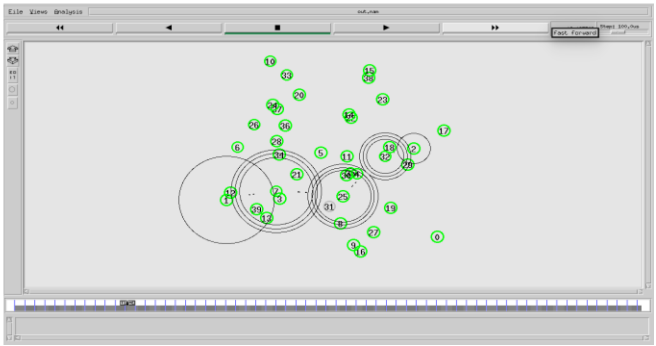

Fig -9: Animator showing scenario creation having 40 nodes

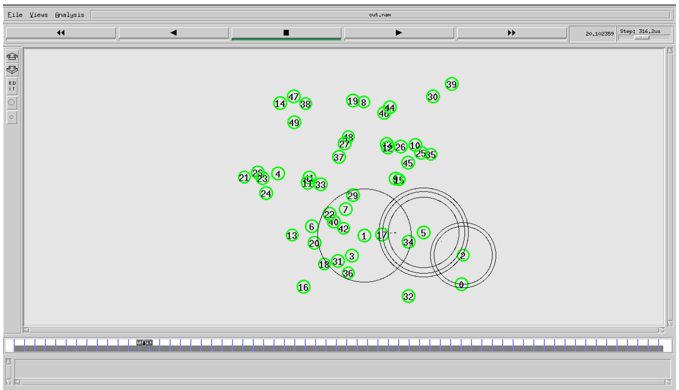

Fig -10: Animator showing scenario creation having 50 nodes

4.0 Results

The tables below Table 2 and Table 3 show the results obtained for distributed cache updating for existing DSR and the proposed DSR protocol for node scenario with 10, 20, 30, 40 and 50 nodes respectively. The performance metrics parameters are throughput, end-to-end delay, Overhead, PDR and overall average energy consumption of the network. These results are obtained for varying node densities with the objective to show the behavior of the DSR protocol at changing network densities with constant velocities and hence allowing to determine the effects over the performance and overall efficiency in different network conditions. The throughput is obtained in Kbps, specifies the total throughput for a particular network scenario for the entire simulation duration, in the same way, end-to-end delay is measured in milliseconds, the overhead measured in bytes, PDR obtained as the ratio of received packets to total packets sent where R gives the total number of packets received and S specify total number of packets sent, F specifies total packets forwarded and the last field specifies the total packets lost, where the fields under PDR are tabulated w.r.t number of packets and not the size of packets.

Table -2: Results obtained for distributed cache updating for existing DSR protocol

Sl No	Number of nodes	Throughput(kbps)	End-to-end delay(msec)	Overhead	PDR (Packet delivery ratio)				
					S:	R:	R/S:	F:	Loss
1	50	9.59	38.699	4.434	260	228	0.8769	368	32
2	40	8.29	249.593	7.934	259	194	0.7490	764	65
3	30	10.45	52.456	4.445	250	247	0.9880	569	3
4	20	10.46	40.412	3.367	261	248	0.9503	399	13
5	10	7.51	4271.995	8.579	252	178	0.7063	969	74

Table -3: Results obtained for distributed cache updating for proposed DSR protocol

Sl No	Number of nodes	Throughput(kbps)	End-to-end delay(msec)	Overhead	PDR (Packet delivery ratio)				
					S:	R:	R/S:	F:	Loss
1	50	10.19	18.577	5.153	261	242	0.9272	443	19
2	40	7.92	48.167	8.293	250	188	0.7520	817	62
3	30	10.81	30.6481	4.549	257	257	0.9698	565	8
4	20	10.35	21.3534	3.020	260	246	0.9462	339	14
5	10	7.29	1737.614	9.757	263	173	0.6578	1117	90

Table -4: Overall average energy consumption

Sl	Number	Existing DSR	Proposed
1	10	771.429	642.857
2	20	632.857	385.741
3	30	871.429	416.286
4	20	1014.29	671.429
5	50	578.751	471.429

The Table 4 tabulates the results obtained for overall average power consumption for all network scenarios over both existing and proposed DSR. The results obtained are in milliwatts (mW). The Chart 1, Chart 2, Chart 3, Chart 4 and Chart 5 below represent the results as graphs which are gathered from same as tabulated in Table 2 and Table 3 for performance metrics parameters like throughput (kbps), end-to-end delay (millisecond), overhead (bytes), PDR (ratio of R/S) and packet loss (number of packets) respectively.

Chart -1: Graphical representation of throughput (Throughput [kbps] vs number of nodes)

Chart -2: Graphical representation of End-to-end delay (Delay [msec] vs number of nodes)

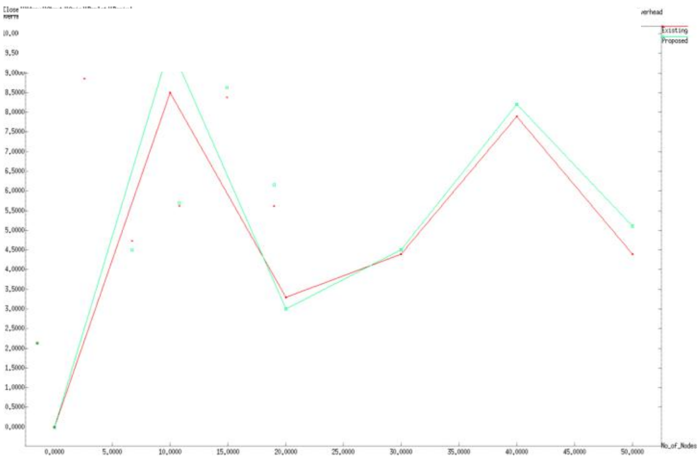

Chart -3: Graphical representation of Overhead (Overhead [bytes] vs number of nodes)

Chart -4: Graphical representation of PDR (PDR [R/S <= 1] vs number of nodes)

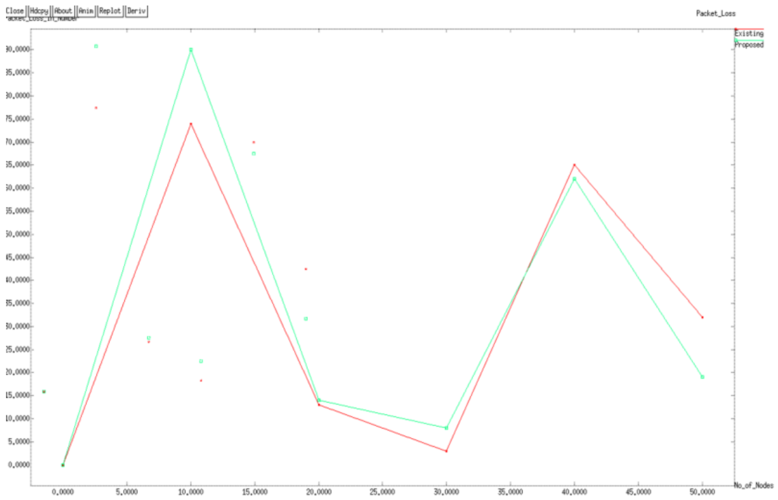

Chart -5: Graphical representation of Packet loss (Packet loss [number of packets lost] vs number of nodes)

The graphs are plotted with the performance metrics on the y axis that are the results obtained vs for the network scenarios with varying number of nodes thus making it possible to analyses the results graphically The points on the graphs show the results obtained to the above-mentioned performance metrics for a given network topology. Two lines of different colors are used to plot the graph where the line in red color represents cache updating for existing DSR and the line with green color represents cache updating for proposed DSR.

4.1 Performance comparison

Performance comparison is carried out in order to showcase the developments achieved based on the analysis and implementation of the work in hand over the existing work. Such an analysis brings out the complete understanding over the fields that have achieved improvements and over the fields where the work needs improvements. A similar performance comparison is carried out in the work where it seen the distributed cache updating for DSR is implemented over the existing DSR protocol and the proposed DSR. The Table mentioned below draws out such performance comparisons in terms of percentage enhancement or drawback to the existing DSR. This kind of comparison is also advantageous in drawing out different performance parameters individually such that the areas with drawback are focused to improvement. The Table 5 below shows the performance comparison of distributed cache updating for existing DSR and proposed DSR as percentage deviation showing how less or more the proposed DSR is efficient for each of the performance metrics parameters for all network scenarios to that where distributed cache updating for existing DSR is implemented.

Table 1: Performance comparison of distributed cache updating for existing DSR and proposed DSR

Performance metric Parameters	Percentage evaluation of Proposed DSR to existing DSR in (percentage %)				
	Number of nodes				
	50	40	30	20	10
Throughput	> 5.88	< 4.67	> 3.33	< 1.06	<3.01
Delay	< 51.9962	< 80.7018	< 41.5737	< 47.1607	< 59.3255
Packet Loss	< 40.62	< 4.61	> 62.5	> 7.14	> 21.6216
Overhead	> 16.215	> 4.47	> 2.33	< 10.30	> 13.73
PDR	> 5.4249	> 0.398	< 1.876	< 0.718	< 7.37
Average Energy Consumption	< 18.18	< 48.52	< 70.69	< 40.67	< 20.43

5.0 Conclusion

The wireless nodes in the MANETs face the issue of cache updating due to the changing network parameters causing link breakage. For efficient communication between the source and destination or the intermediate nodes requires a reliable route for packet transmission. Caching of routes in the node's queue will quicken the process of route discovery and forwarding packets. But, in the event of link failure, for avoiding caching of stale routes the cache updating mechanism deletes the stale entries making way for fresh entries as and when the routes are discovered. For this a priority-based queue processing mechanism is used in the work for cache updating. The Throughput at more density is better than when the nodes are less present in the network. The proposed work outperforms the cache updating for existing DSR with overall decrease in end-to-end delay of 60.10%, and 26.13% of reduction in average energy consumption, having negligible increase in packet loss of 0.5% and PDR almost the same. 0.56% reduction in throughput and 6.99% increase in overhead. At large network density the proposed DSR outperforms the Existing DSR. With reducing density, a trade-off can be observed between throughput and overhead as DSR protocol performs better at greater densities and high mobility.

REFERENCES

[1] X. Yu, "Distributed cache updating for the dynamic source routing protocol," in IEEE Transactions on Mobile Computing, vol. 5, no. 6, pp. 609-626, June 2006

[2] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," in The Kluwer International Series in Engineering and Computer Science, Springer US, 1996, pp. 153-181.

[3] S. Das, R. Castaneda, J. Yan, and R. Sengupta, "Comparative performance evaluation of routing protocols for mobile, ad hoc networks," in Proceedings 7th International Conference on Computer Communications and Networks (Cat. No.98EX226), IEEE Comput. Soc, 1998.

[4] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," in Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking - MobiCom '98, ACM Press, 1998.

[5] X. Zhang, T. Lu, M. Fan, and W. Wang, "DSR optimization in trusted multihop ad hoc networks adopting IEEE 802.11 technology," in 2011 7th International Conference on Wireless Communications, Networking and Mobile Computing, IEEE, Sep 2011.

[6] A. Ahuja, S. Agarwal, J. Singh, and R. Shorey, "Performance of TCP over different routing protocols in mobile ad-hoc networks," in VTC2000-Spring. 2000 IEEE 51st Vehicular Technology Conference Proceedings (Cat. No.00CH37026), IEEE, 2000.

[7] Y.C. Hu and D. Johnson, "Exploiting congestion information in network and higher layer protocols in multihop wireless ad hoc networks," in 24th International Conference on Distributed Computing Systems, 2004. Proceedings, IEEE, 2004.

[8] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot, "Optimized link state routing protocol for ad hoc networks," in Proceedings. IEEE International Multi Topic Conference, 2001. IEEE INMIC 2001. Technology for the 21st Century., IEEE, 2001.

[9] N. P. Bobade, "Performance evaluation of AODV and DSR on-demand routing protocols with varying MANET size," International Journal of Wireless & Mobile Networks, vol. 4, no. 1, pp. 183/196, Feb 2012.

[10] X. Yu and Z. Kedem, "A distributed adaptive cache update algorithm for the dynamic source routing protocol," in Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies., IEEE, 2005.

[11] Y.C. Hu and D. B. Johnson, "Caching strategies in on-demand routing protocols for wireless ad hoc networks," in Proceedings of the 6th annual international conference on Mobile computing and networking - MobiCom '00, ACM Press, 2000.

[12] K. Han, "Cooperative caching algorithm based on grouping nodes in mobile ad hoc networks," in the 2010 IEEE International Conference on Information and Automation, IEEE, Jun. 2010.

[13] M. Marina and S. Das, "Performance of route caching strategies in dynamic source routing," in Proceedings 21st International Conference on Distributed Computing Systems Workshops, IEEE Comput. Soc, 2001.

[14] X. Yu and Z. Kedem, "Reducing the effects of mobility on TCP by making route caches quickly adapt to topology changes," in 2004 IEEE International Conference on Communications (IEEE Cat. No.04CH37577), IEEE, 2004.

[15] X. Murugant, S. Balaji, P. Sivasankar, and S. Shaumugavel, "Cache based energy efficient strategies in mobile ad hoc networks," in 2005 IEEE International Conference on Personal Wireless Communications, 2005. ICPWC 2005., IEEE, 2005.

[16] K. Fawaz and H. Artail, "A two-layer cache replication scheme for dense mobile ad hoc networks," in 2012 IEEE Global Communications Conference (GLOBECOM), IEEE, Dec 2012.

[17] K. Fall, K. Varadhan, et al., "The ns manual (formerly ns notes and documentation)," The VINT project, vol. 47, pp. 19/231, 2011.

[18] M. Tarique, K. Tepe, and M. Naserian, "Energy saving dynamic source routing for ad hoc wireless networks," in Third International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt'05), IEEE, 2005.

[19] T.C. Huang and C.-C. Chan, "Caching strategies for dynamic source routing in mobile ad hoc networks," in 2007 IEEE Wireless Communications and Networking Conference, IEEE, 2007.

[20] A. Saeed, L. Khan, N. Shah, and H. Ali, "Performance comparison of two anycast based reactive routing protocols for mobile ad hoc networks," in 2009 2nd International Conference on Computer, Control and Communication, IEEE, Feb 2009.

[21] L. K. Malviya and D. Tiwari, "LMP-DSR: Load balanced multipath dynamic source routing protocol for mobile ad-hoc network," in 2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT), IEEE, Jul 2013.

[22] M. Appiah and R. Cudjoe, "A comparative study of reactive and proactive routing protocols on a mobility model in mobile ad hoc network (MANET)," in 2018 International Conference on Smart Computing and Electronic Enterprise (ICSCEE), IEEE, Jul 2018.

[23] W. Ding, R. Xu, B. Xu, C. Xiao, and L. Zhao, "A performance comparison of routing protocols for tramcars in mining industry," in 2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), IEEE, Jul 2019.

[24] S. Chiaravalloti, F. Idzikowski, and L. Budzisz, "Power consumption of wlan network elements," Tech. Univ. Berlin, Tech. Rep. TKN-11-002, 2011.

[25] C. Knight, J. Davidson, and S. Behrens, "Energy options for wireless sensor nodes," Sensors, vol. 8, no. 12, pp. 8037/8066, 2008.

[26] M. G. Calle Torres, "Energy consumption in wireless sensor networks using GSP," Ph.D. dissertation, University of Pittsburgh, 2006.