Agent based Load Balancing for Fault Tolerant Networks

Sahana B[1], Abhay Deshapande [2]
[1] Assistant *Professor, Dept. of ECE,RVCE, Bengaluru 560059, India*
[2]*Associate Professor, Dept. of ECE, RVCE, Bengaluru 560059, India*
(E-mail: [1]sahanab@rvce.edu.in, [2]abhayadeshpande@rvce.edu.in)

Abstract— This paper proposes the Supervised Reinforcement Learning (SRL) based algorithm for balancing the overload conditions in the network by using reward based system. The rewards are allocated to the links by a reward-assigning agent. The Agent collects the traffic load information from the network and assigns reward to each link. The Agent algorithm runs on a dedicated router, which is a part of the network, and it works on the currently implemented routing protocol which updates the link weights based on the rewards assigned periodically. The Epsilon greedy algorithm is chosen as the action selection strategy for implementing the agent. The simulation is carried for a network with 9 nodes and is observed that throughput is increased from 2746 Kbps to 3141 Kbps when agent algorithm is implemented .Packet Delay is reduced from 0.112ms to 0.033ms and Jitter has high variation without agent implementation whereas it is almost constant with agent implementation. Packet loss is reduced by 2.74 % with agent algorithm.

Keywords— MANET (Mobile Adhoc Network), SRL (Supervised Reinforcement Learning), QOS (Quality of Service)

INTRODUCTION

Load balancing is an effective approach to improve the performance of fault tolerant networks. In the distributed systems need of load balancing arises due to the traffic variation appears as a result of server task allocation, client-server communication, traffic in the physical link between different routers. A technique of forming a fault tolerant network by overcoming the bottleneck issue through load balancing approach is discussed in this paper. High Load Condition in a network is the one where the network continues to give the service at the cost of quality, delay, packet loss and blocking of new connections. In a congested network, the throughput is always compromised. Congestion occurs when bandwidth is insufficient and network data traffic exceeds its limit. The problems on the network due to high load conditions are: packet loss, buffer overflow and jitter. So, to balance the load in different links of the network and to make the network fault tolerant and to reduce the packet loss probability, load balancing is required. Network Load Balancing provides means of connecting multiple servers with a better scalability with enhanced features of availability and manageability. Benefits of Load Balancing are Scalability and Increased availability. Traditional routing algorithm faces the challenge of route faults due to varying conditions in the network environment. One of the major problems on the network is congestion which leads to buffer overflow, increase in latency and packet loss. So, there is a need for modern approaches in routing algorithms which incorporate techniques to make network fault tolerant. In this paper, an agent based SRL algorithm is proposed. In this algorithm, a reward based policy function is proposed to deal with the challenges that may occur in a fault tolerant network due to unwanted congestion. It is observed that QOS parameters such as throughput, packet loss, delay and jitter are improved considerably by making use of an agent algorithm. This paper mainly focuses on distributing the workload across the network and hence to reduce the probability of overloading conditions. It also maximizes throughput, minimize the jitter, delay and the packet loss. The main focus of the work is to automatically recognize the load on complex networks and the links are rewarded by the agent based on the data collected from the network.

The rest of this paper is organized as follows. In section 2, literature survey is discussed. Methodology is provided in section 3. Analysis of simulation result is dealt in section 4. Finally conclusion and future work is dealt in section 4 and 5 respectively.

LITERATURE SURVEY

[1] Different routing algorithms made the adhoc networks to achieve high quality of service (QOS) at different traffic conditions. Quality of service (QOS) doesn't refer to just one layer of the network. It is applied across the protocol stack. [2]Depending on the number of optimal paths derived, routing protocols are categorized into single path and multipath routing. In single path routing only one route is invented whereas in multipath routing, source and destination use multiple routes are used for communication. Multipath routing always provides alternative route whenever there is a link failure. The load-balanced routing protocols developed for MANET [5] [6] improve the performance of the network in terms of packet delivery ratio, minimizing traffic congestion and throughput. These protocols consider different load metrics in obtaining the optimal route. [3] Any network system should be adaptable for component upgrades and system failures. To achieve this and to optimize the performance, systems need to learn from experience. In the network simulation described in simulating a network, if a machine can't complete the next step required by the job, it must search among its neighbors for machines that can complete that step. [4] Proposes a reward based function for geographic routing. In the heterogeneous network architecture, the cluster heads are equipped with transceivers with Ultra wideband (UWB) range of frequencies. Sensor nodes play important role in this type of networks. Each node in heterogeneous network is capable of obtaining its own location along with sink and its neighbors. The nodes in these networks share the energy information and the location continuously. The Markov decision process is used to solve the routing problems with reinforcement learning. In reinforcement learning, the decision at each stage is decided by the agent. The advantage of reinforcement learning is that decisions are taken without global network information. In [7], main issues of fault management have been identified in SDN and affected layers have been classified accordingly. Trade-offs of various approaches and how it suits for different situation has been proposed in this paper. In [8] SDN has a centralized controller architecture which makes the network prone to single point failure and bottle neck

issues are discussed. The paper [9] talks about SDN which has a single point controller, which leads to single point failure and cause a serious issue in cases of production use and for network reliability. This paper [10] defines the requirements for upcoming industrial networks and explains the challenges that need to be faced. This paper [11] presents an algorithm for controller load balancing, based on distributed architecture, in the absence of any centralized component.

METHODOLOGY

SRL based Algorithm

Any SRL model consists of key elements namely position, position matrix, reward, agent and action. In this model a router is surrounded by 'm' neighbouring routers with the load of each link is known to it as shown in figure 1. The optimal routing path between a source and destination is found using certain routing strategy.

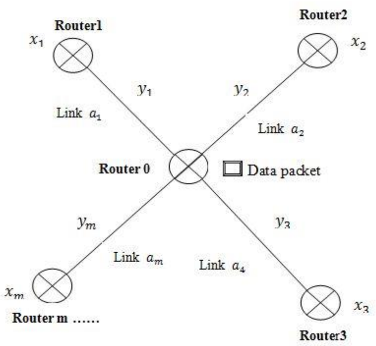

Fig. 1 Data Routing

A. Position: The position refers to the index of the router. The routing direction always gives the indication of destination.
B. Position Matrix: gives the information about the forwarding a packet from one router to the other and it is deterministic.
C. Reward and Agent: The reward process is an important process in learning based algorithms. The reward is -1 before a packet reaches the destination, and it is +1 when the it is forwarded to the destination. Also reward is +1 when data is forwarded to a lighter link and it is -1 when it is forwarded to a link of higher weight. The agent defines the process of rewarding and it makes the policy. Reward is the combination of distance reward and load balancing reward. The distance reward is for the shorter distance. The load balanced reward is for the lower traffic. In general the distance reward is defined as in (1).

$$R_{distance} = p_i - p_j - 1 \quad (1)$$

Where p_i is closest distance between destination address and the i[th] router and $p_j$is the next router having the same distance to the destination. Load balancing reward is depending on the load traffic ratio which is defined as in (2).

$$q_n = \frac{Q_n}{Q_{Th-n}} \quad (2)$$

Where q_n- load traffic ratio, Q_n- absolute load traffic and Q_{Th-n}- maximal load traffic separately in the n[th] network link. In general the total reward is defined as in (3).

$$R = W_1 R_{distance} + W_2\, R_{load\_balance} \quad (3)$$

Where W_1 and W_2 are weight coefficients. The policy function of the agent is defined depending on the strategies of route selection. The flow process is shown in figure 2.

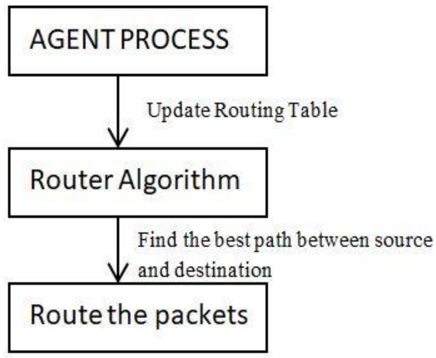

Fig. 2: Flow Chart of action process

There is 'n' number of routers in a network. An agent is software which is implemented in one of the routers of network. This agent is implemented over an existing routing protocol. Agent keeps a track of the link weights and updates them whenever it finds a link is overloaded and periodically. Whenever there are more than two routers sending packets over the same link, traffic increases on the link and hence agent updates the link weight and re-routes the packet through another link. There is an assigned time given in the code after which agent updates the link weights and packets are routed accordingly. The source and destination routers are asked from the user and the shortest path is given between source and destination.
In every 8 seconds the matrix containing the weights is updates. A series of random numbers is generated and modulo of number and numeric 2 is noted. 0 is given as -1 and the complete data is saved in another matrix. This matrix is then added to the old weight matrix and if any number in the matrix is below 1 then 2 is added to the number and the new updated weight matrix is displayed.
The user is again asked for further detecting shortest path between source and route. Based on user input, further cycle continues.

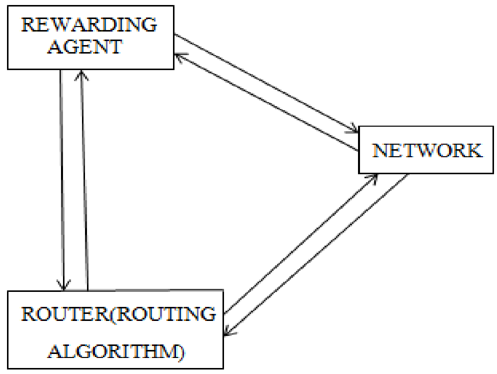

Fig. 3 Scheme of SRL based algorithm

In figure 3, the Rewarding Agent is logically connected to the network and it detects the traffic in all the links, based on that it gives rewards to each link. The rewarding agent is a core router that is physically connected to all the routers which updates all the routing tables periodically. Figure 4 defines the flow of the design process. An existing routing protocol is implemented to find the best path for routing a packet to its destination. Then it is verified that routing was successful or not. Further an agent is implemented in a router over the existing protocol which keeps track of the link load and it periodically updates the link weights. Based on which, later packets are routed where traffic is least.

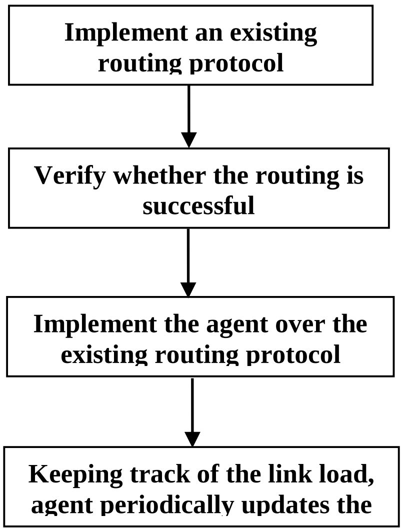

Fig. 4 Flow chart for design process

SIMULATION RESULTS

The simulation of the agent algorithm is carried out in the network simulator v.2.

A. Network Routing Protocol (without agent)

Consider a 9 node network as shown in figure 5. 15000 packets are sent from node 0 to node 3 and node 6 to node 3.

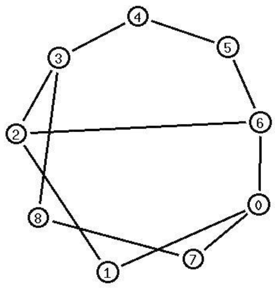

Fig. 5 Network topology (without agent) for NS

The packets are sent from node 0 to node 3. The packet is routed through node 1 and node 2 as shown in figure 6.

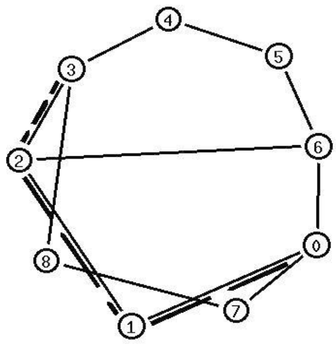

Fig. 6 Packet routing (node 0 to node 3)

The packet transmission starts from node 6 to node 3. It is routed through node 2 as shown in figure 7.

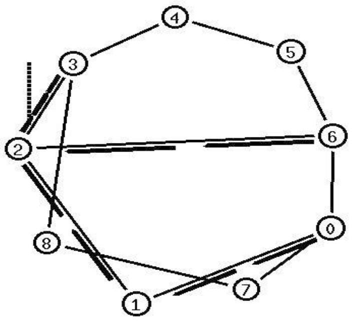

Fig. 7 Packets routing (node 0 to node 3 and node 6 to node 3)

Since many packets are transferred on the link between node 2 and node 3 and this exceeds the link bandwidth, thus results in buffer overflow and packet loss as shown in figure 8

.

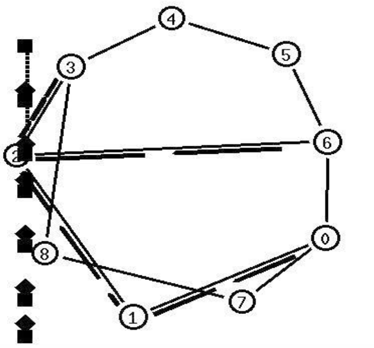

Fig. 8 Packet loss due to overloading

The problems in the network such as link overload, jitter, buffer overflow, and delay cause severe issues in providing the agreed Quality of Service. To overcome the problems discussed, the agent algorithm is implemented which rewards the link based on the traffic present on it. The agent algorithm is a software tool which is running in a router. Further based on the rewards, packets are transmitted through a link which is rewarded highest and weighs least. Hence, jitter, buffer overflow, packet loss and delay problems are overcome.

B. Network Routing Protocol (with agent)

Consider a network topology which consists of 9 nodes as shown in the figure 9. 15000 packets are sent from node 0 to node 3 and node 6 to node 3.

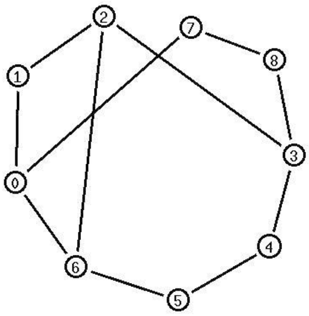

Fig. 9 Network topology (with agent) for Network Simulator

The transmission of packets takes place from node 6 to node 3. It is done through node 2 as shown in figure 10. Since many packets are transferred on the link between node 2 and node 3 and this exceeds the link bandwidth, thus results in buffer overflow and packet loss as shown in figure 10.

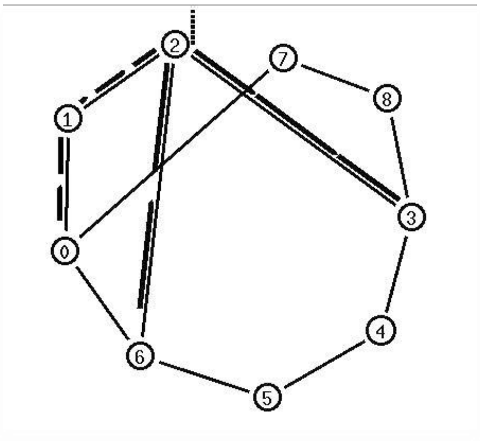

Fig. 10 Packets queued due to overloading

The agent algorithm is implemented which rewards the link based on the traffic present on it. A negative reward to the link between node 2 and node 3 as in fig 11.

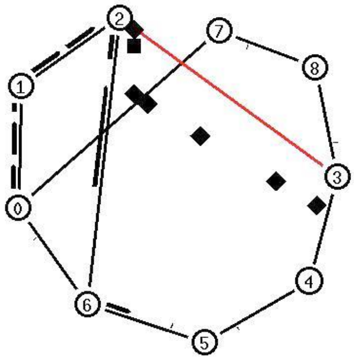

Fig. 11 Packet loss due to overloading

Hence, the routing algorithm starts transmitting the packets through different links as shown in figure 12. Later, agent senses that link between node 2 and node 3 has less traffic and gives positive reward and this advice the routing algorithm to transmit the packet through this link as shown in figure 13.

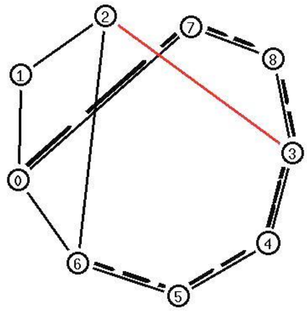

Fig. 12 Packet routing (reduced overload)

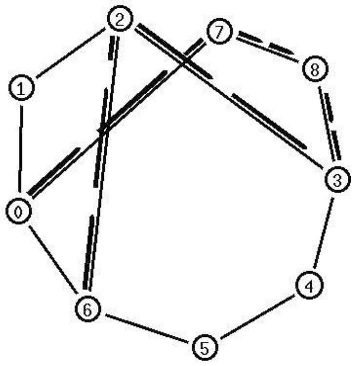

Fig.13 Optimal packet routing

Hence, the problems discussed are overcome by implementation of agent algorithm in the network. The simulation of the algorithm is performed using Network Simulator v.2 The different metrics considered for comparing the implementation of routing protocol with and without agent algorithm

i. Delay
ii. Jitter
iii. Packet Loss
iv. Throughput

The delay is measured in terms of second which specifies the time taken by a bit to travel from source to destination. Delay is introduced in the network due to buffering of packets which arises due to overloading conditions. The agent implementation reduces the overloading conditions which thus reduces the buffering in turn reduces the delay in the network.

Delay in the Network without the implementation of agent algorithm as shown in figure 14.

(i)Delay

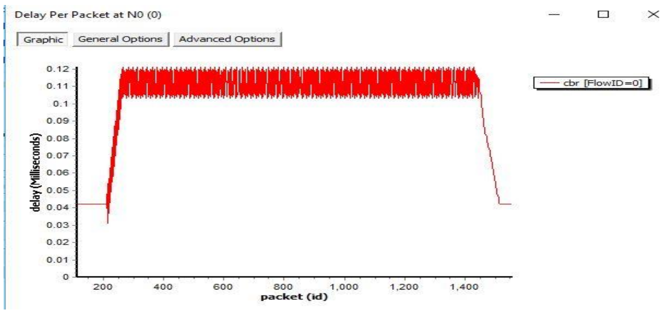

Fig. 14 Output for delay (without agent)

Delay in the Network with the implementation of agent algorithm as shown in figure 15.

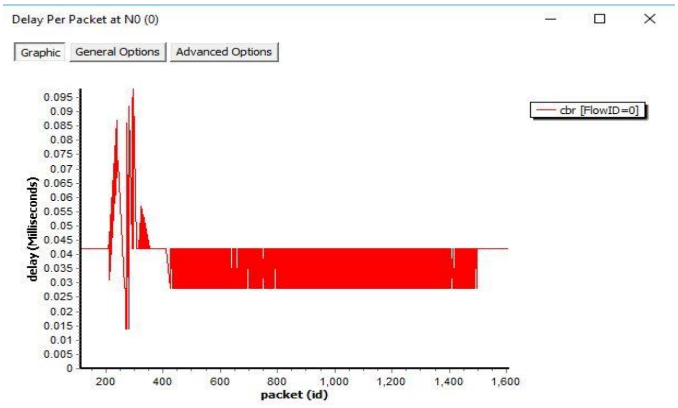

Fig. 15 Output for delay (with agent)

(ii) Jitter

Jitter is the measure of variation of different delays packets arrived at the destination. Though the packets at the source are sent out at time intervals with fixed time gap, at the receiver a variation is found in the time gaps due to network features like congestion, poor queuing approaches or configuration issues. Thus, by implementing agent algorithm which reduces network congestion, the jitter can be minimized. To compare the jitter in the network, the simulation results from the Network Simulator v.2 are attached herewith. Jitter of packets without the implementation of agent algorithm as shown in figure 16.

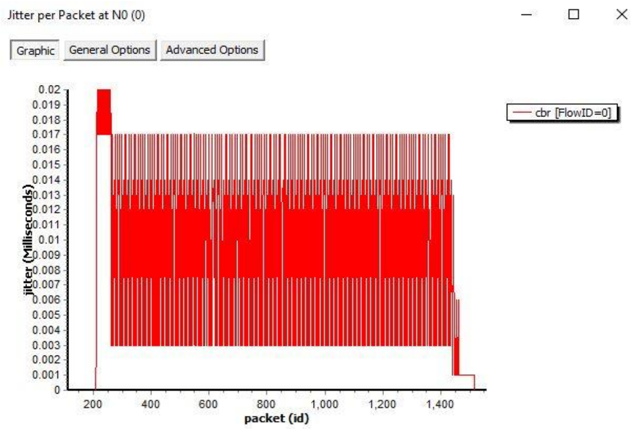

Fig. 16 Output for jitter (without agent)

Jitter of packets with the implementation of agent algorithm as shown in figure 17.

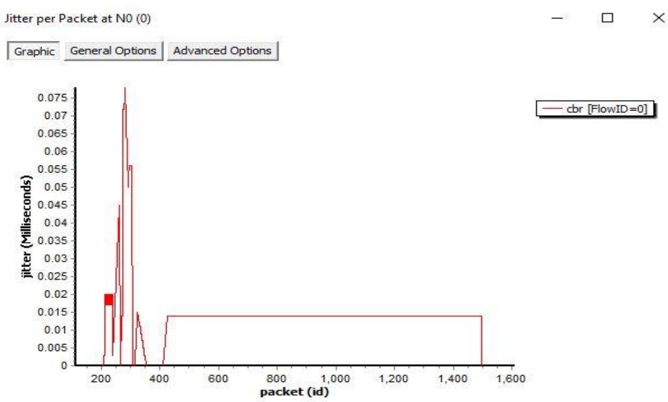

Fig. 17 Output for jitter (with agent)

(iii)Packet Loss

When a packet doesn't reach the destination, it is called as packet loss. Packet loss is mainly due to congestion in the network. The difference between the number of packets sent and the number of packets received gives the measure of packet loss.

To compare the packet loss in the network, the simulation results from the Network Simulator v.2 are attached.

Packets lost - without the implementation of agent algorithm as shown in figure 18 and 19.

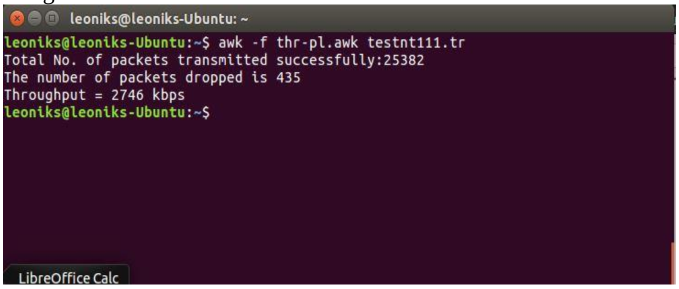

Fig. 18 Output for packet loss (without agent)

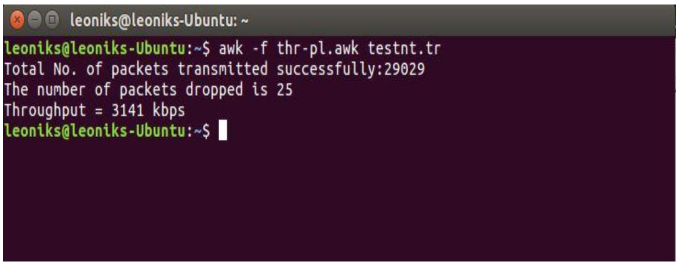

Fig. 19 Output for packet loss (with agent)

(iv) Throughput

Throughput is the measure of successful transmission in a given period which is measured in bits per second. The throughput, which depends on successful transmission of packets, in turn depends on the network congestion. Thus, it can be maximized by implementation of the agent algorithm.

To compare the throughput in the network, the simulation results from the Network Simulator v.2 are attached herewith.

Throughput - without the implementation of agent algorithm is as shown in figure 20.

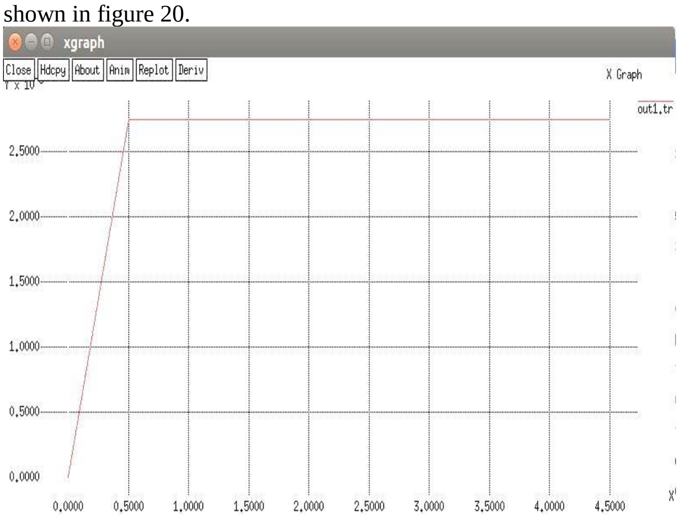

Fig. 20 Output for throughput (without agent)

Throughput - with the implementation of agent algorithm as shown in figure 21

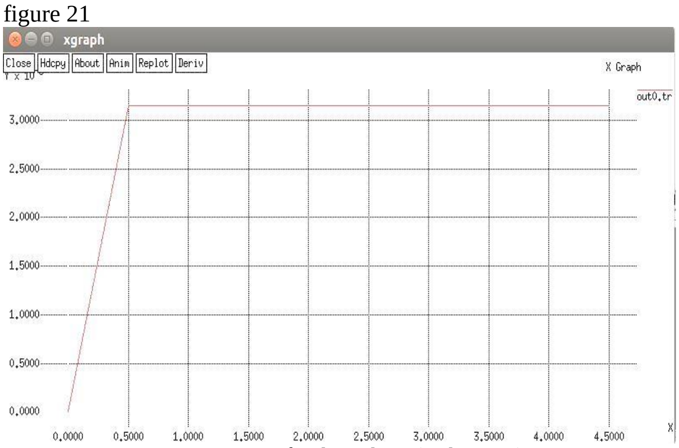

Fig. 21 Output for throughput (with Agent)

Table 1Table of Network metrics obtained from Network Simulator

Parameters	**Value (Without Agent)**	**Value (With Agent)**
Throughput	2746 Kbps	3141 Kbps
Delay	0.112 ms	0.033 ms
Jitter	High Variation	Almost constant
Packet Loss	435 packets	25 packets

The comparison of the quantitative improvement in the network metrics is shown in table 1.

i. Throughput is increased when agent algorithm is implemented from 2746 Kbps to 3141 Kbps i.e. 15% improvement.
ii. Packet Delay is reduced from 0.112ms to 0.033ms i.e. 71% improvement which successfully accomplishes the objective.
iii. Jitter has high variation without agent implementation whereas it is almost constant with agent implementation.
iv. Packet loss is reduced from 435 packets to 25 packets.

CONCLUSION

In this paper, in order to reduce congestion in the network, various algorithms are compared and it is concluded that SRL based algorithm with reward based system is highly efficient for balancing the load in a fault tolerant network. The epsilon greedy algorithm was concluded to be the optimal action selection strategy. The agent algorithm is thus implemented using supervised reinforcement learning algorithm with reward based system. The agent algorithm performs all the calculations based on the data received from the network and then gives the reward to the links. The reward based approach of the agent algorithm helps the router to subsequently learn how to forward a data packet to destination efficiently. This leads to improvised efficiency of the router that leads to following developments in the network metrics: Throughput is maximized, Jitter and delay are minimized, Packet loss is minimized, The congestion in the network is controlled to a large extent, Traffic issues are addressed, The system has become fault tolerant, and The delay in data transmission is addressed. The bandwidth is better utilized. Workload is distributed across the network and hence the probability of overloading conditions is reduced.

FUTURE WORK

The agent can be implemented using a prediction based mathematical model. The network will be subdivided into regions and an agent algorithm will be implemented in each region. All the agents in a network will be interlinked and controlled by an agent. This improves efficiency of the agent implementation. The rewards are given based on traffic on the links. In future, rewards can also be awarded based on additional metrics- buffer capacity and processing time of the routers. The network topology and the information about the metrics are received from the network. In future, these metrics can be calculated by the agent and then rewards will be awarded to the links.

REFERENCES

[1] [1] Shrirang Ambaji Kulkarni. G. Raghavendra Rao, Quality of Services Issues for reinforcement Learning Based Routing Algorithm for Ad-hoc Networks, Journal of Computing and Information Technology – volume 20, issue 4, 2012, pp: 247 – 256.

[2] Simranjeet Kaur, Mukesh Kumar, Review on Load Balancing in Mobile Adhoc Networks, International Journal of Advanced Research in Computer Science and Software Engineering, volume 5, issue 4, 2015, pp: 59 – 64.

[3] Shimon Whiteson, Peter Stone, A Top Down approach to autonomic computing using adaptive routing and scheduling, IBM Corporation – Machine Learning in Network Systems, 24 January 2006.

[4] Shaoqiang Dong, Prathima Agrawal, Krishna Sivalingam, Reinforcement Learning based Geographic Routing protocol for UWB Wireless Sensor Network, IEEE Global Telecommunications Conference, 2007. IEEE GLOBECOM ’07, 26 December 2007, pp: 652 – 656.

[5] Maheshwari, G., Gour, M., & Mishra, S. K., Load Balancing Congestion Control Scheme to Improve the Capability of AOMDV Protocol in MANET, International Journal of Advanced Research in Computer Science and Electronics Engineering (IJARCSEE), volume 3, issue 10, 2014, pp. 428-432.

[6] Tarunpreet Bhatia and A . K . Verma, Simulation and Comparative Analysis of Single Path and Multipath Routing Protocol for MANET, Anveshanam- The Journal of Computer Science & Applications, volume 2, issue 1 , 2013.

[7] P. César da Rocha Fonseca and E. Souza Mota, "A Survey on Fault Management in Software-Defined Networks", IEEE Communications Surveys & Tutorials, vol. 19,no. 4, pp. 2284-2321, 2017.

[8] S. Chattopadhyay, N. Sett and S. Nandi, "FLIPPER: Fault-tolerant distributed network management and control", proceedings of Integrated Network and Service Management (IM), 2017International Federation for Information Processing IFIP/IEEE Symposium, Lisbon, Portugal, 2018.

[9] V. Pashkov, A. Shalimov and R. Smeliansky, "Controller failover for SDN enterprise networks", proceedings of Science and Technology Conference (Modern Networking Technologies) (MoNeTeC), 28-29 Oct. 2014, Moscow, Russia.

[10] D. Henneke, L. Wisniewski and L. Wisniewski, "Analysis of realizing a future industrial network by means of Software-Defined Networking (SDN)", proceedings of 2016 IEEE World Conference on Factory Communication Systems (WFCS), 3-6 May 2016, Aveiro, Portugal.

[11] Y. Zhou, M. Zhu, L. Xiao, L. Ruan, W. Duan, D. Li, R. Liu and M. Zhu, "A Load Balancing Strategy of SDN Controller Based on Distributed Decision", proceedings of 2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications, 24-26 Sept. 2014 ,Beijing, China.

Sahana B holds M.Tech degree in Computer Network Engineering from VTU and She is currently pursuing his Ph.D. program in Fault Tolerant systems. Presently she is working as Assistant Professor in the Department of ECE at RVCE, Bangalore, India.

Abhay Deshapande has obtained his Ph.D. from V.T.U Belgaum with specialization in wireless communication. He has got teaching experience of more than 10 years. He has published /presented more than 15 papers in international/national journals /conferences in the field of engineering. Presently he is working as Associate Professor, RVCE, Bangalore