RESEARCHERID
THOMSON REUTERS

[Vijaya, 5(8): August 2018]
DOI- 10.5281/zenodo.1403497

ISSN 2348 – 8034
Impact Factor- 5.070

GLOBAL JOURNAL OF ENGINEERING SCIENCE AND RESEARCHES

AN EFFICIENT ROUTE STACK OPTIMIZATION BASED ON LOAD BALANCING INQUEUING NETWORK

G.K.Vijaya[*1] & Dr. S. Bharathidass[2]

[*1]Research Scholar (Part Time), Dept of Statistics, Periyar EVR College, Tirchirappalli, Tamilnadu

[2]Asst. Professor, Dept of Statistics, Periyar EVR College, Tirchirappalli, Tamilnadu

ABSTRACT

Queuing network is consisting of several interconnected node is network, commonly queuing network have three types, they are opened, closed and mixed queuing network. In our work fully consider an open queuing network that means Mobile Ad Hoc Network (MANET), because MANET is a infrastructure less network so all the nodes are moveable. Also MANET is a gathering of portable wireless nodes framing a brief network without the assistance of any settled infra-structure. They are self-designing and self-sorting out multi hop wireless network, where the network topology changes powerfully. The nodes of the MANET contain constrained assets like memory, transmission capacity and etc. Due to the asset requirements, the movement or load must to be legitimately distributed over the nodes. Load adjusting is the spreading of problems over the nodes in a reasonable way, which implies that the nodes of the network must not be queuingstacked to send more packets. That is, no nodes must be stacked excessively to transmit packets except if condition appears. In this paper consider stack adjusting is accomplished in light of three systems: piping, piping stack and queuing strategies. The piping approach depends on the essential pipelining idea which transmits the packets all the possible path in the node line using the numerous ways found utilizing AODV convention. In this approach, a node keeps up two queues: common queues and priority queue, all the packets transmission happens from priority queue to the standard path. So finally output generated in to network simulator based on its requirement.

Keywords: *- Queuing network, Stack, Load balancing, Pipelining, Stack adjustment, MANET*

I. INTRODUCTION

The pipeline approach is to consider the queuing network concept also each node have load balancing capacity. Every one of this node, aside from the strength and Weakness, requires data from different components in the general framework. The framework checking module has to think about the condition of the assets. The memory and vitality modules need to think about the individual level of, separately, the memory stockpiling and the battery.

A node is mostly made out of a few tasks {τ1,...,τn} and it likewise remains on assets utilized by undertakings to achieve their execution. The term asset can be connected to both programming and equipment components used by the assignments amid their calculation procedure. In a node, the assets are spoken to by sensors. Also, a node requires to be refreshed by any development in its condition, and it ought to have the capacity to reconfigure its conduct whenever.

Load balancing is a crucial part in rush hour gridlock building and mentions the strategy of scattering activity load consistently finished the network. It is a necessary approach with a particular true objective to accomplish the ideal use of network assets and improved network execution. Load balancing has been a run of the mill issue since the rise of disseminated frameworks.

[Vijaya, 5(8): August 2018] ISSN 2348 – 8034
DOI- 10.5281/zenodo.1403497 Impact Factor- 5.070

This methodology can either be static, where it is just connected disconnected before restarting the framework, or dynamic. The dynamic type of reconfiguration can be either manual combined by a client) or programmed (joined by astute programming operators inside the frame. Considered as a consistent framework, the administrations performed by a node are reliably executed by a settled number of programming errands. These assignments are for the most part implemented under, time limitations regarding due dates that ought to be met to maintain a strategic distance from mishaps in some essential circumstances, memory requirements since installing frameworks don't utilize great memory; and 3) vitality imperatives which are in respect to the used batteries.

In MANET, without a scholarly approach for coordinating network activity, the movement load in the network turns out to be unevenly dispersed. This may achieve clog at nearby hotspots, corruption in the execution of the network and severe packet misfortune. Uneven load scattering is generally produced by rough client demands or rough conveyance of nodes, where the last might be the consequence of the impromptu and portability nature of the MANET.

In the accompanying areas pipelining task for accomplishing load balancing among the nodes of the MANET are introduced. Initial, an impromptu on request remove vector routing protocol utilized for finding courses among the nodes are added. Second, a concise presentation about essential pipelined activity is presented. At last the proposed pipelining task is displayed.

II. RELATED WORKS

The load-balancing approach is the standard movement line, the course vitality, and the bounce tally alongside the related weight esteems [1-2]. The weights reliable with these parameters might be settled or adaptable to the network status. By taking these three parameters together, the movement is veered off from high loaded courses towards courses having higher vitality and slightest load [3]. In this approach, a load adjusted routing path is picked among every possible path based on weight esteem processed for each path. On a reasonable path, the higher the weight esteem, the higher is its appropriateness for movement dispersal.

The pipeline of reconfiguration is an answer for the sheltered and achievable utilization of reconfiguration demands [4-5]. Made out of an arrangement of modules, this arrangement takes care of the considerable number of issues beforehand specified. The original module tunes in and acknowledge the reconfiguration asks for that utilization of prepared equipment and programming assets in the framework [6]. It isn't conceivable to accept demands that utilization nonexistent or idle assets.

At that point, its permits required dependability of the framework's conduct by maintaining a strategic distance from great info streams of solicitations from the framework's condition [7-8]. The module acknowledges from the second the tenders that won't prompt memory floods. To recognize from the third the solicitations that won't expend the battery before reloading. The last module manages continuous limitations to permit another framework's arrangement that regards all due dates of new and old undertakings [9]. The use of this arrangement on a node requires adjusting the real programming design by including a middleware.

The ability of the current node does not consider, it just remains on its combination to the pipelined suggestion [10], yet additionally the utilization of an official operator permitting the coordinating of the middleware usefulness independently. Also, contrasting the novel commitments of this paper, the proposed node design preserves every one of the requirements identified with the vitality, memory and ongoing [11-12].

In straight pipeline processor succession of preparing, steps are directly orchestrated that plays out specific capacity over the information stream [13-14]. The consecutive pipelines are utilized as a part of number juggling calculation, guideline execution and memory get to. Nonlinear pipeline additionally called dynamic pipeline plays out a few capacities at particular circumstances. These pipelines incorporate a feed forward or criticism associations [15].

[Vijaya, 5(8): August 2018] ISSN 2348 – 8034
DOI- 10.5281/zenodo.1403497 Impact Factor- 5.070

It is involved around AODV Protocol in Mobile Ad hoc Networks. In this protocol, a weight work centered on path length, movement load, vitality level and freshness of every one course is ascertained and put away in the course reserve [16-17]. A blockage mindful multi-path routing protocol for load balancing has been proposed. Exactly when the source node needs to forward the data packet to the goal, it uses the receptive course disclosure strategy where the few paths are notable using multi-path Dijkstra calculation.

III. MATERIAL AND METHODS

Queuing network construct multipath routing based on the rule execution could be accomplished by more than one path. This routing protocol uses both safeguard path and node disjoint path to discover the courses. The disjoint node path does not have any regular node, aside from the source and the goal. The safeguard path is the briefest path between the sender and receiver that sidesteps at least one node in the real path. The routing protocol that ascertains protection frequent paths limit the course disclosure time, and overhead brought about in path support. In node disjoint path routing the source node adjusts the way disengagements since interchange path to the goal is contained inside this node

.

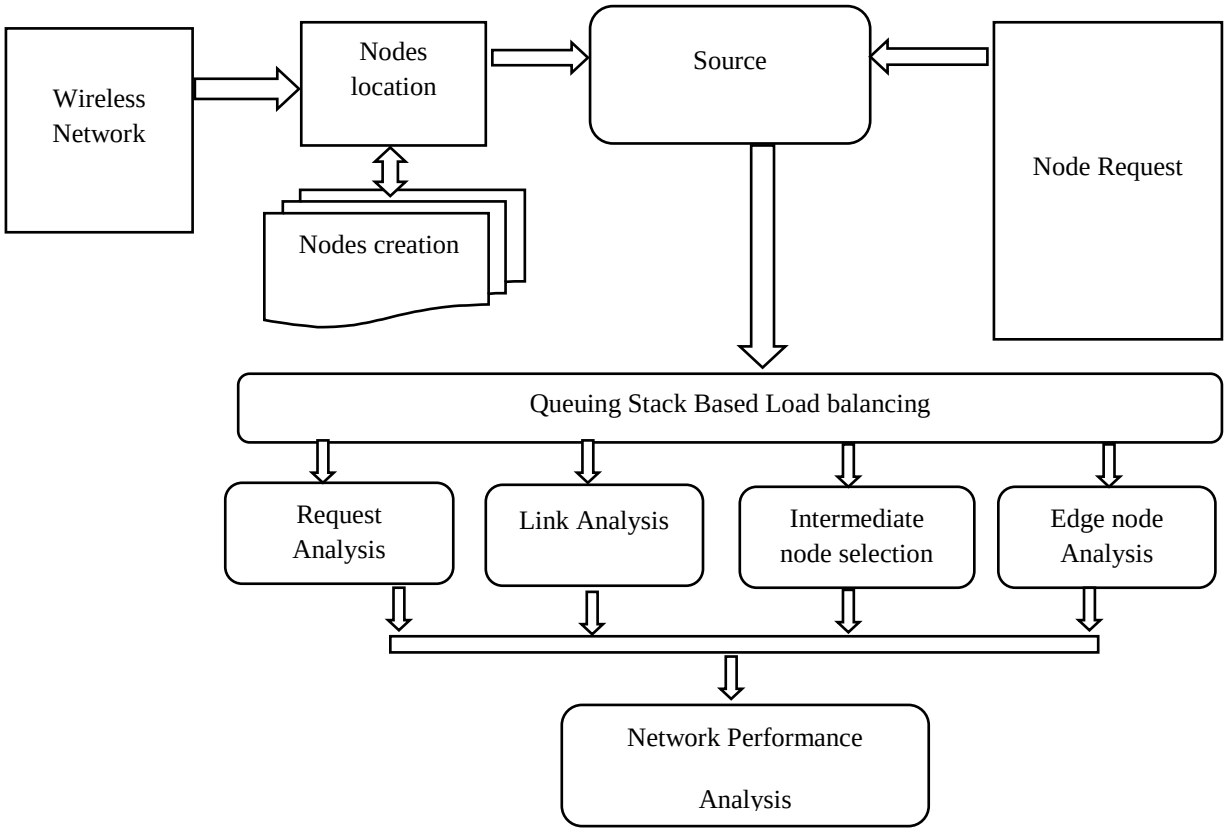


Figure 3.1 Work flow Diagram

Figure 3.1 represents the framework model of the proposed load balancing procedures. Consider a mXn open network location including 'n' nodes and let throughput per unit time be 'm'. Expect that the source 'S' needs to transmit message to the goal 'D'. At first the source starts course revelation system to discover courses to 'D' by communicating RREQ parcels. After accepting RREP bundle, the association with 'D' is set up and the load balancing method is started to decently appropriate load to the goal.

[Vijaya, 5(8): August 2018] ISSN 2348 – 8034
DOI- 10.5281/zenodo.1403497 Impact Factor- 5.070

Queuingnetwork Based Route Request and Route Response Algorithm

Id← node's id, Did← Destination id, RRT← RREQ Table, RDER ← Route Discover Error
Begin
Transmissions RREQ
If Did≠ id
Beforeonward RREQ
Else
Make RREP
Onward RREP
At In-between node check RRT for pathway to source
If found
Accelerative RREP complete min hop neighbor to source
Eliminate entry of the neighbor from RRT
Else
Produce RDER and advancing to downstream neighbor
End If
End If
End

Load Balancing Based Pipelining Under the Queuing Network

Pipelining is an execution method in which a few directions are covered amid the guideline preparing. By and large, the PC pipeline is partitioned into a few phases where each stage executes the segment of the instruction in the grouping. At the point when a period finishes its execution, the outcomes are passed to the following step for additionally preparing and acquire the following guideline from the last stage. Then again in the non-pipelined preparing, the instruction is handled simply after the past direction has been finished it's handling. Figure 3.2 demonstrates a run of the mill structure of pipelined and non-pipelined preparing. For the most part, the guideline in a pipeline is said to be finished when it leaves the last stage. Pipelining does not lessen the season of the direct execution. Instead, it builds the throughput of the guideline. Give us a chance to accept that there is n pipeline arranges each is having measure up to delay, at that point handling the number of instructions every second increment by a factor of n separately.

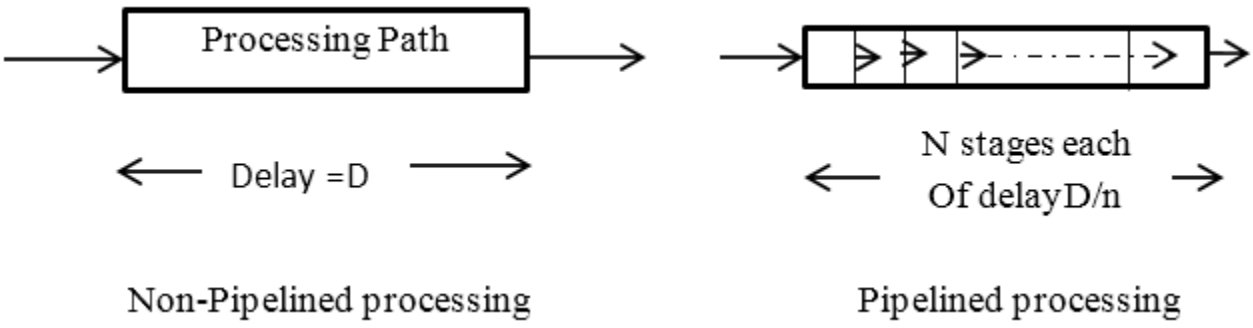


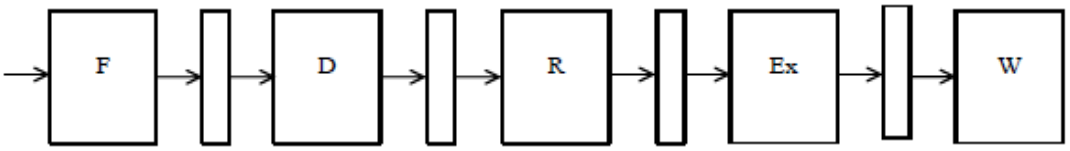
Figure 3.2 Different between pipeline and non-pipeline

On the off chance that the length of the pipeline is adjusted consummately, at that point the time taken by the direction on the pipelined processor is equivalent to

$$T = \frac{Timetakenbytheinstructionontheunpipelinedmachine}{Totalnumberofpipelinestages}$$

[Vijaya, 5(8): August 2018] ISSN 2348 – 8034
DOI- 10.5281/zenodo.1403497 Impact Factor- 5.070

In reality, a guideline pipeline is separated into three to five phases to be specific: Fetch, unravel, Read, execute and compose individually. Figure 3.3 demonstrates the different aspects of the pipeline approach.



F- Fetch, D- Decoder, R-Read, Ex-Execute, W-Write
Figure 3.3 Steps of Pipeline in Manet

Piping is the way of storing data that is being prepared for execution in an arranged manner. The requesting depends on the sender; data compose and need banners. The pipeline stages don't sit tight for the whole procedure to finish its calculation, yet begins handling new data when it wraps up the prior data. In like manner, all the pipeline stages work in parallel bringing about an expanded throughput. The parcels are transmitted in First in First out (FIFO) way. In the general funneling approach the first guideline considered for execution is prepared under factors, memory, bringing memory, stockpiling spot, and transitory stockpiling. In the interim, the second direction is continued sitting tight for execution until the point that the primary guideline completes its execution. Thus the second guideline needs to sit tight for 't' seconds and the estimation of 't' isn't steady for all directions. Under consistent conditions, the execution time stays steady where the estimation of 't' contrasts as the factors and tasks contained in the guideline varies from others which suggests that the last deferral won't be consistent.

Pipeline Operation Algorithm in Network

```
Begin
        PIPE {
    L (n) =active
        Iflively, then check d for all d=1 to x Compute storage S for all x.
        If {S (i) =x or S (i)>x} {
    Supply S (i) with d (i)
    Communicate S (i);
        Else if {S (i) <x}
    S (i) =d (1) X=x-1 i=i+1
        For all x that goes to S (i)
    Communicate d (1)
        Repeat till i=x
        If i==x Free S (i) i=0
        End If
        End for
        End If
        End If
End
```

In the storage room S(i) of the line is equivalent to or more prominent than the number of packets „d" to be transmitted, at that point the packets are put away in the accessible storage room, and the packets are transferred one by one to the goal using the setup path. Then again, if the storage room is not as much as the number of packets 'd' at that point just the initial packet is put away in the accessible storage room. After the transmission of a packet, the following packet available is put away in the line. This procedure is rehashed until all the accessible packets (d) have been effectively transmitted. This calculation is utilized for all the numerous paths found. At long last, the proposed channeling task is as per the following: The estimate for load balancing at first checks for the dynamic course that is to check whether the association with the node has been set up. Once, the association has been set up, the accessibility of information for transmission is verified. At that point, the measure of storage room for the accessible information is registered.

[Vijaya, 5(8): August 2018] ISSN 2348 – 8034
DOI- 10.5281/zenodo.1403497 Impact Factor- 5.070

IV. RESULT AND DISCUSSION:

Ns2 is highly extensible. It not only supports most commonly used IP protocols but also allows the users to extend or implement their protocols. The latest ns2 version supports the four ad hoc routing protocols, including DSR and AODV. It also provides powerful trace functionalities, which are crucial in our thesis since various information need to be logged for analysis. The full source code of ns2 can be downloaded and compiled for multiple platforms such as Windows, and Cygwin. The proposed Load Balancing based on Pipelining under the Queuing Network(LBPQN) algorithm is compared with existing directed acyclic graph (DAG) method. Also here given brief simulation result is given below.

Table 4.1 Parameter Criteria

PARAMETERS	**VALUE**
Version	Ns-all-in-one 2.28
Area	1200m x 1200m
Broadcast Area	250 m
Transfer model	UDP
Data size	512 bytes

4.1 Packet Delivery Ratio

It is defined as the ratio of the number of the number of packets sent by the source to the packets received at the destination.

PDR = Σ Number of packets receive / Σ Number of packets sent *100

Table 4.2 Effect of the packet delivery ratio

Number of Nodes	**DAG in %**	**LBPQN in %**
10	12	19
20	25	36
30	36	49
40	45	72
50	68	86

ISSN 2348 – 8034
Impact Factor- 5.070

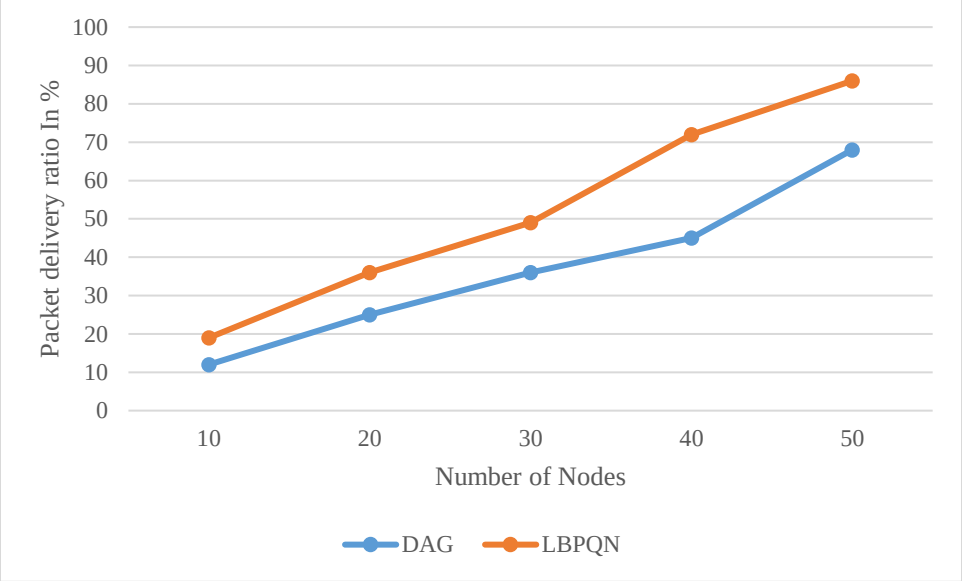


Figure 4.1 Effect of the rate of packet delivery ratio

In above Figure 4.1, Moreover, LBPQNshows the highest packet delivery ratio when compared to others.

4.2 Routing Overhead

This metric represents the ratio of the amount of direction finding related control packet transmissions to a number of data communications.

Routing Overhead= Total number of control packets / Total number of data packet

Table 4.3 Effect of the Routing overhead

Number of Nodes	DAG in %	LBPQNin %
10	26	16
20	46	32
30	66	55
40	83	76
50	94	90

ISSN 2348 – 8034
Impact Factor- 5.070

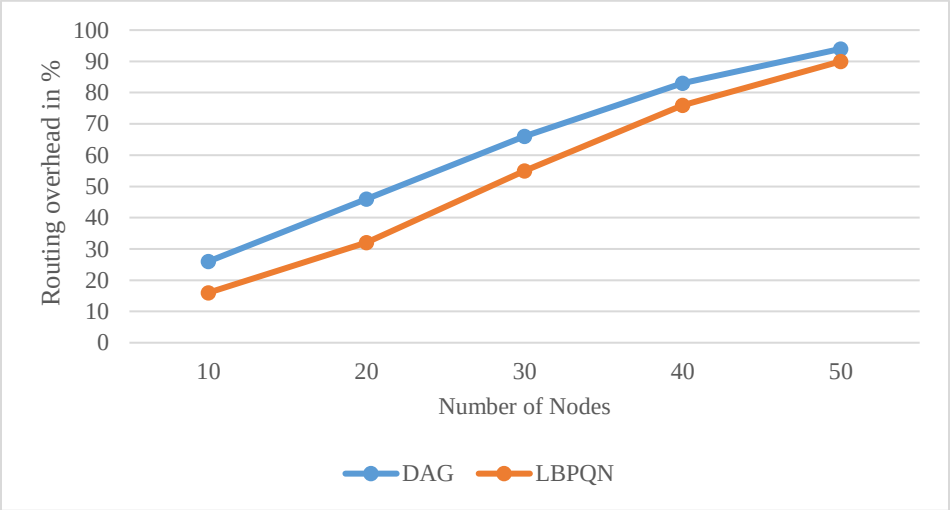


Figure 4.2 Effect of the Routing overhead

In the above illustration 4.2, Moreover, LBPQN has shown lowest routing overhead cost when compared with others.

4.3 End to End Delay

The time is taken by the data packet to reach from source to destination node.

End to End Delay=Arrival time – Sent time / Total number of connections

Average End to End Delay= Total End to End Delay / Total number of packets received.

Table 4.4 Effect of the End to End Delay

Number of Nodes	DAG in %	LBPQN in %
10	19	16
20	29	23
30	49	42
40	67	55
50	82	77

ISSN 2348 – 8034
Impact Factor- 5.070

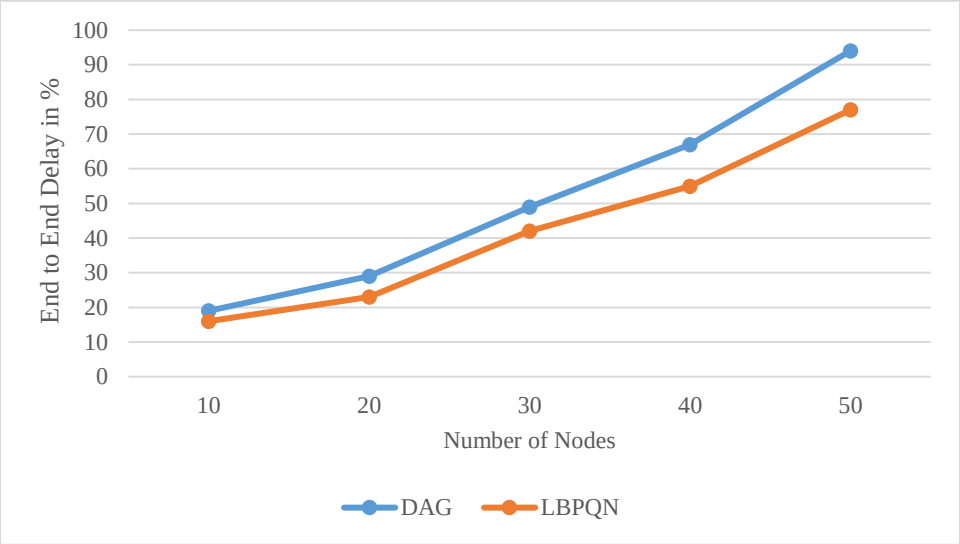


Figure 4.3 Effect of the End to End Delay

In the above figure 4.3, show that LBPQN proves a minimum end to end delay rather than the others.

4.4 Throughput

It is defined as the total amount of data, that the destination receives them from the source which is divided by the time it takes for the destination to get the final packet.

Throughput= Total number of transferred packets / Time taken

Table 4.5 Effect of the Throughput

Number of Nodes	DAG in Packets	LBPQN in Packets
10	11589	12698
20	12589	13657
30	13968	13974
40	14874	14985
50	14980	15120

ISSN 2348 – 8034
Impact Factor- 5.070

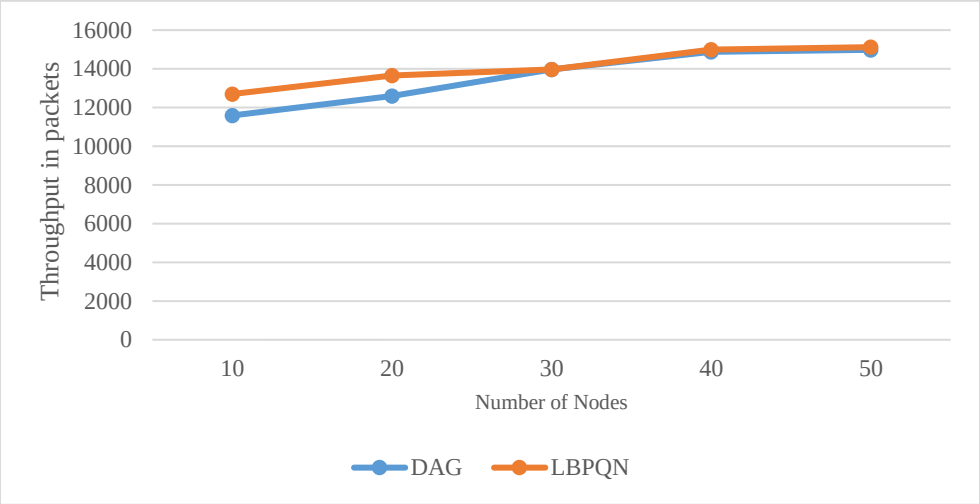


Figure 4.4 Effect of the Throughput

The results are captured in the above Figure.4.4 is LBPQN experimental result shows that it is higher throughput compared with others.

V. CONCLUSION

In this work presented the piping approach in light of queuing network that is utilized to disperse the activity equally over the nodes of the MANET. First, the method utilizes specially appointed on request separate vector routing with multipath augmentation to discover various courses to the goal. At that point funneling approach is actualized to appropriate the load over the network utilizing the multiple paths found. At that point, reproductions are done to assess the execution of the method and are contrasted, and the current piece is stockpiling task. The outcomes demonstrate that the proposed load balancing approach beats the bit storing activity as far as progress rate, a most extreme transmission unit, throughput, and delivery ratio.

REFERENCES

1. X. Wang, Z. W. Li, and W. M. Wonham, "Dynamic multiple-period reconfiguration of real-time scheduling based on timed DES supervisory control," IEEE Trans. Ind. Informat., vol. 12, no. 1, pp. 101–111, Feb. 2016.
2. Husamelddin A. M. Balla, Chen Guang Sheng, "Reliability Enhancement in Cloud Computing Via Optimized Job Scheduling Implementing Reinforcement Learning Algorithm and Queuing Theory" IEEE on Data Intelligence and Security (ICDIS), 2018.
3. Lee, YJ & Riley, GF "A Workload-Based Adaptive Load-Balancing Technique for Mobile Ad Hoc Networks", IEEE Communication Society, WCNC 2005, pp. 2002-2007.
4. M. Gasmi, O. Mosbahi, M. Khalgui, and L. Gomes, "New pipelined-based solutions for optimal reconfigurations of real-time systems," in Proc. Eur. Simulat. Model. Conf., Porto, Portugal, 2014, pp. 361–367.
5. PriyankaGoyal, VintiParmar& Rahul Rish "MANET: Vulnerabilities, Challenges, Attacks, Application",IJCEM International Journal of Computational Engineering & Management, 2011.
6. Sathish G, Lui S., Marco C. Hard real-time communication in bus-based networks. Proceedings of the 25th IEEE International Real-Time Systems Symposium (RTSS 2004), 2004
7. M. Hussin, N. A. W. A. Hamid, and K. A. Kasmiran, "Improving reliability in resource management through adaptive reinforcement learning for distributed systems," J. Parallel Distrib. Comput., vol. 75, pp. 93–100, 2015.
8. SupreetKaur&VarshaKumari, "A Proposed Model for Load Balancing in MANET", International Journal of Computer Applications, 2014.

9. L. Sha, R. Rajkumar, J. Lehoczky, and K. Ramamritham, “Mode change protocols for priority-driven preemptive scheduling,” Real Time Syst., vol. 1, no. 3, pp. 243–264, Dec. 1989.
10. N. K. Samari and g. Michael schneider, “A Queueing Theory-Based Analytic Model of a Distributed Computer Network” IEEE Transactions On Computers, VOL. c-29, NO. 11, 2006.
11. Paola Flocchini, T. Mesa Enrique, “Distributed Minimum Spanning Tree Maintenance for Transient Node Failures”, IEEE transactions on computers, vol. 61, no. 3, 2012.
12. Shaobo Wu, Jianwei Niu, “Delay-Aware Energy Optimization for Flooding in Duty-Cycled Wireless Sensor Networks” IEEE Transactions on Wireless Communications, Vol-21.
13. Willig. “(m, k)-firm deadlines over markovian channels (extended version)”. Telecommunication Networks Group Report Series 05-003, 2005.
14. Wang Guodong, Wang Gang&Zhang ELGR: An Energyefficiency & Load-balanced Geographic Routing Algorithm for Lossy Mobile Ad Hoc Networks‟, Journal of Aeronautics, Volume 23, Issue 3, Pages 334-340, Jun, 2010.
15. X. Liu, W. Tong, X. Zhi, F. ZhiRen, and L. WenZhao, “Performance analysis of cloud computing services considering resources sharing among virtual machines", J. Supercomput., vol. 69, no. 1, pp. 357– 374, 2014.
16. Mohammad Gharbieh, Hesham ElSawy, “Spatiotemporal Stochastic Modeling of IoT Enabled Cellular Networks: Scalability and Stability Analysis” IEEE Transactions on Computers, Vol-11, and No. 11, 2016.
17. Majid Khabbazian,, ‘Bounding Interference in Wireless Ad Hoc Networks With Nodes in Random Position’, IEEE/ACM Transactions on Networking, PP. 1-14, 2015.
18. Murdock 1. R., Koenig 1. R. Open systems avionics network to replace MIL-STD-1553.Aerospace and Electronic Systems Magazine, 2001, 16(8):15-19
19. L. Bai, N. Wu, Z. Li, and M. C. Zhou, “Optimal one-wafer cyclic scheduling and buffer space configuration for single-arm multicluster tools with linear topology,” IEEE Trans. Syst., Man, Cybern., Syst., vol. 46, no. 10, pp. 1456–1467, Oct. 2016.
20. Krishna Pandit, Jens Schmitt, ’Network Calculus meets Queueing Theory - A Simulation Based Approach to Bounded Queues’ IEEE. Vol-45, 2004.