

RHLB: Improved Routing Load Balancing Algorithm Based on Hybrid Policy

Muna Mohammed Jawad

Noor Mohammed Mahdi

Control and Systems Engineering Department, University of Technology, Baghdad, Iraq

[Muna Al-Nayar@yahoo.com](mailto:Muna-Al-Nayar@yahoo.com)

noormohammer57@gmail.com

Submission date:- 14/10/2018

Acceptance date:- 30/12/2018

Publication date:-21/2/2019

Abstract

A network is defined as a set of nodes that are associated with a way to handle and transfer data and messages from source to destination. The congestion in the network occurs when a lot of traffic occurs, leads to delay, packet loss, bandwidth degradation, and high network overhead. Load balancing algorithms have been designed to reduce congestion in the network. Load Balancing is the redistribution of workload between two or more nodes to be executed at the same time. Two policies of load balancing algorithms: static and dynamic load balancing. This paper proposes a load balancing algorithm based on the hybrid (static and dynamic) policy using Network Simulator (version 2). The hybrid policy is used to improve network performance by redistributing the load between overloaded nodes to other nodes that are under loaded when congestion occurs. The simulation results show that the proposed algorithm used performance of the network with regard to throughput, packet delivery ratio, packet loss and the end-to-end delay.

Keywords: Network, Congestion, Load balancing algorithm, Static load balancing, Dynamic load balancing, Hybrid load balancing algorithm.

1. Introduction

A Network comprises of a number of nodes and links for processing packets from source to destination [1]. For connection to happen, packets necessity to be transported from source to destination, a router is a device which assistance in forwarding packets from one node to another in a network or from one network to another [2]. Routing is defined as the operation of moving packets from source to destination using the better path through the network [3].

Congestion in routing is the main problem in the network. If the load on the network (the number of packets transmitted to the network) is larger than the capacity of the network (the number of packets a network can handle). When congestion in the network occurs, the throughput decreases packet delay increases in the network, and the packet drop increases. So in these conditions, if there is no any technique to reduce the congestion in the network can be lead towards the performance collapses and all end to end communication stops [4].

Load balancing is a mechanism utilized to remove congestion by migrating the workload from overloaded node to lightly loaded node to ensure a good overall system performance [3].

1.1 Literature Review

Many works of literature have been addressed load balancing algorithms. **U. Patil and R. Shedge**, used centralized and a decentralized strategy in order to form a hybrid dynamic method. They noticed that the limitation of the system is that in the case of larger number of nodes, communication overhead will be very large. So, to avoid this disadvantage, a design of a simple hybrid dynamic load balancing algorithm was proposed [5]. **M. A. Mehta and D. C. Jinwala**, proposed a hybrid Dynamic Load Balancing (DLB) method which sits between the centralized and decentralized approaches. This technique was partitioned the distributed system nodes into virtual groups, named clusters. A Cluster is useful in improving the scalability and in reducing the communication overhead. In each cluster, a node is designated as a Super Node (SN) in order to define the dynamic threshold value periodically. These clusters are communicated

between them via a Central Master Node (CMN). When any node becomes overloaded, firstly, it searches for a lightly loaded node in its cluster in a decentralized fashion. But, if it does not find a lightly loaded node in the same cluster, its overload will be relocated to a lightly loaded node in another cluster by helping of CMN. The simulation results demonstrate that the proposed algorithm performs faster than the traditional centralized and decentralized approaches [6]. **H. J. Younis et al.**, proposed a hybrid algorithm in cloud environment that takes features of both random and greedy algorithms to improve the performance in heterogeneous cloud computing environment. The proposed algorithm considers the current resource information to achieve the objectives. The simulation results demonstrate that the proposed algorithm add a significant improvements on average response time and average processing time compared with other algorithms. [7]. **R. Saini and A. Bisht**, introduced a Hybrid load balancing algorithm (that focused on two commonly used load balancing techniques i.e., Round-Robin algorithm (Static Load balancing) and Least Load algorithm (Dynamic Load balancing)) which tries to overcome the disadvantages of both the algorithms. These two algorithms are analyzed and compared with proposed algorithm. The advantage of Static load balancing algorithm is more stable but disadvantages, it is not cooperative and also has low resource utilization, on other hand, dynamic load balancing algorithm is more complex and Communication overheads occur more and more as number of packets increase. But dynamic load balancing algorithm is always considered better than static algorithms as they are closely related to real-life scenarios and real-time processing. The simulation results demonstrate that the proposed algorithm performs better in terms data and bandwidth that is more evenly distributed as compared to Round-Robin and Least load algorithms [8]. **S. Subalakshmi and N. Malarvizhi**, proposed an Enhanced hybrid load-balancing algorithm in cloud computing .this proposed contains the features of both Throttled and Equally Spread Current Execution (ESCE) load balancing algorithms. When Comparing the result of the proposed algorithm with the existing algorithms on identified parameters, they found that response time, data processing time and data transfer cost are minimized [9].

2. Load Balancing

Load balancing is a very important topic in our life, since the Internet traffic increases day by day, and rapid development of technologies have emerged. Therefore, there is a high need for availability and rapid response. Goals of load balancing are to Substantial improvement in performance, increase flexibility of the system so as to adapt to the modifications, Stability maintenance of the system, maximizing throughput, minimizing response time and avoid overload [10] [11]. Load balancing can be achieved by statically or dynamically mechanisms [12]:

2.1 Static load balancing mechanism:

Load balancing is performed by prior information with regard to the system. The distribution of workload will be (depending on the performance workload) without consideration to the current case of the node. As the load is allocated to a node, it can't be transferred again to another one.

There are five types of Static load balancing techniques these are Round Robin algorithm, Central Manager Algorithm, Threshold algorithm, randomized algorithm and Least Connection Algorithm [2] [13] [14]. Table 1 lists the main pros and cons of Static Load Balancing Algorithm [2] [15]:

Table 1: The pros and cons of Static Load Balancing Algorithm

Static Load Balancing Algorithm	
Pros	Cons
Performs better in terms of complexity issue.	Compromises with the result as the decision are solely made on statically gathered data.
It is more stable in comparison to dynamic.	The algorithm is non-preemptive.
System overhead is minimized.	It has low resource utilization.
Algorithm is simple and easy to implement.	The task cannot be migrated while execution.
	The overall performance of the system decreased due to load fluctuations.

2.2 Dynamic load balancing mechanism:

Dynamic load balancing algorithm monitors changes on the system workload, and then redistributes of works among nodes at run-time through transferring workload from overloaded node to under loaded node.

There are two types of dynamic load balancing: Central Queue Algorithm and Local Queue Algorithm [13] [14]. Table2 lists the main pros and cons of Dynamic Load Balancing Algorithm [2] [15]:

Table 2: The pros and cons of Dynamic Load Balancing Algorithm

Dynamic Load Balancing Algorithm	
Pros	Cons
No single web server will be overloaded.	Dynamic load balancing algorithm is complex hence not very easy to implement.
Improving the overall performance of the system by migrating the load dynamically.	Communication overheads occur more and more as a number of processes increase.
The algorithm is preemptive.	Needs more data to make the decision.

Each of the mentioned algorithms have its advantages and disadvantages as a result a hybrid load balancing algorithm is suggested to eliminate the drawbacks of dynamic and static load balancing methods, and also they are being used to aggregate the benefits and merits of static and dynamic methods in order to design a new one. This implies that combining the benefits of two or more already existing algorithms either dynamic or static algorithms are able to present a new one with more advantages.

3. The Proposed Hybrid Algorithm For Routing Load Balancing

This section presents the structure of the proposed Routing Hybrid Load Balancing (RHLB) with flowcharts. The RHLB algorithm uses the concepts of both static and dynamic load balancing algorithm, which has two stages: static stage has been used when the load is normal and has no congestion where dynamic stage which is used when the system has a heavy load and congested load. RHLB works with a general structure a network has a number of nodes and links between these nodes such that:

Network Net = (N, E) where:

N= {N_i: i number of nodes in the system}

E= {e_{ab}: there is a link between node (a) and node (b), with specified Bandwidth BW}

$P = \{ \text{all paths from source to a destination stored in source node} \}$

The algorithm begins when there are packets have to be sent from (Source node) to (Destination node)a

4.1 Stage1 (static stage):

Step 1: At the beginning, a node (Source: N_i) chooses the shortest path $\in P$ to send the packets to destination (D) via neighbor node (N_j).

Step 2: If the numbers of packets in e_{ij} queue $< BW_{ij}$ then continue transferring.

Where e_{ij} is a link between nodes N_i & N_j .

And BW_{ij} is the Bandwidth of e_{ij} .

Step 3: Else Calculate the load difference of e_{ij} such that:

- Find all neighbors of node (N_i). Neighbors which is denoted by

Neighbor= $\{N_k: N_k \in N \text{ and } e_{ik} \in E \text{ where } N_k \in \text{path leading to D}\}$

- Calculate the load average of neighbors using the form below:

$$L_{avg} = (L_{ij} + \sum L_{ik} \in N_k) / (BW_{ij} + \sum BW_{ik}) \quad (1)$$

Where L_{ij} is the Load of e_{ij}

L_{ik} is the load of e_{ik} & $N_k \in \text{Neighbor}$

And BW_{ik} is the bandwidth of e_{ik}

- Calculate the load difference using the form below:

$$L_D = (L_{ij} + L_{avg}) \quad (2)$$

- If $L_D < \text{zero}$ then e_{ij} is not overloaded and packets are waiting in queue to be sent to a node (N_j).
- If $L_D \geq \text{zero}$ or queue size of e_{ij} is full then e_{ij} is overloaded and have an excess load so find another node ($N_k \in \text{Neighbor}$) to send the packets to it, go to stage2 (i.e **Stage2 starts**).

4.2 Stage 2 (Dynamic stage)

To design an algorithm for dynamic load balancing, it has to concentrate on three issues they are:

Step 1: Node Selection i.e selection for a new node in order to retransfer the packets to it. This is done by arranging neighbors of node (N_i) in ascending order according to a load of (e_{ik}), and chooses the node (N_k) with the minimum load of (e_{ik}) to be the next node.

Step 2: the Selection of the amount of excess load to be delivered or sent. I.e. choose which packets are eligible to be transferred to other nodes. By arranging or sort, the messages in the queue of (e_{ij}) in descending order according to the number of packets that is still in queue. Choose the largest message (M) with more packets still in the queue.

Step 3: (distribution of the excess load): Send the message (M) to (e_{ik}) queue in order to be sent to the destination node (D) through node N_k

Step4: Repeat the algorithm (stage1 and stage 2) until $N_i = D$ then finish and reach the destination.

Fig.1 shows the flowchart of stage1 (Static stage) and fig.2 shows the flowchart of stage 2 (Dynamic stage) for RHLB algorithm.

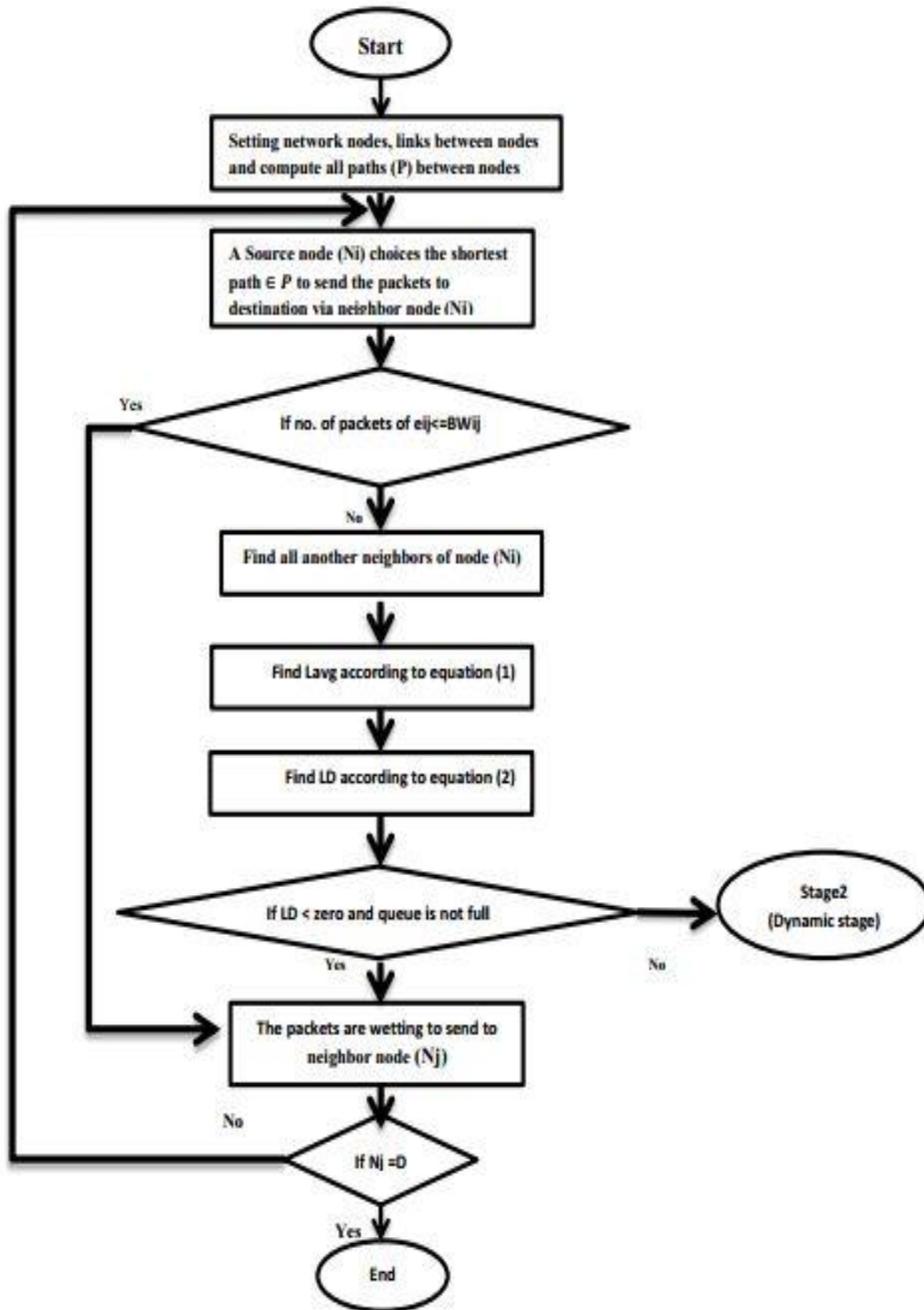


Figure1. The flowchart of static stage for RHLB algorithm

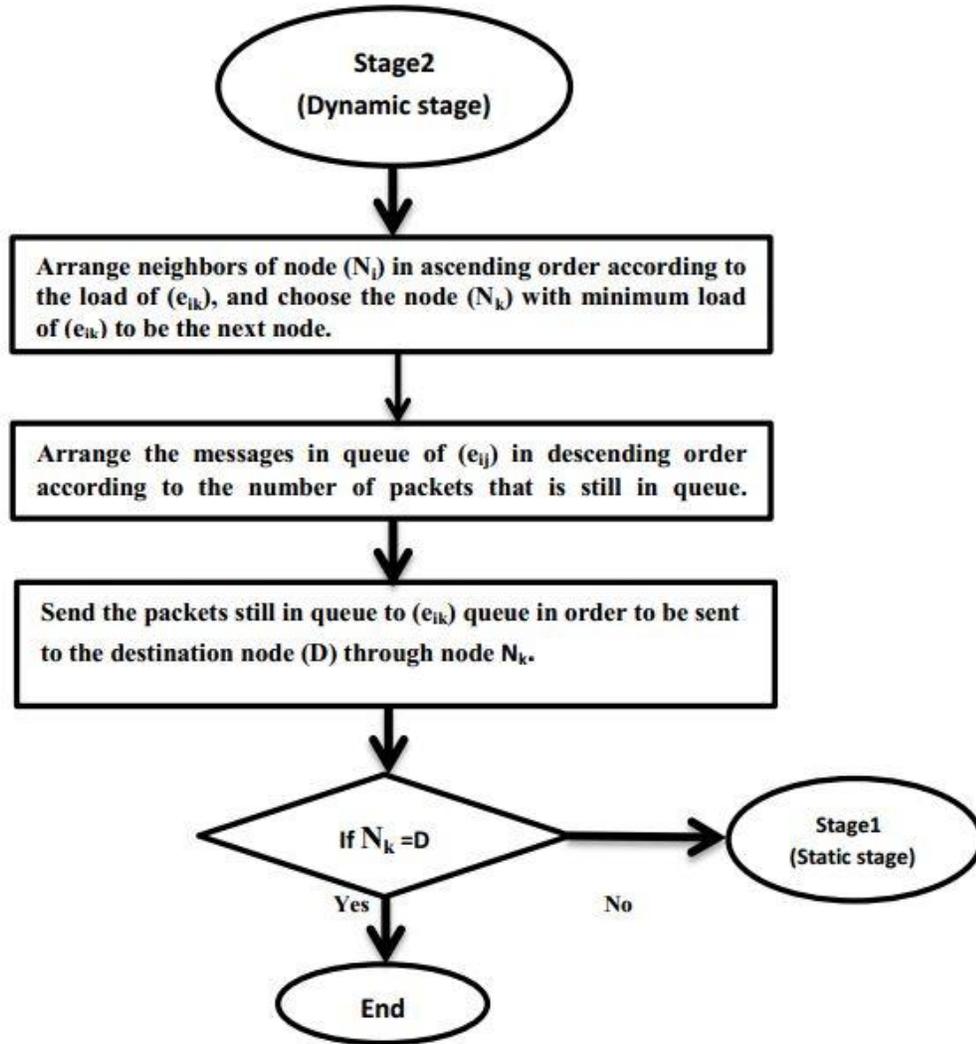


Figure2. Stage 2 (Dynamic stage)

5. Results

This section displays the outcomes of some scenarios to examine the RHLB algorithm. The RHLB algorithm is implemented using Network Simulator (Version 2). The network consists of eight nodes (0-7) and nine links between them as shown in fig.3 Node (0) is the source node and node (7) is the destination node.

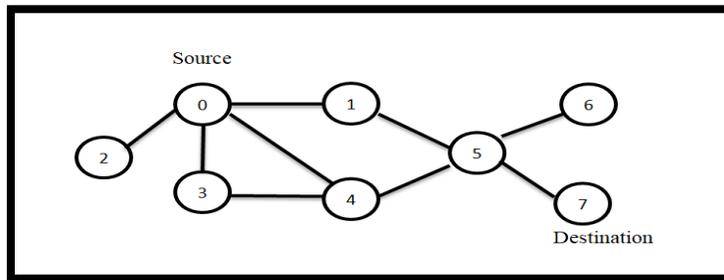


Figure3. Shows the network topology used in RHLB algorithm

The performance of network evaluated using different metrics these are throughput, the end to end delay, packet delivery ratio, and packets loss, which can be shown as below:

- Throughput= Receive size / (Stop Time – Start Time) (3)
- Packet delivery ratio = Number of packet receive / Number of packet send (4)
- Packet loss=Number of packet send – Number of packet receive (5)
- End to end delay=End time– Start time (6)

The performance of the network is calculated by taking the various simulation parameters are given in Table 3.

Table 3: Simulation Settings.

Parameter	Value
Number of nodes	8,10,20,30,40
Simulation Time	10 sec
Routing protocol	TCP
Packet Size	512 byte
Delay	3ms
Queue Type	Drop-tail
Queue size	20

In the first scenario, the performance of the network is calculated by taking the simulation time (0-10) sec and we compare the results of RHLB algorithm with static and dynamic load balancing algorithms with the same environment. RHLB results show that with the start of the algorithm the number of packets sent was equivalent to the static and dynamic algorithms because all nodes are under loaded but when the number of packets increases then nodes will be congested because of the more transfer occurs and most of the nodes will be overloaded so the performance of the network will be enhanced using RHLB algorithm as compared with the performance of the network with using static and dynamic algorithms due to the migration of packet from the overloaded node to the under loaded node. The results of RHLB are showing that performance of throughput and PDR is improved as compared to static and dynamic load balancing algorithm as shown in figure (4 and 5). The packets loss and end to end delay are minimized as compared to the static and dynamic load balancing algorithm as shown in figure (6 and 7). Whereas table 4 summaries these results.

Table4. Summary of the results

Parameter	Results of Static load balancing algorithm	Results of dynamic load balancing algorithm	Results of RHLB algorithm
Packets Sent	7176	8228	10115
Packets Received	4456	7307	10095
Throughput (bit per sec)	5727.14	5237.88	7959.99
Packet Delivery Ratio (PDR)	62.0959 %	88.8065 %	99.9654
Packets loss	2720	921	20
End to End delay (sec)	0.006640	0.003160	0.00130729

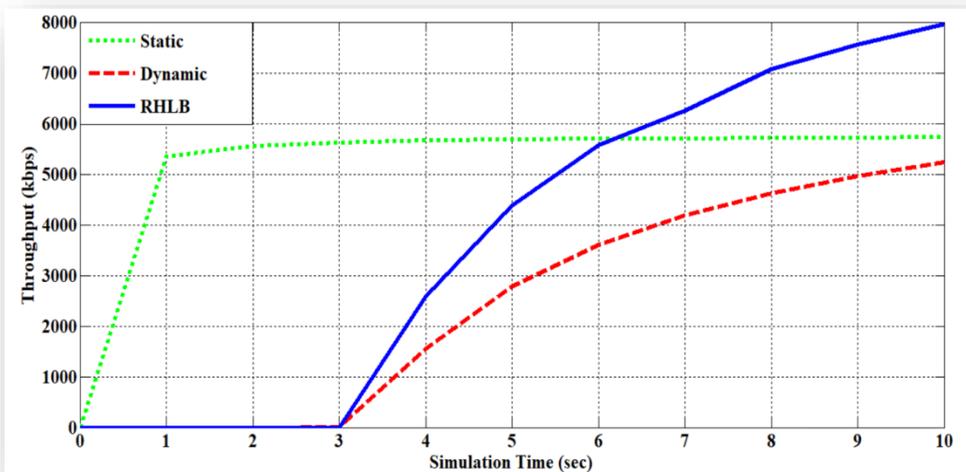


Figure4. Throughput

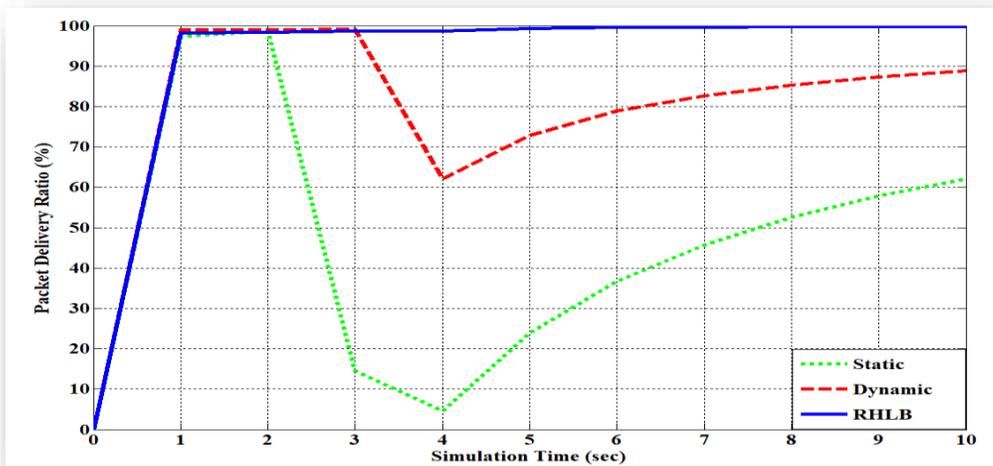


Figure5. Packet Delivery Ratio

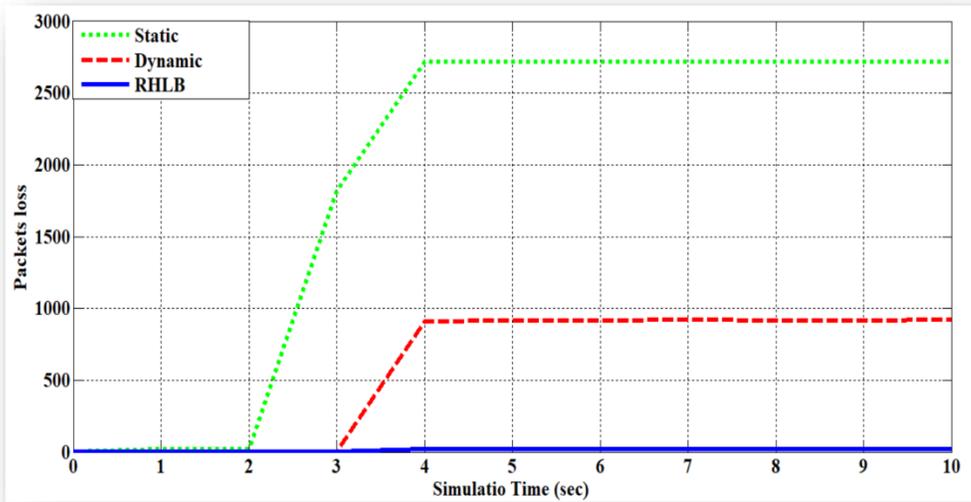


Figure6. Packet Loss

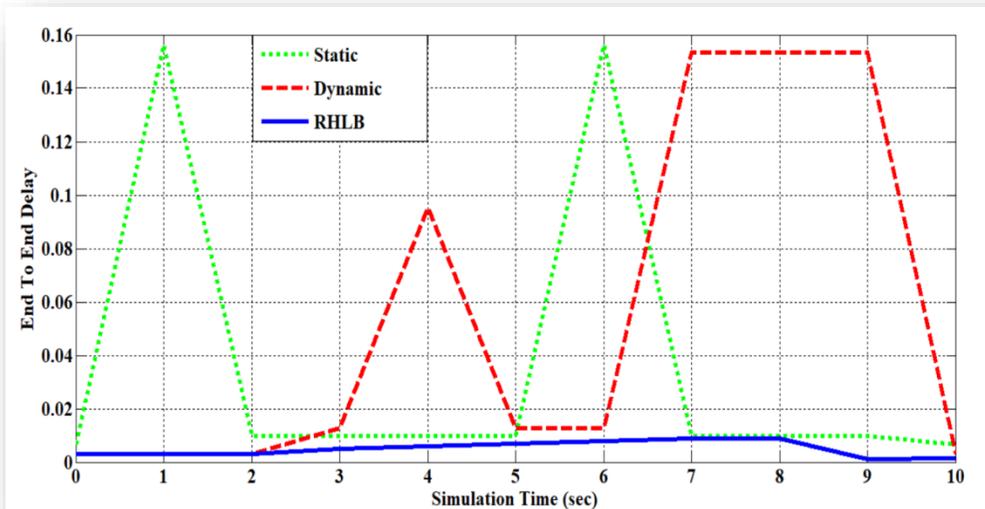


Figure7. End-to-End Delay

In the second scenario, the RHLB is examined against network sizes (increase number of nodes in a network) namely 10 nodes, 20 nodes, 30 nodes, and 40 nodes. The network will be composed of throughput versus the number of nodes for RHLB algorithms shown in Fig. 8, Packet delivery ratio versus the number of nodes shown in Fig. 9, Packet loss versus the number of nodes shown in Fig. 10 and the end to end delay versus the number of nodes shown in Fig.11.

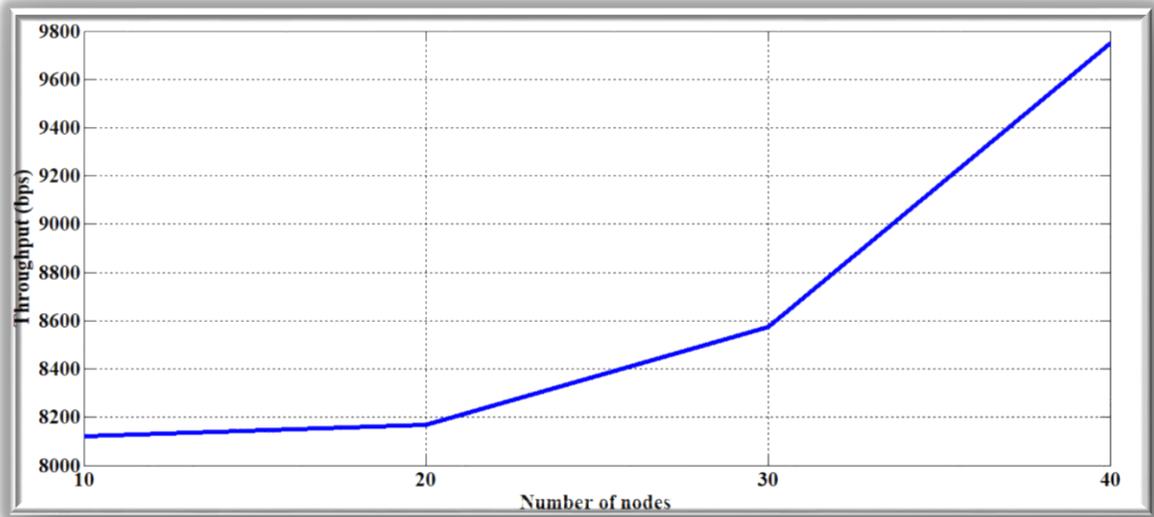


Figure8. Throughput vs. no. of nodes

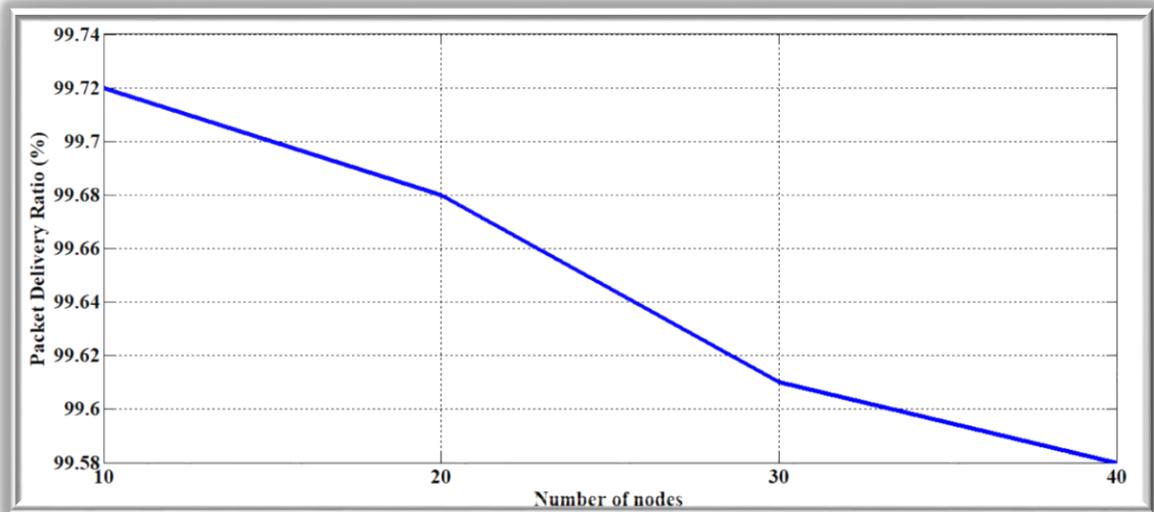


Figure9. Packet delivery ratio vs. no. of nodes

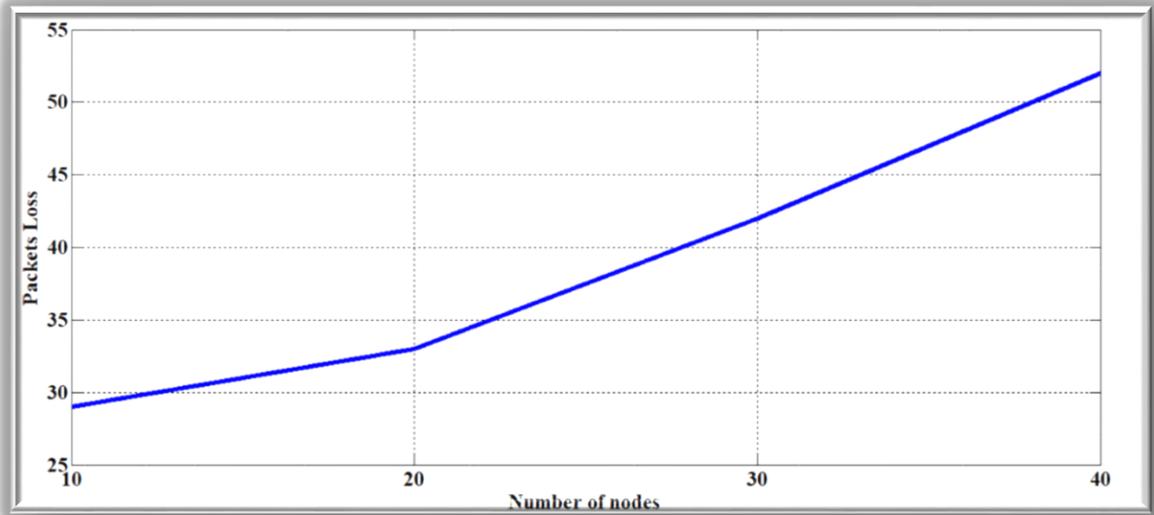


Figure10. Packet loss vs. no. of nodes

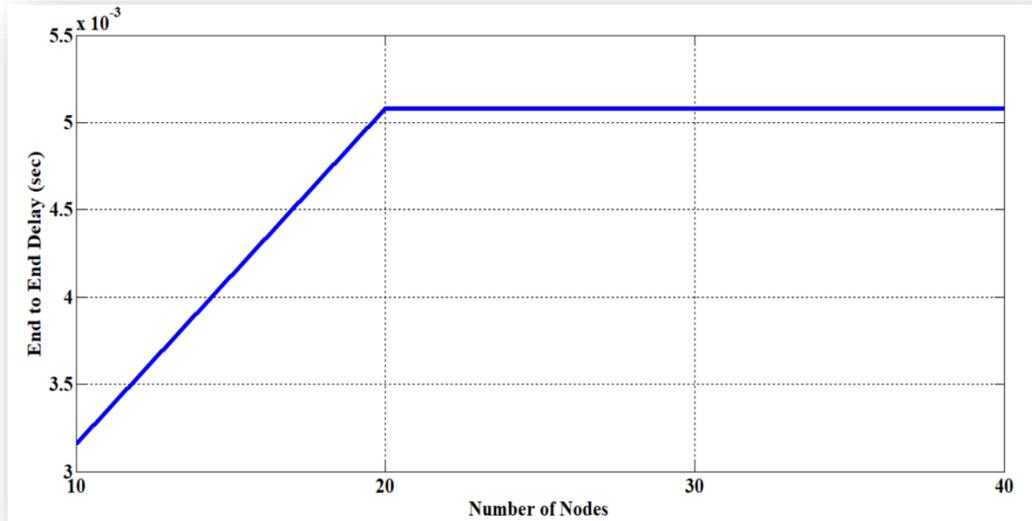


Figure11. End to End Delay vs. no. of nodes

5. Conclusion

For the network, load balancing technique plays a very vital role in order to achieve the QoS solutions. The load balancing is an operation that tries to improve network utilization by choosing lightly loaded links for routing new requests. When this process success it can eliminate delay and link congestion. The hybrid algorithm is introduced to limit the congestion and improved QoS performance as compared to the static and dynamic load balancing technique. The results are compared by considering four important performance metrics such as throughput, packet delivery ratio, packet loss and the end to end delay. In all cases, the proposed load balancing approach shows improved performance when compared to static and dynamic methods, it has a higher throughput, higher packet delivery ratio, lower delay, and lower packets loss also using such algorithm will maximize the fairness of load distribution in the network, while two

stages static and dynamic will minimize the decision-making process due to using static stage when the network is not congested and dynamic stage when the network is congested.

CONFLICT OF INTERESTS.

- There are no conflicts of interest.

6. References

- [1] N.Krishna Chaitanya, and S.Varadarajan,” A NEW QUEUING TECHNIQUE IN ADAPTIVE MULTIPATH ROUTING FOR CONGESTION CONTROL WITH PACKET PRIORITY IN WIRED DATA NETWORKS “, International Journal of GEOMATE, Vol.12, Issue 33, pp.104-108, May 2017.
- [2] Monika Kushwaha, and Saurabh Gupta.” Various Schemes of Load Balancing in Distributed Systems- A Review “, International Journal of Scientific Research Engineering & Technology (IJSRET), Volume 4, Issue 7, ISSN 2278 – 0882, July 2015.
- [3] N.Krishna Chaitanya, and S.Varadarajan,”Load distribution using multipath routing in wired packet networks: Comparative study”, ELSEVIER, No.8, PP.234-236, 2016.
- [4] USMAN AHMAD, MD ASRI BIN NGADI, ISMAIL FAUZI BIN ISNIN,” SURVEY ON END TO END CONGESTION CONTROL TECHNIQUES IN DIFFERENT NETWORK SCENARIOS”, Vol.77, No.3, ISSN: 1992-8645, E-ISSN: 1817-3195, 2015.
- [5] UrjashreePatil, and RajashreeShedge, “Improved Hybrid Dynamic Load Balancing Algorithm for Distributed Environment”, International Journal of Scientific and Research Publications, Volume 3, Issue 3, ISSN 2250-3153, March 2013.
- [6] Mayuri A. Mehta, Devesh C. Jinwala,”A Hybrid Dynamic Load Balancing Algorithm for Distributed Systems”, JOURNAL OF COMPUTERS, VOL. 9, NO. 8, pp. 1825- 1833, AUGUST 2014.
- [7] Hafiz Jabr Younis, Alaa Al Halees, and Mohammed Radi, “Hybrid Load Balancing Algorithm in Heterogeneous Cloud Environment”, International Journal of Soft Computing and Engineering (IJSCE), Volume-5, Issue-3, and ISSN: 2231-2307, July 2015.
- [8] Rashmi Saini, and Ashish Bisht, “A Hybrid Algorithm for Load Balancing”, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 5, Issue 7, ISSN: 2277 128X, July 2015.
- [9] S. Subalakshmi, and N. Malarvizhi ,” Enhanced Hybrid Approach for Load Balancing Algorithms in Cloud Computing “,International Journal of Scientific Research in Computer Science, Engineering and Information Technology ,Volume 2 , Issue 2 ,ISSN : 2456-3307, 2017 .
- [10] Er. Ritu Aggarwal,” Managing Load Balancing In Cloud Computing”, International Journal of Innovative Research and Advanced Studies (IJIRAS), Volume 5, Issue 2, ISSN: 2394-4404, February 2018.
- [11] Ali Akbar Neghabi, Nima Jafari Navimipour, Mehdi Hosseinzadeh, and Ali Rezaee, “Load balancing mechanisms in the software defined networks: a systematic and comprehensive review of the literature”, IEEE. Translations and content mining are permitted for academic research only, 2169-3536 (c) 2018.
- [12] Siham Hamadah,” A Survey: A Comprehensive Study of Static, Dynamic and Hybrid Load Balancing Algorithms “, International Journal of Computer Science and Information Technology & Security (IJCSITS), ISSN: 2249-9555, Vol.7, No.2, Mar-April 2017.
- [13] Sherihan Abu Elenin, and Masato Kitakami,” Comparing Static Load Balancing Algorithms in Grid”, Springer-Verlag Berlin Heidelberg, LNCS 6874, pp. 170–177, 2011.

- [14] Zahra Mohammed Elngomi, and Khalid Khanfar,” A Comparative Study of Load Balancing Algorithms: A Review Paper”, International Journal of Computer Science and Mobile Computing, Vol.5, Issue.6, pg. 448-458, June- 2016.
- [15] Navpreet Singh, and Dr. Kanwalvir Singh Dhindsa, “Hybrid Scheduling Algorithm for Efficient Load Balancing In Cloud Computing “, Int. J. Advanced Networking and Applications, Volume: 08, Issue: 05, ISSN: 0975-0290, Pages 3181-3187, 2017.

تحسين خوارزمية لتوازن حمل التوجيه على اساس سياسة الهجين

منى محمد جواد نور محمد مهدي

قسم هندسة التحكم والنظم، جامعة التكنولوجيا، بغداد، العراق

noormohammer57@gmail.com Muna Al-Nayar@yahoo.com

الخلاصة

يتم تعريف الشبكة على أنها مجموعة من العقد المرتبطة بطريقة للتعامل مع البيانات والرسائل ونقلها من المصدر إلى الوجهة. يحدث الازدحام في الشبكة عند حدوث الكثير من الزيارات، ويؤدي إلى التأخير، وفقدان الحزمة، وتدهور عرض النطاق الترددي، وارتفاع حمل الشبكة. تم تصميم خوارزميات موازنة التحميل لتقليل الازدحام في الشبكة. موازنة التحميل هي إعادة توزيع عبء العمل بين عقدتين أو أكثر ليتم تنفيذها في نفس الوقت. اثبتت من السياسات من خوارزميات موازنة التحميل: موازنة تحميل ثابتة وديناميكية. تقترح هذه الورقة خوارزمية موازنة التحميل على أساس الهجين (سياسة ثابتة وديناميكية) باستخدام شبكة المحاكاة (الاصدار الثاني). يتم استخدام السياسة المختلطة لتحسين أداء الشبكة من خلال إعادة توزيع الحمل بين العقد الزائدة إلى العقد الأخرى التي يتم تحميلها عند حدوث الازدحام. وتبين نتائج المحاكاة أن الخوارزمية المقترحة تستخدم أداء الشبكة فيما يتعلق بالانتقال، ونسبة توصيل الرزم، وفقدان الرزم، والتأخر من طرف إلى آخر.

كلمات الدالة: - الشبكة، خوارزمية موازنة الحمل، موازنة الحمل الثابتة، موازنة الحمل الديناميكية، توازن الحمل الهجينة.