Load Balancing In Wireless Mesh Networks Using Modified RED Algorithm

[1]Avinash Chandra Mishra, [2]Prayag Tiwari, [3]Kirti Singh, [4]Alok Kumar

[1](Avinash Kumar Mishra) Software Engineer, Infosys, Pune, India; [2](Prayag Tiwari) Electrical Engineer, L&T Limited, Noida, India; [3](Kirti Singh) Dept.of Electrical and Electronic Engineering, JSS College, Noida, India; [4] (Alok Kumar) Software Engineer, Infosys, Pune, India
Email: Aviraj017@gmail.com, Prayagforms@gmail.com, Kirtisingh.3007@gmail.com, Singhdodger@gmail.com

ABSTRACT

Wireless Mesh Networks (WMNs) are potentially the most powerful in providing internet connection in a cost-effective manner. WMNs have unique characteristics like static nodes, and sharing of the wireless medium. They consist of gateways which provide connectivity to the backbone network (preferably the internet). Thus, the traffic volume is expected to be very high and due to limited capacity of these gateways they are considered as a potential bottleneck. One of the techniques to improve their capacity is balancing the load arriving at the gateways known as load-balancing. There are many methodologies for Load-Balancing but they do not provide optimal solution as many of them do not consider the real world situations and many lead to non optimal paths and reduced gateway throughput. The proposed scheme for Load-Balancing uses the concept of multiple gateways and combines it with the concept of multiple queues at each gateway. Prioritization of the real time data packets arriving at each gateway queue is done and thus QoS is achieved along with load balancing. A new schema is proposed based on understanding of current research work which simulated in artificial environment using Network Simulator2 (NS2). The proposed solution also meets the requirement of QoS and effectively balances load on gateways and is suited for real world scenarios. The simulation result of the proposed schema has been compared with the standard existing solutions and it has been shown that our proposed schema outperformed the existing RED algorithms. The efficiency of RED is found to be 73.3% and our proposed schema has 80.3% efficiency which turns out to be 7% higher for the used scenerios

Keywords : Load balancing ; Modified RED algorithm; Wireless Mess Network

1. Introduction

The Wireless Mesh Networks (WMNs) are the new kind of multi-hop network architecture and are gaining large popularity due to rapidly increasing demand for internet connectivity. Also, they do not require Line-of-Sight communication which enables them to provide internet connectivity to residential and remote areas. The application scenarios for WMNs include disaster recovery, broadband internet access etc. WMNs comprise of Gateways, Routers and Clients. The Gateways provide wired connection to the backbone network which is preferably the internet. The Routers provide packet delivery to the Clients. The Gateways are considered as a subset of Routers. WMNs are cost effective way of providing internet connection. WMN is one of the hot topics for research both in academia and industry. Many of industry's bigwigs such as Nokia, Intel and Motorola etc are developing their own mesh devices with customary protocols for WMNs. The increased commercial interest in WMNs has driven IEEE to establish a new task group IEEE 802.11s for WMN standardization [5]. Although WMNs are based on the concept of mobile ad-hoc networks there are some key issues. One of such issue is that of Load Balancing at Gateways. As we know that the Gateways are the only component providing clients the connection to outside world, the load is expected to be very high. There are many proposals to this problem but they are mostly ineffective the reason being their inability to adapt to the real world. They result in suboptimal paths resulting in reduced throughput and in some cases increased congestion.

2. Implementation of Load balancing Routing Protocol

The proposed architecture provides QoS by dividing traffic into different categories [8], marking each packet with a *code point* that indicates its category, and scheduling packets according to their code points. The Assured Forwarding mechanism is a group of code points that can be used in a network to define three classes of traffic, each of which has three drop precedences. Those drop precedences enable differential treatment of traffic within a single class. Assured Forwarding uses the RED [1] mechanism by enqueuing all packets for a single class into one physical queue that is made up of three virtual queues (one for each drop precedence). Different RED parameters are used for the virtual queues, causing packets from one virtual queue to be dropped more frequently than packets from another. A packet with lower drop precedence is given better treatment in times of congestion because it is assigned a code point that corresponds to a virtual queue with relatively lenient RED parameters. For example, one code point might be used for assured traffic and another for best effort traffic. The assured packet virtual queue will have higher minimum and maximum thresholds than those of best effort queue, meaning that best effort packets will enter the congestion avoidance and congestion control phase prior to assured packets.

2.1 Network Traffic Prioritization

In this system each packet competes equally with the other packets for the sources. This means that a user having a video conference and a user downloading a movie have the same priority and thus, the same probability of being successfully reaching their destination or being dropped.
The new load-balancing algorithm i.e. Modified RED is IP QOS (Quality of Service) [6] based architecture. This architecture prioritizes the packets based on the user requirements. It basically assigns priority based on the type of data being carried by the packet. If the packet carries real time data it is given higher priority than the packet which is carrying pieces of a download.
In the proposed System we make use of two types of Routers viz. the Edge and the Core. The Edge Router has two responsibilities- it examines the incoming packet and marks the packet with code point (determines the desired level of service). The Core Router also has two responsibilities- it examines the incoming packet for the code point and forewords the incoming packet according to the same.

2.2 Packet Classification

In a network, packets are generally differentiated on a flow basis by the five flow fields in the Internet Protocol (IP) packet header: source IP address; destination IP address; IP protocol field; and source and destination ports.Packet classification is a means of identifying packets to be of a certain class based on one or more fields in a packet. The identification function can range from straightforward to complicate. The different classification support types include: IP flow identification based on the five flow parameters: Source IP Address, Destination IP Address, IP protocol field, Source Port Number, and Destination Port number, Identification based on IP Precedence or DSCP field etc.

Packet identification is also based on other TCP/IP header parameters, such as packet length. Identification based on source and destination Media Access Control (MAC) addresses. Application identification based on port numbers, Web Universal Resource Locator (URL) addresses, and so on.

2.3 IP Precedence

The IP Precedence field in the packet's IP header is used to indicate the relative priority with which a particular packet should be handled. It is made up of three bits in the IP header's Type of Service (ToS) byte.

IP Precedence values	IP Precedence bits	IP Precedence names
0	000	Routine
1	001	Priority
2	010	Immediate
3	011	Flash
4	100	Flash Override
5	101	Critical
6	110	Internetwork Control
7	111	Network Control

Table 2.3 Precedence of Packets

In the proposed schema the priority is set using the Code Point.

2.4 Code Point

As we know each router has a physical queue into which the packets are queued during transmission. But sometimes we need more than a single physical queue. The code point system has virtual queues inside the physical queues. Each combination of the physical and virtual queue determines a certain level of service for example the 2nd virtual queue in the 1st physical queue is numbered as 02 and it certainly determines a different level of service form the combination 11 which gives the first virtual queue in the second physical queue. The level of service depends on the increasing or decreasing priority form the first to the last queue.The architecture has three major components. One is the policy and resource manager, which handles the creation of network policies and distribution of those policies to the routers. The other components are edge routers and core routers. Proposed schema attempts to restrict complexity to only the edge routers of a domain. A policy specifies which traffic receives a particular level of service in the network. Although a policy and resource manager is a necessary component of the network that allows an administrator to communicate policies to the edge and core devices, it is unimportant for the ns implementation. Instead, policy information is simply specified for each edge and core device through the Tcl scripts.

Edge Router Responsibilities:
1 Examining incoming packets and classifying them according to policy specified by the network administrator.
2 Marking packets with a code point that reflects the desired level of service
3 Ensuring that user traffic adheres to its policy specifications, by shaping and policing traffic.

Core Router Responsibilities:

1 Examining incoming packets for the code point marking done on the packet by the edge routers.
2 Forwarding incoming packets according to their markings. (Core routers provide a reaction to the marking done by edge routers.)

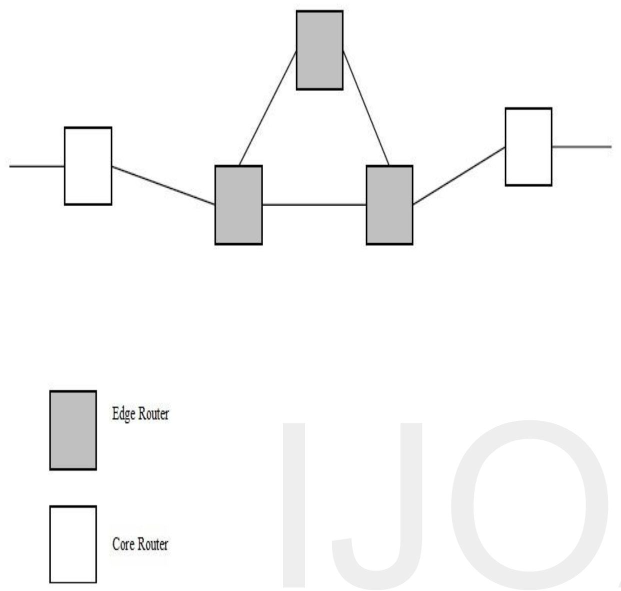

Fig 2.4 Edge and Core Routers

2.5 Classes used to in the Schema

In order to propose the new schema in NS [11], four classes were used- one for the basic router functionality, one for the policy table, one each for edge and core routers.
Each class is defined below:

i. dsRed Class:

The dsRed is the base of the implementation. It defines the dsREDqueue class, which is the parent class for the edgeQueue and coreQueue classes. The purpose of having this class is that it implements all functionality and declares all parameters that are common to both edge and core routers. The queue structure consists of four physical queues, each containing three virtual RED Queues, refered to as precedence levels. Each physical queue corresponds to a class of traffic and each combination of a queue and precedence level is associated with a code point. The code point provides the drop precedence. Packets are enqueued in a certain queue and precedence number according to their code point marking. They are treated according to the corresponding parameters for that queue and precedence number, thus a code point specifies a certain level of service.
It uses the concept of four physical queues and there are three virtual queues inside each physical queue. Not all physical and virtual queues need be used, as in our case we have used only three physical queues and each queue has only two virtual queues. The number of queues can be increased but only after changing the specific parameters and recompiling NS.

ii. Edge Class:

It implements the edge router. It models an edge router. The edge queue class, as a child of the dsRED Queue class is responsible for maintaining multiple physical and virtual queues and processing those queues according to their parameters. Its additional responsibilities are to mark packets with code points and policing the traffic.

iii. Policy Class:

The policy class is used by the edgeQueueclass to handle all policy functionality. The policy class handles the creation edge router policies. A policy determines the treatment that a traffic aggregate will receive at the edge device. The edge devices use policy information to determine with what code point to mark packets.
A policy is said to be established between a source and destination node. All flows matching that source-destination pair are treated as a single traffic aggregate.

iv. Core Class:

This class implements the core router. It is intended to work after the edge router has done its job. It forwards packets according to the marking done on them by the edge router. Packets having code points signifying low priority are dropped at a higher rate than the packets marked with code points having high priority.

3.1 Block Diagram for the Schema

The packets from the source arrive at the edge router. The number of nodes through which the packets travel is not important as only the source-destination pairing is important. For each source destination pair a code point is assigned to each packet. All the packets at the edge router are assigned the code point. After the code point has been assigned the packets are forwarded to the next node. This node is called the core router and it checks the code point for every packet and tries to send them to their destination. But due to a finite queue length all packets are not sent and as a result some are dropped. The dropped packets are discarded as they can no longer be examined. The remaining packets are sent to their intended destination. The main aim of the edge-core router pair is to reduce the packet drop. Packet drop can also occur at the edge router for the obvious reasons but it is advised to reduce this for effective functioning and high throughput.

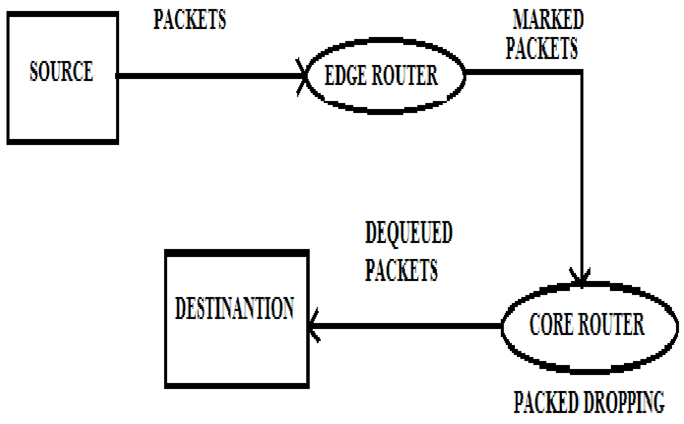

Fig 3.1 Block diagram of the proposed schema

3.2 Algorithm of the Proposed Scheme

The algorithm of the proposed schema is as follows:

Setting Parameters at Edge Router

1. *define the number of physical and virtual queues*
2. *make entry for every source and destination pair into the policy table and assign code point*
3. *define parameters for the policer*
4. *make entry in the phb table for every code point*
5. *set red parameters for every virtual and physical queue pair*

Setting Parameter at Core Router

1. *assign weight to the physical queue*
2. *set mean packet size*
3. *maintain phb entry and red parameters*

Dropping Packets at Core Router

Drop packets according to the core router

Case 1

If packet size is less than minimum threshold,

There is no packet drop

Case 2

If packet size is between minimum and maximum threshold

Drop the packets according to drop probability

Routing the packets

Send the packets to their destination according to the policy table entry

The proposed algorithm uses the priority based packet dropping for load balancing. The parameters at edge and core routers must be set before the packets can be dropped at the core router. The parameters include the number of physical and virtual queues, the policer and PHB table parameters, red parameters etc. The packets are then dropped depending on their packet drop probability.

4. Test Cases

The proposed scheme is implemented in NS-2 using the following network scenario. This scenario consists of three source nodes and three destinations each. The nodes 1,2,3 are sources and 7,8,9 are the destination nodes. The sources send the udp packets to the destinations. As a result, the sources are attached with a cbr agent each. Each cbr agent is attached to a udp agent which serves as the origin of the packets. Each source is attached to the edge router i.r the node 4 using drop tail queuing. The edge router is attached to the node 5 using the edge link. The basic function of the edge router is to mark the packets with the code point. The node 5 is in turn attached to node 6 which serves as the core router. The core router checks each packet and sends it to the intended destination. As a result, when the queue size is less than the minimum threshold all packets reach their destination. As the queue size increases and becomes greater than the minimum threshold, the packets with maximum drop priority are dropped. But when the queue size becomes greater than the maximum threshold, more packets are dropped depending on their priority.

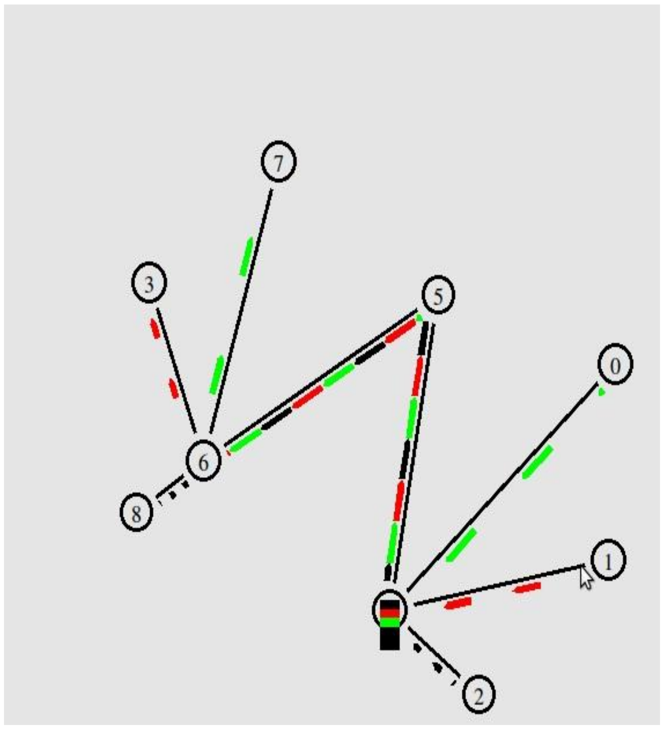

Fig. 4. Simulation of the Proposed Schema after Packets Start Dropping

The test cases are formed by changing different parameters like the queue size, packet size, packet delay etc and a number of different scenarios were formed for the same network structure. The results from all the scenarios were combined to arrive at the final result. The queue size is the maximum number of packets which can be stored at a router at a given time. Packet size is the size of the packets which will be sent from source to the destination. For UDP packets there are no acknowledgements so their size need not be set.

4.1 Assumptions

The following assumptions are taken:

1 All the traffic sources are assumed to be the UDP sources.
2 The packets size is fixed to be 1000 bits.
3 RED parameters are different for different sources.
4 9 nodes are used where 3 nodes act as source and destination and 3 nodes are used as routers for modified Red functionality.
5 The simulation environment is isolated for the outside world.
6 Simulation duration 60 seconds.

	Test Case 1	Test Case 2	Test Case 3	Test Case 4
Packets Sent	101392	101392	101392	101392
Packets Received	81169	80934	81238	82340
Packets Dropped	20223	20458	20154	19532
Throughput	80.05	79.82	80.12	81.12

Table 4. Packet Statistics of Proposed Schema

4.2 Performance Evaluation

For the load balancing schemas the throughput is the percentage of total sent packets received at the destination. The graph below shows the throughput for the four test cases used in our scenario.

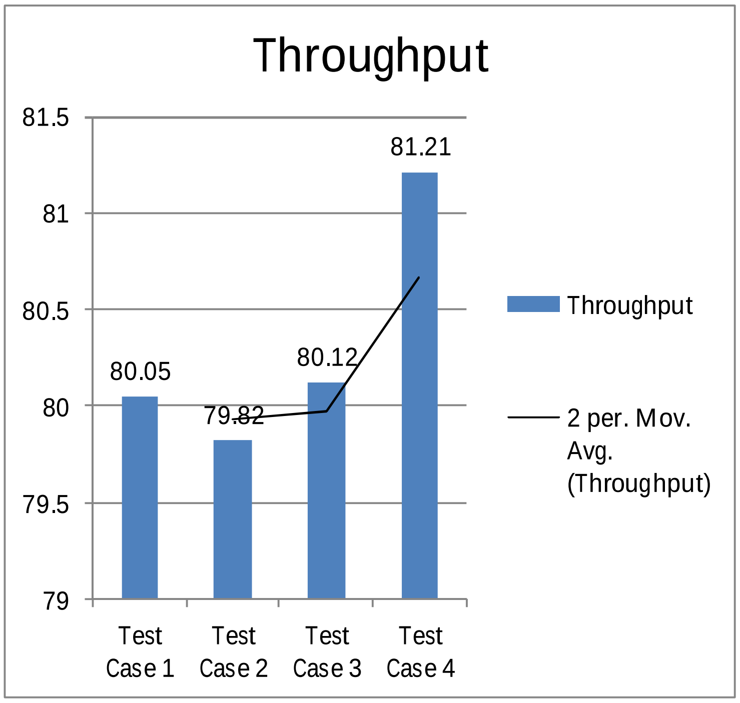

Fig. 4.2 Throughput of Modified RED

The dark bars show the throughput in each of the four test cases used and the thin black line shows the average of the throughputs with insertion of a new case. From calculations we find that the mean throughput comes out to be 80.3% for the schema considered in this project and the test cases used to evaluate it.

4.3 Comparison between RED and Modified RED

The RED is one of the best routing protocols available but, it has some serious problems. When the queue size is between the minimum and maximum threshold, it starts to drop tha packets randomly. As a result, all packets whether they carry real-time traffic or non-real-time traffic have the same dropping probability. Also when the average queue size is greater than the maximum threshold, all packets are dropped. Use of such conditions leads to loss of data. The modified RED solves this problem by providing each packet with the code point. The code point determines the packet drop probability and thus the type of packet being carried by it.Modified Red uses the concept of two types of routers- edge and core- which are used in co-ordination with each other for congestion control and avoidance. The edge router marks the packet with the code point and the core router forward the packet based on the assigned code point.
From the figure 5.4 we find that the average throughput for RED is 73.3%, which is clearly less than the modified RED algorithm.

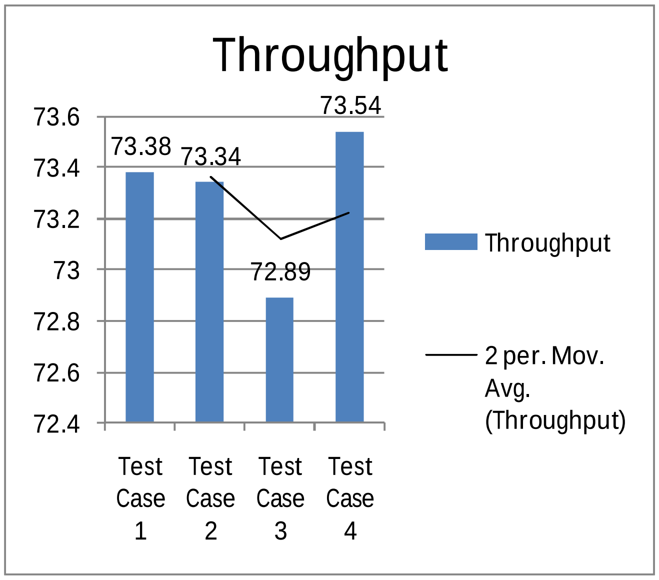

Fig. 4.3 throughput of RED

Thus, the modified Red is an improvement over the existing Random Early Detection algorithm.

IV. Conclusion

The proposed schema meets the requirement of QoS and effectively balances load on gateways and is suited for real world scenarios. The efficiency of RED is found to be 73.3% and our proposed schema has 80.3% efficiency which turns out to be 7% higher for the used scenerios. The proposed schema has two areas where future work can be concentrated. We have statically assigned the priorities to the incoming packets. The packet priorities can be automatically generated and assigned to them. The number of physical and virtual queues can be increased depending on the requirement by changing the necessary parameters.

References

[1] Shrinivas Vegesna, *IP Quality of service*, pp. 230-245, Indianapolis: Cisco Press, 2000.
[2] Cordeiro C. and Agrawal D.P., *Ad hoc & Sensor Networks Theory and Applications* , pp. 189-192, World Scientific Publishing , Spring 2006
[3] Liu C., Hou C., Guo Z., Zhu S., “Load-Balancing Algorithm to Improve Throughput Capacity in Wireless Mesh Backbone Networks”, *Proceedings of the Institute of Electrical and Electronics Engineers (IEEE 2006)*, pp 3-5, New Jersey, USA, 2006.
[4] Nandiraju N., Nandiraju D., Santhanam L., Wang J.F., Agrawal D.P., *Wireless Mesh Networks: Current Challenges and Future Directions of Web-in-the-sky*, pp. 235-239, IEEE Wireless Communications Magazine, Cincinnati, USA, April 2006.
[5] Nandiraju D., Santhanam L., Nandiraju N., and Agrawal D.P., “Achieving Load Balancing in Wireless Mesh Networks Through Multiple Gateways”, *Proceedings of the Institute of Electrical and Electronics Engineers (IEEE 2006),* pp.808-811, Cincinnati, USA, 2006.
[6] Shinoda Y., Nguyen L. T., “Load Balancing using the Quality of Service(QoS) in Wireless Mesh Networks ”, *Presented at the Institute of Electrical and Electronics Engineers (IEEE 2006)*, pp. 2-5, March 2006.
[7] Dongmei S., Bing H. “Probabilistic Load Balancing in Wireless Mesh Networks”, *Proceedings of the Institute of Electrical and Electronics Engineers (IEEE 2008)*, pp. 3-6, May 2006.
[8] Liu C., Shu Y., Zhang L., Li J., “Efficient Multiple Gateways Load-Balancing and QoS Routing in Wireless Mesh Networks”, *Proceedings of the Institute of Electrical and Electronics Engineers (IEEE 2008)*, pp.3-5, Cincinnati, USA, 2006.
[9] Tokito H., Sasabe M., Hasegawa G., Nakano H., “Routing method for Gateway Load balancing in wireless mesh networks”, *Proceedings of the Institute of Electrical and Electronics Engineers (IEEE 2008)*, pp. 3-5, 2008.
[10] Ramachandran K., Hasegawa G., Tokio H., "Design and

Implementation of Infrastructure Mesh Networks", *Presented at Institute of Electrical and Electronics Engineers(IEEE 2009),* pp. 3-5, Gualdeloupe, France, 2009.