Simulative Analysis with QRED to Decrease Packet Loss

Erekar Bala Krishnarjuna Rao
erekarbalakrishnarjuna.rao@gmail.com

CH. Vijaya Bhaskar
chvbhaskar@sreenidhi.edu.in

M. Nagaraju
nagarajum@sreenidhi.edu.in

***Abstract*—** Communication networks are facing packet loss at routers, where different approaches are used to reduce. Similarly RED is one of them, that existing RED [1] [2] algorithm and its variants are found in flow controlling. For minimizing dropping of packets and reducing buffer overflow. This paper propose a new routing algorithm in which additional FIFO controlled queue buffer before existing RED algorithm, to increases performance and throughput of the router. It is experimented and improvements in results are shown with help of OMNet++.

Keywords- *RED, Queue, Delay, Packet flow, OMNeT++*

*****

I. INTRODUCTION

Communication is one of the most important aspects in nowadays life. Networking is a set of computers or devices that are connected to each others with that can exchange data. Internet, intranet, and extranet are three types of networking. Think of a topology as a network's virtual shape or structure. Which are logic but practically not easy to see in real networks architecture. Routers are network devices that literally route data around the network. By examining data as it arrives, the router can determine the destination address for the data; then, by using tables of defined in router, the router determines the best way for the data to continue its journey. Unlike bridges and switches, which use the hardware-configured MAC address to determine the destination of the data, routers use the software-configured network address to make decisions? A router is used to route data packets between two networks. It reads the information in each packet to tell where it is going. If it is destined for an immediate network it has access to, it will strip the outer packet, readdress the packet to the proper Ethernet address, and transmit it on that network. If it is destined for another network and must be sent to another router, it will re-package the outer packet to be received by the next router and send it to the next router. This approach makes routers more functional than bridges or switches, and it also makes them more complex because they have to work harder to determine the information. The routers are defined with static and dynamic routing protocols. Routing explains the theory behind this and how routing tables are used to help determine packet destinations. Routing occurs at the network layer of the OSI model. They can connect networks with different architectures such as Token Ring and Ethernet. They can transform information at the data link level. Routers do not send broadcast packets or corrupted packets. If the routing table does not indicate the proper address of a packet, the packet is discarded. A network protocol defines rules and conventions for communication between network devices.

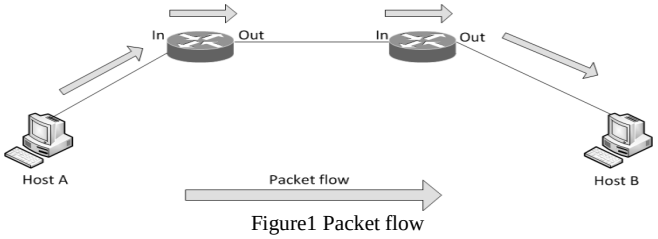

Figure1 Packet flow

Protocols for computer networking all generally use packet switching techniques to send and receive messages in the form of packets. Network protocols include mechanisms for devices to identify and make connections with each other, as well as formatting rules that specify how data is packaged into messages sent and received. Some protocols also support message acknowledgement and data compression designed for reliable and/or high-performance network communication. Hundreds of different computer network protocols have been developed each designed for specific purposes and environments. Routing protocols are special-purpose protocols designed specifically for use by network routers on the Internet. Routing protocols fall into two categories, Interior and Exterior. Interior protocols called IGPs (Interior Gateway Protocols), refer to any routing protocol used exclusively within an Autonomous System, providing Intra-AS routing. Each IGP represents a single routing domain within the AS. Exterior protocols called EGPs (Exterior Gateway Protocols) are routing protocols that facilitate routing between and across different AS'. Some examples of IGP protocols are: RIP, OSPF [11], and IGRP. Exterior Gateway Protocols, such as BGP (Border Gateway Protocol), are designed to serve as a conduit for communication between autonomous systems. BGP is the most popular inter-autonomous system (or Inter-AS) routing protocol used throughout the Internet community. There are several routing mechanisms that may be used as input sources to assist a router in building its route table. Typically, routers use a combination of the following routing methods to build a router's route table:

Directly connected interfaces are routes that are local to the router. That is, the router has an interface directly connected to one or more networks or subnets. These networks are inherently known through the routers configured interface

 ISSN: 2321-8169
2867 - 2873

attached to that network. These networks are immediately recognizable and traffic directed to these networks can be forwarded without any help from routing protocols. Directly connected routes are always the best method of routing because the router knows the network this datagram is destined for firsthand and does not rely on some other means to learn this route. However, when traffic is destined to networks beyond a router locally attached links help is needed. Static routes are routes to destination hosts or networks that an administrator has manually entered into the router's route table. Static routes define the IP address of the next hop router and local interface to use when forwarding traffic to a particular destination. Because this type of route has a static nature, it does not have the capability of adjusting to changes in the network. If the router or interface defined fails or becomes unavailable, the route to the destination fails. Static routes conserve bandwidth because they do not cause routers to generate route update traffic; however, they tend to be time consuming because a system administrator has to manually update routes when changes occur in the network. Dynamic routing protocols not only perform these path determination and route table update functions but also determine the next-best path if the best path to a destination becomes unusable. The capability to compensate for topology changes is the most important advantage dynamic routing offers over static routing. Although there are specific advantages and disadvantages for implementing them, they are not mutually exclusive [5].

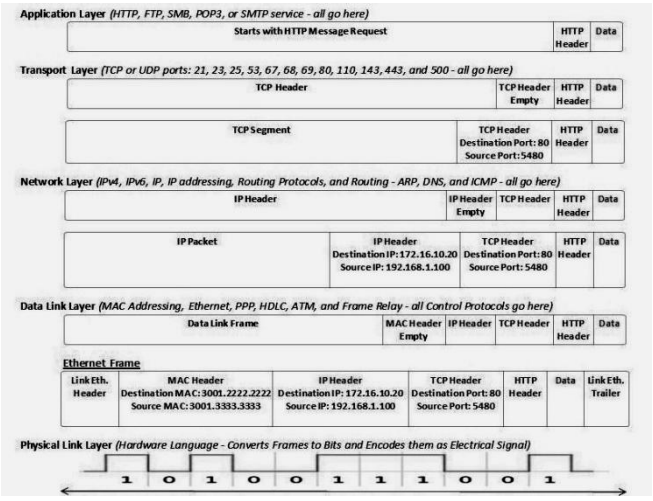

Figure2 layers using protocols

In the current Internet, dropped packets serve as a critical mechanism of congestion notification to end nodes. The solution to the problem is for routers to drop packets before a queue becomes full, so that end nodes can respond to congestion before buffers overflow. Router along with these uses RED for buffer control and drop packets. Where RED drop the packets based on its buffer threshold value. At present single congestion control mechanism cannot solve all of the problems due to the wide number of parameters that have impact on system's performance. In addition to that in today's high speed network, the nature of congestion is not really known and one can't easily characterize the different levels of congestion along with the facts that what is an extreme condition of congestion, how long it lost and what does is the percentage of dropped packets. On the Internet, people cannot rely on end users to incorporate proper congestion control. Router mechanisms must be provided to protect responsive flows from non-responsive ones, and prevent internet meltdown.

Traffic on the Internet [4][6] tends to fluctuate and to be greedy. Ideally, a router management algorithm should allow temporary bursty traffic, and penalize flows that persistently overuse bandwidth. Also, the algorithm should prevent high delay by restricting the queue length, avoid underutilization by allowing temporary queuing, and allocate resource fairly among different types of traffic [22]. In practice, most of the routers being deployed use simplistic Drop Tail algorithm, which is simple to implement with minimal computation overhead, but provides unsatisfactory performance. To attack this problem, many algorithms are proposed, such as Random Early Drop (RED) [2], Flow Random Early Drop (FRED) [24], BLUE [25], Stochastic Fair BLUE (SFB) [26], and CHOKe (CHOose and Keep for responsive flows, CHOose and Kill for unresponsive flows) [27]. Most of the algorithms claim that they can provide fair sharing among different flows without imposing too much deployment complexity. Most of the proposals focus on only one aspect of the problem (whether it is fairness, deployment complexity, or computational overhead), or fix the imperfections of previous algorithms, and their simulations setting are different from each other. These all make it difficult to evaluate, and to choose one to use under certain traffic load. But not making efforts to control queue flow at these algorithms. For each of these algorithms, three aspects are discussed: (1) resource utilization (whether the link bandwidth is fully utilized), (2) fairness among different traffic flows, and (3) implementation and deployment complexity. We proposed an algorithm to limit these problems, using Queue before RED can help in reducing delay, loss of packets. So this queue stores and forward packets to RED to reduce its flow and control its buffer size to overcome the packet loss problem. It helps in affective delay problems. In this, we use OSPF routing protocol for routing packets, ICMP ping for client server communication and Queue to store and forward pings. As RED has min. and max. threshold levels based on which packet dropping is done. To control and maintain these levels, queue is added before RED, so it tries to eventually hold threshold levels in between min. and max., which means an average flow of packets forwarded to RED with the help of Queue in FIFO style, to reduce its dropping chances.

	Drop Function	Control Variable	Changes from original RED
FRED	Single linear	Per-flow queue length	Per-flow queue length, number of active flow
FBRED	Single linear	Average queue length	Per-flow Max_{drop}
SRED	3 segment step	instantaneous queue length and number of active flow	Step drop function, number of active flows, instantaneous queue
CBT-RED	Single linear	Average queue length	Class based threshold
XRED	Single linear	Average queue length	Priority based drop
BRED	4 segment step	Per-flow queue length and number of active flows	Per-flow queue length, number of active flows, step drop function
DSRED	Two linear	Average queue length	Two linear drop function with different slope,
BLUE	Step function	Link utilization and packet loss	Step increase/decrease function, link rate, packet loss
REM	Exponential function	Link rate mismatch and buffer difference	Exponential function, link rate mismatch and buffer difference
SFB	Step function	Instantaneous queue length	Organize sub-queue in Bloom filter

Figure3 RED variants summary [21]

II. BACKGROUND WORK

In packets forwarding, network components play an important role. When a device has multiple paths to reach a destination it always selects one path by preferring it over others. Routing is done by special network devices called routers, but routers have limited functionality and limited scope. Routers use routing algorithms to find the best route to a destination. When we say "best route," we consider parameters like the number of hops, time delay and communication cost of packet transmission. A router is always configured with some default route. A default route tells the router where to forward a packet if there is no route found for specific destination. In case there are multiple paths exist to reach the same destination, router can make decision based on the following information:

*Hop *Count *Bandwidth *Metric *Prefix-length *Delay

Routes can be statically configured or dynamically learnt.

Unicast routing: Most of the traffic on the Internet and Intranets are sent with destination specified, known as unicast data or unicast traffic. So routers just have to look up the routing table and forward the packet to next hop.

Broadcast routing: Broadcast packets are not routed and forwarded by the routers on any network. Routers create broadcast domains. But it can be configured to forward broadcasts message for destined to all network devices.

Multicast routing: It is special case the data is sent to only nodes which want to receive the packets. Router must know that there are nodes who wish to receive multicast packets (or stream) then only it should forward.

Anycast packet forwarding is a mechanism where multiple hosts can have same logical address. Anycast routing is done with help of DNS server. Whenever an Anycast packet is received it is enquired with DNS to where to send it.

Because the router maintains different routing protocols which are responsible for routing and forwarding packets between hosts and it also works as an interface between three layers (link layer, network layer and transport layer). Packet loss exists in between these three layers. To check effective loss consider router as the best thing to study packet loss. This loss is regular due to delay, congestion, buffer overflow, or link failure, we get different values to distinguish packet loss and delay in wired and wireless medium. But the packets get dropped. The packet loss in wireless is more because of its infrastructure. The wireless network packet loss is high, in wired network packet loss can be shown with throughput. OMNet++ [7] [8] is one of the network simulators used for simulation projects, we use this to give a practical vision to packet loss. Where in RED, ICMP and OSPF detects the changes in topology, such as link failures and converges on a new loop-free routing structure within seconds. It computes the shortest path tree for each route using a method based on the shortest path first algorithm. The OSPF [11] routing policies for constructing a route table are governed by link cost factors (external metrics) associated with each routing interface. Cost factors may be based on the distance of a router (round-trip time), data throughput of a link, or link availability and reliability, expressed a simple unit with less numbers. This provides a dynamic process of traffic load balancing between routes of equal cost. A router can't forward a packet to all possible destinations in the way that a bridge can.

Static routers: These must have their routing tables configured manually with all network addresses and paths in the internetwork.

Dynamic routers: These automatically create their routing tables by listening to network traffic. Routing tables are the means by which a router selects the fastest or nearest path to the next "hop" on the way to a data packets final destination. This process is done through the use of routing metrics.

There are various management algorithms proposed to overcome congestion issue such as Random Early Detection (RED), Gentle Random Early Detection (GRED), and Adaptive Gentle Random Early Detection (AGRED) for TCP/IP networks since last two decades (Zhu et al., 2002). Network congestion increases queuing delay, packet loss and it degrades the throughput (Hosam, 2009). RED is existing algorithm which was proposed by sally Floyd and Van Jacobson in 1993 to deal packet dropping(S. Floyd & Jacobson, 1993). In RED average queue length which is calculated using a low pass filter with Exponential Weighted Moving Average (EMWA), is used to make the decision regarding dropping a packet and failed to control direct flow. Stabilized RED (SRED) is another variant of RED which was developed by Ott et al. in 1999 (Ott, T.V.Lakshman, & Wong, 1999). The buffer utilization in the proposed algorithm is stabilized without concerning the load level and limiting the queue flow size(Ryu, Rump, & Qiao, 2003). In (Sally Floyd, 2000), Gentle Random Early Detection (GRED) was proposed in order to increase throughput and reduce the undesired oscillation in buffer size of router by enhancing parameter settings of RED. In this packet dropping decision is based on parameter setting of max probability. Although ARED provides the advantage of automatic parameter setting in response to changes of traffic but it lacks the clarity regarding best policy of parameter setting and averaging queue (Hosam, 2009). GRED proposed to deal with RED issues but still packet loss rate is high in GRED. Adaptive Gentle Random Early Detection (AGRED), proposed by (Mahmoud Baklizi, Hossein Abdel-jaber, 2012) to deal with packet loss issue in GRED. AGRED modified the calculation of dropping probability formula and was evaluated using simulator. All

these algorithms used but not made to control the affective queue size in RED and reduce dropping. The solution to the queues problem is for routers to drop packets before a queue becomes full, so that end nodes can respond to congestion before buffers overflow. By dropping packets before buffers overflow, queue management allows routers to control when and how many packets to drop. Queue management mechanism can provide the following advantages for responsive flows.

Reduce packet drop in routers and provide greater capacity to absorb naturally-occurring bursts without dropping packets.

Delay lowering will reduce the delays seen by flows and performance is better when the end-to-end delay is low.

Flow that does not use congestion control may receive more bandwidth than a flow. Some scheduling algorithms like FQ (Fair Queuing) and CBQ (Class Based Queuing) by themselves do nothing to Control the overall queue size or the size of individual queues flow. Random Early Detection, or RED, is one among used for routers that will provide the Internet performance advantages cited in the [RED93]. In contrast to traditional algorithms, which drop packets only when the buffer is full, the RED algorithm drops arriving packets probabilistically. The probability of drop increases as the estimated average queue size grows. Note that RED responds to a time-averaged queue length, not an instantaneous control of flow. On the other hand, if the queue has been relatively full, indicating persistent congestion, newly arriving packets are more likely to be dropped.

(a) Estimation of Queue Size

RED estimates the average queue size, in the forwarding path using a simple exponentially weighted moving average.

(b) Packet Drop Decision

In the second portion of the algorithm, RED decides whether or not to drop an incoming packet. It is RED's particular algorithm for dropping those results in performance improvement for responsive flows. RED parameters, minth (minimum threshold) and maxth (maximum threshold), Minth specifies the average queue size *below which* no packets will be dropped, while maxth specifies the average queue size *above which* all packets will be dropped. There is no averaging queue to maintain these thresholds. There are three approaches for addressing this issue [28]:

Static Threshold. Low rates of packet loss are assumed to be congestive, while rates above some predefined threshold are deemed malicious.

Traffic modeling. Packet loss rates are predicted as a function of traffic parameters and losses beyond the prediction are deemed malicious.

Traffic measurement. Individual packet losses are predicted as a function of measured traffic load and router buffer capacity. Deviations from these predictions are deemed malicious.

III. PROPOSED SYSTEM

In communication networks, a topology is a usually schematic description of the arrangement of a network, including its nodes and connecting lines. There are two ways of defining network geometry: the physical topology and the logical (or signal) topology.

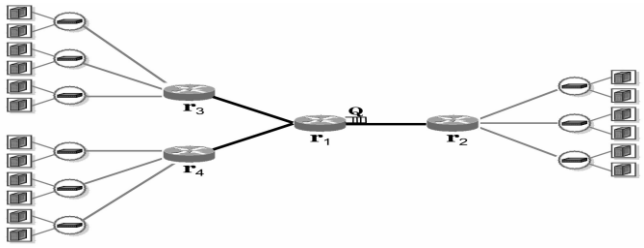

Figure4 simple topology

We consider a dump bell topology which can easily designed to work. Network consists of individual homogeneous routers interconnected via directional point-to-point links. This model is an intentional simplification of real networks (e.g., it does not include broadcast channels or independently failing network interfaces) but is sufficiently general to encompass such details if necessary. We assume that the bandwidth, the delay of each link, and the queue limit for each interface are all known publicly. Within a network, we presume that packets are forwarded in a FIFO fashion, based on a local auto forwarding table. These forwarding tables are auto updated via a distributed link-state routing protocol OSPF . This is critical, as we depend on the routing protocol to provide each node with a global view of the current network topology:

A. Design Using OMNet++

During the design phase, we have taken n number of clients and a server who in turn are connected with three routers (R1, R2 and R3) and two switches (switch1 and switch2).

Server and Client modules are modified and separate message ping files are coded and used for sending the reply and get response. This is implemented using ICMP ping protocol, the.MSG files will have server ping.MSG and client ping.MSG, from server to reply and client receive at the same time request sever and get responses or availability. The routers are designed by Queue and RED combined mechanism for reducing router level loss and delay. It also has some buffer size as router should have storage of tables, routing information, etc., These are taken by usage of INET package. Switches are used for an Ethernet flow and connection, it helped in adding routers in between and also fiber, or wireless channels can be used in place of ehter to get separate result values. The experiment is an explicit study (checking) of packet loss between nodes. The project consists of n number of clients connected to a server in wired medium where the three routers are placed in between them, the routers are especially designed to check packet loss. The routers are used to calculate an effective packet loss over the network.

Initially RED algorithm was having minimum and maximum buffer threshold values, based on which dropping of packets done in router. It maintains the size of buffer average at RED. We are using Queue RED to store packets before it reaches RED. Those packets are queued and forwarded slowly to router. This system maintains flow and storage control in RED (buffer), thus it is named as QRED (Queue RED), with the help of RED and Queue packet loss is defined very efficiently. When the buffer size of queue exceeds it result in packet loss, that packets are dropped. This lost packet information is maintained by RED and Queue control may have flow problem. In a single client communication there won't be any packet loss, but if n number of clients at a time

 ISSN: 2321-8169
2867 - 2873

ping or send requests to the server, the responsiveness of the server and the reply of ping will affectively lead to packet loss in the network. This loss is studied by QRED.

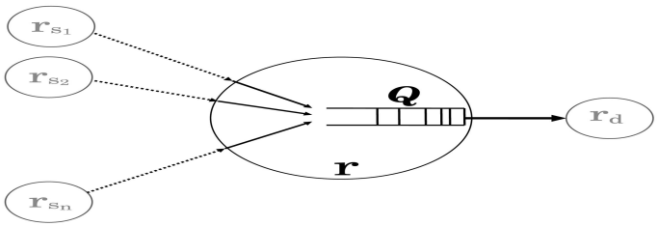

Figure5 Queue validations with QRED in router

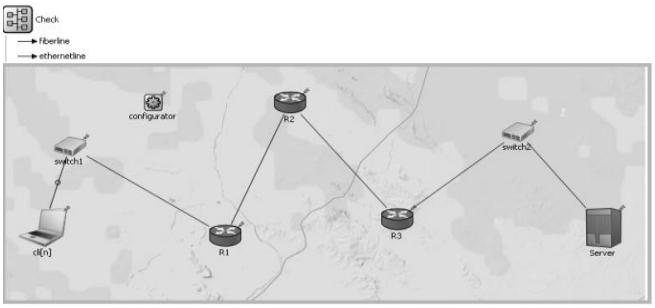

Figure6 Architecture design (NED)

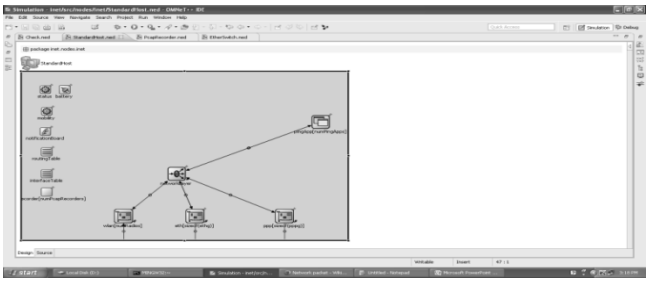

Figure7 Client and Server (same properties)

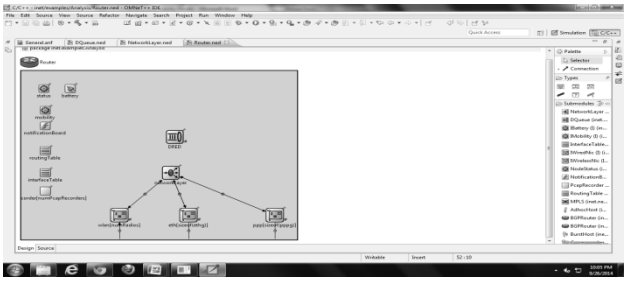

Figure8 Router with DRED

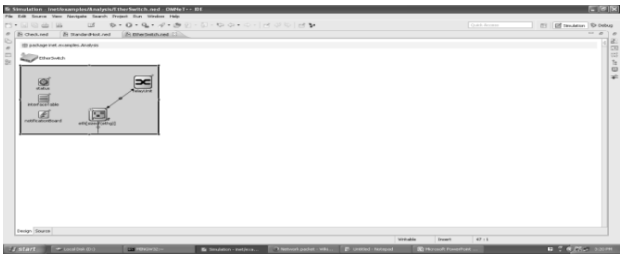

Figure9 Switch

B. Implementation Using OMNeT++

The networks have different possibilities for communication and implementing wide varieties of algorithms for packet flow over a network, but the major need of packet loss detection at the router is neglected. It is possible to estimate correct values for a wired communication but not possible to count the values based on its infrastructure. Major responsibilities of a router is to route the packet, and also find the best path, but there are some chances of packet loss at router as it doesn't have any queue algorithms to reduce direct pressure on some existing algorithm of router (static or dynamic algorithms). Router mainly, has RED for packet dropping check and flow control in congestion and buffer problem situations, to reduce direct pressure on red using a queue mechanism to control flow and maintain buffer values, to reduce immediate pressure on end server. By this, flow control is reduced and monitored by QRED and packet loss is reduced and also detecting loss of packets at the router level and improving efficient server responses to clients. First clients flow F() is given to Queue, then Queue forwards it in FIFO manner to RED. In RED the buffer size is always averaged so it drops fewer packets. If Queues size exceeds Tmax level in queue, then it affects the max. threshold of RED which in turn lead in packet drop. These dropped packets are taken as loss packets to study loss rate at server, clients and router. Results obtained are used for calculating loss rate, buffer overflow, congestion and delay.

Here we use ICMP ping messages for clients to ping server and get response and follow OSPF routing protocol in router. We used ICMP because it gives error in connection or loss of signal and also unreachable status of the server, ICMP is the best to have studied on packet loss checking. OSPF has good flowing of packets in sequential manner in priority usage and queuing in making best path search in the networks to reach source perfectly.

The design of the project is done by using OMNet++ and INET frame package. All connections can be seen in figure3.1.1. We expect that the knowledge about the loss patterns may help design decisions for enabling or improving Quality of Service (QoS) that support in networks.

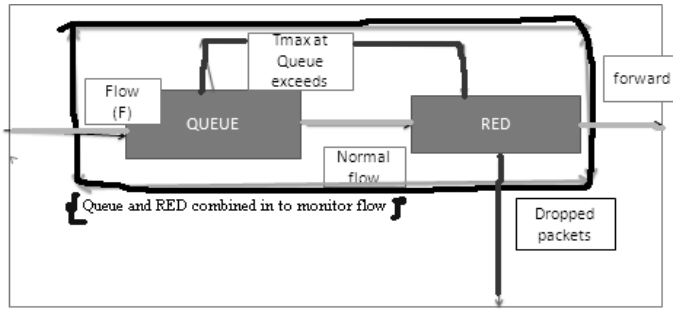

Figure10 QRED flow diagram in the router

OMNet++ is one of the best open source simulation software's. It is very friendly in approach. OMNeT++ is a public-source, component-based, modular and open-architecture simulation environment with strong GUI support and an Embeddable simulation kernel. OMNeT++ provides component architecture for models. Components (modules)

 ISSN: 2321-8169
2867 - 2873

are programmed in C++, and then assembled into larger components and models using a high-level language (NED).
The major concepts in preparing a project using the network simulator they are,
-NED file
-INI file
-MSG file
-Cc file
Ned (network descriptor) is used for designing required network for simulation. INI (information file) it defines the major functionalities of the network. MSG (message file) is used for sharing data between nodes as packets or datagram's. CC (C++ code) the entire code of the project writes in this file.

OMNeT++ has some frameworks for designing of projects and INET [12] is one helps that in better implementation and design.

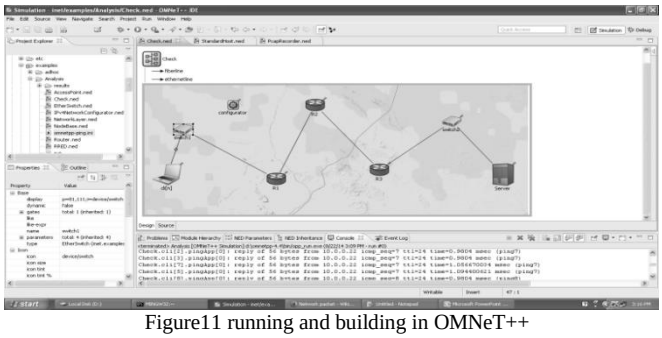

Figure11 running and building in OMNeT++

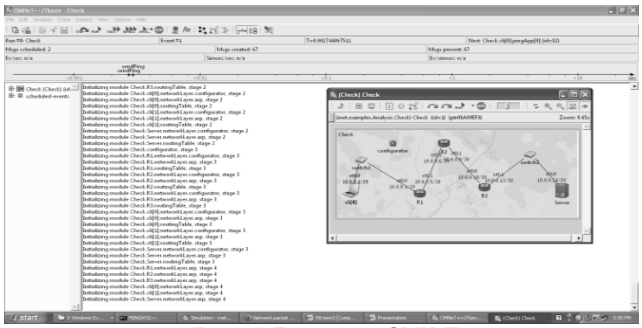

Figure12 Execution in OMNeT++

IV. USING THE TEMPLATE

A. *RED (Random Early Detection)*

Random Early Detection (RED) [13] algorithm was first proposed by Sally Floyd and Van Jacobson in [14] for Queue Management (QM) [15] and then standardized as a recommendation from the IETF in [16]. Essentially, RED algorithm has two separate parts. One is for computing the average queue size, which determines the degree of bursts that will be allowed in the router queue. It takes into account the period when the queue is empty (the idle period) by estimating the number m of small packets that could have been transmitted by the router during the idle period. After the idle period, the router computes the average queue size as if m packets had arrived to an empty queue during that period. Later it marks packets that are moving through router by an additional buffer size are maintained by router in this case loss of packets chances are exists.

B. *QRED (Queue Before RED)*

So by QRED the routers on networks are modeled such that the queue before RED [17] design will maintain the packet flow and control the threshold level, regular level limit is maintained and stores acceptable packets and forwards through the router to clients and server. QRED derived from a model that characterizes the behavior of end-to-end connections with multiple routers in between. When drop probability at router decreases, packet loss decreases and hence sending rate at end host increases. In this, we consider F flow levels and make them to act to queue port in the router which will be having RED this mean Queue and RED work together and after Tmax time limit if flow at router exceeds it drops and increases flow at server, by this server buffer size increased to receive pings from n clients so loss occur at server by this we get some information regarding dropped or lost packets in the network flow. We use this information for analysis.

Here we can overcome regular dropping of packets at the router level, but not on the server side. As flow constraints are checked at the router and Queued to reduce loss at router, to get Qos at Server.

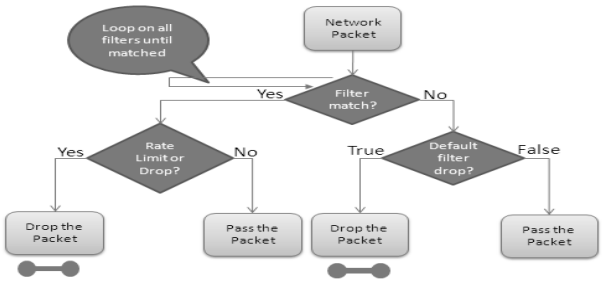

Figure13 Flow diagram at router

C. *RESULTS*

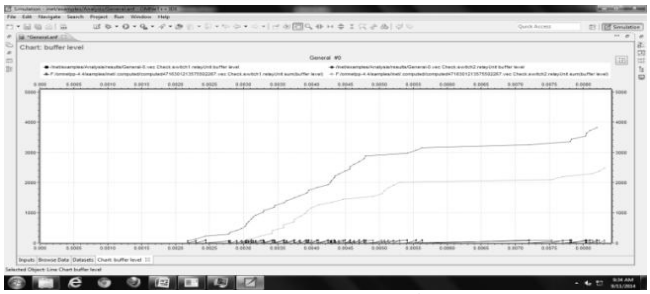

Figure14 delay readings of QRED

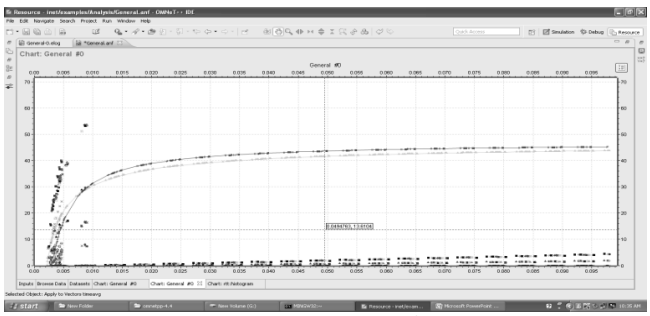

Figure15 Loss readings of QRED

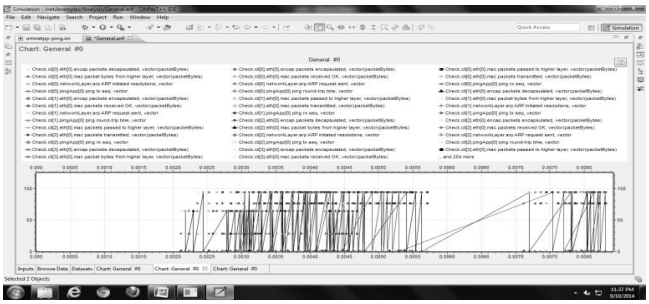

Figure16 clients ping buffer congestion

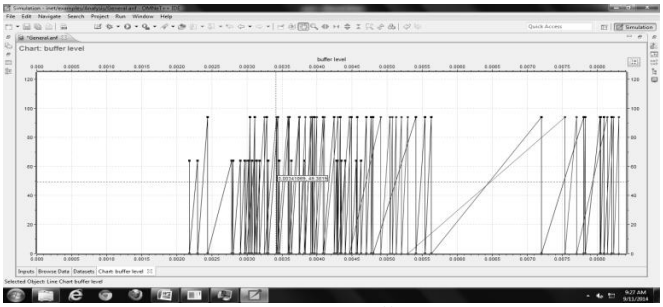

Figure17 servers ping buffer congestion

CONCLUSION AND FUTURE SCOPE

To the best of our knowledge, this paper is one of the attempts to distinguish between existing RED and our proposed approach in router, dropping packets due to buffer overflow and congestion. Previous work has approached this issue using a static threshold, which is fundamentally limiting in control of flow. Using the same, we additionally used queue to control flow in FIFO style before RED. So it come to the aid of the packet loss than existing, considering work done in wired, can be implemented in others. If we standardize it at host's level it may have better results.

REFERENCES

[1] An algorithm for increasing the robustness of RED's active queue management S Floyd, R Gummadi, S Shenker -, 2001.

[2] Random early detection gateways for congestion avoidance S Floyd, V Jacobson - 1993 - ieeexplore.ieee.org

[3] PING attack–How bad is it? S Kumar - , 2006 - Elsevier

[4] Internet Packet Loss: Measurement and Implications for End-to-End QoS, Michael S. Borella, Debbie Swider, and Suleyman Uludag and Gregory B. Brewster

[5] End-to-end packet delay and loss behavior in the Internet JC Bolot -, 1993

[6] Internet research needs better models S Floyd, E Kohler -, 2003.

[7] An overview of the OMNeT++ simulation environment A Varga, R Hornig -, 2008

[8] The OMNeT++ discrete event simulation system A Varga (ESM' …, 2001).

[9] "Controlling Queue Delay" describes a new AQM algorithm by Kathie Nichols and Van Jacobson.:

[10] A novel adaptive traffic prediction AQM algorithm Z Na, Q Guo, Z Gao, J Zhen, C Wang , 2012.

[11] OSPF Monitoring: Architecture, Design, and Deployment Experience. A Shaikh, AG Greenberg -, 2004.

[12] Steinbach, Till, Kenfack, Hermand Dieumo, Korf, Franz and Schmidt, Thomas C. (2011), "An Extension of the OMNeT++ INET Framework for Simulating Real-time Ethernet with High Accuracy",SIMUTools 2011.

[13] Performance analysis of the random early Detection algorithm V. Sharma j. Virtamo p. Lassila1, finland

[14] S. Floyd, V. Jacobson. Random early detection gateways for congestion avoidance. IEEE/ACM Transactions on Networking (TON) August 1993.

[15] Self-Tuning Random Early Detection Algorithm to Improve Performance of Network Transmission Jianyong Chen, Cunying Hu, and Zhen Ji

[16] FC: Recommendations on Queue Management and Congestion Avoidance in the Internet.

[17] Congestion Control Algorithms in High Speed Telecommunication Networks. Aun Haider, Harsha Sirisena, Krzysztof Pawlikowski † and Michael J. Ferguson‡

[18] www.goole.com, search packet forwarding in network, information is by web.

[19] Minimizing Packet Loss by Optimizing OSPF Weights Using Online Simulation Hema Tahilramani Kaur, Tao Ye, Shivkumar Kalyanaraman, Kenneth S. Vastola

[20] IoS Quality of Service : Some IoS tips for Internet Service Providers (ISP) or Access Points (AP) Mehmet S¨uzen E-mail: mehmet dot suzen at physics dot org Memo's Island (Dated: January, 2005)

[21] comparative study of Congestion control techniques In high speed networks. Shakeel ahmad, dli mustafa, bashir ahmad, arjamand bano And al-sammarraie hosam.2009.

[22] Jain, R., and Ramakrishnan, K.K., Congestion Avoidance in Computer Networks with a Congestionless Network Layer: 1988.

[23] D. Lin and R. Morris. Dynamics of Random Early Detection. 1997.

[24] W. Feng, D. Kandlur, D. Saha, K. Shin, Blue: A New Class of Active Queue Management Algorithms U. Michigan 1999.

[25] A. Demers, S. Keshav, and S. Shenker. Analysis and Simulation of a Fair Queueing Algorithm, 1990.

[26] Rong Pan, Balaji Prabhakar, Konstantinos Psounis. CHOKE, A Stateless Active Queue Management Scheme for Approximating Fair Bandwidth Allocation. IEEE INFOCOM 2000.

[27]]Detecting Malicious Packet Losses Alper T. M_zrak, Student Member, IEEE, Stefan Savage, Member, IEEE, and Keith Marzullo, Member, IEEE.