IJERT
ISSN : 2278-0181
International Journal of Engineering Research & Technology

Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 7 Issue 06, June-2018

An Improved Network Congestion Control System

Michael C. Emmanuel
Department of Computer Science
University of Port Harcourt
Nigeria

L. N. Onyejegbu
Department of Computer Science
University of Port Harcourt
Nigeria

Abstract **- Congestion is a known issue for packet-switched networks like the Internet. It is not unusual for packets to arrive in bursts from different sources as packets crisscross the network, and congestion ensues. Typically, congestion happens when packets arrive in a burst or when sources are sending more data than the network can accommodate. The reality is that switching devices have buffers and buffers can help to absorb burst traffic, nevertheless, if burst traffic persists, buffers fill up and incoming packets are dropped. Expanding the buffer size may intensify the problem in light of the fact that huge buffer size can result to huge delay and an inevitable congestion collapse. This is a known problem in multifarious networks. The proposed system sets to counteract congestion in the network by combining traffic shaping and network border monitoring and control mechanisms to ensure congestion does not occur. The system uses exchange of feedback messages among edge routers to control unresponsive traffic tending to enter the network, thereby, forestalling congestion in the network.It used leaky bucket algorithm for traffic shaping and rate control algorithm for network border monitoring. The rate control algorithm involved two phases; a slow start phase and a congestion avoidance phase. It transitions from slow start phase to congestion avoidance phase when it detects imminent congestion. This introduced communication overhead. The application of leaky bucket algorithm served as additional layer of congestion control and reduced the communication overhead introduced by the rate control algorithm. The system was developed with Java programming language and object oriented methodology was applied.**

General Terms - Internet, congestion, network, algorithm, packet, flow control, traffic shaping.

Keywords - Network congestion, congestion collapse, quality of service, congestion recovery, traffic management, congestion recovery.

1. INTRODUCTION

The way we live has considerably been affected by the rise of the internet. New technologies has seen to the invasion of smart devices around us. Today many individuals have smartphones, tablets, etc while some even own more than one piece, coupled with the emergence of more smart gadgets like smart TVs, smart billboards and even smart cars, it suffices to say that the Internet infrastructure must be stretched more than ever before. However, despite the apparent increasing pressure on limited network resources, there has not been a proportional increase in the investment in network infrastructure to sustain this growth of internet of things. What this means is great pressure on available network infrastructure which had really never been enough. According to Bradley [1] the rise of the internet as well as the proliferation of "bandwidth-hungry" smart gadgets around the world has posed challenges of QoS on available network resources. QoS for networks is an industry-wide set of standards and mechanisms for ensuring high-quality network performance for critical applications. By using QoS techniques, network administrators can use existing resources efficiently and ensure the required level of service quality.

The transmission control protocol (TCP) host-based congestion control mechanisms have been key to the robustness of the Internet. However, the Internet is no longer a small user community, it has grown exceedingly and it is no longer practical to rely on hosts using end-to-end congestion control for best-effort traffic,hence, a need to device means of implementing congestion control within the network.The new argument is that the network must now contribute in controlling its own resource and handling congestion since end-to-end approach has left the Internet with lots of pitfalls.

A number of traffic shaping techniques exists which attempts to improve packet transmission. Traffic Shaping involves buffering traffic temporarily and dropping packets when reasonable to avoid congestion. It involves enforcing a limit on the bandwidth size a connection uses. Although these mechanisms attempt to prioritize network traffic for improvements, they are, basically, host-to-host implemented network control mechanisms and alone, are not able to prevent network congestion. Due to this evident limitation, an optimized network-based congestion control system is proposed, which advances QoS beyond the end-to-end devices to the network core and ensuring unresponsive traffic are prevented from entering the network in the first place. The proposed system uses a traffic shaping mechanism (leaky bucket algorithm) as additional congestion control layer to regulate the flow of potential burst traffic arriving at the network. Leaky bucket algorithm regulates irregular flow of traffic into a constant flow thereby minimizing chances of network congestion.

1.1 Misconceptions about Congestion Control

Clearly, congestion ensues when data traffic exceeds available network resources. Common assumption is that as resources become more affordable congestion will be handled automatically. According to Jain [2], this created grounds to misconception that:

As memory becomes more affordable congestion will be resolved automatically since it occurs because of a limited buffer capacity.

Congestion will be resolved as high-speed connections become available because it results from slow links.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 7 Issue 06, June-2018

Congestion will be resolved automatically as faster processors emerge and are deployed because it is caused by low processing speed.

Contrary to these beliefs, more congestion and poor performance may be the case if concerted efforts is not made towards developing an appropriate protocol design. High memory capacity devices and low memory capacity devices are both vulnerable to network congestion. For high memory capacity devices, the packets are buffered and delayed in long queues such that they time out and would have been retransmitted. Whereas, for low memory capacity devices, excess traffic begin overflow and discard.

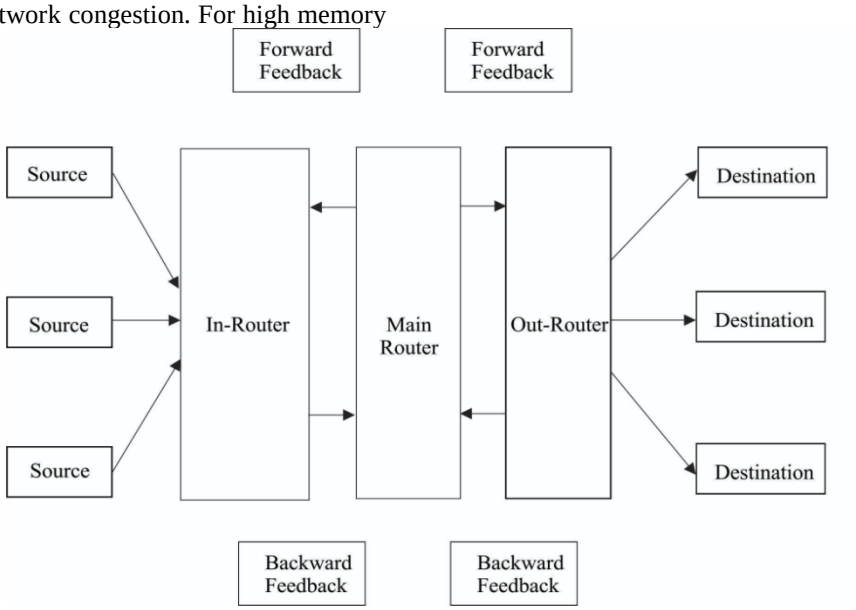

Fig 1. Architecture of the Existing System (Sharadehandra et al. 2017).

2.1 Review And Related Works

Floyd [3] proposed an "equation-based congestion control for unicast applications". According to them, (TCP) has been doing well in managing most "best-effort traffic" in the internet today. Nevertheless, a congestion control technique that is compatible with TCP and avoids reacting to packet drop by slowing down rate of transmission by half will be a good way of handling best-effort unicast streaming multimedia traffic. With this mechanism, the source regulates rate of sending depending on measurement of loss event. Meanwhile, loss event is the situation where some packets are dropped within a one round-trip time.

Yang [4] opined that in shared networks (eg. Internet), the hosts should react to congestion by adjusting their rate of transmission thereby preventing a collapse and maximizing the available network resources. According to Yang [4], the internet today is robust as a result of TCP's host-based congestion control techniques. Much as TCP congestion control performs well for large data transmission applications, it would not work well for other newer applications which will find TCP behavior of halving transmission rate in response to congestion as harsh. As such, since internet traffic is mainly TCP based, it becomes pertinent that emerging congestion control techniques still be TCP compatible. Yangthen examined the fairness, responsiveness and assertiveness of TCP, General Additive Increase and Multiplicative Decrease (GAIMD) and some other two typical TCP-Friendly congestion control protocols. These protocols are analysed and simulated and their responses to network changes closely observed. Yangconsidered the integral instabilities in a static network environment and studied "protocol responsiveness and aggressiveness by evaluating their responses to a step increase of network congestion and a step increase of available bandwidth".

Mawhinney [5] designed a congestion management technique that smoothen packets transfer in a network. The technique observes data packets leaving the network for congestion indications, and effectively controlling the host applications to relieve the network of congestion. The design is quite similar to the Congestion Controlling using Network Border protocol designed by Sharadchandra [6] except that, for the former, the rate controlling is implemented in the host end and not in the network. According to Mawhinney [5], the system prevent congestion from reaching the point where data is lost or packets are dropped. The host sessions is divided into mission critical and non-mission critical sessions. The session types are prioritized in a way that during periods of congestion the mission critical session remain unaffected while the non-mission critical sessions are controlled. The mission critical sessions basically enjoy a reserved specific size of bandwidth up to point of congestion.

Kloth [7] developed techniques for congestion control in IP network such as fiber channel network. Methods are provided to detect congestion in a host. As a controller sends packets through a link, the time elapsed between the sending and the receiving is measured. It the time elapsed is

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 7 Issue 06, June-2018

large, the path to destination is presumed to be congested and the port is blocked from receiving subsequent data. In the fibre channel, a data sequence at a buffer controller is received with a source matching one of many ports coupled to the buffer controller and destination accessible via a link also coupled to the buffer controller. The link is shared by traffic from numerous ports attached to the buffer to reach different destinations. The received packets are forwarded to the link and a transmission acknowledgment is received. According to Kloth [7], depending on the time the acknowledgement is provided, the port associated with the received packets is blocked.

In another example, a congestion controlling setup in a fibre channel network is presented. The setup involves a buffer controller and various input ports. The various input ports are designed to receive packets having destinations accessible via a shared resource. The buffer controller is designed to receive packets from of the various inputs, send the packet through the link and receive acknowledgement from the link.

According to Rejaie [8], the stability and robustness of today's internet is mostly a function of end-to-end congestion control schemes and internet traffic today is mostly TCP and will remain so for quite a while. It is important, therefore, to have new applications to be "TCP friendly". Rejaie [8] presented a rate adaptation protocol (RAP) which was TCP Friendly and employs the TCP additive-increase, multiplicative-decrease AIMD algorithm. RAP was well suited for unicast real-time playback streams and the key goal was to separate network congestion control from application-level reliability while maintaining fairness and TCP compatibility.

Evaluating RAP through thorough simulation show that bandwidth is usually evenly shared between RAP traffic and TCP traffic. Rajaie [8]stated that "unfairness to TCP traffic is directly determined by how TCP diverges from the AIMD algorithm". Though RAP performs similarly to TCP in some situations, however, a simple rate control scheme is devised to widen the scope.

Sharadchandra [6] presented a congestion controlling system that uses the network border patrol (NBP). NBP is a congestion control mechanism implemented on the network layer that controls congestion collapse by "a combination of per flow rate monitoring at an out-router and per flow rate control at an in-router", using feedback exchange instructions from the feedback controller in the out-router. The out-router sends backward feedback to the in-router to inform about the rate flow's packets are exiting the network and the in-router regulates, accordingly, the rate packets are entering the network. According to (Sharadcandra et al., 2017), there are two unique types of routers introduced in the network called edge routers. Edge router maybe viewed as in-router or as out-router depending on the active flow direction. An edge router, for example, acting on a flow entering into a network will be termed an in-router, while an edge router acting on a flow leaving a network will be termed an out-router. NBP uses feedback exchange between routers to manage problematic packets attempting to enter the network.

3.1 Architecture of the Existing System

According to Sharadchandra [6], the issues of end-to-end congestion control schemes can be addressed by taking congestion control away from host systems (end-to-end) to the network using a system they called network border patrol (NBP). NBP involves entry and exit routers (in-routers and out-routers respectively) which compares the frequency packets are entering and leaving the network. The existing system employed a Rate Control Algorithm and a Time Sliding Windows (TSW) Algorithm. Rate control algorithm monitors and regulates the entry and exit flows and Time Sliding Window algorithm ensures an in-order delivery of packets. NBP assumed by Sharadchandra [6], basically compare the rates packets from applications are entering and leaving the network. NBP assumes the network is buffering when packets are leaving slower than they are entering the network. This means more packets are arriving than the network can accommodate. NBP comes in by "patrolling the network's borders", observing the flow rate at the input and output of the main-router and making sure there's a balance in the entry and exit rates. This way congestion collapse is prevented as unresponsive flows are not allowed to enter the network. However, Sudhakar [9]stated that "NBP introduced an added communication overhead, in order for an edge router to know the rate at which packets are leaving the network and must exchange feedback with other edge routers". Figure 3.1 shows the Internet architecture adopted by NBP and figure 3.2 shows the architecture of the existing system.

3.2 Rate Control Algorithm

Rate-control-algorithm functionally regulates flows of packets into the network. It observes also, the rate at which packets are leaving the network and uses feedback mechanisms to put up alert whenever there is indication of imminent network congestion. The out-router determines how rapidly packets are leaving the network by means of rate monitoring.

```
currentRTT = currentTime – p.timestamp;
if (currentRTT < e.baseRTT)
e.baseRTT = currentRTT;
deltaRTT = currentRTT – e.baseRTT;
RTTsElapsed = (currentTime – e.lastFeedbackTime) / currentRTT;
e.lastFeedbackTime = currentTime;
for each flow f listed in p
rateQantum = min (MSS / currentRTT, f.egressRate / QF);
if(f.phase = = SLOW_START)
if (deltaRTT x f.ingressRate < MSS x e.hopcount)
f.ingressRate = f.ingressRate x 2 ^ RTTsElapsed;
else
f.phase = CONGESTION_AVOIDANCE
if (f.phase = = CONGESTION_AVOIDANCE)
if(deltaRTT x f.ingressRate < MSS x e.hopcount)
f.ingressRate = f.ingressRate + rateQuantum x RTTsElapsed;
else
f.ingressRate = f.egressRate – rateQantum;
```

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 7 Issue 06, June-2018

According to Sharadchandra [5], flows may be in slow start phase or congestion avoidance phase. New flows coming into the network start with the slow-start phase and advance to the congestion-avoidance phase only when the flow has identified imminent congestion. Whenever an in-router receives a backward feedback, rate control algorithm is initiated. The backward feedback bears a time-stamp, list of flows from the in-router to the out-router and observed rates of each flow by the out-router.

3.3 Time Sliding Window Algorithm

Generally speaking a sliding window is a sub-list that runs over an underlying collection. A sliding window protocol is a kind of packet-based data transmission protocol. The algorithm comes in handy where consistent in-order delivery of packets is necessary. Basically, a unique consecutive succession number is assigned to each slice of the transmission and packets are arranged in the right order by the receiver. Duplicate packets are discarded and missing ones are identified.

On arrival of Forward Feedback at out-router
Start timer
if(packets arrived = = true)
Send CurrentPacket

Wait Packet arrival
if (Packet Forwarded = = true)
Send Backward FeedbackAcknowledgement to in-router

Retry Packet forwarding
If (packet to forward = 0)
Stop timer
Forward next packet

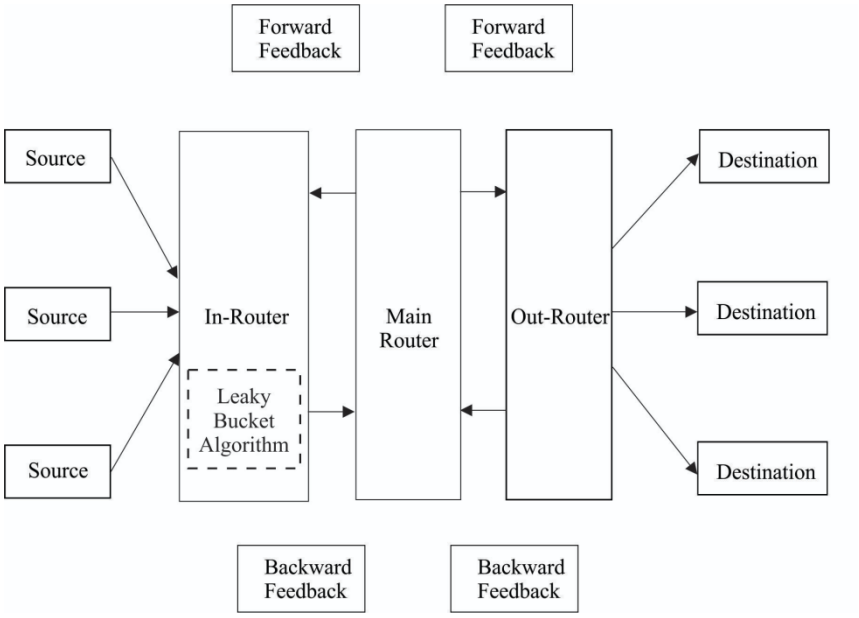

Fig. 2. Architecture of the Proposed System.

3.4 Architecture of the Proposed System

The proposed system employs additional layer of congestion control mechanism; leaky bucket algorithm, to efficiently avert congestion collapse in an IP based network. Leaky bucket algorithm is a traffic shaping algorithm and is applied to smoothen-out burst traffic. In this case, it steps in to smoothen out initial flow of burst traffic, thereby, preventing the rate control algorithm from triggering fully into the "thorough" but "slow" congestion-avoidance stage all the time (the congestion-avoidance stage is triggered any time a burst traffic is experienced). Leaky bucket helps to avoid added communication overhead introduced by the fully invoked rate control algorithm.

Leaky bucket normalizes the packet flow at the input, outbound to the output port of the router. See figure 3.3, Assume a bucket that has been punctured at the bottom, regardless of the burst rate of in-flow to the bucket, it will release the flow at a controlled rate. It is also assumed the capacity of the bucket is unlimited. Hence, no packet loss as a result of the bucket getting over filled. collaborative filtering recommendation system architecture including computational structures and model training algorithms. The system design will also capture the major functional building blocks needed to understand the process of building an online book recommender software system. The architectural design of the proposed system is illustrated in figure 3.3.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 7 Issue 06, June-2018

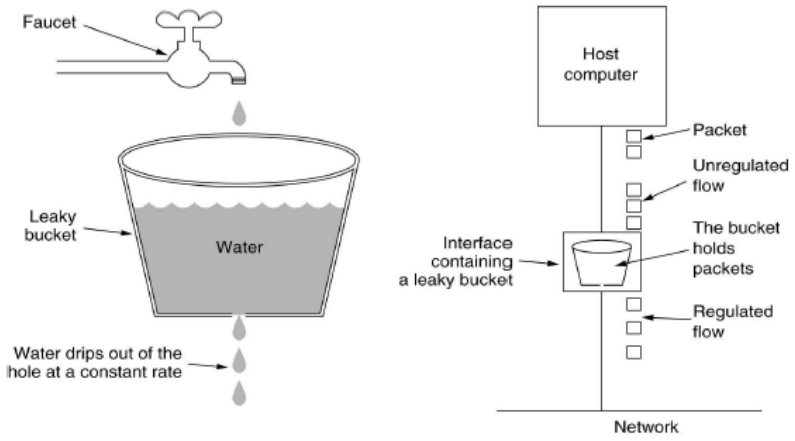

Fig. 3 Graphical representation of the leaky bucket algorithm as adopted..

3.5 Leaky Bucket Algorithm

The adopted leaky bucket algorithm is a traffic shaping mechanism used to control transmission rate in a network. The algorithm works similar to the way actual leaky bucket holds water. It receives data to a maximum capacity and releases data at a set rate and size of packet. When the bucket runs empty the leaking stops; so the input rate can vary but the output rate remains constant. Burst traffic is smoothened-out into regular rate by averaging the data rate.Leaky bucket algorithm in the proposed system steps in to regulate burst traffic, thereby, preventing the rate control algorithm from triggering fully into the "thorough" but "slow" congestion-avoidance stage all the time (the congestion-avoidance stage is triggered any time a burst traffic is experienced). This is to avoid added communication overhead introduced by the fully invoked rate control algorithm. Fig. 3 graphically describes the Leaky bucket algorithm.

Algorithm

Step - 1: Let counter = x

Step - 2: At every clock tick, initialize the x to 'n'.

Step - 3: If size of packet in the front of queue is less than n, send the packet into the network and decrement x by size of packet.

Repeat Step- 3 until n is less than the size of packet.

Step - 4: Reset the x and go to Step - 1.

4. IMPLEMENTATION

The proposed system is composed of 5 modules, they are: Source module, in-router module, main-router module, out-router module and destination module.

Source Module: This module sends the packet to the in-router.

In-router Module: This module is an edge router. It combines leaky bucket algorithm and rate control algorithm to regulate the rate packets are entering the network.

Router Module: This module accepts the packet from the in-router and routes it to the out-router.

Out-router Module: This is a monitoring module; it is also an edge router. It operates on packets leaving the network. Essentially, rate monitoring allows an out-router to determine the rate packets are leaving the network. The out-router ensures in-order delivery of packets using the time sliding window (TSW)algorithm. Out-router contains a rate monitor and a feedback controller.

Destination Module: This Module accepts packet from the out-router router and deliver to the destination.

5. DISCUSSION OF RESULT

The five modules are run concurrently simulating an active network from source to destination. The result from the proposed system is compared with the result from the existing system. Figure 4.shows result from the existing system. Data is entered in a host simulated by the source module. Similarly to the test case of the existing system, a string of all the letters of the alphabet in capitals are sent to the network. The in-router module receives all the packets from the source module and transmit same to the main router module. The in-router exchanges forward and backward feedback with the out-router to perform rate monitoring and rate control to avoid network congestion. The main router routes the received packets to the out-router. There are no dropped packets. The out-router module is rate monitoring router. It observes the rate at which packets are entering and leaving the network. The out-router communicates to the in-router of imminent network congestion. At the destination all packets are delivered. Figure 5 shows the result from the proposed system. In this case leaky bucket algorithm is applied.Table 1 compared sample of 25 packets of same data showing that the average delivery time per packet of the proposed system is 0.4secs against the average delivery time of the existing system which is 1.56secs. This shows that the case with leaky bucket applied (figure 5) transmits packets faster than the case where leaky bucket was not applied (figure4).

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 7 Issue 06, June-2018

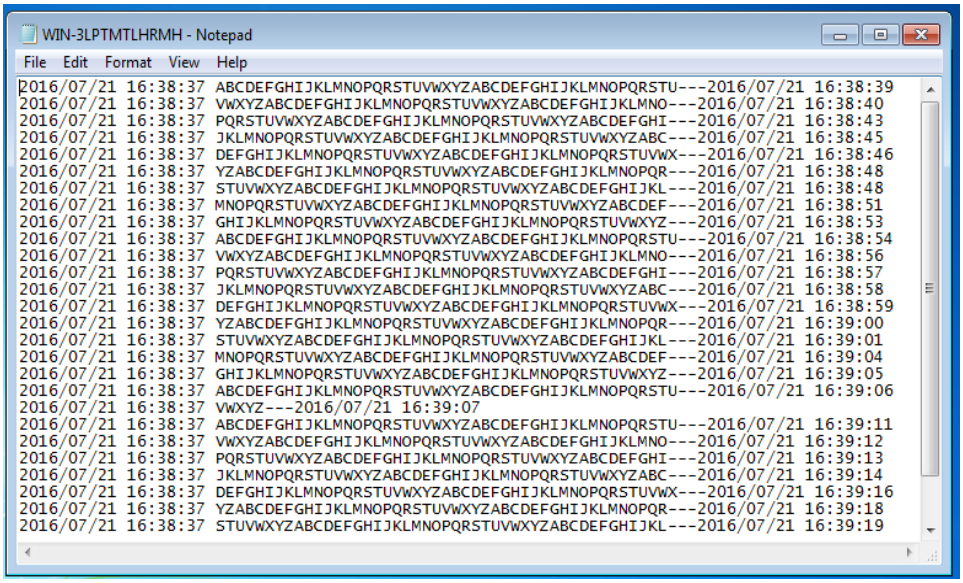

Figure 4:Result of packet transfer time from the existing system with no Leaky bucket algorithm.

Figure 5: Result of packet transfer time from the proposed system with Leaky bucket algorithm.

Table 1: Comparative analysis between Sharadchandra et al. 2017 and proposed system.

S/N	Sharadchandra *et al.* (2017) Packet Delivery Time	Delivery time difference (secs)	Proposed System Packet Delivery Time	Packet Delivery time Difference (secs)
1	16:38:39	N/A	19:46:16	N/A
2	16:38:40	1	19:46:18	2
3	16:38:43	3	19:46:20	2
4	16:38:45	2	19:46:21	1
5	16:38:46	1	19:46:22	1
6	16:38:48	2	19:46:23	1
7	16:38:48	0	19:46:23	0
8	16:38:51	3	19:46:24	1
9	16:38:53	2	19:46:24	0
10	16:38:54	1	19:46:24	0
11	16:38:56	2	19:46:24	0
12	16:38:57	1	19:46:24	0
13	16:38:58	1	19:46:24	0
14	16:38:59	1	19:46:25	1
15	16:39:00	1	19:46:25	0

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 7 Issue 06, June-2018

16	16:39:01	1	19:46:25	0
17	16:39:04	3	19:46:25	0
18	16:39:05	1	19:46:25	0
19	16:39:06	1	19:46:26	1
20	16:39:07	1	19:46:26	0
21	16:39:11	4	19:46:26	0
22	16:39:12	1	19:46:26	0
23	16:39:13	1	19:46:26	0
24	16:39:14	1	19:46:27	0
25	16:39:16	2	19:46:27	0
26	16:39:18	2	19:46:27	0
	Total delivery time	39	Total delivery time	10
	Average delivery time/packet	1.56	Average delivery time/packet	0.4

6. CONCLUSION

The use of the network border patrol framework which applies rate control mechanism at the edge of the network has shown to be effective unlike existing congestion schemes which were basically end-to-end or host-to-host. The traditional host based control mechanism as a solution to network congestion control has not been effective and is discouraged in this work. The proposed system prevents congestion collapse through a combination of rate regulating and control at the in-router, and per-flow rate monitoring at the out-router to ensure packets are not entering the network faster than they are leaving. The adopted leaky bucket algorithm serves as added layer to the control mechanism and ensures that the rate control algorithm does not worry about burst traffic which can cause delay. The forward and backward feedback packets exchange between the edge routers ensure a smooth run of the system.

7. REFERENCES

[1] Bradley, M. (2018). The Value of QoS on Computer Networks. Retrieved from https://www.lifewire.com/definition-of-qos-817890

[2] Jain, R. (1990). Congestion Control in Computer Networks: Issues and Trends. IEEE Network Magazine. 1-2.

[3] Floyd, S., Handley, M., Padhye, J. and Widmer, J. (2000). Equation-Based Congestion Control for Unicast Applications. ACM SIGCOMM Computer Communication Review, Proceedings of the conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, (30) 4; 1-3.

[4] Yang, P., Luo, W., Xu, L., Deogun, J. S., Lu, Y. (2011). TCP Congestion Avoidance Algorithm Identification. University of Lebraska. 310-311.

[5] Mawhinney, T. N. and Davis, J. D. (2004). Network congestion control system and method. Paradyne Corporation. 2-9.

[6] Sharadchandra, C., Qureshi, R., Fulzele, P., Meshram, C. and Jidkuntalawar, P. (2017). Congestion Controlling using Network Border Protocol. International Journal of Modern Trends in Engineering and Research. 4 (1); 114-116.

[7] Kloth, R., J. (2008). Methods and apparatus for network congestion control. CiscoTechnology Inc. 1, 3.

[8] Rejaie, R., Handley, M. & Estrin, D. (1999). RAP: An End-to-End Rate-Based CongestionControl Mechanism for Real-time Streams in the Internet. IEEE. ISBN: 0743-166X. 1337-1339

[9] Sudhakar, M., Ganesh, G. & Rajan, J. (2012). Avoiding Congestion Control using Network Border Patrol. International Journal of Computer Science and Information Technologies. 3 (5); 5243-5246.