Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 8, August 2014, pg.93 – 99

RESEARCH ARTICLE

APAQM: An Alternate Path AQM Congestion Control Algorithm

B.Kiruthiga[1], Dr. E. George Dharma Prakash Raj[2]

[1]Department of Computer Science, Bharathidasan University, Trichy, TamilNadu, India
[2]Department of Computer Science, Bharathidasan University, Trichy, TamilNadu, India

[1] kiruthigabaskaran13@gmail.com, [2] georgeprakashraj@yahoo.com

Abstract- In the TCP network environment system the researcher and developer mainly focus on congestion control. The existing paper provides an analysis of congestion metric with or without flow of information in queue based AQM algorithms, on the basis of the algorithms, proposed a new algorithm APAQM (An Alternative path Active Queue Management Congestion control Algorithm). An Alternative path Active Queue Management Congestion Control Algorithm (APAQM) is used to find the alternate path at shortest distance for sending a packet to source node, during occurrence of congestion. APAQM (An Alternative path Active Queue Management Congestion control Algorithm) is a technique that consists of it Congestion control in internet routers.

Keywords: Congestion control, Queue, APAQM, Internet

I. INTRODUCTION

Congestion in Internet occurs when the large volume of data is being routed on low bandwidth lines or across networks that have high latency and cannot handle large volumes. Then the result is slowing down of packet movement, packet lost or dropped. The increased use of multimedia applications also results in bursty flows in the internet. So there is a requirement of regulating bursty flows in the very large network, the Internet. To regulate these bursty flows, resource allocation must be done efficiently. Router must be able to handle the above problem. The buffer in the routers is to be used effectively by using an efficient Active Queue Management Mechanisms.

A. Active Queue Management

AQM algorithm is a solution to the problem of congestion control in the Internet routers. Researchers and the IETF (Internet engineering task force) proposed active queue management (AQM) as a mechanism for detecting congestion inside the network. Further, they have strongly recommended the deployment of AQM in routers as a measure to preserve and improve WSN/WAN performance. Active Queue Management tries to prevent congestion and provides quality of service to all users.

Active Queue Management (AQM) algorithms have been designed to be able to actively control the average queue length in routers supporting TCP traffic, and thus to be able to prevent congestion and resulting packet loss as much as possible. The purpose of these algorithms is to provide a mechanism to detect network congestion early and to start dropping packets from the router queue before this congestion will affect the network throughput too much.

B. AQM algorithms can be classified as being either reactive or proactive

- A reactive AQM algorithm focuses on congestion avoidance, i.e., active early detection of and reaction to congestion. Congestion can occur in this case, but it will be detected early. Decisions on actions to be taken are based on current congestion.
- A proactive AQM algorithm focuses on congestion prevention, i.e., intelligent and proactive dropping of packets, resulting in prevention of congestion from ever occurring and ensuring a higher degree of fairness between flows. Decisions on actions to be taken are based on expected congestion.

II. RELATED WORK

A. RED (Random Early Detection)

RED algorithm for RED Gateways was first of all proposed by Sally Floyd and Van Jacobson. RED is an active queue management scheme that provides a mechanism for congestion avoidance. Unlike traditional congestion control schemes that drop packets at the end of full queues, RED uses statistical methods to drop packets in a "probabilistic" way before queues overflow. Dropping packets in this way slows a source down enough to keep
the queue steady and reduces the number of packets that would be lost when a queue overflows and a host is transmitting at a high rate.

RED makes two important decisions. It decides when to drop packets and what packets to drop. RED keeps
track of an average queue size and drops packets when the average queue size grows beyond a defined threshold. The average size is recalculated every time a new packet arrives at the queue.

RED uses time-averaging meaning that if the queue has recently been mostly empty, RED will not react to
a sudden burst as if it were a major congestion event. However, if the queues remain near full, RED will assume congestion and start dropping packets at a higher rate.

B. GRED (Gentle RED)

Similar to RED, the GRED algorithm mainly aims to manage and control the congestion networks at the early stage. GRED implements its algorithm by stabilizing the aql at a certain level. GRED employs a similar approach used by RED in calculating the *Dp*. However, GRED utilizes three thresholds, namely, minimum, maximum, and doublemaximum. Generally, GRED reacts with the arriving packets based on one of the following scenarios:

1) When the *aql* at the router is below the *minthreshold*, no packets are dropped.

2) If the *aql* is between the *minthreshold* and *maxthreshold*, the router will drop the arriving packets randomly, similar to RED.

3) If the *aql* is between the *maxthreshold* and the *doublemaxthreshold*, the packets are dropped randomly with higher probability compared with the previous case.

4) If the *aql* is equal or greater than the *doublemaxthreshold*, the GRED router drops the arriving packets with *Dp* equal to one (i.e., arriving packets are dropped).

C. BLUE

BLUE uses flow and queue events to modify the congestion notification rate. This rate is adjusted by two factors: packet loss from queue congestion and link utilization or underutilization. A key difference the BLUE algorithm has from RED is that uses packet loss rather than the average queue length.

BLUE maintains a single probability, Pm, to mark (or drop) packets. If the queue is continually dropping packets due to buffer overflow, BLUE increases Pm, thus increasing the rate at which it sends back congestion notification or dropping packets. Conversely, if the queue becomes empty or if the link is idle, BLUE decreases its marking probability. This effectively allows BLUE to "learn" the correct rate it needs to send back congestion notification or dropping packets.

The typical parameters of BLUE are d1, d2, and freeze time. d1 determines the amount by which Pm is increased when the queue overflows, while d2 determines the amount by which Pm is decreased when the link is idle. Freeze time is an important parameter that determines the minimum time interval between two successive updates of Pm. This allows the changes in the marking probability to take effect before the value is updated again.

D. CHOKe

CHOKe is stateless AQM algorithm similar to RED used incoming packages to punish streams with the highest demand for bandwidth. Just as in the case of RED, there are two threshold values: Minth and maxth. When a new package is arrived the new average queue length is calculated. When the average queue length is less than Minth, all packets are placed in the buffer. When the average queue length is greater than Minth, CHOKe pulls randomly one packet from the FIFO buffer ("CHOKe victim") and verifies if the package comes from the same stream as an incoming packet. If the both packets belong to the same stream, both are removed (this situation is called"CHOKe hit"). Otherwise, the randomly selected package ("CHOKe victim") is returned to the buffer and the arrived packet is placed in the queue with P probability. This probability is calculated in the same manner as in the case of RED algorithm. This event is called the "choke miss".

E. AdaptiveCHOKe

The AdaptiveCHOKe is an extension of CHOKe (CHOose and Keep responsive flows, CHOose and Kill unresponsive flows). It enforces the concept of queue-based and flow information. It is desirable for AQM schemes to act without storing a loss of information otherwise it becomes an overhead and non-scalable. This algorithm also calculates the average queue size of the buffer for every packet arrival. It also indicates two thresholds on the buffer, a minimum threshold minth and a maximum threshold maxth.

The average queue size is compared with these thresholds for every arriving packet. If average queue size is less than minth, every arriving packet is queued. If average queue size is greater than maxth, every arriving packet is dropped. This results in queue size below maxth. When the average queue size is greater than minth, every arriving packet is compared with a randomly selected packet from the queue for their flowid. If they have the same flowid, both are dropped. Otherwise the randomly selected packet is placed in the same position in the buffer. In this algorithm, the arriving packet is dropped with a probability depending on the average queue size.

F. P-CHOKe

P-CHOKe (PiggybackingCHOKe) enforces the concept of queue-based and flow information. In the P-CHOKe algorithm, for every packet is arrival it forward to the gateway. Then select randomly a packet from the queue and compare with arriving packet through gateway. If there is equal flowid, then drop the packet, otherwise forward the packets to the receiver. Then the receiver sends acknowledgement to the router, it collects all the acknowledgements and then sends to the receiver based on their flowid.

III. PROPOSED WORK

Congestion in Internet occurs when the large volume of data is being routed on low bandwidth lines or across networks that have high latency and cannot handle large volumes. Then the result is slowing down of packet movement, packet lost or dropped. When the heavy traffic is occurred the path will be failure and packet will be dropped or lost. In this situation to overcome the above stated problem by using the APAQM Algorithm.

A source node sending a packet to the destination node, during the time of congestion is occurred in a path of the intermediate node then the packet lost or dropped. So, the destination node cannot receive that packet. To overcome this problem by using APAQM (An Alternate Path AQM Congestion Control

Algorithm).The APAQM (An Alternate Path AQM Congestion Control Algorithm) technique is used to finding the alternate path when the packet lost or dropped in a path. Using this APAQM (An Alternate Path AQM Congestion Control Algorithm) the packets are successfully reached the destination.

APAQM (An Alternate Path AQM Congestion Control Algorithm) is proposed by using the OLSR Protocol and A* algorithm. The APAQM (An Alternate Path AQM Congestion Control Algorithm) is finding the best and shortest alternate path during congestion occurred.

The following steps are involved in APAQM algorithm

Step 1: Start Procedure

Step 2: Analyzing the route

Step 3: Selecting the path

Step 4: Packet arrival

Step 5: Queue length analysis

Step 6: Packet drop

Step 7: Alternative path selection (using OLSR and A*)

Step 8: Forward the packet to the destination

Step 9: Data is received then acknowledgement send to the source node

Step 10: End Procedure

IV. SCREEN SHOTS

Node flooding

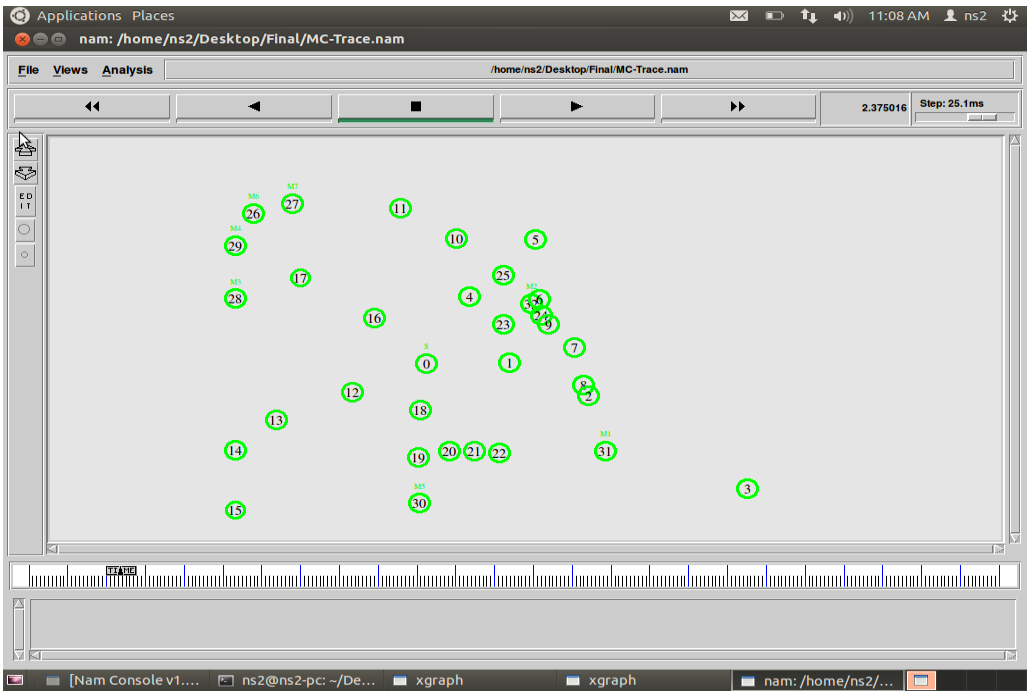

Fig.1 Example for Node flooding

Packet dropping occurred

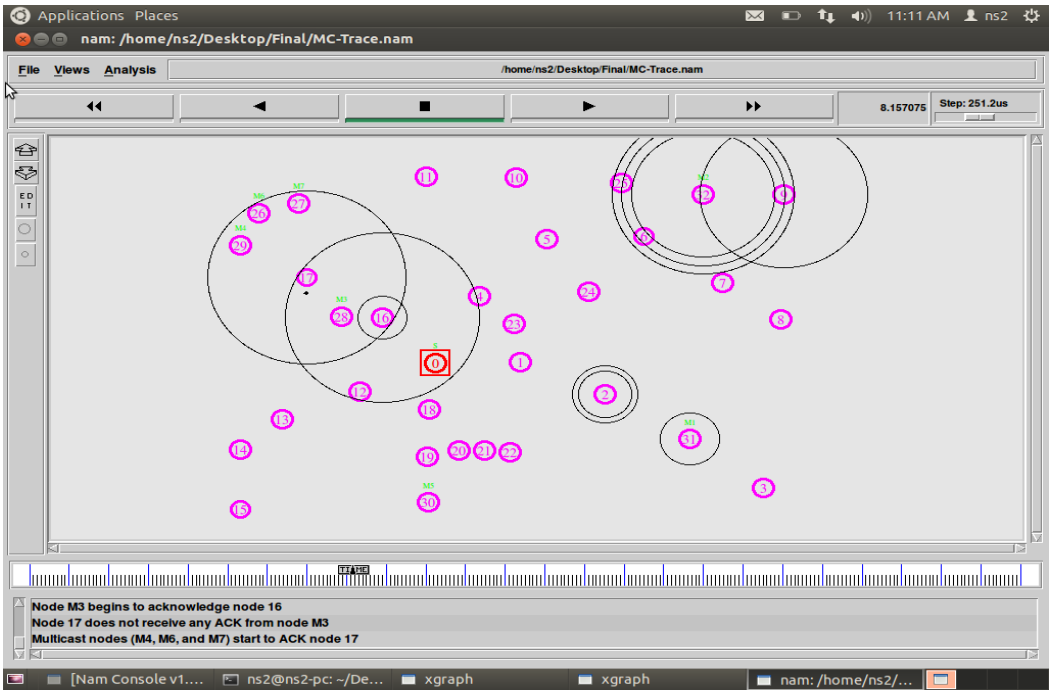

Fig.2 Example for packet dropping

Finding the alternate path

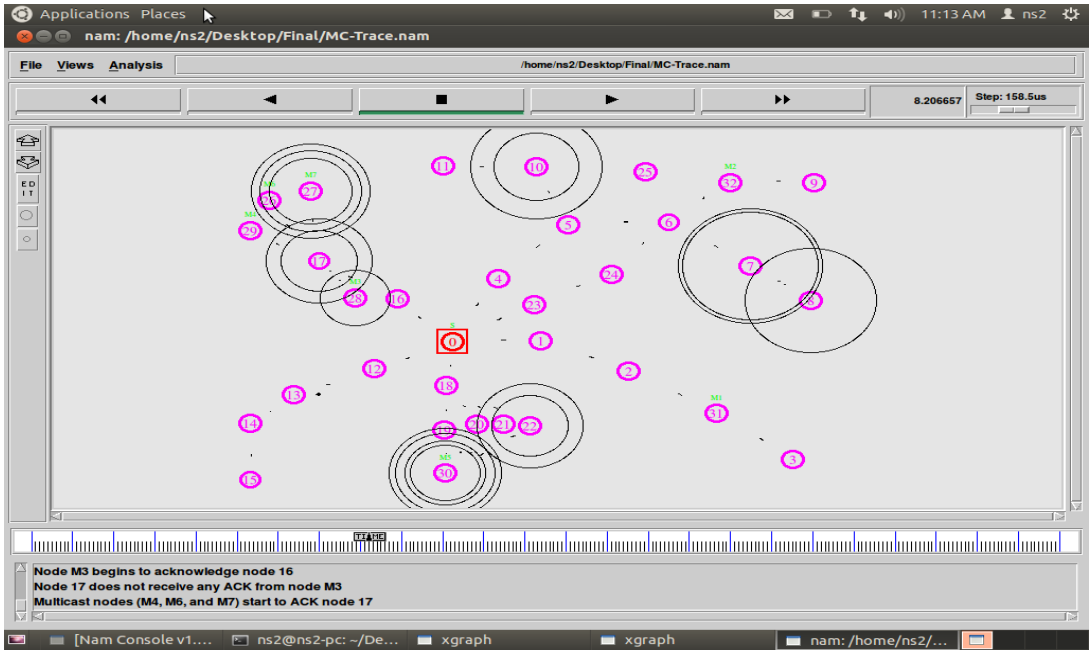

Fig.3 Example for finding alternate path

Acknowledgement sending

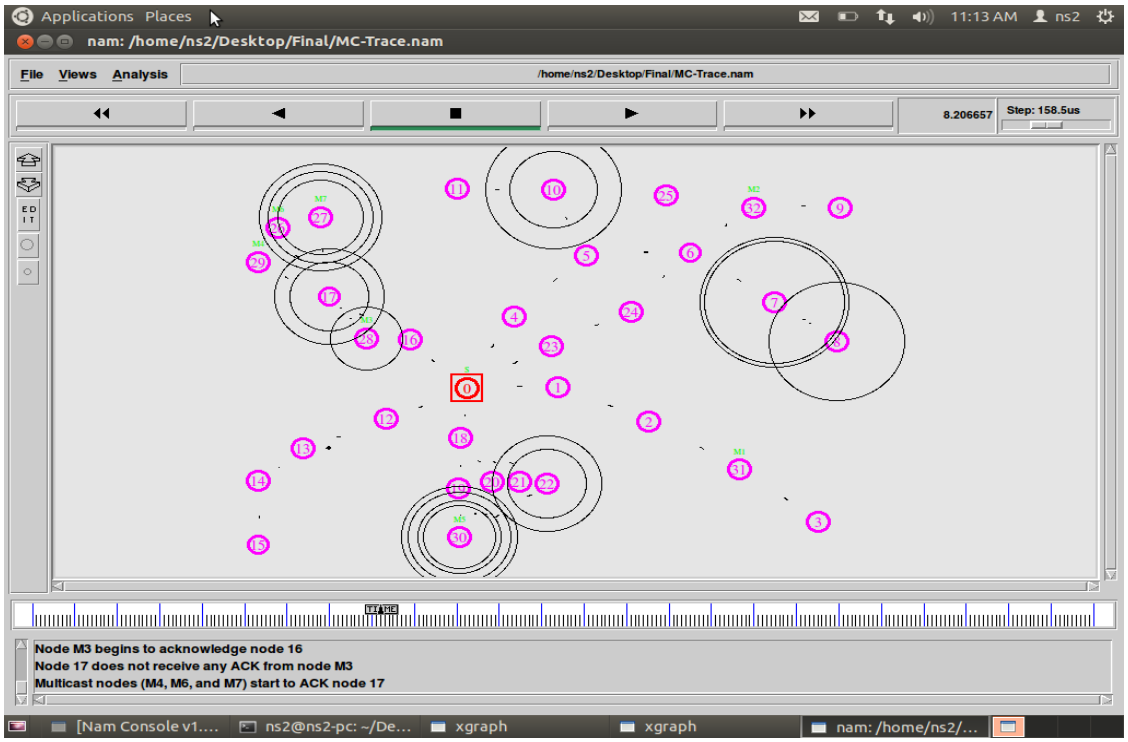

Fig.4 Example for acknowledgment sending process

Packet delivery ratio

The packet delivery ratio for the simulation result stated here is calculated regarding the following formula in relation to time.

$$\sum \text{Number of packet receive} / \sum \text{Number of packet send}$$

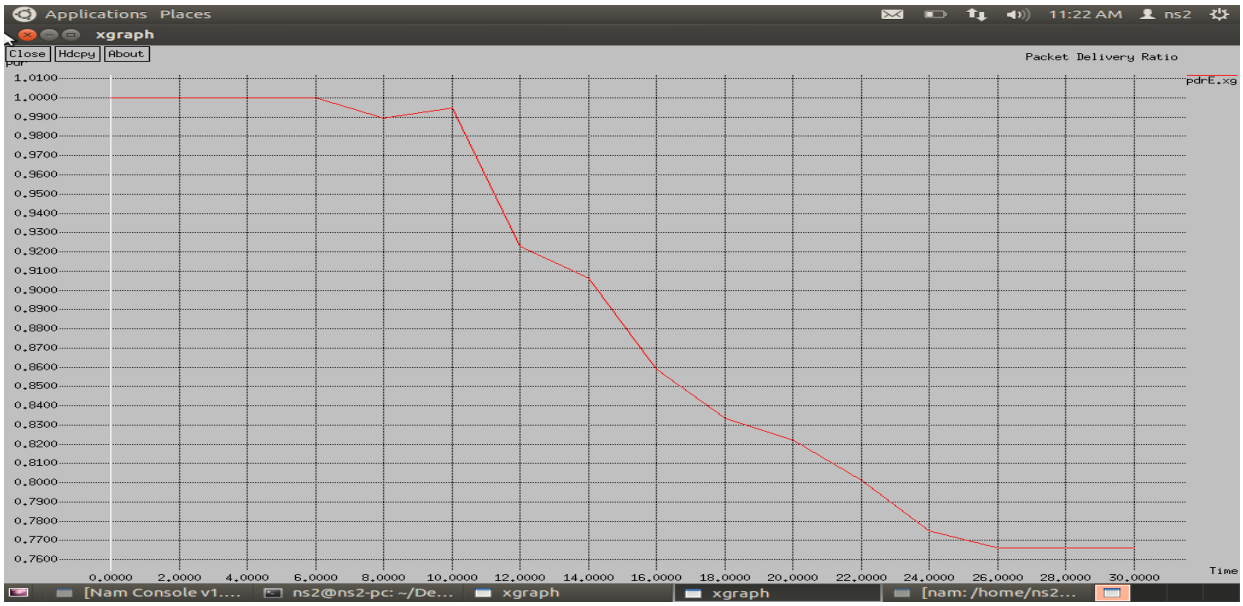

Fig.5 Example for packet delivery ratio

Average energy

The energy level for the simulation result stated here is calculated regarding the following formula in relation to time.

Average energy: Total packets – Lost arrival packet

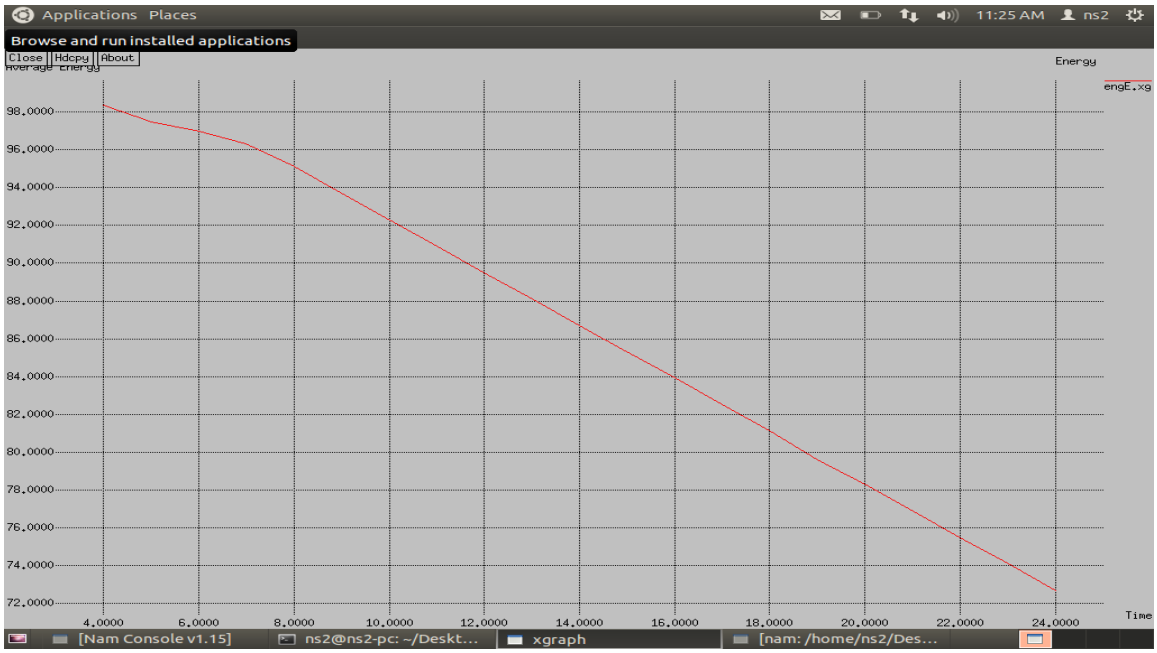

Fig.5 Example for average energy ratio

V. CONCLUSION

The Experimentation is done by Network Simulator2. From the Simulation it is found that APAQM obtains high Packet delivery Ratio and average energy ratio is good. APAQM reduce packet drop and system overhead.

REFERENCES

[1] Alshimaa H.Ismail, Zeiad Elsagheer, and I.Z.Morsi,"Survey on Random Early Detection Mechanism and Its Variants", IOSRJCE, vol.2, Issue.6, Aug-2012.
[2] G. Sasikala, E. George Dharma Prakash Raj, "P-CHOKe: A Piggybacking-CHOKe AQM Congestion Control Method", IJCSMC, Vol. 2, Issue. 8, August 2013.
[3] Addisu Eshete, Yuming Jiang, "Generalizing the CHOKe Flow Protection", ELSEVIER, Computer Networks 57(2013), pg: 147 to 161.
[4] Mahmoud Baklizi, Hussein Abdel-Jaber, Mosleh M. Abu-Alhaj, Nibras Abdullah, Sureswaran Ramadass and Ammar Almomani " Dynamic Stochastic Early Discovery: A New Congestion Control Technique To Improve Networks Performance" ICIC International, vol.9,number 3, March 2013,pg: 1113-1126.
[5] Omid Seifaddini, Azizol Abdullah, and Hamid Vosough "RED, GRED, AGRED CONGESTION CONTROL ALGORITHMS IN HETEROGENEOUS TRAFFI TYPES" ICOCI, 2013, 28-30 August, pg:139-144.
[6]Adam Domanski, Joanna Domanska, Jerzy Klamka, "Analysis of CHOKe – Family Active Queue Management" ISSN 1896 – 5334, Vol.25,Issue 1,2013,pg:49-66.
[7] Satnam Singh, Balveer Singh "Performance & Analysis of REM, RED, & GREEN AQM Algorithms in Congestion Control" IJARCSSE, Vol.3, Issue 11, Nov 2013, pg: 184-188.
[8] Jun Huang, Donggai Du, Qiang Duan, Yanguang Zhang, Yanxiao Zhao, Hongshan Luo, Zhuoqi Mai and Qian Liu "Modeling and Analysis on Congestion Control for Data Transmission in Sensor Clouds" International Journal of Distributed Sensor Networks, 2014, Aritcle ID 453983, 9 pages.

