International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 4 Issue 09, September-2015

Bandwidth Congestion Control using Page Redirection Approach via Virtual Circuit/Server: An Alternative Approach to Improve Network Performance

Yekini Nureni Asafe
Computer Engineering Department
Yaba College of Technology
Lagos, Nigeria

Okikiola Folasade Mercy
Computer Technology Department
Yaba College of Technology
Lagos, Nigeria

Akinyele Akinleye Okedola
Computer Engineering Department
Lagos State Polytechnic
Ikorodu, Nigeria

***Abstract* — In a Computer Internetwork, congestion control is concerned with allocating the resources in a network such that the network can operate at an acceptable performance when the demand exceeds or is near the capacity of the network resources. These resources include bandwidths of links, buffer space that is memory, and also processing capacity at intermediate nodes. Although resource allocation is necessary even at low head, the problem becomes more important as the load increases because issues of fairness and low overhead become increasingly important. Without proper congestion control mechanisms, the throughput may be reduced considerably under heavy load. In this paper our concern is on bandwidth congestion controlled during the peak period in a busy website, the author is of the opinion that creating different websites on different servers, or mirror (website) the application on different servers and redirecting different users to different servers during peak period can solve the bandwidth congestion problem. We simulated a congestion control scheme which is based on page redirection. PHP which is used to manipulate the database and creating a dynamic page, JavaScript which is used to enable quick communication with the server side scripting language, XHTML serves as a basis for web development and CSS for beautification and styling the webpage. Dreamweaver used in designing and editing codes respectively.**

Keywords— Bandwidth, Computer Internetwork, Congestion Control, Page Redirection, Virtual Circuit/server.

I. INTRODUCTION

Congestion occurs on shared networks when multiple users contend for access to the same resources such as bandwidth, buffers, and queues. Congestion in a network may occur when the load on the network is greater than the capacity of the network. Congestion control is concerned with allocating the resources in a network such that the network can operate at an acceptable performance level when demand exceeds or is near the capacity of the network resources. These resources include bandwidths of links, buffer space that is memory, and also processing capacity at intermediate nodes [1].

A recent research has shown that the popularity of the internet has augmented the need for more bandwidth throughout all tiers of network. . So far the internet has survived but there have been a number of incidents throughout the years where serious problems have disabled large part of the network. There has been some serious discussion given to the potential of a large-scale internet collapse due to network overload.

When too much traffic is offered, congestion sets in and performance degrades sharply, for example in an educational institution website, where many users from different part of the world log on-to the website to check quick links (such as admission status, POST UTME schedule for the year, online application, payment format etc.) and also to carry out various tasks as touching the college without regard for impending or existing congestion. As more users enter the freeway, congestion gets worse. Congestion control is the efforts made by network nodes to prevent or respond to overload conditions. Also, it is intended to keep a fast sender from sending data into the network due to lack of resources in the network (e.g. available link capacity, router buffers)

Congestion occurs when the demand is more than the available resources. Therefore, it is believed that as resources become less expensive, the problem of congestion will be solved automatically. This has led to the following myths by [2].

- Congestion is caused by a shortage of buffer space and will be solved when memory becomes cheap enough to allow infinite large memories.
- Congestion is caused by slow links. The problem will be solved when high-speed links become available.
- Congestion is caused by slow processors. The problem will be solved when the speed of the processor is improved.
- If not one, then all of the above developments will cause the congestion problem to go away.

Contrary to these beliefs, without proper protocol redesign, the above developments may lead to more congestion and thus reduce performance. The following discussion explains why.

The congestion problem cannot be solved with a large buffer space. Cheap memory has not helped the congestion problem. It has been found that networks with infinite memory switch are susceptible to congestion as networks with low memory switches [3].

Congestion is a dynamic problem that cannot be solved with static solutions alone. We need protocol designs that protect networks in the event of congestion, and that is the major objective of this research work. We simulated a congestion control scheme which is based on page redirection. PHP which is used to manipulate the database and creating a dynamic page, JavaScript which is used to enable quick communication with the server side scripting language, XHTML serves as a basis for web development and CSS for beautification and styling the webpage. Dreamweaver used in designing and editing codes respectively.

A. Statement of Problem

Dynamic regulation of the computer internetwork resources has been an issue of concern to network administrator and computer engineer in the present due to computer user's demand for internetwork, and limited bandwidth availability. This research work is working towards fair, effective and dynamic regulation of network resources for avoiding the loss of Quality of Service (QoS).

B. Purpose Of The Study

The purpose of this study is to implement a Master-slave Architecture (Virtual Circuit/server) which is a means of transporting data over a packet switched network in such a way that it appears as though there is dedicated physical layer link between the source and the destination end systems of the data..

C. Objective of the Study

The general objectives of this research work are:

- To guarantee stable operation of packet networks in order to avoid congestion collapse.
- To keep networks working in an efficient status. E.g. high throughput, low loss, low delay and high utilization.
- To provide fair allocation of networks bandwidth among competing flows in steady state.
- To ensure quality of service.
- To allocate resources for best-effort service.

D. Scope of the Study

The scope of this study is on network redirection and how it can be monitored in any organization. It will be narrowed down to the network analysis of the data center (a data center is a small unit offering/rendering cloud service to one or more organizations).

II. LITERATURE REVIEW

A. What Is Network Congestion?

Scholars have defined network congestion in one way or the other, for example (Keshar, Srinivasan, 1991) defines congestion as the loss of utility to a network user due to high loads. Congestion occurs in data networks when transfer requests exceed the actual overall capacity of the network nodes. This may translates into degraded quality of service for the user. It may also result in packet loss, delay or blockage of new connections. (Sandvile, 1997). In a network with shared resources, where multiple senders compete for link bandwidth, it is necessary to adjust the capacity allocated to and the data rate used by each sender in order not to overload the network (Chen, 2011).

B. Congestion Control of TCP/IP Networks

An excessive amount of packets arriving at a network bottleneck leads to many packet drop outs. The lost packets often trigger retransmission, which implies that even more packets are sent into the network. Thus network congestion can severely deteriorate the network throughput. Such situations have occurred in the early Internet, leading to the development of the TCP congestion control mechanism [4]. In the 80's TCP/IP links on the Internet became increasingly congested and a new concept of "conservation of packets" was presented where and when "conservation of packets" is observed then TCP flows are generally stable. This "conservation of packets" was implemented by a congestion window where further packets would not be sent once the congestion window was full until another packet was removed. This congestion window could be dynamically realized as the connection was established and as conditions change. These changes are widely credited with preventing ongoing TCP collapse. Over the past few years, large amount of research have been conducted for developing a combination of end-to-end rate (window) adaptation and network-layer dropping or signaling techniques for the TCP/IP and ATM networks to ensure that the network can operate without collapsing due to congestion. However, it has become clear that existing TCP congestion avoidance mechanisms, while necessary and powerful, are not sufficient to provide good service in all circumstances [5]. Most of the current congestion control methods are based on intuition and ad hoc control techniques together with extensive simulations to demonstrate their performance. The problem with these approaches is that very little is known about why these methods work and very little explanation can be given about when they fail due to the ad hoc nature of many of these schemes. In the existing TCP/IP networks, the most studied and used mechanisms to address the congestion problem are the Active Queue Management (AQM) algorithms. These techniques attempt to prevent congestion and regulate the queue length by sending congestion signals (i.e., dropping packets) in a proactive manner. The AQM algorithms are based on a first-in-first out (FIFO) queue system, and the basic idea is to calculate and update the packet drop probability. Therefore, the incoming packets to a queue can be dropped before the buffer overflows.

During the past few years, other advanced algorithms have been studied by using the queue length and input rate jointly to achieve better performance. The AQM algorithm is the REM, which uses a linear combination of the queue mismatch and input rate mismatch to calculate the drop probability, and the input rate mismatch is equivalently simplified to the queue variance between two adjacent queue length samples [6].

C. Congestion Control of Sensor-Actuator Networks

During the past few years, sensor networks have received extensive attention in research and applications due to their capability of self-organization and distributed computing. However, sensors are passive devices that can only collect data from the environment without interaction [7].

Therefore, actuators are introduced to make decisions and perform appropriate actions according to the sensor measurements, which lead to the notion of sensor-actuator networks (SAN). The actuators are usually highly mobile and need to carry out execution activities. Hence, it is required that the actuators have powerful battery, processor and memory to perform both decision making and actuation. As mentioned earlier, a network of multi-agent systems (NMAS) is composed of a number of distributed intelligent agents. Each agent may consist of a group of sensors, decision makers' and actuators. The exchange of information and process of decision making are made autonomously. In this sense, the NMAS can be viewed as a wireless sensor-actuator network (SAN).

Among the many research issues in sensor-actuator networks, congestion control is one of the most predicament problems that need to be solved. Although, the existing TCP/IP congestion control algorithms perform quite well on the Internet, the unique properties of the sensor-actuator networks requires the design of appropriate protocols and protocol stacks in general, and of a congestion control mechanism in particular [8]. The congestion control algorithms for SAN need to be highly energy-efficient, to prolong system lifetime, improve fairness, and improve QoS in terms of throughput (or link utilization)and packet loss ratio along with the packet delay. A SAN consists of one or more sinks and perhaps tens or thousands of sensor nodes scattered in a large area of interest. Congestion restraint generally follows two steps: congestion detection and congestion control.

Congestion Detection. Accurate and efficient congestion detection plays a vital role in congestion control in sensor networks. There are various detection techniques that have low costing terms of energy and computation complexity. An energy efficient congestion detection method known as CODA discussed in Congestion detection and avoidance in sensor networks deals with various degrees of congestion depending on the sensing application. In a hop-by-hop back-pressure congestion detection, if the sink is congested, back-pressure spatially spreads the congestion and helps alleviate congestion quickly. In addition, hop-by- bop control supports in-network data processing. Once congestion is detected, the receiver will broadcast a suppression message to its neighbors. The hop-by-hop back-pressure cans immediately response to the congestion at the intermediate node without incurring the round trip delay that reduces feedback effectiveness [9].

D. Congestion Control approach

Once the congestion at a node is detected, the node informs its source nodes of the congestion and a series of actions are performed at the node side or at the sink side to remedy the congestions. A node priority index is introduced to reflect the importance of each node. There are two congestion control approaches [10].

- Short term control: Here in this control method when node experience congestion, its immediate downstream node split the real-time traffic onto its alternate upstream node in proportion to their weight factor. This approach will eventually carry the newly created real-time data flows at a slower rate along the primary route, allowing the congested node to be relieved and thus alleviate the congestion.
- Long-term control: It is called long term congestion control in that the source node will dynamically adjust to the changing conditions and select the best upstream node as its primary route to send further packets. Consequently, both the real-time and non-real-time data flows will follow the changed or updated primary route. The pump slowly fetch quickly (PSFQ) method is another method worth mentioning that is developed for the sensor networks [11].

F. Congestion Problem and Solutions

Congestion can be classified as a single resource problem or a distributed resource problem depending on the number of resources involved. See figure 1.

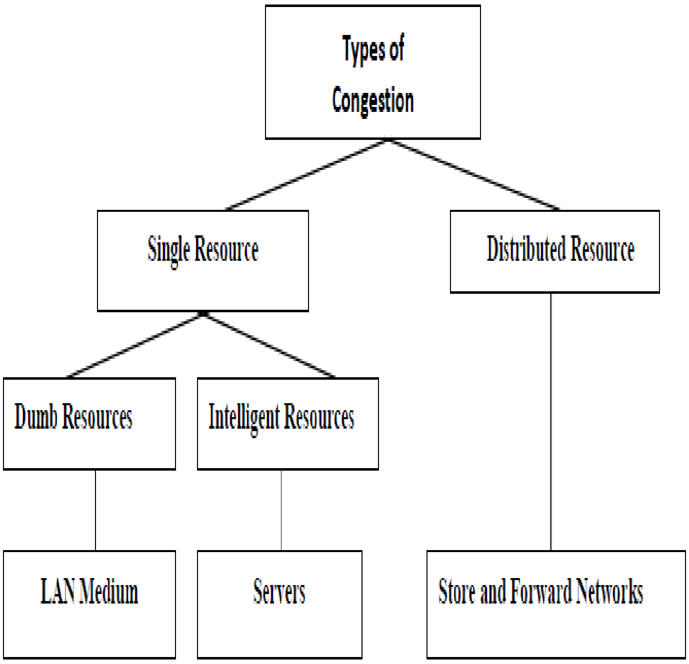

Fig. 1. Congestion resources

The single resource involved here may be a dumb resource, such as a LAN medium. In this case all the intelligence required to solve the congestion problem has to be provided by the users. Various LAN access methods, such as Carrier Sense Multiple access with collision Detection (CSMA/CD), token access, register insertion, and so on, are examples of solutions to the problem of single, dumb resource congestion. If the resource is intelligent (for example, a name server), it can allocate itself appropriately. The problem is more difficult if the resource is distributed, as in the case of a stored-and- forward network. For example, considering the links as the resources, the user demands have to be allocated so that the demand at each link is less than its capacity.

III. RESEARCH METHODOLOGY & DESIGN

A. Description Of the Existing System

In the existing system, many systems-n are connected to a server at the same time for a purpose. All the systems (up to system n = 5) connect to the server at same time for different purposes either to access data or to post data see figure 2. The server is a computer system, and it has to process one data at a time.

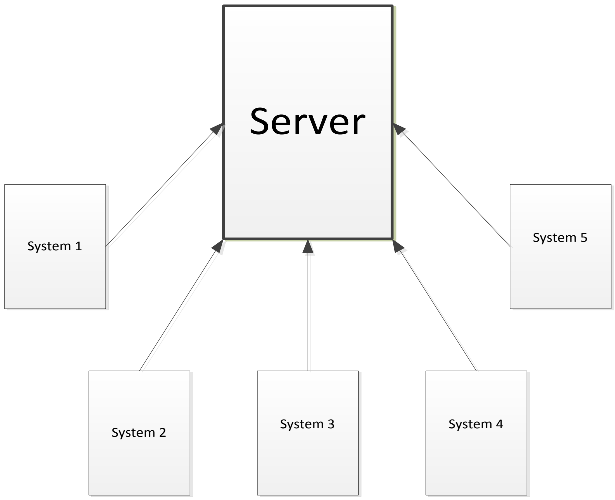

Fig. 2. Three - Tier – Architecture

Once the server is attending to one, the rest will be waiting, and this could lead to bandwidth congestion or deadlock in networking, which can result to:

- Slow data access rate
- Delay to the client
- Delay in retrieving information by the client side from the server side.
- Maximum execution time error for pages accessed on the server

All this and more makes a website to look nonfunctional and unreliable before the users accessing it. It makes users lose interest in the application.

B. *Proposed System*

Based on the challenges identify from the existing system, we proposed a new system which is not going to be the total design of the case website but, it will be a module that can be integrated to the website to enable page redirection during peak periods. the module will be configured in such a way that a particular activities A will be redirected to server X, activities B to server Y, while other activities will be on server Z based on the inference made by the central server C .

The central server C will handle the redirection. The central server C will create a mirror view of the website on the other servers along with the various data required. The central server holds only the database and few source codes written in CGI or a server side language such as PHP, python, java, node is and the likes along with Hypertext Markup Language. Once a request is sent to the central server C, it gets the counter number for the page requested, and check if it exceeds a particular congestion number, if true, it redirect to the corresponding server that will handle the task.

There will be an administrative panel that allows the setting of these parameters. The administrative panel will count the number of visitors accessing the server in real time, making it possible to stop a particular IP that the school doesn't want on their site. Each of the sub servers will be running Internet Service Provider (ISP) server on them. Creativity emerges as the data or user tracking can be trapped on real time bases to enable the system function as expected.

All servers should handle task referenced from the central server C, this will in turn reduce bandwidth congestion and add the following advantage to it:

- A backup process is trivial because virtually all servers comprise of the same information.
- The central server is lightweight due to reduction of traffic.
- Virtually all servers handle the same number of request due to the counter method used in the central server.
- Power outage and other hazardous events do not hinder the flow of transaction.

The proposed system will be built with particular reference to the flowchart given in figure 3 below.

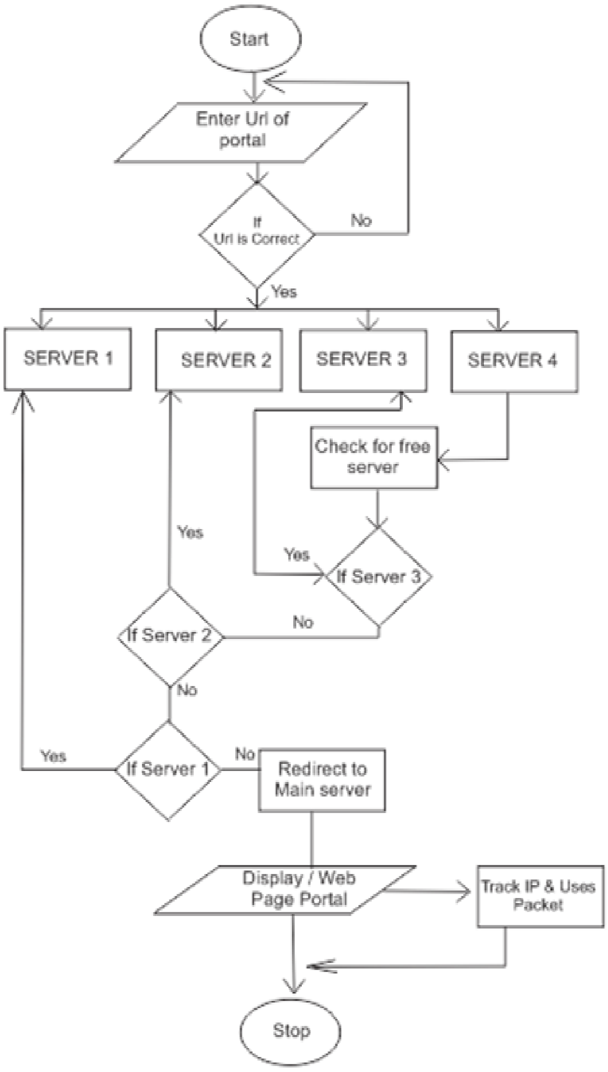

Fig. 3. Proposed System Flowchart

C. *Design, Coding, and Testing*

This is an important process in which the solutions to the problems are identified. It shows the blue print of the new system and how it's going to work. The method used in this design is the 'Three - Tier – Architecture'. See figure 4.

Fig. 4. Three - Tier – Architecture

The three-tier architecture is a framework that shows how a dynamic web works. It shows the three views which are

- The client view that is built with HTML, CSS, and JavaScript,
- The server side script and
- The database.

The client side form submit query or request to the scripting page, see figure 5, while the scripting page verifies the report and if found ok (Processing), it sends the request to the database vice-versa.

Fig. 5. Control Panel

D. Input Design

Input design will be used to design forms that are used in the collection of data for the online application. Data are entered via the keyboard into the system. Some rules are created which users have to follow which in return gives the user the appropriate answer when filling the forms.

- Admin registration: this is the form that allows the registration of admin (Staff that manages the system). These admin staffs are the ones that will set the server IP, redirect clients and specify the IP's that should not access the server.
- Admin Login: In order for the registered staffs or admin to gain control of the system, the user has to login via an authenticated credential such as username and password, staff ID and the likes. The input fields are the textbox for the username and the password.
- Server Settings and IP Configuration: the links to be redirected to have to be configured, this section allows the configuration of those links.
- IP stop: Stop IP is a very vital area in every system, this is done on redirect parameter link which serves as the default page for most blocked IPs. When an IP is sending a wrong signal, it can be stopped from the administrative panel. Once an IP is stopped or blocked it will be using this page as a base.

E. Database Design

Database name is congestion-c. The database /DB/ Dbase to be used is MySQL. This database will be located in a WAMP (Window Apache Mysql PHP) server.

TABLE I. TBL_ADMIN

Field	Type	Size	Primary Key
Serial	Int	10	Yes
Code	Varchar	15	No
Name	Varchar	39	No
Email	Varchar	39	No
Pass	Varchar	39	
Phone	Varchar	39	
Address	Text		
Sex	Varchar	8	

The table I named tbl_admin will be used to store the various Administrator details:

TABLE II. TBL_BRAIN

Field	Type	Size	Primary Key
Serial	Int	10	Yes
Code	Varchar	15	No
URL	Text		No
IP	Varchar	25	No
ServerName	Text		No
Admin	Varchar	39	No
Date	Timestamp		On_current_time

Tbl_brain: this table helps to store the details of all servers in use and the URL to get to them.

TABLE III. TBL_BRAIN

Field	Type	Size	Primary Key
Serial	Int	10	Yes
Code	Varchar	15	No
URL	Text		No
IP	Varchar	25	No
ServerName	Text		No
Admin	Varchar	39	No
Date	Timestamp		On_current_time

Tbl_stp: this table shows those IP that should not access the website.

Coding: There is no software without a source code. The source code is the spirit of the system. In this case, server side language such as PHP, HTML, and Asynchronous JavaScript was used to create the web application. The code is written using Dreamweaver editor as the editor.

F. Processing Design

This stage is focused on how input data must be processed before sending to the database for storage. This is the link between the input and data file of any system. Data processing usually takes place in the central processing unit (C.P.U) of the system. Validation is done to ensure that there is no wrongly entered data. This stage is performed based on certain conditions that must have been included in the code. The system starts by the user entering the web link in the URL. Once the URL link is correct, it checks for the less populated server. First, it checks Server A through Server C, to confirm if there is any less populated one in order to reduce traffic.

On getting the right one it displays the web portal for the client. It tracts the IP address of the user and packet data strength on the site. See figure 6.

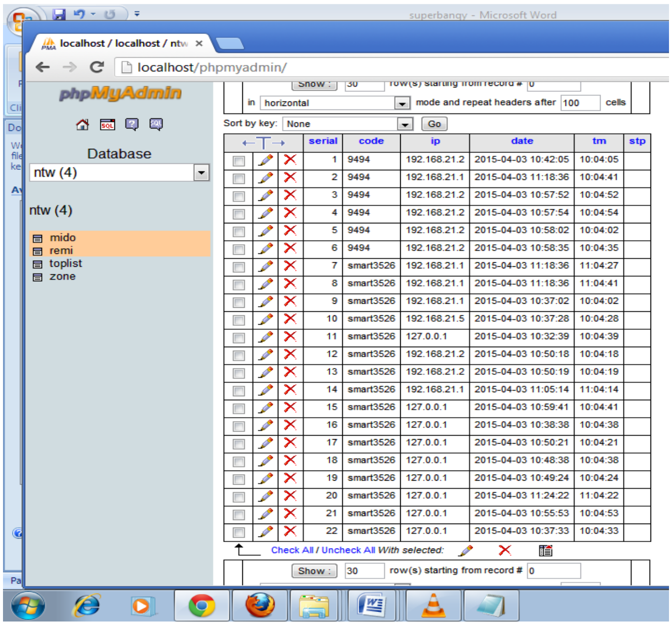

Fig. 6. Congestion Controlling Processing Interface

IV. CONCLUSION

Congestion is not a static resource shortage problem; rather it is a dynamic resource allocation problem. Simply placing more memory in the nodes, or creating faster links, or faster processors will not solve the congestion problem. In any intermediate system where the total rate is higher than the output rate, queues will build up. Therefore, explicit measures to ensure that the input rate is reduced should be built into the protocol architectures. Congestion control is not a trivial problem because of the number of requirements, such as low overhead, fairness, responsiveness, and so on. In particular, congestion schemes are called on to work under unfavorable network conditions and are required to ensure that the result is socially optimal. The proposed research work will demonstrate the implementation of system on how bandwidth congestion can be controlled in different fields. The conclusion is that congestion is a dynamic problem. It cannot be solved with static solutions alone. We need protocol designs that protect networks in the event of congestion. The explosion of high- speed networks has led to more unbalanced networks that are causing congestion. In particular, packet loss due to buffer shortage is a symptom, not a cause of congestion. And this project work has guaranteed a proper recommendation to ensure adequate functionality on the network.

REFERENCES

[1] Chen.L, LowS. H., Chiang M., & Doyle J. C., (2006).Cross-Layer Congestion Control,

[2] Routing and Scheduling Design in Ad Hoc Wireless Networks". In 25th IEEE Inter-National Conference on Computer Communications, (Infocom), (Pp.1-13).

[3] Athuraliya, S., Low. S., Li. V., & Yin. Q.,(2001). Rem: "Active Queue Management". InIEEE Network Magazine, Vol. 15 of 3, May, (Pp. 48-53).

[4] Hollo, C. t., MisraV., Towsley D., & Gong W., (2003). "On designing Improved Controllers for AQM Routers Supporting TCP Ows ". In Proceedings of Twentieth Annual JointConference of the IEEE Computer and Communications Societies (Infocom) Vol. 3, April, (Pp. 1726-1734).

[5] Wan, C. Y., Eisenman, S. B., & Campbell. T. (2003)"Coda: Congestion Detection and Avoidance in Sensor Networks". In Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (Sensys), (Pp. 266-279).

[6] Wang, H., W. Y, Mao, Y., Jing, Y., & Guo, L.(2008)."Robust Congestion Control For Uncertain Time-Varying Delay Network Systems". In Control and Decision Conference (Ccdc), Pp. 739-743).

[7] Zhang, N., Yang, M., Jing, Y., & Zhang, S. (2009)."Congestion Control for DiffServ Network Using Second-Order Sliding Mode Control". In IEEE Transactions on IndustrialElectronics, Vol. 56 of 9, (pp. 3330-3336).

[8] R. Chen & Khorasani, K.(2007). "An Adaptive Congestion Control Technique for Networked Control Systems". In IEEE Conference on Industrial Electronics and Applications, (pp. 2791-2796.).

[9] Jacobson, V. (1988). "Congestion Avoidance and Control". In ACM Annual Conference of Special Interest Group of Data Communications, August, (Pp. 314-329).

[10] Jain, R., Ramakrishna, K.K. (1988), "Congestion Avoidance in Computer Networks With a Connectionless Network Layer". In Innovations In Internetworking, C. Partridge, Ed., Norwood, Ma: Artech House, (Pp. 140-156).

[11] Ran Jain. (1986). "Divergence of Timeout Algorithms for Packet Retransmission". Proc. 5th Annual Int'l. Phoenix Conference on Computer and Communication. Scottsdale. Az. (Pp. 174-179).

[12] Nagle, J. (1987). "On Packet Switches with Infinite Storage". IEEE Trans. on Communication. Vol. COM-35, No. 4, April, (Pp. 435-438).

[13] Mishra, P., & Kanakia, H.(1992). "A Hop By Hop Rate-Based Congestion Control Scheme". In Conference Proceedings on Communications Architectures & Protocols (Sigcomm), (Pp.112-123)