Volume No:1, Issue No:7 (December-2015) ISSN No : 2454-423X (Online)

International Journal of Research in Advanced Computer Science Engineering

A Peer Reviewed Open Access International Journal

www.ijracse.com

Intelligent Network Traffic Control in High Speed Networks Using Fuzzy Logic

Karunakar Saka
M.Tech,
Department of CSE,
Sri Vaishnavi College of Engineering, Srikakulam.

B.V.Ramana
Assistant Professor,
Department of CSE,
Sri Vaishnavi College of Engineering, Srikakulam.

Sahu Smita Rani
Assistant Professor & HOD,
Department of CSE,
Sri Vaishnavi College of Engineering, Srikakulam.

Abstract:

The tremendous growth of the internet and the advances of computer technology have been pushing forward computer networks for high speed and broad bandwidth. Congestion, being a non-linear and dynamic problem requires robust, possibly intelligent, control methodologies to obtain satisfactory performance. Fuzzy logic helps to increase throughput, reduction in packet drop and delays. In this paper, we illustrate the power of the methodology by the successful application of fuzzy based congestion control in the two diverse networking technologies of ATM and TCP/IP.. By the fuzzy logic technique, QoS (Quality of Service) in communication is assured by good performances of our scheme such as max-min fairness, low queuing delay and good robustness to network dynamics. The conclusion is that the results and comparisons have verified the effectiveness and made a created a new benchmark that our traffic management scheme using fuzzy-logic can achieve better performance than the existing protocols that depend entirely on the estimation of network parameter.

Keywords:

Congestion control, fuzzy logic control, quality of service, max-min fairness, robustness, traffic management.

1. Introduction:

Congestion is an important issue that can arise in packet switched network. Congestion is a situation in Communication Networks in which too many packets are present in a part of the subnet, performance degrades. Congestion in a network may occur when the load on the network (that is, the number of packets sent to the network) is greater than the capacity of the network (that is, the number of packets a network can handle).. Traffic congestion control is one of the effective approaches to manage the network traffic [1], [2].Historically, TCP (Transmission Control Protocol) Reno [3], [4] is a widely deployed congestion control protocol that tackles the Internet traffic. It has the important feature that the network is treated as a black box and the source adjusts its window size based on packet loss signal [5]. However, as an implicit control protocol, TCP encounters various performance problems (e.g., utilization, fairness and stability) when the Internet BDP (Bandwidth-Delay Product) continues to increase.

These have been widely investigated with various proposed solutions such as the AQM (Active Queue Management) schemes [6]–[10] whose control protocols are also implicit in nature. As an alternative, a class of explicit congestion control protocols has been proposed to signal network traffic level more precisely by using multiple bits. Examples are the XCP [6], RCP [11], JetMax [12] and MaxNet [13]. These protocols have their controllers reside in routers and directly feed link information back to sources so that the link bandwidth could be efficiently utilized with good scalability and stability in high BDP networks.

Specifically, JetMax and MaxNet signal n congestion by providing the required fair rate or the maximum link price, and then the final sending rate is decided by sources according to some demand functions or utility functions. XCP feeds back the required increment or decrement of the sending rate, while RCP directly signals sources with the admissible sending rate according to which sources pace their throughput. The advantages of these router-assisted protocols are that 1) they can explicitly signal link traffic levels without maintaining per-flow state, and 2) the sources can converge their sending rates to some social optimum and achieve a certain optimization objective [12].

However, most of these explicit congestion control protocols have to estimate the bottleneck bandwidth in order to compute the allowed source sending rate or link price. Recent studies show that misestimating of link bandwidth (e.g., in link sharing networks or wireless networks) may easily occur and can cause significant fairness and stability problems [14], [15]. There are some latest protocols on wireless applications such as QFCP (Quick Flow ControlProtocol) [16] and the three protocols called Blind, Errors' andMAC [17]. They have improved on the estimation error while having high link utilization and fair throughput.

However, they still have the fundamental problem of inaccurate estimation resulting in performance degradation. In addition, their bandwidth probing speed may be too slow when the bandwidth jumps a lot. Also, they cannot keep the queue size stable due to Oscillations, which in turn affects the stability of their sending rates. The contributions of our work lie in: 1) using fuzzy logic theory to design an explicit rate-based traffic management scheme (called the Intel Rate controller) for the high-speed IP networks; 2) the application of such a fuzzy logic controller using less performance parameters while providing better performances than the existing explicit traffic control protocols; 3) the design of a Fuzzy Smoother mechanism that can generate relatively smooth flow throughput; 4) the capability of our algorithm to provide max-min fairness even under large network dynamics that usually render many existing controllers unstable.For the remainder of the paper, the following notations and symbols pertain.

Existing System:

Existing System shows that misestimation of link bandwidth (e.g., in link sharing networks or wireless networks) may easily occur and can cause significant fairness and stability problems. There are some latest protocols on wireless applications such as QFCP (Quick Flow Control Protocol) and the three protocols called Blind, ErrorS and MAC. They have improved on the estimation error while having high link utilization and fair throughput. However, they still have the fundamental problem of inaccurate estimation resulting in performance degradation. In addition, their bandwidth probing speed may be too slow when the bandwidth jumps a lot. Also, they cannot keep the queue size stable due to oscillations, which in turn affects the stability of their sending rates.

Proposed System ;

This paper propose a distributed traffic management framework, in which routers are deployed with intelligent data rate controllers to tackle the traffic mass. Unlike other explicit traffic control protocols that have to estimate network parameters (e.g., link latency, bottleneck bandwidth, packet loss rate, or the number of flows) in order to compute the allowed source sending rate, our fuzzy-logic-based controller can measure the router queue size directly; hence it avoids various potential performance problems arising from parameter estimations while reducing much consumptionof computation and memory resources in routers. As a network parameter, the queue size can be accurately monitored and used to proactively decide if action should be taken to regulate the source sending rate, thus increasing the resilience of the network to traffic congestion.

Procedure:

(1) Upon the arrival of a packet, the router extracts Req_rate from the congestion header of the packet.
(2) Sample IQSize q(t) and update e(t) and g(e(t)).
(3) Compute the output u(t) and compare it with Req_rate.
(3a) If u(t) < Req_rate, it means that the link does not have enough bandwidth to accommodate the requested amount of sending rate. The Req_rate field in the congestion header is then updated by u(t).
(3b) Otherwise the Req_rate field remains unchanged.
(4) If an operation cycle d is over, update the crisp output u(t) and the output edge value of D.

2. TRAFFIC MANAGEMENT PRINCIPLE AND MODELING:

We consider a backbone network interconnected by a number of geographically distributed routers, in which hosts are attached to the access routers which cooperate with the core routers to enable end-to-end communications. Congestion occurs when many flows traverse a router and because it's IQSize (Instantaneous Queue Size) to exceed the buffer capacity, thus making it a bottleneck in the Internet. Since any router may become bottleneck along an end-to-end data path, we would like each router to be able to manage its traffic.

As per its application,every host (source) requests a sending rate it desires by depositing a value into a dedicated field Req_rate inside the packet header. This field can be updated by any router en route. Specifically, each router along the data path will compute an allowed source transmission rate according to the IQSize and then compare it with the rate already recorded in Req_rate field. If the former is smaller than the latter, the Req_rate field in the packet header will be updated; otherwise it remains unchanged. After the packet arrives at the destination, the value of the Req_rate field reflects the allowed data rate from the most congested router along the path if the value is not more than the desired rate of the source. The receiver then sends this value back to the source via an ACK (Acknowledgment) packet, and the source would update its current sending rate accordingly. If no router modifies Req_rate field, it means that all routers en route allow the source to send its data with the requested desired rate.

In order to implement our new controller in each router, we model a typical AQM router in Figure 1 with M sources sending their Internet traffic to their respective destinations. For i = 1, 2, . . . , M, u_i(t) is the current sending rate of source i; ui(t) is the sending rate of source i determined by the routers along the end-to-end path; y(t) is the incoming aggregate controlled flow rate; v(t) is the incoming aggregate uncontrolled flow rate, and c(t) is the link bandwidth (measured in bps).

For a particular source-destination pair i, τfi1 is the time delay of a packet from source i to the router, and τfi2 is the time delay of the packet of source i from the router to the destination i, while τbi is the feedback delay from destination i back to source i. Obviously, τpi = τfi1 + τfi2 + τbi is the RTPD (Round Trip Propagation Delay). Considering other delays en route (e.g., queueing delay), source i may update its current rate u_(t) according to the ui(t) when the ACK packet arrives after one RTT (Round Trip Time) τi.

Considering the possible dynamics of both incoming traffic and link bandwidth in the router in Figure 1, we model the bottleneck link with a queue in which both the controlled arrival rate y(t) and the service rate c(t) may vary with respect to time. Let q(t) be the router IQSize. The variations in y(t) and/or c(t) can cause changes in the queue size of a router, as expressed in the following differential equation.

q(t) = y (t) + v (t) − c (t) q (t) > 0 [y (t) + v (t) − c (t)]+ q (t) = 0
where [x]+ = max (0, x).

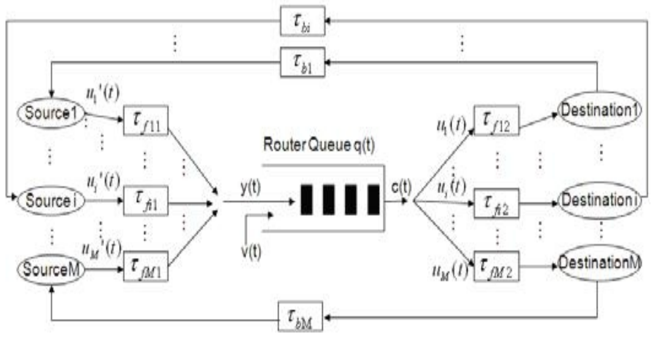


Figure 1: System Model of an AQM router

3. The IntelRate Controller Design:

Figure 2 depicts the components of our fuzzy logic traffic controller for controlling traffic in the network system defined in Fig. 1. Called the IntelRate, it is a TISO (Two-Input Single-Output) controller. The TBO (Target Buffer Occupancy) q0 >0 is the queue size level we aim to achieve upon congestion. The queue deviation e(t) = q0 −q(t) is one of the two inputs of the controller. In order to remove the steady state error, we choose the integration of e(t) as the other input of the controller, i.e. g (e (t)) = e (t) dt. The aggregate output is y (t) = ui (t − τi). Under heavy traffic situations, the IntelRate controller would compute an allowed sending rate ui(t) for flow i according to the current IQSize so that q(t) can be stabilized around q0. In our design, IQSize q(t) is the only parameter each router needs to measure in order to complete the closed-loop control.FLC is a non-linear mapping of inputs into outputs, which consists of four steps, i.e., rule base building, fuzzification, inference and defuzzification. The concepts of fuzzy set and logic of FLC were introduced in 1965 by Zadeh, and it was basically extended from two-valued logic to the continuous interval by adding the intermediate values between absolute TRUE and FALSE.

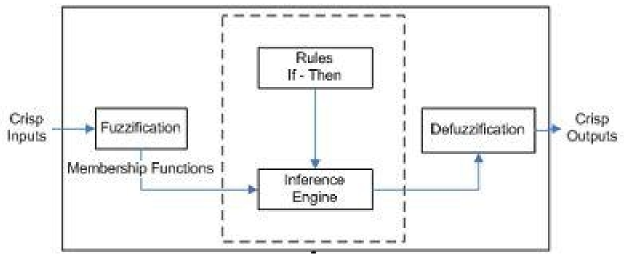


Figure 2: The IntelRate closed-loop control system

Volume No:1, Issue No:7 (December-2015)
ISSN No : 2454-423X (Online)
International Journal of Research in Advanced Computer Science Engineering
A Peer Reviewed Open Access International Journal
www.ijracse.com

4. Performance Evaluation:

The capability of the IntelRate controller is demonstrated by performance evaluations through a series of experiments. We will first describe the simulated network and performance measures of a case study in Section A. Section B demonstrate the system robustness upon large network changes.Queuing jitter control and the effect of short-lived traffic will be discussed in Sections C and D. Section E evaluates the bandwidth utilization and packet loss rate of the IntelRate controller. Finally, Section F discusses the choices of some design parameters.For the following evaluation, we choose N = 9, m = 8 and the delay of TBO≤10ms, while B = 10q0 and d = 50ms using the design principles for the IntelRate controller discussed inSection III. As noted there, some of these “optimum/good” values are the results after many experiments on different combinations. Due to space limitation, they are not presented.

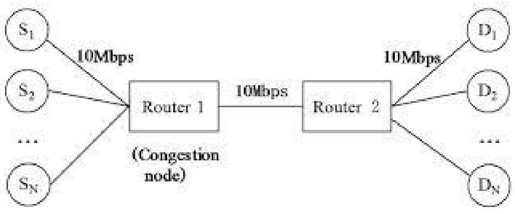


Figure 3: Simulated Network

The single bottleneck network in Fig. 3 is used to investigate the controller behavior of the most congested router. We choose Router 1 as the only bottleneck in the network, whereas Router 2 is configured to have sufficiently high service rate and big buffer B so that congestion never happens there. The numbers in Fig. 3 are the IDs of the subnets/groups attached to each router. Their configuration is summarized in Table II, in which there are M = 11 subnet pairs, which form the source destination data flows in the network, and they run various Internet applications such as the long-lived ftp, short-lived http, or the unresponsive UDP-like flows (also called uncontrolled ftp flows). Since the link bandwidths we want to simulate have a magnitude of Giga bits per second, we need to use 20 flows in each subnet to generate enough traffic to produce congestion. All flows within each group have the same RTPD and behavior, but different from the flows of other groups. The RTPD includes the forward path propagation delay and the feedback propagation delay, but does not include the queuing delay, which may vary according to our settings of TBO size in the experiments.

The reverse traffic is generated by the destinations when they piggyback the ACK information back to the sources. The TBO and the buffer capacity B in Router 1 in each experiment are set according to the approaches discussed in Section III. We also adopt some typical values from the experiments of existing works so that we can make our experiments more meaningful. In particular, all the ftp packets have the same size of 1024 bytes [19] while the http packet size is uniformly distributed in the range of [800, 1300] bytes.

Table 1: Sources Characters:

Subnet ID	Source ID	Flow NO.	RTPD(ms)
ftp group 1	1-20	ftp 1-20	80
ftp group 2	21-40	ftp 21-40	120
ftp group 3	41-60	ftp 41-60	160
ftp group 4	61-80	ftp 61-80	200
ftp group 5	81-100	ftp 81-100	240
http group 1	101-120	http 1-20	80
http group 2	120-140	http 21-40	120
http group 3	141-160	http 41-60	160
http group 4	161-180	http 61-80	200
http group 5	181-200	http 81-100	240
uncontrolled ftp	201	UDP 1	160

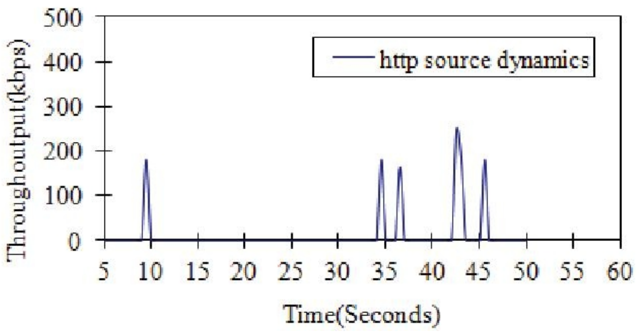
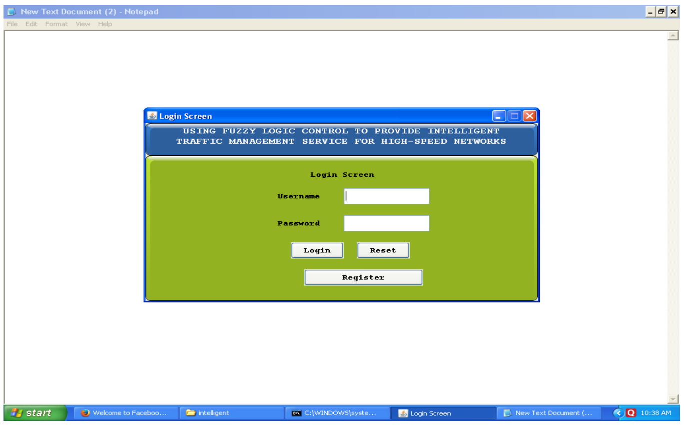


Figure 4: HTTP Sessions’s example

5. Screen Shots:



USING FUZZY LOGIC CONTROL TO PROVIDE INTELLIGENT
TRAFFIC MANAGEMENT SERVICE FOR HIGH-SPEED NETWORKS

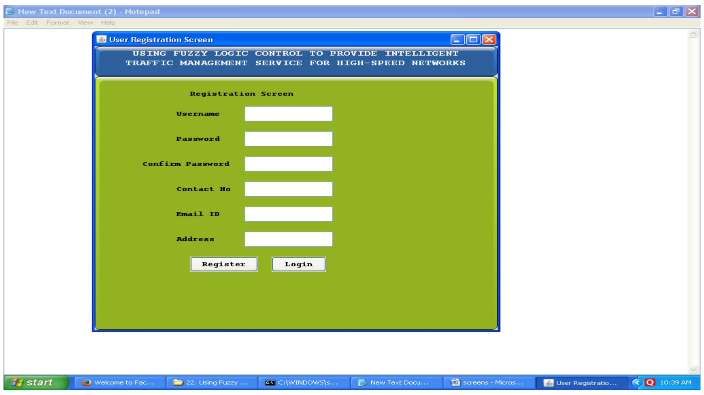
Login Screen

Username

Password

Click on register button to register as a new user:

Registration screen:


User Registration Screen

USING FUZZY LOGIC CONTROL TO PROVIDE INTELLIGENT TRAFFIC MANAGEMENT SERVICE FOR HIGH-SPEED NETWORKS

Registration Screen

Username

Password

Confirm Password

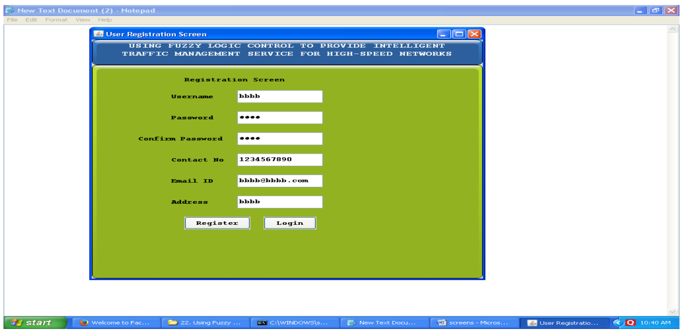
Contact No

Email ID

Address

Register Login

Register as a new user:


User Registration Screen

USING FUZZY LOGIC CONTROL TO PROVIDE INTELLIGENT TRAFFIC MANAGEMENT SERVICE FOR HIGH-SPEED NETWORKS

Registration Screen

Username bbbb

Password ••••

Confirm Password ••••

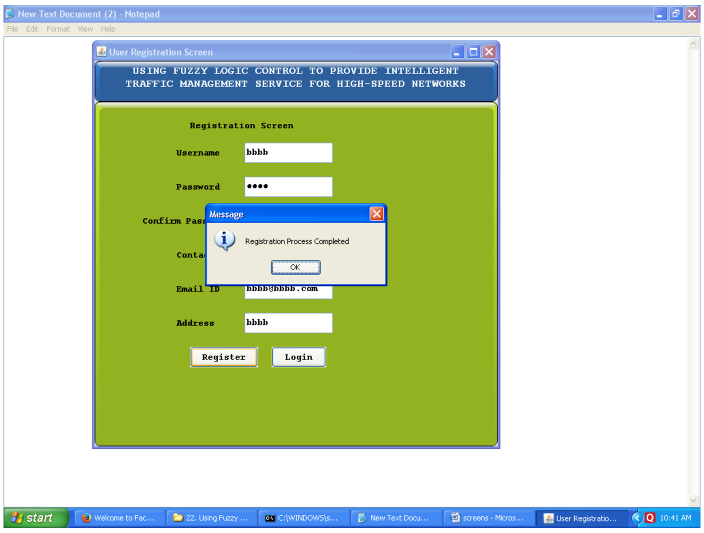
Contact No 1234567890

Email ID bbbb@bbbb.com

Address bbbb

Register Login

After successful registration:


User Registration Screen

USING FUZZY LOGIC CONTROL TO PROVIDE INTELLIGENT TRAFFIC MANAGEMENT SERVICE FOR HIGH-SPEED NETWORKS

Registration Screen

Username bbbb

Password ••••

Message

Registration Process Completed

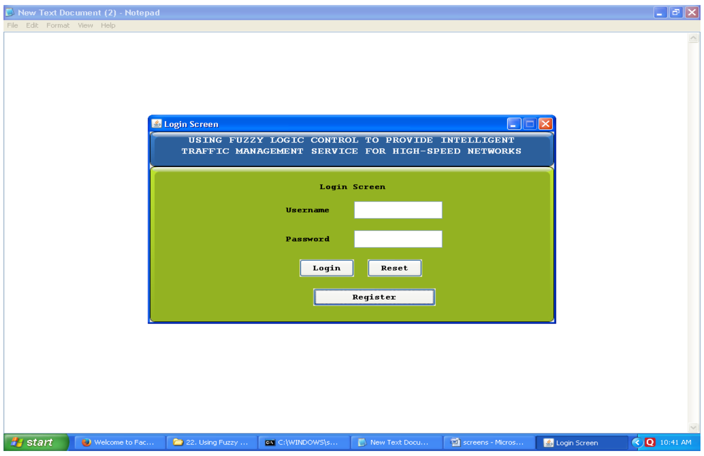
OK

Email ID bbbb@bbbb.com

Address bbbb

Register Login

Click on login to login as a user:


Login Screen

USING FUZZY LOGIC CONTROL TO PROVIDE INTELLIGENT TRAFFIC MANAGEMENT SERVICE FOR HIGH-SPEED NETWORKS

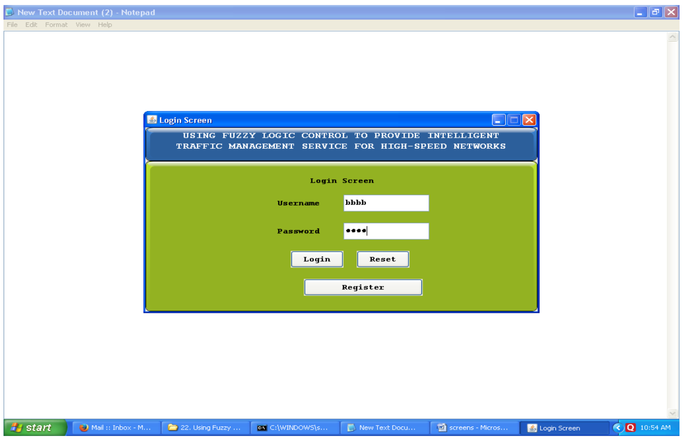
Login Screen

Username

Password

Login Reset

Register


Login Screen

USING FUZZY LOGIC CONTROL TO PROVIDE INTELLIGENT TRAFFIC MANAGEMENT SERVICE FOR HIGH-SPEED NETWORKS

Login Screen

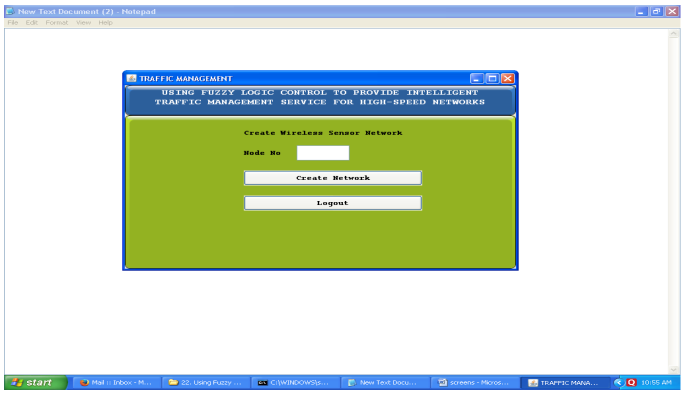
Username bbbb

Password ••••

Login Reset

Register

Home screen after successful login:


TRAFFIC MANAGEMENT

USING FUZZY LOGIC CONTROL TO PROVIDE INTELLIGENT TRAFFIC MANAGEMENT SERVICE FOR HIGH-SPEED NETWORKS

Create Wireless Sensor Network

Node No

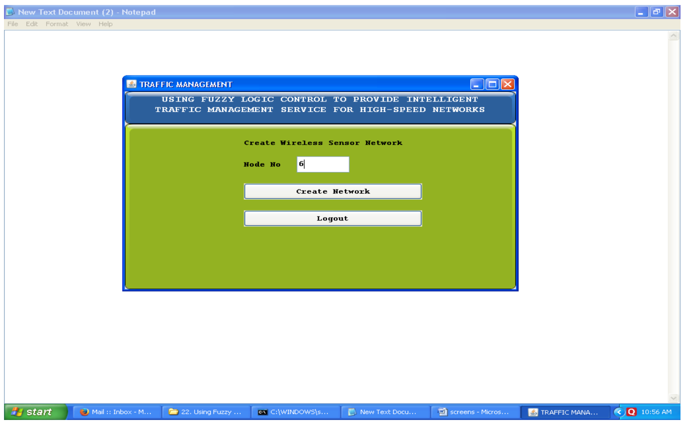
Create Network

Logout

Enter the number of nodes that has to be created in the network then click on create network button:

Volume No:1, Issue No:7 (December-2015) ISSN No : 2454-423X (Online)

International Journal of Research in Advanced Computer Science Engineering
A Peer Reviewed Open Access International Journal
www.ijracse.com

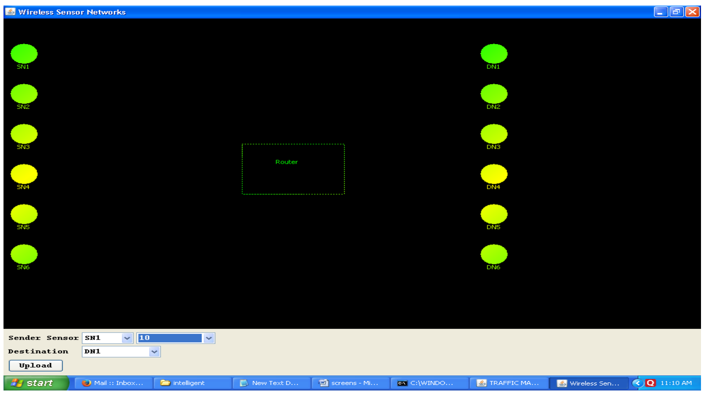

TRAFFIC MANAGEMENT

USING FUZZY LOGIC CONTROL TO PROVIDE INTELLIGENT TRAFFIC MANAGEMENT SERVICE FOR HIGH-SPEED NETWORKS

Create Wireless Sensor Network

Node No 6

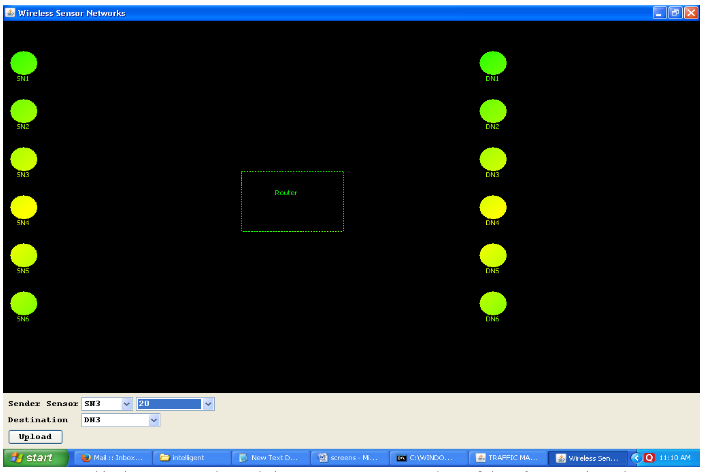
Network with 6 source nodes & 6 destination nodes:



Sender Sensor SN1 10

Destination DN1

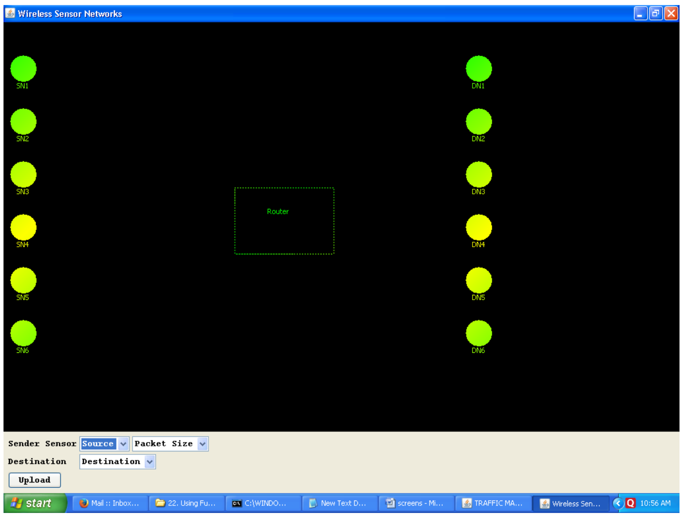
Select a sender node, destination node and the packet size (data rate), repeat the same logic for multiple source and destinations then upload a file.
(from source node 1 to destination node 1 the file has to be transferred with the data rate 10kb, from source node 3 to destination node 3 the file has to be transferred with the data rate 20kb)



Sender Sensor SN3 20

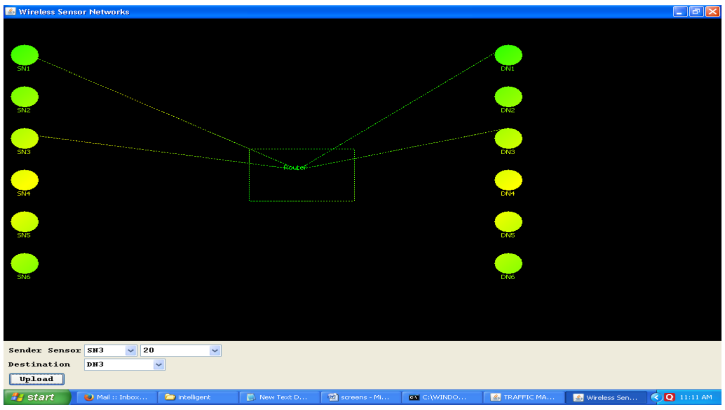
Destination DN3

Now click on upload button to send a file from both sn1 to dn1 and also sn3 to dn3 with the data rate of 10 and 20 respectively.



Sender Sensor Source Packet Size

Destination Destination



Sender Sensor SN3 20

Destination DN3

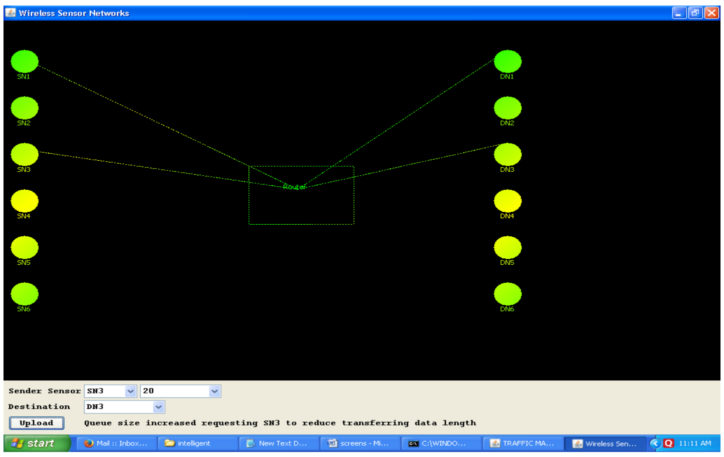
Select a sender node, destination node and the packet size (data rate), repeat the same logic for multiple source and destinations then upload a file.
(from source node 1 to destination node 1 the file has to be transferred with the data rate 10kb, from source node 3 to destination node 3 the file has to be transferred with the data rate 20kb)

Volume No:1, Issue No:7 (December-2015) ISSN No : 2454-423X (Online)

International Journal of Research in Advanced Computer Science Engineering

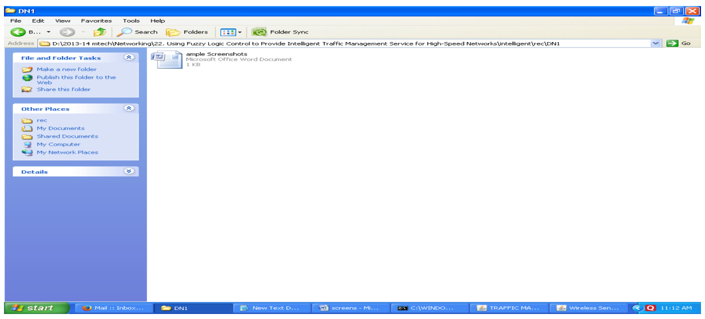
A Peer Reviewed Open Access International Journal

www.ijracse.com



Sender Sensor SN3 20
Destination DN3
Upload Queue size increased requesting SN3 to reduce transferring data length

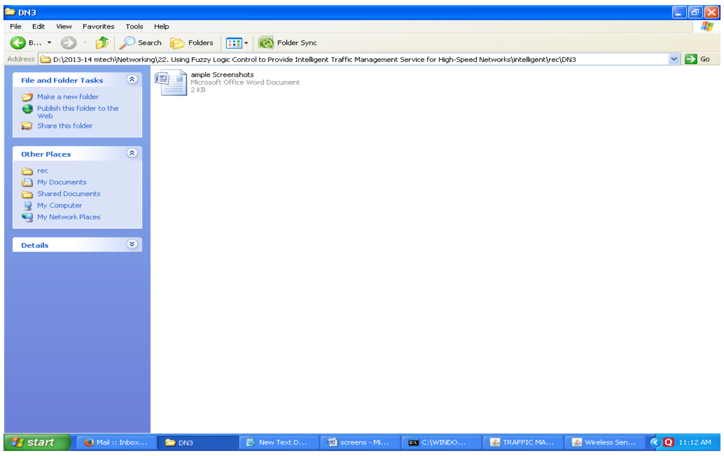
We can check the received data at all the specified destination nodes in the rec folder.
Received data at dn1. (it has received only 1kb in certain amount of time with data rate of 10)



D:\2013-14 mtech\Networking\22. Using Fuzzy Logic Control to Provide Intelligent Traffic Management Service for High-Speed Networks\intelligent\rec\DN1

ample Screenshots
Microsoft Office Word Document
1 KB

Received data at dn3. (it has received 2kb in certain amount of time with data rate of 20)



D:\2013-14 mtech\Networking\22. Using Fuzzy Logic Control to Provide Intelligent Traffic Management Service for High-Speed Networks\intelligent\rec\DN3

ample Screenshots
Microsoft Office Word Document
2 KB

5. Conclusion:

A novel traffic management scheme, called the IntelRate controller, has been proposed to manage the Internet congestion in order to assure the quality of service for different service applications. The controller is designed by paying attention to the disadvantages as well as the advantages of the existing congestion control protocols. As a distributed operation in networks, the IntelRate controller uses the instantaneous queue size alone to effectively throttle the source sending rate with max-min fairness. Unlike the existing explicit traffic control protocols that potentially suffer from performance problems or high router resource consumption due to the estimation of the network parameters, the IntelRate controller can overcome those fundamental deficiencies. To verify the effectiveness and superiority of the IntelRate controller, extensive experiments have been conducted in OPNET modeler. In addition to the feature of the FLC being able to intelligently tackle the nonlinearity of the traffic control systems, the success of the IntelRate controller is also attributed to the careful design of the fuzzy logic elements.

References:

[1]M. Welzl, Network Congestion Control: Managing Internet Traffic. John Wiley & Sons Ltd., 2005.

[2]R. Jain, “Congestion control and traffic management in ATM networks: recent advances and a survey,” Computer Networks ISDN Syst., vol. 28, no. 13, pp. 1723–1738, Oct. 1996.

[3]V. Jacobson, “Congestion avoidance and control,” in Proc. 1988 SIGCOMM, pp. 314–329.

[4]V. Jacobson, “Modified TCP congestion avoidance algorithm,” Apr. 1990.

[5]K. K. Ramakrishnan and S. Floyd, “Proposals to add explicit congestion notification (ECN) to IP,” RFC 2481, Jan. 1999.

[6]D. Katabi, M. Handley, and C. Rohrs, “Congestion control for high bandwidth-delay product networks,” in Proc. 2002 SIGCOMM, pp. 89–102.

Volume No:1, Issue No:7 (December-2015) ISSN No : 2454-423X (Online)

International Journal of Research in Advanced Computer Science Engineering
A Peer Reviewed Open Access International Journal
www.ijracse.com

[7]S. H. Low, F. Paganini, J.Wang, et al., “Dynamics of TCP/AQM and a scalable control,” in Proc. 2002 IEEE INFOCOM, vol. 1, pp. 239–248.

[8]S. Floyd, “High-speed TCP for large congestion windows,” RFC 3649, Dec. 2003.

[9]W. Feng and S. Vanichpun, “Enabling compatibility between TCP Reno and TCP Vegas,” in Proc. 2003 Symp. Applications Internet, pp. 301– 308.

[10]M. M. Hassani and R. Berangi, “An analytical model for evaluating utilization of TCP Reno,” in Proc. 2007 Int. Conf. Computer Syst.Technologies, p. 14-1-7.

[11] N. Dukkipati, N. McKeown, and A. G. Fraser, “RCP-AC congestion control to make flows complete quickly in any environment,” in Proc.2006 IEEE INFOCOM, pp. 1–5.

[12] Y. Zhang, D. Leonard, and D. Loguinov, “JetMax: scalable max-min congestion control for high-speed heterogeneous networks,” in Proc. 2006 IEEE INFOCOM, pp. 1–13.

[13] B. Wydrowski, L. Andrew, and M. Zukerman, “Max-Net: a congestion control architecture for scalable networks,” IEEE Commun. Lett., vol. 7, no. 10, pp. 511–513, Oct. 2003.

[14] Y. Zhang and M. Ahmed, “A control theoretic analysis of XCP,” in Proc. 2005 IEEE INFOCOM, vol. 4, pp. 2831–2835.

[15]Y. Zhang and T. R. Henderson, “An implementation and experimental study of the explicit control protocol (XCP),” inProc. 2005 IEEE INFOCOM, vol. 2, pp. 1037–1048.

[16] J. Pu and M. Hamdi, “Enhancements on router-assisted congestion control for wireless networks,” IEEE Trans. Wireless Commun., vol. 7, no. 6, pp. 2253–2260, June 2008.

[17] F. Abrantes, J. Araujo, and M. Ricardo, “Explicit congestion control algorithms for time varying capapcity media,” IEEE Trans. Mobile Comput., vol. 10, no. 1, pp. 81– 93, Jan. 2011.