Research Paper

Volume : 3 | Issue : 5 | May 2014 • ISSN No 2277 - 8179

Fuzzy Logic Based Traffic Management For High Speed Networks

Engineering

KEYWORDS : Congestion control, Fuzzy logic, Network traffic management

Anil C PG Student, Dept. of Computer Science, NSS College of Engineering, Palakad.

Sruthy Manmadhan Asst. Professor, Dept. of Computer Science, NSS College of Engineering, Palakad.

ABSTRACT

Network traffic management is a core area of research that is of great importance in the field of communication. Although many models have been proposed since time being, all those techniques have their own shortcomings. This paper proposes a new scheme for traffic management by exploiting the possibilities of fuzzy logic. The proposal applies fuzzy logic for predicting the maximum allowable sending rate by observing the queue size of router. Along with the queue size of router, the processing power of router is also taken as a criterion in determining the sending rate. This is justified by the fact that different routers in a network are capable of processing packets at different rates. Fuzzy logic is being employed because; the instantaneous queue size may not be obtained as a crisp measurement always. Also, employing fuzzy logic enables us to make the network more adaptive to current traffic conditions. A new fuzzy controller is to be modeled to implement the proposed system. The paper also suggests an appropriate fuzzy controller for testing the efficiency of the proposed model.

1. INTRODUCTION

Network traffic management can prevent a network from severe congestion and degradation in throughput-delay performance. Congestion in a network may occur when the load on the network is greater than the capacity of the network. Congestion control is 'adapting speed of transmission to match available end-to-end network capacity'. In order to make use of the bandwidth those high-speed networks offers, we need to have a proper and efficient mechanism to control congestion in these nodes.

Many classic ways to handle congestion control have been proposed, and is being made use today for network traffic management. There are mainly two classes of approaches: implicit congestion control and explicit congestion control. Here, this study suggests application of fuzzy logic in determining the optimum source sending rate. The ability of fuzzy logic to handle a loosely-defined input and to produce crisp output from those inputs makes it suitable for application in our proposed system. The queue size variation is used as the input to the fuzzy logic controller. Later, after applying fuzzy solving steps, we produce the desired receiving rate of the router as a crisp output.

2. PROPOSED SYSTEM

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 Congestion occurs Congestion occurs when many flows traverse a router and causes its 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.

2.1 OPERATION PRINCIPLE

Inside each router, the distributed traffic controller acts as a data rate regulator by measuring and monitoring the IQSize. 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 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 the new controller in each router, a typical router is modeled, as in Fig.1, with M sources sending their traffic to their respective destinations. $U_i^l(t)$ for i=1,2,...,M is the current sending rate is source i. $U_i(t)$ is the sending rate of source i determined by the routers along the end-to-end path. For a particular source-destination pair i, T_{fi} is the time delay of a packet from source i to the router, and T_{f2} is the time delay of the packet of source i from the router to the destination i, while $T_{bi}$is the feedback delay from destination i back to source i. Obviously, $T_{pi}= T_{fi1} + T_{fi2} + T_{bi}$is the RTPD (Round Trip Propagation Delay). Considering other delays en route (e.g., queueing delay), source i may update its current rate $U^l(t)$ according to the $U_i(t)$ when the ACK packet arrives after one RTT (Round Trip Time) $T_{i.}$

2.2 THE CONTROLLER DESIGN

Fig.2.2 depicts the components of the fuzzy logic traffic controller for controlling traffic in the network system. 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. Other input is the processor speed. The aggregate output is $y(t) = ui(t - \tau i)$. Under heavy traffic situations, the controller would compute an allowed sending rate $ui(t)$ for flow $i$according to the current inputs so that $q(t)$ can be stabilized around $q0$

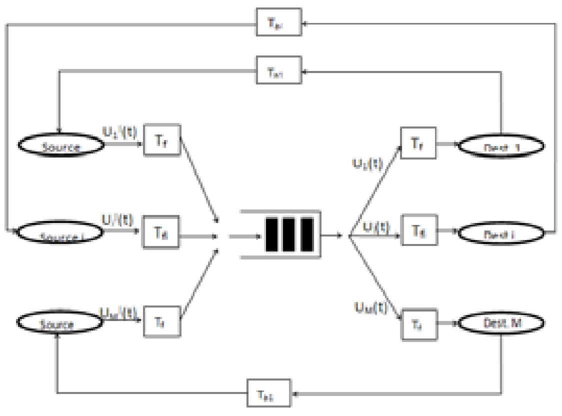


Fig 2.1. Router Model

A. Problem Definition &Linguistic Variable Description Here, we define the problem and will give the description of Linguistic Variables (LV) of the fuzzy system.

The Problem Definition

Produce a crisp value for the rate of packet flow (r (t)), taking router queue deviation (e (t)) and processing capacity of the router (p (t)) as the two inputs.

Linguistic Variables

1. LV for Queue Deviation (e (t)):

a. Very Small (VS)
b. Small (SS)
c. Medium (MM)
d. Large (LL)
e. Very Large (VL)

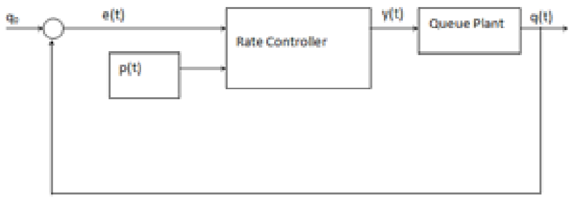


Fig.2.2. Fuzzy Controller

2. LV for Processing Speed (p (t)):

a. Low (LL)
b. Average (AA)
c. High (HH)

3. LV for Rate of Packet Flow (r (t)):

a. Minimum (MI)
b. Optimal (OO)
c. Maximum (MX)

C. Fuzzy Set Description

Here, we define the fuzzy sets used to construct fuzzy values. Here, for making the design simple, we make following assumptions:

- Maximum queue size is 3MB (3072 KB)
- Processing speed of routers vary between 150MHz to 800MHz
- Maximum Rate of packet flow is 1Mbps

1. Fuzzy set for Queue Deviation

a. VS: for 0<= e(t) <= 500
b. SS: for 500<= e(t) <= 1024
c. MM: for 1024<= e(t) <= 2048
d. LL: for 2048 <= e(t) <= 2560
e. VL: for 2560 <= e(t) <= 3072

2. Fuzzy set for Processing Speed

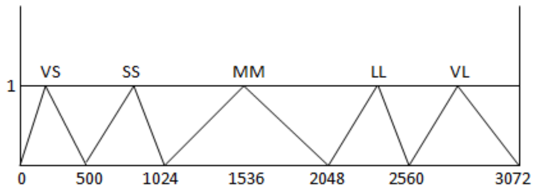
a. LL: for 150 <= p(t) <=450
b. AA: for 450 <= p(t) <=650
c. HH: for 600 <= p(t) <= 800

3. Fuzzy set for Rate of packet flow

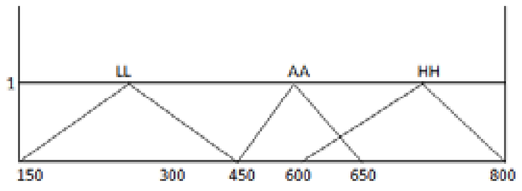
a. MI: for 0<= r(t) <= 400
b. OO: for 400 <= r(t) <= 900
c. MX: for 800<= r(t) <= 1024

D. Fuzzy Member Functions

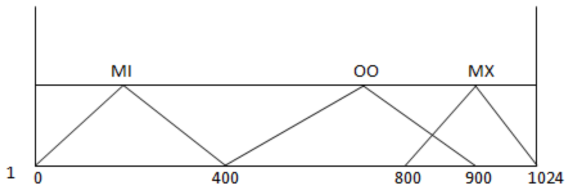
1. For Queue Deviation



2. For Processing Speed



3. For Rate of Packet Flow



D Fuzzy Rules

Following rules are applied on fuzzy input sets:

a. If e(t) is 'VS' OR p(t) is 'LL', then r(t) is MI
b. If e(t) is 'VS' AND p(t) is 'AA', then r(t) is MI
c. If e(t) is 'VS' AND p(t) is 'LL', then r(t) is MI
d. If e(t) is 'SS' AND p(t) is 'HH', then r(t) is MI
e. If e(t) is 'SS' AND p(t) is 'AA', then r(t) is OO
f. If e(t) is 'SS' AND p(t) is 'LL', then r(t) is OO
g. If e(t) is 'MM' AND p(t) is 'HH', then r(t) is OO
h. If e(t) is 'MM' AND p(t) is 'AA', then r(t) is OO
i. If e(t) is 'MM' AND p(t) is 'LL', then r(t) is OO
j. If e(t) is 'LL' AND p(t) is 'HH', then r(t) is OO
k. If e(t) is 'LL' AND p(t) is 'AA', then r(t) is OO
l. If e(t) is 'LL' AND p(t) is 'LL', then r(t) is MX
m. If e(t) is 'VL' AND p(t) is 'HH', then r(t) is MX
n. If e(t) is 'VL' AND p(t) is 'AA', then r(t) is MX
o. If e(t) is 'VL' AND p(t) is 'LL', then r(t) is OO
p. If e(t) is 'VL' OR p(t) is 'HH', then r(t) is MX

2.3 EVALUATION DETAILS

The proposed traffic management scheme targets high-speed networks and network protocols such as IP. The network should consist of interconnected routers that run the proposed algorithm. The efficiency of proposed system can be tested by employing a simulated network consisting of a sender, receiver and a set of interconnected routers.

A. TOOLS USED:

NS2:

The network to be used for evaluation can be simulated using NS2(Network Simulator 2). NS2 is an open-source event-driven simulator designed specifically for research in computer communication networks. NS2 contains modules for numerous network components such as routing, transport layer protocol, application, etc. To investigate network performance, researchers can simply use an easy-to-use scripting language to configure a network, and observe results generated by NS2.

Simulation of wired as well as wireless network functions and protocols (e.g., routing algorithms, TCP, UDP) can be done using NS2. In general, NS2 provides users with a way of specifying such network protocols and simulating their corresponding behaviors. NS2 provides users with an executable command ns which takes on input argument, the name of a Tcl simulation scripting file. Users are feeding the name of a Tcl simulation script (which sets up a simulation) as an input argument of an NS2 executable command ns. In most cases, a simulation trace file is created, and is used to plot graph and/or to create animation.

NS2 consists of two key languages: C++ and Object-oriented Tool Command Language (OTcl). While the C++ defines the internal mechanism (i.e.,a backend) of the simulation objects, the OTcl sets up simulation by assembling and configuring the objects as well as scheduling discrete events (i.e., a frontend). The C++ and the OTcl are linked together using TclCL. Mapped to a C++ object, variables in the OTcl domains are sometimes referred to as handles. Conceptually, a handle (e.g., n as a Node handle) is just a string (e.g., o10) in the OTcl domain, and does not contain any functionality. Instead, the functionality (e.g., receiving a packet) is defined in the mapped C++ object (e.g., of class Connector). In the OTcl domain, a handle acts as a frontend which interacts with users and other OTcl objects. It may defines its own procedures and variables to facilitate the interaction. Note that the member procedures and variables in the OTcl domain are

Volume : 3 | Issue : 5 | May 2014 • ISSN No 2277 - 8179

called instance procedures (instprocs) and instance variables (instvars), respectively. Before proceeding further, the readers are encouraged to learn C++ and OTcl languages. NS2 provides a large number of built-in C++ objects. After simulation, NS2 outputs either text based or animation-based simulation results. To interpret these results graphically and interactively, tools such as NAM (Network AniMator) and XGraph are used.

B. SIMULATION NETWORK DESIGN

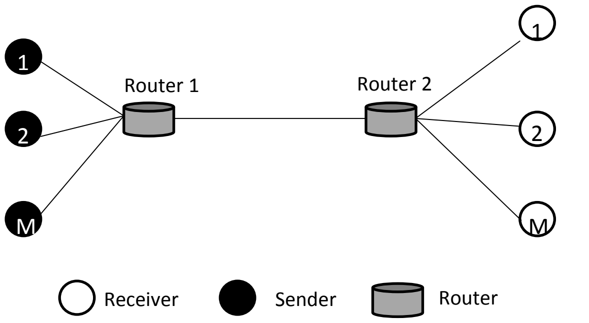


Fig.2.3. Simulator Network Design

The network consists of M senders transmitting data to M destinations, via two routers. Each router will be configured to run the proposed fuzzy-based algorithm to manage the network traffic accordingly. The result of running the proposed algorithm can be collected using data collection tools in NS2, and can be compared with normal traffic management technique.

3. CONCLUSION

The study proposes a fuzzy-based technique for efficient traffic management in IP based networks. The scheme uses router queue size and processing capacity of router as the two inputs in deciding optimum packet transfer rate through network so that the network resources are efficiently utilized, and congestion is avoided. Since the two inputs taken here can be directly measured from the router node itself, the scheme doesn't impose any additional overhead in measuring network parameters like link utilization, bottleneck link, etc. This makes the proposed scheme attractive and places it above the techniques that make use of network parameters to decide data flow rate. The performance evaluation using simulated network shows that the proposed system is effective in handling network traffic intelligently, thus producing maximum throughput.

REFERENCE

[1]Jungang Liu, Oliver W. W. Yang,"Using Fuzzy Logic Control to Provide Intelligent Traffic Management Service for High-Speed Networks", ieee transactions on network and service management, vol. 10, no. 2, june 2013 | | [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. | | |