CMR UNIVERSITY
Private University Established in Karnataka State by Act No. 45 of 2013

School of Engineering Technology

Main Campus, Off Hennur-Bagalur Main Road, Chagalahatti, Bengaluru-562149

A
MINI PROJECT REPORT

"TRAFFIC MANAGEMENT IN NETWORKS USING FUZZY LOGIC"

Submitted to
CMR University School Of Engineering Technology, Bagalur
for the partial fulfillment of the Requirement for the Award of the Degree of

B.TECH
IN
COMPUTER SCIENCE

Submitted by:
SARANYA .T (16UG08047)
IDIGA SREEJA (16UG08013)
PRIYANKA K R (16UG08036)

Under the Guidance Of

Prof. BHULAKSHMI DASARI

DEPARTMENT OF
COMPUTER SCIENCE AND ENGINEERING CMR UNIVERSITY BAGALURE
2018-19

CMR UNIVERSITY
Private University Established in Karnataka State by Act No. 45 of 2013

School of Engineering Technology

Main Campus, Off Hennur-Bagalur Main Road, Chagalahatti, Bengaluru-562149

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

Certified that the project work entitled ***TRAFFIC MANAGEMENT IN NETWORKS USING FUZZY LOGIC*** *carried out by Ms.* ***SARANYA.T REG.NO16UG08047, Ms.IDIGA SREEJA REG.NO 16UG08013,Ms. PRIYANKA.K.R REG.NO 16UG08036*** *in partial fulfillment for the award of Bachelor of Engineering / Bachelor of Technology in* ***COMPUTER SCIENCE*** *of the CMR University, Bagalur during the year 2018-19. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said Degree.*

Prof.BHULAKSHMI	***Prof.SMITHA RAO***	***DR.NAGRAJ M K***
Signature of the Guide	*Signature of the HOD*	*Signature of the Dean*

External Viva

Name of the examiners	**Signature with date**
1	
2	

DECLARATION

WE, SARANYA.T ,IDIGA SREEJA,PRIYANKA KR *student of CMR university school of engineering and technology, bagalur hearby declare that the dissertation entitled* ***" TRAFFIC MANAGEMENT IN NETWORKS USING FUZZY LOGIC"*** *embodies the report of my project carried out independently by me during sixth semester of* **B.TECH in COMPUTER SCIENCE,** *under the supervision and guidance of* ***Prof. BHULAKSHMI DASARI*** *Department of Computer Science and Engineering and this work has been submitted for the partial fulfillment of the requirements for the award of the B.Tech degree.*

I have not submitted the matter embodies to any other university or institution for the award of other degree.

Date :

Place :

SARANYA T

(REG.NO 16UG08047)

IDIGA SREEJA

(REG.NO 16UG08013)

PRIYANKA KR

(REG.NO 16UG08036)

ACKNOWLEDGEMENT

I express my foremost gratitude to ***Dr.NAGRAJ M K,*** *Dean for his constant support.*

I express my foremost gratitude to ***Prof.SMITHA RAO****, HOD, Dept of Computer Science & Engineering, CMR University, Bagalur for her inspiration, direction and discussions in successful completion of this dissertation.*

I express my foremost gratitude to my guide ***Prof.BHULAKSHMI DASARI****, Dept of Computer Science & Engineering, CMR University, Bagalur for his/her inspiration, adroit guidance, constant supervision, direction and discussions in successful completion of this dissertation.*

My sincere thanks to all teaching and non-Teaching Staff of ***Computer Science & Engineering Department*** *for all the facilities provided, without which, I could not have progressed with my work. Thanks to my parents who have been a great source of strength in the completion of this dissertation.*

SARANYA T

(USN: 16UG08047)

IDIGA SREEJA

(USN:16UG08013)

PRIYANKA KR

(USN:16UG08036)

ABSTRACT

In view of the fast-growing Internet traffic, 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 consumption of 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. The communication QoS (Quality of Service) is assured by the good performances of our scheme such as max-min fairness, low queuing delay and good robustness to network dynamics. Simulation results and comparisons have verified the effectiveness and showed that our new traffic management scheme can achieve better performances than the existing protocols that rely on the estimation of network parameters.

TABLE OF CONTENTS

CHAPTER 1
PREAMBLE

1.1 Introduction

Network traffic management can prevent a network from severe congestion and degradation in throughput delay performance. Traffic congestion control is one of the effective approaches to manage the network traffic. Historically, TCP (Transmission Control Protocol) Reno, 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. 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, RCP, JetMax and MaxNet. 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 network 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. 2) The sources can converge their sending rates to some social optimum and achieve a certain optimization objective. 15CSP05 3 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 mis-estimation 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. There are some explicit protocols that appear to compute the sending rates based solely on the queue size, but in fact they still need to estimate the number of active flows in a router, and this consumes CPU and memory resources. Examples are the rate-based controllers for packet switching networks and the ER (Explicit Rate) allocation algorithm for ATM (Asynchronous Transfer Mode) networks. For the API-RCP controller, both the original method (a truncated network model) and the improved method face a memory problem when dealing with many flows (that numbers in millions) arriving to a core router every hour. In some other controllers the TBO (Target Buffer Occupancy) is designed to be as high as 3 times of the BDP, which can cause large queuing delay and thus degrading network performance, and this becomes even worse in the high-speed networks. Historically, the ER allocation algorithms in ATM networks also share the same problems because they need to evaluate the link bandwidth and/or the numbers of active VCs (Virtual Circuits). Some others adjust the source sending rates in binary-feedback switches or explicit feedback switches according to a few queue thresholds, which may cause unfairness as well as high cell loss rate.

1.2 Literature Survey

1. **Network Congestion Control- Managing Internet Traffic**:

Author: John Wiley & Sons Ltd., 2005.

Abstract: This book is the result of an attempt to describe a seemingly complex domain in simple words. In the literature, all kinds of methods are applied to solve problems in congestion control, often depending on the background of authors – from fuzzy logic to game theory and from control theory to utility functions and linear programming, it seems that quite a diverse range of mathematical tools can be applied. In order to understand all of these papers, one needs to have a thorough understanding of the underlying theory.

2. **Congestion control and traffic management in ATM networks- recent advances and a survey**

Author: R. Jain

Abstract: Congestion control mechanisms for ATM networks as selected by the ATM Forum traffic management group are describe. Reasons behind these selections are explained. In particular selection criterion for selection between rate based and credit based approach and the key points of the debate between these two approaches are presented. The approach that was finally selected and several other schemes that were considered are described.

3. **Congestion avoidance and control**

Author: V. Jacobson

Abstract: In October of '86, the Internet had the first of what became a series of 'congestion collapses'. During this period, the data throughput from LBL to UC Berkeley (sites separated by 15CSP05 5 400 yards and three IMP hops) dropped from 32 Kbps to 40 bps. Mike Karels1 and I were fascinated by this sudden factor-of-thousand drop in bandwidth and embarked on an investigation of why things had gotten so bad. We wondered, in particular, if the 4.3BSD (Berkeley UNIX) TCP was mis-behaving or if it could be tuned to work better under abysmal network conditions. The answer to both of these questions was “yes”. Since that time, we have put seven new algorithms into the 4BSD TCP: round-trip-time variance estimation exponential retransmit timer backoff slow-start more aggressive receiver ack policy dynamic window sizing on congestion Karn's clamped retransmit backoff fast retransmit. Our measurements and the reports of beta testers suggest that the final product is fairly good at dealing with congested conditions on the Internet. This paper is a brief description of (i) - (v) and the rationale behind them. (vi) Is an algorithm recently developed by Phil Karn of Bell Communications Research, described in [KP87]. (viii) Is described in a soon-to-be-published RFC. Algorithms (i) - (v) spring from one observation: The flow on a TCP connection (or ISO TP-4 or Xerox NS SPP connection) should obey a 'conservation of packets' principle. And, if this principle were obeyed, congestion collapse would become the exception rather than the rule. Thus congestion control involves finding places that violate conservation and fixing them. By 'conservation of packets' I mean that for a connection 'in equilibrium', i.e., running stably with a full window of data in transit, the packet flow is what a physicist would call 'conservative': A new packet isn't put into the network until an old packet leaves. The physics of flow predicts that systems with this property should be robust in the face of congestion. Observation of the Internet suggests that it was not particularly robust.

4. **Modified TCP congestion avoidance algorithm**:

Author: V. Jacobson

Abstract: Originally TCP was designed for early, low bandwidth, short distance networks, so Standard TCP did not utilize the maximum bandwidth in today’s high bandwidth network environments. Therefore a lot of TCP congestion control mechanisms also known as TCP variants have been developed for today’s long distance high bandwidth networks. In this paper 15CSP05 6 the experimental results evaluating the performance of TCP Reno, High Speed TCP, BIC TCP, TCP CUBIC and Compound TCP in short and long distance high bandwidth networks are presented. Results show that TCP CUBIC shows the highest performance in good put whereas TCP Compound shows the highest performance in protocol fairness and TCP friendliness as compared to the other stat of the art congestion control mechanisms.

5. **Proposals to add explicit congestion notification (ECN) to IP:**

Author: K. K. Ramakrishnan and S. Floyd

Abstract: This note describes a proposed addition of ECN (Explicit Congestion Notification) to IP. TCP is currently the dominant transport protocol used in the Internet. We begin by describing TCP's use of packet drops as an indication of congestion. Next we argue that with the addition of active queue management (e.g., RED) to the Internet infrastructure, where routers detect congestion before the queue overflows, routers are no longer limited to packet drops as an indication of congestion. Routers could instead set a Congestion Experienced (CE) bit in the packet header of packets from ECN-capable transport protocols. We describe when the CE bit would be set in the routers, and describe what modifications would be needed to TCP to make it ECN-capable. Modifications to other transport protocols (e.g., unreliable unicast or multicast, reliable multicast, other reliable unicast transport protocols) could be considered as those protocols are developed and advance through the standards process.

1.3 Problem statement

Historically, TCP (Transmission Control Protocol) 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. 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. 15CSP05 7 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.

1.4 Objective

The contributions of our work lie in, 1) Using fuzzy logic theory to design an explicit rate-based traffic management scheme (called the IntelRate controller) for the high-speed IP networks; The application of such a fuzzy logic controller using less performance parameters while providing better performances than the existing explicit traffic control protocols; The design of a Fuzzy Smoother mechanism that can generate relatively smooth flow throughput; The capability of our algorithm to provide max-min fairness even under large network dynamics that usually render many existing controllers unstable.

The queue size can be accurately monitored. Used to proactively decide if action should be taken to regulate the source sending rate.QoS (Quality of Service) is assured by the good performances of our scheme such as max-min fairness, low queueing delay and good robustness to network dynamics.

1.5 Methodology

1. Sender

2. Receiver

3. Router Queuing Scheme

4. Network traffic analysis

MODULES DESCRIPTION

1. **Sender**: Server module is the main module for this project. Inside each router, our distributed traffic controller acts as a data rate regulator by measuring and monitoring the IQ Size. 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. In addition there is also Message Log, where all the alerts and messages are stored for the references. This Message Log can also be saved as Log file for future references for any network environment.

2. **Receiver**: 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.

3. **Router Queuing Scheme:** In this module, each router along the data path will compute an allowed source transmission rate according to the IQ Size and then compare it with the rate already recorded in Req_rate field. If the former is smaller System model of an AQM router, 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. 15CSP05 10

4. **Network traffic Analysis:** Using fuzzy logic theory to design an explicit rate-based traffic management scheme (called the IntelRate controller) for the high-speed IP networks; The application of such a fuzzy logic controller using less performance parameters while providing better performances than the existing explicit traffic control protocols; The design of a Fuzzy Smoother mechanism that can generate relatively smooth flow throughput.

CHAPTER 2

GENERAL ASPECTS AND TECHNOLOGY

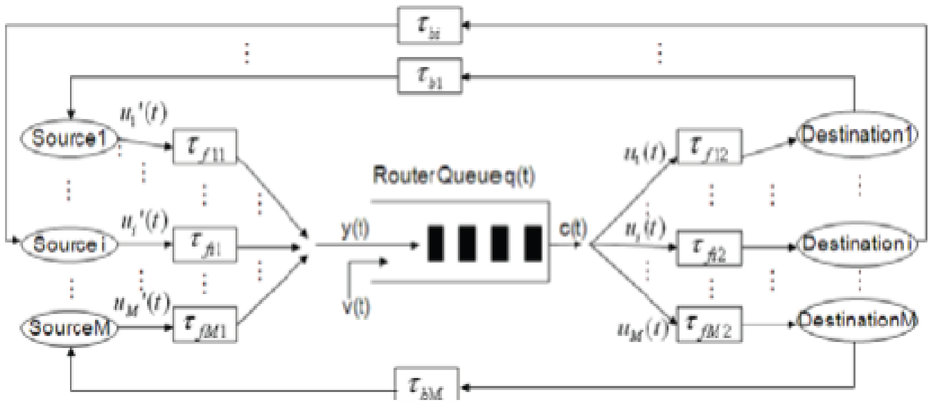
Recent studies show that mis-estimation of link bandwidth (e.g., in link sharing networks or wireless networks) may easily occur and can cause significant fairness and stability problems. There are some latest protocols on wireless applications such as QFCP (Quick Flow Control Protocol) and the three 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.

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, RCP, JetMax and MaxNet. 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 network 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**.**

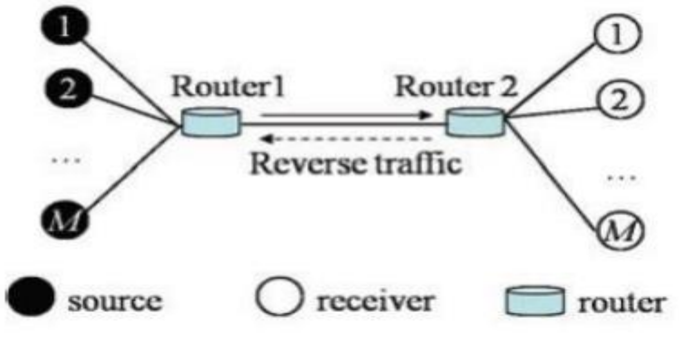
CHAPTER 3

IMPLEMENTATION

SYSTEM ARCHITECTURE:



SIMULATION SETUP:



CODE:

PRIVATE XML

```
<?xml version="1.0" encoding="UTF-8"?>

<project-private xmlns="http://www.netbeans.org/ns/project-private/1">

    <editor-bookmarks xmlns="http://www.netbeans.org/ns/editor-bookmarks/2" lastBookmarkId="0"/>

    <open-files xmlns="http://www.netbeans.org/ns/projectui-open-files/2">

        <group/>

    </open-files>

</project-private>
```

PROJECT PROPERTIES:

```
annotation.processing.enabled=true

annotation.processing.enabled.in.editor=false

annotation.processing.processors.list=

annotation.processing.run.all.processors=true
annotation.processing.source.output=${build.generated.sources.dir}/a
p-source-output

application.title=UssingFuzzyLogic

application.vendor=Mindsoft

build.classes.dir=${build.dir}/classes

build.classes.excludes=**/*.java,**/*.form

# This directory is removed when the project is cleaned:

build.dir=build

build.generated.dir=${build.dir}/generated

build.generated.sources.dir=${build.dir}/generated-sources

# Only compile against the classpath explicitly listed here:

build.sysclasspath=ignore

build.test.classes.dir=${build.dir}/test/classes

build.test.results.dir=${build.dir}/test/results

# Uncomment to specify the preferred debugger connection transport:

#debug.transport=dt_socket

debug.classpath=\

    ${run.classpath}

debug.test.classpath=\
    ${run.test.classpath}
# This directory is removed when the project is cleaned:
dist.dir=dist
dist.jar=${dist.dir}/UssingFuzzyLogic.jar
dist.javadoc.dir=${dist.dir}/javadoc
endorsed.classpath=
excludes=
includes=**
jar.archive.disabled=${jnlp.enabled}
jar.compress=false
jar.index=${jnlp.enabled}
javac.classpath=# Space-separated list of extra javac options
javac.compilerargs=false
javac.deprecation=false
javac.processorpath=\
    ${javac.classpath}
javac.source=1.6
javac.target=1.6
javac.test.classpath=\
    ${javac.classpath}:\
    ${build.classes.dir}
javac.test.processorpath=\
    ${javac.test.classpath}
javadoc.additionalparam=
javadoc.author=false
javadoc.encoding=${source.encoding}
javadoc.noindex=false
javadoc.nonavbar=false
javadoc.notree=false
javadoc.private=false
javadoc.splitindex=true
javadoc.use=true
javadoc.version=false
javadoc.windowtitle=
jnlp.codebase.type=no.codebase
jnlp.descriptor=application
jnlp.enabled=false

jnlp.mixed.code=default

jnlp.offline-allowed=false

jnlp.signed=false

jnlp.signing=

jnlp.signing.alias=

jnlp.signing.keystore=

main.class=

manifest.file=manifest.mf

meta.inf.dir=${src.dir}/META-INF

mkdist.disabled=false

platform.active=default_platform

run.classpath=\

    ${javac.classpath}:\

    ${build.classes.dir}

# Space-separated list of JVM arguments used when running the
project.

# You may also define separate properties like run-sys-
prop.name=value instead of -Dname=value.
```

```
# To set system properties for unit tests define test-sys-prop.name=value:

run.jvmargs=

run.test.classpath=\

    ${javac.test.classpath}:\

    ${build.test.classes.dir}

source.encoding=UTF-8

src.dir=src

test.src.dir=test
```

GENEFILES PROPERTIES:

```
build.xml.data.CRC32=38d1a0a9

build.xml.script.CRC32=265b43bf

build.xml.stylesheet.CRC32=8064a381@1.75.1.48

# This file is used by a NetBeans-based IDE to track changes in generated files such as build-impl.xml.

# Do not edit this file. You may delete it but then the IDE will never regenerate such files for you.

nbproject/build-impl.xml.data.CRC32=38d1a0a9

nbproject/build-impl.xml.script.CRC32=d8aa5f25

nbproject/build-impl.xml.stylesheet.CRC32=876e7a8f@1.75.1.
```

CHAPTER 4

RESULTS AND DISCUSSIONS

The queue size can be accurately monitored. Used to proactively decide if action should be taken to regulate the source sending rate. QoS (Quality of Service) is assured by the good performances of our scheme such as max-min fairness, low queueing delay and good robustness to network dynamics.

To verify the effectiveness and superiority of the IntelRate controller, extensive experiments have been conducted in OPNET modeler.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 surce sending rate with max-min fairness.

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

Jungang Liu, Student Member, IEEE, and Oliver W. W. Yang, Senior Member, IEEE, “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.