ACTIVE QUEUE MANAGEMENT (AQM) AND ADAPTIVE NEURO FUZZY INFERENCE SYSTEM (ANFIS) AS INTRANET TRAFFIC CONTROL

Mila Kusumawardani

Telecommunication Engineering, State Polytechnics of Malang,
INDONESIA.

wardani.mila@ymail.com

ABSTRACT

Congestion (traffic congestion) is one problem that requires continuous development of control system, including the application of artificial intelligence. This study proposes intranet traffic congestion control using algorithm of Active Queue Management (AQM) and Adaptive Neuro Fuzzy Inference System (ANFIS). The control improved router's ability in determining congestion level through marking probability, enabling packet receiver to identify traffic load. Computer network representing intranet and traffic generation was formed based on simulation using Borland C++ Builder. The average of queue length (first variable: average) was simulated between 0-750 packets with minimum limit of 250 packets. Comparison between current and previous packet lengths (second variable: change) was calculated between (-200) and 200 packets. ANFIS was designed, trained, and tested using Mat lab, consisting of 2 inputs (average and change), and was achieved from reply packet as the result of traffic generation while the output produced was probability. Probability was suitable with certain congestion levels (low, medium, and high). Results of simulation showed that with the probability, traffic generation reduced transmission that, in turn, affected lower and delayed queue length. Transmission with low congestion showed queue length of 20.512%, medium congestion reduced queue length to an average of 44.971%, and high congestion reduced queue length up to 48.52%. Reduced queue length shortened delay to an average of 0.00445 seconds.

Keywords: Traffic, congestion, AQM, ANFIS

INTRODUCTION

These guidelines include complete descriptions of the fonts, spacing, and related information for producing your proceedings manuscripts. Please follow guidelines and email your paper Communication using the media of computer network based on TCP/IP (Transmission Control Protocol/Internet Protocol) in a private or global network has become a very essential need. This causes high demand for communication which means high traffic in network. Further, this results in compaction that, in turn, causes traffic congestion. Therefore, there are some methods designed to control congestion.

The essentiality of congestion control in computer network, including intranet, facilitates the development of queue control mechanism for router called Active Queue Management (AQM). Hollot, Misra, Towsley, and Gong (2002) conducted a study on controller's performance in router AQM named Random Early Detection (RED), Proportional (P), and Proportional Integral (PI). This controller had a weakness in delayed stabilization. A further study to handle this weakness was conducted by Aoul, Nafaa, Negru, and Mehaoua (2004). In this study, fuzzy adaptive was set to the controller AQM. Another study by Piedra, Chicaiza, López, and García (2008) analyzed the application of artificial neural network in internet traffic engineering with one of the purposes was to minimize congestion.

From previous studies, it was identified that the mechanism of AQM at router encouraged the development of various controllers, including adaptive controller using artificial intelligence. Smart system used was fuzzy which has been known not having learning and adapting ability. This ability is known possessed by artificial neural network that, from previous studies, was proven applicable to internet traffic engineering. Integration of the two methods creates hybrid method neuro-fuzzy that combines the strengths of both methods.

Based on the above description, this study was aimed at discussing congestion control at computer network traffic. The control was expected to enable router to identify congestion level through marking probability.

CONTEXT AND REVIEW OF LITERATURE

Review of Literature

A study by Feng (1999) about the development of congestion control and queue management algorithm in internet showed that the mechanism of AQM – RED was very possible to be developed to attain improved sensitivity in determining congestion level in network.

Another study about controller of adaptive fuzzy was conducted by (Aoul et. al. 2004). This study introduced adaptive fuzzy control algorithm named Fast Adaptive Fuzzy AQM Controller (FAFC). Compared to classic adaptive controller (RED and PID), FAFC could give quicker response.

A study to predict AQM using neural network was conducted by Alasem, Hossain, and Awan (2007). This study was focused on network congestion control related to the increase number of traffic with varied QoS demands. This study was designed to control adaptive Smith Predictor using neural network for AQM, with Smith Predictor used to handle system weakness such as delay.

In 2007, Khouki and Cherkaoui conducted a study aimed at providing smart solution to congestion control at wireless ad hoc network. This study developed fuzzy logic of congestion control mechanism to support multimedia application and non-real time traffic service. Fuzzy logic was used in the threshold management of buffer so that it could adapt to the dynamic condition of traffic.

A study by (Piedra et.al. 2008) was conducted on neural network application of traffic engineering. This study showed various approaches in neural network application to manage resources and to control internet congestion.

Traffic Congestion Control

Congestion is defined in two different perspectives: from users' point of view and from network's point of view. From users' point of view, Keshav (1991) claims that network is considered experiencing congestion when users' utility decreases as the result of increased network load. Yang and Reddy (1995) defines congestion from the point of view of network, and they define congestion as reduced performance due to saturated resources (communication link, processor cycle, and buffer of memory). Reduced performance can be in the form of high delay or low throughput.

Chiu and Jain (1989) classify congestion control approaches into two, namely congestion avoidance and congestion recovery. The mechanism of congestion avoidance enables the network to operate in an optimal area (low delay and high throughput), keeping the network from congestion. On the contrary, the mechanism of congestion recovery allows the network to recover from congestion status (high delay and losses, low throughput). Although one network uses the strategy of congestion avoidance, the congestion recovery scheme is still needed to maintain throughput, especially when there are changes in the network that possibly cause congestion.

Congestion control can be network-assisted (Chrysostomou, 2006). In this congestion control, router gives explicit feedback to sender about the congestion status in the network. This feedback can be in the form of single bit that indicates the occurrence of congestion or in the form of multi-bits that gives feedback of complete information about the network condition. Congestion information can be achieved from router that makes marking on a field in the data packet header, indicating the occurrence of congestion and, in turn, sending information back to the source as notification.

The method of giving binary feedback to the source about the network condition is Explicit Congestion Notification (ECN) introduced by Ramakrishnan, Floyd and Black (2001). This method provides a mechanism to early detect the occurrence of network congestion. ECN works together with AQM, a router used to detect congestion before overflow occurs. ECN enables router to mark the packet by setting the bit at IP header and TCP to produce signaling between sender, router and receiver.

IP header produces 4 codepoints from 2 bit in field ECN. ECN-Capable Transport (ECT) codepoints '10' dan '01' are set by the sender to indicate the ability of transport protocol (ECN capable). Sender can use one of the available codepoints. Not-ECT codepoint ('00') indicates that the packet data does not use ECN. Codepoint '11' was set by the router to indicate congestion at the end-node.

AQM (Active Queue Management)

Congestion control mechanism used before AQM was Drop Tail (DT) that dropped all of the entering packets over the buffer's capacity. DT has a weakness that causes global synchronization, lockout, and full queues (Chrysostomou, 2006). Global synchronization occurs when a number of sources concurrently reduce the rate and cause dropping packet of the router. Phenomena lockout occurs when DT enables a small number of connections to dominate allocated queues.

Using DT queue management, full queues can occur for a long period of time that is when router sends congestion signals to sources to notify that buffer is full. The most possible solution to reduce the weakness of DT is having packet drop before the queue is full, so that the source can respond to the congestion before buffer overflows. This encourages the development of congestion control mechanism to manage queues in router, and this leads to the invention of AQM mechanism that offers better performance than what DT has offered.

Different from DT that notifies congestion after overflow, AQM provides information about congestion when it is predicted going to occur. Another difference is that when DT only does packet dropping, AQM can do packet dropping or packet marking as the signal of congestion. For the mechanism of packet marking, AQM work together with ECN in setting header's bit and send it to the source.

The main characteristics of AQM, as mentioned by (Hollot, et. al., 2002), include:

a. Efficient queue utilization. Queue has to avoid overflow that can cause packet loss and unwanted retransmission; otherwise, this can cause emptiness due to link underutilization.
b. Queue delay. One thing that needs to be maintained is low queue delay.

c. Robustness. AQM scheme is needed to manage system behavior to be tough against variations that may occur, including a variation in the number of TCP sessions or a variation in the propagation delay and link capacity.

RED is the first AQM algorithm invented (Floyd & Jacobson, 1993). RED arranges both minimal and maximal thresholds to do drop/mark in router queue (as shown in Figure 1 and 2). When the average queue length exceeds threshold, RED will do packet dropping/marking with drop/mark probability based on the average queue. When it exceeds maximal threshold, every packet will be rejected.

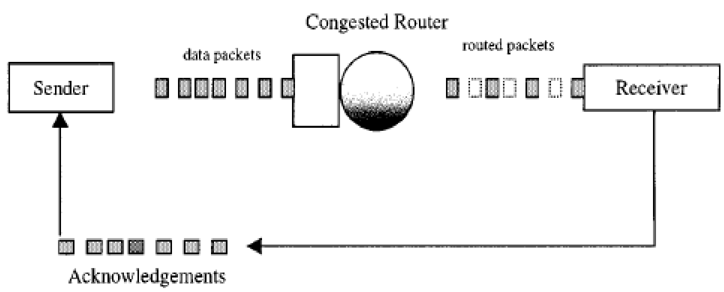

Figure 1. Congestion Control System

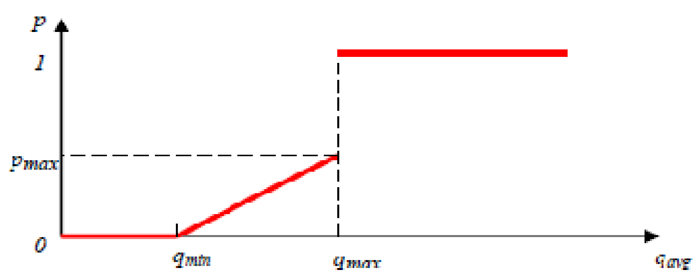

Figure 2. Rule of AQM-RED Control

The average queue length is updated using the method of Exponential Weighted Moving Average (EWMA), as presented in equation 2.1.

$$q_{avg}^{new} = (1-w).\,q_{avg}^{old} + w.\,q_{inst} \;\ldots\ldots\; (2.1)$$

Notes:

q_{inst} = instantaneous queue length

w = averaging weight

Durresi, Sridharan, and Jain (2007) state that the node that receives congestion notification of Explicit Congestion Notification (ECN) will respond by reducing packet delivery with varied reduction factor depending on the level of congestion received. This is shown in Table 1.

Table 1. Reduction Based on Congestion Level

Congestion Level	*Reduction Factor*
Incipient Congestion	20%
Moderate Congestion	40%
Severe Congestion	50%

Adaptive Neuro Fuzzy Inference System (ANFIS)

ANFIS is an adaptive network that functionally is the same as fuzzy inference system using the method of TSL/Sugeno. The network consists of a number of connected nodes. Each node is responsible for the processing unit, and link among nodes represents communication link. Nodes are adaptive, the output of which is affected by the modified parameter. The learning rule defines the way parameter is updated to reduce errors.

The architecture of ANFIS, as shown in Figure 3, comprises five layers. Layer 1 and 4 are square nodes, which indicate their adaptive characteristic. Other layers are circle and not adaptive.

ANFIS is represented mathematically using fuzzy model of Sugeno with 2 inputs (x and y) and 2 if-then rules. Regarding the premises and consequences in the if-then rule, parameter layer 1 is the premise while parameter layer 4 is the consequence.

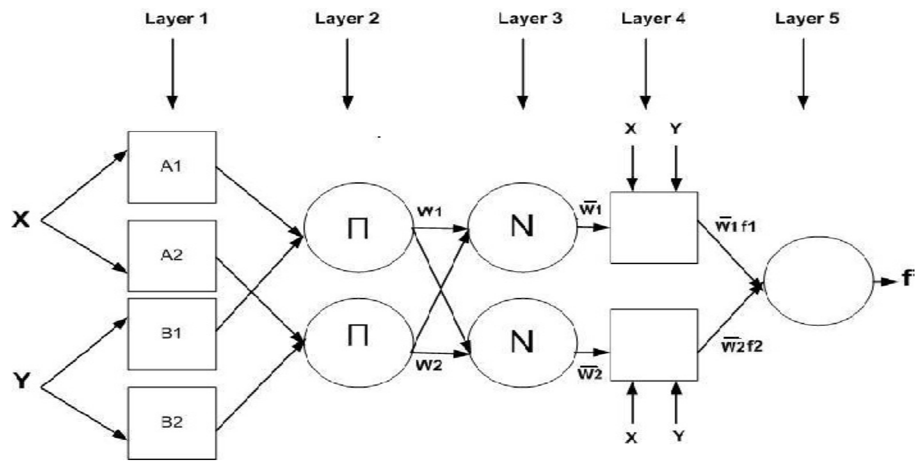

Figure 3. Architecture of ANFIS

DESIGN

System is the simulation of computer network with changeable number of hosts and servers suited to simulation's need. Between the hosts and the servers there are two routers. Router 2 (close to the server) applies AQM and ANFIS algorithm. AQM enables the router to indicate traffic congestion. The mark signaling congestion is the probability named marking probability.

ANFIS is used to identify the value of probability. Input needed by ANFIS consists of two variables, namely queue length average and queue length change. Probability will determine the amount of traffic reduction as presented in Table 1.

Network Design

The system designed comprises N host as the sender, 2 routers, and M server as the receiver, with the values of M and N the same or different. Host sends request to server through router while server receives request and sends reply back to host. Under an assumption that reply data have bigger capacity than request, router getting congestion control is router 2 (see Figure 4).

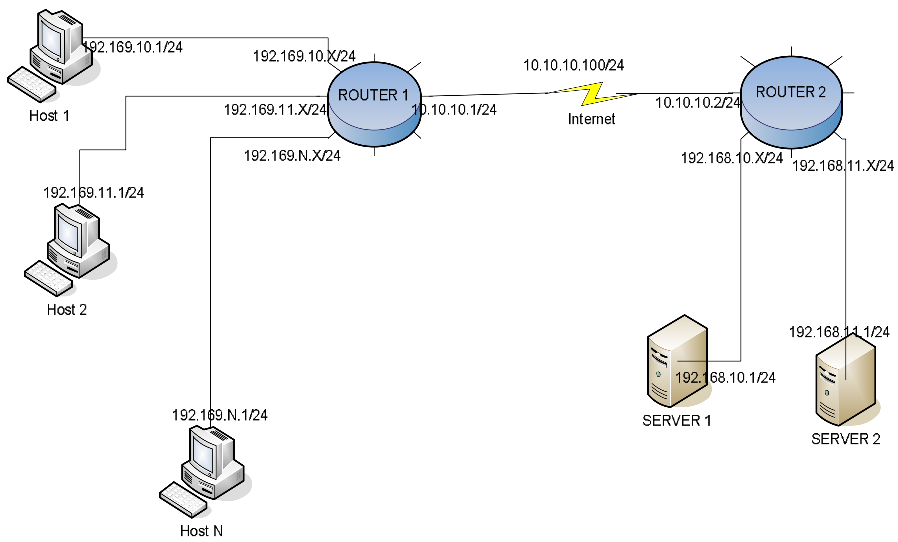

Figure 4. Network Design

The system algorithm includes:

1. Initiation of router memory
2. Traffic generation (request) in host
3. Delivery of request packet from host to router 1
4. Forwarding request packet from router 1 to router 2
5. Delivery of request packet from router 2 to server
6. Traffic generation (reply) in server
7. Delivery of reply packet from server to router 2
8. Calculation of queue length average (avg) by router 2
9. Determination of marking probability
10. Notification of congestion based on marking probability
11. Forwarding reply packet from router 2 to router 1
12. Delivery of reply packet from router 1 to host

Design of Traffic Generation System

The algorithm of traffic generation consists of:

1. Formation of socket
2. Determination of application port
3. Determination of IP source and target
4. Determination of data packet

Design of Congestion Control Using ANFIS

Both inputs of ANFIS (average and change) become the premise parameters at if-then rule; this is shown in Table 2. Each premise parameter is divided into three Membership Functions (MF). Output of ANFIS is marking probability (valued between 0.00 and 1.00) which becomes the consequent parameter as shown in Table 3. The consequent parameter is divided into three membership functions.

Table 2. Premise Parameter

Input		*Premise Parameter*	
		A1	Low
X	Average	A2	Medium
		A3	High
		B1	Reduced
Y	Change	B2	Unchanged
		B3	Increased

Table 3. Consequent Parameter

Output		*Consequent Parameter*	
	0.00 - 0.35		*Low Congestion*
Marking Probability	0.36 - 0.70	Level of Congestion	*Medium Congestion*
	0.71 - 1.00		*High Congestion*

Design of Congestion Notification System

Notification regarding congestion is received from ANFIS output that is changes into 2 bit each of which places bit 6 (bit ECT, ECN Capable Transport) and bit 7 (bit CE, Congestion Experienced) at IP header, as shown in Table 4. Notification is conducted by router 2.

Table 4. Notification of Congestion at IP Header

Bit 6 (ECT)	*Bit 7 (CE)*	*Congestion Level*
0	1	Low Congestion
1	0	Medium Congestion
1	1	High Congestion

Notification is sent by router 2 to router 1 so that router 1 knows router 2's congestion level. Responses given by router 1 reduce next packet delivery, suiting it to reduction factor shown in Table 1.

Formation of ANFIS Structure

ANFIS structure consists of 2 inputs (average and change) and 1 output (marking probability) with the following data:

Tabel 5. Input (premise parameter 1)

Parameter	*Value*	
Variable	Average	
	MF1	250-450
Range	MF2	400-600
	MF3	550-750

Table 6. Input (premise parameter 2)

Parameter		*Value*
Variable		Change
Range	MF1	(-200) – 0
	MF2	(-20) - (20)
	MF3	0-200

Table 7. Rule Formation

		Average		
		Low	Medium	High
Change	Reduced	1-Low	4-Low	7-Medium
	Unchanged	2-Low	5-Medium	8-High
	Increased	3-Medium	6-High	9-High

Through ANFIS Editor's windows, training data is loaded from workspace. The number of coupled data of input-output used for training is 25. Results of the training are illustrated in Figure 5, showing an error average of 0.0043.

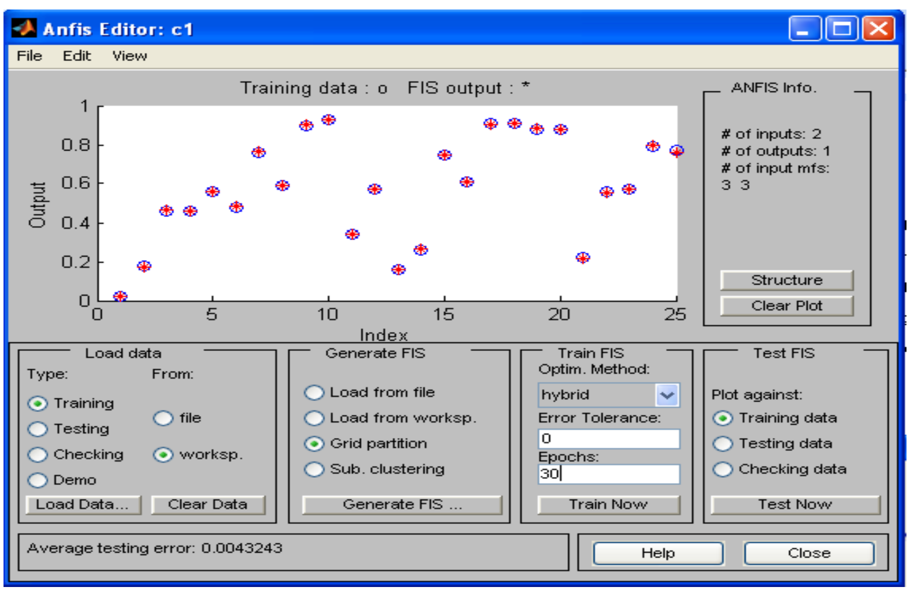

Figure 5. Results of Data Training

SIMULATION AND ANALYSIS

Simulation

Simulation was conducted by providing variations in the input of the number of host (user) delivering data packet of request because this would directly affect the number of queue. The amount of request data was 65 kB. In this simulation, there were up to 200 user 3. The configuration used for the simulation was: a number of host (H1, H2, ..Hn) connected to

ISSN-L: 2223-9553, ISSN: 2223-9944
Vol. 4 No. 5 September 2013

router 1; then, router 1 was connected to router 2. Router 2 was connected to a number of servers (S1, S2, ..Sn).

Results of the simulation were analyzed on its queue average and the change as well as the output probability produced by ANFIS. This output was compared to the results achieved from fuzzy. Analysis was also conducted to identify the queue length after the application of ANFIS. This was done in order to see the applicability of the existing theory. Further analysis was related to delay before and after ANFIS was applied.

Results gained from simulation with 50 users are illustrated in Figure 6 and table 8-9.

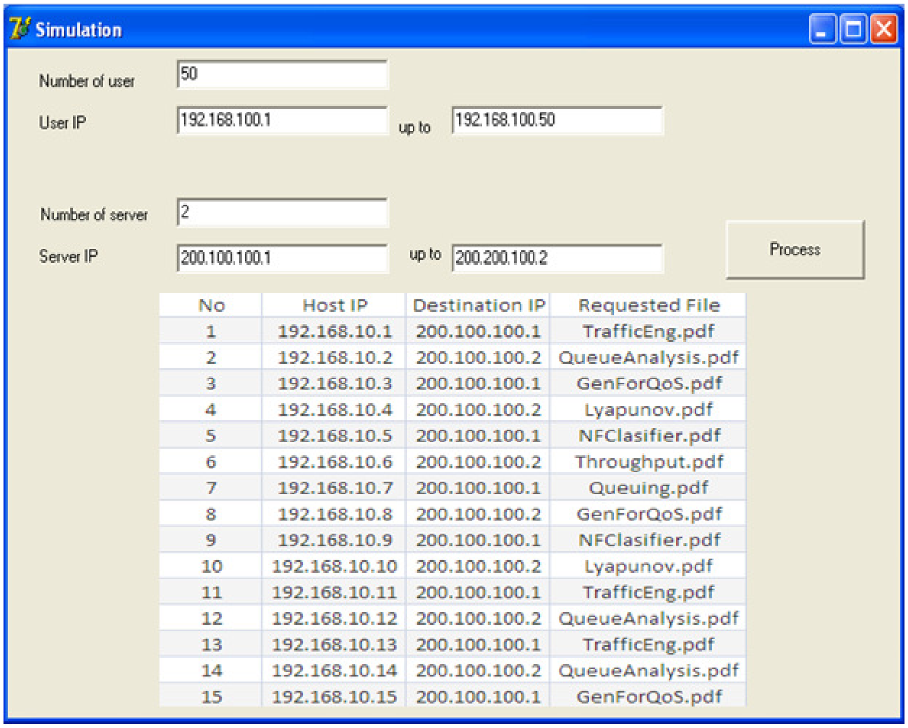

Figure 6. Input of 50 User

Table 8. Average of Queue Length

Second	*Average Before ANFIS*	*Change*	*Average After ANFIS*	*Marking Probability*	*Notification*
1	782	--	626	1	1 0
2	653	-129	458	0,74	1 0
3	672	19	404	0,88	1 1
4	1043	371	522	1	1 1
5	915	-128	366	1	1 0
6	934	19	281	1	1 1
7	805	-129	242	1	1 0

Table 9. Delays Before and After ANFIS

Delay (second)	
Before ANFIS	*After ANFIS*
0,0077599999	0,0014999998
0,0072200000	0,0026400001
0,0078400001	0,0038000001
0,0070000002	0,0017800000
0,0071200002	0,0034600003
0,0071200002	0,0022200001
0,0082000000	0,0001499997

Analysis

The first analysis conducted was comparing the value of marking probability achieved from using ANFIS and the value achieved from using fuzzy. Using the simulated example with 50 users, results of the comparison are shown in Table 10. From the findings, it was identified that ANFIS was capable of providing more accurate values than what fuzzy did; this was because ANFIS had the capability to learn while fuzzy produced more random probability.

Table 10. Comparison of Outputs (marking probability) between ANFIS and Fuzzy

Input		*Output (marking probability)*	
Average of Queue Length	*Queue Length Change*	*Average of Queue Length*	*Queue Length Change*
782	-129	1,00	1,00
653	129	0,74	1,00
672	19	0,88	0,996
1043	371	1,00	1,00
916	-128	1,00	1,00
934	19	1,00	1,00
805	129	1,00	1,00

Marking probability was achieved from the previous analysis which was related to the reduced number of data packets transmitted as shown in Table 1. Impact of this reduction was shorter queue length (congestion level), as shown in Table 11. Generally, it could be identified that at level low and medium, the reduction of transmission that occurred was higher than as determined; yet, at the high level, the transmission did not meet minimal requirement.

Table 11. Reduction of Data Packet Transmission

Number of Users	*Reduction of Transmission at Congestion Level (%)*		
	Low	*Medium*	*High*
3	23,587		
5	17,436	20,000	
10		69,941	44,955
50			48,499
100			49,946
200			49,969
Average	20,512	44,971	48,290

Further effect that occurs due to transmission reduction was reduced delay, as presented in Table 12. From the table it could be identified that, as a whole, with varied numbers of users, reduced delay could reach as much as 61.782%.

Table 12. Reduced Delays

Number of Users	*Average of Reduced Delays*	
	Second	*%*
3	0,00130	28,713
5	0,00273	52,836
10	0,00546	72,666
50	0,00524	69,800
100	0,00601	73,771
200	0,00595	72,903
Average	0,00445	61,782

CONCLUSION

Based on the abovementioned analysis and discussion, it could be concluded that:

1. Training ANFIS can reach the lowest error value of 0.0043 which is predicted not to cause mistakes in output of ANFIS which is known as probability marking.

2. Marking probability is achieved from ANFIS or fuzzy. Results when using ANFIS is more varied than results when using fuzzy. Thus, it can be proven that, in determining marking, ANFIS shows a higher accuracy than fuzzy does.
3. Marking probability influences the reduced number of transmitted data packets. The average percentage of reduced transmission at low congestion was 20.51%, at the low congestion 44.971%, and at high congestion 48.290%.
4. Reduced number of delivered packets impacts on reduced delay. Based on the simulation, it could be seen reduced delay of 0.00445 seconds.

SUGGESTIONS

For further research, it is suggested that:

1. The number of training data of ANFIS is increased in order to gain a higher possibility of less error.
2. Variation of data (number of host, number of servers, and capacity of application) be increased in order to get better variation of delay and to meet various values of marking probability.

REFERENCES

Alasem, R., Hossain M. A. & Awan, I. (2007). Intelligent AQM Predictive Controller Using Neural Network. *NGMAST (Next Generation Mobile Application, Service, and Technology) Conference. (*DOI :10.1109/NGMAST.2007.4343422

Aoul, Y. H., Nafaa, A., Negru, D. & Mehaoua, A. (2004). FAFC: Fast Adaptive Fuzzy AQM Controller for TCP/IP Network. *Global Telecommunication Conference-GLOBECOM'04 IEEE.* Vol 3, 1319-1323. ISBN: 0-7803-8794-5. DOI: 10.1109/GLOCOM.2004.1378200

Chiu, Dah-Ming & Jain, R. (1989). Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Network*. Computer Network and ISDN System 17. North Holland:Elsevier Science Publisher B.V*

Chrysostomou, C. (2006). *Fuzzy Logic Based AQM Congestion Control in TCP/IP Network.* (Unpublished doctoral dissertation), University of Cyprus

Duressi, A., Sridharan, R. & Jain, R. (2007). Adaptive Multi-level Explicit Congestion Notification. *International Journal High Performance Computing and Networking, 5*(1/2), 3-11.

Feng, W. (1999). *Improving Internet Congestion Control and Queue Management Algorithm.* (Unpublished doctoral dissertation), University of Michigan.

Floyd, S. & Jacobson, V. (1993). RED Gateways for Congestion Control. *IEEE/ACM Transaction on Networking, 1*(4), 397-413. DOI: 10.1109/90.251892

Hollot, C. V., Misra, V., Towsley, D. & Gong, W. (2002). Analysis and Design of Controlers for AQM Router Supporting TCP Flows.*IEEE Transactions on Automatic Control*, *47*(6), 945-959. DOI: 10.1109/TAC. 2002. 10083860

Keshav, S. (1991). *Congestion Control in Computer Network*. (Unpublished master's thesis), University of California, Berkeley

Khouki, L. & Cherkaoui, S. (January 23-25, 2008). Intelligent Solution for Congestion Control in Wireless Ad Hoc Networks. Paper presented at the *Wireless On-demand Network System Conference, Garmisch-Partenkirchen, Germany, and Proceeding* (pp. 10-19).

Piedra, N., Chicaiza, J., Lopez, J. & Garcia, J. (2008). Study of the Application of Neural Network In Internet Traffic Engineering. *Information Science and Computing, Institute of Information Theories and Applications FOI ITHEA*, 33-47.

Ramakrishnan, K., Floyd, S. & Black, D. (2001). The Addition of Explicit Congestion Notification (ECN) to IP – RFC 3168. *The Internet Society, RFC Editor*, United States

Welzl, M. (2005). *Network Congestion Control – Managing Internet Traffic.* New Jersey: John Wiley & Sons Inc.

Yang, C. & Reddy, A. (1995). A Taxonomy for Congestion Control Algorithms in Packet Switching Network. *IEEE*, *9*(4), 34-45. DOI: 10.1109/65.397042