IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)
e-ISSN: 2278-2834,p- ISSN: 2278-8735.Volume 14, Issue 6, Ser. I (Nov.-Dec. 2019), PP 45-53
www.iosrjournals.org

An Application of Hedge Algebras in Active Queue Management for TCP/IP Multiple Congestion Networks

Thu May Duong[1], Hien Trinh Nguyen[2], Hop Le The[3]
[1]Department of Communications and Networking, University of Information and Communication Technology,Thai Nguyen, Vietnam
[2]Department of Computer Science, University of Information and Communication Technology,Thai Nguyen, Vietnam
[3]Northern MobiFone Network, The MobiFone Telecommunications Corporation, Vietnam

Abstract: *In this paper, authors proposed a new algorithm to improve the Random Exponential Marking on TCP/IP network by using Hedge Algebras Controller. Using NS-2 simulator, the results are verified and compared to some traditional AQM algorithms on the same multiple-congested TCP/IP network. The operational efficiency of AQM algorithms were evaluated in a large network traffic conditions, dynamical changes and large transmission delay.*

Date of Submission: 20-12-2019 Date of Acceptance: 03-01-2020

I. Introduction

The explosion of customer information needs and the trend of integration of service networks on a TCP / IP network has posed tremendous challenges for the current communication network. Customers increasingly require diversity in the number of services as well as perfection in service quality. Therefore, on the core TCP / IP network, it is necessary to perform management problems such as routing, resource management, quality management, and network traffic control. These problems need to be answered immediately, effectively and intelligently for different types of services [9],[13].

In the field of congestion control and traffic management, primitive algorithms for TCP/IP such as slow start, congestion avoidance, fast recovery, fast retransmit do not satisfy the needs in reality because they are source side controlling congestion algorithms. Therefore, Active Queue Management (AQM) algorithms (installing on routers on the TCP/IP network) were proposed to enhance the capacity of TCP/IP in bandwidth management and congestion control. The purpose of this problem is to provide a mechanism for proactively discarding packets of incoming traffic in an appropriate manner based on the current status of the queue or / and the speed of incoming traffic sources. The removal of this package is intended to maintain the average (or immediate) length of the queue at an appropriate reservation level. Stabilizing the length of the queue will make a number of TCP / IP network operational parameters such as packet loss rate, route utilization, average delay and variation of variation delay within a reasonable range. This will both ensure that there is no congestion on the network and at the same time provide and maintain the best quality of service for different traffic flows.

Currently, there are more than 80 AQM algorithms published and can be divided into three main categories (queue management based on queue length, queue management based on incoming traffic speed, queue management based on a combination of both length and speed of incoming flow) [12], [13].

In order to improve the effectiveness of AQM algorithms, many other methods have been published. Most of these methods are upgrades or combinations of traditional methods. The results have somewhat met the requirements of the AQM problem. However, these methods still need to be improved to simplify implementation, improve intelligence in maintaining the average queue length in dynamic network conditions and ensure fairness when discarding packets for incoming traffic. Therefore, many recent AQM algorithms have been proposed, including the use of artificial intelligence tools such as neural networks and fuzzy logic [7], [9].

Hedge algebra is a fuzzy logic-like approach. Therefore, it is applied in many practical problems, especially in the construction of controllers. Its advantage is to ensure the orderly relationship between linguistic values appeared in the system of inference rules [4]

Because of those reasons, in this paper, authors propose a model using Hedge Algebras Controller (HAC) to improve the REM_AQM. Using NS-2 simulator, the results are verified and compared to some traditional AQM algorithms (ARED, PI, REM) on the same multiple-congested TCP/IP network topology.

II. REM Algorithm

REM is a traditional AQM algorithm based on both queue length and flow rate using a congestion metric called "price", which is calculated from system performance parameters such as queue length, link capacity, speed of the packet arrived. REM periodically samples the router's queue and updates congestion data to reflect the difference between the actual queue length versus the reference queue length, the speed of incoming packets and the link capacity at each connection. For the k-th interval of the router queue, the congestion metric p(kT) at the time kT is calculated by [2]:

$$p(kT) = \max(0, p(k-1)T + \gamma(\alpha(q(kT) - q_{\text{ref}}) + x(kT) - c)) \quad (1)$$

where c is the capacity of link, *q(kT)* is the actual queue length, $q_{ref}$is the reference queue length and *x(kT)* is the speed of the incoming packet. The dropping packet probability is calculated as follows [2]:

$$prob(kT) = 1 - \phi^{-p(kT)} \quad (2)$$

The parameters for REM are described in Table 1.

Table 1.REM parameters

Parameters	**Description**
q_{ref}	Reference queue length
α and γ	Constants for computing the "congestion price"
ϕ	Constant for computing the drop probability
T	Sampling interval

III. Hedge Algebra Controller For Rem_Aqm Problem

In this section, we describe how to build the HAC for the REM problem. The principle model of the HA-based controller is described as in Figure 1 [6].

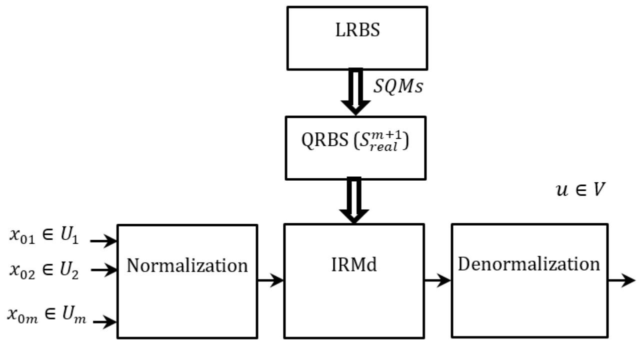

Figure 1. The diagram of the HA-based controller

- Normalization: standardize values of the input variables in the semantic domain.
- LRBS: Linguistic rule-based system of the controller (like SAM table in fuzzy logic).
- QRBS: Quantifying rule-based system of linguistic values which is computed by Semantic Quantitative Mapping functions (S^{m+1}_{real}).
- IRMd (Interpolation Reasoning Method): Interpolation on the "hypersurface" S^{m+1}_{real} .
- Denormalization: convert semantic control value to the domain of variable real value of the output variable.

This controller is also based on the "price" parameter as for the REM algorithm. In this HAC, we use two inputs, one for the price at the current time ($P_r(kT)$) and one for the price at the previous cycle time ($P_r(kT-T)$). Based on these two input values, the HAC will determine the value of the droppingpacket (DP) that represents the output of this controller.

According to HA approach, the HAC described in Figure 2 is designed by the following steps [3], [4], [12]:

Step 1: Identify the components of the hedge algebras for input and output variables.

Hedge algebras for linguistic variables $P_r(kT)$, $P_r(kT-T)$, *DP*include:

1) The set of generating element G = {S, B}, with c^- = S (Small) and c^+ = B (Big).
2) The elements 0, W and 1 are the smallest elements, the neutral element and the greatest element respectively.
3) The set of selected hedges: H^-= {L (Little)} and H^+ = {V (Very)}.
4) The fuzzy parameters of hedge algebras are determined by Trial and error according to Table 2.

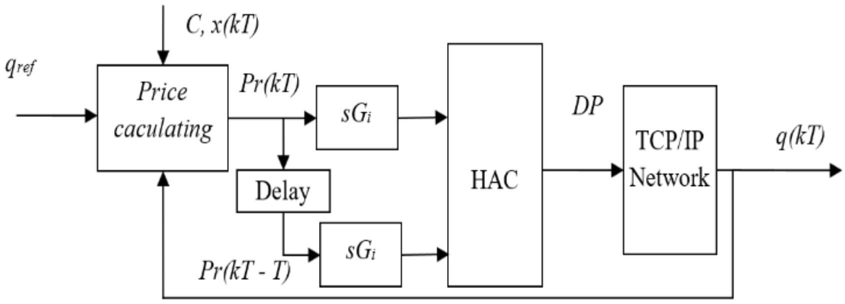

Figure 2. HAC for REM AQM problem

Table 2. Fuzzy parameters for the HAC variables

	$P_r(kT)$	$P_r(kT-T)$	DP
$\theta=fm(S)$	0.5	0.5	0.5
$\alpha=\mu(L)$	0.32	0.21	0.73

Table 3. Sign relationship of the hedges and the generating elements

	V	L	S	B
V	+	+	−	+
L	−	−	+	−

Step 2: Set up the rule set with linguistic label of hedge algebras

The linguistic labels in hedge algebras is shown as follows:

$P_r(kT-T)=\{VS, S, \mathrm{W}, B, VB\}$

$P_r(kT)=\{VS, S, \mathrm{W}, B, VB\}$

$DP=\{VS, S, LS, \mathrm{W}, LB, B, VB\}$

Table 4. Linguistic rule-based system of the controller

$P_r(kT)$ / $P_r(kT-T)$	***0***	*VS*	*S*	***W***	*B*	*VB*	***1***
0	***0***	***0***	*VS*	*VS*	*S*	*LS*	***W***
VS	***0***	*VS*	*VS*	*S*	*LS*	***W***	*LB*
S	*VS*	*VS*	*S*	*LS*	***W***	*LB*	*B*
W	*VS*	*S*	*LS*	***W***	*LB*	*B*	*VB*
B	*S*	*LS*	***W***	*LB*	*B*	*VB*	*VB*
VB	*LS*	***W***	*LB*	*B*	*VB*	*VB*	***1***
1	***W***	*LB*	*B*	*VB*	*VB*	***1***	***1***

Rule set deduced according to HA approach is qualitatively constructed by the control rules shown in Table 4.It can be interpreted for the approximation inference rule applied to the controller as follows:

If $P_r(kT)$ *= VS and* $P_r(kT-T)$ *= VS then DP = VS*

If $P_r(kT)$ *= VS and* $P_r(kT-T)$ *= S then DP = VS*

If $P_r(kT)$ *= B and* $P_r(kT-T)$ *= B then DP = B*

If $P_r(kT)$ *= VB and* $P_r(kT-T)$ *= VB then DP = VB*

Step 3: Computing the semantically quantifyingvalues from language terms in rule table (Table 4), building input and output relationship surface (S_{real})

With an identifying set of fuzzy parameters, semantically quantifyingvalues is recursively determined by the Semantically Quantifying Mapping function υ as follows [1], [6]:

$$\upsilon_{fm}(W)=\theta= fm(c^-), \upsilon_{fm}(c^-)=\theta-\alpha\, fm(c^-)=\beta\, fm(c^-)$$

$$\upsilon_{fm}(c^+)=\theta+\alpha\, fm(c^+) \tag{3}$$

$$\upsilon_{fm}(h_j\,c)=\upsilon_{fm}(c)+sgn(h_jc)\left\{ fm(h_jc)-\frac{1}{2}\left[1+sgn(h_1,h_j)(\beta-\alpha)\right] fm(h_jc)\right\}$$

Where:

- *fm: X → [0, 1]* is called the fuzzy measurement:

$$fm(h_j c) = \mu(h_j)\, fm(c), c \in G, j \in \{-1, 1\} \quad (4)$$

- *sgn: X → {–1, 1}*is the sign function which is recursivelydefined as follows:

If $h \in H$, $c \in \{c^-, c^+\}$ then $sgn(c+) = +1$ and $sgn(c-) = -1$ $\{h \in H+ \mid sgn(h) = +1\}$ and$\{h \in H^- \mid sgn(h) = -1\}$
$sgn(hc) = sgn(h) \times sgn(c)$

- $sgn(h_1, h_j)$ is determined through sign table (Table 3)

Basing on (3) and the parameters in Table 2, we can compute the semantically quantifying values of the language terms in rule table (Table 5) as follows:

Table 5. Semantically quantifying values of input and output variables

Pr(kT-T) \ *Pr(kT)*	0.0000 (***0***)	0.2312 (*VS*)	0.3400 (*S*)	0.50000 (*W*)	0.6600 (*B*)	0.7688 (*VB*)	1.0000 (***1***)
0.0000 (***0***)	0.0000 (***0***)	0.0000 (***0***)	0.0365 (*VS*)	0.0365 (*VS*)	0.1350 (*S*)	0.2336 (*LS*)	0.5000 (*W*)
0.3121 (*VS*)	0.0000 (***0***)	0.0365 (*VS*)	0.0365 (*VS*)	0.1350 (*S*)	0.2336 (*LS*)	0.50000 (*W*)	0.7664 (*LB*)
0.3950 (*S*)	0.0365 (*VS*)	0.0365 (*VS*)	0.1350 (*S*)	0.2336 (*LS*)	0.5000 (*W*)	0.7664 (*LB*)	0.8650 (*B*)
0.50000 (*W*)	0.0365 (*VS*)	0.1350 (*S*)	0.2336 (*LS*)	0.5000 (*W*)	0.7664 (*LB*)	0.8650 (*B*)	0.9636 (*VB*)
0.6050 (*B*)	0.1350 (*S*)	0.2336 (*LS*)	0.5000 (*W*)	0.7664 (*LB*)	0.8650 (*B*)	0.9636 (*VB*)	0.9636 (*VB*)
0.6879 (*VB*)	0.2336 (*LS*)	0.5000 (*W*)	0.7664 (*LB*)	0.8650 (*B*)	0.9636 (*VB*)	0.9636 (*VB*)	1.0000 (***1***)
1.0000 (***1***)	0.5000 (*W*)	0.7664 (*LB*)	0.8650 (*B*)	0.9636 (*VB*)	0.9636 (*VB*)	1.0000 (***1***)	1.0000 (***1***)

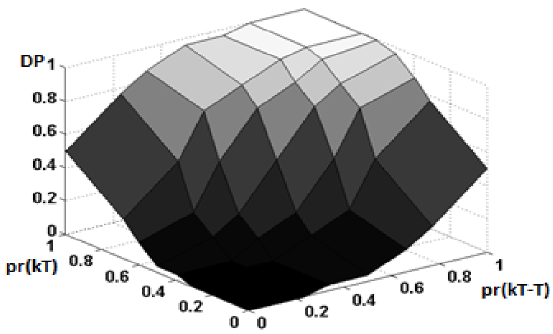

Figure 3. S_{real} of the controller HAC

IV. Simulation Results For Evaluating AQM Algorithms

In this section, we use NS-2 simulator to compare the performance of the proposed AQM with other methods such as REM, PI, ARED. Parameter values of these AQM is established by the proposals [2], [5],[9]. This is to produce a fair comparison that uses the establishments proposed by those authors.

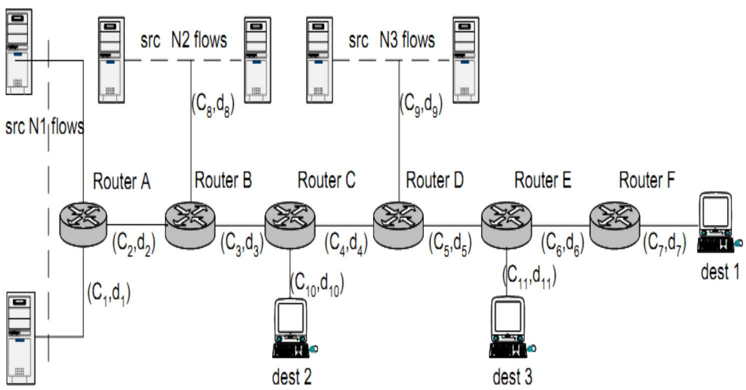

Figure 4. The multiple congestion network topology for simulation

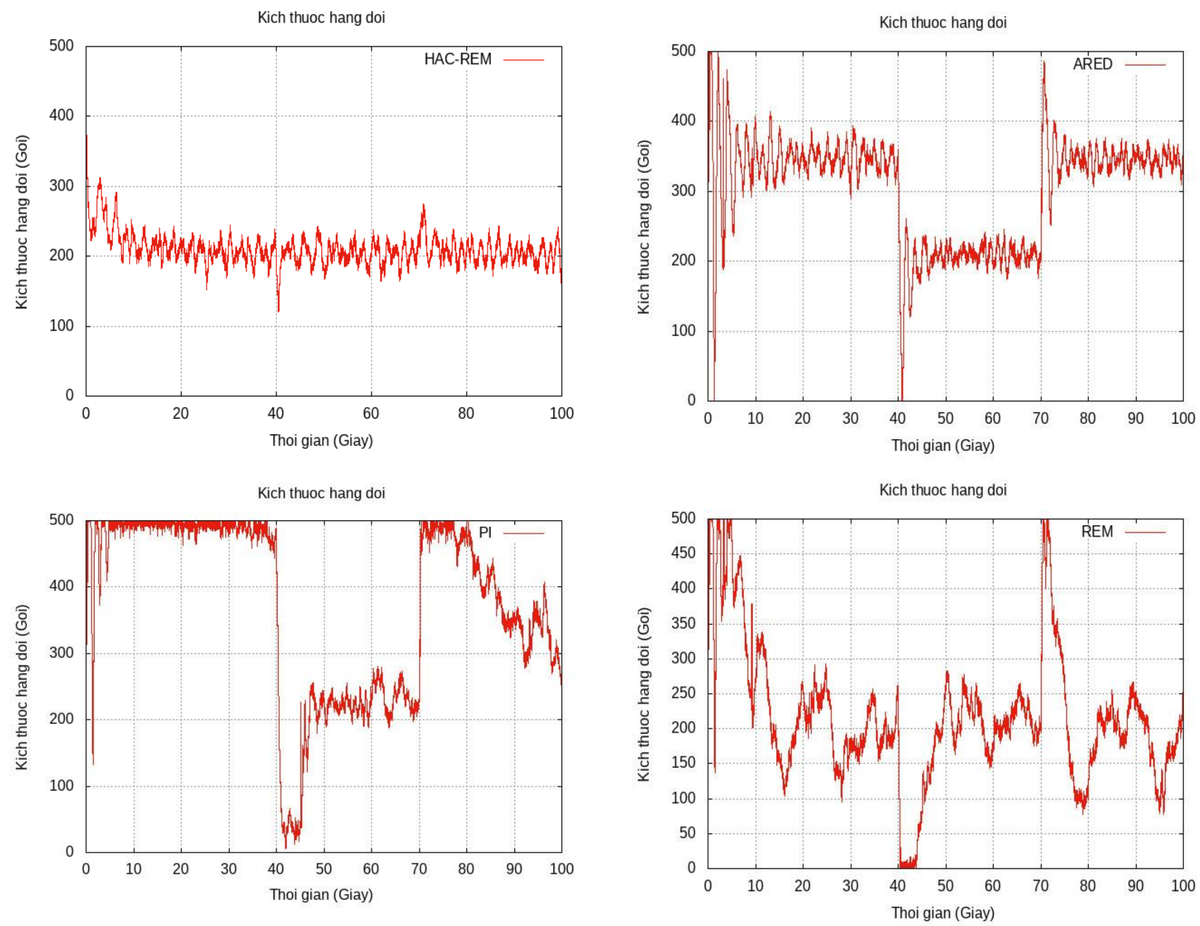

Figure 5. Queue lengths of AQM schemes at N=800, RTT=30

The network topology for the simulation is presented in Figure 4. AQM algorithm is set up with the use of TCP/Newreno. Simple Droptail is used for all other access links. Traffic and delay parameters are as follows: (C1, d1) = (C8, D8) = (C9, d9) = (100Mbps, 5ms), (C2, d2) = (C4, d4) = (C6, d6) = (15Mbps, 10ms), (C3, d3) = (15Mbps, 60ms), (C5, d5) = (15Mbps, 30ms), (C7, d7) = (C10, d10) = (C11, D11) = (200Mbps, 5ms). Flow traffic N1 ends at destination 1, flow traffic N2 ends at destination 2 and destination 3 is the end point of flow traffic N3. This network generates a cross traffic. The number of flows is initially set up as N1=100, N2=50 and N3=100. Next, the number of sources is increased to N1 = 500, N2 = 50 and N3 = 100. Finally, the number of flows is N1=500, N2=100 and N3=200. To illustrate the dynamic change of TCP / IP network. We stop half the source at 40 seconds and continue at 70 seconds to measure the response of AQM algorithms. The queue in the router has a maximum size of 500 packets. AQM algorithms all have a common goal of maintaining an average queue length of 200 packets. Figure 4 show that both bottleneck links where cross traffic exists (for example, between router B and router C, between router D and router E) behave similarly. Therefore, we choose the bottleneck link between router D and router E to analyze the results obtained.

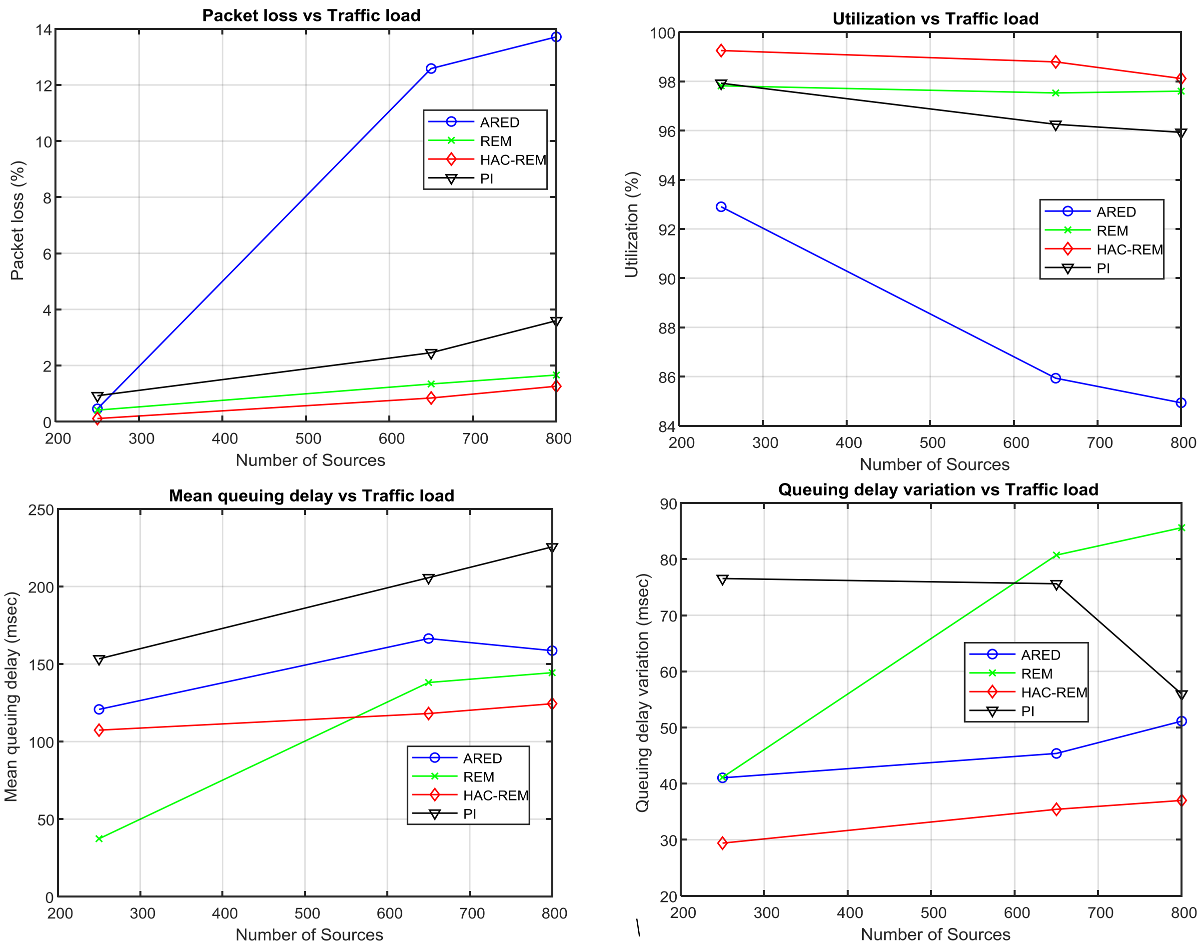

Figure 6. The changes of network parameters when the number of loads change

Figure 5 shows the change of queue length relative to AQM algorithms when the total number of traffic is the largest (N = 800). The results are presented in order from left to right and from top to bottom as HAC-REM, ARED, PI and REM. It is clear that HAC-REM control proves to be more effective than remaining AQM algorithms. It can be seen that when the total amount of traffic on the network is large and sudden changes, traditional AQM algorithms can not afford to meet. This causes the queue length to vary greatly and does not follow the reference value (200 packets).Within 100 seconds of simulation, the average length of the queue according to the HAC-REM method is the most stable. Specifically, when the traffic load decreases at 40 seconds and increases at 70 seconds, the variation of the average queue length according to the HAC-REM algorithm is minimal. This result is in contrast to the results from conventional AQM algorithms such as PI, ARED and REM.

Figure 6 illustrates the changes of the parameters during the operation of the TCP/IP network with the total number of traffic ranging from 250, 650, 800. From left to right in turn are packet loss ratio, the utilization, mean queuing delay and delay variation compared to traffic load.

Theoretically, when the router maintains the queue size at a stable value, the average packet loss rate will decrease. Queue delays are kept at an appropriate value and the variance of latency decreases. Experimental results in Figure 6 also show this. As the traffic load increases, by maintaining a more stable queue, HAC-REM is able to achieve maximum utilization of the route, along with minimal packet loss rates and minimal delay variability. The remaining AQM algorithms exhibit a slow response to queue adjustments. This results in poor network performance parameters. ARED produces the highest packet loss rate and lowest utilization. REM and PI make high queuing delay and high delay variations. Both these delay parameters greatly influence the quality of service.

Next, we evaluate the efficiency of AQM algorithms given when changing the transmission delay at the bottlenecks. We measure the network parameters when the Round-Trip Time (RTT) is 30 ms, 120 ms and 200 ms respectively. The average queue length with RTT of 200 ms is shown in Figure 7. The total number of traffic sources remains at 800. The changes of network parameters are depicted in Figure 8.

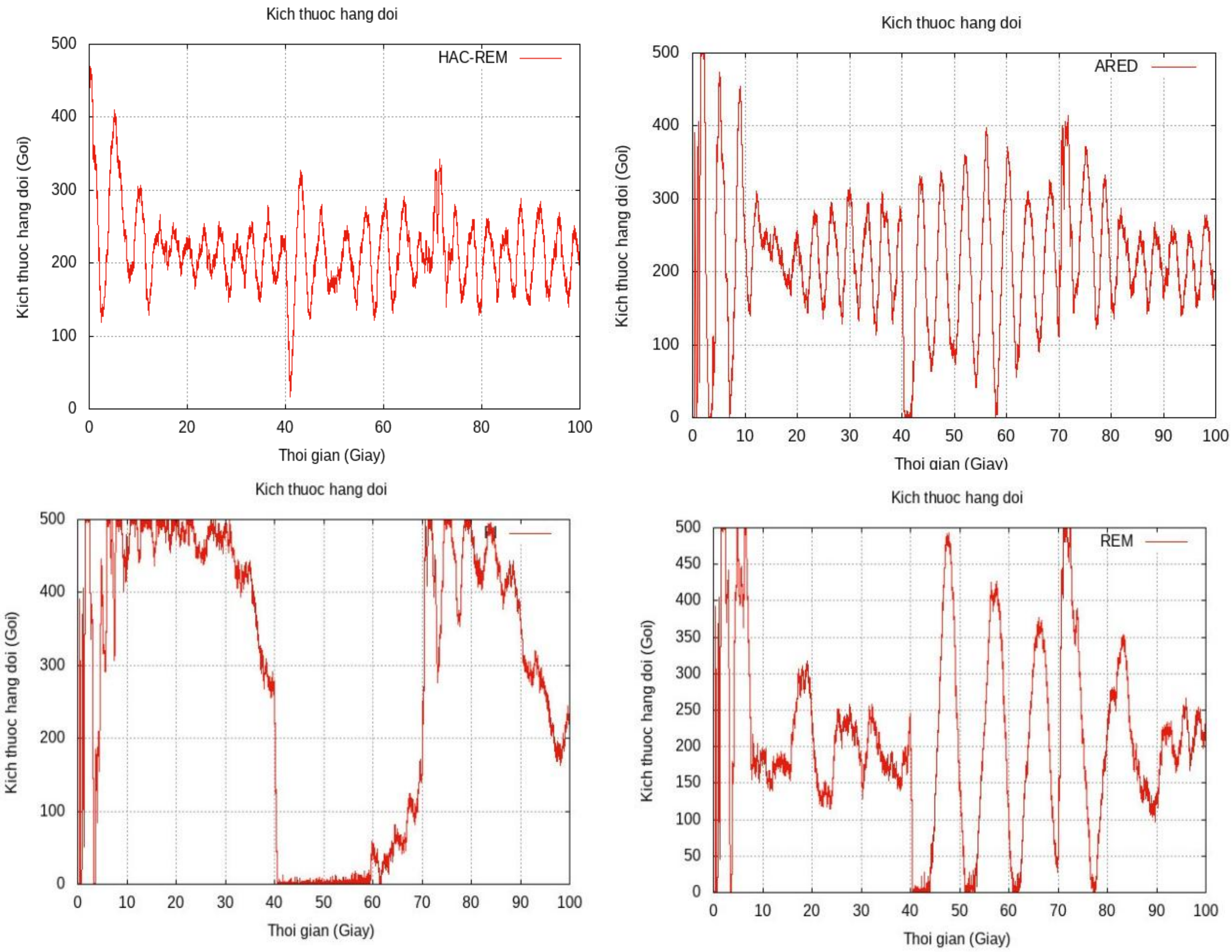

***Figure 7**. Queue lengths of AQM schemes at N=800, RTT=200*

As illustrated in Figure 7, the average length of the queue fluctuates sharply as the RTT value changes. However, HAC-REM is still the best stability algorithm. Maintaining a more stable queue length compared to other methods makes the HAC-REM algorithm have the least packet loss rate (see Figure 8). The results also show that HAC-REM achieves a high linear utilization rate, maintaining a low queuing delay and low delay variations.

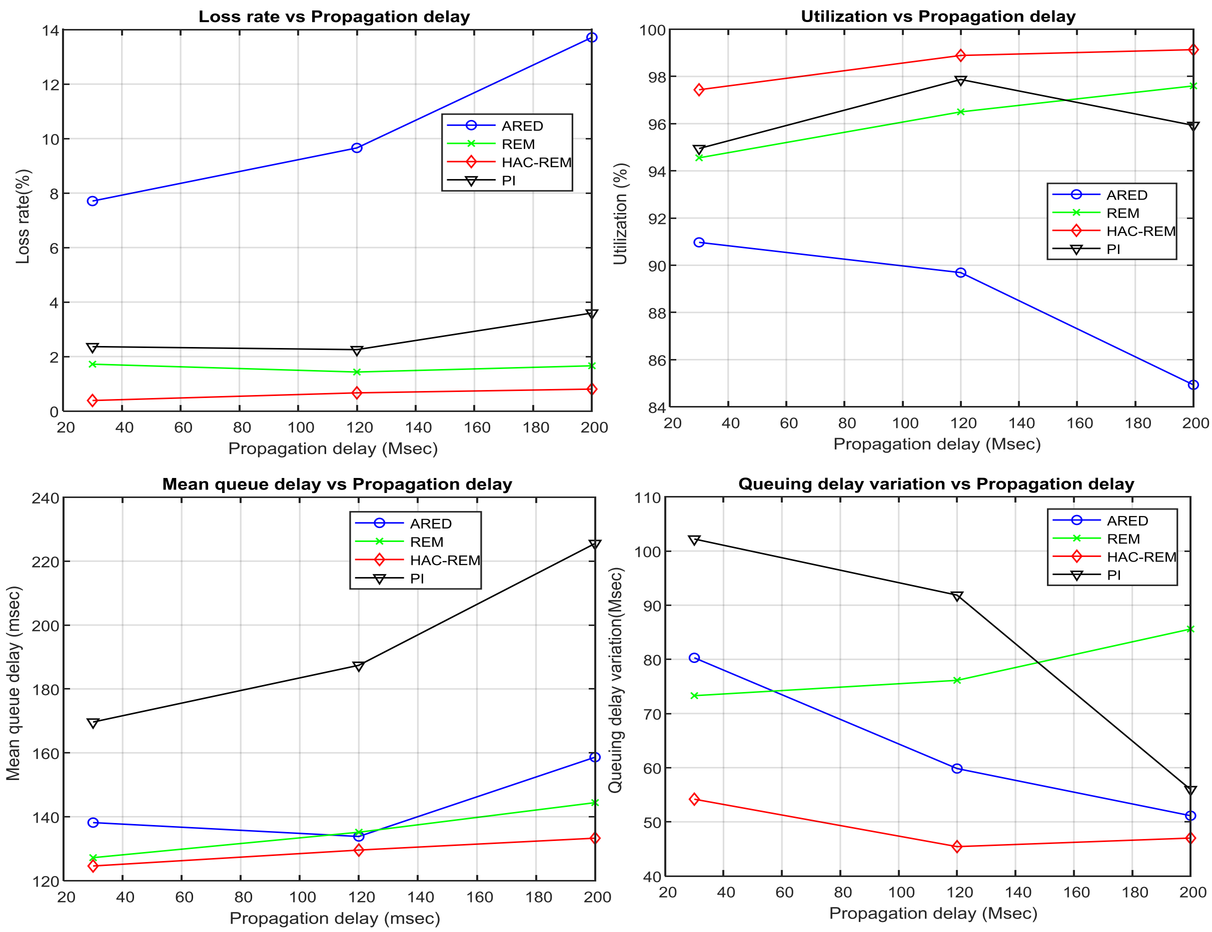

Figure 8. The changes of network parameters when RTT change

V. Conclusion

Soft computing has now become a useful tool in building systems that require intelligence with the ability to make accurate decisions in the absence of information. By accepting a certain degree of error, soft computing tools make the system design process simpler and produce better results. As one part of soft computing, HA soon proved its potential in practical control problems. However, the using of HA to solve communication network problems has not been developed extensively.

In this study, we propose a HAC for the AQM problem based on queue length and load. In essence, this is an improvement of the REM algorithm. Through simulation, it shows good performance of the HAC-REM algorithm. Compared to traditional AQM algorithms, HAC-AQM maintains a more stable queue length, thus making network performance parameters better even with the big traffic flow, big changing of network dynamics and large transmission delays.

This result is only the first step in our research process. Some issues can be further studied both theoretically and practically. In the future, we are particularly interested in some approaches as follows: Using optimal algorithms to find the optimal parameters of HA; Further compare the effectiveness of the proposed AQM method with other methods; Installing the proposed AQM algorithms and experimentally evaluate the effectiveness of these algorithms on actual routers.

Acknowledgment

This research was supported by the ICTU research grant (under project # T2019-07-03). We thank our colleagues from the ICTU who provided insight and expertise that greatly assisted the research.

References

[1]. Aoyama. T, A new generation network: Beyond the Internet and NGN, Communications Magazine, IEEE, Volume: 47 (5): 82 – 87, 2009.

[2]. Athuraliya S., Lapsley D. E., Low S. H, Random early marking for Internet congestion control. IEEE/ACM Transactions on Networking, Vol. 15(3): 48-53, 2001.

[3]. Binh Lam Hoang, Fei Luo, Duy Nguyen Tien, Phuong Huy Nguyen, "Dissolved Oxygen Control of the Activated Sludge Wastewater Treatment Process Using Hedge Algebraic Control", Biomedical Engineering and Informatics (BMEI), 7th International Conference (BMEI 2014), pp 827 – 832, 2014

[4]. C.H. Nguyen, N. L. Vu, X. V. Le, "optimal hedge-algebra-based controller: Design and application", Fuzzy Set. Syst., 159, pp. 968-989, 2008

[5]. Chrysostomou. C., Fuzzy Logic Congestion Control in TCP/IP Tandem Networks , ISCC'06 Proceedings 11th IEEE Symposium on Computers and Communications, pp.123 – 129, 2006.

[6]. Duy Nguyen Tien, Phuong Huy Nguyen, Thi Mai Thuong Duong, Thu May Duong (2018), Designing an Adaptive PID Controller for Dissolved Oxygen Control of the Activated Sludge Wastewater Treatment using Hedge Algebras, International Journal on Future Revolution in Computer Science & Communication Engineering ISSN: 2454-4248, Volume: 4 Issue: 5, pp. 177-184.

[7]. Huy Nguyen Phuong, Dung Le Ba, An application of genetic fuzzy system to improve the Random Exponential Marking algorithm in Active queue Management, 5th International Conference on BioMedical Engineering and Informatics (BMEI'12), pp. 1552-1556.

[8]. Mikel Irazabal, Elena Lopez-Aguilera and IlkerDemirkol (2019), Active Queue Management as Quality of Service Enabler for 5G Networks, 2019 European Conference on Networks and Communications (EuCNC), pp. 421-426.

[9]. Huy Nguyen Phuong. Thi Mai Thuong Duong, Thu May Duong (2016), An application of Genetic Fuzzy System in Active Queue Management for TCP/IP multiple congestion networks, International Journal of Engineering and Applied Sciences (IJEAS), pp. 47-52

[10]. Rezaee, A.A. &Pasandideh, F (2018), A Fuzzy Congestion Control Protocol Based on Active Queue Management in Wireless Sensor Networks with Medical Applications, Wireless Pers Commun (2018) 98: 815. https://doi.org/10.1007/s11277-017-4896-6

[11]. S. H. Low, F. Paganini, and J. C. Doyle, Internet congestion control, IEEE Control Systems Magazine, Vol.22(1):28-39,2002.

[12]. Sharma, Anup & Kumar Behera, Ashok. (2016). A Survey on Active Queue Management Techniques. International Journal of Engineering and Computer Science. 10.18535/ijecs/v5i11.47.

[13]. Thiruchelvi, G. and J. Raja, A survey on active queue management mechanisms, Int. J. Comput. Sci. Network Secur., Vol 8: 130-145, 2008.

[14]. Vukadinović, Dinko, Mateo Bašić, Cat Ho Nguyen, Nhu Lan Vu, and Tien Duy Nguyen, "Hedge-algebra-based voltage controller for a self-excited induction generator", Control Engineering Practice, pp. 78-90, 2014

Thu May Duong. "An Application of Hedge Algebras in Active Queue Management for TCP/IP Multiple Congestion Networks." IOSR Journal of Electronics and Communication Engineering (IOSR-JECE) 14.6 (2019): 45-53.