



This item was submitted to Loughborough's Institutional Repository (<https://dspace.lboro.ac.uk/>) by the author and is made available under the following Creative Commons Licence conditions.



CC creative commons
COMMONS DEED

Attribution-NonCommercial-NoDerivs 2.5

You are free:

- to copy, distribute, display, and perform the work

Under the following conditions:

BY: **Attribution.** You must attribute the work in the manner specified by the author or licensor.

Noncommercial. You may not use this work for commercial purposes.

No Derivative Works. You may not alter, transform, or build upon this work.

- For any reuse or distribution, you must make clear to others the license terms of this work.
- Any of these conditions can be waived if you get permission from the copyright holder.

Your fair use and other rights are in no way affected by the above.

This is a human-readable summary of the [Legal Code \(the full license\)](#).

[Disclaimer](#) 

For the full text of this licence, please go to:
<http://creativecommons.org/licenses/by-nc-nd/2.5/>

The Mechanism of Adapting RED Parameters to TCP Traffic

Wu CHEN, Shuang-Hua YANG

Department of Computer Science
Loughborough University, Loughborough, LE11 3TU, UK
Emails: {w.chen2, s.h.yang}@lboro.ac.uk

Abstract— **Random Early Detection (RED) can stabilize the queue within a given target range and thus achieve high throughput in the routers. However, the average queue size is quite sensitive to network scenarios and it is difficult to adapt RED parameters to changing network traffic. In this paper we use a previously developed dynamic model of TCP behavior and linear feedback model of TCP/RED to analyze and design a mechanism for RED parameter tuning in response to changing network conditions like traffic load, link capacity and round-trip time. Even though the values of four key RED parameters are determined by varying network conditions, they can be tuned independently without consideration of the interactions among these RED parameters. Simulation results show that this mechanism can stabilize the queue and maintain high link utilization in a wide variety of network conditions.**

Keywords-RED; Congestion Control; Parameter tuning; Control theory

I. INTRODUCTION

Congestion control plays an important role in ensuring good network performance of the Internet. Internet congestion control comprises two main parts: TCP and Active Queue Management (AQM). Among many AQM algorithms proposed over the last decade, RED has been widely implemented in major commercial routers, especially in edge routers [1][2]. It is seldom deployed in core routers because they do not typically run congested links [3]. One of the main goals of RED is to stabilize the queue size within a given target while maintaining high link utilization [4]. The average queue size varies with traffic load as well as round-trip time and link capacity, and consequently parameterizing RED to obtain good performance under variable congestion scenarios can be very difficult. Such difficulties discourage network administrators from activating RED in their routers [5]. Certainly, there are many parameter tuning techniques for RED proposed in the literature [6]-[10], but they were either developed on the basis of empirical investigations and analysis, or are only applicable under certain assumptions. In particular, Adaptive RED (ARED) [6] provided a simple guideline for adjusting only one of the RED parameters in

response to the changing traffic load. So it is applicable only under a narrow range of round-trip times and link capacities. As demonstrated in other AQM or AQM-based approaches, such as the Proportional Integrator (PI) controller [9], Random Exponential Marking (REM) [11] and BLUE [12], parameter setting still remains a critical unsolved problem. In addition, the Web performance of these AQMs is not as good as ARED when dropped packets are used as indications of congestion [13], and their transient response would become much slower when network scenarios change dynamically [14].

In [10] the authors proposed a method to set initial RED parameters from a Control Theoretic perspective. However, they did not illustrate how to adapt these RED parameters to changing network scenarios. By improving their Control Theoretic analysis of TCP/RED systems, we have developed an Auto-Parameterization RED (AP-RED) to provide a simple, scalable and systematic algorithm for parameter tuning as a function of network traffic conditions such as link capacity, round-trip time, and the number of TCP flows. Theoretic analysis and nonlinear simulations using a *ns-2* simulator [15] have demonstrated that it is robust, adaptive to TCP dynamics, and produces desirable transient performance.

The rest of the paper is organized as follows. Section II illustrates a pre-developed nonlinear dynamic model of TCP and a linear feedback model of TCP/RED system. Section III describes AP-RED and presents stability analysis. Section IV introduces the calculation method for network parameters. Simulation results to validate the algorithm are presented in Section V. Finally, we conclude in Section VI.

II. MODEL

In [16], a nonlinear dynamic model of TCP behavior was developed using fluid-flow and stochastic differential equation analysis. By ignoring timeout mechanism, [10] uses a simplified version to describe the model by the following differential equations:

$$\begin{aligned}\dot{W}(t) &= \frac{1}{R(t)} - \frac{W(t)W(t-R(t))}{2R(t-R(t))} p(t-R(t)) \\ \dot{q}(t) &= \frac{W(t)}{R(t)} N(t) - C\end{aligned}\tag{1}$$

where \dot{x} denotes the time-derivative of x and
 $W \doteq$ expected TCP window size (packets);
 $q \doteq$ expected queue size (packets);
 $R \doteq$ round-trip time (seconds);
 $C \doteq$ Link capacity (packets/second);
 $N \doteq$ Number of TCP flows;
 $p \doteq$ probability of packet mark/drop;

According to these equations, [10] further describes the behavior of $\delta W \doteq W - W_0$, $\delta q \doteq q - q_0$, and $\delta p \doteq p - p_0$ in a linear feedback control model of TCP/RED by linearizing variables (W, q, p) at its equilibrium point (W_0, q_0, p_0) . The linear control model is depicted in Fig. 1.

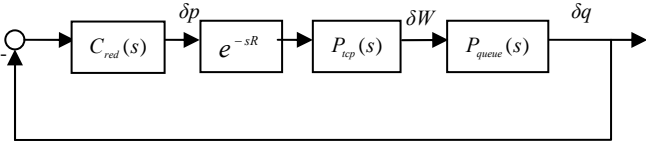


Fig. 1. Feedback control model of TCP/RED system

In the above model $P_{tcp}(s)$ denotes the linearized TCP dynamics, $P_{queue}(s)$ denotes the queue dynamics, e^{-sR} denotes the delay term and $C_{red}(s)$ denotes RED control strategy. They were given by the following equations:

$$\left\{ \begin{array}{l} P_{tcp}(s) = \frac{RC^2}{2N^2} \frac{1}{s + \frac{2N}{R^2C}} \\ P_{queue}(s) = \frac{N}{R} \frac{1}{s + \frac{1}{R}} \\ C_{red}(s) = \frac{L_{red}}{s/K + 1} \end{array} \right. \quad (2)$$

where

$$L_{red} = \frac{p_{max}}{\max_{th} - \min_{th}};$$

$$K = -\frac{\log_e(1 - \alpha)}{T_s};$$

p_{max} = Maximum drop probability;

\max_{th} = Maximum threshold;

\min_{th} = Minimum threshold;

α = Average queue weight;

T_s = Sampling interval.

III. AP-RED ALGORITHM AND STABILITY ANALYSIS

A. The Algorithm

The objective of AP-RED is to maintain the average queue size within the limits of \max_{th} and \min_{th} , and to provide high link utilization in widely varying network scenarios. The rationale behind setting \max_{th} and \min_{th} as a fraction of bandwidth-delay product is to follow the theoretic analysis and rule-of-thumb in accommodating bursty traffic [6][7][17]. The rationale behind setting p_{max} is to achieve a desirable equilibrium point based on theoretic analysis. The rationale behind setting α is to maintain the stability of the system from a Control Theoretic perspective.

Thus, consider the initial RED parameters α_0 , p_{max0} , \max_{th0} , and \min_{th0} that stabilize the queue under the initial network scenario of TCP load N_0 , round-trip time R_0 and link capacity C_0 . When the network scenario varies we present the following algorithm for adjusting RED parameters.

$$\max_{th} = k_r k_c \max_{th0} \quad (3)$$

$$\min_{th} = k_r k_c \min_{th0} \quad (4)$$

$$p_{max} = \left(\frac{k_n}{k_r k_c} \right)^2 p_{max0} \quad (5)$$

$$\alpha = \begin{cases} \frac{k_n}{(k_r k_c)^2} \alpha_0 & N \leq RC/2 \\ \frac{1}{k_r k_c} \alpha_0 & N > RC/2 \end{cases} \quad (6)$$

where

$$k_r = R/R_0;$$

$$k_c = C/C_0;$$

$$k_n = N/N_0$$

and constraint p_{max} within the range [0.01,0.5]

i.e. if $p_{max} > 0.5$ $p_{max} = 0.5$

if $p_{max} < 0.01$ $p_{max} = 0.01$

Compared with the automatic setting of queue weight α based on link capacity in [6], our tuning algorithm of α is based not only on link capacity, but also on round-trip time and traffic load. As α is adapted to congestion scenarios, a larger α can improve transient response while a smaller α provides a sufficient stability margin [10]. When \max_{th} and \min_{th} are reduced according to the changing network parameters, the queuing delay is decreased correspondingly. In contrast a larger \max_{th} and \min_{th} can still stabilize the queue within a given target range when round-trip time increases or link capacity increases.

B. Determining the Equilibrium Point

The purpose of adjusting p_{\max} is to keep the queue at a desirable equilibrium point. At a new equilibrium point where $\dot{W} = 0$ and $\dot{q} = 0$, from (1) we have

$$W^2 p = 2 \text{ and } W = \frac{RC}{N}$$

Then we obtain

$$p = \frac{q - \min_{th}}{\max_{th} - \min_{th}} p_{\max} = 2 \left(\frac{N}{RC} \right)^2 \quad (7)$$

Hence at the equilibrium point the drop probability is determined by network scenario parameters rather than other RED parameters.

Similarly, at the initial equilibrium point q_0 we obtain

$$p_0 = \frac{q_0 - \min_{th0}}{\max_{th0} - \min_{th0}} p_{\max0} = 2 \left(\frac{N_0}{R_0 C_0} \right)^2 \quad (8)$$

Since we tune p_{\max} in terms of (5), from (7) and (8) we have

$$q = k_r k_c q_0$$

Thus the equilibrium point q moves in proportion to the \max_{th} and \min_{th} . It still stays between \max_{th} and \min_{th} when network condition changes.

C. Stability Analysis

1) Stability Proposition

In this section we derive a simplified version of stability proposition based on the analysis given in [10].

Stability Proposition: Let L_{red} and α satisfy:

$$L_{red} \alpha \leq \begin{cases} \frac{0.8N^3}{(RC)^5}, & N \leq RC/2 \\ \frac{0.4N^2}{(RC)^4}, & N > RC/2 \end{cases} \quad (9)$$

where

$$\alpha \ll 1$$

Then the linear feedback control system in Fig. 1 is stable.

Proof:

Consider the frequency response of the compensated loop transfer function

$$\begin{aligned} L(j\omega) &= C_{red}(j\omega)P_{tcp}(j\omega)P_{queue}(j\omega)e^{-j\omega R} \\ &= \frac{L_{red} \frac{(RC)^3}{(2N)^2} e^{-j\omega R}}{\left(\frac{j\omega}{K} + 1\right)\left(\frac{j\omega}{2N} + 1\right)\left(\frac{j\omega}{1} + 1\right)} \end{aligned}$$

$$\approx \frac{L_{red} \frac{(RC)^3}{(2N)^2}}{\frac{j\omega}{K} + 1} e^{-j\omega R} \quad \forall \omega \in [0, \omega_g] \quad (10)$$

where

$$\omega_g = 0.1 \min \left\{ \frac{2N}{R^2 C}, \frac{1}{R} \right\} \quad (11)$$

Thus we obtain

$$\left| L(j\omega_g) \right| \approx \frac{L_{red} \frac{(RC)^3}{(2N)^2}}{\sqrt{\frac{\omega_g^2}{K^2} + 1}} < \frac{L_{red} \frac{(RC)^3}{(2N)^2}}{\frac{\omega_g}{K}} \leq 1 \quad (12)$$

where

$$\frac{L_{red} (RC)^3}{(2N)^2} \leq \frac{\omega_g}{K} \quad (13)$$

We again use (10) to obtain

$$\angle L(j\omega_g) \approx \angle \frac{L_{red} \frac{(RC)^3}{(2N)^2}}{\frac{j\omega_g}{K} + 1} - \omega_g R \geq -90^\circ - 0.1 \frac{180^\circ}{\pi} > -180^\circ$$

This and (12) indicate that the closed-loop system is asymptotically stable according to the Nyquist stability criterion [18].

Given $\alpha \ll 1$, we have

$$\text{Log}_e(1 - \alpha) \approx -\alpha \quad (14)$$

For a stable congested queue we have

$$T_s = \frac{1}{C} \quad (15)$$

Thus, from (1), (14) and (15) we have

$$K = \alpha C$$

Consequently, given any $N \leq RC/2$, from (13) we obtain

$$\frac{L_{red} (RC)^3}{(2N)^2} \leq \frac{0.2N}{\alpha C}$$

i.e.,

$$L_{red} \alpha \leq \frac{0.8N^3}{(RC)^5}$$

Similarly, given any $N > RC/2$, we obtain

$$L_{red} \alpha \leq \frac{0.4N^2}{(RC)^4}$$

This completes the proof.

2) Stability analysis

For initial RED parameters α_0 , $p_{\max0}$, \max_{th0} , \min_{th0} that satisfy (9), given $N_0 > R_0 C_0 / 2$, we have

$$\frac{p_{\max0}}{\max_{th0} - \min_{th0}} \alpha_0 < \frac{0.4N_0^2}{(R_0 C_0)^4} < \frac{0.8N_0^3}{(R_0 C_0)^5}$$

and given $N_0 \leq R_0 C_0 / 2$, we still have

$$\frac{p_{\max 0}}{\max_{th0} - \min_{th0}} \alpha_0 < \frac{0.8N_0^3}{(R_0 C_0)^5}$$

Given $N \leq RC/2$, from (3)-(6) we have

$$\begin{aligned} L_{red} \alpha &= \frac{p_{\max}}{\max_{th} - \min_{th}} \alpha = \frac{k_n^3}{(k_r k_c)^5} \frac{p_{\max 0}}{\max_{th0} - \min_{th0}} \alpha_0 \\ &< \frac{k_n^3}{(k_r k_c)^5} \frac{0.8N_0^3}{(R_0 C_0)^5} = \frac{0.8N^3}{(RC)^5} \end{aligned}$$

Similarly, Given $N > RC/2$ we have

$$L_{red} \alpha < \frac{0.4N^2}{(RC)^4}$$

So the stability proposition in (9) can always be satisfied when we tune RED parameters according to equations (3)-(6). This provides a useful guideline for tuning RED parameters whilst maintaining the stability.

The derivation of our stability proposition is similar to that proposed in [10]. So we have an intuitive sense that our algorithm can provide a similar performance to that in [10]. However, the RED parameters in [10] are tightly coupled to each other and a complex calculation is needed to obtain suitable RED parameters for every changed network scenario. In contrast RED parameter tuning becomes much easier with the algorithm as shown in (3)-(6).

IV. NETWORK TRAFFIC MEASUREMENT

The online statistic characteristics of heterogeneous network scenarios used in AP-RED include round-trip time R , number of TCP flows N and link capacity C . Based on packet-level traces and flow-level statistics we know they are measurable and do not change dramatically over a relatively short time. For example, the characteristics of R , N , C and average package size etc have been observed in OC3MON [19] and the IPMON system [20]. It is feasible for us to use the methods described in the literature to obtain the network parameters.

In [20] the number of flows is calculated per minute. Those packets with the same 5-tuple information (source address, source port, destination address, destination port, and protocol) are classified as the same flow. The start time of a flow is the first time when a packet with new 5-tuple information is observed, and the flow ends when no packets with the same 5-tuple information are seen for a time interval 60s. On the other hand we can use a shorter time interval to calculate the traffic loads. Although there is some difference for the calculated value of traffic loads with different time intervals, in terms of [21] the variation of proportionality factor k_n is small with different time intervals.

In [20][22] round-trip time is measured as the time elapsed between a SYN packet and the first ACK packet that completes the three-way handshake. This is a rough estimate because we only compute the round-trip times for flows of which we observe the SYN/ACK pair. Fortunately, the deviation of the estimate is minimized because we use a proportionality factor

k_r . In addition, RED also has its own robustness under changed scenarios [23].

In [24] link capacity can be sampled by calculating the ratio of departed TCP packets to the router's busy time. Then this capacity is smoothed by a low pass filter.

The measurement methods can be deployed in routers. For example, Cisco NetFlow is configured on most Cisco routers to provide a highly condensed and detailed view of all real time network traffic [25]. Because the statistic characteristics of network parameters are not changing dramatically we do not need to predict the network scenario parameters R , N and C in advance and could measure these parameters less frequently. In practice we need to set a suitable interval time to obtain these network parameters on a trial-and-error basis. Because it is easy to obtain these network parameters in edge routers, the feasibility of our RED tuning algorithm is ensured, especially in edge routers.

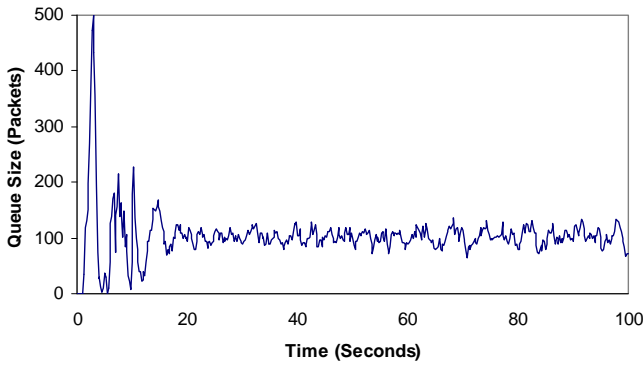
V. SIMULATIONS

Although AP-RED is based on the analysis of a linearized model, we verify the algorithm by using a *ns-2* simulator to capture the stochastic, nonlinear nature of the network dynamic. As the prominent algorithm for RED parameter tuning [13], ARED is compared with AP-RED in the simulations. These simulations use a simple dumbbell topology with a single bottleneck link shared by persistent FTP flows, short HTTP sessions and reverse-path traffic. Each HTTP sessions repeatedly make short file transfers. Between two consecutive transfers, there is a 'think time' that starts after the last byte of the first file has been acknowledged. The transfer size is exponentially distributed with a mean of twelve 1 KB-packets. The think time is exponentially distributed with a mean of 500ms. The presence of these short-lived flows introduces noise into the queue. The number of HTTP flows in our simulations is set to twice that of FTP flows. The presence of reverse-path traffic introduces ack (acknowledgement) compression and the loss of ack packets, and thus increases the burstiness of the forward-path traffic. In all simulations the reverse-path traffic consists of 20 persistent FTP flows. The average packet size is set to 500 bytes, and the buffer size is set to 500 packets.

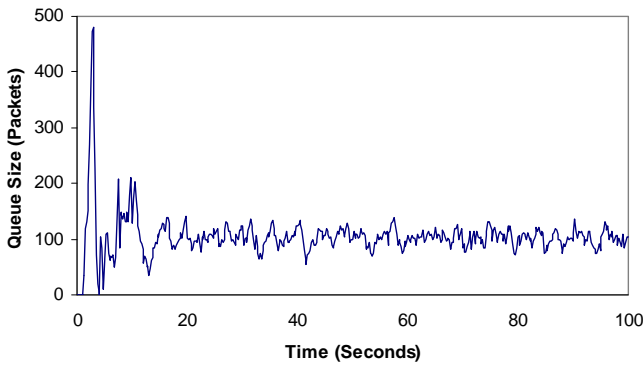
A. Experiment 1

In the first experiment we have $N_0 = 50$ FTP flows with round-trip propagation delay $T_{p0} = 120$ ms. The link capacity is set to 10 Mbits/second, that is, $C_0 = 2500$ packets/second. The corresponding parameters of AP-RED are $p_{\max 0} = 0.05$, $\min_{th0} = 50$ packets, $\max_{th0} = 150$ packets and $\alpha_0 = 10^{-4}$, which satisfy the stability proposition (9). We choose \min_{th} , \max_{th} and α for ARED the same as the corresponding parameters for AP-RED above, and keep these ARED parameters unchanged in the following experiments. Fig. 2(a) plots the instantaneous queue size of ARED. Observe a stable queue with small fluctuations around the equilibrium point of 100 packets. Fig. 2(b) shows that the instantaneous queue size of AP-RED is also stable. Since both ARED and AP-RED are

essentially RED, we observe quite similar performance of their queues in Fig. 2.



(a) ARED



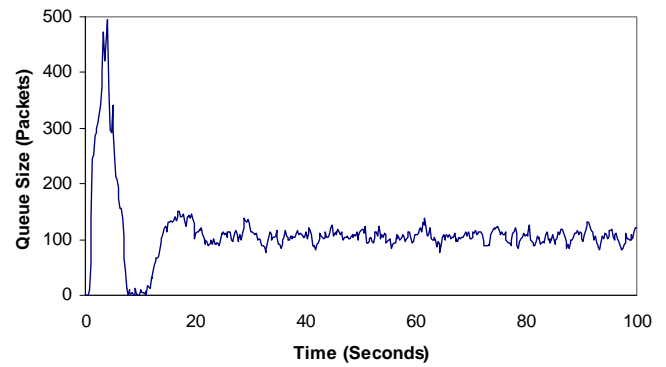
(b) AP-RED

Fig. 2. Comparison of ARED and AP-RED under network scenarios: $N= 50$, $C= 2500$ packets/second and $T_p = 120$ ms.

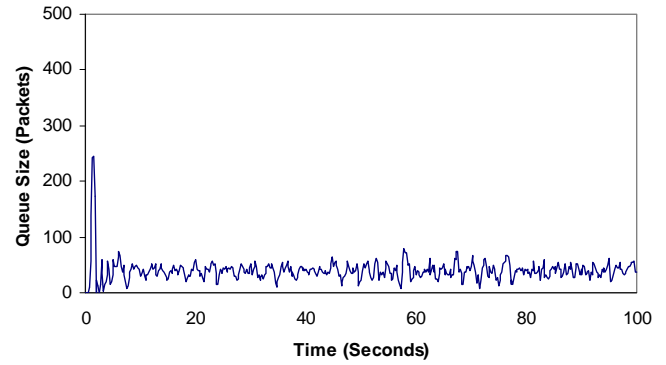
B. Experiment 2

In the second experiment the round-trip propagation delay decreases to $T_p=100$ ms and the link capacity decreases to $C=1250$ packets/second. The number of FTP flows decreases to $N=30$. The plot in Fig. 3(a) shows that the queue oscillations of ARED become smaller than that in Fig. 2(a). This is because RED becomes more stable when round-trip time or link capacity decreases [23]. When we use T_p / T_{p0} to approximate k_r in AP-RED algorithm, \min_{th} / C and \max_{th} / C are in proportion to T_p . So are the queuing delay and the round-trip time. With the same network scenario, AP-RED parameters are changed to $p_{max} = 0.1$, $\min_{th} = 20$ packets, $\max_{th} = 60$ packets and $\alpha = 3.5 * 10^{-4}$, which are derived from the tuning algorithm (3)-(6). We can see that \min_{th} and \max_{th} decrease when R and C decrease. Fig. 3(b) shows that the queue size of AP-RED reduces to around 40 packets, compared to 100 packets in Fig. 3(a), whilst the queue can be stabilized at the new equilibrium point between the updated \min_{th} and \max_{th} . Consequently, a lower queuing delay is obtained. At the same time, the large queue weight used in AP-RED should improve the transient response. As we observe, the queue settles to

around its equilibrium point after 5 seconds, compared to 20 seconds in ARED.



(a) ARED

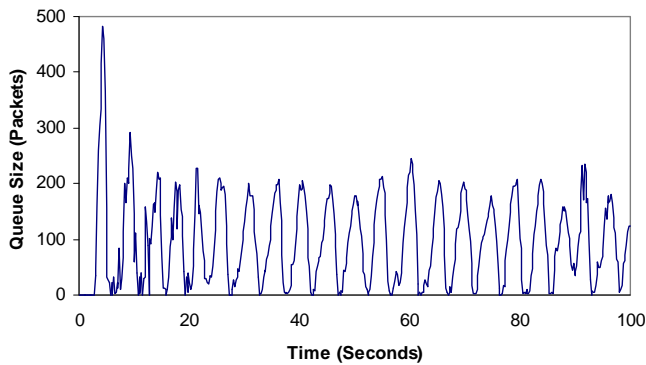


(b) AP-RED

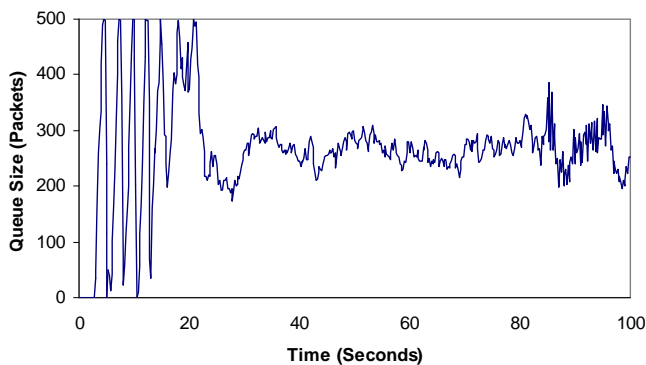
Fig. 3. Comparison of ARED and AP-RED under network scenarios: $N= 30$, $C= 1250$ packets/second and $T_p = 100$ ms.

C. Experiment 3

In the third experiment the round-trip propagation delay increases to $T_p=250$ ms and the link capacity increases to $C=3000$ packets/second. The number of FTP flows increases to $N=100$. The increase in the round-trip propagation delay and the link capacity should reduce the stability of the system or even make the system unstable [23]. The plot in Fig. 4(a) displays deterministic oscillations of ARED, which lead to large delay jitter and low link utilization. With the same network scenario, AP-RED parameters are adjusted to $p_{max} = 0.032$, $\min_{th} = 125$ packets, $\max_{th} = 375$ packets and $\alpha = 3.2 * 10^{-5}$. These parameters are derived from the tuning algorithm (3)-(6) and still satisfy the stability proposition (9). We can see that \min_{th} and \max_{th} increase when R and C increase. Fig. 4(b) displays that the queue of AP-RED can be stabilized around a new equilibrium point between the updated \min_{th} and \max_{th} . Although the queue size becomes large, it remains greater than zero and smaller than the total buffer size as desired. Thus high link utilization is maintained.



(a) ARED



(b) AP-RED

Fig. 4. Comparison of ARED and AP-RED under network scenarios: $N=100$, $C=3000$ packets/second and $T_p=250$ ms.

VI. CONCLUSIONS

In this paper we developed AP-RED to provide a scalable and systematic mechanism for RED parameter tuning. The advantages of AP-RED are that it provides a simple mechanism, from a theoretic standpoint, for constantly tuning four key RED parameters instead of tuning only one of them, like p_{max} , under varying network conditions, such as traffic load, round-trip time and link capacity. It is unnecessary to consider the interaction among these RED parameters when they are changed in response to changed network conditions. Stability analysis and nonlinear simulations showed that AP-RED could stabilize the queue and maintain high link utilization in dynamic network scenarios. Multi-service situations involving networks with multiple bottlenecks and connections have not been considered in this study. More research is required in order to extend AP-RED to the multi-service. We took a single bottleneck link in a dumbbell topology in our simulations in order to best compare the proposed algorithm with ARED. For a more realistic topology, further work is required.

REFERENCES

[1] "Distributed Weighted Random Early Detection," [Online] Available: http://www.cisco.com/en/US/docs/ios/11_1/feature/guide/WRED.html.

[2] M. Dischinger, A. Haebleren, K. P. Gummadi, and S. Saroiu, "Characterizing Residential Broadband Networks," in Proceedings of Internet Measurement Conference, San Diego, USA, October 2007.

[3] S. Leinen, "Use of RED in practice?" [Online] Available: <http://mailman.postel.org/pipermail/end2end-interest/2006-March/005851.html>, March 2006.

[4] S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance," IEEE/ACM Transactions on Networking, vol. 1, No. 4, pp. 397-413, August 1993.

[5] M. May, J. Bolot, C. Diot and B. Lyles, "Reasons not to deploy RED," in Proceedings of 7th International Workshop on Quality of Service, London, UK, pp. 260-262, 31 May-4 June 1999.

[6] S. Floyd, R. Gummadi and S. Shenker, "Adaptive RED: an algorithm for increasing the robustness of RED's active queue management," [Online] Available: <http://www.icir.org/floyd/papers/adaptiveRed.pdf>, August 2001.

[7] H. Sirisena, A. Haider and K. Pawlikowski, "Auto-Tuning RED for Accurate Queue Control," in Proceedings of IEEE GLOBECOM, Taipei, Taiwan, November 2002.

[8] T. Ziegler, S. Fdida, and C. Brandauer, "Stability Criteria for RED with Bulk-data TCP Traffic," 2001. Technical Report, August 1999. <http://www-rp.lip6.fr/publications/production.html>.

[9] C. Hollot, V. Misra, D. Towlsey, and W. Gong, "Analysis and design of controllers for AQM routers supporting TCP Flows," IEEE Transactions on Automatic Control, vol. 47, No. 6, pp 945-959, June 2002.

[10] C. Hollot, V. Misra, D. Towlsey, and W. Gong, "A Control Theoretic Analysis of RED," in Proceedings of IEEE INFOCOM, Anchorage, USA, April 2001.

[11] S. Athuraliya, V. H. Li, S. H. Low, and Q. Yin, "REM: Active Queue Management," IEEE Network, vol. 15, No. 3, pp 48-53, May 2001.

[12] W. C. Feng, Kang G. Shin, Dilip D. Kandlur, Debanjan Saha "BLUE Active Queue Management Algorithms," IEEE/ACM Transactions on Networking, vol. 10, No. 4, pp. 513-528, August 2002.

[13] L. Le, J. Aikat, K. Jeffay, and F. D. Smith, "The Effects of Active Queue Management and Explicit Congestion Notification on Web Performance," IEEE/ACM Transactions on Networking, vol. 15, No. 6, pp. 1217-1230, December 2007.

[14] C. Wang, B. Li, Y. T. Hou, K. Sohaby and Y. Lin, "LRED: A Robust Active Queue Management Scheme Based on Packet Loss Ratio," in Proceedings of IEEE INFOCOM, Hong Kong, China, March 2004.

[15] "The Network Simulator - ns-2," [Online] Available: www.isi.edu/nsnam/ns/.

[16] Vishal Misra, Wei-Bo Gong, and Don Towsley, "Fluid-based Analysis of a Network of AQM Routers Supporting TCP Flows with an Application to RED," in Proceedings of IEEE ACM/SIGCOMM, Stockholm, Sweden, 28 August-1 September 2000.

[17] S. Floyd, "Discussions of Setting Parameters," [Online] Available: <http://www.icir.org/floyd/REDparameters.txt>, November 1997.

[18] G. F. Franklin, J. D. Powell and A. Emami-Naeini, "Feedback Control of Dynamic Systems," Prentice Hall, 2002.

[19] K. Thompson, G. Miller, and R. Wilder, "Wide Area Internet Traffic Patterns and Characteristics," IEEE Network, vol. 11, No. 6, pp. 10-23, November-December 1997.

[20] C. Fraleigh, S. Moon, B. Lyles, C. Cotton, M. Khan, D. Moll, R. Rockell, T. Seely, and C. Diot, "Packet-Level Traffic Measurements from the Sprint IP Backbone," IEEE Network, vol. 17, No. 6, pp. 6-16, November-December 2003.

[21] G. Iannaccone, C. Diot, I. Graham, N. Mckeown, "Monitoring Very High Speed Links," in Proceedings of IEEE ACM/SIGCOMM, San Diego, USA, August 2001.

[22] H. S. Martin, A. McGregor, and J. G. Cleary, "Analysis of Internet Delay Times," in Proceedings of Passive and Active Measurement Workshop, Auckland, NewZealand, April 2000.

[23] S. H. Low, F. Paganini, J. Wang, and J. C. Doyle, "Linear stability of TCP/RED and a scalable control," Computer Networks, vol. 43, No.5, pp. 633-647, December 2003.

- [24] H.G. Zhang, C.V. Hollot, D. Towsley, and V. Misra, "A Self-Tuning Structure for Adaptation in TCP/AQM Networks," in Proceedings of IEEE/Globecom, San Francisco, USA, December 2003.
- [25] "Netflow White Papers," [Online] Available: http://www.cisco.com/en/US/products/ps6601/prod_white_papers_list.html.