Correlation Based Optimization of Random Early Detection

Rahul Vaidya and Shalabh Bhatnagar

Abstract— **The performance of Random Early Detection (RED) algorithm is known to be highly sensitive to network conditions and RED parameter settings. Under non-optimal settings the performance of RED tends to that of traditional Tail-Drop gateways. A black box parameter optimization scheme, with direct measurement of loss function, is proposed. We present a Correlation Based (CB) Optimization algorithm, suitable for optimizing the RED parameters online. Network simulations are carried out to validate our approach. Simulation experiments show significant improvement in the performance of routers with negligible computational overhead.**

I. Introduction

Various AQM techniques like REM [1], RED [2] have been suggested to eliminate packet loss, increase router utilization and decrease queuing delays. Of these, RED has been widely implemented in the internet. However, it has been seen that the network parameter settings affect the performance of the RED gateway [3], [4], [5], [6]. Also, the optimum parameter settings are dependent on the network conditions. Since the network conditions change dynamically, the RED parameters need to be tuned on-line for optimal performance.Certain schemes for adaptive tuning of RED based on simplified models have been proposed [4], [7], [3]. Variants of RED, which heuristically manipulate a certain RED parameter have also been proposed. Adaptive RED [3] adapts the maximum dropping probability of packets so that the average queue length is within the target range. In [6], a heuristic algorithm based on random recursive sampling is proposed for optimizing RED parameters. The optimization step there is carried out in 'control plane' (and not on the router) where the loss function is estimated using network simulation. The optimum parameters are then fed back into the router after the algorithm has converged. This requires the knowledge of the whole network topology as well as the source behaviors. This is, however, infeasible in today's internet.

In this paper we discuss a direct measurement approach for optimization of RED. Here we consider RED as a black box and the loss function is measured directly over the router. Any suitable gradient based optimization algorithm can then be employed for obtaining the optimal network setting. In this approach we do not resort to simulation of the network. Also we do not make any simplifying assumptions regarding the network topology and charecteristics.

Department of Computer Science and Automation, IISc, Bangalore. E-mail: {rahulv,shalabh}@csa.iisc.ernet.in. This work was supported in part by Grant Number SR/S3/EE/43/2002-SERC-Engg. from Department of Science and Technology, Government of India.

0-7803-8909-3/04/$20.00 ©2004 IEEE

However, since the optimization is to be carried using direct measurements of loss function over the router, the optimization algorithm used needs to be efficient, have smooth convergence and be computationally inexpensive. In this paper, we propose a Correlation Based optimization algorithm that uses SPSA [8] type perturbations. Simulation results show that RED under CB Optimization performs better than both the traditional RED as well as the algorithm in [3].

In § II we discuss traditional RED and the algorithms in [3] and [6]. We then present the CB Optimization algorithm in § III and § IV. The implementation and other details of CB Optimized RED algorithm are explained in § V. Simulation experiments for validating our approach are given in § VI. Finally § VII provide the concluding remarks.

II. Traditional RED and Adaptive tuning of RED

In RED gateways, the decision of dropping the arriving packet is made on the value of the average queue length at the time of arrival of the packet. The average queue length($\bar{q}$) is calculated using the equation $\bar{q} = (1 - w_q)\bar{q} + w_q q$ where q is the instantaneous queue length and w_q is a positive constant less that 1. The arriving packet is always dropped if $\bar{q} > max_{th}$ and is never dropped if $\bar{q} < min_{th}$. When $\bar{q}$ is between min_{th} and max_{th}, the probability of dropping the arriving packet varies linearly with the average queue length from zero (at min_{th}) to max_p (at max_{th}). The idea behind the algorithm is that at equilibrium the average queue length stabilizes at a value at which the rate of decrease in source rates (due to packet drops) balance the increase in the source rates due to successful delivery of packets.

There are four parameters in RED which affect the performance of the gateway. The 'low pass filter' parameter, w_q, determines the amount of transient congestion to be allowed before any action is taken. The parameter max_p determines the probability with which the packets are dropped at any given value of average queue length and hence determines the equilibrium average queue length. Since the equilibrium average queue length is also a function of the network load, for any particular value of max_p, the network load determines the equilibrium average queue length. Also the parameters max_{th} and min_{th} determine the average queuing delay experienced by the connection and also router utilization.

Adaptive RED [3] adaptively tunes max_p so that the average queue length is always within a target range. This algorithm uses the intuition that increasing max_p reduces the equilibrium average queue length. It is assumed that effect

of the other parameters is negligible compared to that of max_p. In [6], a heuristic algorithm based on random recursive sampling is used to optimize the RED gateway. The performance of RED gateway is monitored and the adaptive tuning algorithm is triggered whenever the performance is below a target. The loss function for each parameter update is measured using simulation that in turn requires complete knowledge of the topology of the network as well as the individual source behaviors. In our setting, we propose to run our algorithm directly over the router. Thus, network simulation is not required since router performance is directly measured. Also there is no significant increase in computational complexity.

III. BACKGROUND

In this section we first discuss the Correlation Based (CB) scheme and then apply it to the Two-Timescale framework. The CB scheme is based on the observation that $(f(x) - f(y))(x_i - y_i)$ and $(f(x) - f(y))/(x_i - y_i)$ have the same sign [9]. One may consider the algorithm

$$\theta(n+1) = \theta(n) - a(n)\frac{h(\theta(n+1)) - h(\theta(n))}{\delta(\theta(n+1) - \theta(n))}, \quad (1)$$

where, $h(\theta)$ is the loss function and $\theta(n)$ is the parameter update at the nth iteration. This however is numerically ill behaved, the reason being that $\theta(\cdot)$ changes very little over a single iterate. Thus for better behavior of the algorithm, the parameters are simultaneously perturbed [8] at each iteration. Thus the algorithm is given by

$$\theta(n+1) = \theta(n) - a(n)\frac{(h(x(n+1)) - h(x(n)))}{(x(n+1) - x(n))} \quad (2)$$

where $x(n+1)$ is $(\theta(n+1)+\Delta(n+1)\delta)$ and $x(n)$ is $(\theta(n)+\Delta(n)\delta))$. Also, $\delta > 0$ is a small scalar and Δ_i, $i = 1, \ldots, N$ are independent and identically distributed (i.i.d.), bernoulli distributed random variables with $\Delta_i = \pm 1$ w.p. 1/2. Also Δ is the vector $\Delta = (\Delta_1, \Delta_2, \ldots, \Delta_N)^T$.

Now we will discuss the CB scheme in the Two-Timescale framework where the cost function itself the long run average over certain function observations. Suppose for $n \geq 0$, $\zeta_n \in \mathcal{R}^p$ are i.i.d., random variables. Let $h : \mathcal{R}^N \times \mathcal{R}^p \rightarrow \mathcal{R}$ be an associated cost function that is assumed bounded and measurable. Let $J : \mathcal{R}^N \rightarrow \mathcal{R}$ be defined as:

$$J(\theta) = \text{li}$$

the $sgn(\cdot)$ ensures that the impact of sudden change in network conditions during optimization is minimized.

The CB Optimized RED algorithm is triggered when the performance degrades due to change in network conditions. The data averaging step is carried out after every arrival. After every 'L' packet arrivals parameter components are updated directly over the router. The above procedure is continued till the performance has sufficiently improved. The algorithm is as given in Algorithm 1.

CB Optimized RED is disabled for most of the running time of the router and is triggered only when change in network condition result in bad router performance. Hence the computational overhead of the algorithm is negligible. Here we assume that the running time of the algorithm is negligible compared to the time over which the network conditions are stable. Also note that the objective function of the algorithm is determined depending on the priorities of the network administrator and can be set dynamically.

VI. Simulation Results

Extensive simulation experiments were carried out to validate the effectiveness of our approach. We show in this section, results of simulation experiments for different network conditions. All the experiments show a significant performance improvement over both the traditional RED as well as the algorithm in [3]. The CB Optimized RED was implemented in the NS2 network simulator [12], and experiments were carried out. The topology for the experiments is given in Figure. 1.

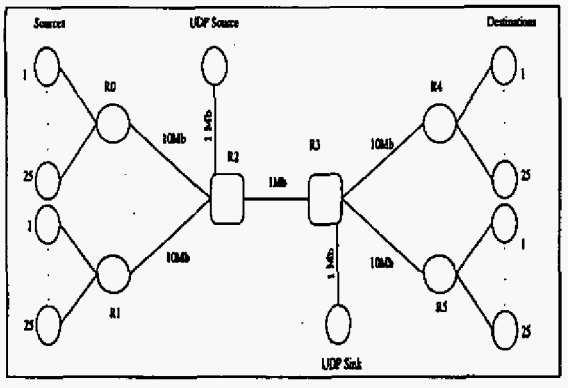
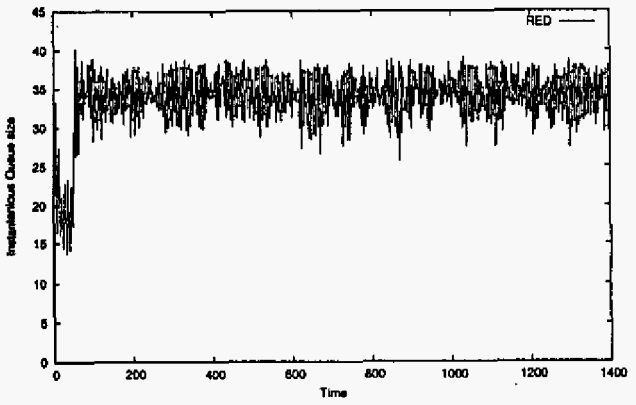


Fig. 1. Network Topology for Simulation

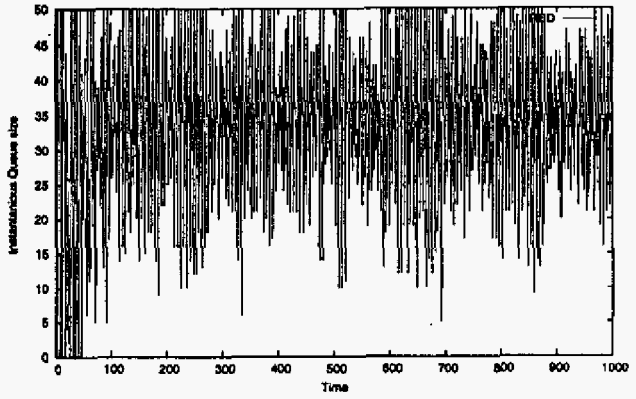
FTP sources with infinite data are considered to be operating over TCP. These are equally distributed over routers R0 and R1. A non-cooperative UDP source is connected to router R2. It is an on-off source occupying 10% of the bottleneck link. The destinations are again equally distributed over routers R4 and R5. The link connecting routers R2 and R3 is the bottleneck link in the network. RED is configured over R2 to manage a buffer of 100 packets. The RED parameters are initialized to the recommended settings, i.e. max_{th} = 30, min_{th} = 10, w_q = 0.001 and max_p = 0.1 respectively. For the step size sequences, we chose $a(n) = a(0)/n$ and $b(n) = 1/n^{0.66}$, respectively. The value of $a(0)$ was chosen to be 0.005 for w_q, 0.1 for max_p and 1 for max_{th} and min_{th}, respectively. The values of the parameter δ for various parameter components were taken to be corresponding values of $a(0)$ for these. This was done for better convergence behavior. A different value of $a(0)$ for a given parameter component merely has the effect of changing the speed of convergence of that component to the optimum value.

For better performance of the network, the mean queue length of packets should be independent of load on the network. Also the variation of queue size should be minimum so that the connections do not experience delay jitters and also the connections have predictable queuing delays. In our experiments, the objective of CB Optimized RED was to minimize the variation of the instantaneous queue length from a predetermined value (20 in this case).

First we consider a network scenario where there is sudden increase in the network load. In our simulation we increase the number of sources transmitting over the router from 10 to 50 at the 50th second since the start of the simulation. The performance of traditional RED, ARED and CB Optimized RED algorithms is as shown in Figs. 2(a), 3(a) and 4(a) respectively.



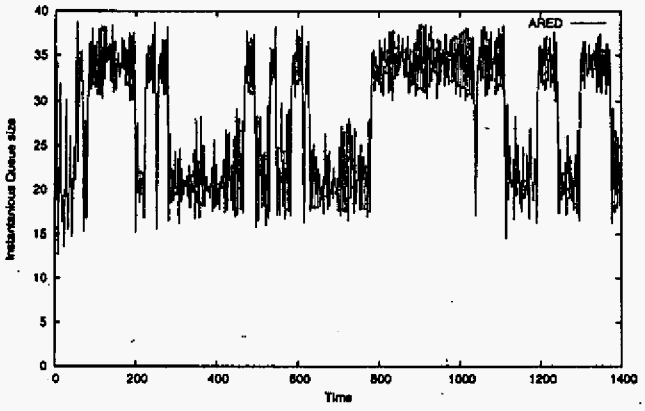
(a) Experiment 1: Instantaneous Increase in Load



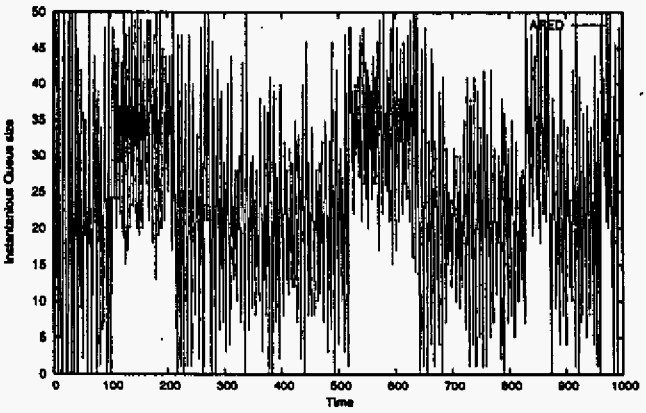
(b) Experiment 2: Increase in Load over an interval

Fig. 2. Performance of Traditional RED

Next we consider a scenario where there is a gradual

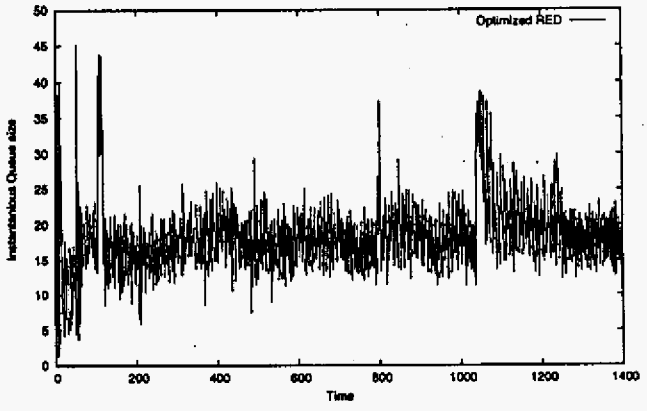


(a) Experiment 1: Instantaneous Increase in Load

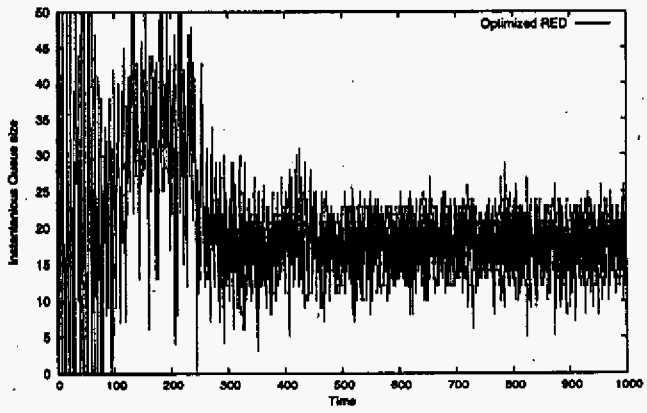


(b) Experiment 2: Increase in Load over an interval

Fig. 3. Performance of ARED



(a) Experiment 1: Instantaneous Increase in Load



(b) Experiment 2: Increase in Load over an interval

Fig. 4. Performance of CB Optimized RED

increase in the network load. Here the number of connections over the router was increased from 10 to 50 over a 50 second interval. The performance with traditional RED, ARED and CB Optimized RED algorithms is shown in Figs. 2(b), 3(b) and 4(b) respectively. The difference in both the experiments is noticeable, with CB Optimized RED significantly decreasing the mean as well as the variance of the instantaneous queue size. From Fig. 4 it is clear that although CB Optimized RED shows slightly large oscillations initially (even though these are lower than both traditional RED and ARED), these significantly reduce with time. Moreover CB Optimized RED reduces the average queue size close to the predetermined queue size (20 in this case).

The numerical comparisons of performance are shown in Table I where the mean and variance of average queue length are computed over the time windows shown in the two figures. It is clearly seen from the table that mean and variance of queue length in the case of CB Optimized RED are significantly lower than corresponding values of these for both traditional RED and ARED respectively. Also, the average queue size in CB Optimized RED is close to the predetermined queue size (20 in this case).

Note that we did not make any assumptions about the network conditions or about the protocol used. Hence the above approach will be equally effective for any other measure of router performance.

VII. Conclusions

Due to the dynamic nature of the internet, the RED parameter settings need to be dynamically tuned rather than setting *a priori*, as was proposed previously. In this paper we suggest a CB scheme for this purpose. This algorithm uses two different timescales and updates all parameter components simultaneously once every fixed number of packet arrivals. The algorithm uses a convolution of the $sgn(\cdot)$ function as gradient

estimate in the update step resulting in less oscillations and smoother convergence.

The simulation experiments show that our algorithm showed significant improvement over both the traditional RED as well as the algorithm in [3]. The objective was achieved in short time interval after our algorithm was triggered. We proposed the implementation of our algorithm directly over RED routers, the advantage being that the measure of loss functions are accurate as well as obtained quickly for different parameter settings as compared to the other schemes where the network needs to simulated for each parameter update.

The proposed algorithm is also adaptable to the changing needs of the network. The objective function can be redefined whenever required to obtain a different corresponding set of optimum parameter values. Moreover, there are no restrictions on the form of objective function as long as it is a measure of router performance. Also the proposed algorithm can be executed over a separate 'control plane', a dedicated computer for optimization of the network. The routers need only transmit one bit per parameter iteration step because of the use of the $sgn(\cdot)$ function in the algorithm. Thus, the network overhead due to our algorithm is also negligible.

Since no assumptions have been made about the network as well as the protocol, the same approach can be applied to various AQM mechanisms, and different protocols as well. For instance, this approach can also be extended to be applied to the DiffServe architecture. Additional parameters can also be handled with minimal extra overheads. The performance of proposed approach when different objective functions are defined for different classes of traffic, however needs to be investigated.

Algorithm	Inst. Load		Load over interval	
	Mean	Variance	Mean	Variance
Traditional RED	33	128	33	140
ARED	27	190	25	165
CB Opt. RED	19	47	20	89

TABLE I

PERFORMANCE COMPARISON OF RED, ARED AND CB OPTIMIZED RED

Algorithm 1 CB Optimized RED

for each packet arrival **do**
  Calculate the average queue size
  if $min_{th} \leq average \leq max_{th}$ **then**
    calculate probability p_a
    With probability p_a mark the packet
  else
    if $average \geq max_{th}$ **then**
      mark the arriving packet
    end if
  end if
  if Optimization Triggered **then**
    $m = m + 1$
    $x(n+1) := (\theta(n) + \delta\Delta(n))$
    $Z(nL + m + 1) = Z(nL + m) + b(n)\,(average - Z(nL+m))$
  end if
end for
for every (L)th packet arrival **do**
  $\theta_i(n+1) = \theta_i(n) - a(n) sgn\left[\frac{Z((n+1)L) - Z(nL)}{x(n+1) - x(n)}\right]$
  if $\theta_i(n+1) < a_{i,min}$ **then**
    $\theta_i(n+1) = a_{i,min}$
  else
    if $\theta_i(n+1) > a_{i,max}$ **then**
      $\theta_i(n+1) = a_{i,max}$
    end if
  end if
  $m := 0$
  $n := n + 1$
  Generate new $\Delta(n) = (\Delta_1(n), \cdots, \Delta(n))^T$ using normalized Hadamard matrix.
end for
Variables:
average: average queue size
$Z(l)$: is given by Equation 7
$\theta_i(n)$: $\theta_i(n)$ as given by Equation 6

REFERENCES

[1] S. Athuraliya, S. Low, V. Li, and Q. Yin, "REM: Active queue management," *IEEE Network*, vol. 15, no. 3, pp. 48–53, 2001.

[2] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, pp. 397–413, 1993.

[3] S. Floyd, R. Gummadi, and S. Shenker, "Adaptive RED: An algorithm for increasing the robustness of RED," Technical report, The ICSI Center for Internet Research Berkeley, California, 2001.

[4] W. Feng, D. Kandlur, D. Saha, and K. G. Shin, "A self-configuring RED gateway," in *Proceedings of INFOCOM*, 1999, vol. 3, pp. 1320–1328.

[5] M. May, J. Bolot, C. Diot, and B. Lyles, "Reasons not to deploy RED," in *Proc. of 7th. International Workshop on Quality of Service (IWQoS'99), London*, June 1999, pp. 260–262.

[6] T. Ye and S. Kalyanaraman, "Adaptive tuning of RED using on-line simulation," *GLOBECOM*, vol. 3, pp. 2210–2214, 2002.

[7] T. J. Ott, T. V. Lakshman, and L. H. Wong, "SRED: Stabilized RED," in *Proceedings of INFOCOM*, 1999, vol. 3, pp. 1346–1355.

[8] J. C. Spall, "Multivariate stochastic approximation using a simultaneous perturbation gradient approximation," *IEEE Transactions on Automatic Control*, vol. 37, pp. 332–341, 1992.

[9] B. Bharath and V. S. Borkar, "Robust parameter optimization of hidden Markov models," *Journal of Indian Institute of Science*, vol. 78, pp. 119–130, 1998.

[10] S. Bhatnagar, M. C. Fu, S. I. Marcus, and S. Bhatnagar, "Two timescale algorithms for simulation optimzation of hidden Markov models," *IIE Transactions*, vol. 33, no. 3, pp. 245–258, 2001.

[11] S. Bhatnagar, M. C. Fu, S. I. Marcus, and I. J. Wang, "Two timescale simultaneous perturbation stochastic approximation using deterministic perturbation sequences," *ACM Transactions on Modelling and Computer Simulation*, vol. 13, no. 4, pp. 180–209, 2003.

[12] NS, "Network Simulator," *http://www-mash.cs.berkley.edu/ns*.