7 August 2002
Electronics Letters Online No: 20020862
DOI: 10.1049/el:20020862

M. De Munnynck, A. Lootens, S. Wittevrongel and H. Bruneel (*Stochastic Modeling and Analysis of Communication Systems (SMACS) Research Group, Department of Telecommunications and Information Processing, Ghent University, Sint-Pietersnieuwstraat 41, B-9000, Gent, Belgium*)

References

1 BRUNEEL, H., and MOENECLAEY, M.: 'On the throughput performance of some continuous ARQ strategies with repeated transmissions', *IEEE Trans. Commun.*, 1986, **34**, (3), pp. 244–249
2 BIRRELL, N.D.: 'Pre-emptive retransmission for communication over noisy channels', *IEE Proc. F*, 1981, **128**, (6), pp. 393–400
3 MOENECLAEY, M., and BRUNEEL, H.: 'Efficient ARQ scheme for high error rate channels', *Electron. Lett.*, 1984, **20**, (23), pp. 986–987
4 BRUNEEL, H., and KIM, B.G.: 'Discrete-time models for communication systems including ATM' (Kluwer Academic Publishers, Boston, MA, USA, 1993)
5 DE MUNNYNCK, M., LOOTENS, A., WITTEVRONGEL, S., and BRUNEEL, H.: 'Transmitter buffer behaviour of stop-and-wait ARQ schemes with repeated transmissions', *IEE Proc. Commun.*, 2002, **149**, (1), pp. 13–17

Scalability problems of RED

Changhee Joo and Saewoong Bahk

Random early detection (RED) can be helpful for Transport Control Protocol to control congestion by dropping packets before its queue becomes full. However, falsely configured RED causes network oscillation and simply works as a tail drop under the heavy load. To overcome the scalability problems of RED, an algorithm is proposed that adjusts the steepness of the drop probability function properly under various network conditions.

Introduction: Transport Control Protocol (TCP) congestion control has been useful to control packet loss and prevent congestion collapse. However, the effectiveness is limited since TCP does not know the exact network status. Therefore an active queue management scheme-in routers is helpful since it can inform TCP of the network status, the most popular being random early detection (RED) [1] that avoids global synchronisation and a bias against bursty traffic by dropping packets before its queue gets full. Despite its prominent advantages, RED is not a substitute for a traditional drop-tail queue owing to difficulty in configuring its parameters under dynamic network environments.

Scalability problem of RED: RED has a linear drop probability function, $f(\bar{q})$, where $\bar{q}$ is the moving average of queue size. The problem comes from the drop probability, p, that is limited by the maximum value, $p_{\max}$. For a fixed $p_{\max}$, there may be so large an amount of traffic that RED needs to increase p over $p_{\max}$, and so small an amount of traffic that $\bar{q}$ stays near the minimum queue threshold, $\min_{th}$.

In [2] it was shown that the relation between the number of active TCP connections (N), $\bar{q}$, and p is given by

$$p = K\left(\frac{N}{\bar{q}}\right)^2 \quad (1)$$

where K is a proportional coefficient. Fig. 1*a* shows (1) and $f(\bar{q})$ of RED, i.e.

$$f(\bar{q}) = p_{\max}\frac{\bar{q} - \min_{th}}{\max_{th} - \min_{th}} \quad (2)$$

where $\max_{th}$ is the maximum queue threshold. Since RED operates at equilibrium point ($\bar{q}$, p) where (1) and (2) intersect, as indicated by arrows in the graph, it cannot reach a proper operating point between $\min_{th}$ and $\max_{th}$ unless $p_{\max}$ is chosen carefully [3]. We show its limitation through NS simulation runs, where we use a simple dumbbell topology with a bottleneck link of 8 Mbits and 40 ms RTT. $\min_{th}$, $\max_{th}$, the queue size, and w_q of RED are set to 20, 60, 80, and 0.002, respectively. We change the number of active connections dynamically from 8 to 128. Since we set $p_{\max}$ to be optimised for $N=32$, RED works well when $N=32$, but fails to keep $\bar{q}$ between $\min_{th}$ and $\max_{th}$ when $N=8$ and $N=128$ as shown in Fig. 1*b*.

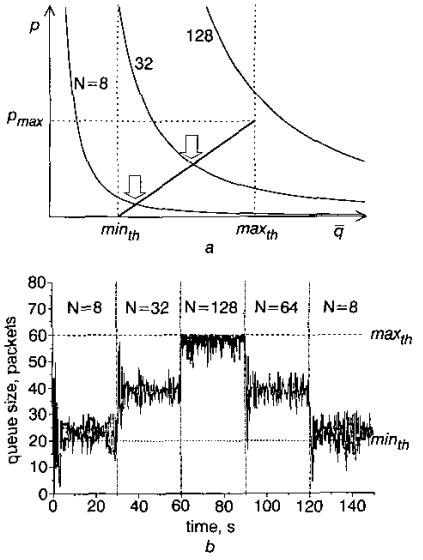

Fig. 1 *$f(\bar{q})$ and simulation results of RED*
a $f(\bar{q})$ Arrows indicate where (1) and (2) intersect
b RED

Self-configuring RED [4] corrects the problem without maintaining any flow states or statistic flow states. It increases $p_{\max}$ when $\bar{q} > \max_{th}$, and decreases $p_{\max}$ when $\bar{q} < \min_{th}$. It adjusts $p_{\max}$ according to the traffic load and overcomes the limit of RED, especially when there is a small amount of traffic. Fig. 2*a* shows its $f(\bar{q})$, and Fig. 2*b* shows the simulation results under the same environments as RED. Self-configuring RED successfully keeps $\bar{q}$ between $\min_{th}$, and $\max_{th}$ when $N=8$ and $N=32$. However, it operates unstably when a large amount of traffic, $N=128$, is offered.

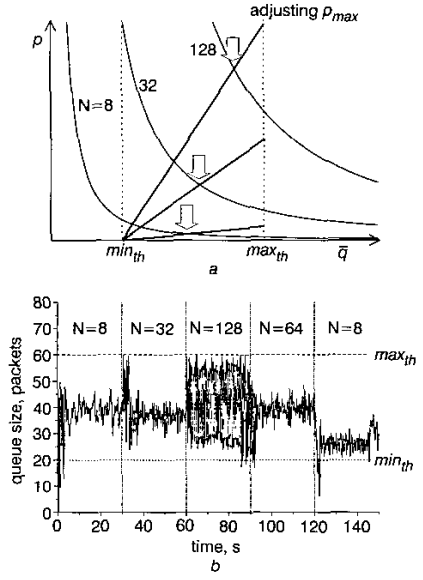

Fig. 2 *$f(\bar{q})$ and simulation results of self-configuring RED*
a $f(\bar{q})$ Arrows indicate where (1) and (2) intersect
b Self-configuring RED

This problem comes from the slope of $f(\bar{q})$. If the slope of $f(\bar{q})$ is too steep (large p_{max}), RED may increase p overly with a small increase of $\bar{q}$ and drops many packets before senders detect the drops. After the senders reduce their rates excessively, RED will get a small queue size and reduce p overly because of the steep slope of $f(\bar{q})$. Again, too small p results in an excessive increase of traffic, which overloads the bottleneck link and makes RED have large $\bar{q}$ and p. This process repeats and makes $\bar{q}$ oscillate with large variation. Since large variation of $\bar{q}$ causes large delay variation, repeated low link utilisation, and frequent queue overflows, it should be avoided. This example of unstable oscillation is shown at $N=128$ in Fig. 2*b*.

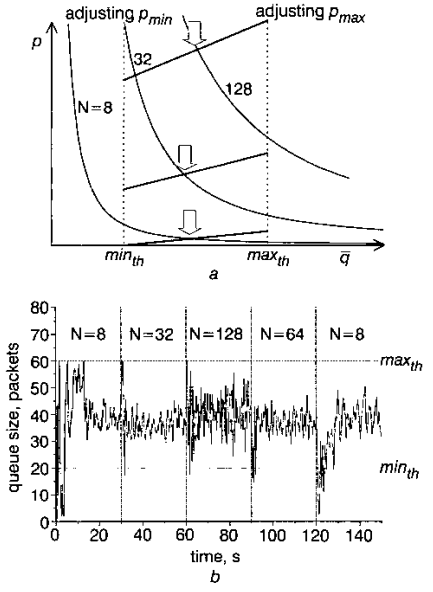

Fig. 3 *$f(\bar{q})$ and simulation results of proposed algorithm*
a $f(\bar{q})$ Arrows indicate where (1) and (2) intersect
b Proposed algorithm

Modification for scalability: Since the oscillation results from the steep slope of $f(\bar{q})$, we can settle the problem by introducing a new parameter, p_{min}, which is the drop probability when $\bar{q}$ is min_{th}. Then $f(\bar{q})$ can be written as

$$f(\bar{q}) = (p_{max} - p_{min})\frac{\bar{q} - min_{th}}{max_{th} - min_{th}} + p_{min} \tag{3}$$

Since we can control the slope through $p_{max} - p_{min}$, we can remove the unstable oscillation by keeping the slope of $f(\bar{q})$ properly. Fig. 3 shows $f(\bar{q})$ and the simulation results of our modification. Our algorithm is allowed to adjust both p_{min} and p_{max}. It increases p_{min} when $\bar{q} > max_{th}$ and decreases p_{min} when $\bar{q} < min_{th}$, and p_{max} is set to $p_{min} + \Delta$, where Δ determines the slope of $f(\bar{q})$, which is critical in the algorithm. Fortunately, a proper slope can be estimated from (1) and (3). Since the change of $\bar{q}$ at time t reflects on p immediately and the change of p at t affects $\bar{q}$ after time T (the interval T should be short enough that $\bar{q}$ does not change significantly, and long enough that senders can react to the change of p), we can get the following by differentiating (1) and (3), and eliminating dp/dt from both equations with assumption of $\bar{q}_{t+T} \simeq \bar{q}_t$,

$$\frac{d(\bar{q}_{t+T})}{dt} = \frac{\bar{q}_t}{p_t}\frac{p_{max} - p_{min}}{max_{th} - min_{th}}\frac{d(\bar{q}_t)}{dt} \tag{4}$$

which shows how much $\bar{q}$ at t affects $\bar{q}$ at $t+T$. As $(\bar{q}_t/p_t)(p_{max} - p_{min}/max_{th} - min_{th})$ is maximised when $p = p_{min}$ and $\bar{q} = max_{th}$, let S_{max} be $(max_{th}/p_{min}) - (p_{max} - p_{min}/max_{th} - min_{th})$. If $S_{max} > 1$, the change of $\bar{q}$ at t causes more change of $\bar{q}$ after T, and $\bar{q}$ will oscillate unstably because of positive feedback.

To confirm our analysis, we simulate RED with different fixed slopes of $f(\bar{q})$ by changing S_{max} from 1/8 to 32. The results are shown in Fig. 4. We measure the standard deviation of $\bar{q}$ according to the increase of N, and regard RED as unstable if the deviation exceeds 1/4 $(max_{th} - min_{th})$, i.e. 10 in our simulations. As N increases, $\bar{q}$ oscillates with less steep slope of $f(\bar{q})$. However, there is not much difference in the variation if S_{max} is less than 1. Hence, we can conservatively propose to have a proper S_{max} bound of 1/2, and adjust the slope of $f(\bar{q})$ accordingly:

$$\Delta = p_{max} - p_{min} = \frac{1}{2}(max_{th} - min_{th})\frac{p_{min}}{max_{th}} \tag{5}$$

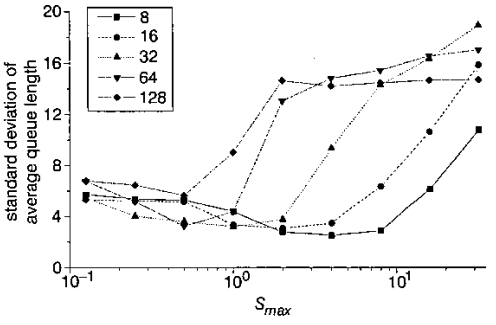

Fig. 4 *Variance of $\bar{q}$ against slope steepness*

Increase and decrease rates of p_{min} are also given in terms of Δ. Our simulation results setting the increase rate to 0.8 Δ and the decrease rate to 0.4 Δ are shown in Fig. 3*b*. Compared with those of RED and self-configuring RED, it is obvious that our modification succeeds in achieving the operating point between min_{th} and max_{th} without oscillation, and hence, succeeds in removing the scalability problem of RED.

Conclusions: RED is not scalable because of its use of fixed p_{max}. Self-configuring RED removes the limits but still has the scalability problem of instability with a large number of connections. We have proposed an algorithm that resolves this problem by keeping the slope of the drop probability function properly, and have shown the improved performance through simulations.

Acknowledgment: This work was supported by KOSEF under grant 1999-1-30200-005-3.

19 March 2002
Electronics Letters Online No: 20020744
DOI: 10.1049/el:20020744

Changhee Joo and Saewoong Bahk (*School of Electric Engineering, Seoul National University, Seoul, Korea*)

References

1 FLOYD, S., and JACOBSON, V.: 'Random early detection gateways for congestion avoidance', *IEEE/ACM Trans. Net.*, 1993, **1**, (4)
2 MORRIS, R.: 'Scalable TCP congestion control'. INFOCOM'00, Tel Aviv, Israel, March 2000
3 FIROIU, V., and BORDEN, M.: 'A study of active queue management for congestion control'. INFOCOM'00, Tel Aviv, Israel, March 2000
4 FENG, W., KANDLUR, D., SAHA, D., and SHIN, K.: 'A self-configuring RED gateway'. INFOCOM'99, New York, NY, USA, March 1999