Conclusion: A coding scheme that is highly successful on pure insertion or deletion channels is presented. It is capable of correcting one error event every 12 or 8 bits, for $R = 1/3$ or $R = 1/4$. It is also useful on mixed insertion/deletion channels where one error event per truncated block can be corrected. Furthermore, it has a significant reversal error correction capability.

12 April 2002
Electronics Letters Online No: 20020597
DOI: 10.1049/el:20020597

T.G. Swart and H.C. Ferreira (*Department of Electrical Engineering, Cybernetics Laboratory, Rand Afrikaans University, PO Box 524, Auckland Park, 2006, South Africa*)

E-mail: hcf@ing.rau.ac.za

References

1 LEVENSHTEIN, V.I.: 'Binary codes capable of correcting deletions, insertions and reversals', English translation in *Sov. Phys.-Dokl.*, 1966, **10**, (8), pp. 707–710, (published in Russian in 1965)
2 BOURS, P.A.H.: 'Codes for correcting insertion and deletion errors'. June 1994, Ph.D. Thesis, Technische Universiteit Eindhoven, Eindhoven, The Netherlands
3 DAVEY, M.C., and MACKAY, D.J.C.: 'Reliable communications over channels with insertions, deletions and substitutions', *IEEE Trans. Inf. Theory*, 2001, **47**, (2), pp. 687–698
4 FERREIRA, H.C., CLARKE, W.A., HELBERG, A.S.J., ABDEL-GHAFFAR, K.A.S., and VINCK, A.J.H.: 'Insertion/deletion correction with spectral nulls', *IEEE Trans. Inf. Theory*, 1997, **43**, (2), pp. 722–732
5 SWART, T.G.: 'Coding and bounds for correcting insertion/deletion errors'. March 2001, M.Eng Thesis, Rand Afrikaans University

Virtual rate control algorithm for active queue management in TCP networks

Hyuk Lim, Kyung-Joon Park, Eun-Chan Park and Chong-Ho Choi

A virtual rate control (VRC) algorithm for active queue management (AQM) to regulate the queue length with small variation and to achieve high utilisation with small packet loss is proposed. Through *ns* simulations, the effectiveness of the proposed VRC algorithm as compared with several well-known AQM schemes such as RED, REM, and AVQ algorithms is shown.

Introduction: There has been a considerable amount of research on active queue management (AQM) for congestion control in TCP networks. The random early detection (RED) gateways and many other algorithms have been proposed for AQM. The RED algorithm uses the averaged queue length for congestion control [1]. The random exponential marking (REM) algorithm uses the accumulation (time integral) of a weighted sum of the instantaneous queue length and the queue occupancy rate [2]. The adaptive virtual queue (AVQ) algorithm uses the virtual queue length, which is also the accumulation of the virtual queue occupancy rate [3].

In this Letter we propose a virtual rate control (VRC), which is based on rate control so that the packet input rate can be maintained around the link capacity. The marking probability of the algorithm is primarily proportional to the queue occupancy rate for fast response to traffic fluctuation. Furthermore, we adopt a notion of a virtual target rate for better agreement of the equilibrium input rate with the link capacity. This algorithm properly regulates the queue length with small variance, which relates to jitter in queueing delay, and achieves high utilisation with low packet loss rate.

Proposed algorithm: AQM algorithms control network congestion by dropping or marking packets with explicit congestion marking (ECN). When TCP sources detect incipient congestion, they reduce their sending rates before a buffer overflow occurs. The aggregate input rate of a queue should be equal to the output link capacity in order to maximise link utilisation and to regulate queue length properly. The aberration of the input rate from the output link capacity should be quickly compensated for because it affects the queue length and leads to increased jitter in queueing delay. We consider a proportional rate control instead of queue length control as follows:

$$p(t) = [\alpha(x(t) - x_r(t))]^+, \quad \alpha > 0 \tag{1}$$

where $p(t)$ is the marking probability, $x(t)$ is the aggregate input rate of the queue, $x_r(t)$ is the target rate, and $[\cdot]^+ \triangleq \max(\min(\cdot, 1), 0)$. The variation of the queue length depends on the difference between the input rate and the target rate, and the proportional rate control attempts to reduce this rate difference. Here the target rate $x_r(t)$ can be set to the link capacity C. However, instead of the constant target rate C we adopt a modified target rate to keep the queue length $q(t)$ around the target queue length q_r. We set the target rate as the sum of the link capacity and the difference between q_r and $q(t)$:

$$x_r(t) = C + \gamma(q_r - q(t)), \quad \gamma > 0 \tag{2}$$

If the queue length becomes smaller than the target queue length, there is more room to accommodate packets, and the target rate is increased. Otherwise, the target rate is decreased. However, the input rate $x(t)$ does not converge to the target rate $x_r(t)$ in equilibrium by using only (1) and (2).

To explain this discrepancy between the input rate and the target rate, we consider an overall steady-state TCP behaviour by a graphical method. Fig. 1 shows the equilibrium of the marking probability and the aggregate input rate when the proportional rate control algorithm (1) is applied to congestion control. We assume that the steady-state throughput $U(p)$ is a strictly decreasing function of the marking probability p. The equilibrium point (p_s, x_s) is at the intersection of the proportional control (1) and the throughput function. Note that the equilibrium input rate x_s is always greater than the target rate x_r, and consequently, there exists a rate error in equilibrium.

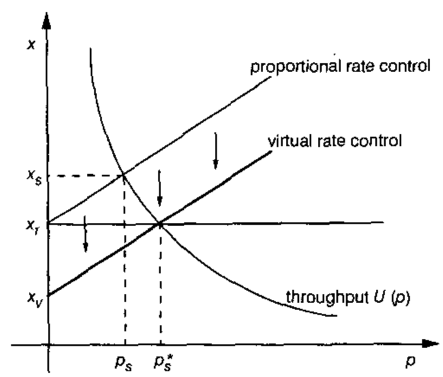

Fig. 1 *Equilibrium points for proportional control and virtual rate control*

To compensate for the rate error, we introduce a notion of a virtual target rate. Instead of using the target rate x_r in (1), the virtual target rate x_v is adopted to lower the proportional control towards the negative direction of the y-axis in Fig. 1, i.e. $p(t) = [\alpha(x(t) - x_v(t))]^+$. The VRC algorithm updates the virtual target rate $x_v(t)$ to minimise the difference between the input rate $x(t)$ and the target rate $x_r(t)$ as follows:

$$\begin{aligned} x_v(t) &= x_r(t) - \Delta x_v(t), \quad t = nT_s \\ \Delta x_v(t+1) &= \Delta x_v(t) + \beta(x(t) - x_r(t)), \quad \beta > 0 \end{aligned} \tag{3}$$

where T_s is the sampling interval. The complete algorithm can be implemented as the following pseudocode.

At every sampling instant

1. /* calculate the target rate x_r */
2. $x_r \leftarrow C + \gamma^* (q_r - q)$
3. /* calculate the virtual target rate x_v */
4. $\Delta x_v \leftarrow \Delta x_v + \beta^* (x - x_r)$
5. $x_v \leftarrow x_r - \Delta x_v$
6. /* calculate the marking probability and mark packets */
7. $p \leftarrow \alpha^* (x - x_v)$

To show that the input rate converges to the target rate by the VRC algorithm in Fig. 1, we consider the rate error $e_s(t) = x(t) - x_s^*$ in equilibrium. Here x_s^* is the target rate in equilibrium, i.e. $x_s^* = C + \gamma(q_r - q_s^*)$ from (2), and q_s^* is the queue length in equilibrium. The rate error $e_s(t)$ is then expressed as follows:

$$\frac{e_s(t)}{e_s(t-1)} = \frac{1 - \alpha\Delta U(p)}{1 - \alpha(1+\beta)\Delta U(p)} \tag{4}$$

where

$$\Delta U(p) = \frac{U(p(t)) - U(p(t-1))}{p(t) - p(t-1)} \tag{5}$$

Note that $\Delta U(p)$ is a linear approximation of the slope of $U(p)$. By the assumption that $U(p)$ is strictly decreasing, there exists a positive constant η such that $\Delta U(p) \leq -\eta$ for $0 \leq p \leq 1$, resulting in $e_s(t)/e_s(t-1) \leq \rho < 1$. Therefore, the rate error $e_s(t)$ converges to zero as $t \to \infty$, and the rate $x(t)$ becomes the target rate x_s^* in equilibrium. Furthermore, if we think of the queue dynamics $q(t+1) = q(t) + (x(t) - C)$, the queue length q_s^* becomes the target queue length q_r in equilibrium from (2).

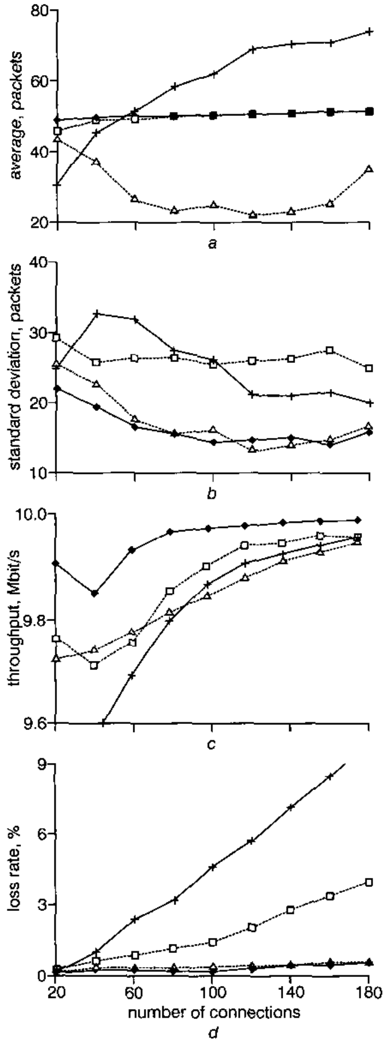

Fig. 2 *Queue length, standard deviation, utilisation, and packet loss rate*

a Average + RED
b Standard deviation □ REM
c Utilisation △ AVQ
d Packet loss rate ● VRC

Simulation: To compare the performance of the VRC algorithm with those of RED, REM, and AVQ, we conducted simulations using the *ns*-2 network simulator. Here we present the simulation results under a single bottleneck topology. A simple bottleneck network configuration was implemented with two routers and a number of TCP connections. The routers are connected through a link capacity 10 Mbits/s with a propagation delay between 50 and 150 ms. The average packet length is set to 1000 bytes. We used the TCP-Reno as the default transport protocol and the ECN marking instead of dropping packets. The target queue length at the bottleneck link is set to 50 packets for REM and VRC, and the min_{th} and the max_{th} for RED are set to 20 and 80 packets, respectively. The utilisation factor γ for AVQ is set to 1 for high utilisation.

We observed queue length, its standard deviation, total utilisation, and packet loss rate for various traffic load by changing the number of connections from 20 to 180 in Fig. 2. The average queue lengths of RED and AVQ are sensitive to traffic load. The average queue length of RED becomes larger as the number of the TCP connections increases. AVQ maintains small queue length. However, the average queue length of AVQ varies significantly depending on the number of connections, because AVQ does not control the queue length directly. REM and VRC keep the average queue lengths very close to the target queue length regardless of the number of connections.,The standard deviation of the queue length of the VRC algorithm is the smallest for most cases. Because the VRC algorithm compensates for the rate change even before the change affects the queue length, the variation of the queue length is reduced. RED results in a poor utilisation compared with the other algorithms. Since the queue length of AVQ is small, AVQ sometimes makes the queue empty, leading to lower utilisation than REM and VRC. The VRC algorithm achieves the highest utilisation for all cases. The loss rates of VRC and AVQ are nearly the same, and are lower than those of the other algorithms. Consequently, these simulation results show that the performance of VRC is better than the other algorithms.

Conclusion: We have proposed the VRC algorithm for active queue management to regulate the queue with small variation and to achieve high utilisation with small packet loss. The proposed algorithm is primarily based on rate control for fast response to traffic fluctuation. By introducing the virtual target rate, the algorithm keeps both the input rate and the queue length around their target levels. The simulation results show the effectiveness of the proposed algorithm.

15 March 2002
Electronics Letters Online No: 20020622
DOI: 10.1049/el:20020622

Hyuk Lim, Kyung-Joon Park, Eun-Chan Park, and Chong-Ho Choi (*School of Electrical Engineering and Computer Science and ASRI, Seoul National University, Seoul 151-744, Korea*)
E-mail: hyuklim@csl.snu.ac.kr

References

1 FLOYD, S., and JACOBSON, V.: 'Random early detection gateways for congestion avoidance', *IEEE/ACM Trans. Netw.*, 1993, **1**, pp. 397–413
2 ATHURALIYA, S., LOW, S.H., LI, V.H., and YIN, Q.: 'REM: Active queue management', *IEEE Netw.*, 2001, **15**, pp. 48–53
3 KUNNIYUR, S., and SRIKANT, R.: 'Analysis and design of an adaptive virtual queue (AVQ) algorithm for active queue management'. Proc. of ACM SIGCOMM, San Diego, CA, USA, 2001, pp. 123–134

2.4 GHz CMOS VCO with multiple tuning inputs

P.B.M. Hammer and P.M. Bakken

A fully integrated 2.4 GHz inductance-capacitance (LC)-tuned complementary metal-oxide-semiconductor (CMOS) voltage-controlled oscillator (VCO) with multiple tuning inputs is presented. The frequency tuning is achieved by use of four parallel, inversion mode NMOS varactors, each with its individual control electrode. It is demonstrated how the use of some of the inputs for coarse tuning can be used to keep down the VCO gain while still covering a wide frequency band. This could be useful as a means of reducing the sensitivity to noise.