ROUND-TRIP TIME ESTIMATION IN COMMUNICATION NETWORKS USING ADAPTIVE KALMAN FILTERING [1]

Krister Jacobsson * **Håkan Hjalmarsson** * **Niels Möller** *
Karl Henrik Johansson *
{krister.jacobsson|hjalmars|niels|kallej}@s3.kth.se

* *Department of Signals, Sensors and Systems, KTH,*
SE-100 44 Stockholm, Sweden

Abstract: Heterogeneous communication networks with their variety of application demands, uncertain time-varying traffic load, and mixture of wired and wireless links pose several challenging problem in modeling and control. In this paper we focus on the round-trip time (RTT), which is a particularly important variable for efficient end-to-end congestion control. Based on a simple aggregated model of the network, an algorithm combining a Kalman filter and a change detection algorithm is proposed for RTT estimation. It is illustrated on real data that this algorithm provides estimates of significantly better accuracy as compared to the RTT estimator currently used in TCP, especially in scenarios where new cross-traffic flows cause a bottle-neck link to rapidly build up a queue, which in turn induces rapid changes of the RTT.

Keywords: Estimation, TCP, Communication Networks, RTT

1. INTRODUCTION

Congestion control is one of the key components that has enabled the dramatic growth of the Internet. The original idea (Jacobson, 1988*a*) was to adjust the transmission rate based on the loss probability. The first implementation of this mechanism, denoted TCP Tahoe, was later refined into TCP Reno. This algorithm (together with some of its siblings) is now the dominating transport protocol on the Internet. The throughput and delay experienced by individual users are depending on several factors, including the TCP protocol, link capacity and competition from other users. There are also lower layers that may affect the achieved delay and bandwidth, particularly if part of the end-to-end connection is a wireless link, see (Mascolo *et al.*, 2001; Sarolahti *et al.*, 2003; Cen *et al.*, 2003; Samaraweera, 1999; Fu and Liew, 2003).

There exist a few studies of the statistics of network quantities (Allman, 2000; Allman and Paxson, 2001), and even estimation of RTT at the link level (Jiang and Dovrolis, 2002). However, there seems to be less work on considering RTT estimation from the perspective of TCP.

The outline of the paper is as follows. In Section 2 a brief presentation of TCP, and especially the importance of accurate RTT estimates, is given. This motivates the RTT model and estimation scheme presented in Section 3. A new algorithm based on a Kalman filter and change detection is proposed for RTT estimation. In scenarios where new cross-traffic flows cause bottle-neck queues to rapidly build up, the algorithm is shown to be particularly useful to track the rapid changes of the RTT. It gives significantly better accuracy compared to the RTT estimator currently used in most TCP versions.

[1] This work was supported by European Commission through the project EURONGI and by Swedish Research Council.

2. TCP AND RTT

TCP is window-based which means that each sender has a window that determines how many packets in flight that are allowed at any given time. The transmission rate is regulated by adjusting this window. For a network path with available bandwidth b and RTT τ, the optimal window size is $b\tau$, in the sense that if all users employ this window there will be no queues and the link capacities would be fully utilized. Using loss probability to measure congestion (as in TCP), means that the capacity of the network cannot be fully utilized. With necessity, queues have to build up (causing increased delays) and queues have to overflow (causing loss of throughput). Several methods to cope with these short-comings have been suggested. In TCP Vegas (Brakmo and Peterson, 1995) a source tries to estimate the number of packets buffered along its path and regulates the transmission rate so that this number is low (typically equal to three). One interpretation of this algorithm is that it estimates the round-trip queuing delay and sets the rate proportional to the ratio of the round-trip propagation delay and the queuing delay (Low *et al.*, 2002). Both round-trip delays are obtained from measurements of the RTT.

Congestion can be indicated in more sophisticated manners than just dropping packets. In random early detection (RED) (Floyd and Jacobson, 1993), packets are dropped before the buffer is full, with a probability that increases with the queue length. RED can thus be seen as a way of indirectly signaling the queue length to the source. In explicit congestion notification (ECN) the links add information concerning their status to the packets. As there is only one bit available in the packet header for ECN, clever coding is required, e.g., random exponential marking (REM) (Athuraliya *et al.*, 2001).

The transmission rate control problem can be solved in a completely decentralized manner, as was recently shown in (Kelly, 1997; Kelly *et al.*, 1998). Each source has a (concave) utility function of its rate. The optimization problem is to maximize the sum of the utility functions for all sources. It is shown that in order to solve this problem each source needs to know the sum of the link prices on the path. The link price is a a penalty function corresponding to the capacity constraint. It is a function of the total arrival rate at the link and can thus be computed locally at each router. This optimization perspective of the rate control problem has been taken in a number of contributions. The developed algorithms can be classified as (1) primal, when the control at the source is dynamic but the link uses a static law; (2) dual, when the link uses a dynamic law but the source control is static; and (3) primal–dual, when dynamic controls are used both at the source and the links, see (Liu *et al.*, 2003; Low and Srikant, 2003) for nice overviews. By appropriate choice of utility function even protocols not based on optimization, such as TCP Reno, can be interpreted as distributed algorithms trying to maximize the total utility (Low, 2000; Low and Srikant, 2003). TCP Vegas can be classified as a primal–dual algorithm with the queuing delay as a dynamic link price.

Network state variables such as queueing delays and RTT are essential for efficient congestion control, as has been recognized by many researchers, cf., (Paganini *et al.*, 2003). This has a simple intuitive interpretation: The ideal window is $b\tau$. Hence, the more accurate estimates of b and τ that are available, the closer to the ideal situation it is possible to keep each flow. It is also clear that when only indirect measures of congestion can be used, throughput will suffer somewhat, since queues have to start to build up in order for the source to detect an increased delay. The queues can be seen as a way to smooth out the uncertainties in the bandwidth and RTT estimates. It follows that one can obtain higher throughput than TCP Reno, since TCP Reno fills up the bottle-neck completely before reacting.

3. RTT ESTIMATION

Motivated by the previous discussion on the importance of accurate RTT estimates, we now take a closer look at short-range RTT from a statistical perspective. The raw RTT measurements, obtained from packet acknowledgements (ACK's), include delays caused by transient effects in the network (attributed to short-lived cross-traffic). The short-lived duration of these flows means that their contribution to the RTT can be considered as noise from the point of view of congestion control. It is thus reasonable to filter them out. In present versions of TCP this is done with a first-order low-pass filter. The average RTT (averaged over a few RTT) often makes sudden changes due to the appearance of a long-lived cross-flow somewhere along the path. It is then important for the filter to react quickly to this change, since otherwise buffers will start to build up with enlarged risk of packet loss and increased delay as consequences. It is impossible with a first-order filter to rapidly adjust to these changes. This motivates the use of a filter based on change detection for RTT estimation, as presented in this section. Before introducing the algorithm, we present a simple model for RTT. The section is ended by experimental evaluations.

3.1 Model

Let $\tau(t)$ denote RTT at the time the ACK is received at the source node. Introduce $d_{l,i}(t)$ for the link delay of link i, $d_{p,i}(t)$ for the corresponding propagation delay, and $d_{q,i}(t)$ for the queue delay. Suppose the considered end-to-end connection has m nodes. Then,

$$\tau(t) = \sum_{i=1}^{m} \left(d_{l,i}(t) + d_{p,i}(t) + d_{q,i}(t) \right).$$

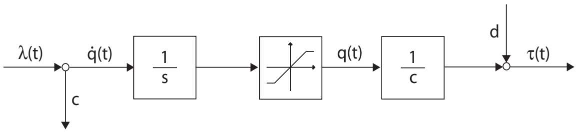

Fig. 1. Network node model.

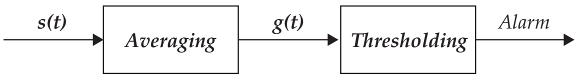

Fig. 2. The principle of change detection.

Assume, for now, that the path through the network remains constant during a session. The propagation and link delay is fairly constant and thus only contributes with a bias to the RTT estimate. All the major RTT dynamics is depending on the evolution of the queue states along the path.

The simple dynamics of the queue length $q_i(t)$ at node i is a saturated integrator, as illustrated in Figure 1. Let c_i denote the link capacity and t_i the time instant for the packet arrival at node i. Then, RTT at the time t when the ACK is received is given by

$$\tau(t) = \sum_{i=1}^{m} \frac{q_i(t_i)}{c_i} + \text{bias},$$

where the bias term represents the link and propagation delays.

3.2 *Change detection*

The RTT measurements have high-frequency characteristics that is desirable to detect. To be able to follow step changes in the RTT mean value due to increased network load, new competing traffic flows, or sudden path changes, more advanced algorithms is needed than the first-order linear filter currently implemented in most TCP versions. We propose an adaptive filter with change detection.

Regard RTT as being composed of a smooth desired RTT signal together with an additative high-frequency noise component. We thus model the desired RTT as a noisy observation of a constant exposed to step changes in the mean. If we denote RTT by y_t and the desired RTT by x_t, we obtain the model

$$\begin{aligned} x_{t+1} &= x_t + \delta_t v_t, \qquad \delta_t \in \{0,1\} \\ y_t &= x_t + e_t. \end{aligned}$$

The noisy characteristic of RTT is thus captured by the measurement noise e_t with variance R_e. The step changes in the desired RTT x_t is modeled as the process noise v_t with variance R_v and the discrete variable δ_t. If a change occurs at time t, then $\delta_t = 1$ otherwise $\delta_t = 0$. To estimate the sequence δ^N of instances when a change occurred is a segmentation problem. The optimal linear estimator is the Kalman filter which is used to estimate x_t. The principle of change detection is illustrated in Figure 2.

To estimate the sequence δ^N of instances of changes, we use a one-sided cumulative sum (CUSUM) algorithm (Gustafsson, 2000). Combining the modified Kalman filter with the CUSUM algorithm yields the following adaptive filter, which we denote the CUSUM Kalman filter:

$$\begin{aligned} \hat{x}_t &= \hat{x}_{t-1} + K_t(y_t - \hat{x}_{t-1}) \\ K_t &= \frac{P_{t-1}}{P_{t-1} + R_e} \\ P_t &= (1 - K_t)P_{t-1} + \delta_{t-1} R_v \\ \varepsilon_t &= y_t - \hat{x}_t \\ g_t &= \max(g_{t-1} + \varepsilon_t - \xi, 0) \end{aligned}$$

$\texttt{if}$ $g_t > h$ $\texttt{then}$
 $\delta_t = 1$ $\texttt{\% alarm}$
 $g_t = 0$
$\texttt{else}$
 $\delta_t = 0$
$\texttt{end}$

The output of the CUSUM Kalman filter is given by the estimate of the (desired) RTT $\hat{x}_t$. The filter has two design parameters: the negative drift ξ and the alarm threshold h. These parameters are tuned to adjust the sensitivity in the detection procedure. The same values have been used in all our initial experiments with satisfying results, so the filter seems fairly robust. Note that also the variances R_v and R_e influence the filter behavior.

3.3 *Experimental evaluation*

The objective is to capture the rapid changes that the RTT undergoes when queues build up when for instance the traffic load increases abruptly. We developed a modified ping tool to monitor RTT time series. By sending a dense stream of small packets the RTT is monitored effectively without affecting the network load too much. Note that data is not easily obtained from a TCP session itself, because a rapid traffic load increase would probably trigger the RTO mechanism in TCP (which would mean lack of samples during the transient). Also, if a queue along the path fill up, packet loss occurrences reduces the TCP sending rate with more sporadic time series as a consequence. Recall that the RTT time in congestion control is often in fractions of seconds (typically $10 - 500$ ms).

We measured the RTT between KTH and the CAIDA web site.This path is normally about 20 hops, including the Atlantic link. The mean RTT is approximately 190 ms. The sending interval (sampling time) of the ping tool was 30 ms. According to our measurements, the path is normally not congested with the result that the RTT almost shows a deterministic behavior with a low variance. However, sporadically the traffic load increased with the result of suddenly increased and fluctuating queues, which propagated to the RTT. The proposed CUSUM Kalman filter was used on the

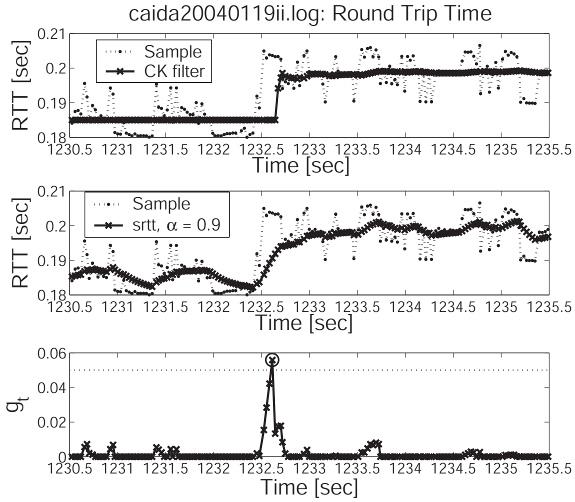

Fig. 3. Experimental evaluation of the proposed RTT estimation algorithm on a small change in RTT.

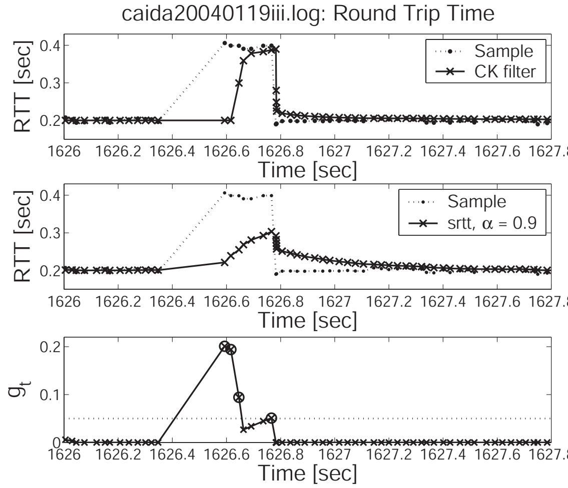

Fig. 4. Experimental evaluation of the proposed RTT estimation algorithm on a large change in RTT. Note how well the proposed algorithm (CK filter) captures the rapid change, while the conventional filter in TCP (srtt) adjusts much slower.

collected data sets. The variances R_e and R_v were set to the variance of the RTT samples. The CUSUM design parameters were set to $\xi = 0.005$ and $h = 0.05$ in all experiments. This was done manually based on initial experimentall results.

A detection of a sudden step in the RTT mean value can be seen in Figure 3. The two upper plots show the sampled RTT values, together with the output of the CUSUM Kalman filter and the output of the conventional TCP first-order filter (with gain set to 0.9 (Jacobson, 1988*b*)). In the lower plot the test statistic g_t is plotted and the change detection alarm is encircled. The CUSUM Kalman filter estimate is smooth, but still manage to detect a mean change as low as 7 %. Note that the TCP filter is more sensitive to noise and periodic fluctuations. As the change in mean is moderate in this example, the TCP filter adapts quite fast. The RTT mean change is probably a result of an abrupt change in the traffic load. An additive static traffic stream as a UDP flow might be the explanation, but a re-routing inside the network is also a possibility.

In Figure 4 we see a large change in the mean RTT of about 100 %. Even if the probing packets are sent with only 30 ms intervals, the RTT measurements do not capture the queue building up. The result becomes a step in the RTT measurements. In this scenario the proposed RTT estimation algorithm reacts after only a few samples, and is much faster than the filter in TCP. Note that this time it was no static mean change, but the queue vanishes after half a second and go down to its original level. The estimation algorithm reacts on the reset and adapts almost immediately while the TCP filter is lagging.

The two given examples show the main features of the CUSUM Kalman filter: smooth estimates but still fast adaption when drastic changes occur. In an application within a transmission protocol, the sampling time is typically larger and the measurements tend to be more spiky. The probability of hitting a plateau as in Figure 4 with more than a few samples is then low. By filtering out such events the RTT estimate could be kept at the appropriate level.

4. CONCLUSIONS AND FUTURE WORK

In this paper we have studied RTT estimation that could be suitable for congestion control applications. Weighting the characteristic of RTT and the application of the estimate together we state that the estimation objective is to keep the estimate smooth but still capture sudden mean changes. A Kalman filter combined with CUSUM change detection is proposed and shown to perform well based on experiments with real data. Long lived rapid changes are detected but short lived are filtered out as noise. The parameters of the CUSUM Kalman filter was tuned manually. Online tuning is an issue to investigate further. In the paper we have only considered the RTT estimation, not how the network metric itself should be used practically in a congestion control algorithm. The optimal window size, based on information from accessible implicit information as estimates of RTT and available bandwidth, will in the future be investigated in parallel with the development of the estimation procedure.

ACKNOWLEDGEMENT

The authors would like to thank Mikael Johansson and Gunnar Karlsson for fruitfull discussions and good advices.

REFERENCES

Allman, M. (2000). A web server's view of the transport layer. *ACM SIGCOMM Computer Communication Review*.

Allman, M. and V. Paxson (2001). On estimating end-to-end network path properties. *ACM SIGCOMM Computer Communication Review*.

Athuraliya, S., V. H. Li, S. H. Low and Q. Yin (2001). REM: active queue management. *IEEE Networking* **15**(3), 48–53.

Brakmo, L. S. and L. L. Peterson (1995). TCP Vegas: end-to-end congestion avoidsance on a global Internet. *IEEE Journal on Selected Areas in Communications* **13**(8), 1465–1480.

Cen, S., P. C. Cosman and G. M. Voelker (2003). End-to-end differentiation of congestion and wireless losses. *IEEE/ACM Trans. on Networking* **11**(5), 703–717.

Floyd, S. and V. Jacobson (1993). Random early detection gateways for congestion avoidance. *IEEE/ACM Trans. on Networking* **1**(4), 397–413.

Fu, C. P. and S. C. Liew (2003). TCP Veno: TCP enhancement for transmission over wireless access networks. *IEEE Journal on Selected Areas in Communications* **21**(2), 216–228.

Gustafsson, F. (2000). *Adaptive Filtering and Change Detection*. Wiley. West Sussex.

Jacobson, V. (1988*a*). Congestion avoidance and control. *ACM Computer Communication Review* **18**, 314–329.

Jacobson, V. (1988*b*). Congestion avoidance and control. In: *Proc. of SIGCOMM*. Vol. 18.4. pp. 314–329.

Jiang, H. and C. Dovrolis (2002). Passive estimation of TCP round-trip times. *ACM SIGCOMM Computer Communication Review*.

Kelly, F. P. (1997). Charging and rate control for elastic traffic. *European Transactions on Telecommunications* **8**, 33–37.

Kelly, F. P., A. K. Maulloo and D. K. H. Tan (1998). Rate control in communication networks: shadow prices, proportional fairness and stability. *Journal of Operational Research Society* **49**, 237–252.

Liu, S., T. Basar and R. Srikant (2003). Controlling the Internet: A survey and some new results. In: *Proc. 42nd IEEE Conference on Decision and Control*. Maui, Hawaii USA. pp. 3048–3057.

Low, S. H. (2000). A duality model of TCP and queue management algorithms. In: *Proceedings of ITC Specialist Seminar on IP Traffic Measurement, Modeling and Management*.

Low, S. H. and R. Srikant (2003). A mathematcial framework for designing a low-loss, low-delay Internet. *Networks and Spatial Economics*. special issue on "Crossovers between Transportation Planning and Tellecommunications".

Low, S. H., L. Peterson and L. Wang (2002). Understanding Vegas: a duality model. *Journal of ACM* **49**(2), 207–235.

Mascolo, S., C. Casetti, M. Gerla, M. Y. Sanadidi and R. Wang (2001). TCP Westwood: bandwidth estimation for enhanced transport over wireless links. In: *MobiCom*. Rome, Italy.

Paganini, F., Z. Wang, S.H. Low and J.C. Doyle (2003). A new TCP/AQM for stability and performance in fast networks. In: *Proceedings of IEEE Infocom 2003*.

Samaraweera, N. K. G. (1999). Non-congestion packet loss detection for TCP error recovery using wireless links. *IEE Proceedings-Communications* **146**(4), 222–230.

Sarolahti, P., M. Kojo and K. Raatikainen (2003). F-RTO: an enhanced recovery algorithm for TCP retransmission timeouts. *ACM SIGCOMM Computer Communication Review*.