Analytical derivation of one-way delay

J.H. Choi and C. Yoo

Delay estimation is a difficult problem in computer networks. Round trip time (RTT) is often used as an approximation of the delay but, because it is a sum of the forward and reverse delays, the actual one-way delay cannot be estimated accurately from RTT. Accurate one-way delay estimation becomes crucial because it serves a very important role in network and application design. A new scheme is proposed for estimating one-way delay. One-way delay, and forward and reverse delay are analytically derived.

Introduction: We address a fundamental problem in computer networks, namely, how to estimate one-way delay time between the sender and the receiver. It would be easy to find the one-way delay if it is guaranteed that there is a global time synchronisation between the sender and the receiver. If there is, the forward delay can be simply calculated by taking the difference between the receiver's clock and the sender's timestamp. The receiver can write this value into the ACK packet's header, and the reverse delay is also obtained similarly. Unfortunately, one-way delay cannot be obtained accurately because the clocks at the end systems are not synchronised with each other. Despite a considerable amount of research effort [1, 2], the clock synchronisation problem has not been solved to a level suitable for calibrating a packet's forward and reverse delay [3, 4]. Furthermore, in heterogeneous and massive networks such as the Internet, it is even more difficult to guarantee the synchronised clock. Therefore, a new scheme is needed for estimating one-way delay without requiring global clock synchronisation.

Delay derivation: Our scheme assumes that the receiver responds with ACK immediately for each arriving packet to the sender (i.e. the receiver does not employ a delayed ACK). Fig. 1 shows a typical interchange of packets and ACKs.

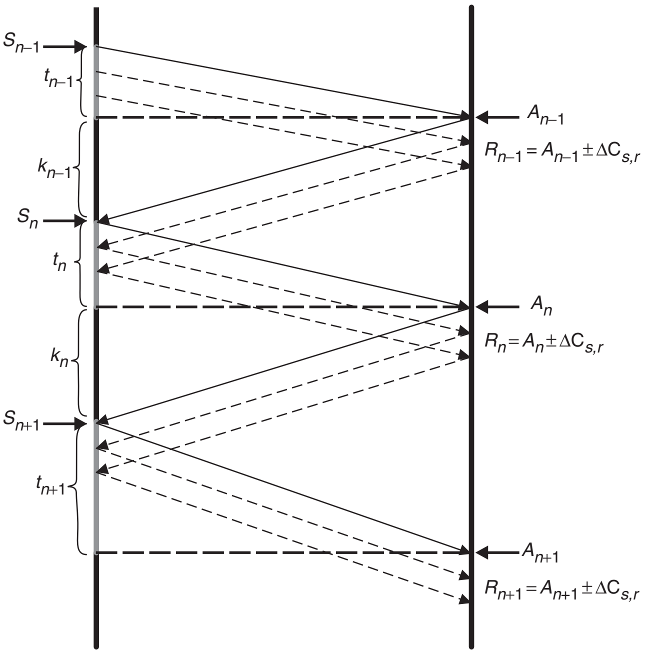

Fig. 1 *Typical packet exchange of sender and receiver*

Let us introduce the terminology for clocks, timestamp and delays used in our estimation scheme:

C_s sender clock
C_r receiver clock
S_n transmission time of the sender for nth packet according to C_s
R_n arrival time of nth packet at the receiver according to C_r
A_n arrival time of nth packet at the receiver according to C_s
t_n forward delay of the nth packet according to C_s; $A_n - S_n$
k_n reverse delay of the nth packet according to C_s; $S_{n+1} - A_n$
$RTT(s, n)$ round trip time of $(n-1)$th packet at the sender according to C_s
$RTT(r, n)$ round trip time of ACK for $(n-1)$th packet at the receiver according to C_r
$\Delta C_{s,r}$ relative offset of a clock C_s at sender with respect to a clock C_r at receiver.

It is very important to obtain the accurate A_n because both forward and reverse delays are calculated based on A_n, and the relative offset $\Delta C_{s,r}$ can be estimated by obtaining accurate A_n. However, A_n is not known at the receiver, and it is the reason that clock synchronisation is difficult in distributed systems. Therefore, our scheme does not use accurate spot of A_n but utilises the fact that the time length is the same and is independent of the clock skew.

We begin our derivation by defining the meaning of *RTT* more clearly.

RTT measured by sender: Sender transmits a packet and measures the time on receiving ACK, and the time difference is *RTT*. Therefore, it denotes $RTT(s, n+1) = t_n + k_n$, and it is an *RTT* measured by the sender, and it is a measured *RTT* at S_{n+1} according to C_s by the sender. Therefore, we obtain the following:

$$RTT(s, n+1) = S_{n+1} - S_n = t_n + k_n \tag{1}$$

RTT measured by receiver: Receiver measures *RTT* by the difference of the time of sending the previous ACK and the current ACK. So it denotes $RTT(r, n) = R_n - R_{n-1}$, and it is a measured *RTT* at R_n according to C_r by the receiver. Also, we get the relation $R_{n+1} - R_n = A_{n+1} - A_n$, since $R_n = A_n \pm \Delta C_{s,r}$ and $R_{n+1} = A_{n+1} \pm \Delta C_{s,r}$ are allowed. From this relation, we know that it is possible to get the forward delay consistent with C_s even though we use the *RTT* calibrated according to C_r by the receiver (we assume that the relative offset is constant on adjacent *RTT* phases). Thus, we obtain:

$$RTT(r, n) = R_n - R_{n-1} = A_n - A_{n-1} = t_n + k_{n-1} \tag{2}$$

Since the *RTT*s measured at the sender and receiver have forward delay t_n as common, subtracting the two equations results in:

$$RTT(s, n+1) - RTT(r, n) = k_n + k_{n-1} \tag{3}$$

By summing the difference from 1 to n, we get the following expression:

$$\begin{aligned} RTT(s, 2) - RTT(r, 1) &= k_1 - k_0 \\ RTT(s, 3) - RTT(r, 2) &= k_2 - k_1 \\ RTT(s, 4) - RTT(r, 3) &= k_3 - k_2 \\ &\cdots \\ RTT(s, n+1) - RTT(r, n) &= k_n - k_{n-1} \\ \sum_{i=1}^{n} RTT(s, i+1) - \sum_{i=1}^{n} RTT(r, i) &= k_n - k_0 \end{aligned} \tag{4}$$

Equation (1), $RTT(s, n+1) = t_n + k_n$ becomes $RTT(s, 1) = t_0 + k_0$, when $n = 0$.

Now we can rewrite the above equations by rearranging k_n and k_0:

$$k_n = RTT(s, n+1) - t_n$$
$$k_0 = RTT(s, 1) - t_0$$

Thus, we get

$$\begin{aligned} k_n - k_0 &= \sum_{i=1}^{n} RTT(s, i+1) - \sum_{i=1}^{n} RTT(r, i) \\ &= RTT(s, n+1) - t_n - RTT(s, 1) + t_0 \end{aligned} \tag{5}$$

Finally, we have the forward delay t_n:

$$t_n = t_0 - \sum_{i=1}^{n} [RTT(s, i) - RTT(r, i)] \tag{6}$$

We also have the reverse delay, k_n

$$k_n = -t_0 + \sum_{i=1}^{n+1} RTT(s, i) - \sum_{i=1}^{n} RTT(r, i) \tag{7}$$

Equations (6) and (7) mean that the forward and reverse delay can be calculated by sender-measured *RTTs* and receiver-measured *RTTs*. Note that the variation of the network condition only depends on the difference of the sender-measured and receiver-measured *RTT*. Also, the equation implies that the variation of one-way delay could be tracked accurately if the receiver measures *RTT*, and returns this value to the sender.

Conclusions: A new scheme has been proposed in which one-way delay can be analytically derived. Much of the work that has tried to calibrate packet transit times has focused on clock synchronisation and timestamp, but to our knowledge no assured solution exists. Therefore, we have chosen a different approach from those of previous work, which focuses on tracking the delay accurately. As a result, we have found that the difference of the sender-measured *RTT* and receiver-measured *RTT* describes the changing network condition and, using this concept, we have designed a new delay estimation scheme.

25 September 2003
Electronics Letters Online No: 20031168
DOI: 10.1049/el:20031168

J.H. Choi and C. Yoo (*Department of Computer Science and Engineering, Korea University, Seoul, Korea*)

E-mail: jhchoi@os.korea.ac.kr

References

1 PAXSON, V.: 'On calibrating measurements of packet transit times'. SIGMETRICS, 1998
2 MOON, S., SKELLY, P., and TOWSLEY, D.: 'Estimation and removal of clock skew from network delay measurements'. INFOCOM, New York, USA, 1999
3 MILLS, D.: 'Improved algorithms for synchronizing computer network clocks', *IEEE/ACM Trans. Netw.*, 1995, **3**, (3)
4 MILLS, D.: 'Network time protocol (version 3): specification, implementation and analysis'. RFC 1305, 1992