Journal of University of Science and Technology Beijing
Volume 10, Number 1, February 2003, Page 65

A method for measuring the clock offset of two hosts in the network

Xingye Yu[1], *Yang Yang*[2], *and Jihai Li*[3]

1) Legend Corporation of Research & Development, Beijing 100085, China
2) Information Engineering School, University of Science and Technology Beijing, Beijing 100083, China
3) Automation Company, Anshan Iron and Steel Complex, Anshan 114001, China
(Received 2002-04-07)

Abstract: In order to detect the performance parameters of the network, for example, the network delay or delay jitter, the clock synchronization relations between the two hosts at two ends along the network must be calculated in advance. Then with the correct temporal relations between the two hosts, multimedia transmission along the network and display can occur by the proper order. A refined method based on Paxson's algorithm is proposed and testified. More accurate results can be attained by the method. By the way, the method can be used in a more complicated environment. Furthermore, an end-to-end network performance tester based on the proposed algorithm is designed and implemented.

Key words: end-to-end measurement; network delay; clock synchronization

In end-to-end measurements, an end host sends probe traffic into the network. The sender inserts a time-stamp the departure time into the probe packet. If the probe packets are simply echoed back to the sender by a router or another end host, it is the echo-based measurement technique. If the receiver is involved in the measurement, *i.e.* it does more than just echo, this is a receiver-based measurement technique. The arrival time of these probe packets is recorded by the receiver and/or by the sender itself. The time-stamps produced by end hosts indicate the time reported by their own clocks, which are not usually synchronized with each other. By analyzing the observed round-trip or one-way transit time structure, one may be able to infer several performance parameters such as delay, bottleneck bandwidth, available bandwidth and hop-by-hop link bandwidth. End-to-end measurements have the advantage of not requiring network elements along a path to cooperate. However the observed time structures are affected by a large number of parameters that are difficult, if not impossible, to model independently.

The one-way delay measurement encounters much more difficult problems, which come from the fact that the clocks used at the sender and receiver usually are not synchronized. Therefore the relative offset and relative skew (to be defined later) must be removed from the observed delay samples. There are several algorithms to estimate and remove relative skew [1] but only one exists for estimating relative offset [2,3]. However, this algorithm does not work if the network path is asymmetric, *i.e.* the series of routers in the upward and downward route are not the same.

A simple way to measure the one-way delay between two hosts is to obtain the half value of the round trip delay. However, the accuracy of this measurement is low because of the asymmetric of the network path and the asymmetric of the payload of the forward and the backward.

A refined method based on Paxson's algorithm is proposed in this paper, and an end-to-end network performance tester based on the proposed algorithm is designed and implemented.

1 Clock terminology

The basic terminology is defined for discussing the characteristics of the clocks used in this study.

First introduce the meta-notation of a subscript "s" denoting the sender host, and "r" denoting the receiver host. Let C_s and C_r refer to the clocks at the sender end and the receiver end.

MTU. A MTU (Maximum Transmission Unit) is the largest packet size, specified in octets that can be sent in a network. TCP uses the MTU to limit the size of each packet in any transmission.

Resolution. A clock's resolution is the smallest unit by which the clock's time is updated (a "tick"). It gives a lower bound on the clock's uncertainty. Note that define resolution with respect to the clock's reported time and not to true time, for example, a resolution of 10 ms only means that the clock updates its notion of time in

Corresponding author:Xingye Yu **E-mail**:yuxy@legend.com

0.01 s increments, not the true amount of time between updates.

Let R_s and R_r refer to the resolutions at the sender end and the receiver end. Estimate a clock's resolution by inspecting the differences between successive packet timestamps. In general, take the smallest positive difference as an upper bound on the clock's resolution. Finally, note that this approach for estimating clock resolution produces at best an upper bound on the clock's true resolution.

Offset. The clock's offset or the clock's accuracy at a particular moment is the difference between the time reported by the clock and the true time (UTC) as defined by national standards.

The relative offset of two clocks is the difference between the times reported by them. For example, define C_r's offset relative to C_s at a particular true time T as $T_r - T_s$, *i.e.* the instantaneous difference between the readings of C_r and C_s at time T. For convenience, sometimes refer to this as C_r's relative offset at time T, with C_s implicitly being the clock to which C_r is compared.

For network measurement, the differences in time are the most important as computed by comparing the timestamps from two different clocks. The process of computing the difference removes any error due to clock inaccuracy with respect to true time; but it is crucial to notice that the differences themselves reflect good approximations to differences in true time.

Skew. The clock's skew at a particular moment is the difference in frequency between the clock and the UTC. The clock's skew measures the change of offset. The relative skew of two clocks is the frequency difference between them.

2 Assessing relative clock offset

2.1 Paxson's algorithm for estimation of relative clock offset

For estimating the relative clock offset between network clocks, Vern Paxson has developed an algorithm [2], which works with the following assumptions:

(1) Both clocks have zero skew with respect to true time.

(2) The measurement time is small enough that the relative offset is unchanged during measurement.

(3) The network path between two hosts is symmetric, which means that the upward route and downward route pass through the same routers.

(4) The links are bandwidth-symmetric, which means that the upward link and downward link between two adjacent routers in the network path have the same bandwidth.

Let C_s and C_r refer to the clocks at the sender s and the receiver r. $\Delta C_{r,s}$ denotes the relative offset between C_r and C_s. The one-way transit time (OTT) across the network from s to r is OTT_1, and the opposite one-way transit time from r to s is OTT_2. Note the assumptions in (3) and (4), then know that $OTT_1 = OTT_2 = \Delta T$. Note that Paxson does not assume that know ΔT itself.

Suppose a packet is sent from host s at time s_1 (with respect to C_s) and arrives in host r at time r_1 (with respect to C_r), and that a second packet sent in the opposite direction is measured to depart at r_2 and arrive at s_2 (**figure 1**). Then have:

$$r_1 = s_1 + \Delta T + \Delta C_{r,s} \tag{1}$$

$$s_2 = r_2 + \Delta T - \Delta C_{r,s} \tag{2}$$

Subtracting the equation (2) from the equation (1) then gives:

$$\Delta C_{r,s} = \frac{(r_1 - s_1) - (s_2 - r_2)}{2} \tag{3}$$

That is, from the raw, measured timestamps of the two packets alone, can estimate $\Delta C_{r,s}$ even if don't know ΔT.

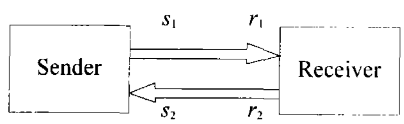

Figure 1 Timestamps of a probe packet.

The accuracy of equation (3) depends, however, on how closely the OTT_s of the two packets and fit in the assumption that they are equal. If a packet is sent through empty queues in the routers, it suffers the minimum end-to-end delay. In order to minimize variations in the OTT_s due to extraneous network delays such as queue, Paxson used the selected packets which have the minimal values (over all of the packets) of $(r_1 - s_1)$ and $(s_2 - r_2)$. By selecting minimal values, the variations are reduced because (most) network-induced noise is additive and positive, and therefore, minimal values tend to have the least noise. Accordingly, can estimate the relative offset by:

$$\Delta C_{r,s} \approx \frac{\min(r_1 - s_1) - \min(s_2 - r_2)}{2} \tag{4}$$

The $\min(r_1 - s_1)$ and $\min(s_2 - r_2)$ can be estimated by the minimum filtering [4]. In practice the internet routes often exhibit significant asymmetries in contrast with the above assumptions, so even in the absence of noise, packets sent in opposite directions along a path may experience considerably different delays. To deal with the problem of bandwidth asymmetry, a refined algorithm is proposed and described in the following.

2.2 Refined algorithm for asymmetric networks

The assumptions (3) and (4) in Paxson's algorithm are deleted in this refined algorithm, which means that $OTT_1 \neq OTT_2$. The other assumptions are the same as those in Paxson's algorithm, then have:

$$r_1 = s_1 + OTT_1 + \Delta C_{r,s} \quad (5)$$

$$s_2 = r_2 + OTT_2 - \Delta C_{c,s} \quad (6)$$

Subtracting the equation (6) from the equation (5) then gives:

$$\Delta C_{r,s} = \frac{(r_1 - s_1) - (s_2 - r_2) + (OTT_2 - OTT_1)}{2} \quad (7)$$

Still select the packets with the minimal values (over all of the packets) of $(r_1 - s_1)$ and $(s_2 - r_2)$, which means that select the minimal value of OTT_1 and the minimal value of OTT_2 in equations (5) and (6).

It is known that the end-to-end delay of a packet consists of the following parts as equation (8) indicated [5,6]:

$$(\text{End-to-end delay}) = OTT = D_c + D_v \quad (8)$$

where D_c includes the propagation delay, during which a packet stays in a physical link which is determined by the light speed, and the queue delay, during which the packet is waiting in the queue of the router when the router consults the tables. D_c can be regarded as a relatively steady value. D_v includes the transmission delay, during which a packet will be sent physically, *i.e.*, the time spent from sending the first bit in a packet until the last bit being sent, and the packet processing delay, the time required for a router to forward a packet, during which the router places a packet in the queue and sends it onto a communication link attached to it. D_v is in proportion to the size of the packet itself as follows [5,6]:

$$D_v = K \cdot s \quad (9)$$

From equations (8) and (9), have:

$$OTT = D_c + K \cdot s \quad (10)$$

Because the network-induced noise is usually additive and positive, select the minimal values of the delays:

$$\min OTT_1 = D_c + K_1 \cdot s \quad (11)$$

$$\min OTT_2 = D_c + K_2 \cdot s \quad (12)$$

From equations (5) and (6), have:

$$\min(r_1 - s_1) = D_c + K_1 \cdot s + \Delta C_{r,s} \quad (13)$$

$$\min(s_2 - r_2) = D_c + K_2 \cdot s - \Delta C_{r,s} \quad (14)$$

Subtracting equation (14) from equation (13) then gives:

$$\Delta C_{r,s} = \frac{\min(r_1 - s_1) - \min(s_2 - r_2) - (K_1 - K_2)s}{2} \quad (15)$$

The value of $\Delta C_{r,s}$ can be actually calculated by the different sizes of the packets and the minimal values of $(r_1 - s_1)$ and $(s_2 - r_2)$.

Equation (15) is the same as Paxson's algorithm when the network bandwidth of the forward route equals the backward route, which means $K_1 = K_2$.

A conclusion can be drawn that this algorithm can calculate the relative offset of the clocks of the two hosts in the network more accurately compared with Paxson's algorithm under the conditions of asymmetric bandwidth of the forward and backward networks and the different unidirectional delays.

3 End-to-end network performance tester

3.1 System description

An End-to-end Network Performance Tester (ENPT) based on the proposed method is designed and implemented in this study. The experiments have been done on the LAN in lab as well as the Campus Area Network (CAN). The following end-to-end network performance parameters can be measured by the tester:

(1) End-to-end network delay, including the maximum value, the minimum value, the average value, the variation value and the delay distribution.

(2) The bottleneck bandwidth of the link path.

(3) The numbers of packet loss or loss rates.

(4) The abnormality error, which includes numbers of out-of-order packets, packet repetition, packet errors.

The whole system comprises of two parts: the console and the end system.

The main task of the console is to control the transmission streams, *i.e.*, sending packets in the sender host according to some stream model and receiving packets in the receiver host, processing the returned results and displaying the corresponding figures and tables.

The end system is realized under linux environment. Its main task is to record the properties of the packet stream in both the sender and receiver, to analyze the primitive data, and to send the results to the console.

3.2 The experimental results and analysis

The experiments have been carried out under the environments of the LAN in lab and the campus area network respectively. The network configurations are as follows in **figures 2** and **3**.

Three models of transmission streams, the Poisson stream, the self-similar stream, and the average stream, are measured under both environments.

Table 1 shows the main results about network delay.

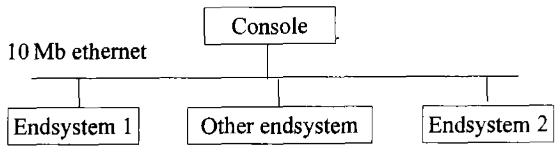

Figure 2 The LAN test environment in lab.

A conclusion can be drawn by table 1 that the delay parameters for the campus area network are all greater than those for the LAN. The conclusion can be simply verified by figure 2 and 3. The network delay as defined in international standard to fulfil the accurate presentation of multimedia streams and the inter-operation with users is less than 150 ms [7]. For both of the campus area network and the local LAN, the maximum delay and the average delay are within the tolerance set in the standard parameters of Quality of Service (QoS). The reality has actually been testified under these two environments.

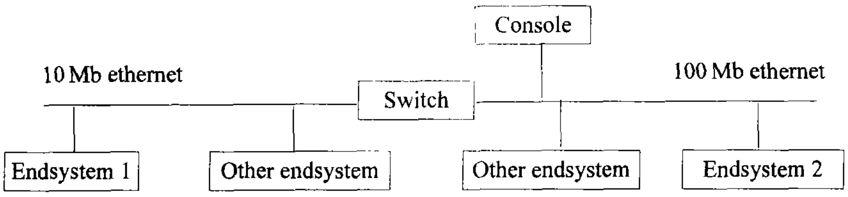

Figure 3 Test environment in the Campus Area Network.

Table 1 experiments results under the LAN and campus area network environments

Measurement environment	Stream model	λ	Hurst	$Delay_{max}$/ms	$Delay_{min}$/ms	$Delay_{aver}$/ms	$Delay_{var}$/ms^2
LAN	Poisson stream	200	—	6.978	0.089	2.620	0.002
	Self-similar stream	15	0.92	5.146	0.088	1.489	0.001
	Average stream	50	—	2.494	0.088	1.844	0.000
CAN	Poisson stream	20	—	17.164	1.320	6.484	0.004
	Self-similar stream	25	0.90	30.232	1.371	6.322	0.029
	Average stream	30	—	7.614	1.188	3.593	0.001

Note: λ and Hurst are parameters in the relative stream models.

4 Conclusion

To carry out real-time multimedia QoS measurement, the clock synchronization problems between two hosts in end-to-end communication have to be overcome beforehand. In this paper, a refined method based on Pasxon's algorithm has been put forward, and the accuracy of the measurement results is improved by this proposed method. Unfortunately, the "actual" value of clock offset between the two hosts in the network can not be obtained in fact, therefore, the accuracy of the proposed algorithm can not be verified in the sense of an absolute view. Finally, an end-to-end network performance tester is designed and implemented, and the results have been analyzed. An additional point is that the delay jitter is also an important parameter that influences the performance of the multimedia presentation, the method of measuring delay jitter is this next objective of research. Furthermore, this final goal is to make research into the method of combining the network performance measurement with multimedia synchronization control.

References

[1] Sue B. Moont, Paul Skelly, and Don Towsley, Estimation and removal of clock skew from network delay measurements, [in] *IEEE INFOCOM'99* [C], New York, 1999.

[2] Vern Paxson, On calibrating measurements of packet transit times, [in] *SIGMETRICS'98* [C], 1998.

[3] Vern Paxson, End-to-End internet packet dynamics [J], *IEEE/ACM Transactions on Networking*, 7(1999), No.3, p. 277.

[4] Allen B. Downey, Using pathchar to estimate internet link characteristics, [in] *ACM SIGCOMM '99* [C], Cambridge, MA, 1999.

[5] John F. Gibbon and Thomas D.C. Little, The use of network delay estimation for multimedia data retrieval [J], *IEEE Journal of selected area communications*, 14(1996), No.7, p. 1376.

[6] J. Kato, A. Shimizu, and S. Goto, End-to-end delay distribution on the internet [J], *IEICE trans. inf. &syst.*, E82-D (1999), No.4, p.762.

[7] Ralf Steinmetz, Human perception of jitter and media synchronization [J], *IEEE Journal of selected area communications*, 14(1996), No.1, p.61.