Next Generation Delay and Performance Measuring Algorithm for an Overall Network

Kazi Md. Abdullah Al Mamun, Gyanendra Prasad Joshi, Sung Won Kim
Department of Information and Communication Eng.
Yeungnam University, Korea
Email: *kazimamun@gmail.com, joshi@ynu.ac.kr, swon@yu.ac.kr*

Abstract - *As a network administrator or as a client, we need to perform the important operation over network such fixing a schedule of video conference, sending data, or doing maintenance related task regarding Quality of Service (QoS). It will be more reliable if we can get the information of the delay, performance statistics and error rate of the overall network that we belong to or we are maintaining. We propose the algorithm to implement such a program. We show that the method can operate in present and future networks in a few seconds. An extended timestamp like clock skew travels entire network randomly and gets the useful information of the entire network like processing time of each node, delay of each traveled path, number of hops it traveled, and error rate.*

1. Introduction

As the improvement of the Quality of Service (QoS) of a network and additional task that is required to maintain the network consistency is becoming important issues, there are a lot of works relating delay and performance measurement. [3] [4]

Clock skew algorithm is used for time synchronization with time server following the procedure in this paper. It is shown that, by using of the same technique in a different manner, we can measure the network delay and performance of a network.

2. Basic of Clock Skew:

Clock offset is a difference between the times of two clocks |Ci(t) – Cj(t)|. Clock skew (sometimes called timing skew) is a phenomenon in synchronous circuits in which the clock signal (sent from the clock circuit) arrives at different components at different times. Round-trip delay and clock offset are defined as

Round-trip delay: D = a – b

Clock offset: O = (a+b)/2

where a = d1(t)-s1(t), b = d2(t)-s2(t), s1(t) = source's sending time, s2(t) = source receiving time, d1(t) = destination's receiving time, d2(t) = destination's sending time. [1] [2]

Christian's algorithm is summarized as follows.

- Client *p* sends request *(mr)* to time server *S*,
- *S* inserts its time *t* immediately before reply *(mt)* is returned,
- *p* measures how long it takes *(Tround=T1 - T0)* from *mr* being sent to *mt* being received
- *p* sets its local clock to *t+Tround/2.* [7] [8]

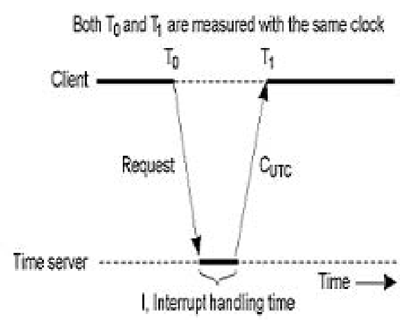

Fig. 1. Christian's algorithm's procedure[9]

3. Proposed Time stamp:

Final Destination Address		Destination Address	
s1(t)	s2(t)	d1(t)	d2(t)
Hop count increase		Processing time	

Table 1. Extended Time Stamp.

We have extended the clock offset measuring timestamp. Suppose source is A which is doing the operation and destination is B which will be changed randomly. [1]

978-0-7695-3322-3/08 $25.00 © 2008 IEEE
DOI 10.1109/NCM.2008.108

Here final destination will be the A's address to get back the time stamp. Note that s1(t) is A's sending time, s2(t) is the A's receiving time, d1(t) is the receiver's receiving time, and d2(t) is the receiver's sending time. Hop count is the number of nodes required for round trip and the processing time is the total processing time of traveled nodes during the round trip.

The procedure of the algorithm is as follows.
Step0: destination address = random IP, set processing time = 0, set Hop count = 0,
Step1: set the s1(t).
Step2: send the packet using routing table.
Step3: set d1(t), increase hop, set d2(t), increase processing time.
Step4: check whether destination address = IP address? If YES set destination address = final destination address or go to step 2.
Step5: check whether final destination = IP address? If yes set s2(t) and stop or go to step 2.

This process will continue during 100 or 1000 times while we store those data in an array

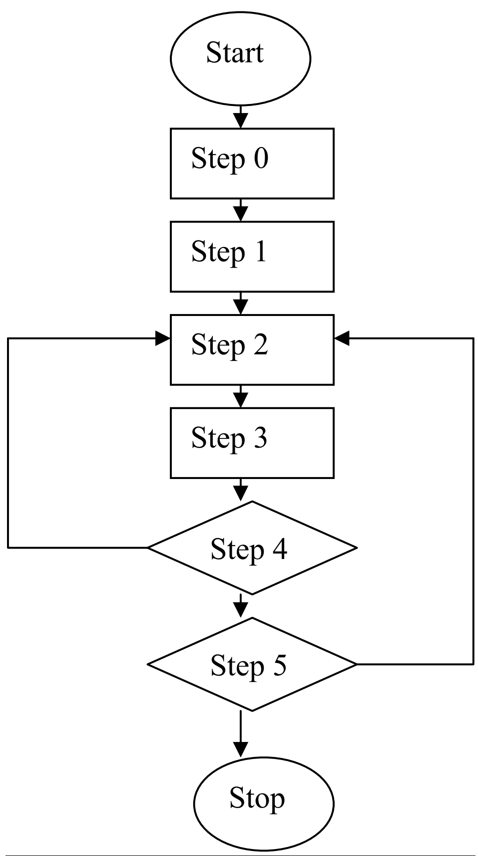

Fig. 2. the flow chart of the algorithm.

5. Simulation Results

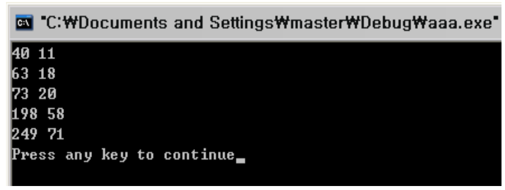

Fig. 3. At time T1, sample output showing processing time at first column and number for hop at 2nd column.

Fig. 4. At time T2, Sample output showing processing time at first column and number for hop at 2nd column.

Fig. 5. At time T3, Sample output showing processing time at first column and number for hop at 2nd column.

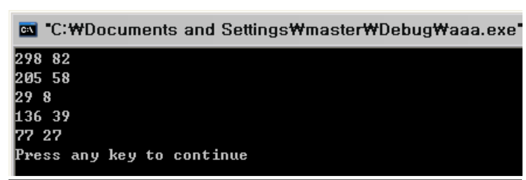

Fig. 6. At time T4, Sample output showing processing time at first column and number for hop at 2nd column.

Fig. 7. At time T5, Sample output showing processing time at first column and number for hop at 2nd column.

In the simulation of Fig. 3 ~ 7, we find out the following result of the performance statistics.

- Maximum Round Trip Delay
- Minimum Round Trip Delay
- Maximum Hop count
- Minimum Hop count
- Total and average processing time of nodes
- Packet loss

From Fig. 3 ~ 7, we can find the maximum delay of the network for one round trip is 707, though the minimum is 16. On other hand minimum hop is 5 to reach the destination though the maximum is 198.

From Fig. 8, we can see the average delay and the No of Hop how vary over time T1~T5. Now the decision can be come out which time will be suitable for our any specific task.

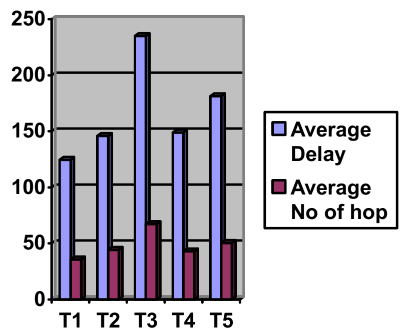

Fig. 8. Average delay and number of hops in different time.

6. Conclusion:

We have presented in this paper the algorithm and the technique to measure the possible delay and performance of a network in different times. In addition, by using one time limit, we can calculate packet loss. The proposed time stamp is so short that in the future it can be used with data packet as well as control packet. And it would not be a load to a network.

Even in the future it can be used with the IPV6 packet header. The results of the simulation of delay measurements presented are based on a test carried out over the particular period of time. Hence regular monitoring of networks is desirable to ensure the QoS of the network and the performance to be maintained and improved. [5] [6]

8. References:

[1] Moon, S.B.; Skelly, P.; Towsley, D., "Estimation and removal of clock skew from network delay measurements," *INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE* , vol.1, no., pp.227-234 vol.1, 21-25 Mar 1999

[2] Neves, J.L.; Friedman, E.G., "Optimal clock skew scheduling tolerant to process variations," *Design Automation Conference Proceedings 1996, 33rd* , vol., no., pp.623-628, 3-7 Jun, 1996

[3] Sun Hongjie; Fang Binxing; Zhang Hongli, "A Distributed Architecture for Network Performance Measurement and Evaluation System," *Parallel and Distributed Computing, Applications and Technologies, 2005. PDCAT 2005. Sixth International Conference on* , vol., no., pp. 471-475, 05-08 Dec. 2005

[4] Shufen Liu; Lu Han; Xinjia Zhang; Kai Nie, "Study of network performance measurement based on SNMP," *Computer Supported Cooperative Work in Design, 2004. Proceedings. The 8th International Conference on* , vol.2, no., pp. 119-123 Vol.2, 26-28 May 2004

[5] Phillips, I.; Sandford, M.; Parish, D., "Processing network delay measurements into network events," *Network Operations and Management Symposium, 2000. NOMS 2000. 2000 IEEE/IFIP* , vol., no., pp.955-956, 2000

[6] Hong Li; Mason, L., "Estimation and Simulation of Network Delay Traces for VoIP in Service Overlay Network," *Signals, Systems and Electronics, 2007. ISSSE '07. International Symposium on* , vol., no., pp.423-425, July 30 2007-Aug. 2 2007

[7] http://everything2.com/e2node/Christian%2527s%2520Algorithm

[8] http://en.wikipedia.org/wiki/Cristian's_algorithm

[9] http://www2.wmin.ac.uk/~lancasd/CCT792/Lectures/L12/img17.html