HOTS: An OWAMP-Compliant Hardware Packet Timestamper

Zhang Shu and Katsushi Kobayashi

National Institute of Information and Communications Technology, Japan
{zhang,ikob}@koganei.wide.ad.jp

1 Introduction

Accurate timestamps on both the sender and the receiver side are crucial for one-way delay (OWD) measurements. Traditionally, the methods of (i) peering with NTP servers, and (ii) connecting to a time source directly, have been used to maintain the accuracy of a measurement system clock. However, it has became clear that such methods suffer from errors to different extents.

In this paper, we introduce the hardware OWAMP [1] timestamper (HOTS), which generates extremely precise clock information for OWAMP test packets on both the sender and the receiver side. Compared with traditional methods, HOTS offers the following advantages: (i) the generated timestamp can be extremely precise because HOTS accepts an external 10-MHz signal as well as the 1PPS signal as input, and (ii) HOTS bypasses all of the software processing, thus minimizing possible errors. We also present the early results of OWD measurements that we made using this timestamper.

DAG [2] is a similar measurement instrument which also uses hardware to generate timestamps for a packet. However, this product is only designed to record the arrival timestamp of a packet and cannot be used to measure OWD.

2 Methodology

2.1 OWAMP Overview

The one-way active measurement protocol (OWAMP) is designed to measure one-way delay, jitter or packet loss. It consists of two inter-related protocols: OWAMP-Control and OWAMP-Test. OWAMP-Control is used to initiate, start and stop test sessions and to fetch their results, while OWAMP-Test is used to define the format of the test packets.

OWAMP test packets are transmitted in UDP datagrams. The header of the packets includes an 8-byte "Timestamp" field where the sender inserts the clock information when an OWAMP test packet is sent.

2.2 HOTS

Simply speaking, HOTS is a packet over SONET (POS) network interface card (NIC) which has a function to generate timestamps for outgoing or incoming packets destined for a specific port. Because it uses a PCI bus, HOTS can be used on most of the PCs.

HOTS has three I/O ports: one bidirectional SC connector and two mini-BNC jacks. The SC connector is used to send and receive packets, just as a normal SC connector does. The two mini-BNC jacks are used to obtain clock information from an external source such as a GPS or CDMA receiver.

HOTS can accept two kinds of signal as input: a 1PPS signal and a 10-MHz signal. The precision of the generated timestamp depends on the accuracy of the provided signals. In our measurements, we used two kinds of GPS receiver to generate these signals: the HP 58503A and the TymServe 2100. Both of them can generate extremely precise clock information.

HOTS maintains two clock-related counters: C_1 and C_2, which respectively hold the seconds and the fractions of a second based on the two kinds of external signals. The counters are operated as follows.

1. When reset (or the interface becomes up), both counters are cleared to zero.
2. In operational mode, C_2 is incremented based on the external 10-MHz signal.
3. When the 1PPS signal is received, C_1 is incremented and C_2 is cleared to zero.

Sender behavior When HOTS is used on the sender side, it works as follows.

1. When receiving a packet from the upper layer (usually the driver program), HOTS checks whether this packet is a UDP datagram and is destined for a specific port.
2. If it is, HOTS generates a timestamp (t_s) based on the two clock counters, inserts the timestamp into the "Timestamp" field of the packet, recalculates the UDP checksum based on the original one and the new timestamp, and then sends the packet to the physical link.
3. If the packet is not a UDP packet or is not destined for a specific port, HOTS does nothing apart from sending the packet to the physical link.

Receiver behavior When used on the receiver side, HOTS behaves as follows.

1. When receiving a packet from the physical link, it checks whether this packet is a UDP datagram and is destined for a specific port.
2. If it is, HOTS generates a timestamp (t_r), and passes the timestamp to the upper layer as well as the received packet. How the timestamp is passed to the upper layer depends on the users. In our measurements, we directly recorded (in the driver program) the timestamp in the body of the OWAMP test packet for simplicity. We also cleared the UDP checksum so that the datagram would not be dropped in the UDP processing because of inconsistent UDP checksums.
3. If the packet is not a UDP datagram or the port number is not a specific one, HOTS simply passes the received packet to the upper layer as other NICs would.

The OWAMP program on the receiver side will receive the packet in the user-space by normal socket API and the OWD can be calculated by

$$D = t_r - t_s \tag{1}$$

HOTS works with both IPv4 and IPv6 OWAMP test packets.

3 One-Way Delay Measurements

3.1 Measured network

We made OWD measurements on the APAN-JP network, which is part of the Asia-Pacific Advanced Network [3]. The topology of the measured network is shown in Fig. 1. PC_1 and PC_2 were the two end hosts between which we sent and received OWAMP test packets. These hosts were located at our institute in Tokyo and a data center in Fukuoka, respectively. There were five routers between the two hosts. The major distances between the two hosts were the 30 km between our institute and downtown Tokyo, and the 900 km between Tokyo and Fukuoka.

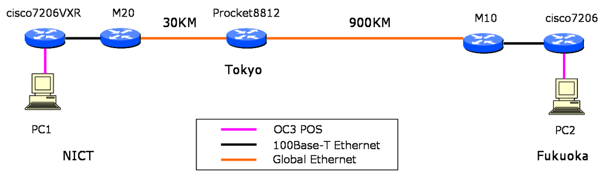

Fig. 1. Topology of the measured network

We periodically sent and received four kinds of test packet: packets of either 64 bytes or 1400 bytes in IPv4 or IPv6. All of these packets were sent once per second.

3.2 Measurement results

Some early results for the IPv4 packets are shown in Fig. 2. From this graph, we can see that the OWD for the IPv4 64-byte packets was usually about 10.7ms, and the OWD for the IPv4 1400-byte packets was several milliseconds longer. For IPv6, the results were similar to those for IPv4 packets.

3.3 Adaptation for other OWAMP implementation

HOTS can be easily used in other OWAMP software, such as the implementation from Internet2 [4], to perform highly precise measurements with the following modifications.

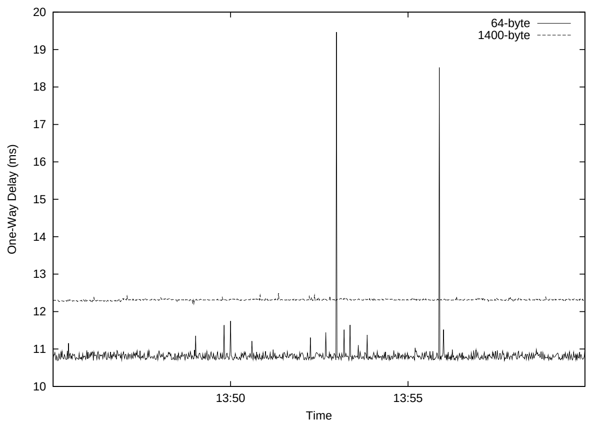

Fig. 2. Typical results for IPv4

- Specify the negotiated port numbers of the test packets on both the sender and the receiver side before transmitting a test packet.
- On the receiver side, use the hardware-generated timestamp when calculating the OWD.

4 Conclusion and Future Work

Highly precise OWD measurement is a challenge because of the difficulty of eliminating errors in process to obtain an accurate timestamp and other overheads. In this paper, we introduced HOTS - a hardware packet timestamper that we developed to measure OWD. HOTS can generate extremely precise clock information for OWAMP test packets provided an accurate time source such as a GPS or CDMA receiver. We presented the results of a preliminary OWD measurement that we did on the APAN-JP network to show its effectiveness.

References

1. S. Shalunov, B. Teitelbaum, A. Karp, J. W. Boote, and M. J. Zekauskas. (2004, Aug.) A one-way active measurement protocol (OWAMP). Internet draft. [Online]. Available: http://www.ietf.org/internet-drafts/draft-ietf-ippm-owdp-10.txt
2. DAG network monitoring interface cards. [Online]. Available: http://www.endace.com/networkMCards.htm
3. Asia-Pacific Advanced Network. [Online]. Available: http://www.apan.net
4. Internet2 OWAMP implementation. [Online]. Available: http://e2epi.internet2.edu/owamp