Precision Timestamping of Network Packets

Jörg Micheel[1,2], Stephen Donnelly[1], Ian Graham[1]

{joerg,sfd,ian}@cs.waikato.ac.nz

[1]WAND, Computer Science
The University of Waikato
Hillcrest Road, Gate 8
Hamilton, New Zealand

[2]NLANR MOAT, SDSC, USCD
10100 John Hopkins Dr
La Jolla, CA 92093-0505

Abstract – **When recording network traffic, accurate timestamping of the arrival of packets is essential for the subsequent analysis of performance metrics. Until the mid-1990s using off-the-shelf network interface cards and computer clocks proved sufficient. With the introduction of ever increasing link data rates the task of proper timestamping becomes increasingly important for continued network research. In the past five years the Dag development team in the WAND research group has undertaken substantial efforts to meet those demands and in this paper we discuss the advantages and limits of this new, hardware driven approach and explain how to interpret high precision timing information for packet arrivals.**

Index term: **computer networks, measurement, timestamping, performance.**

A. INTRODUCTION

Network packet traces are being used to investigate a number of research questions in the Internet area. Timestamping the arrival of packets allows the investigation to retain a notion of the original traffic time-profile on the network link. More specifically, timestamps allow the arrival of one packet within the trace to be correlated with another, to calculate per-link metrics such as utilization figures, or performance of applications observed on this link, such as TCP flow throughput, delay and jitter. Another important area for timestamping is multipoint measurements: the ability to correlate events at one link in the network relative to observations at a different link. Such a scenario will require synchronized clocks, an issue we discuss in this paper.

B. TIME STAMPS AND THE DUCK

When designing the new timestamping system for the Dag [2] measurement cards, coined the Dag Universal Clock Kit (DUCK), we were driven by a number of considerations.

From previous experience with the Dag1 measurement cards and OC3MON[1] we had learned that a 32-bit counter will not be able to provide sufficient resolution to distinguish back-to-back packet arrivals on high-speed network links and, at the same time, run for extended measurement periods without overflowing (wrapping). The two choices were either to provide a reliable mechanism for timestamp overflow notification, or to extend the number of bits in the timestamp. With a 32-bit timestamp and overflow indication an analysis application would have to work out time via calculation, rather than taking straight values for computation. This would result in replication of functionality and raise the chance of error. A timestamp with more bits increases the per-packet overhead in trace records, which, due to limited processing bandwidth on the card, may cause packet loss for runs of small packets.

We were also looking for a more standardized way of representing time across a series of different network interface cards. With increased link bandwidth packet arrival events tend to occur closer to each other, increasing the resolution required in the timestamping mechanism.

An important factor was the ease of handling of timestamps in both hard- and software. In particular, this criterion would rule out the use of floating point numbers to represent timestamps.

As a result, we selected a fixed-point 64-bit timestamp format, which is equivalent to the one used by the NTP protocol [7]. This format utilizes 32 bits

for seconds and 32 bits for fractions of a second. This provides for a measurement interval without wraps of 68 years along with a timestamp resolution of around one quarter of a nanosecond (nS). For comparison, an ATM cell of 53 bytes occupies an 11 nS timeslot on an OC768c network. The one difference of DUCK timestamps from NTP timestamps is that NTP uses January 1st 1900 as zero, which requires all 64 bits to represent current time. Our epoch coincides with UNIX, which uses January 1st 1970. Apart from convenience, the advantage is that the most significant bit will not be required until the year 2038, which allows the interpretation of timestamps as 64-bit signed integers in C. Timestamps thus can be directly subtracted and compared against zero, rather than the two-stage process required for unsigned integers.

As in NTP, we distinguish between the number of bits available in the format of the timestamp and the actual resolution of the clock. In other words, some of the least significant bits may be set to zero. On the Dag3 architecture the DUCK synthesizes a clock running at 16,777,216 Hz, which provides for 24 bits of fraction and sets the least significant byte to zero. This results in a clock resolution of about 60 nS. The Dag4 architecture, dealing with packet arrivals at OC48c line rate, has increased the resolution by an extra bit (reducing the time of one tick to 30 nS). This causes no changes to the format of the trace records or subsequent software post processing modules, which we consider a major advantage.

The implementation of the DUCK is based on a digital frequency synthesiser (DDS), a technology deployed in keyboards and other musical instruments to generate a range of frequencies from a fixed reference, and for tuning of radio frequency sections in devices such as cell phones. A rate register, with a resolution of up to 32 bits, controls the synthetic frequency. For further information please read the chapter on clock generation in [3].

The correct value for the DDS rate register is controlled by an external synchronisation input, normally a 1 pulse per second (PPS) signal from a GPS or CDMA timing receiver. On each PPS event the DUCK time is read and compared with the expected time. This difference is usually small, a few microseconds the most, as the synthetic frequency of the DUCK is ultimately derived from a crystal oscillator with a stability of a few per parts per million (ppm). There are two different errors that the software will correct for: offset and frequency. The controlling algorithm programs a DDS rate, which will minimize the clock offset at the next PPS. Once the PPS is applied, the DUCK will take a few (typically less than 10) seconds to settle. The DUCK's clock is considered to be in sync with the external source when the offset error is within ±10 ticks at the synthetic frequency. In other words, for the Dag3 architecture the maximum offset error will be around ±600 nanoseconds. The threshold of 10 ticks has been chosen from practical considerations, it is a trade-off between the accuracy required and the time it takes for the algorithm to settle. After a few seconds, the controlling algorithm will reach equilibrium and the offset will vary within ±2 ticks (Figures 1-3). For the Dag3 series of cards this implies an error of ±120 nanoseconds.

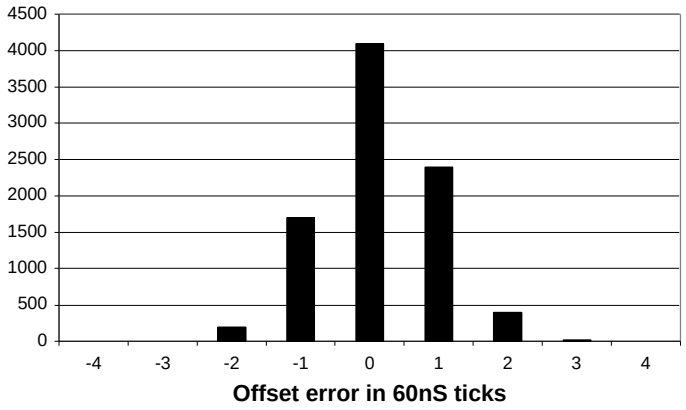

Figure 1. DUCK clock offset for a 24-hour period.

We have carried out experiments to investigate the accuracy of the DUCK in local synchronisation by sending the same packet stream to two or more Dag cards; we have not yet found a simple method to confirm the claimed accuracy of GPS or CDMA time receivers. Figures 2 and 3 show the results of typical experiments. Here the cards are Dag4.11s, and a third Dag in transmit mode acts as the packet source. The cards are running with a synthetic clock rate of 2^{25} Hz, or 29.8 nS per tick.

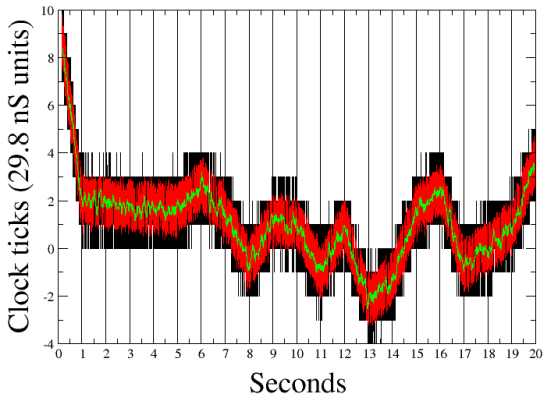

Figure 2. Two card timestamp difference.

In figure 2 the two cards are fed by the same 1 pps signal from a GPS receiver. The operation of the rate control mechanism is clearly seen. One card starts up first, and the second card starts collecting data only when its offset error is down to 10 ticks.

Within the first second the error is reduced to 2 ticks, and then the relative error oscillates with a period of a few seconds, but is the amplitude of the error is never greater than 4 ticks or 120 nS. The oscillations we see are the result of the two cards independently trying to synchronize to the same 1pps input.

In figure 3 the two cards are synchronized to each other; one acts as the master clock with a free running oscillator generating the synchronization pulse, the other card synchronizes its clock to that pulse.

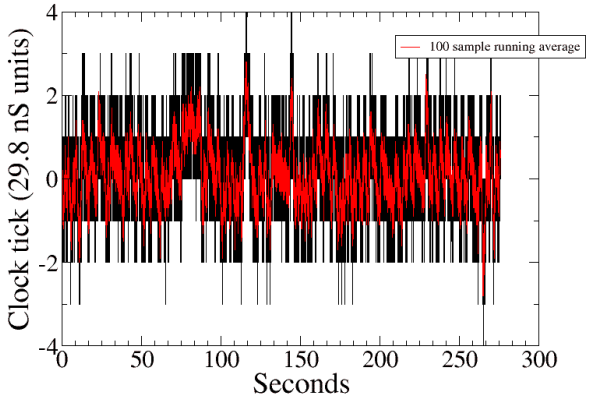

Figure 3. Dag cards in master-slave relationship.

This is a somewhat simpler situation, the master card will drift slowly in frequency, and the slave will attempt to keep in synchronization.

We conclude that the relative accuracy of two Dag cards, synchronized to the same input source, is of the order of 4 ticks of their synthesized clocks. We are working on enhanced algorithms to try to improve this figure.

For measurement integrity it is essential that the PPS be present at all times during a measurement session. Our implementation has been designed for extremely long measurements, lasting weeks or even months. A higher level function, called *DUCK health monitoring*, watches the presence of the PPS source and counts all single PPS outages, the length of the longest outage and how often the PPS was lost for a time period which could affect the accuracy of the timestamping process. This audit trail also preserves the maximum offset and frequency errors encountered. We have recently documented the precision of the timestamping process for a six-week continuous network trace.

C. Errors in the Timestamping Process

As simple as the actual approach of timestamping packets in hardware sounds, it carries in itself quite a number of sources for potential problems, or bugs. A full discussion of the issues that we found during testing and initial deployment will go beyond the scope of this paper. Keys to the elimination of implementation errors are a very tight and careful specification for appropriate test traffic and extensive test runs covering all the possible error conditions. Also, users should never assume equipment to be working *per se* under all conditions. If the required result is to be relied upon, additional checks on the captured trace data should be run to reinforce confidence, as discussed by Paxson [8], chapter 11.2.

If measurements are being taken at a single point, for example to measure inter-packet delays or traffic flows the timing problem is relatively straightforward. All the packets are on the same physical medium and there is no need for very accurate synchronisation of the local clock to an absolute time standard. For many applications a free-running clock oscillator with a stability of a few ppm is quite good enough. However, it is often important that the time resolution of the system be very high.

The situation is very different when transit time measurements have to be made, for example across the global Internet. Calibration or adjustment of local clocks to an absolute standard is essential, but as network delays are of the order of milliseconds the resolution of the timing process can be less good. However, the physical medium may differ at each measurement point, and this factor needs careful analysis, as is discussed in the paragraphs below.

An intermediate case is where traffic measurements are made for test purposes, such as measuring the delay and loss through routers under development in a laboratory. The requirements here are for very high time resolution, and accurately synchronised time at the various measurement points, but not necessarily to an absolute standard. One of the features of the DUCK is that it can issue a pulse each time the internal counter reaches an exact second. Thus one DUCK can synchronise itself to others with simple local cabling.

GPS time receivers provide a cheap and very accurate time standard linked to UTC. Manufacturers such as Trimble [9] claim an absolute accuracy of better than 100 nS to UTC for their specialised GPS time receivers; the main problem with GPS remains

that of finding a suitable site for the antenna. GPS signals are very weak, and the measurement technique relies on seeing several satellites simultaneously, thus the antenna need to be able to view a major portion of the sky down to perhaps 20 degrees above the horizon. This can be an unsurmountable problem for measurement points located in the basements of high-rise buildings.

CDMA [4] time receivers can provide accuracy not much worse than that of GPS, but have an offset due to the distance from the cell site. In metropolitan areas this will be only a few microseconds, but could be much greater in sparsely populated areas where the CDMA cells are much bigger [5].

D. Packet Timing on Network Links

A packet does not represent an atomic event on a network link. On all modern computer networks packets will be serialized into bit streams and transmitted in sequence. Depending on the bit-signalling rate of the physical link it will take a certain amount of time before all of the data belonging to a packet has been placed onto, or retrieved from, the link. This is also referred to as the (de)serialization delay. Below we depict a few common numbers for modern networks:

Data	Link	Serialization
64 byte packet	10BaseT	51200 nS
512 byte packet	10BaseT	410000 nS
1500 byte packet	10BaseT	1200000 nS
64 byte packet	100BaseTX	5200 nS
64 byte packet	1000BaseX	520 nS
ATM cell 53 bytes	OC3c	2720 nS
ATM cell 53 bytes	OC12c	675 nS
ATM cell 53 bytes	OC48c	168 nS

Table 1. Serialization delay for selected packet sizes and networks.

As the resolution of our timestamping engine is now less than the time it takes to transmit or receive a data packet, the question arises to which specific part of the arrival of a packet the timestamp should refer. A spontaneous response we receive frequently is: the beginning. But; not only is it hard to determine the exact start of a packet; associating the timestamp with it is also in breach of tradition; packet filters based on standard network interface cards, such as BPF, timestamp the completion (that is the end) of a packet's arrival. Some metrics even require the ability to measure both the beginning and the end of a packet arrival [6].

This debate is also influenced by practical considerations. Network measurement cards normally use regular physical layer interface chips for cost reasons. These chips are designed to interface with intelligent computer parts, such as microcontrollers and microprocessors, and have internal pipelines of considerable depth. This results in an offset error different for each link configuration (OC3c, OC12c or OC48c), which can be confusing and lead to errors in post processing.

The start of an ATM cell is a well-defined point, and cells have constant length (at least in bytes) but we also support capturing of Packet-over-SONET (PoS). The timestamp for PoS packets as generated in our systems is assigned to the first byte of the HDLC header preceding the packet data. The length of the packets is not known exactly, as the HDLC protocol allows for escaping of some control bytes and the framer does not preserve this information.

E. Packets on the Ethernet and SONET

All Ethernet frames consist of a 64-bit preamble (including, for simplicity of understanding, the start-of-frame delimiter octet), followed by 14 bytes of MAC header (with destination and source address and the 16 bit type/length field), a data portion between 46 and 1500 bytes plus a 32-bit Frame Check Sequence.

The Dag measurement cards report the packet size inclusive of the MAC header and the FCS field. For calculations on packet spacing on the Ethernet the interframe gap has to be taken into account, which sums up to a total of 38 byte times on top of the size of the data payload, or 20 bytes on top of the Dag wire length parameter.

The basic SONET STS-1 (OC1) structure consists of a frame with 9 rows and 90 columns with a total of 810 octets. This frame is transmitted every 125 microseconds, or 8000 times per second. The first 3 columns are reserved for transport overhead (TOH), the remaining 87 columns carry payload.

All OCn carriers (for n=3,12,48,..) represent multiples of the standard frame by expanding the number of columns. The time taken for transmitting a single frame is unchanged, so is the time overhead for the TOH. Beginning with OC3c carriers another overhead column is added to the SONET SPE – POH. The transport overhead inserts a fixed overhead of 463 nS into the transmission of the synchronous payload, the POH accounts for another 51 nS. If TOH and POH are aligned, they will straddle an ATM cell or data packet by a total of 514 nS. Depending on the time it takes to transmit the data (see table 1) this overhead

will add between a few dozen and several hundred percent of overhead to the serialization delay.

The term Packet-over-SONET (PoS) is applied to a family of link configurations which deploy HDLC encoding to provide for packet framing within the SONET SPE. There are three major variants of the encapsulation used to carry PoS data; CISCO HDLC, PPP-over-SONET and CISCO DPT. Our experience has been with CISCO HDLC, which appears to be the most commonly deployed standard.

All three variants of PoS carry a Frame Check Sequence (FCS). For CHDLC and PPP the FCS is configurable as a per-link option of either 16 or 32 bits. SRP uses 32-bit FCS. The Dag measurement cards deliver the decoded frame size inclusive of the FCS field as "wire length".

Aside from the straddle effect, PoS data packets are also subject to stretching by the HDLC byte stuffing procedure, which depends on the data content. Without a trace of the complete payload data it is impossible to work out the exact size of the link level frame as it appears within the SONET SPE.

The details of packet timing on SONET carriers are very involved. For a general picture of per-link performance metrics it can be useful to ignore all the special effects and work with summaries over larger intervals of time. For instance, when doing analysis on intervals of multiples of 125 μseconds the framing jitter can be smoothed.

F. DISCUSSION

The availability of precision timestamping for network packets is fairly new and it is not surprising that researchers are in a learning process to correctly interpret measurement results.

We have indicated a number of areas in which dedicated passive measurement technology can improve (or enable) different kinds network analysis.

From analysis and experience we find the solutions presented in this paper able to scale with increasing network link bandwidth, which is a dominating trend for the Internet.

G. CONCLUSION

Using hardware-based methods we have been able to improve the precision for network packet arrival timestamping by at least an order of magnitude over conventional software-based methods. With ever increasing link bandwidth and reduced packet interarrival times those technologies will gain importance for the analysis of network performance metrics.

The details provided by the new timestamping system highlight artefacts of the specific link layer technology in use at the point of observation. For network analysis, it is important to clearly distinguish those from the more general picture of Internet traffic dynamics.

The provision of global time synchronization (GPS and CDMA) for passive network measurements is an enabling technology for multipoint measurements, which will allow researchers to make the step from isolated observations towards studies of Internet behaviour on a larger scale.

H. ACKNOWLEDGEMENTS

The idea for this paper is based on discussions with Bill Cleveland and Jin Cao of Bell Labs, Murray Hill, who also provided feedback on an early draft of this paper.

We would like to thank Darryl Veitch of EMULab, University of Melbourne, Australia, for his helpful comments on the structure of this paper.

The development of the Dag network measurement cards is a result of the strong support of and collaboration with our colleagues of NLANR MOAT and CAIDA at San Diego Supercomputer Center. Funding has also been provided by the New Zealand Foundation for Research Science and Technology, and Sprint Advanced Technology Laboratories.

A portion of Jörg's time is committed to Passive Measurement and Analysis, an NLANR MOAT project under the National Science Foundation Cooperative Agreement No. ANI-9807479.

I. REFERENCES

[1] Apisdorf, Joel, Claffy, K, Thompson, Kevin, and Wilder, Rick: OC3MON - flexible, affordable, high-performance statistics collection, http://www.nlanr.net/NA/Oc3mon/

[2] Dag Project website, http://dag.cs.waikato.ac.nz/

[3] Donnelly, Stephen F., PhD thesis, work in progress, http://wand.cs.waikato.ac.nz/~sfd/

[4] EndRun Technologies Preacis Cf frequency standard, http://www.endruntechnologies.com/frequency-standard-lowcost.htm

[5] EndRun Technologies White Paper: UTC Time and Frequency Dissemination via the IS-95 CDMA Mobile Telecommunications Infrastructure, http://www.endruntechnologies.com/PTTI2000_WhitePaper.pdf

[6] IETF RFC2679, A one-way delay metric for IPPM, http://www.ietf.org/rfc/rfc2679.txt

[7] Mills, David. Network Time Protocol (Version 3), Specification, Implementation and Analysis. IEFT Request for Comments 21305, March 1992, http://www.iet.org/rfc/rfc1305.txt

[8] Paxson, Vern, PhD Thesis, 1997, ftp://ftp.ee.lbl.gov/papers/vp-thesis/dis.ps.gz

[9] Trimble Navigation, Acutime 2000 GPS Smart Antenna Users Guide, ftp://ftp.trimble.com/pub/sct/timing/acutime2000/acu2000b.pdf