Measuring packet forwarding behavior in a production network

Lars Landmark
University Graduate Center
Kjeller,Norway
larslan@unik.no

Otto Wittner
Uninett
Trondheim, Norway
otto.wittner@uninett.no

Øivind Kure
Norwegian University of Science and Technology (NTNU)
Trondheim, Norway
okure@item.ntnu.no

***Abstract*— Today routers mainly "fast switch" their traffic applying dedicated hardware. However "process switching" which involves the router OS may occur occasionally before caches in hardware modules are updated. Fast switching is preferred as it requires less time and is more deterministic than process switching. Under normal operating, the majority of all packets are fast switched. However, in the event of a miss-configuration and/or system error, routers and switches may start to process switch large portions the traffic. This may lead to higher packet jitter and even loss. This paper shows that by applying active measurement unusual process switching behavior in routers in a production network may be detected. Hence access to router internals is not required. By interpreting measurement statistics intermediate routers are found which add jitter unrelated to traffic or intermediate bottleneck links. The added jitter is caused by a large amount of process switched packets requiring longer and varying processing time. Probe packets applied were time stamped at source and destination by high precision hardware (data acquisition cards) as well as lower precision Linux OS functions. Interestingly, results based on a large number of observations show that hardware and Linux OS timestamps present similar number in terms of jitter measurements.**

Keywords- Process switching; Fast switching; passive hardware measurements; Linux measurements; Intradomain; Experimentation; Measurement; Performance.

I. INTRODUCTION

Various analysis of jitter in a packet stream has been used to infer capacity and residual capacity on bottlenecks. In this paper we investigate a particular path in an operational network, and analyse its jitter and End-To-End delay. We illustrate why the jitter analysis is not reliable as a tool for detecting bottleneck links in the Gbps range and above; The internal processes in the router may be a large source of perturbation. In our particular set of measurements, slow path processing in a router presented the largest source of perturbation. As part of the measurement work we also evaluated the difference between using specialized equipment with high degree of accuracy compared to standard LINUX tools.

Packet forwarding is often assumed to be deterministic, and that the only factor influencing inter-packet times is queuing caused by cross traffic. This assumption is in contrast to our observations, and caught our interest. Furthermore, in [8] it was shown, as in our study, that packets equally spaced in time at source becomes perturbed at the destination. The jitter distribution is concentrated in modes with a long time distance between the modes. The modes do not correspond to any queuing delay caused by cross traffic, and instead it must be caused by processes within the router itself. The forwarding process results in packets not being forwarded with a deterministic time.

In order to clarify the cause of our multi-modal jitter, we also evaluated the End-to-End (E2E) delay. The delay consists of the physical propagation time on cable (i.e. distance), the transmission time, the processing time in the routers and the queuing time. Only the two latter ones should then vary from packet to packet and thus contribute to the jitter. Observed jitter is either caused by cross traffic, or irregular packet forwarding in the router itself. In case of cross traffic, one would expect a probability distribution depending on the traffic load. In case of a repeated time process within a router, packets would be spaced in line with this time process and influence both jitter and E2E delay.

Packet forwarding within a router is traditionally divided in two forwarding paths, fast switching (fast path) and process switching (slow path). Fast switched packets benefit from parallel processing, cache lookup without the need of consulting main router tables and forwarding at interrupt level. Packets not eligible for fast switching must wait in the central processor input queue until they are scheduled for processing. Hence, process switched packets are associated with a higher router delay [17]. A packet is fast switched if a cache entry exists. A cache entry is established for the first packet, and remains in the cache until aged out.

There exist several techniques for aging cache entries; Least Recently Used (LRU) and Most Recently Used (MRU) are two common ones. In addition to aging a specific cache entry, cache entries are also randomly aged. Cisco routers for instance, randomly invalidate 1/20th of the cache each minute, or 1/5th in case of low memory [13]. As a result, not all packets within the same flow are necessarily fast switched, leading to different End To End timers and jitter.

In a stable operational network routers are intended to fast switch the majority of the traffic. Identifying misbehaving router is of interest since a consequence of process switched packets is reduced throughput and increased jitter. High jitter is often associated with application problems. However, irregular packet forwarding, or high a process switching ratio, may also lead to packet loss at the destination. This problem increases with the difference in intermediate link capacity. Take one example: a packet train leaves a 1Gbps link and is further forwarded over a 10Gbps link or a higher core network link. In case of packets being process switched at a 10Gbps router, the packets are queued until a cache is installed. When a relevant cache entry is installed, packets leave back-to-back out onto the next link. When such a 10Gbps train of packets later is to be forwarded onto a link with lower link speed and a narrower link buffer, packets may be lost.

Understanding and controlling network resources is therefore important, and often referred to as network resource management. Packet dispersion, E2E delay and other techniques are often used for inferring network characteristics, such as residual path capacity and link bottlenecks. Both E2E delay and packet dispersion are based on assumptions that not always hold true. As an example, packet dispersions used to infer a path link bottleneck assume that packets are deterministically forwarded by all routers. When this is not true, it is important to detect errors to further take action on the observations. This is especially important in high capacity networks, as internal router timers overshadow link capacity timers.

The rest of this paper is organized as follows. Section 2 and 3 outline our measurement technique and test bed. Section 4 describes and discusses our hardware acquisition card measurement results. Section 5 presents observed difference for Linux and hardware card results. The paper concludes with a summary in Section 6.

II. MEASUREMENT TECHNIQUE

Our observation set covers measurements over three days, and consist of approximately 26 million observations. The set holds observations for multicast and unicast packets, hence enabling investigations of difference between multicast and unicast forwarding. Multicast and unicast UDP packets were sent with a constant packet inter departure time at 10 milliseconds using Rude & Crude packet generators [9].

Due to process scheduling inaccuracies in end systems, packets were not scheduled for transmission with an exact 10 milliseconds time interval. This inaccuracy was however eliminated in the end calculation.

For our jitter measurements, we used a technique similar to the packet dispersion technique in [4]. Packet dispersion is widely used to infer bottleneck, residual bandwidth etc. The principle idea behind is to measure the time between two consecutive packets at the receiver when transmitted back-to-back at the source. In our case, packets

$$\text{Jitter} = (Tdst_i - Tdst_{i-1}) - (Tsrc_i - Tsrc_{i-1}) \quad (1)$$

were not sent back to back, but spaced by 10 milliseconds. Timestamps were acquired at both source and destination. The network introduced jitter was calculated as shown in equation 1. $Tdst_i$ and $Tdst_{i-1}$ is packet i's and i-1's arrival times at the destination respectively. Similar goes for source src. If three consecutive packets experience equal processing and queuing delay, the jitter will be zero. On the other hand, if a single packet is delayed, time is added between the preceding packet and the delayed packet, while time is reduced equally much between the delayed packet and the succeeding packet. The added and reduced delay does not need to happen in sequence, as every packet is dependent on every previous and later packet. Consequently, the sum of all jitter samples is zero.

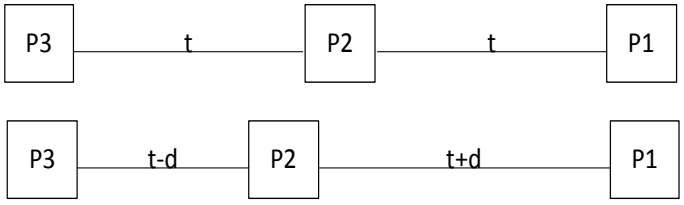

Figure 1: Network added jitter. Packet P2 is delayed causing added time d between P1 and P2 and accordingly reduced time d between P2 and P3.

Figure 1 exemplifies the argument for three packets, where the first and last packet is not changed.

A packets E2E delay from source to destination consists of a sum of components from along the path:

1. *Transmission time*, i.e required time to serialize all bits of a packet on the cable.
2. *Propagation delay*, i.e time for transmitting a bit over the transmission line. Propagation velocity over an optical fiber is 2/3 of the speed of light.
3. *Processing time*, i.e required time a router need for transmitting a packet from the incoming interface to the associated outgoing interface queue.
4. *Queuing time*, i.e. duration a packet waits for other packets to be dispatched at the cable before itself being dispatched.

For a given packet size both transmission time and propagation delay would remain the same, and add a constant time to each packet. Processing delay and queuing delay are then the only cause to the variance in observed jitter and E2E delay. As a consequence, the window of variance should remain similar for E2E delay and jitter.

III. TEST-BED

The measurements were performed over UNINETT, the Norwegian national research and educational network (NREN) which connects universities, colleges and research institutions to the Internet. The UNINETT core interconnects the main Norwegian cities with 10 and 2.5 gigabit per second (Gbps) links in ring structures. Capacity on access links to institutions varies from 1 Gbps and upwards. Access links in our test case were 1 Gbps. The cable distance may roughly be approximated by the road distance as physical cables are mainly following

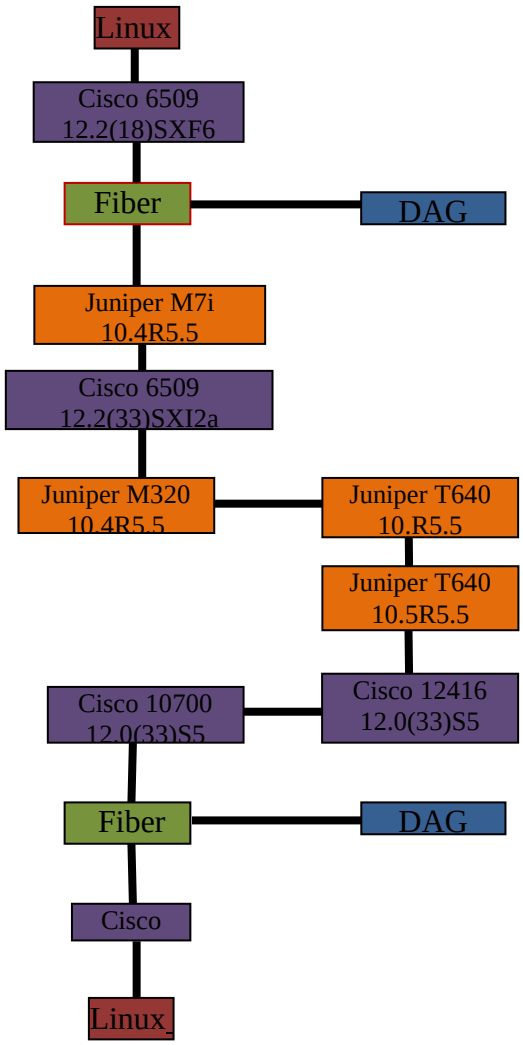

Figure 2: Routers in our measured path

the roads, railroad and power cables. A 932 km road distance implies a propagation delay 5.66 milliseconds in our measurements. The probe packets, as seen from the GPS synchronized acquisition cards (DAG) applied, were visiting 7 routers along the path (i.e Time To Live was reduced by 7) and an unknown number of switches. The measured path consists of routers from Cisco and Juniper using different IOS as shown in figure 2.

IV. RESULTS

Measurements reported in this section are based on multicast and unicast probe traffic captured by the DAG cards. Probability distributions are estimated by the use of histograms with a bin width of 5 microseconds.

A. End-To-End Delay

Figure 3 to 4 show the E2E delay distribution. The window of E2E variance is approximately 200 micro seconds. The E2E shape, in figure 3 and 4, shows dissimilarities for the two path direction. The dissimilarity is assumed caused by different traffic load on the two path directions.

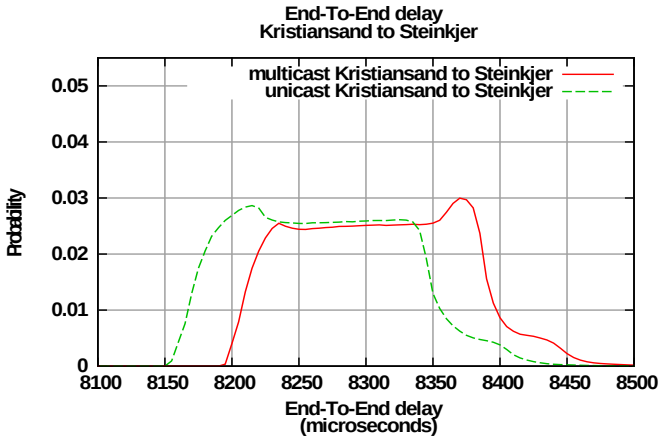

Figure 3: End To End delay for multicast and unicast from Kristiansand to Steinkjer.

In [6], it is shown that paths with low traffic load show a gamma probability distribution, medium traffic show generalized Pareto while high traffic load show Lognormal. Our E2E delay measurements do not correspond to any of these, but are more similar to a uniform distribution, especially from Kristiansand to Steinkjer as illustrated in figure 3 compared to figure 4. This indicates that the E2E variance is more related to packet processing within the routers themselves than traffic queuing.

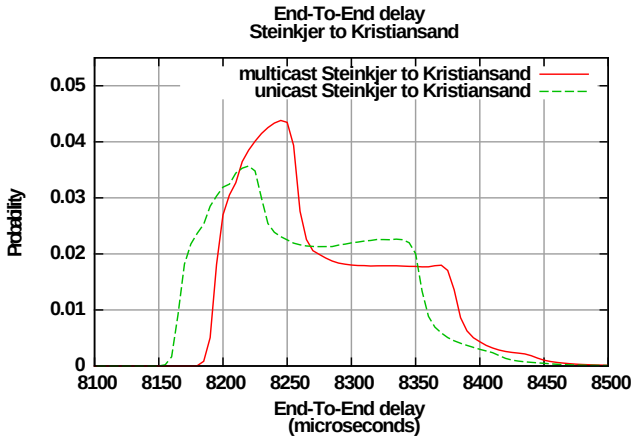

Figure 4: End To End delay for multicast and unicast from Steinkjer to Kristiansand

B. JITTER

Figure 5 and 6 display the jitter between two consecutive packets. Our observation shows a multi-modal jitter distribution. The jitter observations appear within a similar window of variance as observed for E2E delay, i.e. 200 microseconds.

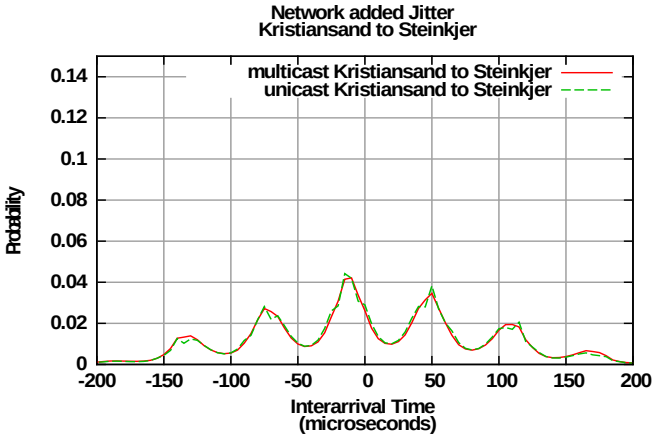

Figure 5: Network added jitter for unicast and multicast from Kristiansand to Steinkjer.

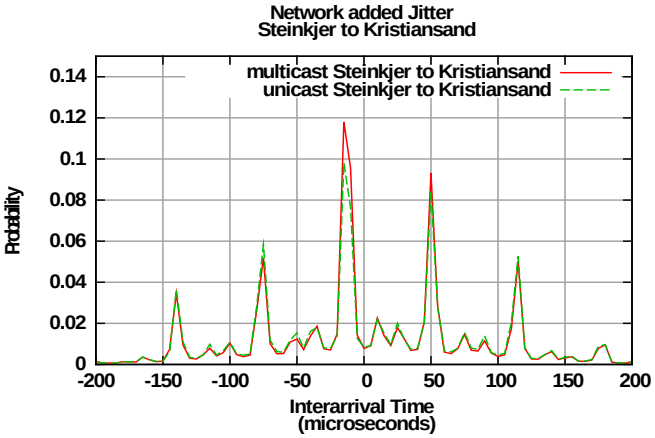

Figure 6: Network added jitter for unicast and multicast from Steinkjer to Kristiansand.

Each mode is spaced by approximately 60 microseconds in both path directions. Such repeated jitter may indicate a repeated time process in the network. Multi-mode patterns resembling those in Figure 5 and 6 is by earlier work [5] explained to be caused by multiple link bottlenecks along a path. The link bottlenecks were inferred by use of packet dispersion and cross traffic. If cross traffic intertwine between two packets, and the new inter-packet time is found to be a multiply of a given packet size on a known link, then the path bottleneck link capacity can be inferred. The distance between the jitter modes was further used to infer additional but higher capacity bottlenecks. In our observations the distance between the modes is 60 microseconds which would indicate a 200Mbps link. However, the lowest capacity on our path was 1Gbps. In [8] jitter with multiple modes similar to our observations are shown. The authors conclude that the number of modes is a result of inter-departure packet time at the source. Fewer modes are observed with increased packet ratio from source. In our test-bed we only have one inter-departure packet rate, and similar to the work in [8] have only one set of modes. Based on the discussion above we can only conclude that the observed jitter is introduced by the intermediate routers themselves. The regular modes are most likely caused by a router internal process, and probably by process switching of packets. To evaluate our conclusion statistics from all routers along the path were collected. They did indeed confirm that the (old and end-of-life) cisco 10700 router did process switch the majority of its traffic.

Fast switching and process switching differ in their packet forwarding time. For instance, a cache miss results in process switching with an added time compared to fast switching. In our case, at most four modes are observed with a constant 60 micro-seconds time difference. The modes do therefore represent the probability of being process switched. The main mode implies fast switching, while the remaining modes imply process switching. The second mode represents packets being process switched one time. Third mode represents packets waiting in queue for process switching within the same router.

In case of dissimilar routers performing process switching, we would expect to observe dissimilar timers. Steinkjer to Kristiansand shows two additional modes between the main modes illustrating potentially additional routers and the associated process switching time. The additional modes are spaced by a constant, but a more narrow time, presenting a second, but faster process switching router.

According to UNINETT, the Cisco 10700 router was in its end of life, and was in the process of being changed to Juniper MX80. After changing the router, we collected a new measurement set. The new measurement set was collected via acquisition software in Linux only, hence with reduced timing accuracy. To understand the level of this inaccuracy we studied measurements collected with Linux software in parallel with DAG hardware during the first measurement period (before exchanging Cisco 10700 to Juniper MX80). The results are further described in section 5, but the main observation is that Linux and DAG show equal distributions for jitter. Hence the second data set has value. We therefore did additional jitter-analysis based on the Linux software acquired measurement.

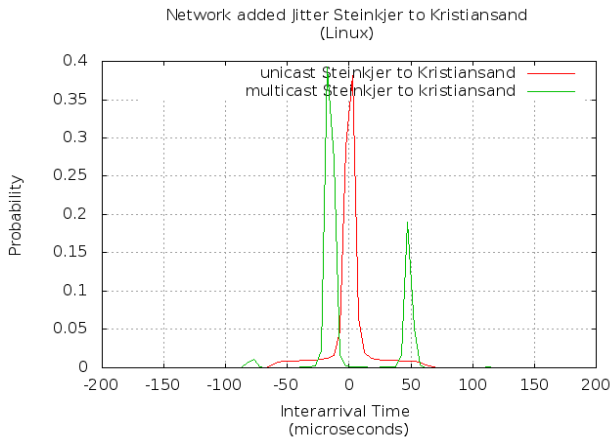

Figure 7: Network added jitter for unicast and multicast after changing the flawed router

Figure 7 and 8 show these jitter distribution in both directions for multicast and unicast. It is clear now, especially for unicast, that packets are mainly fast forwarded. Multicast, on the other hand has a slightly higher chance of being process switched than unicast, which comply with more protocol logic, and/or that their cache entry is more likely exchanged with unicast.

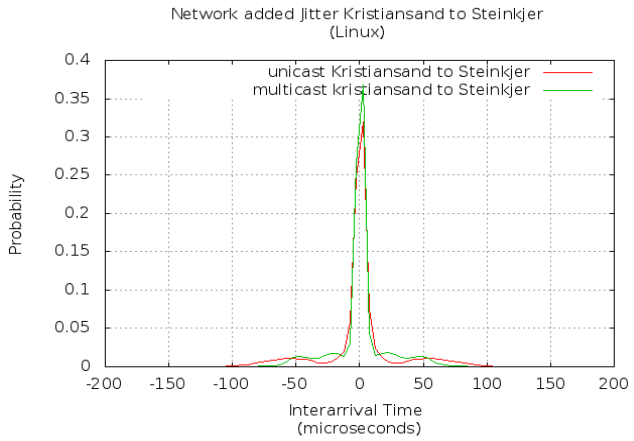

Figure 8: Network added jitter for unicast and multicast after changing the flawed router

V. HARDWARE V.S. SOFTWARE ACQUSITION

During the first measurement period we captured packets both with hardware DAG cards and via software tools (tcpdump and crude) in Linux. As Linux applies system timers when timestamping packets it was assumed that Linux would provide higher jitter variability than DAG.

Figure 9 shows the probability distribution for jitter acquired from a DAG card and from the Linux application layer. Previous work [6], using E2E delay, have shown large difference in DAG and Linux measurements. In our work however, when focusing on jitter, significant differences are no longer observable in the jitter distributions.

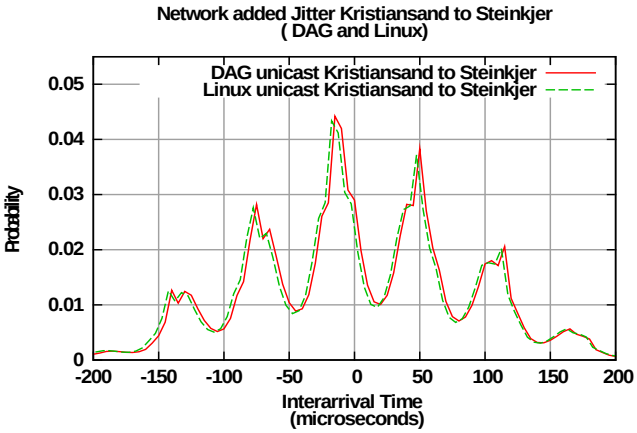

Figure 9: Network added Jitter for DAG and Linux

Hence investments in high precision hardware for jitter observations seem unnecessary if the number of observations is ensure to be large enough.

VI. CONCULSION

In this paper we have shown that by applying commonly available measurement techniques and analysis methods, we are able to infer flawed routers with suboptimal packet forwarding along a path. Such flawed routers are not necessarily detected by ordinary network management routines since no alarms are generated even though packets may be lost.

Our multi-mode jitter observation turns out to reveal the difference between fast and process switched packets. The main mode presents fast switched packets, while additional modes show the probability of being process switched. The time distance between the modes represent the processing time for process switching.

Two measurement methods were also explored, software acquisition at the Linux application layer and acquisition with high precision hardware (DAG cards). DAG and Linux delivered similar probability distributions for jitter measurements, suggesting that low cost software tools may be sufficient in this context.

Future work includes repeated measurements along multiple paths in UNINETT to enable better pinpointing of suboptimal network components. High accuracy inter-domain measurements between Norway and New Zealand are also in progress.

ACKNOWLEDGMENT

Our thanks go to Olav Kvittem, Arne Oslebo, Trond Skjesol and Jon K Hellan for facilitating the experiments at UNINETT, and for our valuable discussions.

REFERENCES

[1] Fast-Path Multicast Forwarding on Cisco 12000 Series Engine 2 and ISE Line Cards, available: http://www.cisco.com/en/US/docs/ios/12_0s/feature/guide/mcast.html, cited January 04 2012

[2] D. Estrin, D. Farinacci, A. Helmy, D. Thaler, S. Deering, M. Handley, V. Jacobsen, C. Liu, P. Sharma, L. Wei, "Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification", RFC 2117

[3] E. Kohler , R. Morris , B. Chen , J. Jannotti , M. F. Kaashoek, "The click modular router", ACM Transactions on Computer Systems (TOCS), v.18 n.3, p.263-297, Aug. 2000

[4] C. Dovrolis, P. Ramanathan, D. Moore, "Packet dispersion techniques and capacity estimation methodology", In IEEE/ACM Transactions on Networking, December 2004

[5] S.katti, D. Katabi, C. Blake, E. Kohler, J. Strauss, "MULTIQ: Automated Detection of Multiple Bottleneck Capacities Along a Path", IMC04, October 25-27, 2004, Taormina, Sicily, Italy

[6] A. Hernandez, E. Magana, "One-way Delay Measurement and Characterization" In-ternational Conference on Networking and Services (ICNS '07) p. 114, 2007

[7] Endace DAG cards, available: http://www.endace.com/endace-dag-high-speed-packet-capture-cards.html, cited januar 04 2012

[8] D. A. Freedman, T. Marian, J. H. Lee, K. Birman, H. Weatherspoon, and C. Xu. " Exact temporal characterization of 10 Gbps optical wide-area network", Appears in Proceedings of the 10th ACM SIGCOMM Internet Measurement Conference (IMC), November 2010, Melbourne, Australia

[9] Rude & Crude packet generator, available: http://rude.sourceforge.net/, cited January 04 2012

[10] D. L. Mills. The nanokernel. In Proc. Precision Time and Time Interval (PTTI) Applications and Planning Meeting, pages 423-430, November 2000.

[11] D. Mills, U. Delaware, J.Martin, J. Burbank, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905

[12] G. Jin, B. L. Tierney," System Capability Effects on Algorithms for Network Bandwidth Measurement", IMC03, October 27-29 2003, Miami Beach, Florida, USA

[13] "How to Choose the Best Router Switching Path for Your Network", available http://www.cisco.com/en/US/tech/tk827/tk831/technologies_white_paper09186a00800a62d9.shtml, cited januar 2012

[14] N. Hohn, K. Papagiannaki, and D. Veitch, "Capturing Router Congestion and Delay", IEEE ACM Transaction On Network. 17 (2009), 789–802.

[15] K. Papagiannaki, S. Moon, C. Fraleigh, P. Thiran, and C. Diot, "Measurement and analysis of single-hop delay on an IP backbone network," IEEE Journal on Selected Areas in Communications, vol. 21, no. 6, Aug. 2003.

[16] Kim, C., Caesar, M., Gerber, A., and Rexford, J. Revisiting route caching: The world should be flat. In PAM (2009).

[17] Hohn, N., Papagiannaki, K., and Veitch, D."Capturing Router Congestion and Delay" IEEE ACM T. Network. 17 (2009), 789-802.