

**CESNET Technical Report 13/2009**

**Precise Timestamp Generation Module and its  
Applications in Flow Monitoring**

TOMÁŠ MARTÍNEK, MARTIN ŽÁDNÍK

Received 11.12.2009

**Abstract**

Precise timestamps assigned to individual packets play an important role for network traffic analysis and measurement of network infrastructure. Moreover, connection of precise timestamps with flow based analysis, allow us to measure quality of end to end and other QoS-oriented applications. This technical report describes a hardware module for precise timestamp generation dedicated for netflow monitoring probe FlowMon. It shows module hardware architecture, measurement of timestamp accuracy and discussion about possible use cases in flow based applications.

*Keywords:* precise timestamps, FlowMon, network applications, FPGA, VHDL

## **1 Introduction**

Development of measurement and monitoring of network traffic is driven, among others, by an increasing demand for quality of service provided by the network. Loss-free, low delay with low jitter links are common denominators of real-time applications such as video and audio communication, on-line gaming, streaming and others. Unlike bandwidth, temporal parameters of network connections is not that easy to improve due to factors such as traffic mix (competing applications) and traffic dynamics (e.g., peaks, alternative routes). Long-term and ongoing measurements are necessary to reveal bottlenecks and scope for optimizations. Since a flow measurement is an established way of observing traffic behavior, its extension with measurement of temporal flow characteristics is only natural, for example see RFC on IPFIX [6]. Gathered flow data may serve as an input of end to end delay analysis and other QoS-oriented applications.

The major challenge is to measure temporal characteristics of high-speed network traffic stably (with low error variance) and accurately (with low error). The reason is that a conventional PC architecture, which are many measurements points based on, cannot assign incoming packets precise timestamp due to several facts:

1. Resolution of software timestamp is in order of microseconds
2. Common crystal oscillator used in PC is not very accurate (about 30 ppm error)
3. A packet seen at the software layer was previously stored in several buffers and its incoming time at the network interface differs significantly packet by packet from its arrival time to software.

4. Assigning each incoming packet a timestamp requires an indispensable processor time which might become the bottleneck of high-speed network traffic measurement.

To address the issue of precise packet timestamping, we propose a hardware timestamp generation module capable of assigning a precise timestamp to each incoming packet immediately at the network interface. Moreover, the module might be connected to a GPS unit, in which case, generated timestamps are also very accurate. To address high-speed traffic measurement, the precise timestamp generation module is instantiated in the flow measurement pipeline of Flexible FlowMon probe [5]. The probe is intended for flow measurement of high speed backbone links and, with its timestamp extension, we envision analysis of reported data to support several applications such as one-way delay and jitter measurement, response time measurement and detection of stepping stones.

Flexible FlowMon probe is based on a commodity PC running Linux OS with PCI-Express x8 network acceleration card COMBOv2<sup>1</sup>. The measurement process with the precise timestamp generation module is implemented in the acceleration card as a firmware for Field Programmable Gate Array (FPGA). The host PC serves as an exporter with capability to perform further aggregation upon reported flows.

The rest of this technical report is organized as follows: second chapter describes hardware support for precise timestamp generation, next chapter provides experimental results on timestamp accuracy, the report concludes with summary and use cases of precise timestamps in flow measurement applications.

## 2 Hardware support for precise timestamp generation

### 2.1 System Architecture

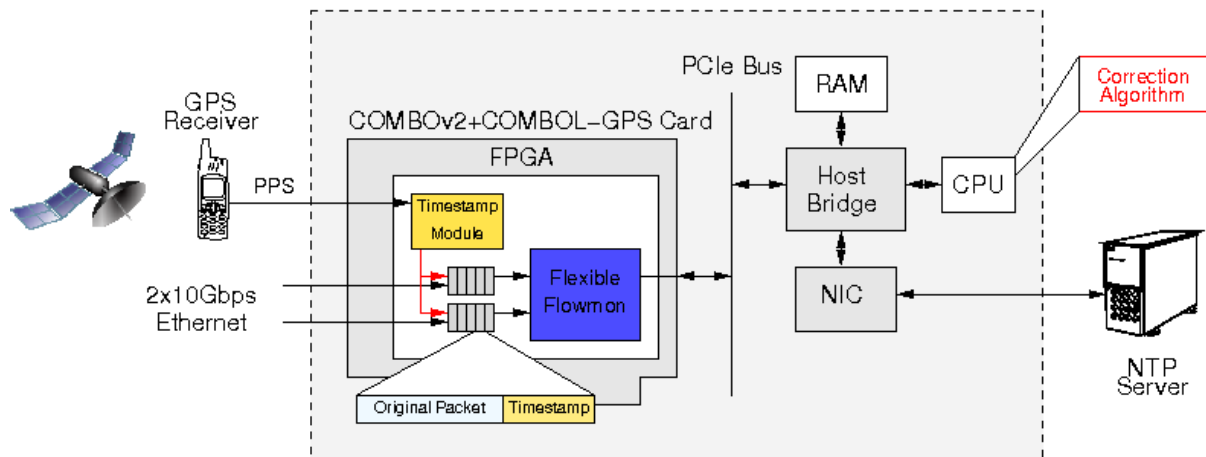
The detailed flow based analysis of network traffic requires a precise timestamp assignment for every incoming packet. It is important for inter-packet gap measurement as well as for global statistics. For these purposes, an extension module for timestamp generation (TSU) is placed directly in the input part of the whole system. As soon as a packet arrives, TSU is asked for timestamp, which is attached to the packet in the form of a specific header (see Figure 1). Afterwards, the packet with timestamp continues to TSU is asked for timestamp which is attached to the packet in form of specific header (see Figure 1). Afterwards, the packet with timestamp continues to the Flexible Flowmon application which evaluates inter-packet gaps and other required statistics.

In order to generate precise timestamps, TSU module uses synchronization pulses from GPS receiver. These pulses are generated every second (PPS - Pulse Per Second) with accuracy defined by GPS vendor. TSU module is responsible for generation of timestamps between two PPS pulses. For more information about precise timestamp generation see chapter Section 2.2.

For correct function, it is necessary to initialize timestamp in TSU module be-

---

<sup>1</sup> <http://www.liberouter.org/hardware.php?flag=2>



**Figure 1.** A system architecture of TSU module connected with flexible flowmon application.

fore its computation starts. Please note, that the initialization of timestamp in order of seconds is sufficient because the TSU module is responsible for generation inside a second. This initialization can be performed using system time or some of more precise sources for example NTP server, atom clocks etc. For more information about initialization process see [4].

Due to inaccuracy in physical clock crystals and its temperature dependency, errors (offsets against a true time) occur during the timestamp generation process. Therefore, generator has to be corrected continually. As the initialization and correction processes are not computation intensive and time sensitive they can running at host CPU without any effect to overall timestamp accuracy.

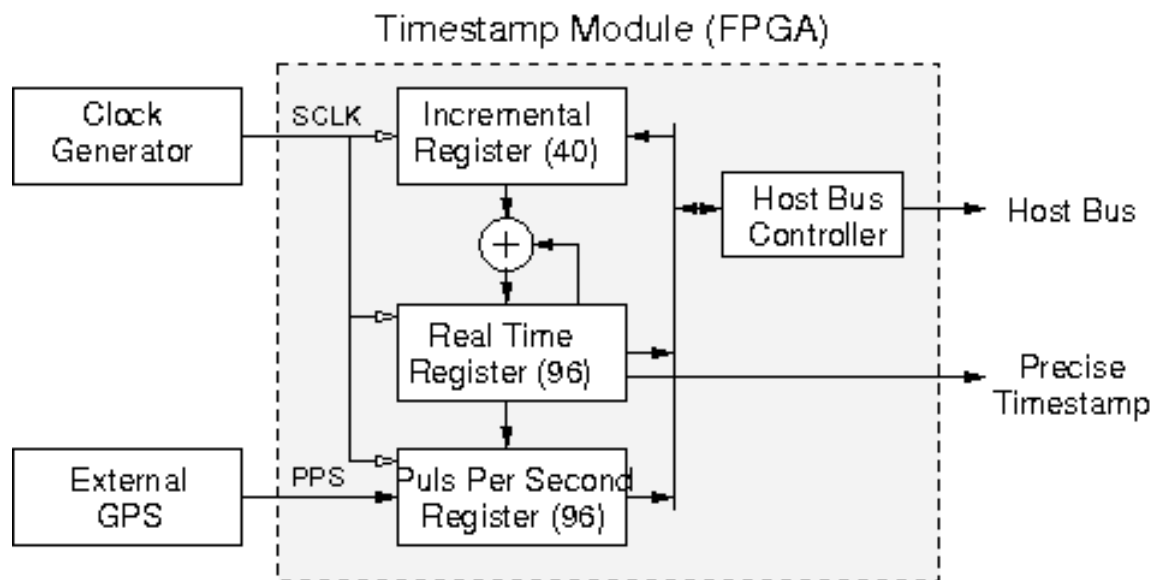
In comparison to approach published in [4] the presented approach was simplified significantly at the level of TSU module as well as the architecture of extension card for GPS signal processing. Instead of PTM card connected to PCI bus, a novel and simpler COMBOL-GPS module was developed (see Figure 2). In COMBOL-GPS card, there is no need for FPGA chip because the PPS signal is propagated down to the COMBOv2 mother card using LSC connector. Similarly, no microcontroller MSP430 is present on COMBOL-GPS for eventual adoption of initialization and correction processes. These can be handled using some of soft-core processors (e.g. MicroBlaze) implemented directly inside COMBOv2 FPGA chip. The COMBOL-GPS contains only precise crystal (1 ppm), clock synthesizer for clock customization and necessary circuits.

## 2.2 TSU Module Architecture

A module architecture for precise timestamp generation is based on the approach described in [4]. The main task of the module is to generate timestamps between synchronization pulses (PPS) generated by GPS receiver. The actual timestamp is stored in *Real Time Register (RTR)*. The timestamp is represented using 96 bits where the first 32 bits represent number of seconds since 1.1.1970 00:00:00 UTC and the remaining 64 bits represent fragment of second. In each tick of SCLK clock signal the value of RTR register is incremented by a constant stored in *Incremental Register (INC\_REG)*.



**Figure 2.** COMBOL-GPS extension card



**Figure 3.** A hardware architecture of precise timestamps generator

The incremental constant should be selected such that the value of RTR register is incremented by one second (i.e. the first 32 bits are incremented by one and lower 64 bits are as close to zero as possible) after receiving next PPS pulse. Unfortunately, this is not always possible because of GPS PPS inaccuracy, crystal inaccuracy and its temperature dependency. Therefore, the incremental constant has to be corrected continuously.

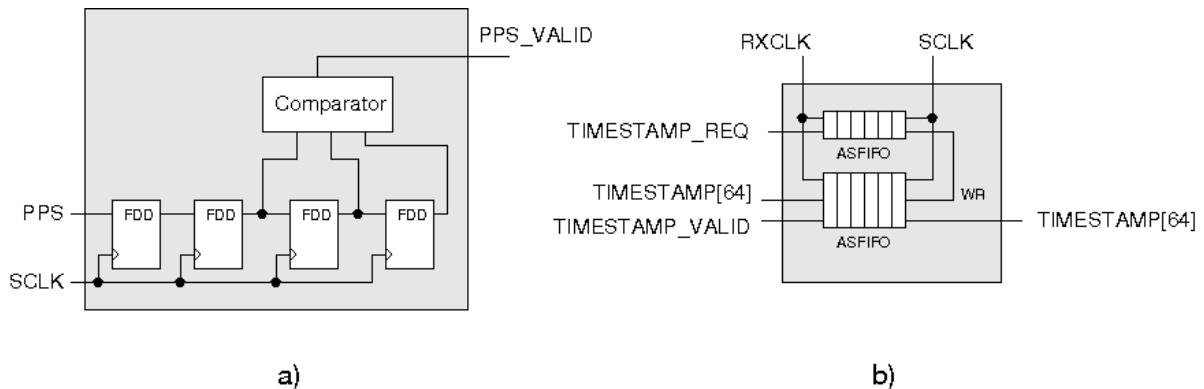
The correction process is based on a feedback control loop. As soon as the PPS signal is generated by GPS, the content of RTR register is saved into auxiliary PPS register. The correction algorithm reads this register every second and calculates

the difference against the required value. Subsequently, this difference is projected into the incremental constant such that the next PPS values should be as close to zero as possible. For more information about correction algorithm see [4].

### 2.3 Asynchronous Inputs/Outputs

For correct TSU module connections with the Flowmon application it is necessary to solve problems with asynchronous inputs/outputs. Two such places exist in the design: 1) Sampling of PPS signal from GPS receiver and 2) Transfer of timestamp between TSU module and an application network interface.

ad 1) The rising edge of PPS signal has to be detected by TSU module correctly avoiding possible glimmers and metastable states. An example of such circuit is shown in Figure 4 a). The PPS signal is resampled to the SCLK clock using pipeline of 4 flip-flops. This circuit considers PPS signal valid only if all 3 last registers contain the same value. Resampling process injects a delay which can be as large as single period of SCLK clock signal. However this disruption dissolves in the whole PPS period (i.e. one second).



**Figure 4.** Sampling circuits: a) PPS signal sampling circuit, b) Timestamp sampling circuit

ad 2) Input packets arrive into the system at frequency corresponding to the input interface (125MHz for 1Gbps Ethernet, 156MHz for 10Gbps). This clock is derived from network interface directly and can differ for each input network interface in the system (it depends on the source which generated a packet). The precise timestamp generated by TSU module at SCLK have to be resampled to the frequency of input network interface. An example of such circuit is shown in Figure 4 b). It is based on two asynchronous fifos where the first one stores requirements for timestamp and the second one contains an appropriate timestamps. Similarly to PPS sampling circuits, the disruption of stability occurring during resampling process corresponds to one period of SCK signal. Please note, this disruption arises with every incoming packet.

### 3 Experimental Evaluation and Results

The most important parameter of the whole system is its accuracy in which the timestamp is assigned to every incoming packet. Lets suppose the system for Net-flow statistics measurement on two 10Gbps ports shown in Figure 1. It is based on COMBOv2 card, where TSU module is implemented together with Flexible FlowMon application. As GPS receiver, we use Garmin 18 LVC, with external antenna connected using coaxial cable. TSU module operates at SCLK frequency 156.25MHz generated by crystal Connor Winfield P142-156.25MHz with accuracy  $\pm 30$  ppm and temperature stability  $\pm 20$  ppm.

*An accuracy of the whole system depends on all elements (namely):*

- Accuracy of PPS signal generated by GPS
- Accuracy of PPS signal resampling
- Accuracy of timestamp generation
- Accuracy of timestamp resampling to the network interface frequency

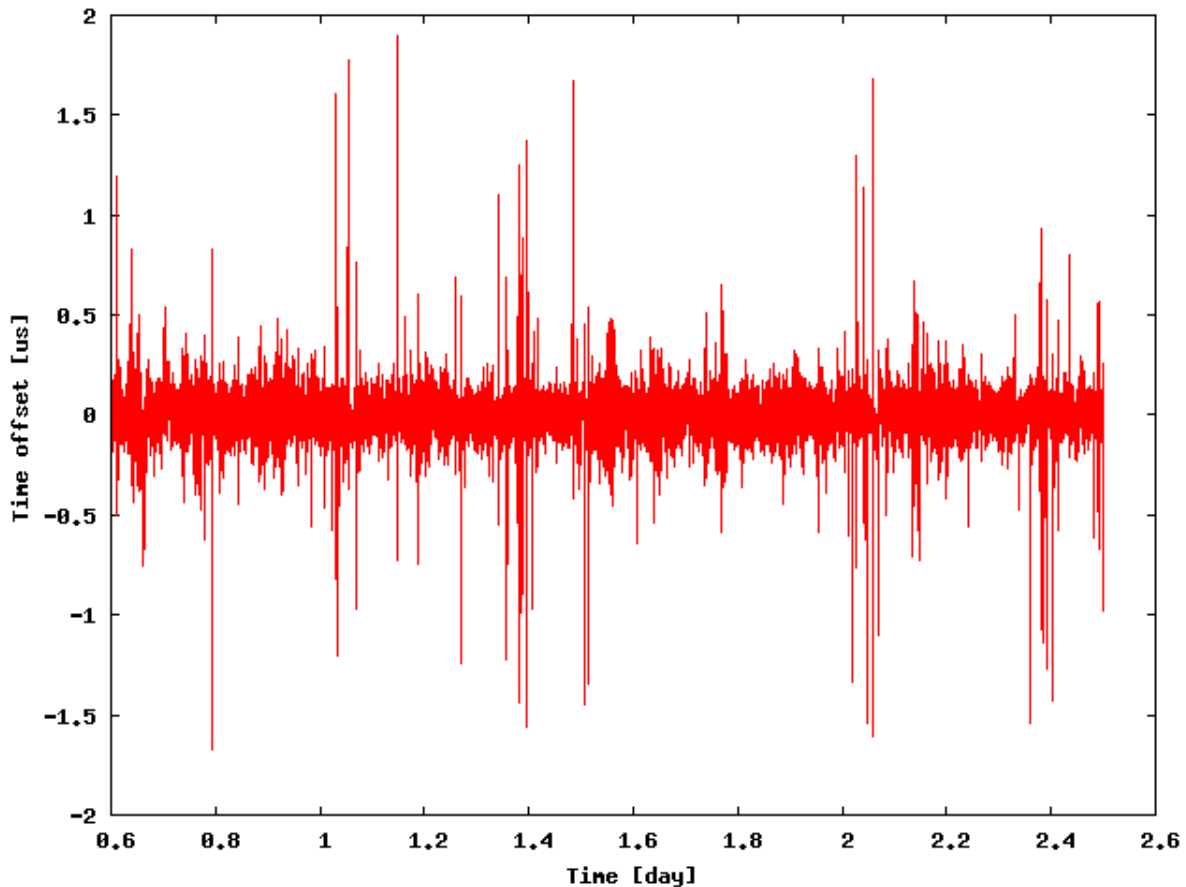
In the following part, we discuss impact of individual elements on accuracy of the whole system.

The uncertainty of PPS signal generation is primarily dependent on used GPS receiver. It is defined by vendor and it is usually in the range of  $\pm 1$   $\mu$ s. On the other hand, GPS receivers designated as time source achieve lower uncertainty in the range of  $\pm 50$  ns. Moreover, a very specific GPS calibration devices can achieve uncertainty up to  $\pm 1$  ns. Please note that this does not represent cumulative error but rather an offset from true time, which is generated in GPS control center using Celsius clocks with extreme accuracy and stability. The GPS receiver Garmin 18 LVC used in our test generates PPS signal with inaccuracy  $\pm 1$   $\mu$ s.

The PPS signal from GPS has to be resampled at the input of the TSU module. An example of resampling and detection circuit is described in chapter Section 2.3. Based on the circuit function, it was derived that the time offset can achieve up to SCLK period length, in our case:  $1/156.25 \times 10^6$  s = 6.4 ns.

The main objective of TSU module is to generate timestamps within a second (between two PPS pulses). Accuracy of this module is dependent on accuracy of SCLK clock, its temperature stability, incremental constant resolution and correction mechanism. As it is not easy to evaluate the impact of all factors, therefore we decided to measure TSU module accuracy experimentally. Our measurement operates in following steps: 1) PPS register is read every second and 2) Time offset within the second (lower 32bits) represents deviation from the PPS signal generated by GPS. The results of 2 days measurement are shown in Figure 5. The time offsets usually fluctuated in the range  $\pm 0.2$   $\mu$ s. In specific moments, peaks up to  $\pm 1.5$   $\mu$ s occurred. Unfortunately, this measurement is also influenced by PPS signal sampling and its inaccuracy, which can be up to  $\pm 1$   $\mu$ s.

The last stage is resampling of timestamp from SCLK clock domain to the network interface clock domain. As derived in chapter Section 2.3, the time offset can achieve up to SCLK period length, in our case:  $1/156.25 \times 10^6$  s = 6.4 ns.



**Figure 5.** Time offset of TSU module disciplined by GPS receiver

Finally, we can summarize achieved results. If we are interested in absolute accuracy, the offset of timestamp from true time corresponds to sum of GPS receiver offset, offsets of PPS signal and timestamp resampling circuits and TSU module. As the offsets of resampling circuits are negligible, the resulting offset against the true time is roughly  $\pm 2.7 \mu\text{s}$ .

For purposes of inter-packet gap measurement the relative accuracy is more important than absolute one. We measured the maximal deviation between two consecutive time offsets (two PPS signals) and it was  $1.9 \mu\text{s}$ . If we add GPS receiver offset, then the relative inaccuracy is 2.9 ppm however  $2.9 \mu\text{s}$  at most.

## 4 Conclusions

In this work, we implemented a module for precise timestamp generation in FPGA chip. Measurement of the generator shown that absolute inaccuracy from true time is in the range of  $\pm 2.7 \mu\text{s}$  which is sufficient for most network applications. Relative inaccuracy of generator reaches 2.9 ppm (max.  $2.9 \mu\text{s}$ ) and is much better than using only common crystal without any correction (30 ppm).

In our future work we will concentrate to increasing accuracy. Especially using GPS receiver with more accurate PPS pulse generation (e.g.  $\pm 50 \text{ ns}$ ). Then, we will connect more precise crystal (1 ppm) as SCLK clock generator and we will try to improve correction algorithm.

## 5 Discussion

The ultimate goal of flow measurement is to provide detailed information about traffic while maintaining abstraction at the flow level. Therefore a flow record is being extended with new statistical variables (defined by IPFIX) which either describe packets' property or flow temporal characteristics. A concept of Flexible FlowMon [5] based on COMBO boards has founded a flow measurement platform capable of accommodating various flow characteristics to support broader variety of applications. Incorporating precise packet timestamping improves the quality of measured temporal characteristics and opens flow measurement to a new set of applications. Several examples are discussed in the light of flexible flow measurement combined with precise timestamping.

One-way delay and jitter measurement is a first application that has been so far a domain of packet capture rather than a flow measurement. The objective of one-way delay measurement is to optimize routing, prioritization, buffering, and other parameters by measuring the delay between two nodes in the network. A common practice is to generate a specialized packet that is being captured, timestamped, and transmitted again at points of interest while traversing the network. An alternative approach may be build upon a set of flow measurement probes deployed throughout the network, passively gathering temporal characteristics on all flows of our interest. Such a measurement is non-invasive (no extra packet have to be launched in the network) and at the same time performed upon multiple hosts and applications using real network traffic.

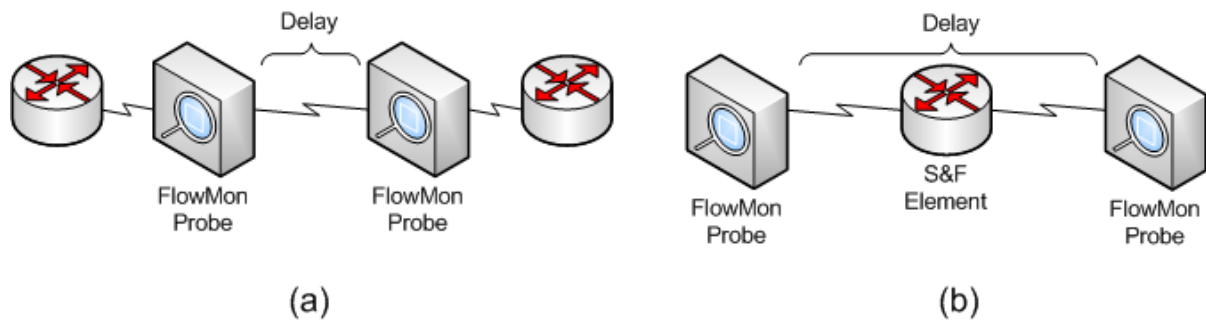
In order to support this type of measurement Flexible FlowMon offers gathering of first order statistical variables on temporal characteristics as well as storing timestamp of selected packets. A typical flow record for this type of measurement may be composed of start and end of flow timestamps and several raw timestamps of selected packets from within the flow.

The ability to store timestamps of mid-flow packets is convenient to overcome a starting period of a flow during which the delay might be longer than for the rest of the flow. Longer delay of the first packet in the flow is a side effect of path establishment (for example an OpenFlow switch has to query a routing server) or caching. Following example shows this phenomenon on round trip time reported by ping command:

```
PING XXX (XXX): 56 data bytes
64 bytes from XXX: icmp_seq=0 ttl=109 time=188 ms
64 bytes from XXX: icmp_seq=1 ttl=109 time=184 ms
64 bytes from XXX: icmp_seq=2 ttl=109 time=180 ms
64 bytes from XXX: icmp_seq=3 ttl=109 time=178 ms
64 bytes from XXX: icmp_seq=4 ttl=109 time=177 ms
```

Precise timestamping allows for correlation of gathered data, i.e., revealing a flow trajectory as well as calculation of contributing delays caused by each measured segment. Flow data correlation requires accurate timestamp from global point of view otherwise a correlation of data from adjacent nodes would be prone to errors, caused by rearrangement of timestamps and packets seen. This means that if a node receives a packet prior to another node its timestamp must precede a timestamp

assigned by a subsequent node. To avoid these errors a sufficient distance (in the case of directly connected probes, Figure 6 (a)) and/or delay (caused by buffering, Figure 6 (b)) has to be met. Flexible FlowMon with precise timestamping reduces a required delay to  $5.5 \mu\text{s}$ , which is equivalent to 1.1 km of cable (given the signal propagation speed of 200000 km/s).



**Figure 6.** Two alternatives of FlowMon probes deployment

Further, flexibility and precise timestamping can be used to measure response time of application servers using real traffic, i.e., real queries and real workload. An ongoing measurement accompanied with automated analysis system improve allocation of computational resources per application, warn administrator during flash crowds events or analyze various incidents, e.g., DoS, stalled queries, service failures. The challenge is to correctly pair query and response packets. Pairing first synchronization packets is easy but does not tell much about the response of a service since TCP establishment process is handled purely by operating system. A pair can be established only for certain packets using sequence and acknowledgment numbers. Subsequently it is possible to compute first order statistics on a response times.

Flexible FlowMon allows to setup its metering process to support response time measurement by enabling aggregation of both direction into a single flow record. By correctly defining triggers and operations per each field of a flow record a correct pair can be established and statistics computed. Due to precise timestamp module the precision of statistics is sufficient enough to measure even short response times which may be as short as tens of microseconds.

Detection of stepping stones [2], [1] is another exemplary use of precise timestamping combined with flexible flow measurement. The goal is to reveal hosts in a monitored network that have been compromised and are used by attackers as mediators – stepping stones – to achieve anonymity so that their final activities are hardly traced back. In both cited publications detection of stepping stones is based on the measurement of several flow characteristics – packet or temporal. Combination of both groups is descriptive enough to determine flows that are closely coupled, i.e., are likely to be a part of stepping stone chain. Among most important characteristics belongs so called ON/OFF periods which are short periods when the flow is active or inactive and volume of data transferred during these periods.

Flexible FlowMon can support detection of stepping stones if it is set up with a short inactive timeout allowing to measure the lengths of activity periods. More-

over, a flow record may contain first order statistics on inter-packet gaps and packet lengths, which are the main discriminators of the stepping stones detection methods. Clearly, an accurate timestamp is vital to distinguish between prime and resulting flows and to provide a high quality input for subsequent detection methods.

## References

- [1] ZHANG, Y.; YANG, J.; YE, C. Modeling and Detecting Stepping-Stone Intrusion. In *International Journal of Computer Science and Network Security*, vol. 9, no. 7 p. 200–205, 2009.
- [2] ZHANG, Y.; PAXSON, V. Detecting stepping stones. In Proc. *9th USENIX Security Symposium*, Denver, 2000, p. 67–81.
- [3] MILLS, D. L. *Computer Network Time Synchronization: the Network Time Protocol*, CRC Press 2006, 304 p., ISBN 0-8493-5805-1
- [4] SMOTLACHA, V.; NOVOTNÝ, J.; MARTÍNEK T.: *Hardware Supported Precise Timestamps Generation*. Technical Report 11/2008<sup>2</sup>, Praha: CESNET, 2008.
- [5] ŽÁDNÍK, M.; ŠPRINGL, P.; ČELEDA, P. *Flexible FlowMon*. Technical report 36/2007<sup>3</sup>, Praha: CESNET, 2007.
- [6] CLAISE, B. et al. *Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information*. RFC 5101<sup>4</sup>, IETF, January 2008.

---

<sup>2</sup> <http://www.cesnet.cz/doc/techzpravy/2008/hardware-supported-timestamp-generation/>

<sup>3</sup> <http://www.cesnet.cz/doc/techzpravy/2007/flexible-flowmon/>

<sup>4</sup> <http://tools.ietf.org/html/rfc5101>