Advanced Traffic Performance Monitoring Tool for Time-Critical Communication Networks

TrafMon III White Paper

Version 1.0 February 03, 2010

AETHIS Document ID: TRAFMON-III/WP1.0

TABLE OF CONTENTS

LIST OF FIGURES

1. ABSTRACT

The maturity of today digital networks leads to their involvement in more and more demanding contexts. Some applications, like voice and video are relatively sensitive to time critical performances. Often the protocol stack can cope with very short anomalies, at the level of one or a few affected packets. Stringent applications do not even detect such transient data flow disturbances.

Therefore, most of the communication service level criteria are expressed in terms of short to medium term averages, where measurement granularity at a period of tens of minutes is felt already quite precise. Commercial performance evaluation tools often rely therefore on the injection of minimal test traffic superimposed on the actual data flows.

But in user contexts where safety-of-life considerations have to be taken into account, such statistical estimation doesn't suffice. Indeed, the confidence on service quality then relies on precise measurements of the actual flows, nearly eliminating the disturbance introduced by the monitoring. The performance tool has to directly gauge the performance of effective application traffic and can only cope with a fully controlled repatriation of the distributed measurements in a quite compact format.

A representative example is the Galileo Integrity retroaction loop, drastically augmenting the reliability of the GNSS time and positioning services. Its counterpart is a mission ground network, linking the numerous worldwide Galileo Sensor Stations (GSS) to the double Galileo Control Centre (GCC), where precise spacecraft integrity and orbitography parameters are computed in real-time, every second, leading to a feed-back delivered to the users via disseminated UpLink Stations (ULS). Including the necessary GeoSat hop, packets have to travel the world in less than 600 milliseconds without any capacity left for retransmission and under a stringent constraint of continuity (probability of lost or late packet in the order of 10^{-6})! Tele-surgery is also a promising new world of technological challenges for the networking technology.

A first prototype of our TrafMon tool (2005) has been built along these principles, where distributed probes, along the network paths, were gathering basic time information on every packet of the observed data flows. Once online collected centrally, any packet loss and the latency of each packet can be determined and presented in a way where overall trends and performances synthetic metric can be drilled down to the fine granularity of per packet information.

Since then, the tool has been significantly extended. First on the database and Web reporting part, leading to a professional and intuitive presentation of a rich set of performance metrics, always allowing to focus quickly on the few abnormalities. Then on the optional enrichment of collected per packet information permitting full discrimination and qualification the analysed data flows. Finally, it permits simple deployment, where a single probe is now capable of measuring complete request/response transactions round-trip time.

In addition, an auto-discovery utility simplifies the tuning of the TrafMon configuration file to actual context of use.

In parallel, the robustness of TrafMon III has been extensively validated, leading to appropriate fine tuning of performances, through its long-term use in different network environments: a basic test lab at ESTEC, aimed at pre-analysing the Galileo ground network challenge, the ESA OpsNet telemetry link between Redu and ESOC involved in the ground segment of the Integral spacecraft mission.

The TrafMon II is now used in qualifying the Galileo Mission Data Dissemination Network links, where it already revealed its usefulness in fine tuning the traffic shaping behaviour over the WAN and in pinpointing an Ethernet interface duplex mode configuration problem leading to unacceptable rate of packet loss.

2. TIME CRITICAL NETWORKS

2.1 CONTEXT

The TrafMon development aimed at producing a tool for monitoring common network flow performances but which is also well suited to very specific networks i.e., with traffic requirements close to the current limits of feasible, in terms of availability, latency, jitter, integrity and continuity of service.

In addition to the Galileo mission data dissemination network (MDDN), which has to comply with paramount cases of service criticality, some applications are identified below, that have similar requirements.

- Real-time support to critical operations as for instance remote medicine or surgery, or telescope in operation.
- Networks based on convergence, for which voice or video applications do not allow for micro-interruptions that would prevent immediate and clear understanding.
- Applications using multicast: once multicast support comes up in streaming over the high-performance research networks.
- Applications which open many parallel network streams to diverse locations.

Moreover, in a near future, some other users may require the service of time critical networks for their applications as follows.

- Applications used by a relatively small number of technically competent trusted users working with large datasets.
- Applications where there is a large discrepancy between bandwidth available via commodity network connectivity and bandwidth available via high performance networks (e.g., overseas sites in many regions, provided that the overseas site has access to high-performance network connectivity).

The top level issue of those “critical networks” is their latency requirements, which do not allow room for any acknowledgement and hence retransmission. For those time-critical traffic (TCT) networks, end-to-end latency requirements are much less than 1s even for intercontinental communication. For instance, latency of less than 600ms is required for the Galileo MDDN, even from Pacific Ocean area to Europe, and including all communication processes as well as security mechanisms.
This means that even when complete or non-corrupted, any late received packet, or reassembled message, is definitively lost, as no time is left for repetition request and retransmission. This time constraint significantly impacts the quality of service (QoS), especially in terms of overall availability requirements.

In summary, time critical traffic networks have to comply with very high availability, integrity and continuity requirements. The Galileo ground networks form the most well known

example. However, several other present or future applications have been identified, which justify the utilisation of the Fine Grain Traffic Performance Monitoring Software Tool (TrafMon) that is compatible with the monitoring of such outstanding performance requirements.

2.2 TIME CRITICAL TRAFFIC (TCT) MONITORING ISSUES

If communication service providers (CSP) go for the challenge of TCT networks, the quality of service they provide to users will be evaluated, through performance measurements, both from the user and the service provider view-points. Existing current monitoring equipment generally do not provide the level of detail required by the TCT network performance validation. This is the 'raison d'être' of TrafMon.

TCT networks monitoring implies measurement of key performance indicators that are briefly described below, with their related major issues.

- Availability is the probability of the service to be up and running over a certain period. Generally, it is measured through the cumulated service incapability over the considered operational period, during which the service is expected. Most operated monitoring systems are checking the performances over relatively long contractual periods i.e., weeks, months or even years. Very often CSPs ignore the short interruptions, which are excluded from their calculated and published statistics. Some service level agreements (SLA) even include a clause that tells that interruptions of less than, e.g., 5s are not accounted in their availability calculation for any penalty determination. Related representative statistical indicators also encompass the Mean Time between Failure (MTBF) and Mean Time to Repair (MTTR).
- Integrity of service is the probability that the received information is received without degradation over the period considered for service. Comparing with availability, integrity degradations may be caused by external factors to the network as deliberate aggressions on the communication system. From the user viewpoint, non-integer and non-available services generally result in the same inconveniences. However, for the CSP, evidencing degradations due to external causes is essential because of possible SLA clauses leading to non-application of penalty.
- Latency is the time elapsed between the information transmission and the information reception. When a packet or a combination of packets, which build a message, has arrived, it is necessary to check with the transfer time limit acceptable for the user. Moreover and as for Integrity, giving evidence that the latency limit trespassing is due a specific segment of the network may be important for a possible penalty sharing between CSPs, in the frame of SLA(s).
- Effective throughput, which is a function of the link capacity but also of the shaping behaviour of shared network nodes on the path and at the ends, represents the actual amount of payload data that an application can communicate per unit of time

and, of course the way it evolves over time. This requires collecting additional raw observation of the packet sizes.

- Jitter, in its one-point and two-point variants, is an important qualifier for the delivery of continuous stream of analogous signal sampling (voice, video), or for a stable synchronisation of distant clocks.
- Continuity is the probability over a coming period that a service will be provided with the required quality in terms of availability, integrity and latency. The continuity indicator is the most difficult to appraise, as it is concerned by the probability of getting a service within the performance requirements, without any interruption over a specific time slot. The prediction of continuity over the next time slot may be very important if related to safety of life (SoL), as for the Galileo SoL service which aims at supporting manned aircraft landing. Other applications as e.g., telemedicine or tele-surgery also imply SoL continuity requirements. The continuity is a way to quantify the confidence one can have in the quality of communication service while relying on it for SoL use.

Monitoring requirements are variable according to network applications. The variations consist of combinations of key performance indicators, with or without tuning of the above baselines. Some examples are provided below in this respect.

- Monitoring of the service availability for users is built on the latency measurements and on integrity checks.
- Integrity is correlated to security or/and forward error correction process. The information is sometimes encrypted for confidentiality or encoded for preventing integrity degradation up to a certain extent. The criteria for integrity performance evaluation are affected by this data handling. It can be of interest to evaluate the impact of the processes on e.g., individual flow latency, when the time critical traffic(s) are merged among multiple flows.
- Quality of service is often discriminated according to the traffic flows. The monitoring will have to take this discrimination into account.

3. THE NECESSITY FOR A SPECIALISED TRAFFIC PERFORMANCE MEASUREMENT TOOL

This chapter summarises the key requirements for such a monitoring system, as seen from the user viewpoint, whether he/she is concerned by service provision or by service utilisation. Then the lack of suitability of the classical market products is highlighted.

3.1 SUMMARY OF THE REQUIREMENTS ON A TCT NETWORK MONITORING SYSTEM

The performance monitoring system must observe, in a non intrusive way, unidirectional data flows, at least at source and destination sites. The potential impact of monitoring on network performance has tb be negligible. The actual monitoring, achieved centrally but online, both globally and per custom-identified data flow, concerns:

- incurred one-way latency;
- packet loss, whether it comes from network congestion, propagation degradation (corruption) or actual service interruptions.

The requirement for this specific monitoring concerns every single packet from their input interface to their final output. The QOS performance analysis reporting is to be done from the online data base of the basic raw measurements. Key performance indicators can be derived for presenting metrics quantifying service availability, integrity and continuity characteristics.

These above requirements, driving the first TrafMon prototype, have been augmented in further developing the product for widening the applicability of the tool:

- Provision of longer term trend views, although the gathering of per-packet observations leads to quickly explosive amount of data (aggregation, optimal design of database and queries);
- Coping with alternative paths (routes) of the observed packets, not only at intermediate network hops, but possibly also due to clustering a virtual service over multiple end-systems;
- Support of multiple (redundant) TrafMon central servers and possibility to gather raw observations into local probe files, repatriated and centrally processed in batch mode (deferred time);
- Single-probe operation: it shall be possible to probe some traffic flows at a single point, mapping corresponding pairs of ingress/egress packets (request/response or data and acknowledge) in order to detect broken transactions and measure round-trip time — this applies to ICMP Echo request/reply, SNMP request response, NTP query/response, TCP data/ack;

- Further qualification of a flow: at one of the probing points, selected elements of information shall be collected from data flow packets: microsecond timestamp (for fine analysis), packet size (for throughput computation), IP/TCP/UDP addressing and detailed qualifiers (for further qualification);
- TrafMon configuration assistance with auto-discovery of relevant flows.

The central reporting is capable to support service level monitoring, but also troubleshooting in case of abnormal network behaviour. Reporting shall include regularly refreshed, short-term indications as well as longer term evolution representations of the performance metrics or indicators. This implies a flexible zoom-in/zoom-out capability between fully aggregated large time frame and the per-packet behaviour within a fraction of a second, without unacceptable responsiveness imposed by the very large volume of raw observations data.

From the TCT service requirements, some design constraints can be derived. Those are certainly making a clear cut between the existing market products and the actual needs of the user of TCT network monitoring systems as TrafMon.

- **Unambiguous identification of packets**. In order to make an accurate monitoring of the packet losses, it is mandatory to unambiguously identify all transmitted packets.
- **Minimal or no additional monitoring traffic.** The additional monitoring information attached to the above prior requirement shall be accompanied by very limited additional traffic, in order to avoid competition with the TCT or to significantly increase the cost of network operations. This additional information flow may have the opportunity to run on the network with lower priority than the TCT flow. Or it can be locally buffered during a delimited test run without injecting any extra packet, for a posterior collection and production of the monitoring metrics;
- **Capacity to monitor separately the categories of flows**. As written above, the TCT monitoring system should be compatible with flows having distinct priorities, as for instance a hierarchy that is compatible with the Type of Service (TOS) field, which is part of the IP packet format.
- **Granular user selection of additional packet information items**. On a per flow class and per probing point basis, TrafMon is optionally configured with list of additional data items to be gathered together with the basic packet timestamp observations. This information is kept in the central database in support of fine grain analyses.
- **Flexibility in deployment**. TrafMon shall support one or multiple probing points, possibly varying from flow class to flow class. A same probe instance shall be capable of observing multiple points at once (either over alternative routes or at different local hops on the network path) and can also partly reconcile its observations for further compaction of the centralised monitoring data. Furthermore probing points from different probes can be used to monitor successive network hops or alternate routes for the flows.

- **Further support for granular troubleshooting analysis**. Configurable audit logging, complemented by software diagnostic messages at selected level of verbosity shall permit to keep the necessary information on the way packet observations are handled in the distributed TrafMon architecture. Resulting logs provide valuable information to fine analyses around abnormal behaviours, generally by detecting occurrences of unexpected packets and by fully recreating the timeline, within a narrow time span (one second or less), of all competing packets at a given point. Microsecond precision is necessary for recreating sequences in a packet burst. Also, care is to be taken to preserve observation just at the border of a connectivity break. This way early degradation of service quality can still be analysed after communication is restored, even when remote observations within a service break would have been destroyed by timeout.
- **Capability to monitor secured transmissions**. Users may utilise IPsec for their transmission. The monitoring system must be capable of monitoring this traffic as well as plain (unencrypted) traffic. Flow classification can take party of preservation of the IP ToS byte, remaining visible in VPN tunnels.

3.2 NON-ADEQUACY OF EXISTING PERFORMANCE MEASUREMENT TOOLS

The kind of applications that have been identified in section 2.1 form only a minority on the communications services market. Existing network management tools are designed for classical less demanding and less sensitive applications where episodic measurements on samples of messages, often injected on purpose, suffice to derive statistical macroscopic trends.

Most users, indeed, do not detect problems due to short discontinuities of service. Their applications are generally supported by TCP, which asks for re-transmission whenever a packet is missing. If the communication circuit is not saturated, the repetition phenomenon is not remarked as the resulting traffic slow down is negligible.

An experience run in 2004 on the INFONET Frame Relay network, in the context of the Galileo Network Performances Preliminary Testing (GNPPT) for ESA clearly illustrates the limitation. Two week of intensive testing revealed (after heavy manual post processing of gigantic data sets) the occurrence of four micro breaks of connectivity, lapsing from 5 to 15 seconds. The CSP didn't detect anything with its professional collection of NOC tools. But this would have conducted to severe discontinuities in Galileo Integrity based services, due to minutes of algorithms re-convergence.

Even those performance analysis tools available on the market, which monitor every packet at distributed probing points, tend to quickly aggregate their granular observations in the probes themselves to reduce the volume of centralised data to be processed. These produce only statistically smoothed results, hiding the anomalies impacting time-critical traffic based applications.

4. HOW DOES TRAFMON FUNCTION?

System Architecture

Because every packet of target flow classes has to be individually monitored, TrafMon encompasses distributed packet capture probes. This permits to collect timestamp observations on the actual traffic flows without modifying their behaviour.

A same probe can have multiple packet capture interfaces so as to gather observations at different local stages of the network path and/or at different branches of alternative routes. Of course, this also permits to cover different data flows when they do not appear at a single probing point in the infrastructure at a site.

For those bi-directional data flows where pairs of corresponding egress and ingress packets can be matched to measure the round-trip time of an application, a single probe suffice to collect start and end timestamps.

But often, uni-directional latencies of the traffic in each direction are of interest. This implies to place a probe at the source and another at destination. Optionally, TrafMon can collect its timing information from specially crafted IP packets with the TIMESTAMP option present. Here a single probe at destination could suffice, but then without ability to detect packet losses. Furthermore, the limitation in number of IP timestamps adds to the generally poor quality (accuracy and stability) of clock of the routers on the path (generally supporting only SNTP and inducing a serrated evolution of their local time).

Where possible, placing supplementary probes at intermediate node within the network path can also allow analysing the behaviour over each segment, even over alternate routes. For instance, when a link is made of a terrestrial line and a satellite hop, placing an additional TrafMon probe in the intermediate VSAT hub is interesting for discriminating performance figures over each segment.

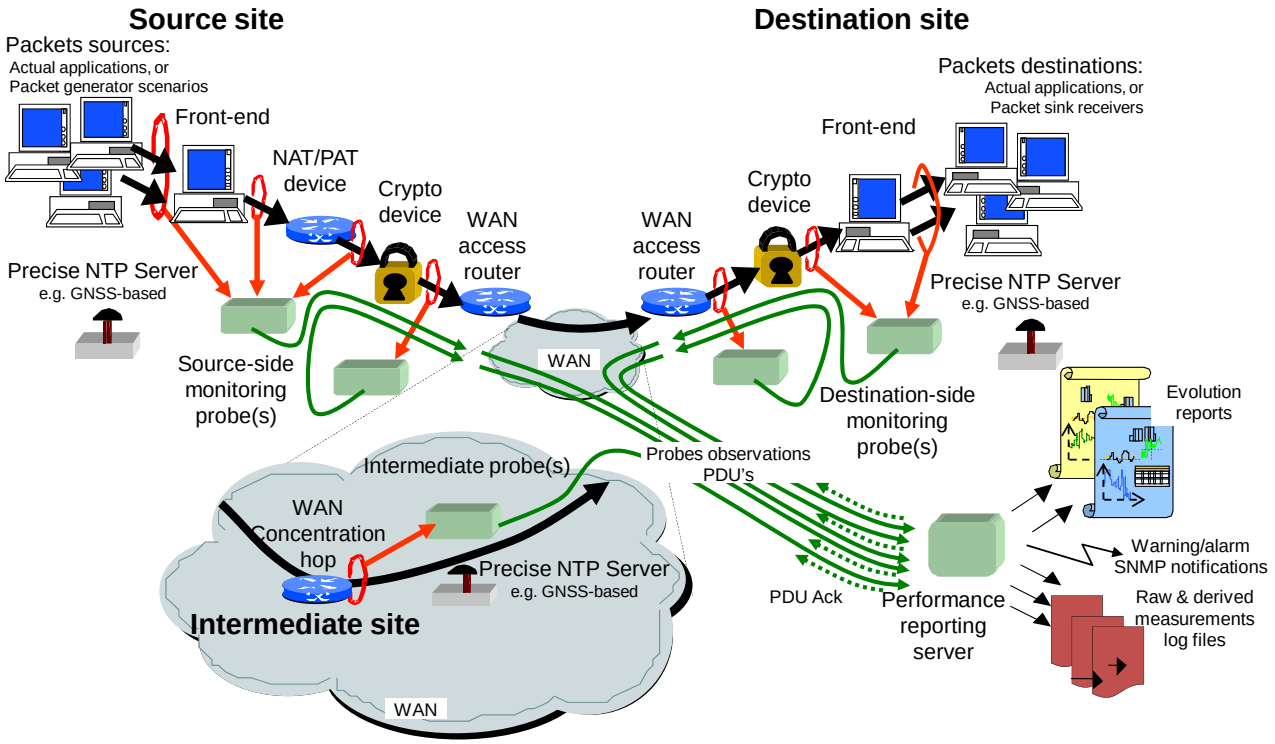

Figure 1: Example of Deployment Flexibility

TrafMon Mechanism

In order to match all partial observations of a same packet at different probing points, every packet need to be unambiguously identified: uniquely within its flow class and for a time span commensurate to the delay between observation gathering and its reconciliation processing. This is achieved by computing a MD5 hash of the packet content or of sufficiently identifying subset(s) of its data fields.

Of course, thanks to the entropy of the MD5 hash, only the first few bytes suffice as packet signature. Users can select between 1 to 5 bytes. Experience shows that 2 bytes leads to too frequent clashes. 3 bytes is a good compromise provided one can cope with about one clash per day, while 4 bytes seem to suffice for avoiding any clash in relatively sustained traffic over a week. This parameter influences the relative size of monitoring data possibly injected online through the same network link as the observed data flows. The shorter this information, the least is the risk of disturbance by the monitoring activity.

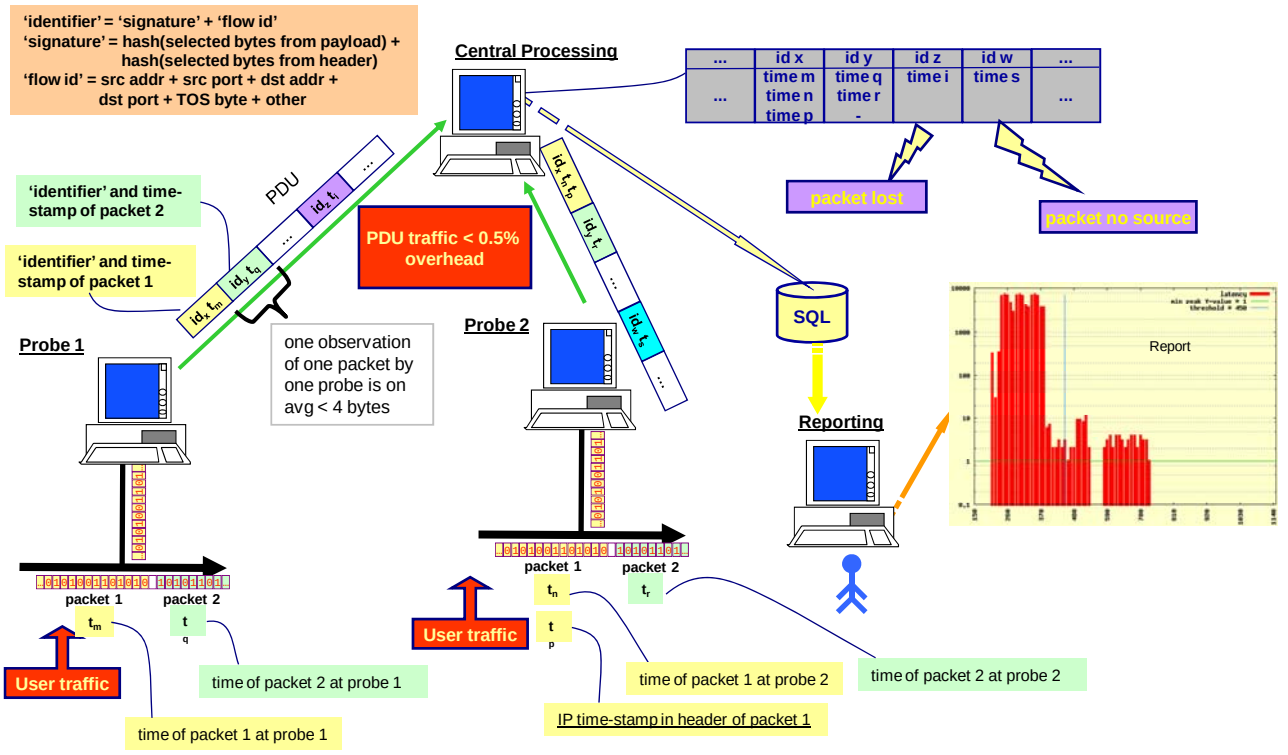

Figure 2: Collecting and Identifying Distributed Observations on Packets

Although it is possible to gather locally the observations into log files, which can be batch repatriated for deferred processing, the normal way is to continuously deliver TrafMon PDU towards one or more central TrafMon processing servers in order to refresh the performance reports in near real-time.

Therefore, besides the hash signature bytes, the flow identification and the packet timestamp have also to be kept compact. The idea is to reach a ratio of LESS THAN 1% OF INJECTED OBSERVATIONS DATA UNITS (TrafMon PDU's and acknowledges) compared to the observed traffic rate, when collecting the minimal information: timestamps for latencies and packet loss reporting.

So, in a PDU, packet observations are grouped by time and flows classes: transitions of time references (at ¼ second) and of flow identifiers are reduced. Each reference timestamp is encoded as an offset from the last time reference, and occupies less than one byte. Furthermore, when two observations on a same packet are collected by a same probe via two probing points, the second timestamp is added as extra information to the base packet record through a simple one byte time offset. This is then even more concise. And a series of IP timestamps is coded as successive 10-bit units disregarding the byte boundaries.

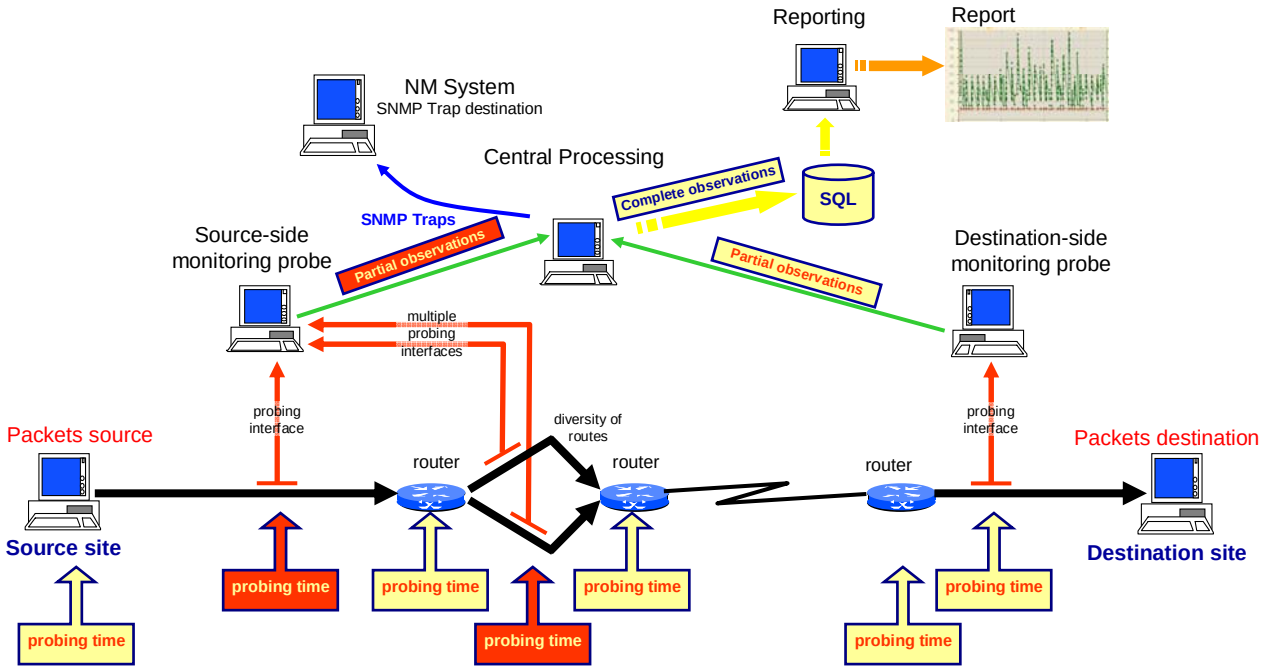

Figure 3: Another Example with Intermediate Probing Points on Alternative Paths

The user can also specify additional custom data to be added to the per packet observation, typically at one of the hops. But this is at the expense of higher monitoring data overhead, so preferably gathered at the TrafMon centralisation site:

- Second timestamp: another hop, possibly with NAT/PAT translation, or an *application-layer source time extracted from the packet payload*, or the *reply time* for transaction-based Single Probe Operation,
- Optional IP timestamps,
- Timestamp at the microsecond precision,
- Exact IP packet length,
- Extra protocol information among: source IP/port, destination IP/port, IP protocol, IP fragmentation, TTL, Type of Service, TCP flags, TCP sequence, TCP ack, TCP window,
- Optional application-layer extracted message number/fragment number.

TrafMon Component Architecture

The probe program and the central processing server program are binary executables written in C code. The handling of optional observations listed above is confined in dynamically loaded plug-in modules in both types of processes.

In order to cope with potentially important burst of user traffic, the probe needs to cope with semi-asynchronous threads of processing, while avoiding the pure parallelism induced by full multi-threading. It is therefore structured in slices, with buffering stages in between:

- Packet capture,
- Packet analysis: signature, flow classification, optional decoders,
- optional merging: reconciliation leading to primary and second timestamp,
- PDU formatting: compact encoding, computation of accuracy confidence level,
- Per-server PDU transmission, with acknowledgement or retry and selective drop on timeout in order to keep observations at the start of a communication break.

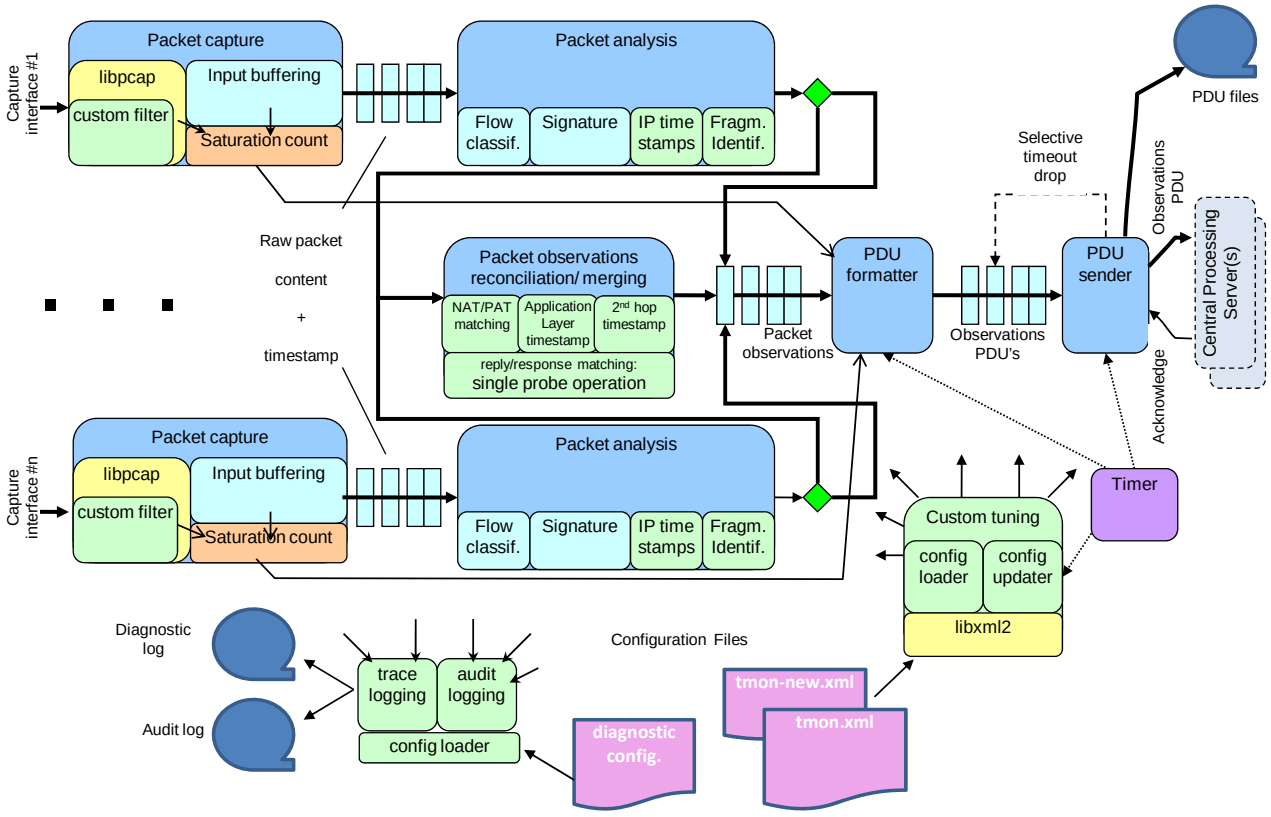

Figure 4: Structure of the TrafMon Probe Program

Each probe interface is specified a capture filter to roughly select the universe of packet potentially of interest. But the flow classification, in the probe analysis stage, consists of matching every packet with a second level of filter expressions. The filtering language is the well-known classical Berkeley Packet Filter (BPF) of libpcap and tcpdump public-domain utilities.

Note that a same flow can correspond to different matching filters when applied to packet at different interfaces of a same or of different probes: e.g. before and after VLAN tagging, before and after address translation...

Support of Single-Probe Operation (SPO) as well coping with potential manipulation/translation of packet headers fields over the network path is achieved through specific packet data selection upon computing packet signature. The user can specify fields in the TCP/UDP/IP headers to be masked. He can also specify one or more portions of data in the overall packet (headers of various layers and payload) to be used for hashing. And specific signing functions have been implemented to cope with the TCP Data/Ack mapping and with the SNMP Request/Reply ASN.1 BER identifier decoding. This permits to produce a data chunk, identical between the captured peer packets, subject to hashing for identifying signature.

The central processing is fed, either online via PDU reception/acknowledgement, or through batch loading of saved PDU files, with raw observations on user traffic packets: either partial or already complete due to single probe operation applied on some observed flows.

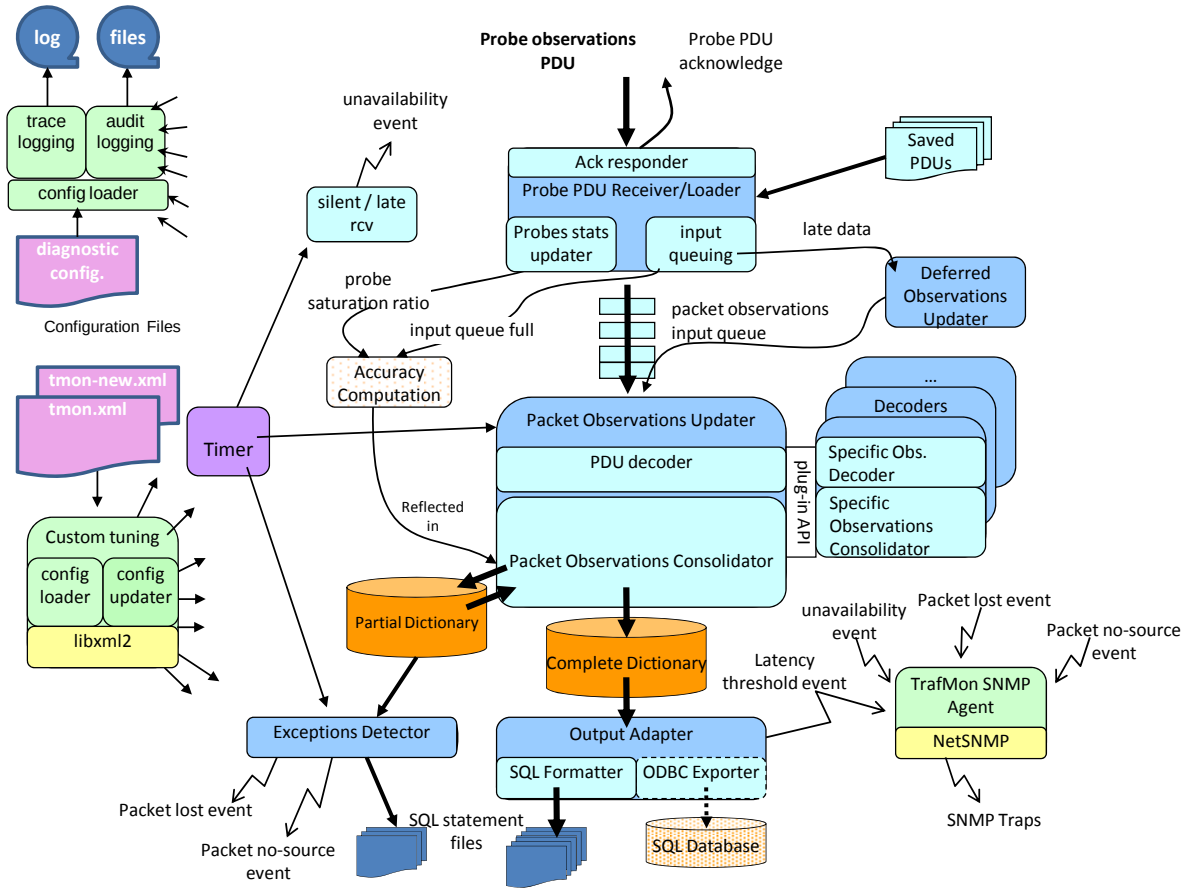

Figure 5: Structure of the TrafMon Central Processing Program

The online reception of observations PDU serves a dual purpose:

- Continuously feeding the reporter with fresh data,
- But also checking that the connectivity with each probe is alive.

For this second role, in absence of traffic, the probes regularly send empty heartbeat PDU's. In classical deployment scenarios, where the TrafMon PDUs are mixed with the observed traffic, this permits to generate useful unavailability event notifications.

In the TrafMon main configuration file, the user specify, for each flow, the sequence of probe flow hop timestamps expected to be collected on a per reporting flow basis. A reporting flow can therefore aggregate observations for more than one probe detected flows. Also, by giving the same hop name to a same probe flow, but at different probing points, one can specify alternate paths taken by a same reporting flow entity.

A packet observation is said complete when a timestamp is known for each of the expected reporting flow hop sequence.

The TrafMon central processing server works with two main internal data structures:

- The partial dictionary, keeps track of only partially reconciled observations on packets, as provided by one or more probes,

- And the complete dictionary, where packet observation records are migrated as soon as the last (possibly sole) probe record has been received.

Regularly (e.g. 1 min for exceptions and 2 min for data), the complete dictionary is scanned for producing SQL INSERT/UPDATE statement for each new record and timed-exhausted ancient partial records give rise to exceptions INSERT statements. The output of the central processing therefore consists of flat files to be executed for data import in an SQL database.

Currently, the reporting function relies on a PostgreSQL DBMS. Due to performance issue in maintaining automatic identifying key, the SQL statements files are post-processed for boosting the speed of data import.

This process of data import occurs typically every quarter, and loads all pending chunks of data into detailed tables. It continues by computing the various metrics available into reports and terminates by refreshing the aggregated tables at different scales. Those tables are used for direct charting of macroscopic performance figures.

Aggregation by buckets over the time scale but also along the metric ordinate leads to a very nice way to present synthetic performance trends. At first glance, it appears strange, but while understanding the double dimension aggregation it reveals very rich in information.

Care is taken, in histograms and availability figures for instance, to use logarithmic scale for focusing on rare cases at the boundaries or outside the normal behaviour. This permits the tool to extend is overall scope to support fine grain analysis of special cases and troubleshooting diagnostic. Also, because the finest granularity of raw data are also kept in the database for a custom-defined time span, it is always possible to zoom down to the per packet microscopic evolution of a metric instance, e.g. at a sub-second time interval.

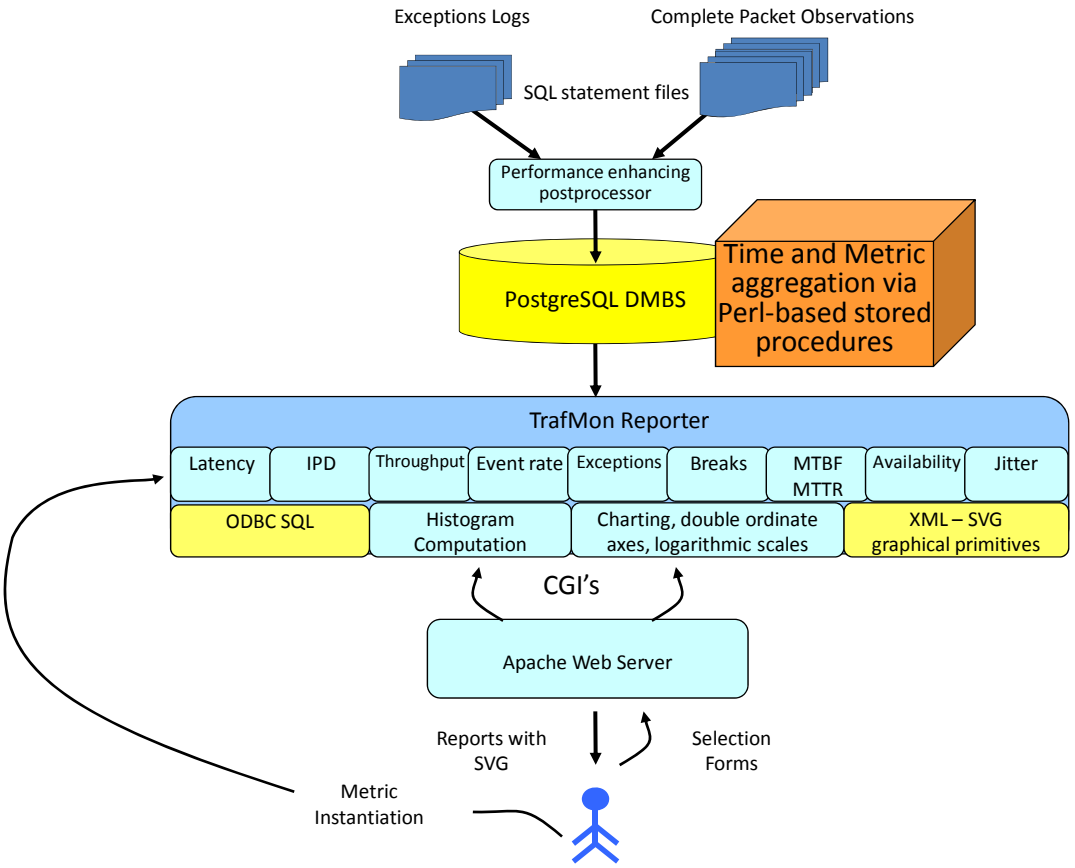

Figure 6: Architecture of the Web-based Interactive Report Writer

The Web interface consists in a series of forms, in Javascript, permitting to establish a query for a metric applied to a report-level dataflow within a given timestamp and a range of ordinate values.

This triggers server-side PHP scripts which in turn query the database, generate the XML/SVG result as interactive charts and tables of value listings.

Flexible zoom-in/zoom-out capabilities in the selectable charts permit to drill down to the per-packet granularity and to drill up to higher spans of time/ordinates.

Additional Troubleshooting Capabilities

The normal TrafMon reporting aims at presenting overall trends as well as pinpointing abnormalities. But three other data sources support on purpose detailed analysis for further diagnosing problems.

The first is the rebuilt of an approximate timeline of packet occurrences, possibly in both directions, at a given probing point and for a narrow time span (typically one or a few seconds). This is important because a packet lost or a slowdown can be due to the simultaneous occurrence of other packets of any flows. This is typically conducted by

manually crafting specific SQL queries on the database. Note that this is helped by the fact that any standard queries used by the TrafMon reporter appear in an optionally visible debug window of the Web interface. Such queries can easily be copy/pasted and modified as needed.

In addition, both TrafMon C programs (probe and central processing server) can generate systematic tracing of their processing stages. This diagnostic information can be tuned on a per functional object module level, from the less verbose (only Errors and possibly Warnings) to full verbosity (Trace0, Trace1, Trace2). Although this information is more aimed at program debugging support, it captures useful knowledge such as the fact that a signature clash occurred, or that a loss is correctly declared and is not due to the loss of its partial observation record, or also that other packets, not matching any flow classifying filter did occur during a time span under scrutiny.

Finally, auditing capability has been added for collecting a tuneable set of information about every processed packet. This audit logging can be imported in a classical spreadsheet tool to present views on the analysed traffic which complements those performance figures provided by TrafMon reports.

Configuration Support

Care has been taken, in designing the TrafMon custom configuration mechanism, to permit the use of a single XML file for all distributed TrafMon components (all probes, central server, database reporting). Even the database list of metric instances for each custom-defined flow comes from a script parsing the main tmon.xml configuration file.

By distributing a new file under the name tmon-new.xml, every permanent TrafMon process automatically reloads its configuration.

It can happen that the user wanting to deploy TrafMon in a given environment does not know a priori which relevant flows are present in the network. For this, a configuration support tool has been developed which can derive, from local or distributed auto learning, the Top-N most important flows and which generates a basic tmon.xml configuration based on this discovery.

5. TRAFMON USER INTERFACE OVERVIEW

This chapter describes the Web reporting interface of TrafMon III, as it results from the third development phase.

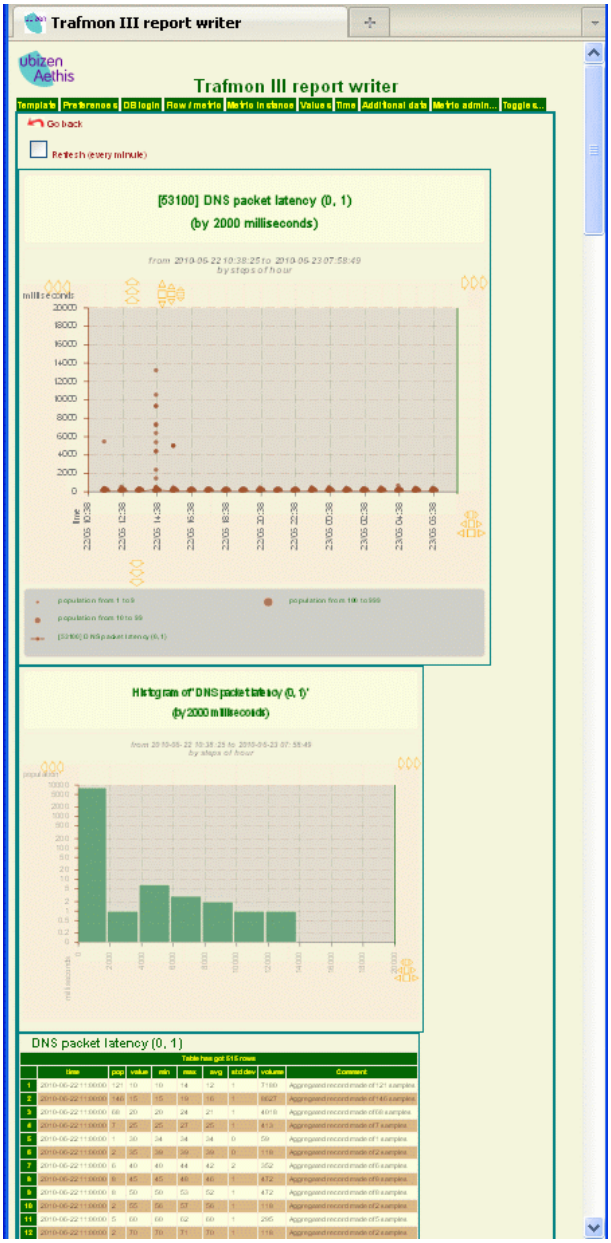

Figure 7: TrafMon III Overall User Interface

The top bar is a collection of buttons, each popping-up reporting-request parameter selection boxes. Laid under it is the resulting report window presenting the selected metric instantiated to the selected flow.

The report server supports access to different databases, in case the TrafMon administrator has directed different data collection periods to different databases. This is advantageous in case full per packet granularity needs to be kept for more than a week, in order to preserve the volume of data to a reasonable size for avoiding drastic slowdown of retrieval query performance.

Once the user has logged-in in a database, the reporter automatically initialises the selected span of the view to the number of days covering the database actual content.

Most of the available metrics are presented as a time graph aggregated in the two dimensions, of a logarithmic histogram over the selected time span and of a table describing the various aggregated buckets.

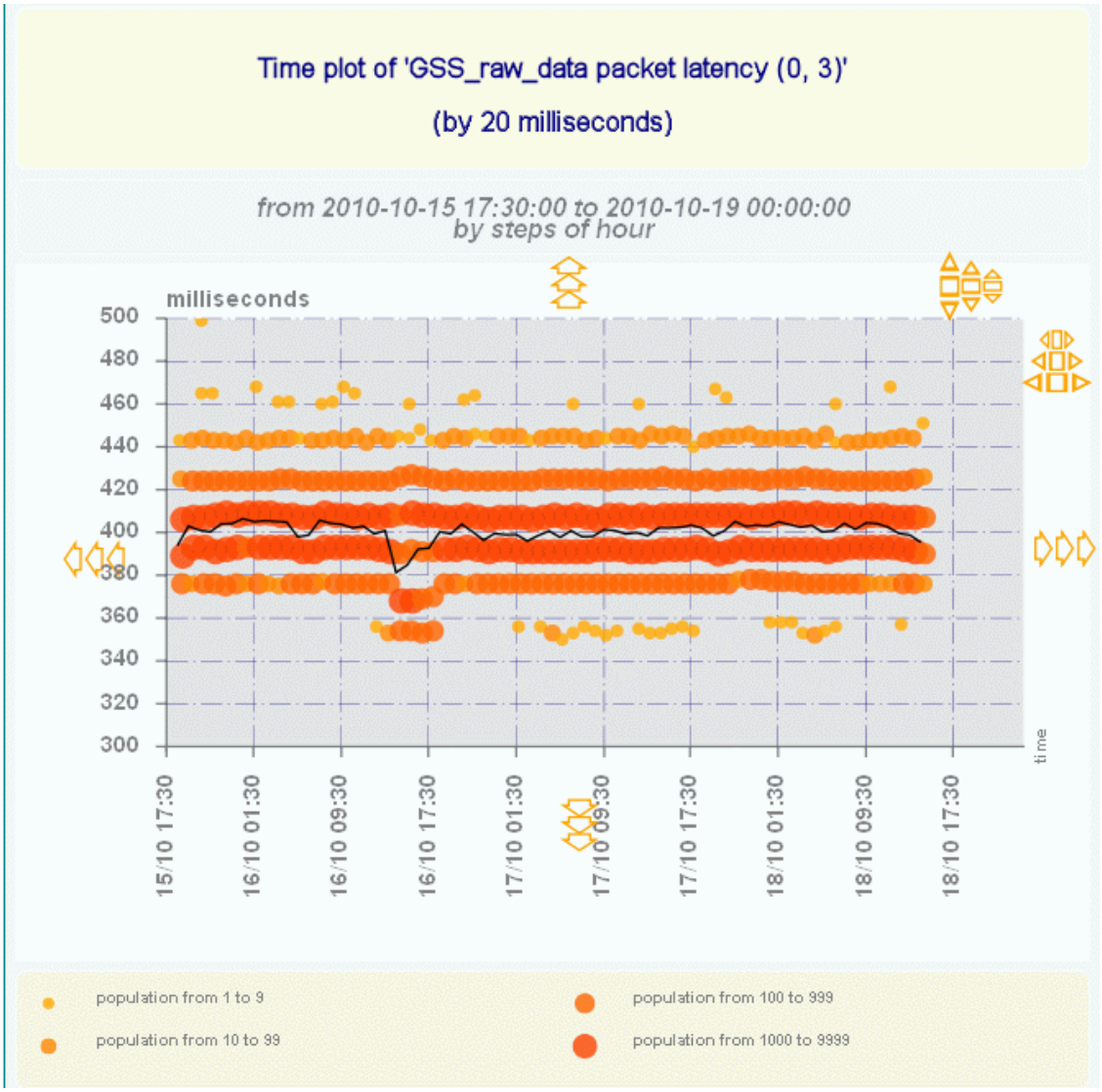

Figure 8: Aggregated Time Plot

The time plot must be interpreted carefully. The time axis consists of discrete intervals whose duration (1 hour in Figure 8) is automatically derived from the displayed time span. But the Y axis is also discrete (buckets of latency steps of 20 ms at Figure 8).

Each bucket for a given time slice and metric value interval, covering a non-null population, is displayed as a bubble whose size (and darkness) is commensurate to the number of individual measurements it represents (four possible sizes at Figure 8). Each such bubble is placed centred at its medium time and at the corresponding bucket average Y-axis value.

In the example of Figure 8, the data flow consists of only one packet sent every second, in the vast majority of the seconds, the observed latency is between (380, 420) ms. But there are about 1% of the seconds where the packet is taking between (360, 380) or (420, 440) ms. Yet there are few occasions where packets are relatively quick (340 ms to 360 ms) or relatively slow (over 460 ms but below 500 ms). Intuitively, one would wrongly think that there are burst of two to five and sometimes more packets sent simultaneously, due to the apparent lines of buckets in the graph.

The corresponding histogram is shown below. It is a Poisson curve. Its logarithmic scale emphasizes the abnormalities.

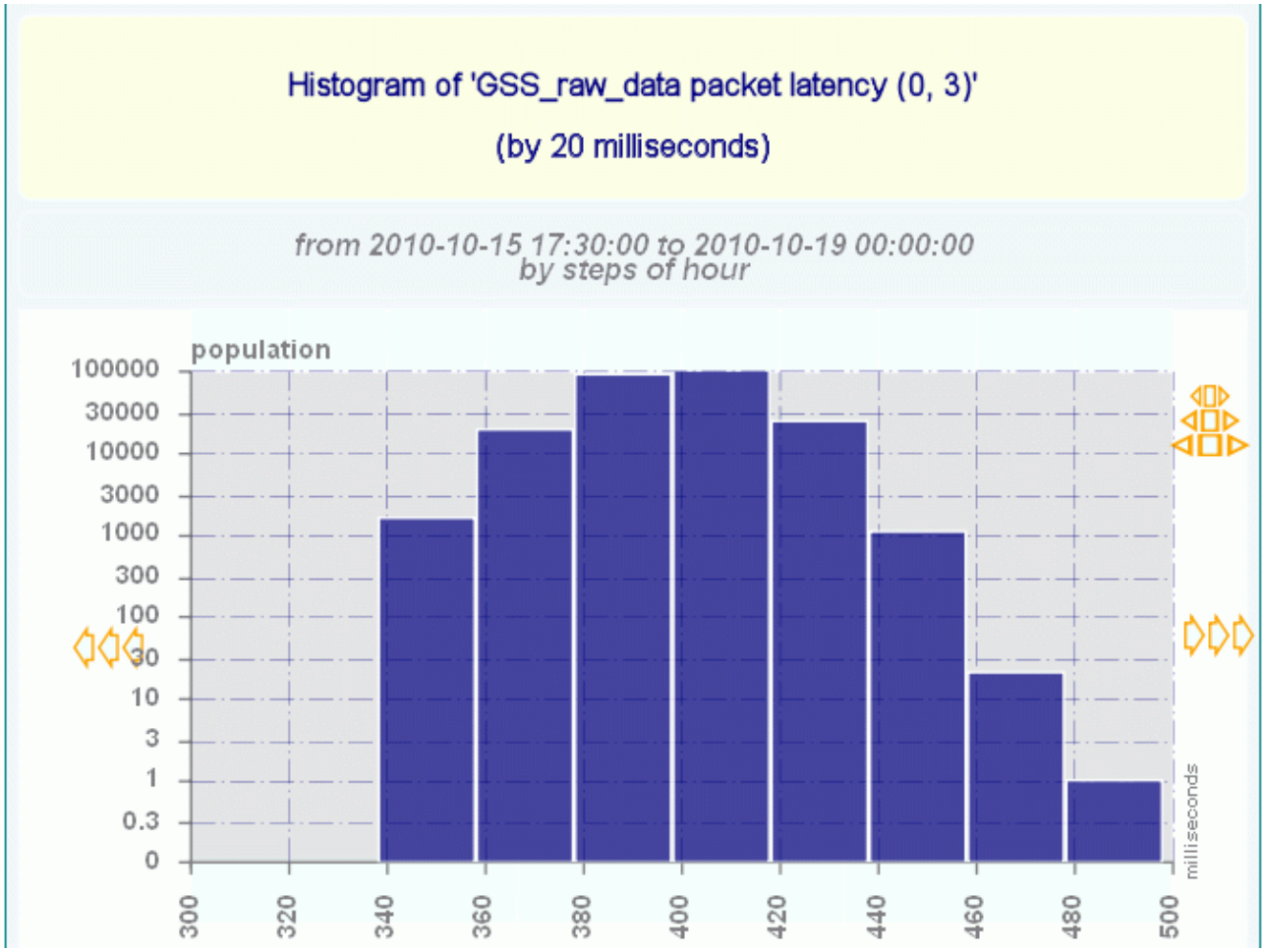

Figure 9: Histogram

Thanks to the use of W3C Scalable Vector Graphics (SVG) primitives, the objects in the charts are sensitive.

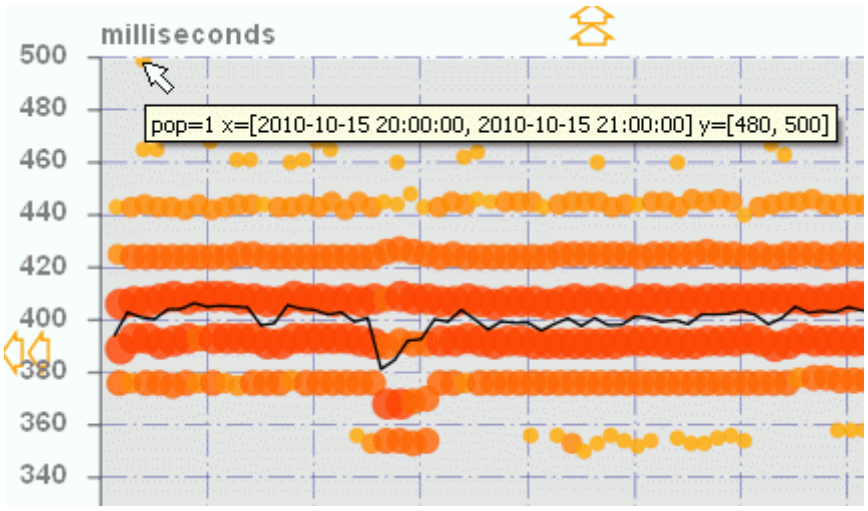

Figure 10: Dynamic Bucket Description

Placing the mouse on a bucket or on a histogram bar shows its description. At Figure 10, we see that the slowest bucket is only made of a single packet (population = 1). This can then be looked at in the table under the charts (Figure 11).

Table has got 393 rows									
	time	pop	value	min	max	avg	std dev	volume	Comment
1	2010-10-15 18:00:00	267	360	366	379	376	2	0	Aggregated record made of 267 samples
2	2010-10-15 18:00:00	2208	380	380	399	389	6	0	Aggregated record made of 2208 samples
3	2010-10-15 18:00:00	1039	400	400	419	406	5	0	Aggregated record made of 1039 samples
4	2010-10-15 18:00:00	82	420	420	438	425	5	0	Aggregated record made of 82 samples
5	2010-10-15 18:00:00	4	440	440	448	443	3	0	Aggregated record made of 4 samples
6	2010-10-15 19:00:00	25	360	375	379	376	1	0	Aggregated record made of 25 samples
7	2010-10-15 19:00:00	1460	380	380	399	393	4	0	Aggregated record made of 1460 samples
8	2010-10-15 19:00:00	1750	400	400	419	407	6	0	Aggregated record made of 1750 samples
9	2010-10-15 19:00:00	344	420	420	439	424	4	0	Aggregated record made of 344 samples
10	2010-10-15 19:00:00	21	440	440	450	443	2	0	Aggregated record made of 21 samples
11	2010-10-15 20:00:00	115	360	375	379	376	1	0	Aggregated record made of 115 samples
12	2010-10-15 20:00:00	1603	380	380	399	393	4	0	Aggregated record made of 1603 samples
13	2010-10-15 20:00:00	1625	400	400	419	407	5	0	Aggregated record made of 1625 samples
14	2010-10-15 20:00:00	243	420	420	439	424	4	0	Aggregated record made of 243 samples
15	2010-10-15 20:00:00	12	440	440	456	444	5	0	Aggregated record made of 12 samples
16	2010-10-15 20:00:00	1	460	465	465	465	0	0	Aggregated record made of 1 samples
17	2010-10-15 20:00:00	1	480	499	499	499	0	0	Aggregated record made of 1 samples
18	2010-10-15 21:00:00	331	360	364	379	376	2	0	Aggregated record made of 331 samples

Figure 11: Buckets Description Table

By clicking on a histogram bar, one zooms to the corresponding range of values, without changing the covered time span. But by clicking on a bubble in the time chart, one zooms to the narrow area corresponding to the bubble time slice and range of values. Note however that, to avoid confusion, the table listing the buckets adapts to the time span, but always cover the full range of values where there exist samples.

The evolution trend of the displayed metric is highlighted by the continuous line drawing the overall averages over successive time intervals.

Note that it is possible to toggle out each portion of the reported information: the bubbles, the average line, the complete time chart, the histogram and the buckets table. Also, when the table would be too long, the rows at the centre are skipped f the display.

While it is possible to manually select the start and end date/time and the value range via dialog boxes, the charts provides numerous navigation capabilities. Note however that each underlying set of SQL queries takes typically a few seconds. Therefore step by step browsing cannot be really reactive (along the time axis).

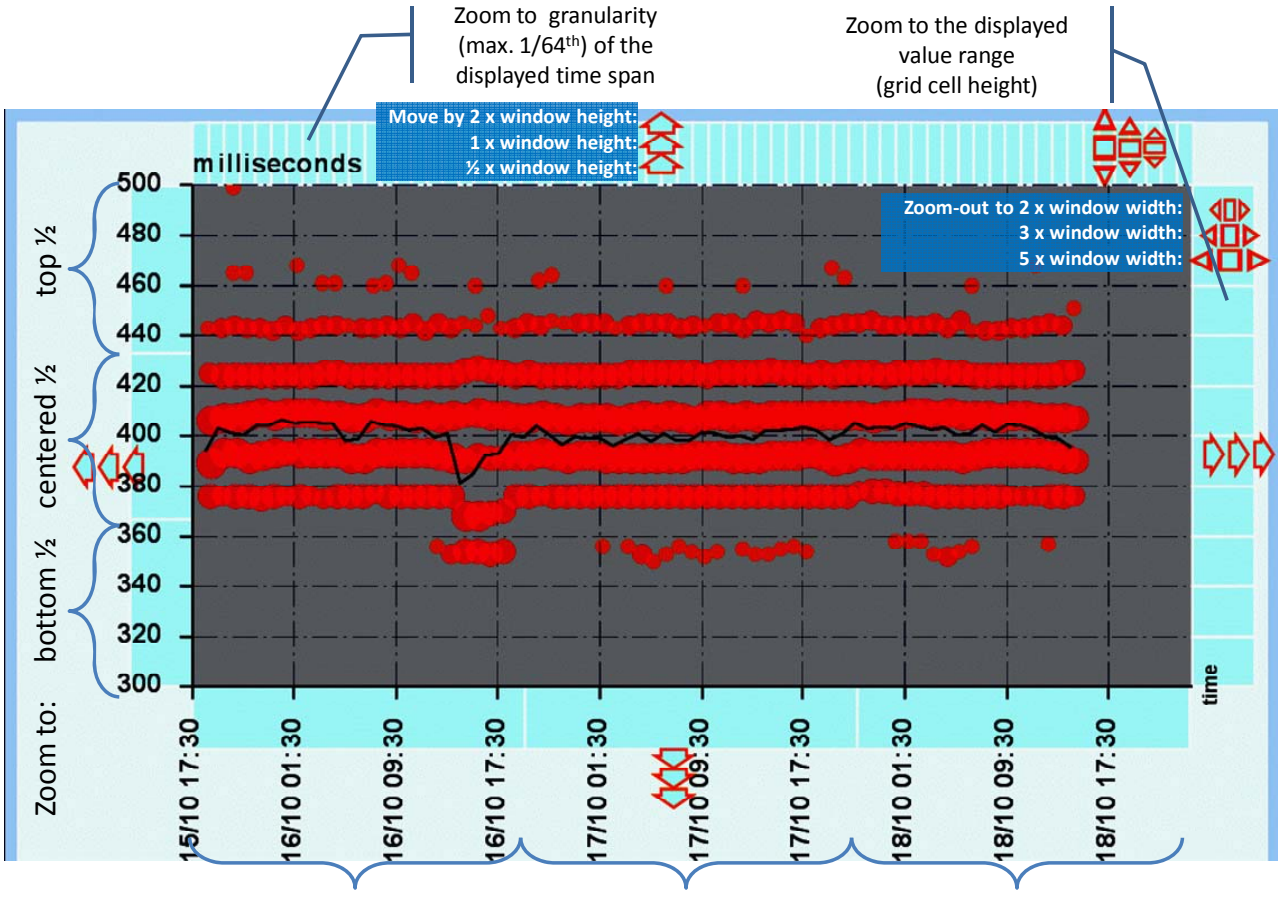

Figure 12: Browsing over time and over ordinate axes

It is also possible to display two metric instances in a combined report. In this case a single chart displays both with two possibly different ordinate scales. Time browsing applies to both, while ordinates browsing buttons are doubled.

Figure 13 shows an example of comparing the uplink and downlink latencies (between ESOC and Redu) of Integral telemetry. But the two metrics could differ, e.g. one-point jitter and inter-packet delay.

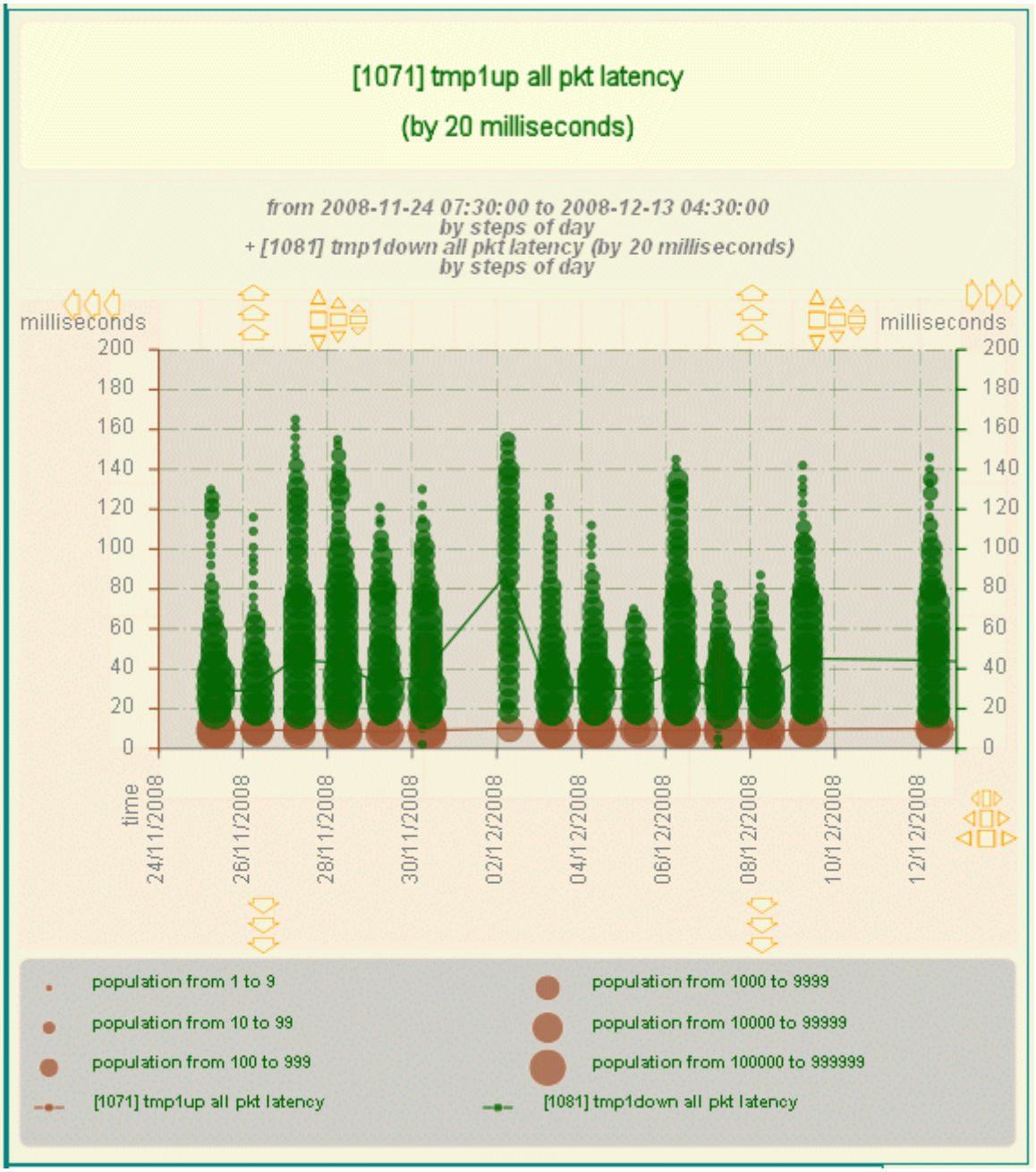

Figure 13: Reporting two Metric Instances