Minimizing Information Loss in Sequential Estimation of Several Quantiles

Jorma Kilpi[1] and Samu Varjonen[2]

[1] VTT Technical Research Center of Finland, jorma.kilpi@vtt.fi
[2] HIIT Helsinki Institute for Information Technology, samu.varjonen@hiit.fi

Abstract. We propose to network monitoring a statistically well justified method for on-line quantile estimation. The method satisfies scalability and robustness requirements and guarantees that the information loss in saved data is minimal. This is achieved by combining a sequential quantile approximation algorithm developed in [5–7] with an information theoretic histogram estimation method developed in [10].

Keywords: Network monitoring, Y.1731, Service Level Agreement (SLA), P^2-algorithm, on-line estimation, quantiles, Minimum Description Length (MDL).

1 Introduction

In network monitoring context at least three simultaneous targets from the operator point of view can be indicated: fulfillment of *Service Level Agreements* (SLA), *security* and *dependability*. The focus of this paper is on dependability but the motivation comes through SLAs. Ethernet networks with ambitious SLAs require good *operation*, *administration* and *maintenance* (OAM)tools.

There is a need to support both *reactive* and *proactive* usage of monitoring data. Reactive usage includes support for the *detection* and for the *localization* of faults, and *comparisons* of the present to the past: is the present state of the network worse, similar or better than the past state(s). Proactive usage includes *predictions* of the next state given the present state. This requires that the information of the present state is sufficient.

Real-time requirements mean that certain computations from the data, and decisions which are based on the information obtained from the data, must be very fast. *Non-real-time* requirements, on the other hand, require either saving large amounts of original data, or sufficient information extracted from the data in real time. A customer claim, for example, may require SLA verification many months afterwards. Therefore, *robustness* and *scalability* are key issues.

Current operator practice is often either to gather mean values and alarm or SLA threshold exceedances, or to gather all raw data. In this paper we propose a statistically well justified method which satisfies scalability and robustness requirements and, whenever possible, guarantees that the information of the present state is sufficient. This is achieved by combining a sequential quantile approximation algorithm developed in [5–7] with an information theoretic histogram estimation method developed in [10].

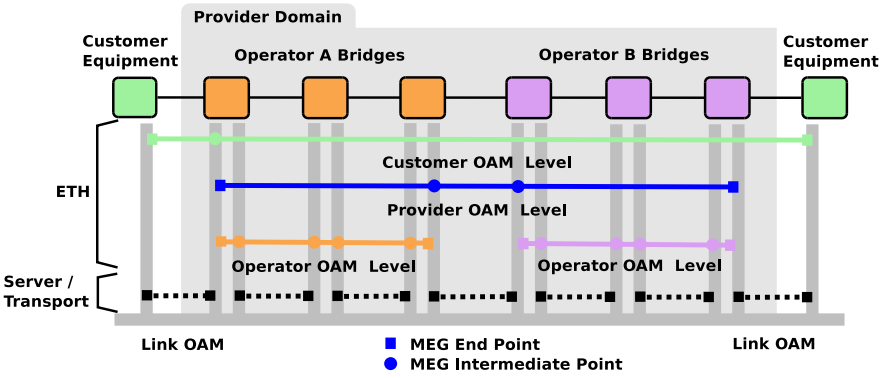

Fig. 1. Example of a Multi-Domain Ethernet Service OAM.

2 Background

2.1 The Context of Data

As the networks become larger and more complex, service providers will need better tools to maintain their networks. IEEE 802.1 [3], ITU SG 13 [4] and the Metro Ethernet Forum (MEF) [5] work in cooperation to develop standards for Ethernet service Operations, administration and management (OAM). In this article we concentrate on the ITU SG 13 protocol Y.1731 [1].

The main scope of OAM protocols is to provide fault and performance management capabilities for the network entities. In OAM protocols the monitoring entities in the network are called Maintenance Entities (MEs). The entities are divided into *Maintenance Entity group end Points (MEPs)* and to Maintenance Intermediate Points (MIPs). MEPs are the active measuring entities and MIPs forward the measuring traffic between MEPs. MIPs may participate in the measurements as responders but MIPs may not initiate measurements. The MEs monitoring in cooperation are organized to Maintenance Entity Groups (MEGs). Multiple MEGs may operate in the same network separated by their Maintenance Entity Level (MEL), as illustrated in the Figure 1.

The fault management of the Y.1731 protocol consists of proactive connectivity checks, in which the participants periodically send Connectivity Check Messages (CCMs) toward other participants. If the expected CCMs do not reach a MEP inside a grace period, the MEP will report an alarm.

The performance management of the Y.1731 consists of frame loss and delay measurements. The frame loss measurements can be carried out in one-way or two-way manner. One-way frame loss measurements are handled by piggybacking the frame counters in the CCMs. Two-way frame loss measurements are handled by exchanging the frame counters in a separate message exchange. The frame delay measurement is a “ping” like message exchange between participating MEPs.

[3] `http://www.ieee802.org/1/pages/802.1ag.html`
[4] `http://www.itu.int/rec/T-REC-Y.1731/en`
[5] `http://metroethernetforum.org/page_loader.php?p_id=29`

The Tool to Obtain Data. In the government and industry funded project called GoodNet [6], which focuses on network dependability, we implemented an open source Y.1731 software. Our implementation conforms to the technical specifications of Y.1731. We have tested the implementation in a test network setup in order to verify the correctness of the behavior of the implementation in various network fault and error cases.

Our implementation gives more flexibility than the commercial implementation to prototype the solutions found during the GoodNet project, such as additional features which are closely related to the Y.1731 daemon, including the topic of this article.

The left plot of Fig. 2 shows the amount of total data (kB/s) as a function of the number of MEPs when the MEPs form a full mesh monitoring topology. The full mesh assumption is a worst case situation. There are CCM, frame loss and frame delay measurements assumed to be running all the time. The polling intensity of 1/s is insufficient for verification of ambitious SLAs at gigabit speeds, but increasing the polling intensity increases the amount of data very fast.

In this paper we focus on frame delays. The range of frame delays is $[0, R]$ where R is called *a loss threshold*: if the actual frame delay is larger than R the frame is considered as lost. Delay values in gigabit speeds are conveniently expressed in microseconds (μs). The value of R may be specified in the SLA, for example, to be $1s = 10^6 \mu s$.

There is a relationship between the frequency of making the frame delay measurement and the typical value of the frame delay itself between two MEPs. Namely, if measurement frames are sent more frequently than the path delay is, the frames may block each other in the queues and this may cause some artefacts. Thus, if the measurement frequency is 100/s, that is 10ms intervals, then the typical delay value must be significantly less than 10ms.

2.2 Sequential Procedure for Simultaneous Estimation of Several Quantiles

The *q*:th quantile of a probability distribution. Let F be the cumulative distribution function (CDF) of a univariate probability distribution on the real axis. Then F is increasing but not necessarily strictly increasing function. For all $0 < q < 1$ *the q:th quantile* x_q of F is defined by *the generalized inverse*

$$x_q = F^{-1}(q) = \inf\{x \in \mathbb{R} \mid F(x) \geq q\} \ . \tag{1}$$

It is customary that the function F^{-1} is simply called the inverse of F, see [11]. If F is strictly incresing (injective, one-to-one), then F^{-1} coincides with the ordinary concept of inverse function.

The sample quantiles and order statistics. Given a sample $X_1, \ldots, X_n$ of n observations, or random variables representing possible observations, *the*

[6] See web-site `iplu.vtt.fi` for GoodNet and past network dependability projects.

empirical CDF is the piecewise constant function defined as

$$F_n(x) = \frac{1}{n}\sum_{i=1}^{n} 1_{\{X_i \le x\}}(x) = \frac{1}{n}\sum_{i=1}^{n} 1_{\{X_{(i)} \le x\}}(x) \ , \text{ for all } x \in \mathbb{R} \ , \qquad (2)$$

where the indicator function 1_A is $1_A(x) = 1$ if $x \in A$ and $1_A(x) = 0$ if $x \notin A$, and the ordered data $X_{(1)} \le \ldots \le X_{(n)}$ is called *the order statistics*. Generally $X_{(i)} \neq X_i$, hence $1_{\{X_{(i)} \le x\}}(x) \neq 1_{\{X_i \le x\}}(x)$ for many $x \in \mathbb{R}$ but, when summed over all observations, the equality in (2) holds for all $x \in \mathbb{R}$. When data is sorted, the original order of occurence is lost, or ignored. This is a step where some information about data source is lost, unless the observations are independent.

The sample quantiles can be defined in terms of (1) applied to (2). Then, for $\frac{i}{n} \le q < \frac{i+1}{n}$, $i = 1, \ldots, n-1$, it holds that $F_n^{-1}(q) = X_{(i)}$. Actually, $F_n^{-1}(q) = X_{(r)}$ where $r \le i$ is smallest such that $X_{(r)} = X_{(i)}$; ties[7] are typical in this context. If $0 < q < \frac{1}{n}$ then $F_n^{-1}(q) = -\infty$ but, if we restrict the domain of F_n to be $[0, R]$, then we can agree that $F_n^{-1}(q) = 0$ in this case. We also have $F_n(x) \ge q$ if and only if $x \ge F_n^{-1}(q)$.

Order statistics, sample quantiles, and the empirical CDF are practically equivalent approaches. The choice between these formulations is simply done according to which is relevant and convenient to the particular purpose [11]. They can be viewed as considering all sorted data, with fixed n, as a parameter.

If the data is *an independent and identically distributed (i.i.d.)* sample from a distribution F, then for any fixed x the $F_n(x)$ can be considered as a random variable. If x is fixed the $F_n(x)$ is *asymptotically* normally distributed with mean $F(x)$ and variance $(F(x)(1 - F(x))/n)$ as $n \to \infty$, while *the exact* distribution of $nF_n(x)$, both x and n fixed, is binomial with parameters n and $F(x)$, see Chapter 2 of Serfling's book [11]. We do not know F and independence is an ideal assumption, but these results are not completely useless though.

For fixed q the $F_n^{-1}(q)$ is, likewise, a random variable in the *i.i.d.* case. Due to generalized inverse one needs to assume that the hypothetical F is strictly increasing near x_q in order that $F_n^{-1}(q)$ would converge to x_q when n increases. In case of frame delays there is a physical justification for this assumption whenever the accuracy and the precision of the timing clock are sufficient. Namely, then we can assume that all values in the range $[0, R]$ have positive probability to occur[8] and this means that the cumulative F can be assumed to be strictly increasing. Then a strong result follows that, for every $\varepsilon > 0$,

$$\mathbb{P}\{|F_n^{-1}(q) - x_q| > \varepsilon\} \le 2e^{-2n\delta_\varepsilon^2} \ , \qquad (3)$$

where $\delta_\varepsilon = \min\{F(x_q + \varepsilon) - q, q - F(x_q - \varepsilon)\}$ estimates the minimum slope of F at x_q, [11].

Description of the P^2-Algorithm. The P^2-algorithm was first developed by Jain and Chlamtac [5] for sequential estimation of a single quantile of a

[7] Observed values X_i and X_j, $i \neq j$, are called *tied*, if $X_i = X_j$.

[8] Except, possibly, very near 0. But this is insignificant.

CDF F, then extended by Raatikainen in [6] for simultaneous estimation of several quantiles, and further analyzed by Raatikainen in [7]. The paper [6] of Raatikainen contains the remarkably short algorithm code in detail.

The extended P^2-algorithm approximates the inverse of the empirical CDF by utilizing a piecewise-parabola[9] to adjust the estimates of quantiles. If the parabolic adjustment cannot be applied, linear adjustment is used instead.

The algorithm maintains an array data structure of size $(2m+3)\times 4$ numbers where m is the number of quantiles x_{q_j}, $j = 1, \ldots, m$, of an assumed CDF F that the algorithm estimates. The probabilities q_j, $j = 1, \ldots, m$, are given as a parameter; they need not be equally spaced but they need to correspond *distinct* sample quantiles. This is because from the definition of sample quantiles it follows that if $\frac{i}{n} \leq q_1 < q_2 < \frac{i+1}{n}$, then $F_n^{-1}(q_1) = X_{(i)} = F_n^{-1}(q_2)$ and ties may cause even further restrictions.

Actually, for initialization the algorithm requires $2m + 3$ distinct observations. If it is known beforehand that the data source can be considered as *i.i.d.* from F, then the values needed for initialization could be just the first $2m + 3$ (distinct) observations. However, in Section 4.1 we will introduce a learning step and, generally, it is better to use the information obtained from the learning step for the initialization. This guarantees good performance from the very beginning. Besides, in Section 3 will show, there may arise a need to perform the initialization step every now and then.

The vector $(\widehat{x}_{q_1}, \ldots, \widehat{x}_{q_m})$ of current estimates can be read out and saved from the data structure *at any time during a measurement*. Regular reading and logging, which is assumed here, leads to time series

$$(\widehat{x}_{q_1}, \ldots, \widehat{x}_{q_m})_k \ , \ \ k = 1, 2, 3, \ldots \ . \tag{4}$$

formed by these vectors, instead of time series of the original data. Reduction in the amount of data is substantial if measured in bytes. However, some information about the data source is necessarily lost.

For further reference the index k of (4) will be called *quantile vector index*. The total number of original data points observed and taken into account between two consecutive readings of the quantile vector is a parameter and it will be denoted by n. The total amount of data from the beginning of the measurement is then kn. The amount of data needed to store reduces to $100\frac{m}{n}\%$.

Computational performance of the P^2-Algorithm. The computational complexity of the P^2-algorithm is $\mathcal{O}(m)$. Our implementation of Y.1731 includes an implementation of the P^2-Algorithm as an additional feature. The right plot of Fig. 2 depicts the time it takes to update the data structure one million (10^6) times for a new observed value. This has been computed on a PC (Intel Core2 Duo, CPU 2000 MHz, on single core) and on a special purpose network device based on ALIX system board (AMD Geode, CPU 498.128 MHz) which would correspond to an assumed MEP. In a Geode-like MEP the single update should take far less than 1 ms even if $m = 20$.

[9] The name $P^2 = PP$ comes from the two P-letters in Piecewise-Parabola [5].

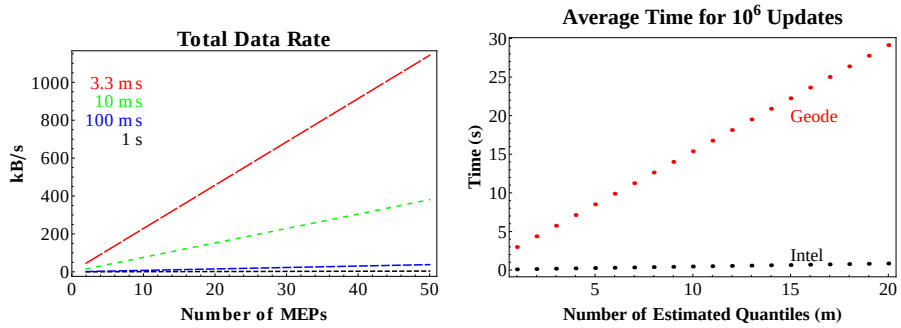

Fig. 2. Left: Total data rate (kB/s) as a function of number of MEPs when they form a full mesh monitoring topology. Right: Average time (s) it takes to update the data structure of P^2-algorithm for one million (10^6) values as a function of m.

We conclude that the P^2-algorithm is computationally simple and fast enough to be run in the MEP which, generally, is assumed to have only limited capacity reserved for computing and data storage.

Statistical performance of the P^2-Algorithm. Time series is called *stationary* if the distribution of the data source does not depend on the time when the observation begins. Stationary time series may be *autocorrelated* which means statistical dependence in time order, also called memory.

Raatikainen [7] already studied the performance of the P^2-algorithm in case of an autocorrelated source by generating sojourn times of M/M/1 queue with fixed traffic intensities. Hence, the performance for a stationary data source with short memory is already known to be good. Long memory case is left for further study.

3 Applying the P^2-Algorithm for Monitoring

Based on our experience and physical intuition of Ethernet network's performance, we assume that the non-stationary phenomena in the observed frame transfer delays between two MEPs are the most relevant issue to be detected and that they are transient. That is, most of the time the frame delays are in some roughly stationary state. After a transient non-stationary phase the current stationary state of the network need not be the same as what it was before the non-stationary phase.

Denote by $\widehat{x}_q^{(k)}$ the k:th estimate of a q-quantile. If all the data were saved, then the $F_{kn}^{-1}(q)$ always have an interpretation and, therefore, if $\widehat{x}_q \approx F_{kn}^{-1}(q)$ there is also an interpretation for the estimated quantiles, even for non-stationary data source. Therefore, in the presence of trends or seasonalities, it would be optimal to measure the relative difference

$$100 \times \left(\frac{\widehat{x}_q^{(k)} - F_{kn}^{-1}(q)}{F_{kn}^{-1}(q)} \right) \% \tag{5}$$

where $F_{kn}^{-1}(q)$ is the q:th sample quantile of cumulative original data up to the point when the k:th estimate $\widehat{x}_q^{(k)}$ is read out from the data structure. Since (5) requires saving of all data, it is presented here only for illustration. The important thing is that neither $F_{kn}^{-1}(q)$ nor $\widehat{x}_q^{(k)}$ looks for future, only to the present and to the past; $F_{kn}^{-1}(q)$ from the very beginning and $\widehat{x}_q^{(k)}$ has some memory due to the data structure of the algorithm.

It is easy to work out non-stationary examples where the relative difference (5) becomes huge. The P^2-algorithm is simply not designed for a non-stationary data source. The funny thing is that this is exactly what we want since the target is to detect non-stationary periods. In this section we present two computationally simple and fast methods to detect non-stationary periods by testing how well $(\widehat{x}_{q_1}, \ldots, \widehat{x}_{q_m})_{k-1}$ predicts $(\widehat{x}_{q_1}, \ldots, \widehat{x}_{q_m})_k$. See also the left plot of Fig. 4 for a flow chart which describes the monitoring with the P^2-algorithm.

The important thing to understand is that, whenever a nonstationary moment is detected, then the data structure of the P^2-algorithm must be initialized. It is the only way to maintain any ability to interpret the data (4) that the P^2-algorithm produces. The main usage of data (4) is for SLA verification but the non-stationary periods are interesting due to dependability and security reasons.

3.1 Bin boundaries

The quantile vector $(\widehat{x}_{q_1}, \ldots, \widehat{x}_{q_m})$ defines $m+1$ variable width bins

$$[0, \widehat{x}_{q_1}],]\widehat{x}_{q_1}, \widehat{x}_{q_2}], \ldots,]\widehat{x}_{q_m}, R]. \tag{6}$$

The bin boundaries fluctuate but, under the stationarity assumption, their fluctuations should typically cancel each other. We can compare just observed $(\widehat{x}_{q_1}, \ldots, \widehat{x}_{q_m})_k$ against the previous $(\widehat{x}_{q_1}, \ldots, \widehat{x}_{q_m})_{k-1}$ by computing

$$S = \sum_{j=1}^{m} \left(\widehat{x}_{q_j}^{(k)} - \widehat{x}_{q_j}^{(k-1)} \right) \quad . \tag{7}$$

We should have $S \approx 0$. Assume for a moment that both $\widehat{x}_{q_j}^{(k)}$ and $\widehat{x}_{q_j}^{(k-1)}$ are estimates of x_{q_j} of F from *i.i.d.* data. Then the triangle inequality allows us to infer first that $|\widehat{x}_{q_j}^{(k)} - \widehat{x}_{q_j}^{(k-1)}| \leq |\widehat{x}_{q_j}^{(k)} - x_{q_j}| + |x_{q_j} - \widehat{x}_{q_j}^{(k-1)}|$ and by (3) we can conclude, after some computations, that for every $\varepsilon > 0$

$$\mathbb{P}\left\{ \left| \sum_{j=1}^{m} \widehat{x}_{q_j}^{(k)} - \widehat{x}_{q_j}^{(k-1)} \right| > \varepsilon \right\} \leq 4 \sum_{j=1}^{m} e^{-2n\delta_{\varepsilon,j}^2} \tag{8}$$

where $\delta_{\varepsilon,j} = \min\{F(x_{q_j}+\varepsilon) - q_j, q_j - F(x_{q_j}-\varepsilon)\}$. Here n, m and q_j, $j = 1, \ldots, m$ are known and for the estimation of $\delta_{\varepsilon,j}$ we can use information obtained from the learning step to be explained in Section 4.1.

3.2 Bin frequencies

Denote by n_j, $j = 1, \ldots, m+1$, the number of observations in the j:th bin between two readings of the quantile vector (4). Furthermore, assume that after the previous reading of the quantile vector the values are always set to zero: $n_j = 0$, $j = 1, \ldots, m+1$. Thus, in the current reading $n_1 + \ldots + n_{m+1} = n$.

There appears to be a problem in the definition of bin counts n_j since the estimated bin boundaries $\widehat{x}_q$ may change for each observed value. However, the probabilities of these bins do not change. Therefore, as long as the assumption of stability of the bin boundaries holds, which is first tested by (8), the bin counts which are computed according to the dynamic bins behave well. Adding of $m+1$ counters to the P^2-algorithm do not increse the computation time significantly.

Using redundant, but convenient, notation $q_0 = 0$ and $q_{m+1} = 1$, define

$$p_j = q_j - q_{j-1}, \;\; j = 1, \ldots, m+1. \tag{9}$$

Then $p_1 + \ldots + p_{m+1} = 1$. If the data source were *i.i.d.*, then the exact distribution of *the bin frequency vector* $(n_1, \ldots, n_{m+1})$ associated with (6) is multinomial with parameters $n_1 + \ldots + n_{m+1} = n$ and $(p_1, \ldots, p_{m+1})$. Multinomial distribution is not easy to use in practice. If we define the matrix $\Sigma = (\sigma_{jr})$ by

$$\sigma_{jr} = \begin{cases} p_j(1-p_j), \; j = r \\ -p_j p_r, \qquad j \neq r \end{cases}, \tag{10}$$

then the *normalized relative* bin frequency vector

$$\sqrt{n}\left(\frac{n_1}{n} - p_1, \ldots, \frac{n_{m+1}}{n} - p_{m+1}\right) \tag{11}$$

converges in distribution to multinormal distribution $N(0, \Sigma)$, for details see Serfling's book [11]. Thus, once q_j, $j = 1, \ldots, m$ are specified, we can compute a multivariate ellipsoid, based on $N(0, \Sigma)$ inside which the vector (11) should be with some prescribed probability.

4 Minimizing Information Loss

The Minimum Description Length (MDL) Principle. So far we have assumed that n, m and q_j, $j = 1, \ldots, m$ are given. The method to find values for m and q_j, $j = 1, \ldots, m$ is based on applying the Minimum Description Length (MDL) principle in the learning step. Learning step is viewed as data compression. The process chain discussed in Section 3 is: raw data $\rightarrow$ P^2-algorithm $\rightarrow$ lossy compression. This is analogous to a low pass filter: only relevant features are saved. We want also to minimize the information loss in this chain.

The formulas (6) and (9) define *a variable bin width histogram (VBWH)*. We simply ask, given a data, what values of m and q_j, $j = 1, \ldots, m$ provide he best VBWH model for the data. The model selection in MDL is based on the concept of *stochastic complexity* which is a joint measure of the ability of a model to

compress the data in uniquely decodable lossless manner and the complexity of the model itself.

The main sources for the MDL theory include the two books of Rissanen [8] and [9], the book of Grünwald [4] or his tutorial article [3]. The specific application we need is already solved in [10] and discussed also in [2].

Stochastic complexity of a VBWH for a data $X_1, \ldots, X_n$ in the given range has been computed in [10]. When translated for our purposes, we get the following. Assume n, m and the range $[0, R]$ are given, then we need to find q_j, $j = 1, \ldots, m$ which minimize the following:

$$\sum_{j=1}^{m+1} n_j \log \left(F_n^{-1}(q_j) - F_n^{-1}(q_{j-1}) \right) + \log \binom{n}{n_1, \ldots, n_{m+1}} + \log \binom{n+m}{n} \quad (12)$$

For an example of the minimization of (12) in case when $m = 3$, see Fig. 3. In this simulated case the data source was a multimodal distribution.

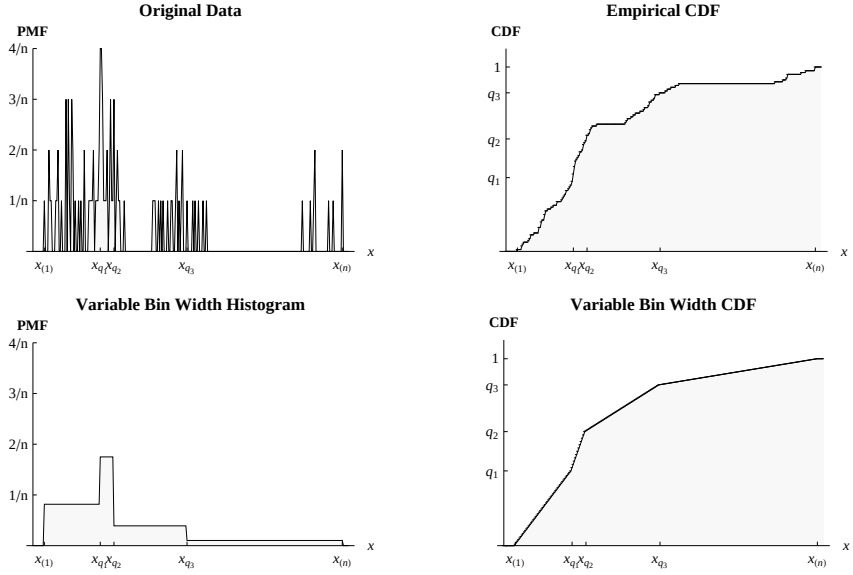

Fig. 3. An example of lossless compression in case when $m = 3$. The top left plot describes the empirical probability mass function (PMF) of the original data, where each observation is given a measure $\frac{1}{n}$. The top right plot is the empirical CDF of the original data, from which the sample quantiles are computed. The bottom left plot is the VBWH for this data after the optimal q_1, q_2 and q_3 are found by minimizing (12). The bottom right plot is the CDF of the VBWH.

Furthermore, encoding m and minimizing it together with (12) removes dependence on m. In [10] the joint minimization is formulated as a recursive dynamic programming algorithm which, unfortunately, is computationally demanding and cannot be performed in a low capacity MEP.

4.1 Learning Step

The P^2-algorithm requires the values q_j, $j = 1, \ldots, m$ as input and the aim is to find those q_j that can be argued to minimize the information loss. The argument comes from the lossless data compression that is performed assuming that a VBWH with parameters $(q_1, \ldots, q_m)$ is 'the' model of the learning data. In the beginning of a (long) measurement some raw data is saved and the MDL approach is applied to this raw data in order to find the VBWH that compresses the data most in a lossless fashion, *i.e.*, when no information is lost, see the right plot of Fig. 4. The amount of learning data is assumed to be much larger than the value of the parameter n shall be so that various choices for n can be considered.

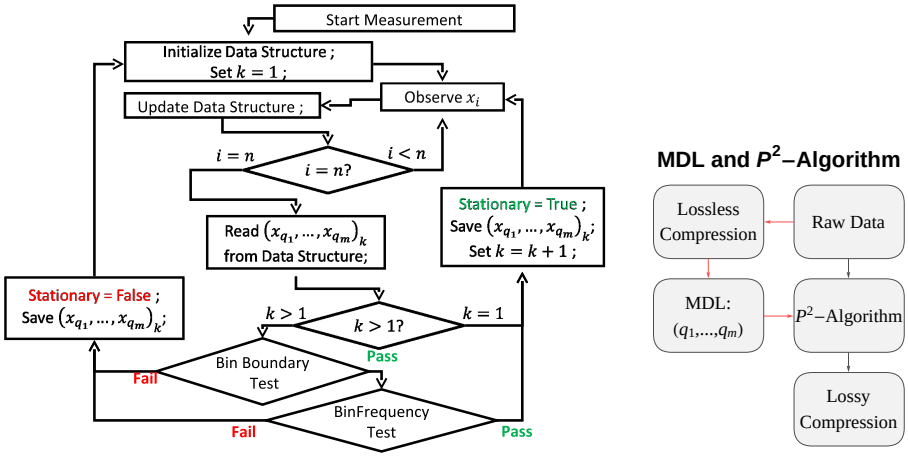

Fig. 4. Left: Flow chart of the usage of P^2-algorithm as described in Section 3. Right: Illustration of combining the MDL principle and the P^2-algorithm.

From the learning data good initialization data set(s) for the P^2-algorithm are computed. It can be that different initialization sets are needed if, for example, daily variation in traffic profile affects significantly to the frame delay values. This can be the case if the network is empty some part of the day. One good initialization data comes directly from the minimization of (12).

The value of $\delta_{\varepsilon,j}$ needed in (8) is essentially the minimum slope that the cumulative distribution model at x_{q_j} has. The cumulative VBWH model of $(\widehat{x}_{q_1}, \ldots, \widehat{x}_{q_m})_{k-1}$ can be used to estimate it in (8). See the bottom right plot of Fig. 3 for an example of the piecewise linear cumulative VBWH. Virtually it does not depend on ε. The sensitivity of the hidden dependence to ε needs to be estimated from the learning data.

The only remaining parameter that is not yet specified is n. The formula (8) could, for example, be used to specify n if the $\varepsilon > 0$ in (8) is fixed and $\delta_{\varepsilon,j}$ is replaced by some $0 < \delta \leq \delta_{\varepsilon,j}$.

5 Conclusions and Open Issues

We have presented an application of information theoretic density estimation to be applied with a sequential estimation of several quantiles. We have also explained how to apply the P^2-algorithm in on-line network monitoring context to detect non-stationary periods. The novel combination of these methods satisfy the scalability and robustness requirements.

Because the computational complexity of the P^2-algorithm is $\mathcal{O}(m)$, the argument probabilities can be a mixture of both q_j that are used for minimizing information loss and values like 0.5, 0.75, 0.9, that are meant for staff use or are specified in an SLA.

The MDL model selection step of minimizing (12) together with m, the number of quantiles, is computationally demanding. Therefore, the learning step needs to be made in a computer with ordinary PC processor capacity.

The method(s) to select an optimal value for the parameter n is an open issue, although (8) can already be used for that purpose. The selection method could, and perhaps should, be adaptive.

References

1. ITU-T SG 13. OAM functions and mechanisms for Ethernet based networks. ITU-T G.8013/Y.1731, ITU, July 2011.
2. Andrew Barron, Jorma Rissanen, and Bin Yu. The Minimum Description Length Principle in Coding and Modeling. *IEEE Transactions on Information Theory*, 44(6):2743–2760, October 1998.
3. Peter Grünwald. A Tutorial Introduction to the Minimum Description Length Principle. In Jae Myung and Mark A. Pitt, editors, *Advances in Minimum Description Length: Theory and Applications*. MIT Press, 2005.
4. Peter Grünwald. *The Minimum Description Length Principle*. MIT Press, 2007.
5. Raj Jain and Imrich Chlamtac. The P^2 Algorithm for Dynamic Calculation of Quantiles and Histograms Without Storing Observations. *Communications of the ACM*, 28(10):1076–1085, October 1985.
6. Kimmo Raatikainen. Simultaneous Estimation of Several Persentiles. *Simulation*, pages 159–164, October 1987.
7. Kimmo Raatikainen. Sequential Procedure for Simultaneous Estimation of Several Percentiles. *Transactions of the Society for Computer Simulation*, 7(1):21–44, March 1990.
8. Jorma Rissanen. *Stochastic Complexity in Statistical Inquiry*, volume 15 of *Series in Computer Science*. World Scientific, 1989.
9. Jorma Rissanen. *Information and Complexity in Statistical Modeling*. Information Science and Statistics. Springer, 2007.
10. Jorma Rissanen, Terry P. Speed, and Bin Yu. Density Estimation by Stochastic Complexity. *IEEE Transactions on Information Theory*, 38(2):315–323, March 1992.
11. Robert J. Serfling. *Approximation Theorems of Mathematical Statistics*. Wiley Series in Probability and Statistics. John Wiley & Sons, 1980.