Telecommun Syst
DOI 10.1007/s11235-009-9210-9

# Assessing the accuracy of using aggregated traffic traces in network engineering

**Lucjan Janowski · Philippe Owezarski**

**Abstract** Aggregated traffic traces are commonly used in network engineering for QoS or performance parameters evaluation. Many performance models come from such aggregated traces. However, real traffic is a marked point process combining two processes: one for the arrival times of packets and the other for their size in bytes. This paper deals with assessing whether aggregated traces are a good representation of real traffic. Based on the analysis of many traffic traces, and focusing only on loss probability, it is shown that the packet drop probability obtained for the aggregated traffic traces can significantly differ from the real packet drop probability obtained for the real traffic traces. Then, a solution which enables one to obtain correct loss probability based on aggregated traffic traces is proposed by determining the correct aggregation scale and traffic parameters to be applied.

**Keywords** Aggregated vs. real traffic · Loss probability · Unit translation · Aggregation scale

## 1 Introduction

Modelling today's Internet traffic is a difficult research problem. Internet traffic consists of numerous different sources, services, access technologies, etc. In addition, traffic models can be different from one link to the other or from one moment to another. Nevertheless, in early 90s the self-similarity property has been discovered as a common property in all traffic traces of any kinds of network technologies (e.g. LAN [9], UMTS [8], etc.) and applications (e.g. video [7], www [5], etc.). Then, numerous different self-similar models have been proposed. An interesting overview is given in [16]. The proposed self-similar models are able to describe the bursty nature and complex long range dependence structure of Internet traffic [9, 16] which were proved to be essential for performance evaluation in networking [19], for instance. Since then, realistic traffic models are used with network engineering tools (as simulators or emulators), for instance, for creating realistic experimental scenarios with realistic background traffic conditions. It is then expected to get, using such realistic traffic models, realistic results, whereas it has been recently shown that using basic models as Poisson or Markov for Internet traffic leads to optimistic performance estimates [14]. Note however, that most of the recent traffic models are based on aggregated traffic traces, essentially because they are easier to compute and use compared to models relying on a proper representation of real traffic.

Traffic aggregation is one of the commonly used representations of Internet traffic. Aggregation means that the traffic is a random variable of the number of bytes or packets sent during a time interval $\Delta$ (called time window). Such representation is commonly used in the traffic visualisation (figures showing a server link utilisation [15] for instance) and modelling (modelling aggregated traffic instead of real traffic [16]). Theoretically, it is obvious that real traffic is indeed different from aggregated traffic. The proper representation of the real traffic is a marked point process that is compounded of two processes: the packet incoming times and the packet sizes. Since the aggregated traffic and the real traffic representations are different it is crucial to check if the results obtained for the aggregated traffic are the same

L. Janowski (✉) · P. Owezarski
LAAS, CNRS, 7 avenue du Colonel Roche, 31077 Toulouse Codex 4, France
e-mail: janowski@kt.agh.edu.pl

P. Owezarski
e-mail: owe@laas.fr

Published online: 23 October 2009

as those obtained for the real traffic. Note that the term "results" needs to be precisely defined since we are not able to analyse all possible results. Moreover, QoS and performance parameters seem to be the most important from the network analysis point of view. Therefore, in this research we focused on comparing drop probability observed for a single sever or router queue fed by the real or aggregated traffic. The drop probability is selected because it is one of the important QoS parameters and already raised significant modelling work [14, 16].

In order to analyse drop probability we have written functions implemented in MATLAB. The algorithm that is used for the real traffic analyses queue length each time a new packet arrives. The queue length for the aggregated traffic can only be observed at the end of each time interval. Note that drop probabilities obtained for the real and aggregated traffic are obtained for the same trace. By the same trace we understand that the aggregated traffic trace is the real traffic trace aggregated over a time window.

The reason why the real drop probability can be different from the aggregated drop probability is related to the different ways of computing queue lengths for the real and aggregated traffic because of the different traffic representations. Note that for the real traffic a new packet can arrive at any time and each packet has its own size. On the other hand, for the aggregated traffic, we observe the system only after constant time intervals. In this later case, the only information we have is the number of packets arriving during this particular time window. As a consequence, counting only the number of packets is not sufficient and packet sizes would be missing. In order to compute the service time we have to know the packet size for all aggregated packets. In any paper considering aggregated traffic, we did not find information about the packet size. On the other hand in order to obtain the same number of sent bytes for the real and aggregated traffic, we have to assume that the packet size equals the mean packet size. Therefore, we think that for the aggregated traffic traces each packet size is assumed to be equal to the mean packet size. It is shown in this paper that such an assumption—similarly as observing the system only at constant intervals—can influence the obtained drop probability.

The paper is organised as follows. In Sect. 2, the danger of using aggregated traces is illustrated with some examples. Section 3 presents the methodology used for analysing the differences obtained for real vs aggregated drop probability. Section 4 describes drop probability as a function of the time window, with detailed analysis of some assumptions that we have to do in order to obtain this result. In Sect. 5, the equation presenting a solution making it possible to obtain the same drop probability for real and aggregated process is proposed. In Sect. 6 we described some consequences of using empirical solution presented in this paper. The last section concludes the paper and presents future research topics. Additionally in Appendix, two proofs are shown.

## 2 Motivation

A common Internet traffic representation is the number of requests, sent bytes or users logged into the system over day, week or other amounts of time. Actually, each source sends packets characterised by particular size and sending time. Therefore, the precise description of Internet traffic is a marked point process $\{(t_l, A_l), l = 0, 1, 2, \ldots\}$ where $t_l$ is the $l$th packet arrival time and $A_l$ is the $l$th packet size or any other feature [18].

The marked point process is difficult to model since the two $t_l$ and $A_l$ processes have to be considered. The difficulty comes from the complicated autocorrelation and correlation function between $t_l$ and $A_l$.

Since modelling the marked point process is difficult, the aggregated traffic is commonly used in related literature [9, 14, 16]. The aggregated traffic is the amount of bytes or packets sent during a time window $\Delta$ given in seconds or milliseconds. The aggregated process is called the byte or packet aggregated count process and denoted by $W_\Delta(k)$ and $X_\Delta(k)$ respectively.

Computing drop probability for different traces, we discovered that the drop probabilities obtained for the marked point process $\{(t_l, A_l)\}$ and the aggregated processes $W_\Delta(k)$ and $X_\Delta(k)$ are not equal.

An example of different drop probability values obtained for the same traffic trace is shown in Table 1.[1] Note that the drop probability obtained for aggregated process is $\Delta$ dependent (different values were obtained for $\Delta = 0.001$ and $\Delta = 0.1$). As a consequence, the drop probability obtained for aggregated process does not equal the real drop probability for all values of $\Delta$.

This example illustrates that the aggregated drop probability (i.e. a drop probability obtained for aggregated traffic) can be erroneous. Therefore in this paper, we describe one way that makes possible obtaining an aggregated drop probability as close as possible from the real drop probability, as well as analysing the possible reasons of such difference.

## 3 Methodology

In order to compare the real drop probability with the drop probability obtained for aggregated traffic, we analysed real traces. They are GPS-synchronized IP header traces captured with a DAG family [4] Ethernet network tap. Two different sources of traces were used. The first was the NLANR project [13] and the second the METROSEC project [11].

From NLANR project, traces captured at different hours 5, 8, 13 and 18 were used. The traces captured within the

[1] The detailed description of the methodology for getting these results is given in Sect. 3.

**Table 1** An example of different drop probability values obtained for the same traffic trace (the analysed traffic trace contains 1 154 282 packets, its duration is 781 seconds and MTU size is 1500 bytes), but different traffic representations and network parameters. $p_{\mathrm{p}}$ is real packet drop probability, $p_{\mathrm{b}}$ is real byte drop probability, $p_{\mathrm{pa}}$ and $p_{\mathrm{ba}}$ are respectively the drop probability obtained for the aggregated process $X_\Delta$ and $W_\Delta$ where $\Delta$ is given in seconds

| $\rho$ (link utilisation) | $B$ (buffer size) [packets] | $p_{\mathrm{p}}$ | $p_{\mathrm{pa}}, X_{0.001}$ | $p_{\mathrm{pa}}, X_{0.1}$ | $p_{\mathrm{b}}$ | $p_{\mathrm{ba}}, W_{0.001}$ | $p_{\mathrm{ba}}, W_{0.1}$ |
|---|---|---|---|---|---|---|---|
| 0.33 | 20 | 0.032 | 0.009 | 0.006 | 0.047 | 0.082 | 0.024 |
| 0.33 | 75 | 0.012 | 0.006 | 0.006 | 0.024 | 0.027 | 0.023 |
| 0.66 | 20 | 0.105 | 0.041 | 0.018 | 0.136 | 0.168 | 0.058 |
| 0.66 | 75 | 0.034 | 0.017 | 0.015 | 0.052 | 0.064 | 0.046 |

METROSEC project in addition contain DDoS attacks and flash crowds. As a consequence, the traces present different characteristics depending on the time of the day and traffic anomalies.

Since non-stationary processes are difficult to analyse [12], we used only traces for which the obtained traffic has an approximately constant number of packets sent during a time interval, and its packet size distribution was similar to a typical packet size distribution (see [6]). As a consequence, we have been working on traces not longer than 1 200 000 packets (which is nevertheless sufficient to get statistically significant results).

The drop probability computation was done using MATLAB for both real and aggregated traces. Each drop probability value was computed by an algorithm simulating single server queue behaviour. For the marked point processes (real trace), we know for each new packet the arrival time $t_l$ and the packet size $A_l$. Note that the service time is packet size dependent and if the $l$th packet is accepted it leaves the queue at time

$$\tau_l = t_l + \tau_k + A_l/C, \tag{1}$$

where $\tau_k$ is the service time (including storing time) of the $k$th packet, i.e. the one which was queued just before the $l$th one, or 0 if the queue was empty at time $t_l$. Note that $k = l-1$ only if the $(l-1)$th packet was accepted and was not served before $t_l$. $C$ is link capacity given in bytes per second. By knowing the arrival and departure time of each packet, it is possible to compute the number of packets stored in the queue at any time. Then, the $l$th packet is queued if the number of stored packets at time $t_l$ is less than $B$, dropped otherwise.

For the aggregated traffic, the queue length is given by [16, see Chap. 9]

$$Q(i+1) = \min(\max(0, Q(i) + X_\Delta(i) - C\Delta), B), \tag{2}$$

where $Q(i)$ is the queue length at $i$th time interval, $C$ is the link capacity, $B$ is the queue buffer size and $X_\Delta(i)$ can be replaced by $W_\Delta(i)$ if the byte queue length is considered. Note that in this paper, as drop probability is considered, and that the information unit handled by routers is packet, we will consider the $X_\Delta(i)$ process. Then, $Q(i)$ represents the queue size in terms of number of packets, $B$ is also expressed in terms of number of packets, and for concordance reasons, $C$ must be expressed in terms of number of packets per second.

In (2), the network parameters $C$ and $B$ are used. Nevertheless, link utilisation ($\rho$) is more significant when considering drop probability than $C$ which is constant over the durations considered in the simulations presented in this paper. In order to maximise the utilisation of our traces, we considered different link capacity $C = \bar{X}/\rho$ for each trace, where $\bar{X}$ is the trace mean bit rate (i.e. $\bar{X} = \sum_{l=1}^{N} A_l/(t_N - t_1)$). Since Internet links are not heavy loaded [2], the considered values of the link utilisation were limited in all our experiments to range $\rho \in (0.25, 0.75)$. The second network parameter $B$ was limited to range $B \in (20, 100)$ packets since the default settings of the queue buffer size of Ethernet link for cisco routers are 40 packets for output queue buffer and 75 packets for input queue buffer [21]. Generally speaking the queue buffer sizes in Internet are short and they should be short for performance purpose [17]. By considering such queue buffer sizes, all the realistic values from the real network point of view are studied. Moreover, for low link utilisation considering long queues and drop probability is questionable since we cannot observe any drops.

In this paper, for each original trace, the *real drop probability* $p_{\mathrm{p}}$ and *aggregated drop probability* $p_{\mathrm{pa}}(\Delta)$ are compared. The *real drop probability* $p_{\mathrm{p}}$ is obtained for the marked point process, buffer size given in packets and link capacity given in bytes per second. Therefore, $p_{\mathrm{p}}$ is the drop probability that we would observe in the real network. The *aggregated drop probability* $p_{\mathrm{pa}}(\Delta)$ is obtained for the aggregated traffic where aggregation window is $\Delta$. In order to compute $p_{\mathrm{pa}}(\Delta)$ the terms of (2) have to be expressed with the same units. It is important to note that link capacities are usually expressed in terms of bytes per second while buffer size is in number of packets [21]. Therefore, in order to compute drop probability for aggregated process (i.e. to use (2)) the link capacity has also to be expressed in terms of packets per second. For computing the drop probability for the aggregated traffic, we have to use the buffer size given in terms of number of packets, the link capacity in terms of packets per second, and an aggregated traffic trace obtained by aggregating the real traffic trace using a particular time window $\Delta$.

Therefore, we can say that this algorithm has two steps, the first is unit translation[2] and the second is traffic aggregation. Differences between *real drop probability* $p_{\mathrm{p}}$ and *aggregated drop probability* $p_{\mathrm{pa}}(\Delta)$ can only be due to link capacity unit (byte per second or packets per second) and/or traffic representation (marked point process or aggregated process). In order to determine the relative impact of link capacity unit and traffic representation, we introduced a third drop probability: the unit translation drop probability $p_{\mathrm{pu}}$. The *unit translation drop probability* $p_{\mathrm{pu}}$ is obtained for the marked point process, the buffer size given in terms of number of packets and the link capacity in packets per second (and not the usual bytes per second unit). The unit translation drop probability $p_{\mathrm{pu}}$ will make possible to isolate possible differences due only to unit translation (and not to real traffic aggregation).

In the following, we will compare the three different drop probabilities: the first one is the real drop probability, the second is the drop probability obtained after the unit translation and the last one is the drop probability obtained for the aggregated traffic trace.

The comparison between those drop probabilities is made by using scattered plots [12]. Each point in the scattered plots represents drop probabilities obtained from two different computations. The $x$ axis represents $p_{\mathrm{pu}}$ or $p_{\mathrm{pa}}(\Delta)$. The $y$ axis represents the real drop probability $p_{\mathrm{p}}$ for all plots. The drop probabilities obtained for $x$ and $y$ coordinates are calculated for the same trace, link utilisation and buffer size. We used 55 different traces. For each trace we considered different link utilisation $\rho$ and queue buffer size $B$ pairs. In order to build the models we analysed 25 different $\rho$ and $B$ pairs, next in order to validate the obtained model we used another 25 $\rho$ and $B$ pairs. Therefore, in total we considered 1375 simulations in order to build the models and 1375 test simulations that were used only to validate the obtained results. Note that, 1375 simulations result in 1375 drop probability values obtained for specific traffic trace representation (for example aggregated traffic trace for which time window is 10 ms). Therefore, in order to obtain figure where two drop probabilities are compared (all Figs. from 2 to 7) we did 2750 simulations since each point represents two different drop probabilities.

Additionally, a linear function and $R^2$ coefficient are estimated. The coefficient of determination $R^2$ is a measure of variability of the data that is predicted by the estimated linear function [12]. In an ideal case, the drop probabilities are identical, the slope of the straight line is 1 and $R^2 = 1$. If the slope of the straight line is lower (resp. greater) than 1, the considered drop probability overestimates (resp. underestimates) the real drop probability. Estimating the real drop probability on the basis of the drop probability obtained for aggregated traffic is less precise for lower values of $R^2$. Analysis of the points positions can help to understand what type of error is achieved.

## 4 Aggregated traffic

The drop probability computed for aggregated traffic is the *aggregated drop probability* denoted by $p_{\mathrm{pa}}(\Delta)$. Recall that in order to compute $p_{\mathrm{pa}}(\Delta)$, the terms of (2) have to be expressed with the same units. It is important to note that link capacities are expressed in terms of bytes per second. Therefore, in order to compute drop probability for aggregated process (i.e. to use (2)) the link capacity has also to be expressed in terms of packets per second. Moreover, the traffic trace has to be aggregated over a time window $\Delta$. Both unit translation and aggregation window size $\Delta$ influence the obtained result. Since this influence is different in its origin we considered those factors in separate subsections.

### 4.1 Units translation

The real values of link capacity $C$ and buffer size $B$ are bytes per second and number of packets, respectively [21] from which the real drop probability is computed. However, in order to compute the aggregated drop probability with (2), the respective units have to be converted. $X_{\Delta}(i)$ being expressed in terms of number of packets the link capacity has therefore to be expressed in terms of packets per second (denoted by $C'$). The other solution could be considering byte aggregation instead of packet aggregation. Nevertheless, the byte aggregation is not very effective since the QoS parameter considered in this paper is the packet drop probability [22]. The packet drop probability cannot be computed from the byte aggregated process since for this process we only know how many bytes are dropped. Note that packets have different sizes and therefore we cannot easily convert number of dropped bytes to the number of dropped packets. That is why, in this paper we are using the term translation instead of conversion.

Translating the number of bytes into the number of packets is not easy since we do not know the *unit packet size*. Note that the packet size is not constant or even regularly distributed (see Fig. 1 or [6] for more details). Therefore the

[2] We are using term translation instead of conversion since converting queue buffer size from bytes to packets is not as simple as a conversion (see Sect. 4.1).

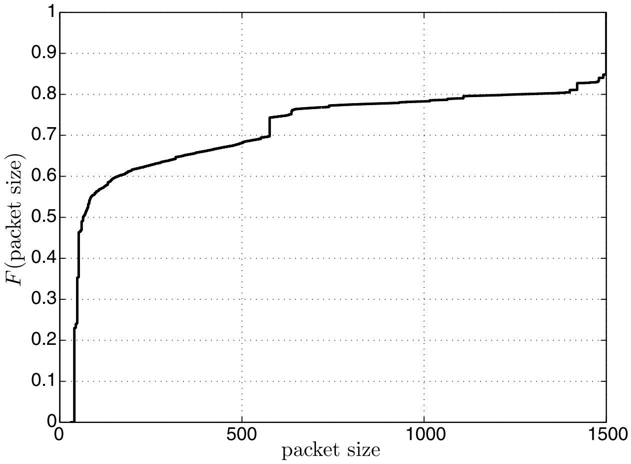

**Fig. 1** An example of the packet size cumulative distribution function obtained for a traffic trace (the analysed traffic trace contains 1 154 282 packets, its duration is 781 seconds and MTU size is 1500 bites)

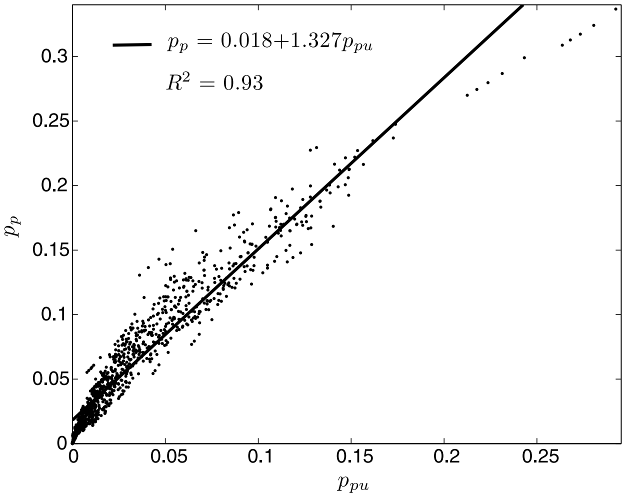

**Fig. 2** The scatter plot of the real drop probability $p_\mathrm{p}$ as a function of the unit translation drop probability $p_\mathrm{pu}$ (the link speed $C' = C/\bar{A}$)

unit packet size is not easy to compute or define. Note that depending on the definition we can obtain different values.

The natural and almost always used way for translating $C$ given in bytes per second unit into the packets per second unit consists in dividing the link speed by the average packet size i.e.

$$C' = \frac{C[\text{bytes per second}]}{\bar{A}[\text{average number of bytes per packet}]}, \tag{3}$$

where $\bar{A}$ is the mean packet size.

The unit translation can influence the results obtained for aggregated traffic. In order to analyse only how the unit translation influences the drop probability value, we computed the drop probability for link capacity $C'$ and real traffic trace i.e. we use exactly the same algorithm of the queue length computation as for the real trace (see Sect. 3) using the link capacity $C'$ instead of $C$. The drop probabilities obtained for $C'$ and the real trace are denoted $p_\mathrm{pu}$ and called *unit translation drop probability*. Note that $p_\mathrm{pu}$ was computed using the same algorithm as $p_\mathrm{p}$. The only difference is the use of $C'$ instead of $C$ as the link capacity value. In Fig. 2, the scatter plot of $p_\mathrm{p}$ as a function of $p_\mathrm{pu}$ is presented.

The slope of the straight line is 1.327 (see Fig. 2), i.e. after unit translation, the obtained drop probability is (in most cases) smaller than the real drop probability. As a consequence the $p_\mathrm{pu}$ underestimates the real drop probability. For example, for $p_\mathrm{pu} = 0.06$, the real drop probability $p_\mathrm{p} = 0.1$ (significant error of 40%). Moreover, the underestimation is stronger for smaller values of drop probability (see Fig. 2).

Why such huge difference were obtained is very interesting question. We cannot give a clear answer especially because for different traces the effect is different. It seems that the packet structure (like bursts' sizes or correlation between packets' sizes) can have a strong impact. On the basis of Fig. 1 we can conclude that for the real trace some large packet bursts must occur. Note that for a traffic containing mainly large packets, if we change the packet size to the mean packet size, we observe smaller drop probability. On the other hand it seems that there are no (or at least less than for large packets) bursts of small packets. Note that for small packets, changing theirs size to the mean packet size should theoretically increase the packet drop probability.

Note that since $p_\mathrm{pu}$ and $p_\mathrm{p}$ have been obtained for the same trace, the relationship between those two values "should" be linear. Nevertheless, the points in Fig. 2 make think we have a non-linear function $p_\mathrm{p}(p_\mathrm{pu})$. Since, it can be caused only by non-linear influence of incorrect unit translation we do not try to find a better function. Instead of that, in the next section, we are proposing some solutions which result in proper relationship.

### 4.2 Time window aggregation influence

In Appendix, two theorems are proved. The first one is:

**Theorem 1** $p_\mathrm{pa}(\Delta) \to p_\mathrm{pu}$ *for* $\Delta \to 0$.

The second one is:

**Theorem 2** $p_\mathrm{pa}(\Delta) \geq p_\mathrm{pa}(2\Delta)$.

Note that $p_\mathrm{pu}$ obtained for the case for which $C' = C/\bar{A}$ is already smaller than the real drop probability $p_\mathrm{p}$. Therefore, on the basis of Theorems 1 and 2 we can conclude that the aggregated drop probability $p_\mathrm{pa}(\Delta)$ is not higher than $p_\mathrm{pu}$. Since $p_\mathrm{pu}$ is already an underestimation of $p_\mathrm{p}$, the results obtained for the aggregated traffic cannot be better.

In Fig. 3 results obtained for four different time windows $\Delta$ are shown.

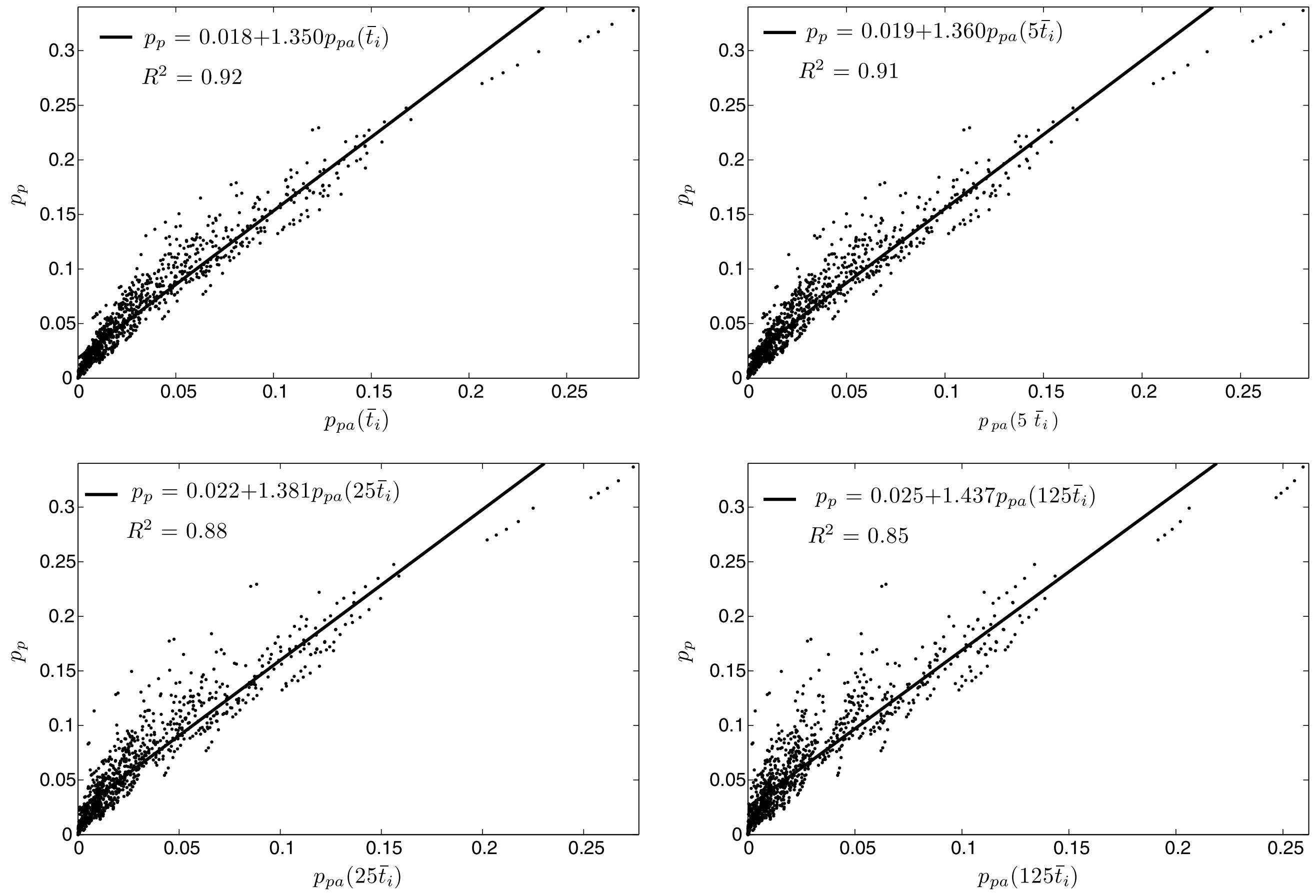

**Fig. 3** The scatter plot of the real drop probability $p_\text{p}$ as a function of the aggregated drop probability $p_\text{pa}(\Delta)$ where the link speed $C' = C/\bar{A}$. Each plot represents different aggregation window from $\bar{t}_i$ (i.e. mean inter-arrival time) to $125\bar{t}_i$

The results obtained for $\Delta = \bar{t}_i$ (i.e. mean inter-arrival time) are similar to those presented in Fig. 2, what is a consequence of Theorem 1. For the increasing values of $\Delta$, the obtained underestimation is stronger (higher slope of the straight line), what is a consequence of Theorem 2. Another consequence of using aggregated traffic is obtaining higher scattering (smaller $R^2$ value). Moreover, the underestimation obtained for smaller $p_\text{pa}$ is stronger. Therefore, we can conclude that using the aggregated drop probability as an estimation of the real drop probability is highly inaccurate. In order to correct this difference we propose a new way of computing the drop probability of the aggregated traffic traces.

## 5 Getting correct drop probability with aggregated traces

The difference between real and aggregated drop probability is caused by two factors: unit translation and aggregation window value. Moreover, on the basis of Theorems 1 and 2 we conclude that $p_\text{pu} \geq p_\text{pa}(\Delta)$ for any $\Delta$. Since $p_\text{pu}$ is underestimating $p_\text{p}$ we cannot find such $\Delta$ for which $p_\text{pa}(\Delta) = p_\text{p}$. Therefore, we proposed to use different link capacity unit translation. Instead of using $\bar{A}$ as a unit packet size we propose to use a *real unit packet size* $A_\text{u}$. The real unit packet size is such packet size, for which the drop probability obtained for the capacity $C_\text{u} = C/A_\text{u}$ is the same as the real drop probability.

The proposed definition of $A_\text{u}$ provides the equality between the real drop probability and the unit translation drop probability. Nevertheless, it does not determine $A_\text{u}$ value. Note, however, that the unit translation drop probability is an increasing function of $A_\text{u}$ since increasing $A_\text{u}$ decreases $C_\text{u}$. Moreover, for $A_\text{u} \to \infty$ we have $C_\text{u} \to 0$ and $p_\text{pu} \to 1$; on the other hand for $A_\text{u} \to 0$ we have $C_\text{u} \to \infty$ and $p_\text{pu} \to 0$. Therefore, we can determine such $A_\text{u}$ for which the unit translation drop probability is identical to the real drop probability. Nevertheless, such value will be unique only for particular trace and queue parameters. Therefore we estimate $A_\text{u}$ for many different traces and queue parameters and using a statistical model we determine $A_\text{u}$ as a function of a specific trace and queue parameters. In order to be more precise we did not use simple linear regression model but much more sophisticated model called Generalized Linear Model (GLZ) [10].

GLZ model, as any statistic model, has two types of variables: the explanatory variables (i.e. those variables that are used to compute the results) and response variable (which is in our case $A_{\mathrm{u}}$). Different $A_{\mathrm{u}}$ values were determined for specific traffic traces and network parameters. The traffic trace can be described by numerous different parameters from the mean packet size (simple one) to Hurst parameter (much more complex). Since we would like to build a possibly simple model we analysed the mean packet size $\bar{A}$ and the average inter-arrival time $\bar{t}_i$. Analysing results, we realised that the network parameters (i.e. buffer size $B$ and link utilisation $\rho$) are influencing $A_{\mathrm{u}}$, too. Therefore, those values were also used as explanatory variables. It would be possible to add numerous different variables describing traffic trace properties. Nevertheless, the obtained results were unexpectedly good (and completely overpassing those obtained for $\bar{A}$) that we did not consider more complicated models.

Since we computed many real unit packet size values (over 500), we can denote them by $\mathbf{A}_{\mathrm{u}}$ (where $\mathbf{A}_{\mathrm{u}}$ is a vector of size $n$, where $n$ is the amount of obtained $A_{\mathrm{u}}$ values). $A_{\mathrm{u}}$ values were obtained for different explanatory variables denoted, for simplification, by a matrix $\mathbf{X}$ (where $\mathbf{X}$ is a matrix of size $n \times k$, where $k$ is the amount of explanatory variables). On the basis of the GLZ we can write [10]

$$g(E\mathbf{A}_{\mathrm{u}}) = \mathbf{X}\beta \tag{4}$$

where $E$ denotes expected value, $\beta$ is a vector of the estimated parameters ($\beta$ size is $k$) and $g(x)$ is a *link function*.

The GLZ model enables estimating parameters even if the response distribution is different from the normal distribution, for example Gamma, Multinomial or Binomial [10]. Moreover, numerous link functions can be considered. The link function is a function of response variable such as identical, exponential, logit, etc. Note that, using link functions (specific functions can be used in case of particular distribution) helps to obtain more precise results since, even for nonlinear problems, the link function transformation can result in a linear function of the explanatory variables [10]. Note however, that by linear function we understand that the estimated parameters are linear and we can use for example the square of an explanatory variable.

The obtained equation is estimated from data using statistic package R [20]. An iterative three-stage modelling is used [3]. The first step is to propose a model by choosing some explanatory variables that can influence the response variable. Note that we cannot know exactly which explanatory variables influence the response variable a priori. Therefore, we can say that the first step relies on a kind of guess. The second step deals with estimating the model parameters $\beta_i$. Note that some of the $\beta_i$ can be 0 with high probability, in our case 95%; in such a case we say that the $i$th explanatory variable is not statistically significant. If any explanatory variable is not statistically significant then we go back to the first step and remove such explanatory variable from the model. As a consequence the final model is composed only with parameters that are statistically significant.

For example, we analysed a model where explanatory variables were $\rho$, $\rho^2$, $\frac{B}{100}$, $\rho\frac{\bar{A}}{1000}$ and $(\frac{\bar{A}}{1000})^2$. Nevertheless, the $\frac{B}{100}$ was not statistically significant. It then does not appear in the final model.

Many different models with all statistically significant parameters can be estimated. In order to decide which of the statistically significant models is the best (i.e. describes data the most precisely and has the lowest number of parameters), information criteria were proposed [1]. In this research we used the Schwarz Information Criterion (SIC).

The final equation is

$$\begin{aligned} A_{\mathrm{u}} = 820.2 - 966.5\rho + 197.1\rho^2 \\ + 450.2\left(\frac{\bar{A}}{1000}\right)^2 + 769.7\rho\frac{\bar{A}}{1000}. \end{aligned} \tag{5}$$

Since, (5) was obtained on the basis of a data set that is limited by definition the parameters for which this equation was computed are limited. The packet mean value for the analysed traces is $\bar{A} \in (350, 800)$ since such values were obtained for the analysed traces. Note that it is not very probable that a mean packet size can be strongly lower or higher than these range limits, since the smallest packet is 40 bytes and the largest 1500 bytes. The link utilisation and buffer size are $\rho \in (0.25, 0.75)$ and $B \in (20, 100)$. The justification for these ranges were described in Sect. 3.

The parameters of (5) are then limited to the most important values considering the real network point of view. Note

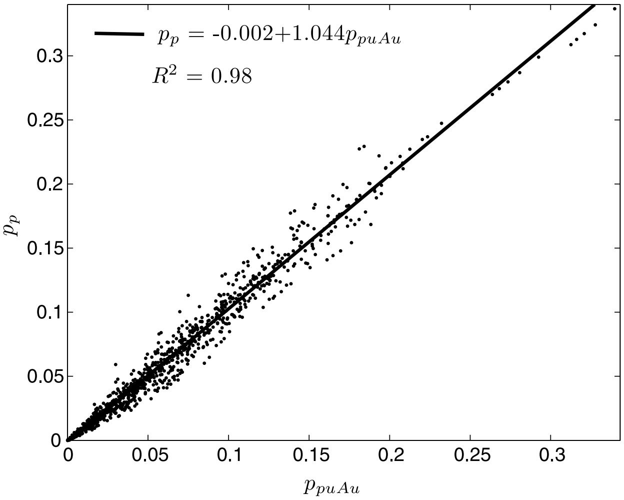

**Fig. 4** The scatter plot of the real drop probability $p_{\mathrm{p}}$ as a function of the unit translation drop probability $p_{\mathrm{puAu}}$ obtained for the proper link capacity $C_{\mathrm{u}} = C/A_{\mathrm{u}}$

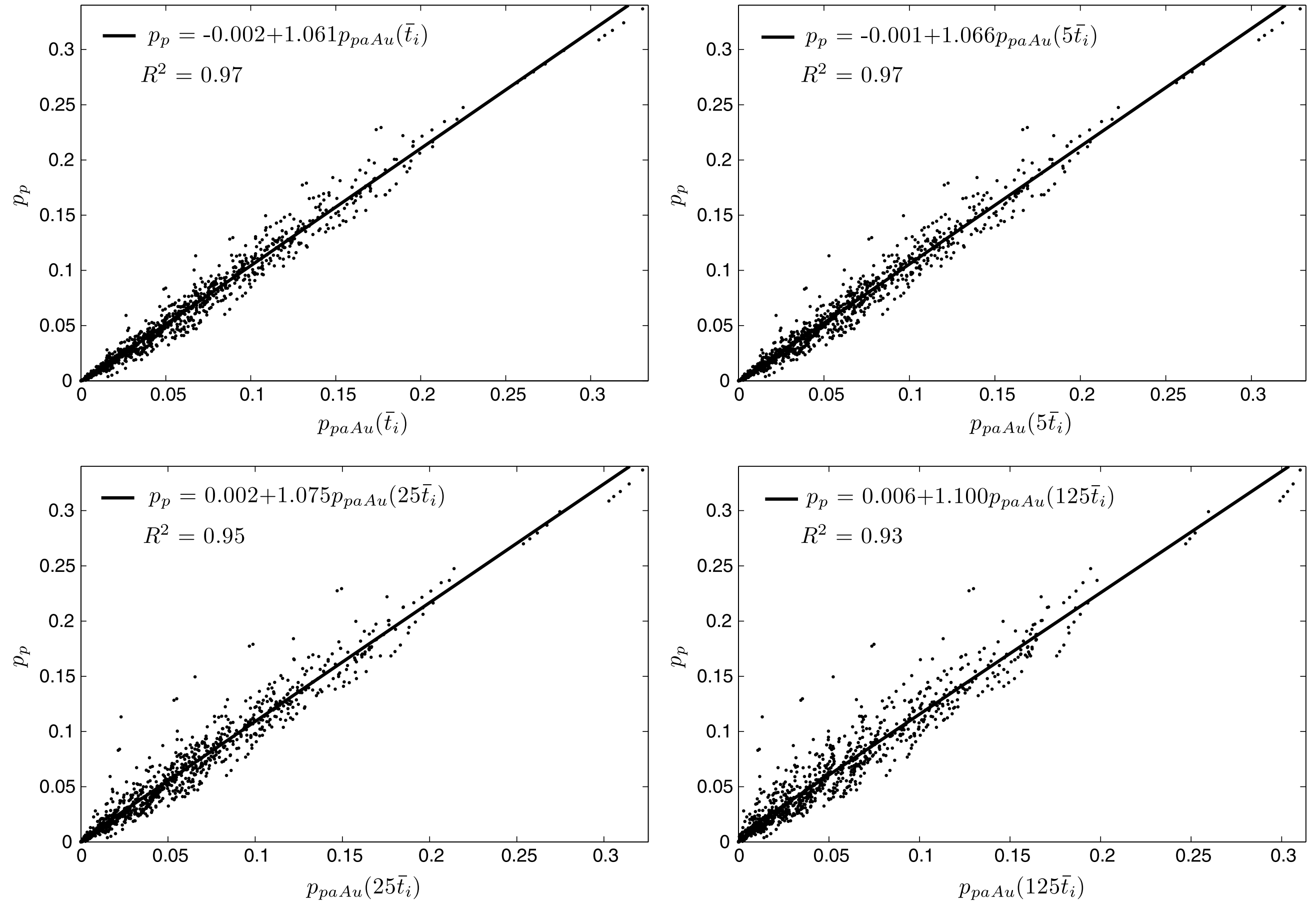

**Fig. 5** The scatter plot of the real drop probability $p_\mathrm{p}$ as a function of the aggregated drop probability $p_{\mathrm{paAu}}(\Delta)$ where the link speed is $C_\mathrm{u} = C/\bar{A}_\mathrm{u}$. Each plot represents different aggregation window from $\bar{t}_i$ (i.e. mean inter-arrival time) to $125\bar{t}_i$

that in the real networks, buffer sizes are small [21] and the link utilisation is low [2]. Therefore, considering values out of our considered range is questionable in practice. Moreover, if the network parameters are such that the real drop probability is 0 then $A_\mathrm{u}$ is ambiguous and considering drop probability is questionable. Therefore, the only limitation of our results is related to different mean packet size and higher link utilisation. The packet size should not differ strongly from the values considered in our research since we analysed measurements obtained for different operational networks and day times. Considering higher link utilisation, as already said, is of very limited importance since in real networks, the link utilisation is low [2].

For the new unit translation (where packet size is given by $A_\mathrm{u}$) the unit translation drop probability is computed and denoted by $p_{\mathrm{puAu}}$. In Fig. 4, the scatter plot of new unit translation drop probability is shown. Note that traffic parameters used in order to obtain these results are different than those used to obtain (5).

The results obtained for $A_\mathrm{u}$ packet size are close to ideal. Firstly, the slope is almost 1, the intercept is close to 0 and $R^2$ is very high. Secondly, the scatter of the results obtained for smaller drop probability is smaller, too. As a consequence the obtained absolute error does not significantly depend on the obtained drop probability value as it was for the $p_{\mathrm{pu}}$.

The second factor influencing aggregated drop probability is time window $\Delta$. Therefore, in Fig. 5 results obtained for $A_\mathrm{u}$ unit packet size and different aggregation windows are shown.

The obtained results show much higher accuracy obtained for link capacity $C_\mathrm{u} = C/A_\mathrm{u}$ and different aggregation windows than those obtained for $C' = C/\bar{A}$. As we expected $p_\mathrm{p}$ underestimation is higher for higher $\Delta$. Nevertheless, for $p_{\mathrm{pa}}(\Delta)$, the difference between slopes of the linear functions obtained for $\Delta = \bar{t}_i$ and $\Delta = 125\bar{t}_i$ is higher (0.087) than for $p_{\mathrm{paAu}}(\Delta)$ (more than two times less: 0.039). Moreover, the obtained results are scattered symmetrically around the linear function (except for few points). Therefore, the conclusion obtained for the aggregated traffic can be applied to the real drop probability. Note that even if $R^2$ is decreasing for higher $\Delta$ the value obtained for the highest $\Delta$ is smaller than the one obtained for the smallest $\Delta$ in case of $p_{\mathrm{pa}}(\Delta)$.

The only disadvantage of (5) is its slight complexity and dependence on $\rho$ that can change. Therefore, we considered another model where the only explanatory variable that we used was $\bar{A}$. Note that higher orders of $\bar{A}$ polynomials were considered but the optimal (base on SIC) is given by

$$A'_{\mathrm{u}} = 196.7 + 0.915\bar{A}. \tag{6}$$

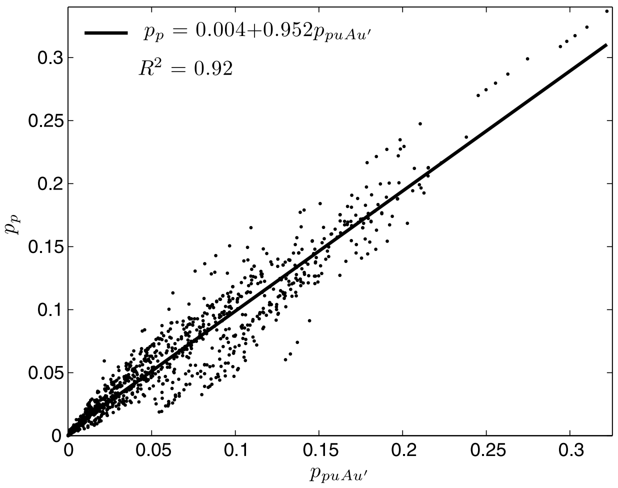

**Fig. 6** The scatter plot of the real drop probability $p_{\mathrm{p}}$ as a function of the unit translation drop probability $p_{\mathrm{pu}A\mathrm{u}'}$ obtained for the proper link capacity $C_{\mathrm{u}} = C/A'_{\mathrm{u}}$

The drop probability obtained for unit packet size $A'_{\mathrm{u}}$ is denoted by $p_{\mathrm{pu}A\mathrm{u}'}$. The comparison between $p_{\mathrm{pu}A\mathrm{u}'}$ and the real drop probability $p_{\mathrm{p}}$ is presented in Fig. 6.

For the simple solution $A'_{\mathrm{u}}$ the obtained results are much more linear and more randomly scattered around the linear regression than those obtained for the link capacity $C' = C/\bar{A}$. The slope of the straight line is much closer to 1. $R^2$ is just slightly smaller. Moreover, the obtained result is overestimation of $p_{\mathrm{p}}$. Therefore, on the basis of Theorem 2, we expect the obtained results for the aggregated traffic to be even better. In Fig. 7 results obtained for $A'_{\mathrm{u}}$ unit packet size and different aggregation windows are shown.

The obtained result is very accurate according to the simplicity of (6). The relationship between the estimated real drop probability $p_{\mathrm{p}}$ and the one that we can easily model $p_{\mathrm{pa}A\mathrm{u}'}(\Delta)$ is linear. Therefore, any conclusions ob-

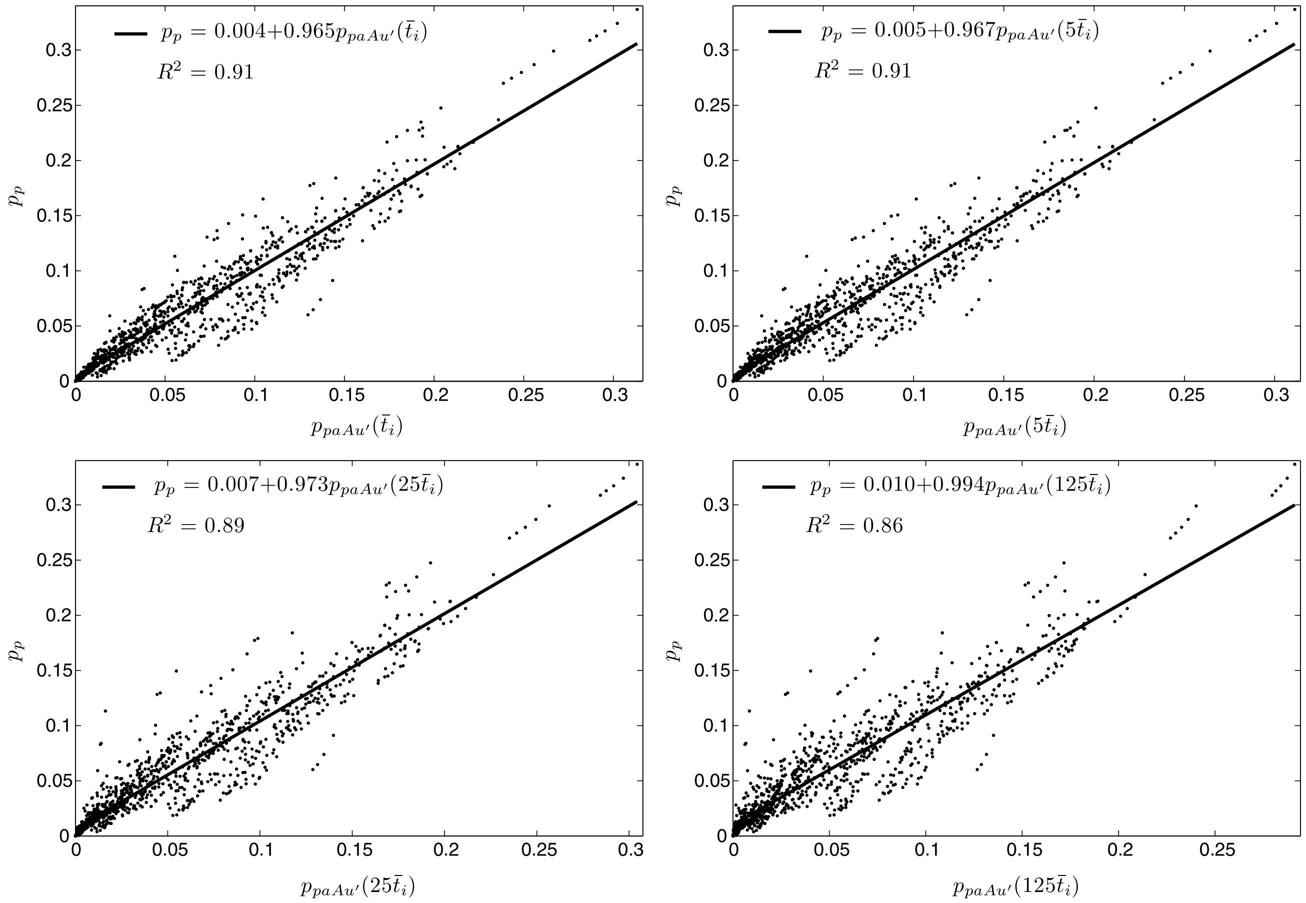

**Fig. 7** The scatter plot of the real drop probability $p_{\mathrm{p}}$ as a function of the aggregated drop probability $p_{\mathrm{paAu'}}(\Delta)$ where the link speed is $C_{u'} = C/\bar{A}'_{\mathrm{u}}$. Each plot represents different aggregation window from $\bar{t}_i$ (i.e. mean inter-arrival time) to $125\bar{t}_i$

tained for the aggregated traffic trace can be applied to the real traffic trace. The only disadvantage is high scattering of the obtained results (much higher than those obtained for $p_{\mathrm{pa}A\mathrm{u}}(\varDelta)$).

The obtained $p_{\mathrm{pa}A\mathrm{u}}(\varDelta)$ and $p_{\mathrm{pa}A\mathrm{u}'}(\varDelta)$ are close to the real value $p_{\mathrm{p}}$ and can be used to estimate the real drop probability since relationships $p_{\mathrm{p}}(p_{\mathrm{pa}A\mathrm{u}}(\varDelta))$ and $p_{\mathrm{p}}(p_{\mathrm{pa}A\mathrm{u}'}(\varDelta))$ are linear. Note that the $p_{\mathrm{p}}(p_{\mathrm{pa}}(\varDelta))$ is not linear since for small $p_{\mathrm{pa}}(\varDelta)$, the increase is much faster than the one obtained for larger $p_{\mathrm{pa}}(\varDelta)$. In Fig. 3, it was shown that the drop probability obtained for the aggregated traffic trace can significantly differ from the real drop probability. Since, numerous traffic analysis are focusing on the aggregated traffic trace, the conclusions that where obtained on the basis of the aggregated traffic trace can be different from the real traffic behaviour. We showed that it is possible to analyse drop probability of the aggregated traffic and obtain results that are close to the real drop probability by using (5) or (6).

## 6 Practical usage of obtained results

Equations (5) and (6) have been estimated on the basis of traffic traces analysis. Nevertheless, any traffic trace analysis is limited because of the limited characteristics of the traces that were used. Therefore, one can question whether we can observe such effect for different traces.

We used special techniques to estimate the equations. Moreover, the obtained equations were tested by comparing the values obtained from equations with values obtained for various queue parameters, the used parameters being different than those used in the equations estimation. Therefore, we are quite sure that the obtained equations will hold for similar traces i.e. traces measured on high speed link with large aggregation level.

Nevertheless, it is not possible to check any network conclusions or effects on any Internet link, simply because only some of the data are available. Therefore, our future work will focus on testing the obtained equations on different traffic traces. Moreover, Internet traffic is changing very fast; we then would like to test the obtained equations in the future in order to check if they are still valid.

We know that the way we obtained (5) and (6) is empirical and not theoretical. The reason is very simple. The complex nature of packet arrival time and packet size distributions makes it very difficult to find a formal proof of what we should do to model aggregated traffic correctly. The complexity is so high that we can find numerous different publications dealing only with packet arrival time distribution. Note that some of those publications are contradictory! Therefore, we preferred to find an empirical model limited to the available traces than rely on particular assumptions about traffic trace nature.

We hope that in future research it will be possible to find $A_{\mathrm{u}}$ using more formal way. Nevertheless this task is difficult and probably, at the beginning, it will be solved for some specific cases only.

## 7 Conclusions

The presented results reveal that estimating the packet drop probability on the basis of the aggregated traffic, for network engineering purposes for instance, can occur in large underestimation of the real packet drop probability. Such results were obtained for different traffic traces and, most importantly, with realistic traffic parameters (low link utilisation and small buffers).

In this paper, we proposed a way to make the drop probability obtained for the aggregated traffic similar to the real drop probability. The proposition is based on determining and using a specific unit packet size for which the link capacity given in packets per second is computed. There are two solutions proposed: one is very accurate (but a bit complicated), whereas the second is very simple, but of course less precise (but still much better than solutions proposed in previous work). The obtained equations were estimated from simulations that, such as any simulations, can be run only for limited ranges of parameters. Therefore, the obtained equations should be used for values that are in those specific ranges. Nevertheless, the considered ranges of network parameters $\rho$ and $B$ are the most common and important ones from the real network point of view. Moreover, the traffic parameters were represented by different traffic traces measured at different time and in different networks. Therefore, we argue that the proposed solutions are quite general and can be used by network engineers and researchers.

In this paper a solution for aggregating traffic traces in order to get an aggregated loss probability similar to the real drop probability (see (5) or (6)) was proposed. Since drop probability is not the only QoS parameter, in the future work, we would like to consider some other QoS parameters as delay or jitter for example. This work will be focused on determining appropriate solutions (similar to (5) or (6)) for such parameters. The final step of the analysis will be finding a common range of aggregation scale and unit translation for aggregated traces which makes all QoS and performance parameters simultaneously similar to the real ones.

At this point, we, at least, showed that using aggregated traces in networking for performance evaluation has to be made very carefully.

## Appendix: The theorem proofs

**Theorem 1** $p_{\mathrm{pa}}(\varDelta) \to p_{\mathrm{pu}}$ *for* $\varDelta \to 0$.

*Proof* In this proof, we assume that any two events are separated by some time, i.e. two events cannot happen at the same time. Note that from a technical point of view this assumption is true since a single queue is served by a single processor that makes single operation at once.

In order to prove this theorem, it is enough to focus on the queue behaviour, since dropping can occur only if queue is full. Note that in order to compare queues obtained for the real trace and the aggregated trace both link capacities have to be given in packets per time unit (for example packet per second). If link capacity is given in bytes per second and we consider packet aggregated process, we are not able to use (2) and automatically we cannot analyse such a system.

The queue length fed by the real trace can increase only when a new packet arrives (i.e. at time $t_i$) and decrease when a packet is served (let denote it by $\tau_i$). Therefore, we focus on how different, from the real traffic, the arrival and service moments for the aggregated trace are. The aggregated traffic trace is a random variable $X_n$ describing how many packets arrived at time $[(n-1)\Delta, n\Delta)$. Since we do not know when exactly the packets arrived or were served, we can assume that all of them arrived or were served at time $(n-1)\Delta$. Therefore, aggregating process can shift events for time not greater than $\Delta$.

Let choose $\Delta = \min_{i,k}(|l_i - l_{i-k}|)/2$ where $l$ is $t$ or $\tau$ and $t_i - t_i$ and $\tau_i - \tau_i$ are excluded. Note that for such $\Delta$, we know that in each time interval only one event can happen. Therefore, if for the real trace a packet arrives and the queue is full (i.e. the packet will be dropped), for the aggregated traffic it has to be full as well. The reason is that it is impossible that the next service is shifted for more than $\Delta$. We know by $\Delta$ definition that shifting for less than $\Delta$ does not change anything for the next event. The same conclusion can be made for non full queue (i.e. in case the packet will be stored). Since the same packets will be dropped or stored for the real and aggregated traces we have to obtain the same drop probability. This finishes the proof. □

**Theorem 2** $p_{\mathrm{pa}}(\Delta) \geq p_{\mathrm{pa}}(2\Delta)$.

*Proof* In order to prove this theorem we consider two processes $X_i$ and $Y_i$ which are packet aggregations obtained for the same real trace and different aggregation windows $2\Delta$ and $\Delta$, respectively. Moreover, we use separate notation for queue length ($Q_i$ and $U_i$) and amount of dropped packets ($S_i$ and $T_i$). To make it clearer see Fig. 8.

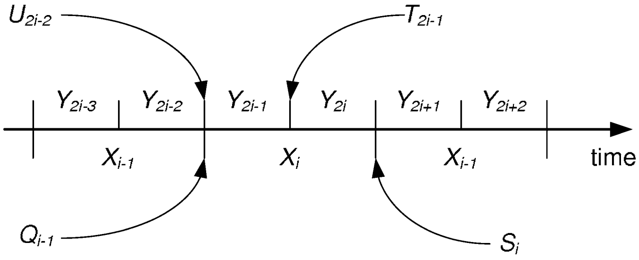

**Fig. 8** The notation used in the proof. $X_i$ and $Y_{2i-1} + Y_{2i}$ are the amount of packets arrived at $i$th interval; $Q_i$ and $U_{2i}$ are queue length at the end of $i$th interval. $S_i$ and $T_{2i} + T_{2i-1}$ are the amount of packets dropped at $i$th interval

Since both those processes are obtained from the same real traffic we have

$$\sum_{i=1}^{N} X_i = \sum_{i=1}^{2N} Y_i \tag{7}$$

where $2N\Delta$ is the trace duration.

Note that $p_{\mathrm{pa}}(2\Delta)$ is estimated as a ratio between the number of dropped packets and the number of sent packets. Therefore, we have

$$p_{\mathrm{pa}}(2\Delta) = \frac{\sum_{i=1}^{N} S_i}{\sum_{i=1}^{N} X_i} \tag{8}$$

and

$$p_{\mathrm{pa}}(\Delta) = \frac{\sum_{i=1}^{2N} T_i}{\sum_{i=1}^{2N} Y_i}. \tag{9}$$

Since in (8) and (9) denominators equal (see (7)), we can change the problem of showing that $p_{\mathrm{pa}}(\Delta) \geq p_{\mathrm{pa}}(2\Delta)$ to the problem of showing that

$$\sum_{i=1}^{2N} T_i \geq \sum_{i=1}^{N} S_i. \tag{10}$$

In order to prove inequality (10) we have to prove two lemmas.

**Lemma 1**

$$\left.\begin{array}{l} Q_{i-1} = U_{2i-2} + a, \\ Q_i = U_{2i} + b, \\ S_i > 0 \end{array}\right\} \quad \Rightarrow \quad S_i \leq T_{2i-1} + T_{2i} + a - b. \tag{11}$$

*Proof* In order to prove Lemma 1 we show that it is impossible that

$$S_i > T_{2i-1} + T_{2i} + a - b. \tag{12}$$

Note that $S_i$ (respectively $T_i$) is given by

$$S_i = \max(0, X_i + Q_{i-1} - 2C\Delta - B). \tag{13}$$

Note that $S_i > 0 \Rightarrow Q_i = B$ (see (2) and (13)). Therefore we can rewrite inequality (12) (according to notation presented in Lemma 1 $a = Q_{i-1} - U_{2i-2}$ and $b - B = -U_{2i}$

and since $X_i = Y_{2i-1} + Y_{2i}$ (see Fig. 8)) and we get the equation

$$Y_{2i-1} + U_{2i-2} - C\Delta + Y_{2i} - C\Delta - U_{2i} > T_{2i-1} + T_{2i}. \quad (14)$$

Note that left-hand side of inequality (14) has three parts. Part 1: $Y_{2i-1} + U_{2i-2} - C\Delta$, Part 2: $Y_{2i} - C\Delta$ and Part 3: $-U_{2i}$. We will recall those parts during the proof.

Let assume that $T_{2i-1} = T_{2i} = 0$ and $U_{2i-1} = 0 \Rightarrow Y_{2i-1} - C\Delta + U_{2i-2} < 0$. Part 1 is then negative so we can assume that it is 0 (note that our task is to show that the left-hand side of inequality (14) is not higher than the right-hand side, therefore we always can overestimate some factors of the left-hand side). Part 2 is positive and equals $U_{2i}$ or is negative. Therefore, in both cases the left-hand side is not higher than 0. Then, for $T_{2i-1} = T_{2i} = 0$ and $U_{2i-1} = 0$, inequality (14) cannot be true.

Let assume that $T_{2i-1} = T_{2i} = 0$ and $U_{2i-1} > 0$. Since $T_{2i-1} = 0$ we have $Y_{2i-1} - C\Delta + U_{2i-2} < B$. Then Part 1 equals $U_{2i-1}$ and since Part 2 + Part 1 = $U_{2i}$ or the sum is negative. Therefore, for $T_{2i-1} = T_{2i} = 0$ and $U_{2i-1} > 0$ inequality (14) cannot be true.

Let assume that $T_{2i-1} > 0 \wedge T_{2i} = 0$. Since $T_{2i-1} > 0 \Rightarrow T_{2i-1} = Y_{2i-1} + U_{2i-2} - C\Delta - B \wedge U_{2i-1} = B$. We can rewrite inequality (14) and get

$$\begin{aligned} &Y_{2i-1} + U_{2i-2} - C\Delta + Y_{2i} - C\Delta - U_{2i} \\ &\quad > Y_{2i-1} + U_{2i-2} - C\Delta - B. \end{aligned} \quad (15)$$

After algebraic transformations we have $Y_{2i} + B - C\Delta > U_{2i}$. Note that the left-hand side of this inequality is $U_{2i}$ or 0. Therefore, for $T_{2i-1} > 0 \wedge T_{2i} = 0$, inequality (14) cannot be true.

Let assume that $T_{2i-1} = 0 \wedge T_{2i} > 0 \wedge U_{2i-1} = 0$. Since $T_{2i} > 0 \wedge U_{2i-1} = 0 \Rightarrow T_{2i} = Y_{2i} - C\Delta - B \wedge U_{2i} = B$. We can rewrite inequality (14) and get

$$Y_{2i-1} + U_{2i-2} - C\Delta + Y_{2i} - C\Delta - B > Y_{2i} - C\Delta - B \quad (16)$$

what is in contradiction with assumption that $U_{2i-1} = 0$. Therefore, for $T_{2i-1} = 0 \wedge T_{2i} > 0 \wedge U_{2i-1} = 0$ inequality (14) cannot be true.

Let assume that $T_{2i-1} = 0 \wedge T_{2i} > 0 \wedge U_{2i-1} > 0$. Since $T_{2i-1} = 0 \Rightarrow U_{2i-1} \le B$ and $T_{2i} > 0 \wedge U_{2i-1} \le B \Rightarrow T_{2i} = Y_{2i} + U_{2i-1} - C\Delta - B \wedge U_{2i} = B$. Therefore we can rewrite inequality (14) and get

$$Y_{2i-1} + U_{2i-2} - C\Delta > U_{2i-1} \quad (17)$$

what is in contradiction with the assumptions since the left-hand side of the above inequality is $U_{2i-1}$ or 0. Therefore, for $T_{2i-1} = 0 \wedge T_{2i} > 0 \wedge U_{2i-1} > 0$ inequality (14) cannot be true.

The last assumption that we can make is $T_{2i-1} > 0 \wedge T_{2i} > 0$. In this case we can rewrite inequality (14) and get

$$\begin{aligned} &Y_{2i-1} + U_{2i-2} - C\Delta + Y_{2i} - C\Delta - B \\ &\quad > Y_{2i-1} + U_{2i-2} - C\Delta - B + Y_{2i} + B - C\Delta - B. \end{aligned} \quad (18)$$

After algebraic transformations, we obtain contradiction with the assumption. Therefore, for all possible values of $T_{2i-1}$ and $T_{2i}$, inequality (14) cannot be true. This completes the proof. □

**Lemma 2**

$$\left.\begin{array}{l} Q_{i-1} = U_{2i-2} + a, \\ Q_i = U_{2i} + b, \\ S_i = 0, \\ b > 0 \end{array}\right\} \quad \Rightarrow \quad T_{2i-1} + T_{2i} \ge b - a. \quad (19)$$

*Proof* Since $b > 0$ we know that $Q_i > 0$, since $S_i = 0$ we know that $X_i + Q_{i-1} - 2C\Delta \le B$. Therefore we know that $Q_i = Q_{i-1} + X_i - 2C\Delta$.

Again the proof will be made by showing that in any case equation

$$T_{2i-1} + T_{2i} < b - a \quad (20)$$

cannot be true.

Using $a$ and $b$ definitions, knowing that $0 < Q_i \le B$ and $X_i = Y_{2i-1} + Y_{2i}$ (see Fig. 8) we can rewrite above equation as

$$Y_{2i-1} + U_{2i-2} - C\Delta + Y_{2i} - C\Delta - U_{2i} > T_{2i-1} + T_{2i} \quad (21)$$

Note that inequality (14) and (21) are identical, and since proof proposed in Lemma 1 can be repeated here, Lemma 2 is true. □

Note that in order to prove the inequality (10), it is enough, if we show that $S_i \le T_{2i-1} + T_{2i}$. On the basis of Lemma 1, it is true for $a \le b$. Therefore, we have to show that if

$$S_i - (T_{2i-1} + T_{2i}) = \alpha > 0 \quad (22)$$

we have

$$\sum_{j=1}^{i-1} S_j - \sum_{j=1}^{2i-2} T_j = \beta \ge \alpha. \quad (23)$$

The condition described by (22) and (23) means: it is possible that the process $X$ lost more packets at $i$th time interval than the $Y$ process at the same time (i.e. at $2i - 1$th and $2i$th time intervals). Nevertheless, in such a case we know that the number of all the previous losses of the $Y$ process (i.e. $\sum_{j=1}^{2i-2} T_j$) is larger than the number of all the previous losses of the $X$ process (i.e. $\sum_{j=1}^{i-1} S_j$). Moreover, we know that $\sum_{j=1}^{i} S_j$ cannot be higher than $\sum_{j=1}^{2i} T_j$ (it was needed to prove Theorem 2).

Note that for $S_i > 0$ we have $Q_i = B$ and therefore, in Lemma 1, we have $b \geq 0$. Since the interesting situation is for $a > b$, therefore we know that $a > 0$. Since we start from the empty queue we have to have such a situation for which $a \leq b$ (at least $a = b = 0$). Note that in this case

$$T_{2i-1} + T_{2i} - S_i \geq b - a = \beta. \tag{24}$$

It means that $Y$ lost $b - a$ more packets than $X$. So in order to obtain value $b$ from the moment where $a = 0$, process $Y$ has to lose at least $b$ packets. Note that this conclusion is true for $S_i > 0$ on the basis of Lemma 1 and for $S_i = 0$ on the basis of Lemma 2.

So in order to get $Q_i > U_{2i}$, the process $Y$ had to lose more packets than the process $X$. Moreover, $Y$ had to lose at least $Q_i - U_{2i} = \alpha$ more packets than $X$ and the $X$ lost given by $S_i - (T_{2i-1} + T_{2i})$ cannot be higher than $\alpha$. Therefore, we know that $S_i \leq T_{2i-1} + T_{2i}$ or $\sum_{j=1}^{i-1} S_j - \sum_{j=1}^{2i-2} T_j \geq S_i - (T_{2i-1} + T_{2i})$. This completes the proof of Theorem 2. □

The prove of Theorem 2 is quite complicated and it seems that one should be able to find a simpler one. Firstly we considered this theorem as obvious! Nevertheless, for any simplification of this prove we could always find such combination of incoming packets that the prove was incorrect.

An interesting fact is that it should be possible to prove that $p_{\text{pa}}(\Delta) \geq p_{\text{pa}}(a\Delta)$ for any $a \geq 1$. Nevertheless, we found this case much more difficult and since Theorem 2 is enough we did not considered it.

## References

1. Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, *19*(6), 716–723.
2. Bhattacharyya, S., Diot, C., & Jetcheva, J. (2001). Pop-level and access-link-level traffic dynamics in a tier-1 pop. In *IMW '01: proceedings of the 1st ACM SIGCOMM workshop on Internet measurement* (pp. 39–53). New York: ACM.
3. Box, G. E. P., & Jenkins, G. M. (1976). *Time series analysis: forecasting and control*. San Francisco: Holden-Day.
4. Cleary, J., Donnelly, S., Graham, I., McGregor, A., & Pearson, M. (2000). Design principles for accurate passive measurement. In *PAM2000 passive and active measurement workshop* (pp. 1–7), May 2000.
5. Crovella, M., & Bestavros, A. (1996). Self-similarity in world wide web traffic: Evidence and possible causes. *Performance Evaluation Review*, *24*(1), 160–169. Also, in *Proceedings of SIGMETRICS'96: the ACM international conference on measurement and modeling of computer systems*, Philadelphia, Pennsylvania, May 1996.
6. Fraleigh, C., Moon, S., Lyles, B., Cotton, C., Khan, M., Moll, D., Rockell, R., Seely, T., & Diot, C. (2003). Packet-level traffic measurements from the sprint ip backbone. *IEEE Network*, *17*(6), 6–16.
7. Garrett, M. W., & Willinger, W. (1994). Analysis, modeling and generation of self-similar VBR video traffic. In *SIGCOMM* (pp. 269–280).
8. Janowski, L., Ziegler, T., & Hasenleithner, E. (2006). A scaling analysis of umts traffic. In *NEW2AN* (pp. 211–222).
9. Leland, W. E., Taqqu, M. S., Willinger, W., & Wilson, D. V. (1994). On the self-similar nature of Ethernet traffic (extended version). *IEEE/ACM Transactions on Networking*, *2*(1), 1–15.
10. McCullagh, P., & Nelders, J. (1991). *Generalized linear models* (2nd edn.). London: Chapman & Hall.
11. METROSEC (2005). *Metrology for security and quality of service*. http://www.laas.fr/METROSEC/.
12. NIST (2002). *NIST/SEMATECH e-handbook of statistical methods*. http://www.itl.nist.gov/div898/handbook.
13. NLANR (2003). *Measurement & network analysis, Auckland-VIII*. http://pma.nlanr.net/Special/auck8.html.
14. Norros, I. (1994) *A storage model with self-similar input*.
15. *Packet statistics for Auckland-VIII 20031209*, 2003. http://pma.nlanr.net/Special/auck8/20031209-p.html.
16. Park, K., & Willinger, W. (2000). *Self-similar network traffic and performance evaluation*. New York: Wiley.
17. Prasad, R. S., Dovrolis, C., & Thottan, M. (2007). Router buffer sizing revisited: the role of the output/input capacity ratio. In J. Kurose & H. Schulzrinne (Eds.), *CoNEXT* (p. 15). New York: ACM.
18. Scherrer, A., Larrieu, N., Borgnat, P., Owezarski, P., & Abry, P. (2007). Non-Gaussian and long memory statistical characterisations for Internet traffic with anomalies. *IEEE Transactions on Dependable and Secure Computing*, *4*(1), 56–70.
19. Sheluhin, O., Smolskiy, S., & Osin, A. (2007). *Self-similar processes in telecommunications*. New York: Wiley.
20. *The R project for statistical computing*. (2008). http://www.r-project.org/.
21. *Troubleshooting output drops with priority queueing* (August 2005). http://www.cisco.com/warp/public/105/priorityqueuedrops.pdf.
22. Wang, Z. (2001). *Internet QoS—architecture and mechanisms for quality of service*. San Mateo: Morgan Kaufmann.

**Lucjan Janowski** is an assistant of professor at Department of Telecommunications (AGH University of Science and Technology). He received the M.Sc. degree in Telecommunication in 2002 and Ph.D. degree in Telecommunications in 2006 both from the AGH University of Science and Technology. During 2007 he worked on a post-doc position in CNRSLAAS (Laboratory for Analysis and Architecture of Systems of CNRS) in France where he was preparing both malicious traffic analysis (under a Gamma-FARIMA assumptions) and anomaly detection algorithms. His main interest is statistics and probabilistic modelling of traffic traces. He has been participating in several commercial and scientific projects. He is an author of several papers.

**Philippe Owezarski** is a full time researcher of CNRS (the French Center for Scientific Research), working at LAAS (Laboratory for Analysis and Architecture of Systems), in Toulouse, France. He got a Ph.D. in computer science in 1996 from Paul Sabatier University, Toulouse III. His main interests deal with high speed and multimedia networking and more specifically on IP networks monitoring, and Quality of Service and security enforcement based on measurements. During year 2000, he spent 9 months working for Sprint ATL in Burlingame, California. There he has been working on the Sprint monitoring IPMON project, and focused mainly on actual TCP flows analysis. Back to LAAS, Philippe Owezarski has been one of the main contributor of a monitoring project in France—METROPOLIS, has been leading a French steering group on IP networks monitoring, and has been leading the French MetroSec project aiming at increasing the robustness of the Internet against DoS and DDoS attacks. Now, he is contributing to the European COST-TMA and ECODE projects which propose to use monitoring as the main support for enforcing QoS optimization and security mechanisms in networks (in particular by using machine learning techniques for improving routing and anomaly detection). He also uses monitoring and honeypots for studying malicious traffic in the Internet and assessing the threads on the network and its users.