VLATKO LIPOVAC, Ph.D.
E-mail: vlatko.lipovac@unidu.hr
VEDRAN BATOŠ, Ph.D.
E-mail: vedran.batos@unidu.hr
University of Dubrovnik
Dpt. of Electrical Engineering and Computing
Branitelja Dubrovnika 29, HR-20000 Dubrovnik,
Republic of Croatia
BORIS NEMŠIĆ, Ph.D.
E-mail: b.nemsic@mobilkom.at
Telekom Austria
Lassalerstrasse 9, 1020 Vienna, Republic of Austria

TESTING TCP TRAFFIC CONGESTION BY DISTRIBUTED PROTOCOL ANALYSIS AND STATISTICAL MODELLING

ABSTRACT

In this paper, a solution is proposed for testing TCP congestion window process in a real-life network situation during stationary time intervals. With this respect, the architecture of hardware and expert-system-based distributed protocol analysis is presented that we used for data acquisition and testing, conducted on a major network with live traffic (Electronic Financial Transactions data transfer), as well as the appropriate algorithm for estimating the actual congestion window size from the measured data that mainly included decoding with precise time-stamps (100ns resolution locally and 1μs with GPS clock distribution) and expert-system comments, resulting from the appropriate processing of the network data, accordingly filtered prior to arriving to the special-hardware-based capture buffer. In addition, the paper presents the statistical analysis model that we developed for the evaluation whether the data belonged to the specific (in this case, normal) cumulative distribution function, or whether two data sets exhibit the same statistical distribution - the conditio sine qua non for a TCP-stable interval. Having identified such stationary intervals, it was found that the measured-data-based congestion window values exhibited very good fitting (with satisfactory statistical significance) to the truncated normal distribution. Finally, an appropriate model was developed and applied, for estimating the relevant parameters of the congestion window distribution: its mean value and the variance.

KEY WORDS

protocol analysis, TCP-IP, testing, traffic congestion, statistical analysis, parameter estimation

1. INTRODUCTION

Transport Control Protocol (TCP) is a connection-oriented and so reliable protocol that much of the Internet traffic uses it at the transport layer (the rest belongs to the connectionless User Datagram Protocol (UDP)). As a (sliding) window-based protocol, it controls the sending rate at end-points, together with queuing mechanisms provided by the routers [1].

It is imperative to predict the performance of the connections; however, it turned out that this might not be a simple task [2]. With this regard, the means of testing the congestion window process is proposed, through experimental approach as we measured the real Internet traffic, processed the data, collected and developed statistical methods to analyze it.

Section 2 presents the architecture of our distributed network test system and the organization of measurements, focusing data acquisition hardware and data processing on top of which the model for the actual congestion window calculus was built. Section 3 introduces statistical methods to analyze stationary distribution of the congestion window process that exhibits the properties of the truncated normal cumulated distribution function (cdf), whose mean and variance are estimated from the acquired protocol data. Section 4 presents the main results of our experimental study and some resulting observations. Section 5 presents the conclusions.

Flow control and managing congestion in TCP/IP networks

As seen in Figure 1, each TCP traffic *Protocol Data Unit* (PDU) – segment is divided into the header and the data.

The TCP window is sometimes referred to as the "TCP sliding window" [3]. This field tells the receiver how many bytes the transmitting host can receive without acknowledgement. Thus, the sliding window

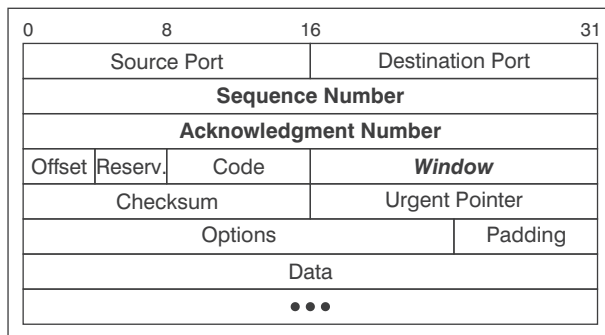| 0	8	16	31
Source Port		Destination Port	
Sequence Number			
Acknowledgment Number			
Offset	Reserv.	Code	*Window*
Checksum		Urgent Pointer	
Options			Padding
Data			
•••			

Figure 1 - TCP segment format

size in TCP can be adjusted on the fly, which is a primary flow control mechanism, allowing a larger amount of data "in flight", so that the sender gets ahead of the receiver (though not *too far* ahead). Actually, the so advertised window informs the sender of the receiver's buffer space.

In original TCP design this was the sole protocol mechanism controlling the sender's rate. However, this simple flow control mechanism keeps *one fast sender* from overwhelming *a slow receiver,* while congestion control keeps a *set of senders* from overloading the *network,* and so must ensure adjusting to bottleneck bandwidth and to its variations, as well as sharing bandwidth between flows.

Therefore, in real life, congestion is unavoidable; when two packets arrive at the same time the node can only transmit one and either buffer or drop the other. Specifically, if many packets arrive within a short period of time, the buffer may eventually overflow, so that an increase in the network load results in a decrease of useful work done due to: undelivered packets, packets consuming resources that are dropped later in the network, spurious retransmissions of packets still "in flight" (leading to even more load)… In the mid-1980s, the Internet converged to a halt until Jacobson/Karels devised TCP congestion control [3], so that the general approaches to avoiding drops of many packets and the so-called congestion collapse include: pre-arranging bandwidth allocations (with drawbacks of requiring negotiation before sending packets and potentially low utilization, differential pricing, i.e. not dropping packets for the highest bidders, and dynamic adjustment with testing the level of congestion with speeding up when there is no congestion and slowing down otherwise (where drawbacks are: suboptimality, complex implementation) [4].

With this regard, the term "congestion window" denotes the maximum number of unacknowledged bytes that each source can have "in flight". This implies that, in order to conduct congestion control, each source must determine the available network capacity so as to know how many packets it can leave "in flight". The congestion-control equivalent of the receiver window principle should presume sending at

the rate of the slowest component in order to adapt the window by choosing its (maximal) size as the minimal out of the two values: the actual congestion window and the receiver window. Thus, upon detecting congestion, the congestion window must be (fast) decreased, as well as increased had there been no congestion detected.

Detecting congestion by a TCP sender can be accomplished in a number of ways. For example, an indication can be if the Internet Control Message Protocol (ICMP) *Source Quench* messages are detected on the network. However, this is not particularly reliable, because during times of overload, the signal itself could be dropped. Increased packet delays or losses can be another indicator, but also not so straightforward due to considerable non-congestive delay variations and losses (checksum errors) that can be expected in the network.

Anyway, no matter how congestion is detected, managing it must start from the fact that the consequences of over-sized window (packets dropped and retransmitted) are much worse than having an under-sized window (somewhat lower throughput). Therefore, upon success with the last window of data, the TCP sender load should increase linearly, and decrease multiplicatively, upon packet loss [4]. This becomes a necessary condition for the stability of TCP [5], [6], [7].

Particular schemes for managing congestion window are out of scope of this paper and will therefore not be further explored here, as our experimental investigations and analyses focused just on the estimation of statistical distribution of the stationary congestion window size.

2. DISTRIBUTED PROTOCOL ANALYSIS SYSTEM FOR TESTING TCP CONGESTION

The available test methods for studying communications networks range from mathematical modelling, through simulation (and/or emulation) to real-life measurements. We based this research on measuring the relevant parameters of test traffic by specialized hardware and analyzing the measurement results statistically.

2.1 Architecture of the test system

LAN (Ethernet 100 Mbit/s to 1 Gbit/s) and WAN (Frame Relay) network infrastructure of a major Austrian bank in/between their offices in three cities was used. The network consisted of dedicated workstations, residing at different locations around the network and exchanging test traffic, as well as of
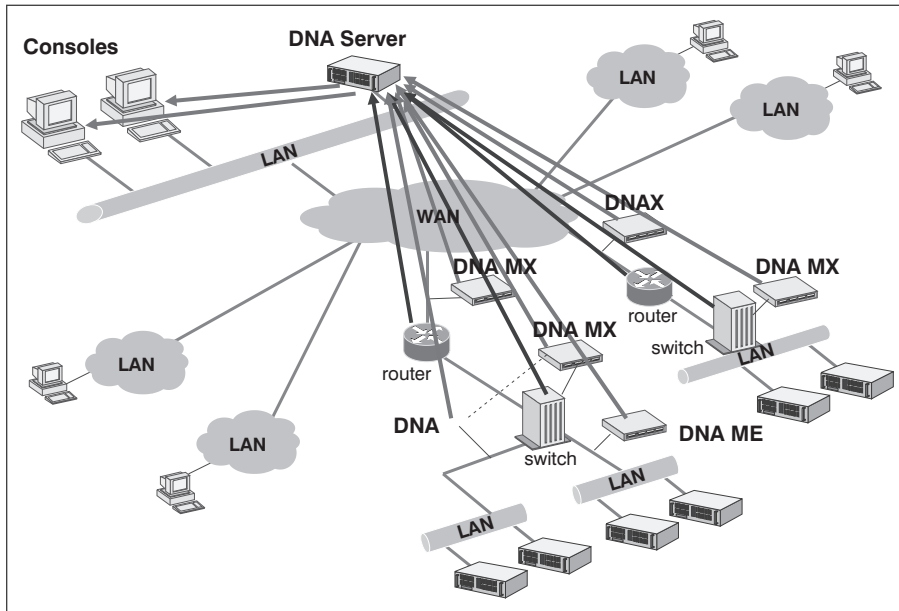
Figure 2 - Test scenario with distributed protocol analysis system spanning distant networks

six distributed test system devices: hardware-based protocol analyzers - namely, *Distributed Network Analyzers* (DNA), products of Agilent Technologies, aimed for protocol data acquisition and analysis. These were dispersed throughout various network segments of interest, either as network resident or temporarily installed for network performance testing. Their interwork was orchestrated by the central application server (Network Troubleshooting Center software of Agilent Technologies), which controlled and integrated the distributed data acquisition modules and their expert protocol analyses, and was accessible via two remote clients - consoles, *Figure 2*.

By examining the precisely time-stamped TCP traffic Protocol Data Units (PDU) – *segments*, using the tools of the non-intrusively attached DNAs, we were able to characterize the network, as well as its endpoints. In this paper, the focus is on the transaction-intensive network application traffic profiles, such as the ones associated with Electronic Financial Transactions (EFT), in particular, where proper network performance is mission critical [1].

Many measurements were carried out throughout the day work time and in various environments: LAN – LAN and LAN-WAN-LAN. For each packet sent or received, the DNAs registered its timestamp, sequence number and size. Multiple DNAs, with their 1Gb/s acquisition system and 256MB capture memory per line interface module (LIM), supporting real-time network data capture and filtering, were combined for time-synchronized multi-port tests, still using the same software features as with stand-alone protocol analyzer, such as e.g. decoding, statistical analysis and expert analysis [8]. Time synchronization was achieved either via the "EtherSync" interfaces (where DNAs were daisy-chained), allowing DNAs to be synchronized to each other within ±100ns, or by means of the external GPS sources, providing synchronization accuracy of $\pm1\mu$s. (For the reference, the IETF's Network Time Protocol (NTP) could provide the synchro-

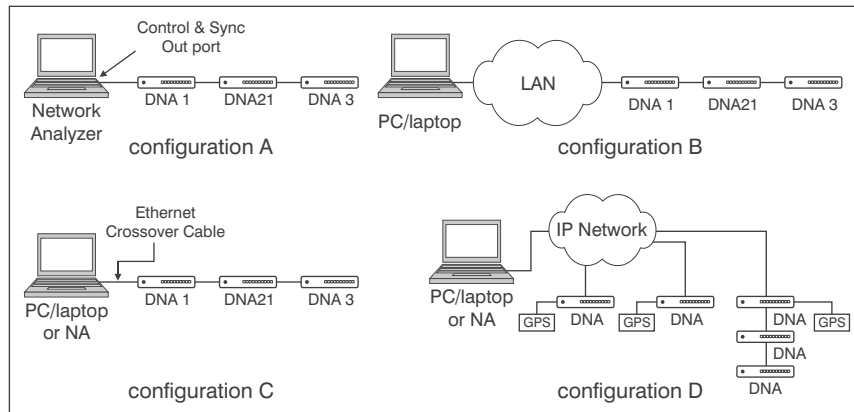

Figure 3 - Multiple DNAs in a network with various time synchronization

nization accuracy of ±20ms or better, depending on the proximity of the Network Time Servers (NTS)).

The scenarios deployed included: multiple daisy--chained DNAs connected to a PC or a protocol analyzer either directly, or via LAN, or multiple DNAs in a network with GPS, rather than NTP time synchronization.

2.2 Data acquisition and managing the capture buffer processing

State-of-the-art hardware-based protocol analyzers are considerably more powerful than their software-based counterparts, as they can analyze and record 100 percent of network traffic with great precision, regardless of the network throughput, from relatively low to high-speed networks such as nGigabit Ethernet etc. In a monitor-through connection, a protocol analyzer is connected between a node and a hub/switch or between two switches, so it non-intrusively sees all traffic occurring between the two devices.

The protocol analyzers used were equipped with both LAN and WAN interfaces, which provided physical connections and high-speed data acquisition hardware to get every frame on the network into any particular protocol analyzer, *Figure 4*.

The capture buffer is a special kind of memory that can be written to at very high speed. The RISC CPU, optimized for speed and accuracy, processes frames from the capture buffer and feeds the information to protocol analyzer measurements running in Windows on the PC. We selected the packet slicing option with capture buffer of each LIM, to enable the protocol analyzers to capture only the first part (e.g. first 100 Bytes) of each packet, containing the relevant header information, so that we could store more packets for a given buffer size (of up to 256 Mbytes). We mostly used the *Full Buffer* option, where data acquisition
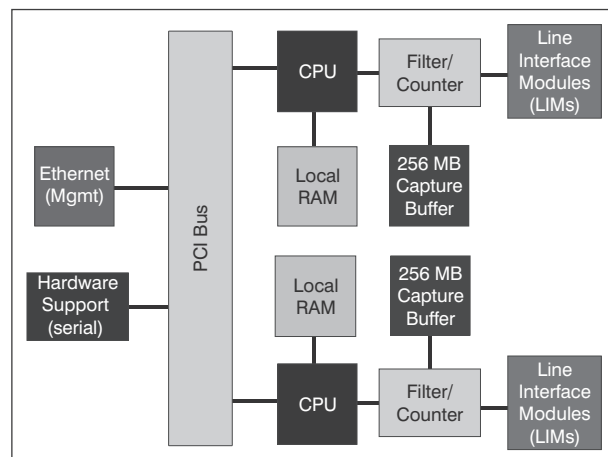

Figure 4 - Hardware-based protocol analyzer system architecture

continued until the buffer was filled, and the acquisition finally stopped.

We enabled the built-in capture filters to control which frames are allowed to enter the capture buffer, and thus focused the analyzer (or just saved space in the capture buffer). As these filters are implemented in hardware, they were also used to trigger an *action*, such as halt or start collecting data on a matched frame, as well as either include or exclude the matched frames from logging into the buffer, and later on to the hard disk of the analyzer PC platform. Among a number of different available filter criteria, we used: addresses, protocols (TCP, IP, etc.), specific fields (such as e.g. window size), frame attributes (such as erroneous or good frames etc.), and, in some instances, frame data bytes (the first 127 bytes can be used).

Associated to capture filters are statistics counters that provide counts of frames, packets and other events matching the selected filter criteria. They were set up and used for getting the precise statistics – histograms of the traffic events that we investigated.

2.3 Measurements and data processing

The very essential measurement of any protocol analyzer is decoding (interpreting various PDU fields of interest). The *Decoding* window for the analyzers used here is presented in *Figure 5*, which shows the very essential information used for characterizing the congestion window – precise timestamps of packet arrivals.

However, state-of-the-art protocol analyzers provide much more information than just decoding. This always includes statistical analysis of traffic, and finally, expert analysis, where the system compares the network problems that occur to information in its knowledge database, and if any error scenario is found in the database that matches the discovered situation, the system suggests possible diagnoses and troubleshooting tips. So, as we monitored the traffic, the expert system transformed the data into meaningful diagnostic information, thus reducing thousands of frames to a handful of significant events, including e.g. router misconfigurations, slow file transfers, *inefficient window sizes*, and connection resets. As can be seen from the *Expert Commentator* display for the case of *excessive retransmissions* of IP packets, *Figure 6*, the related *Warning Event* shows node and connection information in one view, again properly time-stamped (and thus useful in the congestion window analysis).

Following data processing in each DNA, the appropriate post-processing software (Agilent's *Report Center*) was also used to accomplish the multi-segment network baselining and benchmarking, with time correlation of data across the segments of interest. Using these multitasking measurement features, the raw
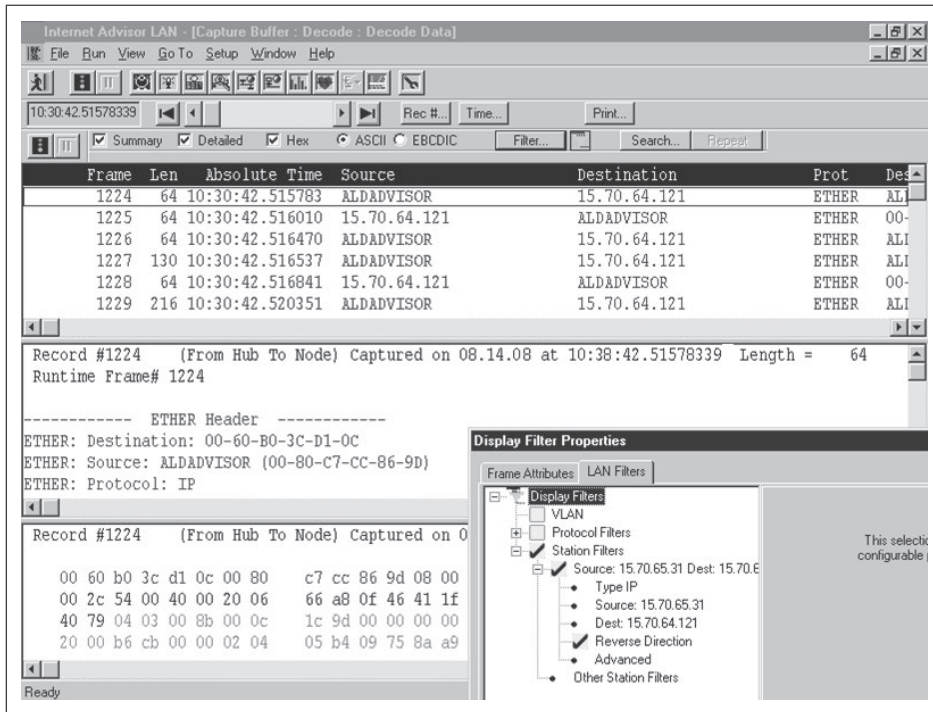
Figure 5 - Decoding of protocol data units, with precise time stamping (resolution of 100ns)
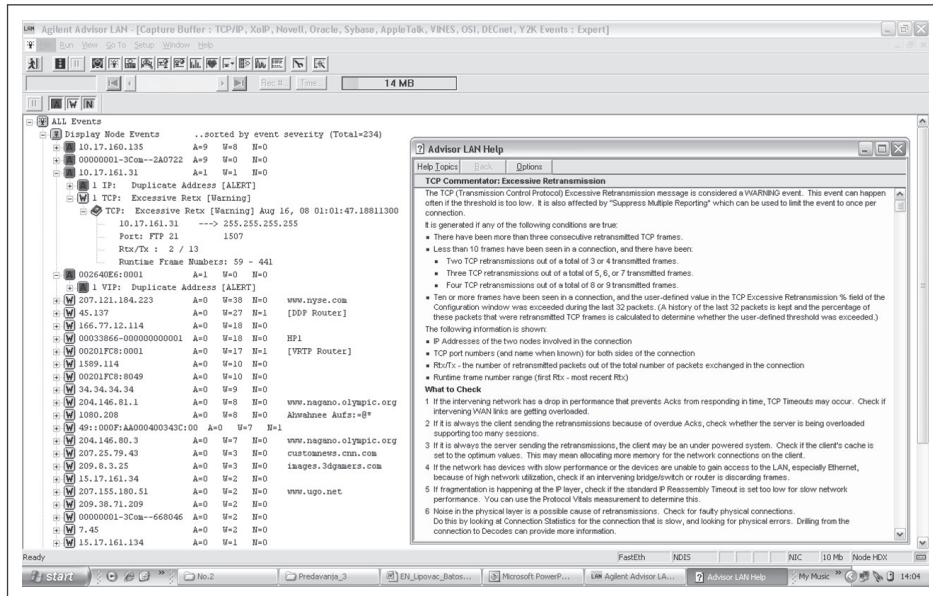

Figure 6 - Expert Commentator measurements window

data could be analysed in different ways concurrently, such as e.g. to get correlated statistics between protocols, nodes (that use these protocols) and connections of each such end-station [8].

In order to estimate the sender's congestion window size from the collected data, we had to identify (by filtering with appropriate criteria) the packets that have been sent from the sender, but have not yet arrived to the receiver, count them (by stats counters) and present as a function of time. The above presented features of the experimental system enabled

fulfilling this task with great precision and accuracy. We made a simple application program that added +1 to the actual congestion window size for each outgoing packet at the time it was leaving the sender, while adding -1 when/if that packet arrived at the receiver's. With the exception of lost packets (that can be traced by various means, the easiest one by the Expert Commentator (Figure 6), thus excluding the lost packets from the calculation), this accumulated sum well approximated the actual congestion window size almost in real time.

3. STATISTICAL ANALYSIS MODEL

In statistics, the Kolmogorov–Smirnov (K-S) test is used to determine whether a certain empirical cdf, based on finite samples, differs from a hypothesized continuous distribution function specified by the null hypothesis [9]. An attractive feature of this test is that the cdf of the K-S test statistics itself does not depend on the distribution function being tested, and also that it does not depend on adequate sample size (as e.g. the chi-square goodness-of-fit does). The higher the extent to which the test in question shows that the speculated hypothesis has (or has not) been nullified - the significance level α (of the difference between the hypothesized values and the sample-based ones), obviously, the less likely it is that the phenomena in question could have been produced by chance alone. Hence, the significance level is the probability that the null hypothesis will be rejected by mistake, when it is true (a decision known as *Type I* error, or "false positive"). Among the popular levels of significance: 5%, 1% and 0.1%, the mid value for our tests was adopted.

The significance *of a result* is also called its *p-value*; the smaller the p-value, the more significant the result is said to be. If a test of significance gives a p-value lower than the α-level, the null hypothesis is rejected. Such results are informally referred to as "statistically significant". The lower the significance level, the stronger the evidence.

3.1 One-sample Kolmogorov-Smirnov test of conformance to normal distribution

A sample Kolmogorov-Smirnov test [9] enables testing of a hypothesis that a certain distribution $F_{n\xi}(x)$ of a random variable ξ conforms to the given continuous cdf $F_{0\xi}(x)$.

$$H_0: F_\xi(x) = P(\xi < x) = F_{0\xi}(x) \tag{1}$$

The empirical cdf $F_{n\xi}(x)$ is derived from the independent samples $(\xi_1, \xi_2, ..., \xi_n)$. The Kolmogorov-Smirnov statistics for a given $F_{0\xi}(x)$ is:

$$D_n = \sup_x \left| \hat{F}_{n\xi}(x) - F_{0\xi}(x) \right| \tag{2}$$

As it follows from the theorem of Glivenko-Cantelli [9], if the observed sample comes from the $F_{0\xi}(x)$ distribution, then D_n converges to 0. Furthermore, as $F_{0\xi}(x)$ is continuous, the rate of convergence of $\sqrt{n}D_n$ is determined by the Kolmogorov limit distribution theorem [9] stating:

$$\lim_{n \to \infty} \Pr(\sqrt{n}D_n < y) = K_\eta(y), \quad 0 < y < \infty \tag{3}$$

where $K_\eta(y)$ is the Kolmogorov cdf (that does not depend on $F_{0\xi}(x)$, as pointed out above).

Moreover, if the significance level of α is pre-assigned, then the tested null-hypothesis is to be rejected at the level α if:

$$\sqrt{n}D_n > y_\alpha \tag{4}$$

where the cut-off y_α is found by equalizing the Kolmogorov cdf $K_\eta(y)$ and 1-α:

$$\Pr(\sqrt{n}D_n \le y_\alpha) = K_\eta(y_\alpha) = 1 - \alpha \Rightarrow y_\alpha = K_\eta^{-1}(1-\alpha) \tag{5}$$

Otherwise null-hypothesis should be accepted at the significance level of α.

Actually, the significance is mostly *tested* by calculating the (*two-tail* [9]) p-value (which represents the probability of obtaining values of the test statistics that are equal to or greater in magnitude than the observed test statistics), by using the theoretical $K_\eta(y)$ cdf of the test statistics to find the area under the curve (for continuous variables) in the direction of the alternative (with respect to H_0) hypothesis, i.e. by means of a look-up table or integral calculus, while in the case of discrete variables, simply by adding up the probabilities of events occurring in the direction of the alternative hypothesis that occurs at and beyond the observed test statistics value. So, if it comes out that:

$$p = 1 - K_\eta(\sqrt{n}D_n) < \alpha \tag{6}$$

then the null hypothesis is again to be rejected, at the assumed significance level α, otherwise (if the p-value is greater than the threshold α), we can state that we do not reject the null hypothesis and that the difference is not statistically significant.

3.2 Two-sample Kolmogorov-Smirnov test for identifying stationary intervals

While the main applications of the one-sample K-S test are testing goodness of fit with the normal and uniform distributions, the two-sample K-S test is one of the most useful and general nonparametric methods for comparing two samples, as it is sensitive to differences in both location and shape of the empirical cdfs of two samples, so it is the most important theoretical tool for detecting points of change.

Let us now consider the test for the series $\xi_1, \xi_2, ..., \xi_n$ of the first sample, and $\eta_1, \eta_2, ..., \eta_n$ of the second, where the two series are independent. Furthermore, let $\hat{F}_{m\xi}(x)$ and $\hat{G}_{n\eta}(y)$ be the corresponding empirical cdfs. Then the K-S statistics is:

$$D_{m,n} = \sup_x \left| \hat{F}_{m\xi}(x) - \hat{G}_{n\eta}(y) \right| \tag{7}$$

The limit distribution theorem states that:

$$\lim_{m,n \to \infty} P\left(\sqrt{\frac{mn}{m+n}} D_{m,n} < z \right) = K_\zeta(z), \quad 0 < z < \infty \tag{8}$$

where again $K_\zeta(z)$ is the Kolmogorov cdf.

3.3 Estimation of the (normal) distribution parameters

Let us consider a normally distributed random variable $\xi \in N(m, \sigma^2)$, where:

$$p_\xi(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-m)^2}{2\sigma^2}} \tag{9}$$

Its cdf $\Phi_\xi(x)$ can be expressed as the *standard* normal cdf $\Phi(x)$ [9] of the ξ-related zero-mean normal random variable, normalized to its standard deviation σ:

$$\Phi_\xi(x) = \Pr(\xi \leq x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(u-m)^2}{2\sigma^2}} du =$$

$$= \Phi\left(\frac{x-m}{\sigma}\right) = \int_{-\infty}^{\frac{x-m}{\sigma}} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{v^2}{2}} dv \tag{10}$$

Normal cdf has no lower limit; however, since the congestion window can never be negative, here we must consider a truncated normal cdf. In practice, when the congestion window process gets in its stationary state, the lower limit is hardly 0. Therefore, for the reasons of generality, here we consider a truncated normal cdf with lower limit l, where $l \geq 0$.

Now parameters m, σ and l are estimated, starting from:

$$\Pr(\xi > l) = 1 - \Phi\left(\frac{l-m}{\sigma}\right) =$$

$$= 1 - \int_{-\infty}^{\frac{l-m}{\sigma}} \frac{1}{\sqrt{2\pi}} e^{-\frac{v^2}{2}} dv = Q\left(\frac{l-m}{\sigma}\right) \tag{11}$$

where: $Q\left(\frac{l-m}{q}\right)$ is the Gaussian *tail* function [9].

The conditional expected value of ξ, just on the segment $(l, +\infty)$ is:

$$E(\xi / \xi > l) = \int_{l}^{+\infty} u \cdot \frac{1}{\sqrt{2\pi}\sigma \cdot Q\left(\frac{l-m}{\sigma}\right)} e^{-\frac{(u-m)^2}{2\sigma^2}} du \tag{12}$$

By substituting: $\frac{u-m}{\sigma} = v$, $du = \sigma \cdot dv$ into (12), we obtain:

$$E(\xi / \xi > l) = \frac{1}{Q\left(\frac{l-m}{\sigma}\right)} \int_{\frac{l-m}{\sigma}}^{+\infty} (\sigma \cdot v + m) \frac{1}{\sqrt{2\pi}} e^{-\frac{v^2}{2}} dv =$$

$$= \frac{\sigma}{Q\left(\frac{l-m}{\sigma}\right)} \int_{\frac{l-m}{\sigma}}^{+\infty} v \frac{1}{\sqrt{2\pi}} e^{-\frac{v^2}{2}} dv +$$

$$+ \frac{m}{Q\left(\frac{l-m}{\sigma}\right)} \int_{\frac{l-m}{\sigma}}^{+\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{v^2}{2}} dv =$$

$$= \frac{\sigma}{Q\left(\frac{l-m}{\sigma}\right)} \int_{\frac{l-m}{\sigma}}^{+\infty} v \frac{1}{\sqrt{2\pi}} e^{-\frac{v^2}{2}} dv + m =$$

$$= \frac{\sigma}{Q\left(\frac{l-m}{\sigma}\right)} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{l-m}{\sigma}\right)^2} + m \tag{13.}$$

Now, if we pre-assign a certain value γ to the above used tail function $Q(\cdot)$, then the corresponding argument (and so m) is determined by the inverse function $Q^{-1}(\gamma)$:

$$Q\left(\frac{l-m}{\sigma}\right) = \gamma \Rightarrow m = l - \sigma \cdot Q^{-1}(\gamma) \tag{14}$$

so that (13) can now be rewritten as:

$$m = E(\xi / \xi > l) - \frac{\sigma}{\gamma} \cdot \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}[Q^{-1}(\gamma)]^2} \tag{15}$$

Substituting m from (14) into (15) results with the following formula for σ:

$$\sigma = \frac{l - E(\xi / \xi > l)}{Q^{-1}(\gamma) - \frac{1}{\sqrt{2\pi}\gamma} e^{-\frac{1}{2}[Q^{-1}(\gamma)]^2}} \tag{16}$$

Finally, substituting the above expression for σ into (14), we obtain the expression for m:

$$m = \frac{\gamma \cdot Q^{-1}(\gamma) E(\xi / \xi > l) - l \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}[Q^{-1}(\gamma)]^2}}{\gamma \cdot Q^{-1}(\gamma) - \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}[Q^{-1}(\gamma)]^2}} \tag{17}$$

Thus it came out that, after developing formulas (16) and (17), we expressed the mean m and the variance σ^2 of the Gaussian random variable ξ, by the mean $E(\xi / \xi > l)$ of the truncated cdf, the truncation cut-off and the tabled inverse $Q^{-1}(\gamma)$ of the Gaussian tail function, for the assumed value γ. As these relations hold among the corresponding estimates, too, in order to estimate $\hat{m}$ and $\hat{\sigma}$, first we need to estimate $\hat{E}(\xi / \xi > l)$ and $\hat{\gamma}$ from the sample data:

$$\hat{E}(\xi / \xi > l) = \frac{1}{\left|\{1 \leq i \leq n: \xi_i > l\}\right|} \sum_{i=1}^{n} \xi_i N_i(\xi_i > l) \tag{18}$$

$$\hat{\gamma} = \frac{1}{n} \sum_{i=1}^{n} N_i(\xi_i \leq l) \tag{19}$$

where N_i denotes the number of occurrences (frequency) of particular samples being larger/smaller-or-equal to l, respectively.

Thus, once we have estimated $\hat{E}(\xi/\xi>l)$ and $\hat{\gamma}$ by (18) and (19), we can then calculate the estimates $\hat{\sigma}$ and $\hat{m}$ by means of (16) and (17), which completes the estimate of the pdf (9).

4. RESULTS OF THE ANALYSIS

Initially, we characterized the network traffic with respect to packet delay variation and packet loss – that were, expectedly, considered as significant influencers on the congestion window. Accordingly, in many tests, for mutually very different network conditions and between various end-points, we noticed significant packet delay variation, *Figure 7*.

However, the expected impact of the packet delay variation [2], [10] on packet loss (and thus on congestion, i.e. to its window size), has not been noticed as significant, *Figure 8a, 8b*.
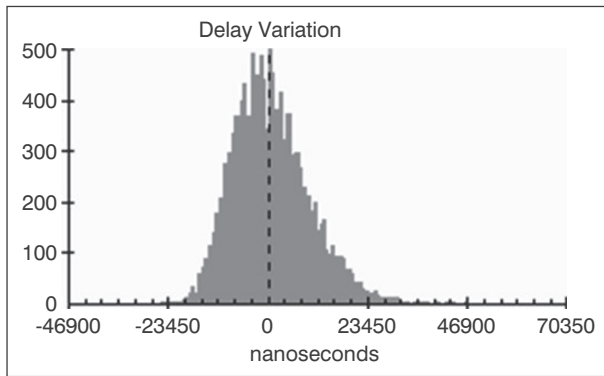

Figure 7 - Typical packet delay variation within a test LAN segment

Still, we noticed some sporadic bursts of packet losses, which we explain as consequence of grouping of the packets coming from various connections. Once the buffer of the router, using drop-tail queuing algorithm, gets in overflow state due to heavy incoming traffic, the most of or the whole burst might be dropped. This introduces correlation between consecutive packet losses, so that they, too (as packets themselves), occur in bursts. Consequently, the packet loss rate alone does not sufficiently characterize the error performance. (Essentially, "packet-burst-error-rate" would be needed, too, especially for applications sensitive to long bursts of losses [2], [5] [6], [10]).

With this respect, one of our observations (resulting from the expert analysis tools referenced in Section 2) was that, in some instances, congestion window values show strong correlation among all six connections. Very likely, this was the consequence of the bursty nature of packet losses, as each packet, dropped from a particular connection, is likely to cause the congestion window of that very connection to be simultaneously reduced [2], [3], [2].

In our real-life analyses of the congestion process stationarity, we considered the congestion window values, calculated from the TCP PDU stream, captured by protocol analyzers, as a sequence of quasi-stationary series with constant cdf that changes only at frontiers between the successive intervals [9]. In order to identify these intervals by successive two-sample K-S tests (as explained in Section 3), we compared the empirical cdfs within two neighbouring time windows of rising lengths, sliding them along the data samples, to finally combine the two data samples
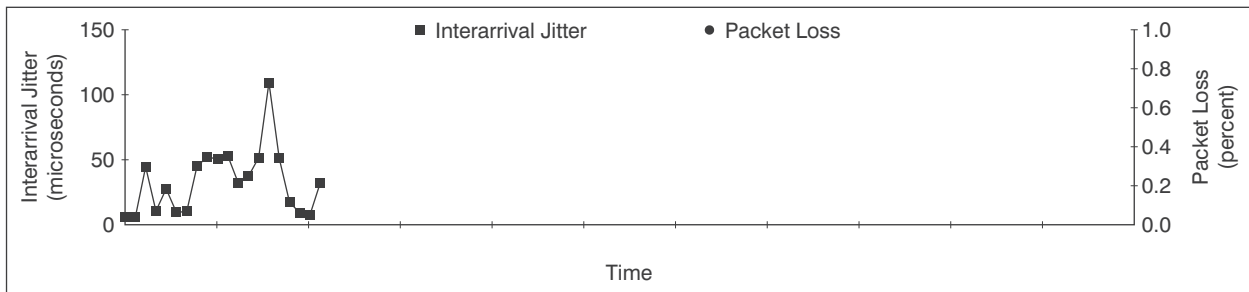

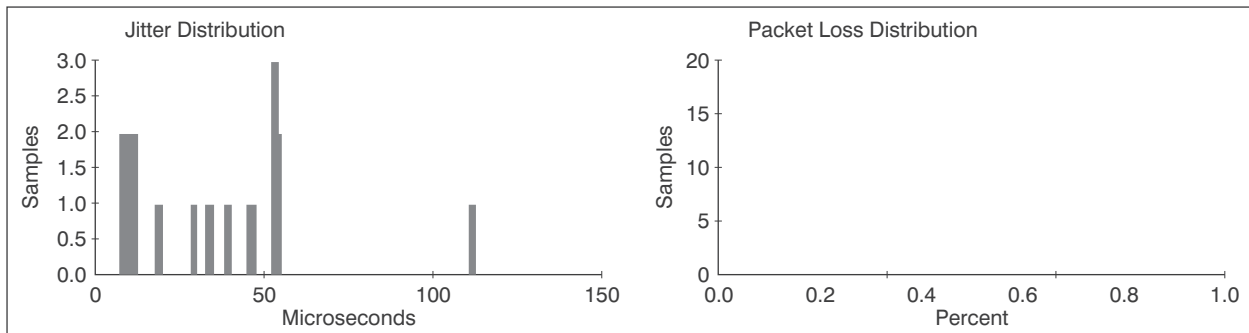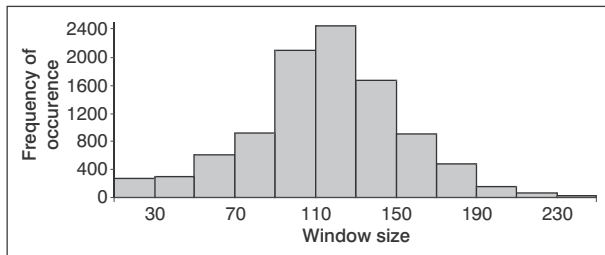Figure 8a - Typical time-diagram of correlated packet jitter and loss measurements

Figure 8b - Typical histogram of correlated packet jitter and loss measurements

Table 1 - Typical values of stationary congestion window size

$Pr(x_i - 20 < x < x_i)$	278	310	624	928	2094	2452	1684	911	478	157	63	21
x_i	30	50	70	90	110	130	150	170	190	210	230	250

Figure 9 - Typical histogram of the congestion window

into a single test series, once the distributions matched.

Typical results (where "typical" refers to traffic levels, network utilization and throughput for a particular network configuration) of our statistical analysis for 10,000 samples of actual stationary congestion window sizes, sorted in classes with the resolution of 20, are presented in *Table 1* and as histogram, in *Figure 9*, visually indicating compliance with the (truncated) normal cdf, with the sample mean within the class of 110 to 130. Accordingly, as the TCP-stable intervals were identified, we conducted numerous one-sample K-S tests and obtained p-values in the range from 0.414 to 0.489, which provided solid indication for accepting (with $\alpha = 1\%$) the null-hypothesis that, during stationary intervals, the statistical distribution of congestion window was (truncated) normal.

As per our model from Section 3, the next step was to estimate typical values of the congestion window distribution parameters. Thus, firstly, by means of (19), we estimated $\hat{\gamma}$ as one minus the sum of frequencies of all samples belonging to the lowest value class (so e.g., in the typical case, presented by *Table 1* and *Figure 9,* we took: $\hat{\gamma} = 1 - 278/10000 = 0.9722$, which determined the value $Q^{-1}(\gamma) = -1.915$ that we accordingly selected from the look-up table). Then we chose the value of $l = 30$ for the truncation cut-off and, from (18), calculated the mean $\hat{E}(\xi / \xi > l) = 117.83$ of the truncated distribution, excluding the samples from the lowest class and their belonging frequencies, from this calculation.

Finally, based on (16) and (17), the estimates for the distribution mean and variance of the exemplar typical data presented above, were obtained as: $\hat{m} = 114.92$ and $\hat{\sigma} = 44.35$.

5. CONCLUSION

In this paper, the means for real-life testing of TCP congestion window process are proposed. Tests were conducted on a major network with live traffic, by means of hardware and expert-system-based distributed protocol analysis and applying the appropriate model that we developed, for statistical analysis of the captured data. It was shown that the distribution of TCP congestion window size, during stationary intervals that we identified prior to estimation of the cdf can be considered as close to the normal, whose parameters we estimated experimentally, following our theoretical model. In some instances, we found that the congestion window values show strong correlation among various connections, as consequence of the intermittent bursty nature of the packet losses. The proposed test model can be extended to include the analysis of TCP performance in various communications networks.

Dr. sc. **VLATKO LIPOVAC**
E-mail: vlatko.lipovac@unidu.hr
Dr. sc. **VEDRAN BATOŠ**
E-mail: vedran.batos@unidu.hr
Sveučilište u Dubrovniku
Odjel za elektrotehniku i računarstvo
Branitelja Dubrovnika 29, 20000 Dubrovnik,
Republika Hrvatska
BORIS NEMŠIĆ
E-mail: b.nemsic@mobilkom.at
Telekom Austria
Lassalerstrasse 9, 1020 Vienna, Republic Österreich

SAŽETAK

ISPITIVANJE ZAGUŠENJA TCP PROMETA POMOĆU DISTRIBUIRANE ANALIZE PROTOKOLA I STATISTIČKOG MODELIRANJA

U ovome radu, predstavljamo koncept i realizaciju praktičnog ispitivanja procesa prozora TCP zagušenja tijekom stacionarnih vremenskih intervala. S tim u vezi, predstavljamo arhitekturu distribuirane analize protokola, temeljene na strojnoj opremi i ekspertnom sustavu, koju smo koristili za praktičnu akviziciju podataka i ispitivanje koja smo proveli na jednoj većoj mreži sa živim prometom (prijenos podataka elektroničkih financijskih transakcija), kao i pogodni algoritam za procjenu aktualne veličine prozora zagušenja iz mjernih podataka – uglavnom dekodiranih jedinica protokola s preciznim vremenskim žigovima (rezolucije 100 ns lokalno i 1 ms s GPS distribucijom takta) i rezultata ekspertne analize odgovarajuće filtriranih podataka iz specijalne hardverski bazirane odvajačke memorije za akviziciju podataka. Osim toga, predstavili smo model statističke analize koji smo razvili za procjenu pripadaju li podatci nekoj posebnoj (normalnoj) kumulativnoj funkciji raspodjele, ili dva skupa podataka iskazuju istu statističku raspodjelu – conditio sine qua non za stabilni interval. Konač-

no, kako su se vrijednosti prozora zagušenja, dobivene na temelju mjernih podataka, sa zadovoljavajućom statističkom značajnošću, pokazale sukladnim ograničenoj normalnoj raspodjeli, razvili smo i primijenili odgovarajući model za procjenu relevantnih parametara raspodjele prozora zagušenja: njegove srednje vrijednosti i varijance.

KLJUČNE RIJEČI

analiza protokola, TCP-IP, ispitivanje, zagušenje prometa, statistička analiza, procjena parametara

LITERATURE

[1] **Lipovac, V.**: *Testing QoS of Multiservice Networks*, CRC Press New York (manuscript, expected to be in print by fall 2009)

[2] **Kumar, A.**: *Comparative Performance Analysis of Versions of TCP*, *IEEE/ACM Transactions on Networking*, Aug. 1998

[3] **Comer, D. E.**: *Internetworking with TCP/IP*, Vol. 1; *Principles, Protocols, and Architecture* (Fifth Edition), Prentice Hall, NJ, 2005

[4] **Mathis, M.**, **Semke, J.**, **Mahdavi, J.** and **Ott, T. J.**: *The Macroscopic Behavior of the TCP Congestion Avoidance Algorithm,* Computer Communication Review, Vol. 27, No. 3, July 1997

[5] **Chen, K.**, **Xue, Y.** and **Nahrstedt, K.**: *On setting TCP's Congestion Window Limit in Mobile ad hoc Networks*, Proc. IEEE International Conf. on Communications, Anchorage, May 2003

[6] **Floyd, S.** and **Fall, K.**: *Promoting the Use of End-to-End Congestion Control in the Internet*, IEEE/ACM Trans. on Networking, Vol. 7, Issue 4, pp. 458 – 472, Aug. 1999

[7] **Balakrishnan, H.**, **Rahul, H.** and **Seshan, S.**: *An Integrated Congestion Management Architecture for Internet Hosts*, Proc. ACM SIGCOMM, Sep. 1999

[8] Network Analyzer Technical Overview 5988-4231EN, Agilent Technologies, 2004

[9] **Kendall, M.**, **Stewart, A.**: *The Advanced Theory of Statistics,* Charles Griffin, London, 1966

[10] **Elteto, T.**, **Molnar, S.**: *On the distribution of round-trip delays in TCP /IP networks*, International Conference on Local Computer Network, 1999

ACKNOWLEDGEMENTS