Analysis and Detection of Bottlenecks via TCP Footprints in live 3G Networks

Philipp Svoboda*, Fabio Ricciato†
*INTHFT Department, Vienna University of Technology, Austria
Institut für Nachrichtentechnik und Hochfrequenztechnik
Technische Universität Wien, Austria
Gusshausstrasse 25/389, A-1040 Vienn a, Austria
Email: psvoboda@nt.tuwien.ac.at
†ftw. Forschungszentrum Telekommunikation
Donau-City-Strasse 1, A-1220 Vienna, Austria
Email: ricciato@ftw.at

***Abstract*— In this paper we evaluate four different metrics for non intrusive bottleneck detection based on TCP counters. This work is based on the full TCP statistics recorded on five days spread over the last one and a half year within the core network of a mobile network operator in Austria. Scatterplots, so called "footprints", were generated counting the number of packets and the number of retransmission for each user during the peak hours. Two of the datasets had a known capacity bottleneck in place. Based on those datasets we benchmarked the different metrics for the detection of a bottleneck event. We preprocessed the traces in order to remove the traffic increase. After this step all metrics were able to detect the special bottleneck case. Even traces separated for more than one year deliver a clear result. The performance of a PSNR metric was similar to the other metrics based on more sophisticated functions.**

I. Introduction

In this paper we evaluate metrics for non intrusive bottleneck detection in a mobile core-network using TCP (Transmission Control Protocol) related counters. The protocol guarantees reliable exchange of data between sender and receiver. This is achieved by the retransmission of data packets. The number of retransmission in the network is an indicator for losses. In wired networks the probability to lose a packet due to a link error is very small, therefore a high number of retransmissions indicate congestion in the network. Due to errors on the radio link the BER (Bit Error Rate) in a mobile UMTS (Universal Mobile Telecommunications System) network is larger than in wired networks. Coupled with a higher delay many retransmissions are caused by the physical errors rather than by network congestion [1, 2]. Retransmissions are a part of normal operation in a mobile environment. Therefore the detection of a bottleneck purely based on packet loss can be ambiguous.

There are two main tracks to analyze available bandwidth, the passive and the active. Active tools generate traffic patterns to evaluate the available bandwidth via certain routes [3]. These tools are able to extract exact figures under all network conditions — passive tools can only work if there is user traffic — however the implementation puts extra load to all network components. Common passive tools record the actual load and compare it with the available bandwidth on the local link. Ref. [1] extended the idea by applying statistical methods on the bandwidth values. The key idea is to extract the second and third order moments of different bandwidth values. A bottleneck is detected via a decreasing variance for higher numbers. The same authors show in [2] that this effect is due to the TCP mechanism. To analyze the effect more deeply, we used an improved version of TCP-Trace presented in Ref. [4] to extract retransmission events on the TCP layer.

A straight forward extension to the use of retransmitted packets, n_i, is a metric that is based on the normalization n_i/N_i, where N_i is the total number of packets for this user. Mobile data traffic is always related to a specific user, therefore it is possible to extract these numbers on a per user base. However the burstiness of the link error can rise false alarms, e.g., small number of N_i and bad radio conditions. To avoid this problem we decided to work with pairs of $[N_i, n_i]$ in scatterplots and applied a distance metric to compare the different scenarios. The goal was to find a metric that can be used as an indicator for hidden, e.g. not at the observed link, bottlenecks based on retransmissions. The anomaly is only visible as a part of the total traffic monitored.

The paper is structured in the following way. Section II presents the measurement setup, which is based on the METAWIN testbed developed at the ftw. Furthermore we use this section to describe the used traces. Section III we analyze the samples. In a first step we analyzed only visually and then in a second step via a metric based approach. Section IV uses a generated sample to benchmark the different metrics. The last section presents a summary and the conclusions.

II. Measurement Setup

The reference network scenario is depicted in Fig. 1. As most access networks, the 3G mobile network has a hierarchical tree-like deployment. The mobile stations and base stations are geographically distributed. Going up in the hierarchy (first BSC (Base Station Controller) / RNC(Radio Network Controller), then SGSN (Serving GPRS Support Node), ultimately

WIOPT 2008, 1st – 3rd Apr 2008, Berlin, Germany.
 ISBN 978-963-9799-18-9
DOI 10.4108/ICST.WIOPT2008.3256

GGSN(Gateway GPRS Support Node)) the level of concentration increases, involving a progressively smaller number of equipments and physical sites. In a typical network there are relatively few SGSNs and even fewer GGSNs. Therefore it is possible to capture the whole data traffic from home subscribers on a small number of Gn/Gi links. For further details of the structure of a 3G mobile network refer to [5].

To meet privacy requirements traces are anonymized by hashing all fields related to user identity at the lower 3G layers (e.g. IMSI, MSISDN), and removing the user payload above the TCP/IP layer.

The input traces were captured on a live GPRS/UMTS network at the Gn interface by the METAWIN monitoring system [1]. It is a monitoring tool designed to record traffic in a mobile core network. Although the underlying protocols and interfaces are similar to normal core networks, the presence of user mobility introduces intermediate protocols between the transport network and the user data (see [5]). Therefore METAWIN has to accomplish two main tasks: decoding the additional protocols and tracking the individual user sessions. In addition to this the system anonymizes all the traces and strips off the payload in order to protect the privacy of the customers. This preprocessing allows to do research on live traces. The system is capable to monitor at three interfaces: IuPs, Gn and Gi. The Gi interface is a normal Ethernet interface between the mobile and the internet service provider. No further user specific information is transmitted. The IuPs interface is used between SGSN and RNC units. Although the user specific information is present, the extraction and reassembling is, due to the high number of links and protocol stacks, very complex at this interface. Due to these facts we decided to extract the Gn interface, because it provides both: few interfaces and the whole user specific information.

The extraction parser called MOTRA dumps the packet traces to a ring-buffer. The TCP statistics were extracted using a modified version of `tcptrace`[2].

This work is based on TCP statistics on the Gn interface refined per user. This is possible because in a mobile network, like GPRS or UMTS as well as EDGE (Enhanced Datarates for Global Evolution) and HSDPA (High Speed Data Packet Access), each data packet is dedicated to a specific user. However, due to ciphering and security only session keys are stored in the packet header of the transport protocol visible on the interfaces (e.g. GPRS Transport Protocol (GTP) on Gn). The connection setup contains the user identifier (IMSI) and the session key. Therefore, the monitor unit has to perform some kind of tracking.

As the traces were captured in a live network they include several error sources like misbehaving terminals, portscans and so on (see [6]). These errors impact the tracking, and reduce the number of tracked packets to about 98%. The remaining 2% of packets could not be addressed to any user and were excluded from the further processing. In addition to this we filtered out the TCP ports 135 and 445 because most of spurious packets in the network address these ports, e.g., from port scans and/or attacks see [7], and could inflate N_i.

[1]More Information on the METAWIN project can be found here: `http://www.ftw.at/ftw/research/projects/`

[2]A diff package can be downloaded from the following homepage `http://userver.ftw.at/~vacirca/`.

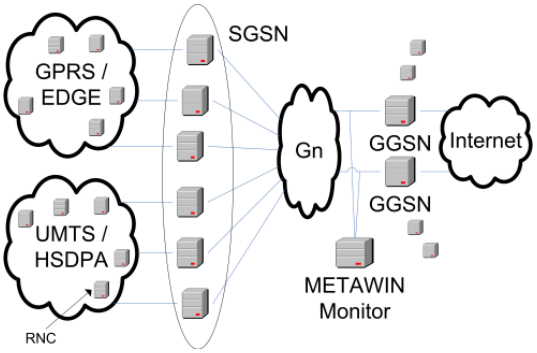


Fig. 1. Measurement Setup

III. Analysis of TCP Footprints

The core dataset used in this work consists of the full TCP statistics for one UMTS SGSN during five different periods: one day in March 2006, one day in September 2006, one day in April 2007, four hours around the maximum load in September 2007 and finally four hours around the maximum load in October 2007. The two most recent traces focus only at a time frame from 7 p.m. to 11 p.m., including the peak hour around 9 p.m. in the evening.

In [8] we discovered that an up-coming bottleneck will be visible around the daily peak throughput rate first. For the following section we reduced the older datasets to the same time frame. The measurements were taken from the live network of a major mobile provider in Austria, EU. Hereafter the datasets will be indicated by $\mathtt{S}$ with an index starting at one for the oldest and ranging up to five for the most recent trace (e.g., $\mathtt{S}_3$ represents the trace taken in April 2007).

Two samples are expected to be different from the rest: $\mathtt{S}_1$ and $\mathtt{S}_4$ were recorded with a bottleneck in place. The first sample was already used in the last publication and showed a clear difference when compared to footprints of normal operation. However, in the following the second trace will be of more interest, as we monitored the same GGSN one week later without the bottleneck and recorded it to $\mathtt{S}_5$. Now we are able to draw a direct comparison between traces taken at the same weekday with and without bottleneck.

The TCP statistics $\langle N_i, n_i \rangle$ were extracted for 30 minutes time bins in order to speed up the processing done with `tcptrace`. As the impact of a bottleneck is more evident during the peak hour we focus our analysis only on the period from 7 to 9 pm, for a total of four bins. We used scatterplots to visualize the process $\langle N_i, n_i \rangle$, i.e., to create "TCP footprints". As both variables span several orders of magnitude we introduced a logarithmic binning with 150 bins on each axis. The color of each pixel represents the number of occurrences within the bin.

We already described this procedure in [8]. There we compared one bottleneck trace with several different footprints

without bottleneck. However, with the new traces we are now able to see the long term evolution of the TCP footprints. This evolution is important to decide whether the approach we chose in [8] is applicable for anomaly detection.

A. Putting TCP Footprints to Scatterplots

In this paragraph we will present the unfiltered and not normalized footprints for the different traces recorded over the last two years. The following Fig. 2 depicts the scatterplots for S_{1-5}, in Figs. 2(a), 2(b), 2(c), 2(d), 2(e), and a GPRS footprint to extend the intra technology comparison in Fig. 2(f).

At a first glance we can see that there is a difference between the leftmost figures in line one and two and the other four footprints. In fact these two depict the bottleneck cases. The two figures have a strong positive correlation between N_i and n_i, especially for larger values of N_i. This was expected: In fact, a capacity bottleneck can be modeled as an element introducing random packet loss with a certain probability p on all flows. Hence the absolute number of retransmissions for each flow will be roughly proportional to the flow size, i.e. $n_i \propto p \cdot N_i$. In contrast to this all other footprints in Fig. 2 (without bottleneck) yield a much weaker correlation. Furthermore there is no significant difference in the shape of the footprints figures 2(b), 2(c), 2(e), which represent the non congested cases ($\mathrm{S}_{2,3,5}$).

It is striking how similar the GPRS sample is compared to the UMTS samples. This is an interesting observation, since the different radio technologies offer different kinds of capacity and therefore one could expect different shapes in footprints. We assume that the TCP footprint is caused by the services used, which are basically dominated by HTTP as we have shown in [9].

The shape of the footprint seems to be invariant for the non congested scenarios. However, the growth in traffic, which took place between the oldest and the most recent dataset, lead to a shift along the x-axis (N_i). In case of the congested footprints, Fig. 2(a) and Fig. 2(d), this shift is clearly visible. Also in these scenarios the shape is similar, however, the footprint in March 2006 shows less variance, which indicates that there was a stronger limitation in place than during September 2007.

Concluding this paragraph we can say that there is a clearly visible difference between a footprint affected by a bottleneck and a footprint recorded under normal operation conditions. In addition we see that there is only a slight difference in the shape after one year of network evolution and also between two fundamentally different technologies like UMTS and GPRS. The only visible difference between the figures is a shift in N_i toward higher values.

B. Matching the TCP Footprints

In this paragraph we apply four different metrics to analyze footprints for bottleneck situations. To make the benchmarking more robust to outliers we applied the following preprocessing steps. First we filtered the datasets using a median filter to remove outliers. In case a region of the scatterplot is only sparse populated a point by point comparison may fail, if bins containing events do not overlap. To avoid this problem we applied a mean filter with a window size of five to smooth the footprint. In a final step we normalized the value of the bins by the number of events in the traces. This step eliminates the growth of the population between the different measurement periods. The datasets are now equal to an empirical binned bi-dimensional probability density function.

In order to cope with the increase in N_i we calculated a center of gravity for each plot. In metrics that need a reference, like the correlation coefficient, we aligned the values for N_i before applying the metric. Please note that this was not done in our last publication. Therefore, the values may differ, in fact some values improve while other values degrade.

a) Correlation Coefficient: The first metric we tested was the 2-D correlation coefficient (Eq.(1)) for all permutations. Hence the parameter is relative simple to compute (see [10]) we decided to use it as a starting point. A correlation coefficient returns a dimensionless value in the interval of -1 to 1. A value of 1 indicates a perfect correlation while a value of 0 indicates that there is no dependency between the two parameters.

$$\Delta_C(A,B) = \frac{\sum_x \sum_y (A_{xy} - \overline{A}) \cdot (B_{xy} - \overline{B})}{\sqrt{\sum_x \sum_y (A_{xy} - \overline{A})^2 \cdot \sum_x \sum_y (B_{xy} - \overline{B})^2}} \tag{1}$$

The results of this first metric are presented in Table I. As the metric is symmetric we only write down the upper half of the resulting matrix.

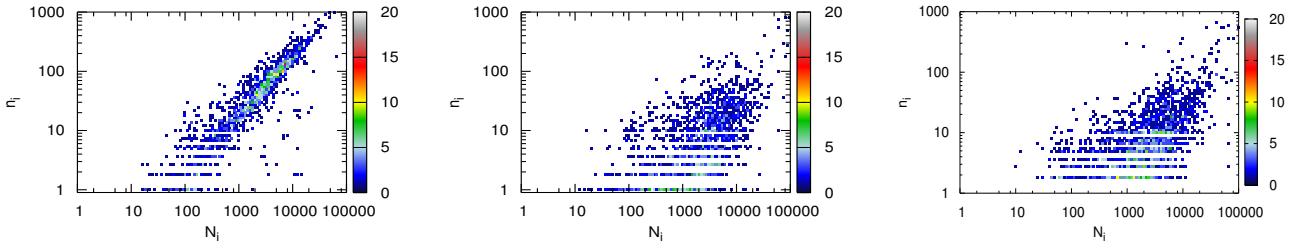
Δ_C	S_1	S_2	S_3	S_4	S_5
S_1	1.00	**0.41**	**0.27**	0.87	**0.43**
S_2	—	1.00	0.93	**0.49**	0.94
S_3	—	—	1.00	**0.38**	0.97
S_4	—	—	—	1.00	**0.48**
S_5	—	—	—	—	1.00

TABLE I

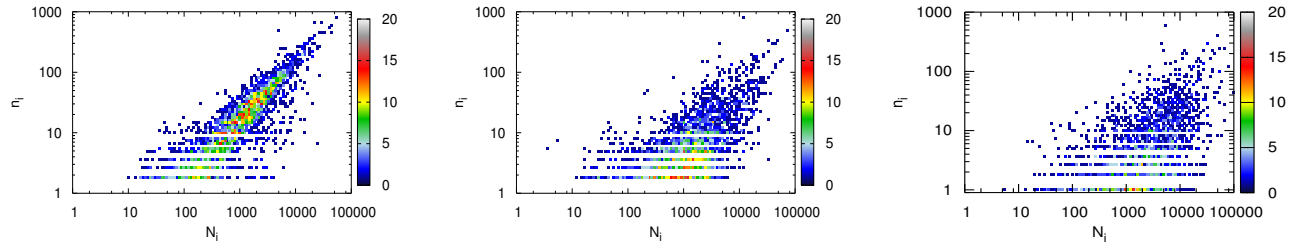
CORRELATION RESULTS (**NORMAL** VS. BOTTLENECK)

The results for the different datasets are similar to what we have already observed visually. The coefficients for non congested datasets, e.g., $\mathrm{S}_{2,3,5}$, are well above 0.9 indicating a high similarity. If a non congested footprints is compared with a congested one we get a value in the order of 0.45. Although sample S_4 originated from a weaker bottleneck condition the correlation coefficient does not exceed 0.5, leaving a gap of 0.4 as detect threshold. In case we compare the bottleneck traces, S_1 and S_4, we also get a high correlation of 0.87. We conclude that this metric is well suited to detect such kind of events.

b) Kullback-Leibler distance: The second metric we evaluated was the Kullback-Leibler (KL) distance between two datasets. The KL-distance is a distance measure between a given probability distribution P and an arbitrary distribution Q. Often P represents some reference data obtained by measurements and Q is generated by a model approximating



(a) S_1 (03-2006, **with** bottleneck) (b) S_2 (09-2006, without bottleneck) (c) S_3 (04-2007, without bottleneck)



(d) S_4 (09-2007, **with** bottleneck) (e) S_5 (10-2007, without bottleneck) (f) GPRS 09-2006 without bottleneck

Fig. 2. Scatterplot of N_i over n_i in the peak hours (log-binning, log scale)

P. Eq. (2), shows the KL-distance metric.

$$\Delta_{KL}(P\|Q) = \sum_{i,j} P(i,j) \cdot \ln \frac{P(i,j)}{Q(i,j)} \qquad (2)$$

The KL-distance is known to be very sensitive against changes, this is why we chose this metric. We applied the distance to our two dimensional domain. The logarithm in this equation was a problem for our datasets with empty bins, as in case Q is equal zero the fraction will be undefined. In this case it is not possible to use the common domain only, as it would neglect outliers outside this region. To overcome this problem we modified the equation in such a way that we only calculated the sum over all bins where P differs from zero. If we still encounter bins where Q equals zero we added a constant offset $c = 0.01$ to these bins of Q.

Note that the KL-distance is not, in general, symmetric (see [11]). We defined a symmetrized version of the metric as in Eq. (3), similar to [12].

$$\Delta_{KL_s}(P,Q) = \frac{1}{2} \cdot (\Delta_{KL}(P\|Q) + \Delta_{KL}(Q\|P)) \qquad (3)$$

The value of Δ_{KL_s} obtained by Eq. (3) is zero, if and only if P equals Q and larger than zero in any other case. A larger value of Δ_{KL_s} indicates a stronger deviation of P from Q.

The second metric calculates the distance values given in Table II. Again the matrix is symmetric and we omitted the lower part of the results. The KL-distance is equal to zero in case of equality, therefore every element on the main diagonal is equal to zero.

The basic results are identical to the first metric. The bottlenecks are clearly detected, note this time lower values indicate higher similarity. Again there is a clear gap between the two cases observed in the network.

Δ_{KL}	S_1	S_2	S_3	S_4	S_5
S_1	0.00	**1.74**	**3.65**	0.32	**1.59**
S_2	—	0.00	0.40	**1.20**	0.34
S_3	—	—	0.00	**1.75**	0.35
S_4	—	—	—	0.00	**1.23**
S_5	—	—	—	—	0.00

TABLE II
RESULTS FROM A SYMMETRIC KL-DISTANCE (**NORMAL** VS. BOTTLENECK)

The interesting result we obtain here is the fact that the manipulation of the center of gravity delivered a degraded performance in this case, compared to our previous publication. These problems come from the different distributions in case of bottleneck and non bottleneck scenarios. Therefore, a shift along the x-axis increases the KL distance. However, the new preprocessing leads to a higher detection sensitivity (larger gap for bottleneck cases).

c) Principle Component Analysis: The third metric is based on a PCA (Principle Component Analysis). In contrast to the first two metrics, this metric does not need a reference. This has several advantages, first we do not have to define a *reference* day in order to use our metric and second as we only consider the shape, we are independent from shifts in N_i, as seen over the last months.

The PCA is a mathematical transformation. Given that variance is a measure of information, the PCA tries to match the new coordinate system accordingly. It applies a linear orthogonal transformation which converts the original coordinates to a new coordinate system so that the projection of the greatest variance is identical with the first coordinate. This first coordinate is also called first principle component.

In practice PCA is often used to reduce dimensionality. This is possible by omitting higher principle components,

PCA	S_1	S_2	S_3	S_4	S_5
λ_1	33.3	26.8	27.7	31.1	25.8
λ_2	5.82	12.5	13.0	8.16	12.4
Frac_{pca}	**5.72**	2.14	2.13	**3.81**	2.08

TABLE III

RESULTS FROM A PRINCIPLE COMPONENT ANALYSIS (**NORMAL** VS. NORMAL)

PSNR	S_1	S_2	S_3	S_4	S_5
S_1	∞	**20.3**	**18.1**	24.9	**19.2**
S_2	—	∞	25.0	**18.9**	27.7
S_3	—	—	∞	**20.4**	30.9
S_4	—	—	—	∞	**21.6**
S_5	—	—	—	—	∞

TABLE IV

PSNR VALUES IN dB (**NORMAL** VS. BOTTLENECK)

depending on the energy left in those variables.

Given a set of N data vectors $\mathbf{x}_1 \dots \mathbf{x}_N$, where each vector $\mathbf{x}_n$ is a single observation of the M variables, here M is the dimension of the underlaying dataset, which is two in our case. We first generate a Matrix **X** of the size MxN, with one row per variable and one column per observation. To apply a rotation we have to remove the mean, in our case the empirical mean of **X**. Now we need a orthonormal transformation matrix **P** of the form:

$$\mathbf{Y} = \mathbf{P}^T\mathbf{X} \tag{4}$$

so that **cov(Y)** is a diagonal matrix. After some matrix manipulations we get the final results,

$$\mathbf{P}\text{cov}(\mathbf{Y}) = \text{cov}(\mathbf{X})\mathbf{P} \tag{5}$$

which shows that the new matrix **P** can be found by calculating the eigenvectors of **cov(X)**. However, as we only have one observation, we have to calculate an empirical covariance matrix.

Table III shows the results of the variance along the new axes in the rotated coordinate systems for each dataset in log scale. In case of a bottleneck, e.g., $S_{1,2}$, the variance in the second direction is smaller. This can also be interpreted with the fact that the footprint turns into a small ellipsoid like figure in the case of a bottleneck. Under normal operational conditions the footprint has a higher variance in the second component and a lower in the first, in other words the footprint is more round in this case. We then calculated the fraction, $\text{Frac}_{\text{pca}} = \lambda_1/\lambda_2$, for the variance in the first and the second component, e.g., the energy in the different components. Under normal operation Frac_{pca} stays around 2, while in a bottleneck scenario Frac_{pca} rises up to 4 in case s_4 or even up to 5 in case of S_1. There are several advantages in this method: first it does not need any preprocessing like the shift in N_i, second it can be used reference free and third it has the capability to detect the difference between the two bottlenecks. In fact s_1 was a quite heavy restriction in traffic, while S_3 was only an up-coming bottleneck, which is visible in the different results. However, the computational effort here is higher than it is in the PSNR case.

d) Peak Signal to Noise Ratio: We used a peak mean square error as a forth metric, also called PSNR (Peak Signal to Noise Ratio). This is a term from engineering, which is used to compare the maximum possible power of a signal and the power of a corrupting noise. It is commonly expressed in logarithmic scale. The PSNR is often used in image processing as a benchmark for image quality.

In a first step the mean square error is calculated over every bin of P and Q. The result is then normalized to the maximum level of P and Q. As we have renormalized these values already before to minimize the effect of a population growth, we can directly compare two scatterplots. This metric needs a reference, therefore we again would have to set a *reference* day in order to perform a benchmark. However, we will not consider this problem here. In case P equals Q the metric will go to infinity, see main diagonal in Table IV. A lower value indicates less similarity between P and Q.

$$\text{MSE}(P,Q) = \frac{1}{m \cdot n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \|P(i,j) - Q(i,j)\| \tag{6}$$

$$PSNR = 10 \cdot \ln \frac{I_{max}^2}{\text{MSE}} \tag{7}$$

The results improve if we align the center of gravity for both scatterplots. This comes from the fact that this metric is extremely sensitive to small shifts, whereas it becomes less sensitive for larger deviations, e.g., an error in the least significant bit of a single pixel results in a PSNR number of about 91dB, a further pixel error results in a degradation of 2.3dB. This metric gains most from a good alignment between the traces.

The values for the PSNR metric are given in Table IV. The numbers are in the order of 20dB in case we compare a bottleneck with a non bottleneck case, else they are around 25-30dB. There is a clear gap of more than 5dB usable for detection. This metric is computationally simple and effective in detecting the bottlenecks.

IV. PERFORMANCE ANALYSIS

In this paragraph we want to benchmark the performance of the different metrics for an artificial generated dataset. The new dataset $S_{new[x]}$ was generated by randomly taken samples from the two datasets S_1 and S_5. The value x indicates the percentage of samples taken from S_1, e.g., $S_{new[80]}$ is generated from samples that originate to 80% from S_1, or in other words this represents a relatively congested scenario.

Table V presents the detection result for the different metrics. In the correlation based metric we set the limit to raise an alarm to the lowest score for a bottleneck free dataset and subtracted 10% as detection threshold. The alarm is triggered in case that approximately 30% of traffic originate from the congested trace. For the KL-distance we took the largest distance from a bottleneck free operation and added 10% as a detection threshold. In this case the alarm was triggered for 25% of congested traffic. Applying the same rules for PSNR leads to a detection limit of 26%.

$s_{new[x]}$	Corr	KL-dist	PSNR	PCA
$x =$	30%	25%	26%	51%

TABLE V

SHARE OF CONGESTED TRAFFIC TO RAISE AN ALARM

At a first glance results for the PCA were a bit weired, it took more than 50% of congested traffic to raise an alarm. To explain this problem we have to go back to the footprint figures. Comparing Fig. 2(d) with Fig. 2(e) we see that the first figure, which shows the congestion, is shifted to higher values of n_i. If we now randomly add a small amount of values from a congested scenario to the normal operation, the variance of both the first and the second principle component is increased. If we reach a certain threshold the outer samples originating from the non congested trace start to disappear rapidly and in conjunction with this the variance of the second principle component starts to drop fast. In fact for 10–20% of congested traffic added, we obtain a lower or better Frac_{pca} than listed in Table III. However, detection on a decrease of Frac_{pca} is not reliable as there only is a small variation.

Concluding this benchmark we found out that KL performs best. However, the simple PSNR performs similar in this scenario. We propose the PSNR metric as it is the best trade off between complexity and detection threshold reached.

V. SUMMARY AND CONCLUSIONS

In this paper we present four different methods for non intrusive bottleneck detection in a cellular mobile core network based on counters for TCP related events. The datasets we used were recorded in a live 3G network at a major operator in Austria.

We worked with five different datasets spanning nearly one and a half year of network evolution, of which two represent known bottleneck problems in the network. A first interesting result was that there is no significant difference in shape between the different non bottleneck cases although there was a huge increase in throughput between 2006 and 2007. We observed only a slight shift in the size of TCP packets send per user, called N_i.

In this paper we used four different metrics, namely: the correlation coefficient, a modified KL-distance, a PSNR method and a principle component analysis. The first three metrics give a distance between the actual footprint and a reference. Therefore, one needs two references, e.g., one with and one without bottleneck. In contrast to these the PCA method needs no reference and can directly tell if there is a bottleneck or not. This last metric identifies the shape of the given footprint. All four metrics allowed a clear detection of the bottleneck.

It was interesting to see that also the second bottleneck, which had a much smaller impact on the footprint, is detected flawlessly. By applying a shift in N_i, e.g., aligning the center of gravity, the detection event even works for traces that where taken more than one year apart.

We concluded that the correlation coefficient is reliable enough to detect a bottleneck footprint. However, in case one does not have a reference bottleneck, and this may be true for most of the operators, the PCA is the best choice because it does not need a reference footprint.

In our following work we want to exploit the fact that the footprints in GPRS do look similar to the UMTS case. We will try to extend the detection to the total traffic in the network.

ACKNOWLEDGEMENTS

This work is supported by the K-Plus initiative of the Austrian government. We thank mobilkom austria AG and the Kapsch Karrier Com for technical and financial support of this work. The views expressed in this paper are those of the authors and do not necessarily reflect the views within mobilkom austria AG.

REFERENCES

[1] Fabio Ricciato, Francesco Vacirca, and Martin Karner. Bottleneck detection in umts via tcp passive monitoring: a real case. In *CoNEXT'05: Proceedings of the 2005 ACM conference on Emerging network experiment and technology*, pages 211–219, New York, NY, USA, 2005. ACM Press.

[2] F. Ricciato, F. Vacirca, and P. Svoboda. Diagnosis of capacity bottlenecks via passive monitoring in 3G networks: an empirical analysis. *Computer Networks*, 57:1205–1231, March 2007.

[3] R. Prasad, C. Dovrolis, M. Murray, and K. Claffy. Bandwidth estimation: metrics, measurement techniques, and tools. *Network, IEEE*, 17(6):27–35, 2003.

[4] F. Vacirca, F. Ricciato, and R. Pilz. Large-scale RTT measurements from an operational UMTS/GPRS network. In *Proc. of the First International Conference on Wireless Internet (IEEE WICON 05)*, 2005.

[5] H. Holma and A. Toskala. *WCDMA for UMTS, Radio Access For Third Generation Mobile Communications, Third Edition*. Wiley, 2004.

[6] Fabio Ricciato. Unwanted traffic in 3G networks. *ACM Computer Communication Review*, 36, 2005.

[7] R. Pang et al. Characteristics of Internet Background Radiation. *Proc. of the International Measurements Conference (IMC'04), Taormina, Sicily, Italy.*, October 2004.

[8] P. Svoboda, F. Ricciato, and M. Rupp. Bottleneck footprints in TCP over mobile internet accesses. *Communication Letters*, 11(11):839–841, 2007.

[9] P. Svoboda and F. Ricciato. "Composition of GPRS and UMTS traffic: snapshots from a live network". *IPS MoMe 2006, Salzburg*, pages 42–44, Feb 2006.

[10] T.C Moon and W.C. Stirling. *Mathematical Methods and Algorithms for Signal Processing*. Prentice-Hall, 2000.

[11] S. Kullback and R.A. Leibler. On information and sufficiency. *Ann. Math. Stat.*, 22:79–86, 1951.

[12] D. Johnson and S. Sinanovic. Symmetrizing the Kullback-Leibler distance. *Technical Report*, 2001.