Anomaly detection in diurnal data

Felipe Mata\textsuperscript{a,*}, Piotr Żuraniewski\textsuperscript{b,c,d}, Michel Mandjes\textsuperscript{b}, Marco Mellia\textsuperscript{e}

\textsuperscript{a}High Performance Computing and Networking Group, Universidad Autónoma de Madrid, Spain
\textsuperscript{b}Korteweg-de Vries Instituut voor Wiskunde, University of Amsterdam, The Netherlands
\textsuperscript{c}TNO, Delft, The Netherlands
\textsuperscript{d}AGH University of Science and Technology, Kraków, Poland
\textsuperscript{e}Dipartimento di Elettronica e Telecomunicazioni, Politecnico di Torino, Italy

A R T I C L E  I N F O

Article history:
Received 14 June 2012
Received in revised form 18 November 2013
Accepted 19 November 2013
Available online 27 November 2013

Keywords:
Anomaly detection
Diurnal pattern
Detrending
Changepoint
VoIP

A B S T R A C T

In this paper we present methodological advances in anomaly detection tailored to discover abnormal traffic patterns under the presence of seasonal trends in data. In our setup we impose specific assumptions on the traffic type and nature; our study features VoIP call counts, for which several traces of real data has been used in this study, but the methodology can be applied to any data following, at least roughly, a non-homogeneous Poisson process (think of highly aggregated traffic flows). A performance study of the proposed methods, covering situations in which the assumptions are fulfilled as well as violated, shows good results in great generality. Finally, a real data example is included showing how the system could be implemented in practice.

1. Introduction

Network operators and service providers have taken a keen interest in managing the Quality of Service (QoS), and how it is perceived by their end-users (Quality of Experience). In this light, a broad range of techniques have been proposed to detect QoS degradation, see e.g. [1]. Some of these specifically focus on the Voice over Internet Protocol (VoIP) service [2], where performance degradation (due to packet loss, and increased delay/jitter) occurs during periods with high loads. Consequently, timely detection of such overload periods is crucial for management of VoIP services [3], as they enable a better cost control if applied in an automated fashion [4]. Such automated techniques rely on the statistical analysis of network traffic measurements, which commonly assumes stationarity of the data. A complication, however, is that network traffic measurements can usually not be considered as stationary, but rather exhibit a, roughly periodic, diurnal (day-night) pattern.

The violation of the stationarity assumption may lead to erroneous conclusions [5], in terms of large amounts of false positives/negatives. To remedy this, we propose in this paper (which builds on the results of [6]) a simple, yet effective methodology for removing the inherent daily pattern; in our study VoIP call counts data serves as the leading example. The methodology relies on the fact that the call arrival process is time-varying Poisson, which we show to be valid for the data of our case study. After removing the daily trend, we obtain standardized samples (i.e., zero mean and unit variance) that are nearly Normally distributed, as long as there is sufficient traffic aggregation—as a consequence, the fit improves when the night periods are removed from the sample (in which the chances of overload are negligible anyway). The (nearly Normal) output samples are not (by approximation) independent, though, which is problematic as this is required in many detection algorithms. To mitigate this effect, we propose an alternative measurement methodology that reduces the correlation for an important class of call holding
time distributions. Specifically, we show that when the call holding time distribution follows a Pareto or Log-Normal distribution, our alternative measurement procedure tends to outperform the traditional approach; this is also the case for mixtures of two Log-Normals and a Pareto distribution, which is actually the best fitting model for our VoIP data. To assess the efficacy of the resulting procedure, we have modified the overload detection methodology presented in [3] to work with Normally distributed input data and extensively tested its performance, including situations in which the independence assumption is violated; these tests convincingly show that our approach works well in great generality.

The rest of the paper is organized as follows: Section 2 presents related work. A description of the dataset is presented in Section 3. After describing how to remove the diurnal trend from the VoIP call count data in Section 4, we present the alternative measurement technique in Section 5. Next, we provide in Section 6 a description and performance evaluation of the overload detection methodology. Finally, Section 7 concludes the paper.

2. Related work

The analysis of traffic in communication networks has attracted much attention; see e.g. [7]; also its evolution in time has been studied [8]. The ubiquitous daily pattern evidently depends on the kind of users that access the network, although it can be deemed as roughly invariant (having a similar shape from day to day, that is) [4].

These users can be divided into two main groups: enterprise users, who access the network in their workplaces, and domestic users, accessing the network from their residences. The enterprise users’ daily pattern is directly related to the office working hours, i.e., the load is larger during working hours, and usually there appear two clearly distinguishable peaks—before and after lunchtime (see [4] for a study on the Spanish Academic Network RedIRIS). The domestic users’ pattern is also influenced by the working hours, but obviously in the opposite way: the load is larger after the usual working hours (see [9] for such a study held within the European project TRAMMS).

This shape invariance is actually observed at different timescales. For instance, on a weekly basis, the shape of the pattern is approximately the same from Monday up to Thursday. On Fridays, we observe a scaled version of the other working days pattern i.e., the shape is essentially the same, but the load is somewhat smaller. Finally, on weekends and holidays, we found almost flat patterns when dealing with enterprise measurements—principally due to applications that are left running and generate traffic without user interaction.

A similar conclusion holds when focusing on VoIP traffic only, see e.g. [8,10]. Here it is noted that these VoIP-related studies primarily focus on call characteristics (in terms of the call arrival process and call holding time distribution) rather than daily/weekly patterns. The call arrival process is widely accepted to be accurately modeled by a time-inhomogeneous Poisson process (roughly stationary at short timescales, ranging from minutes to hours [8,11]). Conversely, there is no consensus as to which model should be used for the call holding times (where it is clear that the exponential distribution is not a good candidate). A broad range of distributions have been proposed, such as the hyper-exponential [8], the inverse Gaussian [10], and the Log-Normal [12]. The trend-removal issue can be approached relying on general traffic forecasting techniques [13], or by time series with seasonal cycles [14].

3. Dataset description

The experiments reported on in this paper are using actual traffic traces collected from an operational network. Using Tstat [15], we monitored IP traffic exchanged by customers in a large Point of Presence (POP) of an operator in Italy where VoIP is deployed. A total of about 22,000 customers were continuously monitored for more than 4 months, starting from November 2010. Tstat was used to identify VoIP flows, i.e., voice calls, and to extract several performance indexes for each call [10]. In particular, in the context of the present paper we are interested in the call arrival process and call holding time distribution. The resulting dataset contains the log of the call arrival epochs and the corresponding durations. Later in this paper, we statistically analyze these, and use the resulting processes/distributions to assess the performance of our algorithm. The dataset containing start and end times of the calls will be referred to as detailed below.

However, storing detailed call records is not practical for history analysis in large-scale networks. In these cases, summarized statistics are stored instead, and detailed logs are kept for short time periods (say several days). Anomaly detection is performed in the summarized datasets, and the detailed records are used for further forensic analysis on the relevant events encountered. Consequently, we build a summarized dataset from the detailed logs for its application in the trend removal methodology described in Section 4. To construct the summarized dataset, we adopt the traditional way that records the number of calls present in the system at equidistant points in time (e.g., \( N_0, N_1, N_2, \ldots \)). The separation \( t \) between records is set to 5 minutes for this dataset, based on the results presented in the following subsection.

3.1. Call arrival process

The Poisson process is the classical model for the arrival process of voice calls. Evidently, at longer timescales this model does not match with reality, due to the absence of a day-night pattern (and a weekly pattern). To cope with this effect, non-homogeneous Poisson processes are used instead, where the arrival rate is usually assumed constant for blocks of time, of say, \( L \) minutes. To verify the ‘local Poisson claim’ for some \( L \), we apply to our detailed dataset a test presented in [11]. To construct the test, we split up a day into disjoint blocks of length \( L \), resulting in a total of \( I \) blocks. Let \( T_{ij} \) be the \( j \)th arrival time in the \( i \)th block. Denoting with \( J_i \) the total number of arrivals within the \( i \)th block, we then define \( T_{i0} = 0 \) and for \( j = 1, \ldots, J_i \) and \( i = 1, \ldots, I \).
Under the null hypothesis (arrival rate is constant within each block), the $R_i$ are independent standard exponential variables; see [11] for further background and a justification of the test.

In Table 1 we present the results of applying the test to different block sizes $L$. We use the Kolmogorov-Smirnov (KS) test to verify the null hypothesis at the 5% significance level. The results presented in the table indicate that the arrival process can be regarded as non-homogeneous Poisson at relatively short timescales only, say less than 10 minutes. This motivates the choice of $t = 5$ minutes for building our summarized dataset.

3.2. Call Holding Time (CHT) distribution

In the literature it is generally concluded that the CHT is poorly modeled by the exponential distribution. Instead, distributions with heavier tails have been proposed. A visual inspection using log-log plots of the empirical Complementary CDF (CCDF) of the sample, which allow us to gain insight in the tail of the distribution, evidences the heavy-tailed nature in our dataset as well. Consequently, in our Goodness-of-Fit (GoF) assessment we restrict ourselves to heavy-tailed distributions (that is, heavier than the exponential distribution). This mixture model is capable of fitting the whole body of the data, while there is a slight lack of fit in the very end of the tail (which is in fact caused by 10 observations out of more than 10$^5$).

To measure the GoF we use again the KS test. Note however, that the applicability of this test is in principle not justified, as we are estimating the parameters of the hypothesized models from the sample [16]; as a result in this case the critical values of KS test statistics, widely used in statistical software packages, are not correct. In order to avoid this (unfortunately very common) mistake we have calculated the KS critical values for each of our mixtures of the distributions by performing a Monte Carlo experiment, being the counterpart of the one described in [17].

Table 1

<table>
<thead>
<tr>
<th>$L$ (min)</th>
<th>Rejection %</th>
<th>$L$ (min)</th>
<th>Rejection %</th>
</tr>
</thead>
<tbody>
<tr>
<td>90</td>
<td>74</td>
<td>25</td>
<td>27</td>
</tr>
<tr>
<td>60</td>
<td>61</td>
<td>20</td>
<td>19</td>
</tr>
<tr>
<td>45</td>
<td>50</td>
<td>15</td>
<td>14</td>
</tr>
<tr>
<td>35</td>
<td>39</td>
<td>10</td>
<td>9</td>
</tr>
<tr>
<td>30</td>
<td>35</td>
<td>5</td>
<td>7</td>
</tr>
</tbody>
</table>

Table 2

<table>
<thead>
<tr>
<th>Distribution Parameters</th>
<th>KS statistic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distribution Parameters</td>
<td>KS statistic</td>
</tr>
<tr>
<td>Pareto + 2 Log-Ns</td>
<td>$p = 0.6793$</td>
</tr>
<tr>
<td></td>
<td>$p = 0.2023$</td>
</tr>
<tr>
<td></td>
<td>$p = 0.1184$</td>
</tr>
<tr>
<td></td>
<td>$\sigma = 63.1807$</td>
</tr>
<tr>
<td></td>
<td>$\sigma = 0.9523$</td>
</tr>
<tr>
<td></td>
<td>$\sigma = 0.5201$</td>
</tr>
<tr>
<td>2 Log-N</td>
<td>$\mu = 0.1089$</td>
</tr>
<tr>
<td></td>
<td>$p = 0.8911$</td>
</tr>
<tr>
<td></td>
<td>$\mu = 3.6421$</td>
</tr>
<tr>
<td></td>
<td>$\sigma = 0.4810$</td>
</tr>
<tr>
<td></td>
<td>$\sigma = 1.5528$</td>
</tr>
<tr>
<td>Weibull + Log-N</td>
<td>$p = 0.9068$</td>
</tr>
<tr>
<td></td>
<td>$p = 0.9032$</td>
</tr>
<tr>
<td></td>
<td>$\lambda = 42.7199$</td>
</tr>
<tr>
<td></td>
<td>$\mu = 4.2964$</td>
</tr>
<tr>
<td></td>
<td>$k = 2.4978$</td>
</tr>
<tr>
<td></td>
<td>$\sigma = 1.5385$</td>
</tr>
<tr>
<td>Log-N</td>
<td>$p = 1$</td>
</tr>
<tr>
<td></td>
<td>$\mu = 4.2218$</td>
</tr>
<tr>
<td></td>
<td>$\sigma = 1.4882$</td>
</tr>
</tbody>
</table>

Fig. 1. CCDF log-log plots of the data and the best fitting models according to the KS statistic value.

known as Lilliefors Normality test. Table 3 summarizes the results (critical values or CVs) of the simulation study in which for each distribution mixture there were 1000 independent runs and the largest sample size was $n = 100,000$ (by using smaller sample sizes we have observed the expected $\sqrt{n}$ dependency, just like in [17], so for the reader’s convenience the table contains CVs multiplied by a $\sqrt{n}$ factor). After substituting the sample size of $n_0 = 1,289,414$ of the analyzed dataset and calculating CVs, we see that the null-hypothesis is rejected at any significance level $\alpha$. This was to be expected, taking into account the effects observed in the tail. The KS statistic can then only be used as a measure of model discrepancy, so as to select the ‘best’ (non-conforming but being close enough) model. Treating the 10 largest observations as outliers and removing them would allow for a much better fit for Pareto + 2 Log-Ns, but in fact such long calls (despite being rare) may be of interest to the system administrator. An alternative approach is to keep them, and to use any of the other fitted distributions which bounds the tail of the survivor function from above (2 Log-N, Weibull + Log-N or Log-N). Following the latter approach means accepting even a larger misfit, but the advantage is that we are conservative rather than too optimistic about the possible call length duration.
Table 3

Critical values for KS test when parameters are estimated from the sample.

<table>
<thead>
<tr>
<th>Distribution</th>
<th>Parameters</th>
<th>20%</th>
<th>15%</th>
<th>10%</th>
<th>5%</th>
<th>1%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pareto +</td>
<td>$\sqrt{n} CV(n)$</td>
<td>0.44564</td>
<td>0.45586</td>
<td>0.46676</td>
<td>0.49232</td>
<td>0.54885</td>
</tr>
<tr>
<td>2 Log-Ns</td>
<td>$CV(n_0)$</td>
<td>0.00039</td>
<td>0.00040</td>
<td>0.00041</td>
<td>0.00043</td>
<td>0.00048</td>
</tr>
<tr>
<td>2 Log-N</td>
<td>$\sqrt{n} CV(n)$</td>
<td>0.54226</td>
<td>0.55955</td>
<td>0.58928</td>
<td>0.64841</td>
<td>0.73210</td>
</tr>
<tr>
<td>CV($n_0$)</td>
<td>0.00048</td>
<td>0.00049</td>
<td>0.00052</td>
<td>0.00057</td>
<td>0.00064</td>
<td></td>
</tr>
<tr>
<td>Weibull</td>
<td>$\sqrt{n} CV(n)$</td>
<td>0.57293</td>
<td>0.60242</td>
<td>0.64042</td>
<td>0.70595</td>
<td>0.79213</td>
</tr>
<tr>
<td>+ Log-N</td>
<td>$CV(n_0)$</td>
<td>0.00050</td>
<td>0.00053</td>
<td>0.00056</td>
<td>0.00062</td>
<td>0.00070</td>
</tr>
<tr>
<td>Log-N</td>
<td>$\sqrt{n} CV(n)$</td>
<td>0.75564</td>
<td>0.79675</td>
<td>0.83370</td>
<td>0.92035</td>
<td>1.07504</td>
</tr>
<tr>
<td>CV($n_0$)</td>
<td>0.00067</td>
<td>0.00070</td>
<td>0.00073</td>
<td>0.00081</td>
<td>0.00095</td>
<td></td>
</tr>
</tbody>
</table>

4. Detrending methodology

4.1. Methodology description and expected results

Our methodology exploits the presence of a weekly pattern to estimate and remove the seasonality from the measurements, so as to obtain a sequence of zero-mean, unit-variance observations. In our setup the measurements are time series of traffic metrics (think of number of active calls) at a given time granularity—5 minutes in our study. These measurements are denoted by $x_i^n$, with $i = 0, 1, 2, \ldots, 2015$ corresponding to the 5-minute intervals within the week (starting on Monday midnight), and $n$ to the week number within the dataset (out of a total of $N$ weeks). The goal is to provide a good estimate $\theta^n$ for the measurement vector of week $n$, $x^n$, using the information available from previous weeks, $x^j$, $j < n$, assuming the differences from week to week in the weekly pattern to be due to random deviations from an average network usage pattern. To this end, we propose the following model for the measurements: $x^n = a^n + e^n$, where $a^n$ denotes the average pattern and $e^n$ are the random deviations from such pattern.

The proposed estimation procedure computes the trend $\theta^n$ as an arithmetic average of the observations of the last $w = 5$ weeks. This window size balances accuracy and robustness to pattern shifts quite well; we have also tested different averaging processes, but the differences in performance are negligible.

We can then remove the estimated pattern from the actual measurements, so as to obtain zero mean residuals. Recalling that our arrival process is locally Poisson (cf. Section 3.1), and because Poisson variates with a high mean are approximated well by the Normal distribution, the resulting residuals (by approximation) form a sequence of zero mean Normally distributed random variables. Note, however, that they are not homoscedastic (that is, they do not have a uniform variance). Therefore they need to be standardized, which can be done by dividing each residual by its standard deviation. Instead of designing another model for estimating the pattern variance, we exploit the fact that for Poisson random variables, the mean and the variance are equal. Hence, we obtain standardized residuals through

$$r^n = \frac{x^n - \theta^n}{\sqrt{\theta^n}}. \quad (2)$$

It is remarked that we use bold font to denote week vectors.

4.2. Model performance results

We have computed the (standardized) residuals in our dataset using (2). For the sake of brevity, we only show the results for one week, which turned out to be highly representative.

Fig. 2(a) shows the estimated pattern, computed from the previous weeks’ samples, superimposed on the actual samples of the week under study. It shows the goodness-of-fit of the estimated pattern relative to the actual measurements, which turns out to be extremely accurate due to the high degree of stability of the weekly pattern. The main differences appear at the troughs of the weekly pattern, at which the samples show abrupt variations. The corresponding residuals are shown in Fig. 3(a).

We found that the variance of the residuals was substantially higher than expected, with some values exceeding 5%. This effect is also observed in the corresponding Gaussian Q-Q plot, shown in Fig. 4(a); the deviations from the straight line indicate that the tails are non-Normal. This discrepancy is mainly caused by the fact that during nights the load decreases drastically, thus leading to large residuals (due to the small denominator in (2)). In the rest of our analysis, we have therefore decided to remove the nights (defined as periods from midnight to 6 a.m.), which is justified as the chances of overload during the nights are negligible anyway. In fact, if the load during the night is high, such as in an international hub where the day-night patterns are less pronounced, we may use all the data without compromising the algorithm’s accuracy. If needed, the time instants the algorithm is able to analyze with high accuracy may be determined by a threshold for the load based on the target accuracy and the probability of having an overload period during the time instants not analyzed. However, we use a fixed night period to be ruled out throughout this article for the sake of clarity.

The corresponding estimated pattern as well as the residuals, with and without night periods, are shown in Fig. 2 and Fig. 3, respectively. We observe that leaving out nights reduces the variance substantially. In addition, Fig. 4, which shows the Gaussian Q-Q plot of the residuals with and without nights, indicates that Normality (approximately) holds in the latter case (that is, the fit is considerably better than in the situation that nights were included). Fig. 5 presents the autocorrelations of the residuals, both in the case that nights were included and the case they were left out. These graphs show that the residuals are not independent—note that the horizontal lines indicating the 95%
confidence interval around 0 are exceeded, particularly for the first lags; we return to this issue later in great detail. In addition, there is a periodic component in the residuals. As this periodicity, which is likely to be due to the inherently simple nature of the detrending procedure, is relatively weak, we do not take it into account in our study.

It is important to realize that statistical overload detection procedures usually assume that the observations used are independent. As a result, when using our residuals in such a test, the high correlation may lead to a significant performance degradation (in terms of high numbers of false positives and negatives). We are inclined to think that

![Fig. 2. Data samples for the week under study and estimated pattern based on previous weeks data samples: (a) nights included; (b) nights removed.](image1)

![Fig. 3. Residuals obtained after standardization with the estimated pattern: (a) nights included; (b) nights removed.](image2)

![Fig. 4. Gaussian quantile–quantile plots of the residuals: (a) nights included; (b) nights removed.](image3)
the correlation in the residuals is due, among others, to the simplicity of our trend estimation model, which is not able to adapt dynamically—at a short timescale, say hours—to deviations from the pattern. As a consequence, when actual measurements are above the estimated pattern from previous weeks, there is a high probability that this situation would remain the same for the next samples, and vice versa. Nonetheless, we have decided to keep the model as simple as possible at the expense of modest performance degradations (which can be controlled as described in Section 6).

5. Measurement alternative

We have observed above that call holding times exhibit heavy-tailed features, involving Log-Ns and potentially Pareto or Weibull distributions. As a result, residuals cannot be assumed independent, and this is a complication as many changepoint detection procedures (such as theCUSUM-based one highlighted in Section 6.1) rely on such an independence assumption. In Section 5.1 we propose an alternative measurement procedure that has the potential of reducing the correlation drastically. As will be demonstrated in Section 5.2 for the call holding times that fit our data best, the correlation indeed go down substantially.

5.1. Alternative procedure

As indicated earlier, the correlations in the residuals of our detrending procedure are strongly affected by the way the measurements are obtained. In this section we analyze an alternative to the traditional one described in Section 3, and compare their performance (in terms of correlation). More precisely, we define \( M_a \) as the number calls that have been present during the interval \([a, a + t]\), for a given \( t \).

To evaluate the performance of this alternative procedure, we compute the correlation between two measurements at different time instants—e.g., \( M_0 \) and \( M_{15} \). To simplify the following computations, we assume that the arrival process is Poisson with constant arrival rate \( \lambda \). Consequently, we obtain

\[
\text{Corr}(M_0, M_{15}) = \frac{\text{Cov}(M_0, M_{15})}{\sqrt{\text{Var}(M_0)}}, \tag{3}
\]

using the Poissonian and stationarity assumptions. To compute (3), we define

\[
A_a = \{ \# \text{ arrivals up to time } a \text{ that depart in } [a, a + t] \},
\]

\[
B_a = \{ \# \text{ arrivals in } [a, a + t] \text{ that depart in } [a, a + t] \},
\]

\[
C_a = \{ \# \text{ arrivals in } [a, a + t] \text{ that are still present at } a + t \},
\]

\[
D_a = \{ \# \text{ arrivals up to time } a \text{ that are still present at } a + t \},
\]

thus obtaining the following identity:

\[
M_a = A_a + B_a + C_a + D_a. \tag{4}
\]

Now notice that calls that are there at the end of the first interval are potentially still present at the beginning of the second interval, so that we only have to take into account \( C_0 \) and \( D_0 \). Also, only calls that arrived before the beginning of the second interval can interact, so for the same reason we only need to include \( A_{15} \) and \( D_{15} \). Consequently, (3) can be rewritten as follows:

\[
\text{Corr}(M_0, M_{15}) = \frac{\text{Cov}(C_0 + D_0, A_{15} + D_{15})}{\sqrt{\text{Var}(A_{15} + B_0 + C_0 + D_0)}}. \tag{5}
\]

Also realize that for all \( a \in \mathbb{R} : C_a + D_a = N_{a,t} \) and \( A_a + D_a = N_a \). Therefore, the numerator in (5) is

\[
\text{Cov}(N_a, N_{a,t}) = \rho \mathbb{P}(S_r > (k - 1)t), \tag{6}
\]

where in the last identity we have defined \( \rho = \lambda \mathbb{E}[S] \). \( S \) being the service time distribution (with finite mean), and \( S_r \) its excess lifetime; in A we provide a demonstration of the equality in (6). It is also true that \( B_a + C_a = F_a \) is the number of arrivals in the interval \([a, a + t]\). This can be used to compute the mean of \( M_a \) for all \( a \in \mathbb{R} \):

\[
\mathbb{E}[M_a] = \mathbb{E}[N_{a,t} + F_a] = \rho + \lambda t = \rho \left(1 + \frac{t}{\mathbb{E}[S]}\right). \tag{7}
\]

Finally, using (6) and (7), we obtain the result for (3):

\[
\text{Corr}(M_0, M_{15}) = \frac{\mathbb{P}(S_r > (k - 1)t)}{\rho (1 + \frac{t}{\mathbb{E}[S]})} = \frac{\mathbb{P}(S_r > (k - 1)t)}{1 + \frac{t}{\mathbb{E}[S]}}, \tag{8}
\]

It is worth noting that in the previous computations we did not make any assumption regarding the service time distribution, which means that (8) holds for any kind of service time distribution given that the arrival process is Poisson.

5.2. Correlations study: impact of service time distribution

We now compare the correlation resulting from (8) with the one from the traditional call count process \( N_t \), assuming specific service time distributions; the main
question is whether or not \( \text{Corr}(M_0, M_k) < \text{Corr}(N_0, N_k) \); if yes, then the correlation is reduced. We restrict ourselves to three service time distributions, viz. exponential, Pareto and log-normal. We rely on the following identity from renewal processes (see [18], Section 14.3) for the excess lifetime distribution:
\[
P(S > y) = \frac{1}{\mathbb{E}[S]} \int_y^\infty P(S > t) \, dt.
\] (9)

5.2.1. Exponential distribution

In this subsection we assume that the call service time \( S \) is exponentially distributed with parameter \( \mu \), i.e., for \( s \geq 0 \), the distribution function reads \( P(S \leq s) = 1 - e^{-\mu s} \). As is well-known, the excess lifetime distribution is exponential with parameter \( \mu \) as well. Now it can be checked easily that verifying \( \text{Corr}(M_0, M_k) < \text{Corr}(N_0, N_k) \) reduces to verifying \( e^{\mu s} < 1 + \mu t \). The latter inequality (obviously) does not hold. As a consequence, for exponential service times, our measurement alternative does not lead to a reduction of the correlations. Recall, however, that the exponential distribution is not suitable for VoIP call holding times (see Section 2).

5.2.2. Pareto distribution

In this subsection we assume that the call holding time \( S \) is Pareto distributed with parameter \( \alpha > 1 \), that is, \( P(S \leq s) = 1 - (1 + s)^{-\alpha} \), for \( s \geq 0 \). As is checked easily, the corresponding excess lifetime distribution is Pareto with parameter \( \alpha - 1 \). It turns out that we need to check whether
\[
f(t) := \left(1 - \frac{t}{1 + kt}\right)^{1-\alpha} < 1 + (\alpha - 1)t :=: g(t).
\] (10)

To find a sufficient solution for this new inequality, we will use the fact that, if \( f(0) < g(0) \), then \( f'(t) < g'(t) \forall t > 0 \) guarantees \( f(t) < g(t) \). Applying this argument three times, we obtain the following sufficient condition: \( k > \alpha/2 \). However, we observed by numerical analysis that the condition is even less restrictive: for some cases with \( k < \alpha/2 \) the inequality still holds, and the alternative procedure is to be preferred.

5.2.3. Log-normal distribution

We now study the correlations when the service time distribution is Log-Normal. The excess lifetime distribution is hard to work with, and therefore we resort to a numerical approach to study this distribution. We include this distribution because it is, in addition to the Pareto distribution, the most commonly used call holding time distribution (Section 2).

We have used numerical integration to obtain the excess lifetime probability for different values of the Log-Normal parameters \( \mu \) and \( \sigma \) at different time lags \( k \). We use these results to compute \( \text{Corr}(N_0, N_k) - \text{Corr}(M_0, M_k) \); see the contour plots in Fig. 6 for the most representative values of \( k \).

The most prominent conclusions of Fig. 6 are:

- At the first lags (\( k = 1 \) in Fig. 6(a) and \( k = 2 \) in Fig. 6(b)), the traditional method is slightly better for large values of \( \mu \) but small of \( \sigma \), while for other pairs \((\mu, \sigma)\) the alternative approach works better.
- However, as we move to larger lags, the situation improves for the proposed alternative, as shown in Figs. 6(c) and (d).

5.2.4. Correlation for call holding time distribution

Above we compared \( \text{Corr}(M_0, M_k) \) and \( \text{Corr}(N_0, N_k) \) for various call holding time distributions. We now consider the best fitting models obtained in Section 3.2 (a simple Log-Normal fit, and the mixture with one Pareto and two Log-Normal components). Fig. 7 shows the difference between both models, indicating that the alternative procedure outperforms the traditional approach at all time lags. The difference between both correlations is maximal at the first lags, which is the timescale of interest for the purposes of the methods presented in this paper. We have then showed that the alternative measurement procedure reduces correlations in situations of practical interest, and it is therefore more suitable to be used in statistical detection procedures.

6. Overload detection methodology

In [3] an overload detection algorithm is studied for an \( M/G/\infty \) queuing system, relying on the testing framework of [19], Section VI.E. It was also pointed out in [3] that the detection procedure in [19] for a changepoint in the mean of i.i.d. Normally distributed samples (with known and constant variance) can be adapted to a changepoint in the variance (with known and constant mean). (Semi-)closed-from results were included in [3], but the performance of the resulting test was not evaluated. Furthermore, the case of a simultaneous change in both mean and variance was not covered in [3].

Below we provide a detailed analysis featuring a procedure to detect a simultaneous change of the mean and variance. We assess its performance in case the independence assumption is fulfilled, but also when it is violated. The latter case is obviously highly relevant in our VoIP context: as we saw in Section 4, our detrending method leaves some autocorrelation in the residuals. If the changepoint detection method developed for the independent case works sufficiently well (at least up to some specific level of correlation), it may be an attractive and viable alternative to the idea of explicitly incorporating correlation into the test (which will lead to a considerably more complicated procedure).

6.1. Changepoint detection

We wish to detect a changepoint in the Normally distributed data, that is, whether during our observation period the parameter vector \((\mu, \sigma)\) (which we denote as the probability model \( P \) ) changes into \((v, q) = (\mu, \sigma)\) (the model \( Q \) ). More formally, we consider the following (multiple) hypotheses. Let \( X_k \) be the sequence of independent observations obtained from the Normal distribution.
are distributed according to a Normal random variable with parameters vector \((\mu, \sigma)\).

\[ H_0: \ (X_i)_{i=1}^n \text{ are distributed according to a Normal random variable with parameters vector } (\mu, \sigma). \]

\[ H_1: \text{ For some } \delta \in \{1/n, 2/n, \ldots, (n-1)/n\}, \text{ it holds that } \ (X_i)_{i=1}^n \text{ is distributed according to a Normal random variable with parameter vector } (\mu, \sigma), \text{ whereas } (X_i)_{i=1}^n \text{ is distributed according to Normal random variable with parameters } (\nu, \eta) \neq (\mu, \sigma). \]

Following the notation used in [3], we consider the likelihood-ratio test statistic (cf. the Neyman-Pearson lemma):

\[
\max_{\delta \in [0,1]} \left( \frac{1}{n} \sum_{i=1}^n L_i - \varphi(\delta) \right), \quad \text{with } L_i := \log \frac{Q(X_i)}{P(X_i)}
\]

(11)

for some function \(\varphi(\cdot)\) which is defined shortly; here \(P(\cdot)\) is to be interpreted as the density of the \(X_i\) under the probability model \(P\), and \(Q(\cdot)\) its counterpart under the model \(Q\).

If the test statistic is larger than 0, we reject \(H_0\); this procedure can be seen as member of the class of CUSUM (cumulative sum) based tests. The function \(\varphi(\cdot)\) is introduced to get an essentially uniform alarm rate with respect to \(\delta\). As in [3], \(\varphi(\cdot)\) is given in the implicit form using Legendre transforms \(I(\vartheta) = \sup_x (\vartheta x - \log M(\vartheta))\) of the moment generating function \(M(\vartheta)\), which is defined as

\[
M(\vartheta) := \int_{-\infty}^{\infty} \mathbb{P}(x) \exp \left( \vartheta \log \frac{Q(x)}{P(x)} \right) dx
\]

(12)

\[
= \int_{-\infty}^{\infty} (\mathbb{P}(x))^{1-\vartheta} (Q(x))^{\vartheta} dx.
\]

It is now a matter of elementary calculus to check that

\[
M(\vartheta) = \int_{-\infty}^{\infty} \exp \left( -\vartheta \left( \frac{x+\mu}{\sigma} \right)^2 - \frac{1}{2} \vartheta (\frac{\sigma^2}{\sigma^2}) \right) \frac{dx}{\sqrt{2\pi \sigma^2}},
\]

so that

\[
\log M(\vartheta) = -\frac{\vartheta(1-\vartheta)(\mu^2 + \sigma^2) + \vartheta \log \eta + (1-\vartheta) \log \sigma}{\frac{1}{2} \log \left( \frac{\vartheta}{\eta} + \frac{1-\vartheta}{\sigma^2} \right)}
\]

(12)

It is possible (using standard computer algebra software) to explicitly find the value \(\vartheta^*\), optimizing \(I(\vartheta) = \vartheta^*(\vartheta) u - \log M(\vartheta^*(u))\), but the formula is lengthy and omitted.

The function \(\varphi(\cdot)\) then follows from

\[
\delta \left( \frac{\varphi(1-\delta)}{\delta} \right) = \chi^*,
\]

(13)

where \(\chi^* = -\log \chi/n\); here \(\chi\) is a measure for the likelihood of false alarms (for instance 0.05). We are now in a position to (numerically) solve Eq. (13), so as to obtain...
the threshold function \( \varphi(\cdot) \), as well as the value of the test statistic \( (11) \).

6.2. Analysis with synthetic data

It is clear that applying the above test to the (approximately Normally distributed) residuals that we generated in Section 4.1, allows us to detect whether or not a given pattern shows substantially higher values than those suggested by the trend. Alternatively, one may be interested in tests that detect whether or not the offered load is getting close to the system’s capacity. Below we point out how to set up a test that focuses on anomalies of the latter kind; a procedure to detect anomalies of the former kind can be set up similarly.

Due to the normalization procedure (2), in our case the parameters in model \( P \) are equal to \((\mu, \sigma) = (0.1)\) while these related to the model \( Q \) will typically be provided by the system administrator (who determines a suitable value based on e.g. the acceptable overload probability level). An example of such calculations, based on the Erlang model, is presented in [3], Section 5, Example 1. The figures provided there were the following: an expected number of users during ‘busy hour’ was equal to 320 and a number of users which, if reached, was considered to be overload was equal to 375 (that was the value of the parameter one tested against if considering a counterpart of the model \( Q \) presented here). For the finite capacity system, with these numbers a blocking probability would be around 0.1%. If we then use these figures to calculate the parameters of the model \( Q \) presented here, we get due to Eq. (2),

\[
v = \frac{375 - 320}{\sqrt{320}} \approx 3.075, \quad \eta = \sqrt{\frac{375}{320}} \approx 1.083 \quad (14)
\]

In our situation, where the expected number of calls is not constant but fluctuates (cf. Fig. 3(b)), to keep a (roughly) constant blocking probability, one would have to constantly update the values of \((v, \eta)\) to be tested against. While this is entirely possible, from a practical standpoint it may be more attractive to stay with just one fixed pair. For the number of calls larger than, say, 100 one would not observe much change in blocking probability despite changing the parameters values, while for lower numbers of calls the blocking probability will be somewhat underestimated.

All the changepoint detection tests presented below use the parameters \((\mu, \sigma) = (0.1)\) (model \( P \)) and \((v, \eta) = (3.075, 1.083)\) (model \( Q \)) unless explicitly stated otherwise. However, we want to underline that the algorithm for changepoint detection in Normally distributed data we provide here is generic, in the sense that any values \((\mu, \sigma)\) and \((v, \eta)\) can be used. Moreover, we may abstract from our VoIP situation, and use the proposed method for any type of data obeying the Gaussianity assumption.

6.2.1. Synthetic independent data

Here we will present the results of an experiment \((E_1)\) in which we draw 200 independent samples from the distribution \( P \), followed by another 200 samples from the distribution \( Q \). We take a window of length 50 samples, that is, we test whether \( H_0 \) should be rejected based on data points \( X_i, \ldots, X_{i+49}, \) for \( i = 1 \) up to 351. The first window in which the influence of the parameters \((v, \eta)\) is noticeable is therefore window number 152. 5000 independent repetitions of this experiment were run to assess the performance of the method. In Fig. 8(a) we see that before the changepoint (red vertical line) the detection ratio, which is in this case an estimate of the false alarm ratio (type I error probability), is about the assumed level of 5%. In addition, when only one observation from the distribution \( Q \) is present (window number 152 as explained above), the detection ratio (which now estimates the power of the test) is about 75%. The detection ratio then increases sharply as more samples from the new distribution appear. Furthermore, the position of the changepoint returned by the test is in general very close to the true one, as indicated on Fig. 8(b). One has to bear in mind, however, that for a less pronounced change the detection ratio would grow slower.

6.2.2. Synthetic dependent data

As it was stated earlier, the considered changepoint detection test is designed for i.i.d. data. Nevertheless, in practice one may be interested in the performance of the proposed algorithm in the case of dependent data. Below we present the results of an experiment \((E_2)\), which is similar to \( E_1 \), but now the observations before and after the changepoint originate from AR(1) processes (autoregressive processes, see for example [20]) with different levels of correlation. The AR(1) process is defined as

\[
X_i - \mu = \phi(X_{i-1} - \mu) + \epsilon_i,
\]

where \( \{\epsilon_i\} \) is a sequence of i.i.d. random variables with zero mean and variance \( \tau^2 \). The process has mean \( \mu \), and autocorrelation function

\[
\gamma(k) = \phi^k, \quad \text{for } k = 0, 1, \ldots
\]

Note, that because of the relationship \( \text{var} X_i = \tau^2 / (1 - \phi^2) = \sigma^2 \) the value of \( \tau \) used to generate the sample after the change will be equal to \( \tau = 1.083 \sqrt{1 - \phi^2} \) (not just 1.083). The findings related to \( E_2 \), for \( \phi \in (0.2, 0.8) \), are presented in Fig. 9 and show how the performance of our method is degraded by increasing the level of autocorrelation. In \( E_3 \) we consider a larger set of autocorrelations: \( \phi \in (0.2, 0.4, 0.6, 0.8) \); the results are given in Table 4.

The detection ratios in \( E_2 \) should be interpreted with care: in case of non-negligible correlation, the relative frequency of detections before the actual change happens is not anymore an unbiased estimator of type-I error probability. This is because if the test incorrectly detected a changepoint using data contained in the \( k \)th window, in the next step window \( k + 1 \) contains the same data apart from the oldest observation (which is dropped) and the appended newest observation which is now highly dependent on several previous ones. As a result, the chance of spurious detection is increased and at the same time, also the power of the test (in terms of the detection ratio) is affected. To assess this effect, we performed an additional experiment \((E_4)\), that is the same as \( E_3 \), apart from the fact that samples of length 50 are generated. In other
words, the size of these samples equals the detection window, thus completely 'regenerating' the input data for each of the 5000 repetitions.

For the somewhat lower value of $\phi = 0.2$ (Fig. 9) the spurious detection ratio is about 10% (versus the prescribed 5%, which is achieved in the case of the ‘regenerated’ samples of $E_4$), while the position of the detected change-point is close to the true one. In other words, for these low values of $\phi$ the detection procedure performs well.

When increasing $\phi$ the performance degrades: the false alarm ratio increases, while also the position of the detected change-point becomes less accurate. If the false alarm ratio is regarded to be too high, a quick fix is to lower the nominal value of $\alpha$. Obviously, again the price we pay lies in the detection ratio. $E_3$ and $E_4$ were redone with $\alpha = 0.005$; see the lower part of Table 4. One can see that for example for $\phi = 0.6$ the false alarm ratio is now close to the prescribed value. This indicates that by
6.3. Comparison with a threshold-based algorithm

Changepoint detection state-of-the-art includes a great variety of algorithms to solve the problem of timely detecting a changepoint with high accuracy, ranging from the simple threshold-based approaches, which provide a low accuracy but are simple, manageable and easy to understand and implement, to more sophisticated algorithms tailored to specific changepoint detection problems, which offer higher accuracy at the expense of larger complexity and narrower applicability. However, when a new changepoint detection algorithm is proposed, it is necessary to compare it against other algorithms in the state-of-the-art to assess its performance.

The class of CUSUM-based changepoint detection algorithms (to which the one presented in this paper belongs) are known to theoretically outperform threshold-based ones. We have applied a threshold-based algorithm to our first experiment ($E_1$) and compared the results, so as to empirically demonstrate this fact. We recall the main information of $E_1$: 200 samples of model $P$ followed by another 200 samples of model $Q$, a detection window of 50 samples, which means that the first window containing the changepoint is window number 152.

Varying the value of the threshold, we influence two key performance metrics of the algorithm: the false alarm ratio and the detection instant. In Fig. 10 we show two cases that illustrate this behavior. If we set the threshold to a large value (Fig. 10(a)), we have no false alarms, however, the detection instant is far from the first detectable instant. As we reduce the value for the threshold, the detection instant approaches the first detectable instant, but at the expense of a larger false alarm ratio. The example of Fig. 10(b) shows a case where the false alarm ratio is similar to the false alarm ratio of our proposed algorithm. However, if we compare Fig. 10(b) with Fig. 8(a), we observe that the threshold based algorithm needs more samples from the second model in the detection window in order to be able to detect the changepoint. Trying to enhance the detection accuracy would lead to high values of the false alarm ratio, which impedes its applicability in live applications.

6.4. Results on a real data trace

In this section we present results obtained by applying our anomaly detection method to a real data trace. To demonstrate its performance, we select one day (the Friday of Week 11) from our repository and give detailed comments about the outcome of the tests. Fig. 11 should be interpreted as follows:

- On the left scale we record the actual and average number of calls (a pattern) based on observations from five previous weeks, as indicated in Section 4.1.
- On the right scale we have the relative position of the anomaly detected in the window of 50 samples which ends at the given time point (meaning that we have
decided to skip the first 49 values as they would require readings from a previous day). A value of 1 means ‘no anomaly detected’, while for example a value of 0.94 observed at time point \( t = 926 \) means that at that moment the system reports an anomaly (detection instant), and declares it has happened 3 observations before \( (t = 923) \) (reported instant, as for a detection window of size 50 distance between two consecutive observations is 0.02 and \( 1 - 0.94 = 0.06 = 3 \cdot 0.02 \)). Later on, we again observe a series of readings with no alarm reported. Then, at the onset of the ‘afternoon peak’ \( (t = 955) \), after some uncertainty at the beginning, we observe a consistent period that the detector reports that the number of calls is significantly higher than average, which is confirmed by a visual inspection. From \( t = 1032 \) on, the system again declares no anomaly.

- Finally, for the sake of comparison we have added results from applying the threshold-based algorithm to the residuals of our methodology. The thresholds were set to \( 2\sigma \) and \( 3\sigma \) and we have depicted the alert generation instants in Fig. 11 with the symbols (\( \circ \)) and (\( \square \)), respectively. In the case of \( 2\sigma \) we can observe that there are reported anomalies at time instants were the actual calls do not deviate from the pattern (between 850 and 900) and afterwards, many anomalies are reported, but as there is not reported time instant with this algorithm, we cannot say whether all these alerts refer to the same anomaly or not. On the other hand, with the threshold set to \( 3\sigma \), we are too much conservative, and only there is one alert. In this situation, we may observe a non-negligible delay between the beginning of the anomalous period, and the moment of the generation of the alert. This is an undesirable property of an online detector, as timely reporting is crucial for solving the problem before many users are affected.

Observe, that the proposed method is capable of not only detecting an overload, defined as a situation when the system approaches its capacity limits, but also the situation that the number of calls, while still being below the aforementioned capacity limits, grows (falls) faster (slower) than the trend (see also the remark at the start of Section 6.2). Such information can be useful for example in call centers, as it may indicate the need for more staff than was initially planned. Note as well that when we have a slope of \(-0.02\) in the left scale graph, this means that we are reporting the same anomaly, i.e., we have the same reported instant but in different windows, because the sliding of the window is compensated by the increment of the distance to the reported instant. In case an action is taken to solve the anomaly, for instance, adding staff in the call center, the algorithm may be reset by substituting the values affected by the taken action with the ones in the pattern, thus avoiding to repeat the report of the anomaly that we know has been solved.

7. Concluding remarks

We have discussed the problem of anomaly detection in the situation that a strong trend (diurnal pattern being a result of human behavior) is present in the analyzed sample. As such a trend is a kind of non-stationarity by itself, its presence in most cases has a negative effect on the performance of any changepoint detection algorithm. Thus, the anomaly detection method we proposed in our paper consists of two steps: trend estimation and removal, which results in obtaining so-called residuals, and then applying a changepoint detection method to those residuals. Our contribution to the first step is verifying and exploiting the fact that the arrival process is non-homogeneous Poisson, which leads to a straightforward, yet effective trend removal procedure. The contribution to the second phase is twofold: a methodology to simultaneously detect changes in mean and variance, and extensive tests for the cases of both independent and correlated input. Besides, we also present and discuss a measurement procedure that leads to potentially significant correlation reduction. Finally, a real data example is included showing how the system could be implemented in practice.

As future work, we plan to further analyze our proposed alternative measurement procedure, including the analysis of more traces from different operators. Furthermore, we plan to extend our detrending methodology to work with multi-service measurements, where the Poissonian assumption typically does not hold.

Acknowledgement

The research of the first author was partially supported by the Spanish Ministerio de Ciencia e Innovación (MICINN) under the FPU fellowship program and the COST Action IC0703 Data Traffic Monitoring and Analysis. The research of the second author was partially supported by the Polish Ministry of Science and Higher Education.

Appendix A. Covariance of a Poisson process

Let \( K_t \) be the number of calls in a stationary M/G/\( \infty \) queue at time \( t \). In this appendix we evaluate \( \text{Cov}(K_\theta, K_\tau) \), for \( \theta \geq 0 \). For any \( t \), the random variable \( K_t \) has a poisson distribution with parameter \( \rho \). In addition,
any call present at time 0 has a residual duration \( S_r \) with distribution function
\[
P(S_r > y) = \frac{1}{E[S]} \int_{y}^{\infty} P(S > \tau) d\tau.
\]

Evidently,
\[
E[K_0 K_t] = \sum_{k=0}^{\infty} k E[K_t \mid K_0 = k] P(K_0 = k)
= \sum_{k=0}^{\infty} E[K_t \mid K_0 = k] k e^{-q} \frac{p^k}{k!}.
\]

Let \( K_t \) be the calls arriving in \((0, t]\) that are still present at time \( t \), and \( K_t \) the number of call present at time \( t \) that were already present at time 0. It is clear that
\[
E[K_t \mid K_0 = k] = E[K_t] = E[K_t] + E[K_t].
\]

Observe that, conditional on \( K_0 = k, K_t \) has a binomial distribution with parameters \( k \) and \( P(S_r > t) \), so that
\[
E[K_t \mid K_0 = k] = k P(S_r > t).
\]

The number of arrivals in \((0, t]\) has a Poisson distribution with mean \( 
\lambda t \). Given there are, say, \( \ell \) arrivals in that interval, each of them arrives at a uniformly distributed position in the interval \((0, t]\). As a result, the probability that an arbitrary call arriving in \((0, t]\) is still present at time \( t \) is
\[
p_t := \int_{0}^{t} \frac{1}{t} P(S > t - s) ds = \int_{0}^{t} \frac{1}{t} P(S > s) ds
= \frac{E[S]}{t} P(S_r < t).
\]

Conclude that \( K_t \) has a Poisson distribution with mean \( \lambda t p_t = \lambda E[S] P(S_r < t) \). Combining the above, we obtain
\[
E[K_0 K_t] = \sum_{k=0}^{\infty} (k P(S_r > t) + \rho P(S_r < t)) k e^{-q} \frac{p^k}{k!}
= P(S_r > t) \sum_{k=0}^{\infty} k e^{-q} \frac{p^k}{k!} + \rho P(S_r < t) \sum_{k=0}^{\infty} k e^{-q} \frac{p^k}{k!}
= P(S_r > t) (\rho + \rho^2) + \rho^2 P(S_r < t) = \rho^2 + \rho P(S_r > t).
\]

Conclude that
\[
\text{Cov}(K_0, K_t) = E[K_0 K_t] - E[K_0] E[K_t] = \rho^2 + \rho P(S_r > t) - \rho^2
= \rho P(S_r > t).
\]

References


Michel Mandjes received M.Sc. (in both mathematics and econometrics) and Ph.D. degrees from the Vrije Universiteit (VU), Amsterdam, the Netherlands. After having worked as a member of technical staff at KPN Research (Leidschendam, the Netherlands) and Bell Laboratories/Lucent Technologies (Murray Hill NJ, USA), as a full professor at the University of Twente, and as department head at CWI, Amsterdam, he currently holds a full professorship (chair of Applied Probability) at the University of Amsterdam, the Netherlands. He is also affiliated (as an advisor) with EURANDOM, Eindhoven, the Netherlands. In 2008, Mandjes spent a sabbatical at Stanford. His research interests include performance analysis of communication networks, queueing theory, advanced simulation methods, Gaussian traffic models, traffic management and control, and pricing in multi-service networks. He is the author of Large Deviations for Gaussian Queues, Wiley, 2007. He is associate editor of Stochastic Systems, Stochastic Models, Queueing Systems, and Journal of Applied Probability.

Marco Mellia (SM’08) received his Ph.D. degree in Telecommunications Engineering in 2001 from Politecnico di Torino. In 1999, he was with the CS Department at Carnegie Mellon University, Pittsburgh (PA) and since April 2001 he is with EE Department of Politecnico di Torino. He has co-authored over 140 papers published in international journals and conferences, and he participated in the program committees of several conferences including IEEE Infocom and ACM Sigcomm. His research interests are in the fields of traffic measurement, P2P applications and energy aware network design.