Gerhard MÜNZ*, Georg CARLE*

APPLICATION OF FORECASTING TECHNIQUES AND CONTROL CHARTS FOR TRAFFIC ANOMALY DETECTION

In this paper, we evaluate the capability to detect traffic anomalies with Shewhart, CUSUM, and EWMA control charts. In order to cope with seasonal variation and serial correlation, control charts are not applied to traffic measurement time-series directly, but to the prediction errors of exponential smoothing and Holt-Winters forecasting. The evaluation relies on flow data collected in an ISP backbone network and shows that good detection results can be achieved with an appropriate choice and parametrization of the forecasting method and the control chart. On the other hand, the relevance of the detected anomalies for the network operator mainly depends on the monitored metrics and the selected parts of traffic.

1. INTRODUCTION

In control engineering, monitoring mechanisms are deployed to observe the properties or behavior of a system and raise an alarm if an important parameter runs out of the range of sound operation. One monitoring goal is the detection of unexpected changes in characteristic properties of the system because such changes may be indications of failures, malfunctions and wearout. The detection must be fast to enable manual intervention, recalibration, or exchange of erroneous system components before severe consequences happen. Network monitoring pursues similar objectives. One aspect is the identification of anomalous traffic behavior which can be a sign of network failures or abuses, for example due to worm or attack traffic.

Change detection methods consider time-series of measurement values and search for points in time at which statistical properties of the measurements change abruptly, i.e. “instantaneously or at least very fast with respect to the sampling period of the measurements” [2]. Before and after the change, the monitored statistical properties are assumed to show no or only little variation. Under these conditions, even small changes can be detected with high probability if they persist for a long duration.

In general, the more a priori knowledge is available, the easier it is to detect changes with high accuracy. For example, parametric methods have more power than non-parametric methods, which means that they allow us to detect more true anomalies at the same false alarm level (i.e., probability of an alarm in absence of any significant change). However, if the model assumption is incorrect,

*Network Architectures and Services – Institute for Informatics, Technische Universität München, Germany,
e-mail: {muenz|carle}@net.in.tum.de

parametric methods lose their decisive power and may lead to wrong decisions. In the case of traffic anomaly detection, we cannot assume that the monitored variables follow a specific distribution, thus the detection should be non-parametric, or at least robust against non-normality. Moreover, changes should be detected very quickly (i.e., online) without requiring any a priori knowledge about their magnitude since such information is usually not available, either.

A practical solution to statistical online change detection are control charts [16]. In a control chart, mean and variability of a monitored variable are characterized by a centerline (CL), an upper control limit (UCL), and a lower control limit (LCL). A change is detected if the measured value exceeds one of the control limits. This decision can be formalized as a hypothesis test with null hypothesis H_0 assuming no significant change and alternative hypothesis H_1 suggesting the opposite.

In this paper, we evaluate the applicability of control charts to the problem of traffic anomaly detection. More precisely, we analyze time-series of byte, packet, and flow counts which can be easily obtained from routers via SNMP (Simple Network Management Protocol), IPFIX [8], or Cisco NetFlow [9]. These measurement time-series are subject to systematic changes, in particular seasonal variation. In addition, dependencies may exist between subsequent observations, noticeable as serial correlation. Both, systematic changes as well as serial correlation, need to be accounted for because most control charts are designed for independent and identically distributed observations. Useful tools are forecasting techniques which predict future values based on what has been observed in the past. In the optimal case, the prediction errors are small, random, and uncorrelated as long as the behavior of the monitored variable does not change. Hence, we can identify changes in the original variable by applying control charts to the prediction errors.

The main contribution of this work consists of a comparison of three different control charts and two different forecasting techniques. The considered control charts are the Shewhart control chart, the CUSUM (cumulative sum) control chart, and the EWMA (exponentially weighted moving average) control chart, which are commonly used in process monitoring. With respect to forecasting, we choose exponential smoothing and Holt-Winters forecasting, two self-adaptive and robust forecasting techniques which are suitable for our purposes. In contrast to many existing publications, our evaluation is not based on synthetically generated anomalies. Instead, we apply our methods to real flow data collected in the backbone network of an Internet Service Provider (ISP) and assess the relevance of the detected anomalies by examining their causes. The evaluation shows that a combination of Shewhart control chart and exponential smoothing enables good detection results under various conditions.

Sections 2 and 3 provide the theoretical background of the deployed control charts and forecasting techniques. Subsequently, in Section 4, we evaluate the capability to detect traffic anomalies with the described methods. Section 5 surveys related approaches of using control charts for traffic anomaly detection before Section 6 concludes this paper.

2. CONTROL CHARTS

A typical control charts contains a center line (CL) representing the average value of the monitored random variable Y under normal conditions. Above and below the center line, the upper and lower control limit (UCL, LCL) define the range of normal variation or in-control state. The decision function detects a change (or out-of-control state) if the measured value y lies outside this range.

The statistical properties of control charts can be deduced from the theory of sequential probability ratio tests (SPRT). If the distribution before and after the change are known, the decision

function can be converted into a condition for the log-likelihood ratio of the observation y:

$$s(y) = \log \frac{p_{\Theta_1}(y)}{p_{\Theta_0}(y)}$$

where $p_\Theta(y)$ is the probability density function of Y with parameter Θ. Θ_0 and Θ_1 are the parameter values before and after the change. If $s(y)$ is positive, the monitored random variable more likely conforms to the distribution after change than to the distribution before change. Hence, we can define a threshold h for $s(y)$ to reject the null hypothesis $H_0 : \Theta = \Theta_0$ and accept the alternative hypothesis $H_1 : \Theta = \Theta_1$ at a given level of significance. The level of significance corresponds to the probability of a false alarm.

If Y is normally distributed with constant variance σ^2 and means μ_0 and μ_1 before and after change, $s(y)$ becomes:

$$s(y) = \frac{\mu_1 - \mu_0}{\sigma^2} \left(y - \frac{\mu_1 + \mu_0}{2} \right)$$

If $\mu_1 > \mu_0$, $s(y) > h$ is equivalent to the decision function:

$$y > \mu_0 + L\sigma \quad \text{with } L = \frac{h\sigma}{\mu_1 - \mu_0} + \frac{\mu_1 - \mu_0}{2\sigma}$$

In this equation, the correspondence to the control chart is obvious: μ_0 is the center line and $\mu_0 + L\sigma$ the upper control limit.

As the variance of a single observation is quite high, change detection methods often consider a sequence of observations $\{y_t | t = a, \ldots, b\}$ to increase the power of the hypothesis test. Under the condition that the observations are independent, the log-likelihood ratio of the sequence is:

$$s(y_a, \ldots, y_b) = \log \frac{\prod_{t=a}^{b} p_{\Theta_1}(y_t)}{\prod_{t=a}^{b} p_{\Theta_0}(y_t)} = \sum_{t=a}^{b} s(y_t)$$

A hypothesis test based on $s(y_a, \ldots, y_b)$ corresponds to a control chart for a test statistic that is calculated from $y_a, \ldots y_b$. An example is the average value $\bar{y}$, which has an important property: regardless of the distribution of Y, $\bar{y}$ is approximately normally distributed if calculated from a large number of observations, thanks to the central limit theorem.

In the following subsections, we introduce three different control charts, namely the Shewhart control chart, the CUSUM control chart, and the EWMA control chart. For more detailed information, we refer to the text books of Montgomery [16] and Basseville [2].

2.1. SHEWHART CONTROL CHART

The Shewhart control chart [26] defines UCL, CL, and LCL for a statistic calculated from N observations $y_{(l-1)N+1}, \ldots, y_{lN}$. An example statistic is the average value $\bar{y}_l$, which is appropriate for detecting changes in the mean:

$$\bar{y}_l = \frac{1}{N} \sum_{t=(l-1)N+1}^{lN} y_t \quad \text{where} \quad l = 1, 2, \ldots$$

If the observations are independent and identically distributed with mean μ_0 and variance σ^2, $\bar{y}_l$ is an unbiased estimator of μ_0 with variance σ^2/N. Hence, the upper and lower control limits can be

defined in the form $\mu_0 \pm L\sigma/\sqrt{N}$ with tuning parameter L. An alarm is raised if $\bar{y}_l$ passes one of the control limits. As already mentioned, $\bar{y}_l$ is approximately normally distributed for large N, thus the control limits for a given false alarm probability α are $\mu_0 \pm \Phi(1-\alpha/2)\sigma/\sqrt{N}$. However, this approximation does not hold for small N or if the observations are serially correlated.

A special case is $N = 1$, the so-called Shewhart control chart of individuals. This chart compares individual observations against the control limits. Obviously, the central limit theorem does not apply, thus the distribution of Y needs to be known exactly in order to define precise limits for a given false alarm probability.

2.2. CUSUM CONTROL CHART

The CUSUM control chart (also called CUSUM algorithm) [18] is based on the fact that $S_t = s(y_1, \ldots, y_t)$ has a negative drift under normal conditions and a positive drift after a change. The CUSUM decision function g_t compares the increase of S_t with respect to its minimum to a threshold h:

$$g_t = S_t - \min_{1 \le i \le t} S_i = \max\left(0, s(y_t) + g_{t-1}\right) = \left[g_{t-1} + s(y_t)\right]^+ \ge h \quad ; \; g_0 = 0$$

An alarm is raised if g_t exceeds the threshold h. To restart the algorithm, g_t must be reset to zero.

From the view of hypothesis testing, the CUSUM control chart repeatedly performs an SPRT where each decision considers as many consecutive observations as needed to accept either H_0 or H_1. The CUSUM control chart implicitly starts a new run of SPRT if H_0 has been accepted, and stops with an alarm in the case of H_1. The threshold h allows trading off the mean detection delay and the mean time between false alarms. If the distribution of Y is unknown, the log-likelihood ratio $s(y_t)$ must be replaced by a statistic $u(y_t)$ with comparable properties: the expectation value of $u(y)$ must be negative under H_0 and positive under H_1 This variant is often called non-parametric CUSUM algorithm.

An appropriate statistic for detecting positive shifts in the mean is $u^+(y) = y - (\mu_0 + K)$. K is called reference value. In order to detect negative shifts as well, we need a second statistic $u^-(y) = (\mu_0 - K) - y$. As a result, we get two decision functions:

$$g_t^+ = \left[g_{t-1}^+ + y_t - (\mu_0 + K)\right]^+ \ge h \quad ; \quad g_t^- = \left[g_{t-1}^- + (\mu_0 - K) - y_t\right]^+ \ge h$$

Typical settings are $K = \sigma/2$ and $h = 4\sigma$ or $h = 5\sigma$, where σ is the standard deviation of Y_t [16].

Compared to the Shewhart control chart, CUSUM detects small but persistent changes with higher probability because little effects accumulate over time. Brodsky and Darkhovsky [5] studied the properties of the non-parametric CUSUM algorithm for a specific family of exponential distributions of $u(y)$. For this distribution family, the detection delay reaches the theoretic minimum if the mean time between false alarms goes to infinity. As we will discuss in Section 5, several existing publications refer to this proof of optimality although the specific requirements are not fulfilled in general.

2.3. EWMA CONTROL CHART

The EWMA control chart (c.f. [23,24]) relies on exponential smoothing of observations. Given the smoothing constant λ ($0 < \lambda < 1$),

$$z_t = \lambda y_t + (1-\lambda) z_{t-1} = \lambda \sum_{i=0}^{t-1} (1-\lambda)^i y_{t-i} + (1-\lambda)^t z_0$$

is a weighted average of all observations up to time t. The initial value is the expected mean under H_0: $z_0 = \mu_0$. If the observations are independent and identically distributed with variance σ^2, the variance of z_t approaches $\frac{\lambda}{2-\lambda}\sigma^2$ for $t \to \infty$, which allows the definition of control limits for z_t:

$$\mu_0 \pm L\sigma\sqrt{\frac{\lambda}{2-\lambda}}$$

λ and L are design parameters of the EWMA control chart. Popular choices are $2.6 \leq L \leq 3$ and $0.05 < \lambda < 0.25$, where smaller λ allow detecting smaller shifts [16].

The EWMA control chart has some interesting properties [16]. It can be tuned to achieve approximately equivalent results as the CUSUM control chart. Secondly, it is quite robust against non-normal distributions of Y, especially for small values of λ (e.g., $\lambda = 0.05$). Finally, after adjusting the control limits, the EWMA control chart still performs well in the presence of low to moderate levels of serial correlation in Y_t.

3. RESIDUAL GENERATION BY FORECASTING

Common to all the control charts presented in Section 2 is the assumption that the observations are independent and identically distributed under normal conditions. This corresponds to the output of a stationary random process that generates uncorrelated values. Such a process is also called pure random process.

There are various reasons why traffic measurement data exposes significant deviation from the output of a pure random process. Non-stationarities result from trends as well as dependencies on the time of day, the day of the week etc. Serial correlation is caused by internal network states which cannot change arbitrarily from one instant in time to the next. For example, the number of packets in the network evolves according to a birth-death process depending on the arrival times and processing times.

We can identify systematic changes in the mean or variance by visually inspecting the measured values over time. Systematic changes as well as serial correlation also have an impact on the sample autocorrelation, which is calculated as follows:

$$r_\tau = \frac{\sum_{i=1}^{N-\tau}(y_t - \bar{y}_t)(y_{t+\tau} - \bar{y}_t)}{\sum_{i=1}^{N}(y_t - \bar{y}_t)^2}$$

In the above equation, N is the number of observations and τ the lag between two instances of time. If r_τ is not decreasing with increasing τ, or if it shows periodic oscillation, the observations do not resemble the output of a stationary random process. In the case of a pure random process, the 95% confidence interval of r_τ is $[-1/N - 2/\sqrt{N}; -1/N + 2/\sqrt{N}]$ for all τ. Hence, if a non-negligible number of r_τ's lie outside this range, the process is not purely random.

Time-series analysis allows modeling and removing systematic changes and serial correlation with help of the Box-Jenkins approach [4] . However, fitting an accurate ARIMA (autoregressive integrated moving average) model is difficult and requires a long series of anomaly-free observations. Therefore, robust forecasting methods based on exponential smoothing are preferred [7], especially for online applications. Forecasting relies on the assumption that the temporal behavior observed in past observations persists in the near future. Hence, an unusually large prediction error is an indicator of a change in the monitored random variable. The prediction errors are also called residuals because they represent the variability not explained by the forecasting model.

In the following subsections, we present two popular forecasting techniques that we will use in Section 4 for residual generation: exponential smoothing and Holt-Winters forecasting. In order to define appropriate limits for the control charts, we need to estimate the standard deviation of the residuals under normal conditions. How this can be achieved is explained in Section 3.3.

3.1. EXPONENTIAL SMOOTHING

Exponential smoothing allows predicting future values by a weighted sum of past observations:

$$\hat{y}_{t+1} = \alpha \sum_{i=2}^{t} (1-\alpha)^{t-i} y_i + (1-\alpha)^{t-1} y_1 = \alpha y_t + (1-\alpha)\hat{y}_t$$

This is the same exponentially weighted moving average as used in the EWMA control chart. The distribution of the weights is geometric and gives more weight to recent observations. Forecasting according to the above equation is optimal for an infinite-order MA (moving average) process, which is equivalent to an ARIMA(0,1,1) process [7]. Yet, exponential smoothing is very robust and also provides good forecasts for other trendless and non-seasonal time-series. The optimal value for α can be approximated by trying different values and choosing the one with the smallest residual sum of squares.

3.2. HOLT-WINTERS FORECASTING

Holt-Winters forecasting combines a baseline component L_i with a trend component T_i and a seasonal component I_t:

$$\hat{y}_{t+1} = L_t + T_t + I_t$$

L_t, T_t, and I_t are recursively updated according to the following equations:

$$L_t = \alpha(y_t - I_{t-s}) + (1-\alpha)(L_{t-1} + T_{t-1})$$

$$T_t = \beta(L_t - L_{t-1}) + (1-\beta)T_{t-1}$$

$$I_t = \gamma(y_t - L_t) + (1-\gamma)I_{t-s}$$

α, β, and γ are smoothing parameters which have to be set to appropriate values in the range $(0, 1)$. s is the length of one season counted in time intervals. The above equations include an additive seasonal component. Alternatively, the seasonal component can also be modeled in a multiplicative way. For more details, we refer to [7] and the references therein.

3.3. CONTROL LIMITS AND STANDARD DEVIATION ESTIMATORS

As we have seen in Section 2, control limits are usually defined relatively to the standard deviation σ of the monitored random variable. If we apply control charts to residual time-series of prediction errors $\epsilon_t = y_t - \hat{y}_t$, the standard deviation has to be estimated. We could calculate the sample variance from a finite set of past residuals. However, this estimation is very sensitive to outliers and does not reflect dynamic changes in the variance. Therefore, we make use of a moving estimator which is based on exponential smoothing. For a given mean μ, the exponentially weighted mean square error (EWMS) is a variance estimator:

$$\hat{\sigma}_t^2 = \rho(\epsilon_t - \mu)^2 + (1-\rho)\hat{\sigma}_{t-1}^2$$

Since the mean of the residuals is approximately zero under normal conditions, we can set $\mu = 0$ in the above equation.

4. EVALUATION

We evaluated the capability to detect traffic anomalies with help of the forecasting techniques and the control charts presented in the previous sections. Our evaluation is based on traffic measurement data collected in the Gigabit backbone network of a regional ISP between September 7 and October 25, 2006. The operation area of the ISP covers parts of Saarland, Rhineland-Palatinate, Hesse (all federal states in Germany), Luxembourg, and Belgium. At measurement time, the offered services ranged from server hosting and colocation to VPNs and modem, ISDN, and DSL dial-in service. Customers were corporate clients, local carriers, roaming providers, and small and medium enterprises. The measurements were performed at a router using unsampled Cisco Netflow.v5 with active and idle flow timeouts set to 150 seconds. The router exported the resulting flow records to a collector which stored them in a database after anonymizing the IP addresses.

Our evaluation is not based on individual flows but on time-series of the number of bytes, packets, and flows counted in equally spaced time intervals. Each flow record was associated with the time interval in which the first packet passed the router. The interval length was set to 300 seconds (i.e., twice the flow timeout) in order to reduce possible distortions in the byte and packet counts that may result from long-lasting high-volume flows which are reported at the period of the active timeout. The flow count was determined as the number of distinct IP-five-tuples (i.e., cardinality of combinations of protocol, source and destination IP addresses and port numbers) to prevent manifold counting of flows reported in more than one records per time-interval.

We implemented the forecasting techniques and control charts in GNU Octave [11]. This approach enabled us to analyze the time-series data with different forecasting methods and control charts. For online traffic analysis, the detection mechanisms can be integrated into a real-time system, for example as a detection module of our traffic analysis framework TOPAS [17]. We determined the reason for the detected anomalies by identifying the responsible flows. Furthermore, we assessed the importance and relevance of the alarms for the network operator.

In the following subsections, we present the results for time-series of overall IP traffic, ICMP traffic, and SMB (Server Message Block) traffic. The objective is to answer the following questions:

- Which forecasting method is the most appropriate to generate residual time-series?
- Which control chart provides the best detection results when applied to these residuals?
- In which part of the traffic and in which metric do we find the most interesting anomalies?

We do not aim at finding the optimal solution, which is difficult regarding the numerous degrees of freedom. Also, the result would be limited to the specific set of measurement data. Instead, we are interested in recommendations that allow achieving good results under various conditions.

4.1. ANOMALY DETECTION IN OVERALL IP TRAFFIC

Figure 1 depicts the time-series of the total number of bytes, packets, and flows in the measurement period. All three metrics show a daily cycle of low values at nighttime and high values at daytime. Furthermore, we observe a weekly cycle with higher traffic on weekdays and lower traffic at weekends. October 3 is a public holiday in Germany, which results in slightly decreased traffic

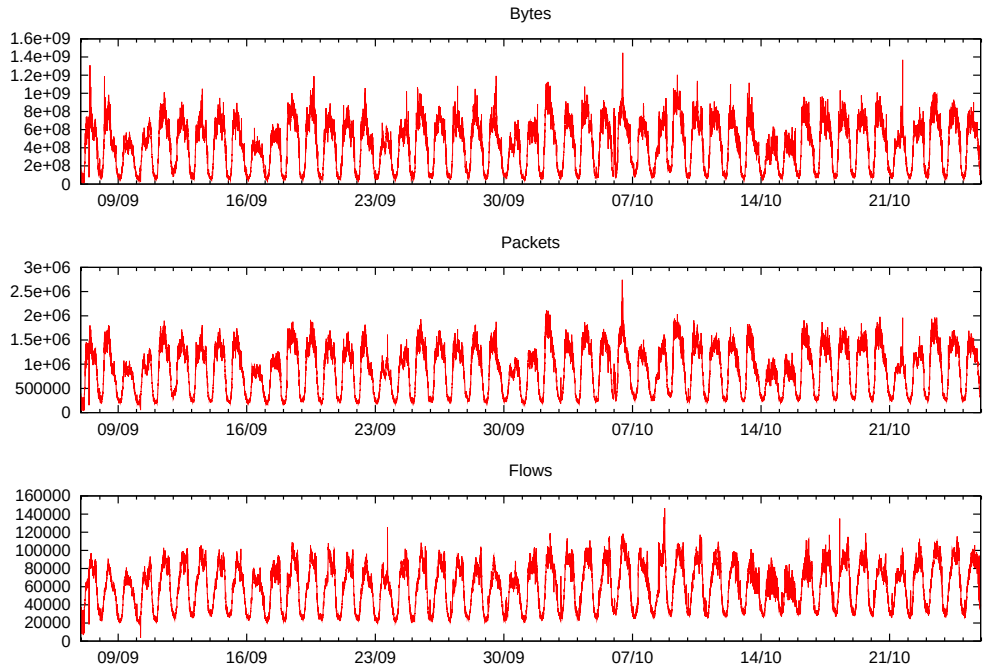

Fig. 1: Measurement time-series of total traffic

volume on this day as well. The regular run of the curves is interrupted by isolated peaks which are obvious traffic anomalies. Most of the time, a peak in one metric coincides with a peak in another metric. Yet, we rarely observe extreme values in all three metrics simultaneously.

In order to cope with the seasonal variation, we apply exponential smoothing and Holt-Winters forecasting and use the prediction errors as residual time-series. Given the measurement time-series y_t, we initialize the Holt-Winters components as follows:

$$L_s = 0 \quad ; \quad T_s = 0 \quad ; \quad I_i = y_i \text{ for } i = 1, \ldots, s$$

The seasonal period is set to $s = 288$ or $s = 2016$ to account for daily or weekly seasonality. With exponential smoothing, we obtain the first prediction value (and residual) in the second time interval ($t = 2$). In contrast, Holt-Winters forecasting requires the first s values for initialization, thus the first residual is generated at $t = s + 1$. To get comparable results, we only count the alarms raised after time interval $t = s$.

Figure 2 shows the residual time-series of byte counts for three different configurations of exponential smoothing ($\alpha = 1$, $\alpha = 0.5$, and $\alpha = 0.1$) and one setup of Holt-Winters forecasting with additive seasonal component ($s = 288$, $\alpha = 0.1$, $\beta = 0.001$, $\gamma = 0.25$). In the case of $\alpha = 1$, the residuals are simply the differences of consecutive measurement time-series values. Except for exponential smoothing with $\alpha = 0.1$, the seasonal variation of the mean is successfully removed. However, the variability of the residuals still depends on the time of day. Regarding the different settings for exponential smoothing, $\alpha = 0.5$ provides the best results: obvious anomalies in the original data appear as peaks, whereas the variability during normal traffic is relatively low. This visual impression is confirmed by the mean squared prediction error, which is the smallest for this setting.

Similar to our examinations of exponential smoothing, we tested various parameterizations of Holt-Winters forecasting with different smoothing constants, seasonal periods of one day and one week, additive and multiplicative seasonal components. The setting shown in Figure 2 effectively reduces the seasonal variation and exposes various anomalies in the measurement data. Yet, the

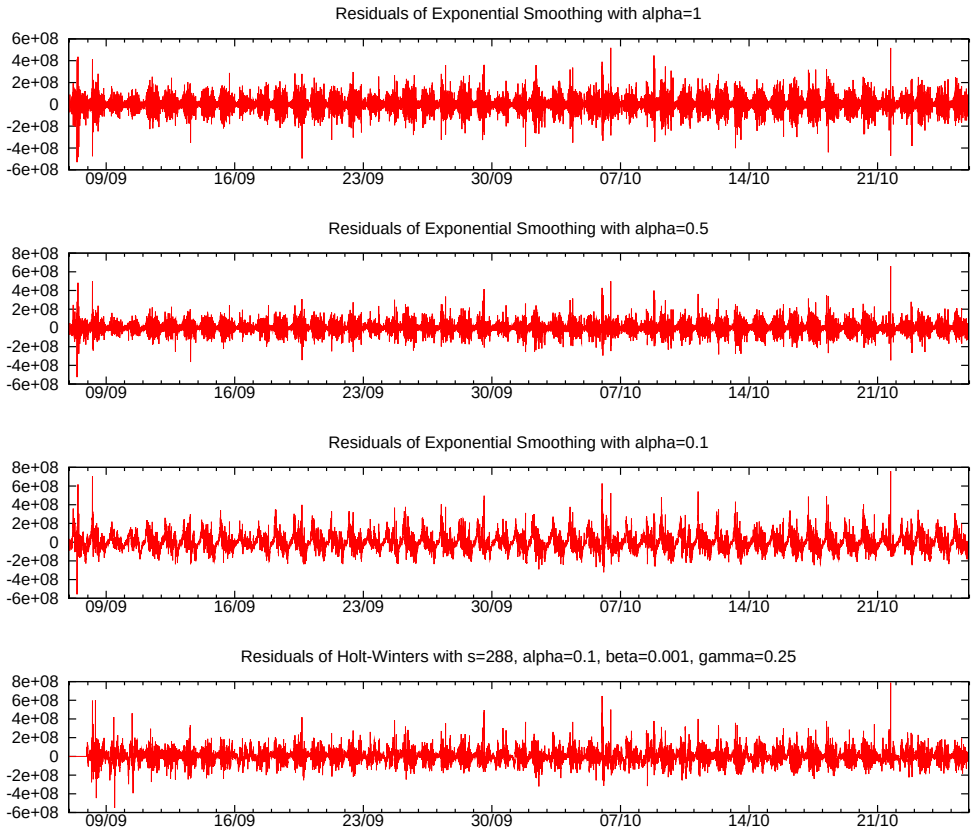

Fig. 2: Residual time-series (prediction errors) of byte counts

additional complexity of Holt-Winters forecasting does not seem to ensure better results than simple exponential smoothing: the residual time-series of the two methods turned out to be quite similar. A possible explanation is that the seasonal period is very long (288 or 2016 intervals), hence the effect of the seasonal variation on consecutive values is very small.

Figure 3 shows the sample autocorrelation of the original byte count time-series and the corresponding residuals. As expected, the seasonality of the original measurements reappears in the autocorrelation plot. On the other hand, the serial correlation in the residual time-series attenuates quite quickly.

We applied the Shewhart control chart of individuals, the two-sided CUSUM control chart, and the EWMA control chart to the residual time-series. Control limits were defined as multiples of $\hat{\sigma}$ which was estimated by EWMS (see Section 3.3). The smoothing constant ρ controls how quickly the limits adapt to variability changes in the prediction errors. For our purposes, $\rho = 0.01$ turned out to be a good setting.

Figure 4 shows the measurement time-series of byte counts on top and three control charts applied to the residuals of exponential smoothing below. The parameters of the control charts are as follows:

- Shewhart: $UCL = -LCL = 6\hat{\sigma}$
- CUSUM: $K = \hat{\sigma}$; $h = 6\hat{\sigma}$
- EWMA: $\lambda = 0.25$; $UCL = -LCL = 5\hat{\sigma}\sqrt{\frac{\lambda}{2-\lambda}}$

The Shewhart control chart shows the residual time-series and the corresponding control limits. The CUSUM control chart depicts the maximum of the two CUSUM statistics g_t^+ and g_t^- as well as the

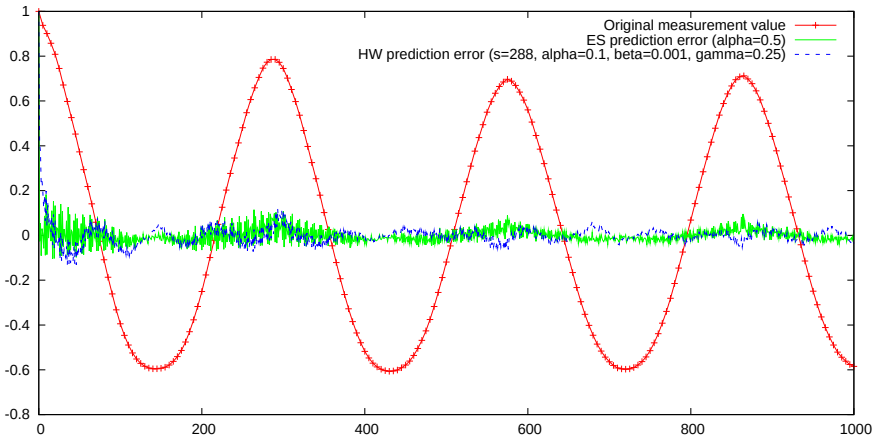

Fig. 3: Sample autocorrelation of byte counts

Table 1: Byte Anomalies Detected in All Control Charts

Day	Cause
08/09	FTP download (640 MBytes at 1 MByte/s on average)
24/09	RSF-1 data transfer (approx. 310 MBytes)
29/09	HTTP download (approx. 160 MBytes)
06/10	High SMTP traffic during 3 hours in the night
11/10	HTTP download (peak of 355 MBytes in one interval)
18/10	HTTP download (peak of 195 MBytes in one interval)
21/10	HTTP download (peak of 524 MBytes in one interval)

threshold h. The EWMA chart finally shows the exponentially smoothed residuals z_t and the control limits.

The dotted vertical lines in Figure 4 mark the intervals in which the corresponding value is beyond the control limits. We obtained 11, 15, and 11 alarms for Shewhart, CUSUM, and EWMA, respectively. Some of them are so close to each other that they can hardly be distinguished in the figure. Table 1 lists the set of anomalies that are detected in all three charts. For each anomaly, we identified the responsible flows and found that most of the alarms were caused by large downloads from web or file servers. What we describe as RSF-1 data transfer in the table is a large flow to UDP port 1195, which has been registered by High-Availability.Com [13] for a high-availability and cluster middleware application. Very probably, these downloads represent legitimate traffic. However, we detected anomalous high SMTP traffic on October 6 lasting for several hours, which is a sign of a mailbomb triggered by spammers or a worm propagating via e-mail. Most of the remaining alarms not mentioned in the table could be explained by the same kinds of HTTP, FTP, and RSF-1 traffic. Though, some of the alarms trigged by the CUSUM and EWMA control charts could not be associated to any unusual pattern in the flow records.

Some of the detected anomalies also appear as extreme values in the original measurement data. Hence, they could be detected with a threshold applied to the byte counts directly. Others, such as the mailbomb, do not cause extraordinarily high byte counts, i.e. they can only be detected in the residuals.

We applied the same control charts to the Holt-Winters residuals and obtained similar results as for exponential smoothing. Furthermore, we examined if more interesting anomalies can be found

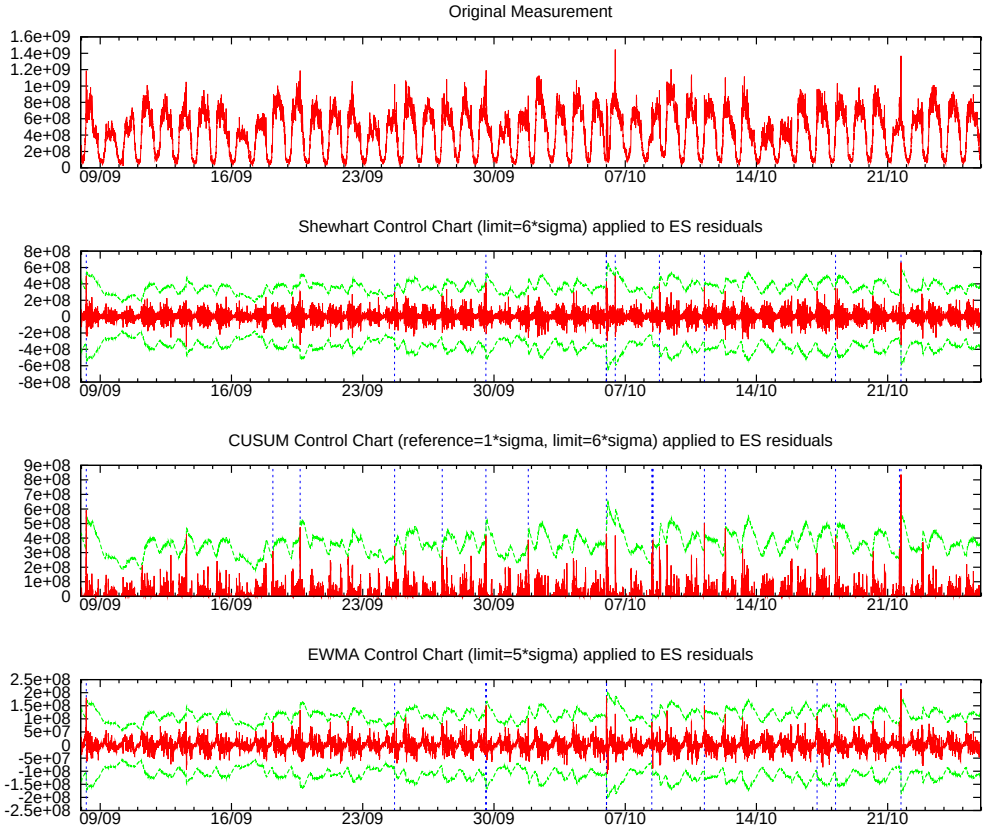

Fig. 4: Control charts applied to ES residuals ($\alpha = 0.5$) of byte counts

in the packet and flow counts or in any ratio of the three basic metrics, such as the average number of bytes per flow. Packet and byte counts triggered the same alarms. A couple of new anomalies were found in the flow counts. One of these alarms is the result of a large number of short SSH connections from one client to multiple servers, a pattern that may be caused by a massive password guessing attempt. Another alarm coincides with a time interval in which the traffic abruptly breaks down, possibly due to a network failure. Regarding the anomalies found in the ratios, we did not notice any improvements compared to the basic metrics of bytes, packets, and flows.

As a result, we conclude that residual generation using exponential smoothing techniques and change detection with the Shewhart control chart of individuals enables the detection of traffic anomalies with relatively low computational complexity. The CUSUM and EWMA control chart did not provide better detection results but raised additional alarms that could not be linked to anomalous traffic behavior. In the EWMA control chart, the moving average flattens short peaks in the residuals and thus hampers their detection. However, such peaks result from abrupt changes in the original measurement data, which are events we definitively want to detect.

An appropriate level of the control limits needs to be determined by experimentation in order to focus on the most significant anomalies. Among the detected anomalies in the overall traffic, the mail-bomb, the password guessing attempt, and the network failure are the most interesting events for the network operator. However, the majority of the alarms is caused by legitimate traffic, independently of the considered metric.

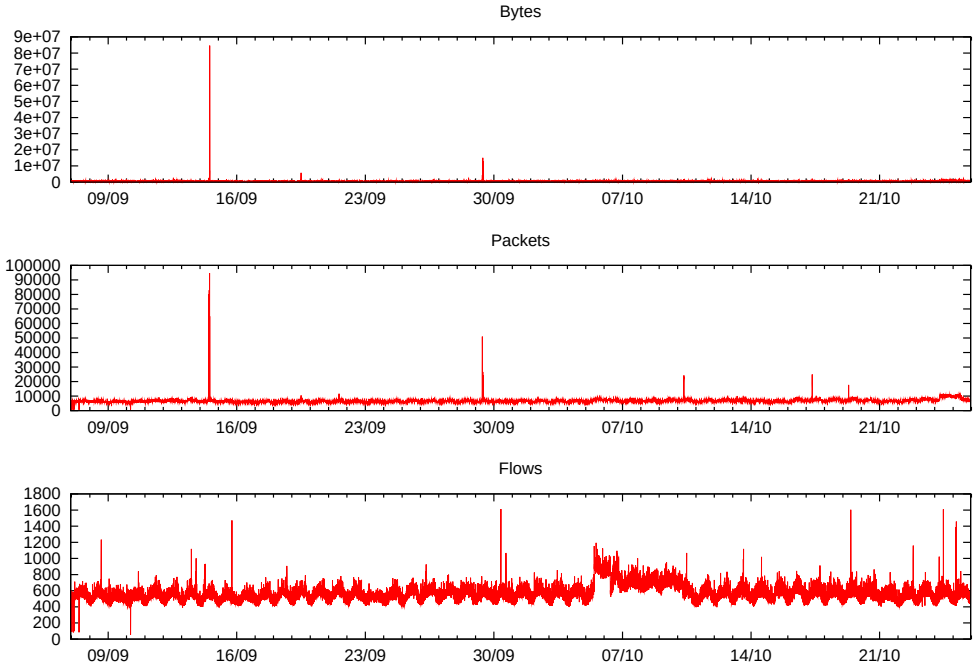

Fig. 5: Measurement time-series of ICMP traffic

4.2. ANALYZING ICMP TRAFFIC

As part of the Internet protocol suite, the Internet Control Message Protocol (ICMP) is mainly used for exchanging error messages, for example, if a certain host cannot be reached due to link or routing problems. ICMP is also used for network testing and debugging purposes (e.g., using ping and traceroute commands) and self configuration in local IP networks. As ICMP is not directly involved in the transport of user and application data, we expect a low and time-invariant level of ICMP traffic under normal conditions. Indeed, we can observe this behavior in the byte and packet time-series shown in Figure 5. In contrast, the number of flows shows daily variation, yet less pronounced than in the overall traffic.

We adopted the most promising approach from Section 4.1, namely the Shewhart control chart applied to the residuals of exponential smoothing ($\alpha = 0.5$), to detect anomalies in the ICMP traffic. Control limits at $\pm 6\hat{\sigma}$, as used before, generated a very large number of alarms for byte and packet counts. Therefore, we increased the control limits to $\pm 8\hat{\sigma}$ in order to focus on the most significant anomalies.

The anomalies found in the byte and packet time-series are listed in Table 2. Many alarms are triggered by both metrics, especially those caused by ping traffic (ICMP echo requests and replies). Sporadic occurrences of ping traffic at moderate rate are not suspicious, hence the corresponding alarms are not of much interest. The extremely high impulse on September 14 is the result of one host pinging another host at very high rate, which can be a sign of an attack. Though, as the ping is very short, we think that it was executed for debugging purposes. Apart from ping traffic, many traffic anomalies are caused by destination unreachable messages, most of them reporting that a large packet could not be fragmented due to the 'don't fragment' bit in the IP header. The corresponding ICMP messages are quite large because they include a section of the dropped packet. Therefore, these anomalies are mainly detected in the number of bytes.

As can be seen in Table 3, the flow count residuals exceed the control limits only six times.

Table 2: Shewhart Alarms for ICMP Traffic: Bytes and Packets

Time	Bytes alarm	Packets alarm	Cause
11/09 09:00	x		Echo replies from/to one host
12/09 15:15	x		Ping (moderate rate)
14/09 13:50	x	x	Ping flood (very high rate)
14/09 14:25	x		Ping flood (very high rate)
14/09 14:40	x		Ping flood (very high rate)
19/09 14:00	x	x	Echo replies from/to one host
21/09 15:30	x	x	Destination unreachable (fragmentation required)
24/09 08:25	x		Destination unreachable (fragmentation required)
29/09 10:40	x	x	Ping (moderate rate)
29/09 11:15	x		Ping (moderate rate)
29/09 11:30	x		Ping (moderate rate)
05/10 16:35	x		Destination unreachable (fragmentation required)
06/10 18:00	x		Destination unreachable (fragmentation required)
10/10 10:00	x	x	Time exceeded from one host
13/10 07:50		x	Destination port unreachable
16/10 16:45	x		Destination unreachable (fragmentation required)
17/10 09:55	x	x	Ping (moderate rate)
19/10 09:30	x	x	Ping (moderate rate)
21/10 23:00	x		Destination unreachable (fragmentation required)
25/10 15:35	x		Ping (moderate rate)

Table 3: Shewhart Alarms for ICMP Traffic: Flows

Time	Cause
13/09 14:45	Destination port unreachable from many sources to one host
15/09 19:40	ICMP scan followed by TCP connections to port 3389 (WBT)
30/09 11:05	ICMP scan followed by TCP connections to port 1433 (MS SQL)
19/10 12:20	ICMP scan followed by TCP connections to ports 3389 (WBT) and 1433 (MS SQL)
24/10 13:20	Ping at moderate rate to five hosts
25/10 05:40	ICMP scan followed by TCP connections to ports 80 and 3128 (HTTP proxies)

None of these alarms coincides with any of the byte and packet alarms. Examining the flow records, four of the alarms can be explained by ICMP echo requests sent by individual hosts to a few hundred IP addresses. Echo replies are returned from a small proportion of the scanned IP addresses only. To these destinations, the scanning host then tries to establish TCP connections on ports 3389, 1433, 80, or 3128 which are used by Microsoft remote desktop (Windows-based Terminal, WBT), Microsoft SQL server, and HTTP proxies, respectively. ICMP scans are typically performed with help of automated network scanners in order to detect active hosts. It is difficult to assess if the observed traffic is harmful or not. Maybe the scans served testing and debugging purposes. This assumption is fortified by our experience that malware and worms usually try to establish TCP connection directly without preceding ICMP scans. However, it has been recently reported that ICMP scans are more and more frequently deployed in advance of an infection attempt [10] as well.

Having a look at Figure 5, we see that the number of flows is increased between October 5 and October 10. After decreasing the Shewhart control limits to $\pm 5\hat{\sigma}$, this anomaly is also detected in the flow count residuals. We examined the flow records and discovered that the increase is caused

by ICMP destination unreachable messages sent from different sources to one specific host. Two different error codes are reported: host unreachable and communication administratively prohibited. The second one is returned by routers or firewalls if a packet is discarded because of a blocked destination IP address or port number. The host receiving all these messages thus had to be emitting a lot of packets to non-existing or blocked destinations. Indeed, we found a lot of outgoing connection requests from this IP address to TCP ports 445 (Microsoft-DS) and 139 (Netbios) during five days. On Microsoft Windows systems, these ports are well-known for many vulnerabilities which are exploited by worms.

All in all, anomalies found in the ICMP traffic give the network operator valuable insights in the current state of the network. An anomalous increase of the number of destination unreachable messages indicates a network failure or the occurrence of a TCP or UDP scan performed by worms or hackers. Large numbers of ICMP flows are mostly caused by ICMP scans which do not represent an instantaneous security threat but often reveal other suspicious traffic, such as connection attempts to specific TCP ports following a scan.

4.3. ANALYZING SMB TRAFFIC

Motivated by the findings in the ICMP traffic, we analyzed TCP traffic to and from port 445. Since Windows 2000, this port is used by Microsoft for file and printer sharing in local area networks via the SMB (Server Message Block) protocol. However, vulnerabilities in this service are also being exploited by worms to infect unprotected computers in the network. A prominent example is the Sasser Worm which has been spreading over the Internet since 2004.

For our analysis, we consider the difference of TCP traffic to and from port 445. The plots in Figure 6 show the corresponding time-series for the number of bytes, packets, and flows. As can be seen, all metrics have small values close to zero most of the time and do not show any seasonal variation. Between October 5 and 9, we observe longer periods of large positive values, which means that many more bytes, packets, and flows are directed to port 445 than returned. During this time, we also observed an increased number of ICMP destination unreachable messages. Indeed, the anomalies in the ICMP and SMB traffic are related to each other: the emitter of the SMB traffic is the receiver of the ICMP traffic. As mentioned in Section 4.2, the host is very probably infected by a worm trying to connect to randomly chosen destinations.

As before, we applied Shewhart control charts to the prediction errors of exponential smoothing. We set the control limits to $\pm 10\hat{\sigma}$ in order to get a reasonably small number of alarms. We obtained 23 alarms for the byte count residuals, 12 alarms for the packet count residuals, and 13 alarms for the flow count residuals. While many of the byte and packet alarms are caused by non-suspicious SMB traffic (e.g., data transfers between two hosts), all of the flow alarms are triggered by scanning activities. Four of the flow alarms are related to the worm-infected host already mentioned, the remaining alarms are caused by short scans originating from different IP addresses. These scans probably belong to worm traffic generated in distant networks, thus only parts of it are observed by the router.

4.4. DISCUSSION OF RESULTS

Our evaluation demonstrates the applicability of forecasting techniques and control charts for detecting traffic anomalies in time-series of byte, packet, and flow counts. The prediction error of exponential smoothing with smoothing constant $\alpha = 0.5$ turned out to be a robust residual generation

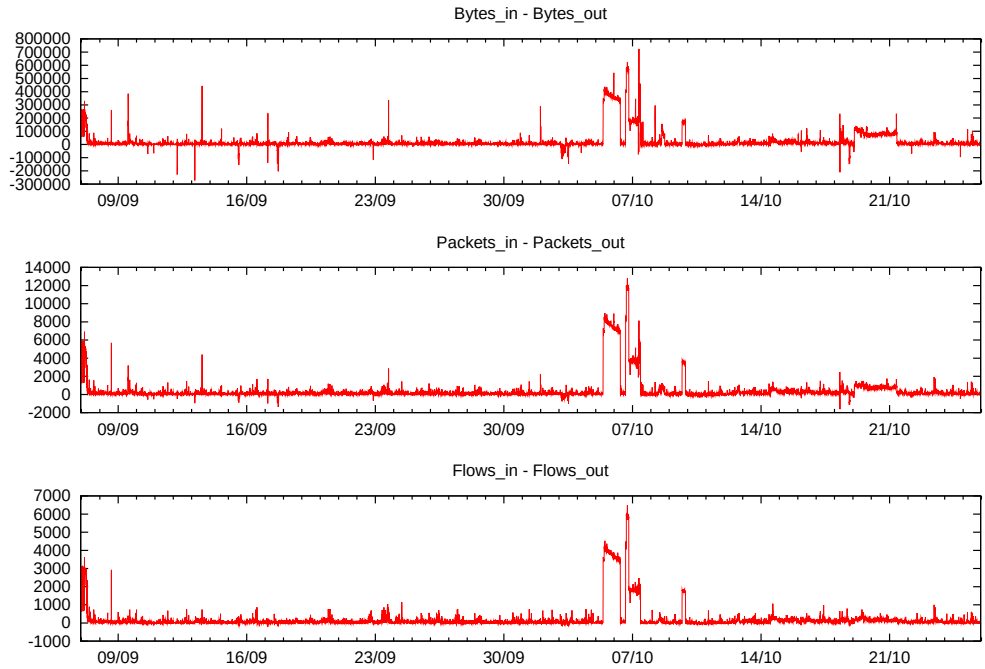

Fig. 6: Measurement time-series of SMB traffic

method which provides good results for various traffic metrics. Among the examined change detection mechanisms, the Shewhart control chart of individuals works fine despite of its simplicity. The lack of knowledge of the residuals' distribution under normal conditions inhibits the calculation of exact control limits for a given false alarm level. Yet, the sensitivity of the detection can be very easily adjusted by defining empirical control limits as multiples of the estimated standard deviation.

From a theoretical point of view, CUSUM and EWMA control charts are better in detecting small sustained shifts in the mean. However, the forecasting-based residual generation is characterized by a differentiation effect: abrupt changes in the measurement time-series result in short impulses in the prediction errors. Therefore, sustained shifts rarely occur in the residual time-series monitored in the control charts.

The relevance of the detected anomalies depends very much on the analyzed traffic and the considered metrics. Most byte and packet anomalies detected in the overall traffic as well as in the SMB traffic were caused by large data transfers. Among these uninteresting alarms, events of actual importance risk to go unnoticed. Therefore, it is advisable to monitor traffic metrics that are less influenced by unpredictable but legitimate behavior of users and applications. Examples are the numbers of ICMP and SMB flows as well as the number of ICMP destination unreachable messages. where most anomalies are caused by suspicious traffic.

In our approach, control limits are calculated relatively to the EWMS estimation of the standard deviation. The benefit of this approach is that the limits dynamically adapt to changes in the residuals. However, we do not stop the update of the limits if an anomaly is detected. Therefore, the control limits are often increased to very high values after an alarm, as can be observed in Figure 4. This problem could be solved by temporarily suspending the update of the EWMS estimator after the detection of an anomaly.

5. RELATED WORK

Hood and Ji [14] convert MIB variables into a measurement time-series, eliminate serial correlation by fitting an AR(2) model, and detect network failures in the AR parameters. Hellerstein et al. use the GLR (generalized likelihood ratio) algorithm to detect anomalies in the number of web server requests per five minutes interval [12]. Systematic changes are eliminated by estimating daily and weekly variations as well as monthly trend from a set of training data. In addition, an AR(2) model is fitted to remove the remaining serial correlation. Brutlag [6] employs Holt-Winters forecasting to model baseline, trend, and daily variation in the outgoing traffic of a web server. Barford et al. [1] apply Holt-Winters forecasting to time-series of packet, byte, and flow counts as a reference anomaly detection approach for their own detection mechanism based on wavelets. The evaluation yields similar detection performance for the two approaches.

Ye et al. use EWMA control charts to detect anomalies in computer audit data [32]. The results are compared to those obtained with a Shewhart individuals control chart applied to the prediction errors of exponential smoothing[†]. Like in our work, the control limits depend on the EWMS estimate of the standard deviation. Paul [19] adopts this method for detecting denial-of-service attacks against web servers.

The optimality of the CUSUM algorithm [5] is frequently brought up to justify its usage for traffic anomaly detection. For example, Wang et al. deploy the CUSUM algorithm to detect SYN flooding attacks. The considered metrics are calculated from the number of TCP SYN, FIN, and SYN/ACK packets [30, 31]. Peng et al. [20] apply the CUSUM algorithm to the number of RST packets returned in response to SYN/ACK packets in order to detect reflector attacks. In [21], the same authors count the number of new source IP addresses to detect distributed denial-of-service attacks. Siris and Papagalou [27] use exponential smoothing to generate prediction errors for the number of SYN packets. The residuals serve as input to CUSUM in order to detect SYN flooding attacks. Similarly, Rebahi and Sisalem [22] use the number of SIP (Session Initiation Protocol) INVITE messages to detect denial-of-service attacks against SIP servers. In order to be optimal, the CUSUM algorithm must be applied to a time-series of independent observations belonging to a specific family of probability distributions. However, none of these works shows that these conditions are fulfilled, hence it is unsure if the CUSUM algorithm actually is the best choice.

The research group of Tartakovsky has proposed several approaches to apply the CUSUM control chart to multivariate data. In [3] and [29], they calculate a chi-square statistic as input for CUSUM in order to detect denial-of-service attacks. For the same purpose, the multichart CUSUM algorithm proposed in [15] and [28] performs separate tests on each component of the multivariate data. Salem et al. apply the multichart CUSUM algorithm to the entries of a count-min sketch to detect SYN flooding attacks and scans [25]. Common to these multivariate methods is the assumption that the components in the multivariate data are mutually independent, which is usually not fulfilled in the case of traffic measurement data. Tartakovsky at al. also downplay the prerequisite of uncorrelated observations arguing that the false alarm rate decays exponentially fast for increasing thresholds [28] under conditions that are to be usually satisfied. Yet, they do not verify if these conditions are fulfilled by the data used in their evaluation.

[†]The authors misleadingly call this approach "EWMA control chart for autocorrelated data" although it actually is a Shewhart control chart.

6. CONCLUSION

We evaluated the applicability of control charts for detecting traffic anomalies. A necessary requirement is the removal of systematic changes and serial correlation from the measurement time-series. We showed that both, seasonal variation and serial correlation can be effectively reduced with robust forecasting techniques based on exponential smoothing. Comparing three different control charts, we determined that CUSUM, although favored by many related works, does not perform better than the simpler Shewhart control chart of individuals when applied to time-series of prediction errors.

Our evaluation based on traffic measurement data collected in an ISP backbone network shows that many anomalies are provoked by legitimate traffic. To increase or decrease the total number of alarms, it suffices to adjust the control limits. Yet, according to our experience, the proportion of alarms that are relevant for the network operator mainly depends on the monitored metrics and the parts of traffic analyzed.

In order to validate our findings, we will conduct similar experiments with measurement data obtained in other networks. Moreover, it will be interesting to examine if dependencies between different metrics can be exploited in a multivariate residual generation process. We observed strong correlation of the number of bytes, packets, and flows in the traffic measurement data, so the detection of changes in the correlation structure may allow us to discover anomalies which cannot be detected in a single metric.

REFERENCES

[1] BARFORD, P., KLINE, J., PLONKA, D., AND RON, A. A signal analysis of network traffic anomalies. In *ACM SIGCOMM Internet Measurement Workshop 2002* (Marseille, France, Nov. 2002).

[2] BASSEVILLE, M., AND NIKIFOROV, I. V. *Detection of abrupt changes: Theory and application*. Prentice-Hall, Inc, 1993.

[3] BLAZEK, R. B., KIM, H., ROZOVSKII, B. L., AND TARTAKOVSKY, A. G. A novel approach to detection of denial-of-service attacks via adaptive sequential and batch-sequential change-point detection methods. In *IEEE Workshop on Information Assurance and Security* (West Point, NY, USA, June 2001).

[4] BOX, G. E. P., AND JENKINS, G. M. *Time-Series Analysis, Forecasting and Control*, 2 ed. Holden-Day, San Francisco, USA, 1970.

[5] BRODSKY, B., AND DARKHOVSKY, B. *Nonparametric Methods in Change-Point Problems*, vol. 243 of *Mathematics and its applications*. Kluwer Academic Publishers, 1993.

[6] BRUTLAG, J. D. Aberrant behavior detection in time series for network monitoring. In *14th Systems Administration Conference (LISA 2000)* (New Orleans, Louisiana, USA, Dec. 2000), U. Association, Ed.

[7] CHATFIELD, C. *The analysis of time series: an introduction*, 6 ed. CRC Press LLC, 2003.

[8] CLAISE, B., BRYANT, S., SADASIVAN, G., LEINEN, S., DIETZ, T., AND TRAMMELL, B. H. Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information. RFC 5101 (Proposed Standard), Jan. 2008.

[9] CLAISE, B., SADASIVAN, G., VALLURI, V., AND DJERNAES, M. Cisco Systems NetFlow Services Export Version 9. RFC 3954 (Informational), Oct. 2004.

[10] DACIER, M. Leurré.com: a worldwide distributed honeynet, lessons learned after 4 years of existence. In *Presentation at Terena Networking Conference* (Bruges, Belgium, May 2008).

[11] GNU OCTAVE HOMEPAGE. http://www.octave.org, 2008.

[12] HELLERSTEIN, J. L., ZHANG, F., AND SHAHABUDDIN, P. Characterizing normal operation of a web server: Application to workload forecasting and capacity planning. In *24th International Computer Measurement Group (CMG) Conference* (Anaheim, California, USA, Dec. 1998).

[13] HIGH-AVAILABILITY.COM HOMEPAGE. http://www.high-availability.com, 2008.

[14] HOOD, C. S., AND JI, C. Proactive network fault detection. In *IEEE Conference on Computer Communications (INFOCOM'97)* (Kobe, Japan, Apr. 1997), pp. 147–1155.

[15] KIM, H., ROZOVSKII, B. L., AND TARTAKOVSKY, A. G. A nonparametric multichart cusum test for rapid detection of dos attacks in computer networks. *International Journal of Computing & Information Sciences 2*, 3 (Dec. 2004), 149–158.

[16] MONTGOMERY, D. C. *Introduction to Statistical Quality Control*, 5 ed. John Wiley & Sons, 2005.

[17] MÜNZ, G., AND CARLE, G. Real-time Analysis of Flow Data for Network Attack Detection. In *Proc. of IFIP/IEEE Symposium on Integrated Management (IM 2007)* (Munich, Germany, May 2007).

[18] PAGE, E. Continuous inspection schemes. *Biometrika 41* (1954), 100–115.

[19] PAUL, O. Improving web servers focused dos attacks detection. In *IEEE/IST Workshop on Monitoring, Attack Detection and Mitigation (MonAM 2006)* (Tübingen, Germany, Sept. 2006).

[20] PENG, T., LECKIE, C., AND RAMAMOHANARAO, K. Detecting reflector attacks by sharing beliefs. In *IEEE 2003 Global Communications Conference (Globecom 2003)* (2003).

[21] PENG, T., LECKIE, C., AND RAMAMOHANARAO, K. Proactively detecting distributed denial of service attacks using source ip address monitoring. In *Networking 2004* (Athens, Greece, May 2004).

[22] REBAHI, Y., AND SISALEM, D. Change-point detection for voice over ip denial of service attacks. In *15. ITG/GI Fachtagung Kommunikation in Verteilten Systemen (KiVS)* (Bern, Switzerland, Feb. 2007).

[23] ROBERTS, S. Control chart tests based on geometric moving averages. *Technometrics 1* (1959), 239–250.

[24] ROBINSON, P., AND HO, T. Average run lengths of geometric moving average charts by numerical methods. *Technometrics 20* (1978), 85–93.

[25] SALEM, O., VATON, S., AND GRAVEY, A. An efficient online anomalies detection mechanism for high-speed networks. In *IEEE Workshop on Monitoring, Attack Detection and Mitigation (MonAM2007)* (Toulouse, France, Nov. 2007).

[26] SHEWHART, W. *Economic Control of Quality Manufactured Product*. D.Van Nostrand Reinhold, Princeton, NJ, 1931.

[27] SIRIS, V. A., AND PAPAGALOU, F. Application of anomaly detection algorithms for detecting syn flooding attacks. In *IEEE Global Telecommunications Conference (Globecom 2004)* (Dallas, USA, Nov. 2004).

[28] TARTAKOVSKY, A. G., ROZOVSKII, B. L., BLAZEK, R. B., AND KIM, H. Detection of intrusions in information systems by sequential change-point methods. *Statistical Methodology 3*, 3 (2006).

[29] TARTAKOVSKY, A. G., ROZOVSKII, B. L., BLAZEK, R. B., AND KIM, H. A novel approach to detection of intrusions in computer networks via adaptive sequential and batch-sequential change-point detection methods. *IEEE Transactions on Signal Processing 54*, 9 (Sept. 2006).

[30] WANG, H., ZHANG, D., AND SHIN, K. G. Detecting syn flooding attacks. In *IEEE Infocom 2002* (June 2002).

[31] WANG, H., ZHANG, D., AND SHIN, K. G. Syn-dog: Sniffing syn flooding sources. In *22nd International Conference on Distributed Computing Systems (ICDCS'02)* (Vienna, Austria, July 2002).

[32] YE, N., VILBERT, S., AND CHEN, Q. Computer intrusion detection through ewma for autocorrelated und uncorrelated data. *IEEE Transactions on Reliability 52*, 1 (Mar. 2003), 75–82.