IDENTIFYING NETWORK FAILURES AND EVALUATING LINK MTBF FROM UTILIZATION LOGS

G. Casale
Neptuny R&D
Via Durando 10-G, I-20158
Milan, Italy

P. Cremonesi
Politecnico di Milano – DEI
Via Ponzio 34/5, I-20133
Milan, Italy

S. Visconti
Neptuny R&D
Via Durando 10-G, I-20158
Milan, Italy

Network failure detection techniques and link Mean Time Between Failure (MTBF) estimates are required to assess the reliability of large communication networks. We present an experience based on utilization logs from a large ISP network comprising hundreds of links, and spreading over a geographic area. The complexity of the network requires accounting also for the mutual dependencies between events on different links. Nevertheless, we show that robust non-parametric data mining methods offer a simple and effective way to accomplish the task.

1. INTRODUCTION

The detection of network failures is of paramount importance for assessing the quality of service in communication networks. Reliability metrics, as the well-known Mean-Time-Between-Failures (MTBF), are widely used to give synthetic information about the frequency of failures either for a service, a server, or any HW/SW component. Despite reliability analysis techniques have been studied for long time [TRIV93], they have recently acquired more and more importance, because of the significant economic losses that may derive from enterprise network unavailability or malfunctions. Unfortunately, the increasing size of enterprise computing infrastructures, which may be composed by several hundreds servers interconnected through large networks, makes it very difficult to *manually* collect, process and analyze reliability data. Hence, accurate reliability assessment of large networks requires *automatic* techniques.

In this paper we discuss the automatic analysis of network failures from link logs from which we can extract utilization data, i.e. the ratio between the current bandwidth usage and the total available bandwidth. The reason why we focus on usage logs is that, very often, link utilization is the only monitored performance metric in communication networks, especially for large or geographic-scale infrastructures. Furthermore, since utilization is a very general performance metric that can be collected for a variety of resource such as CPUs or storage devices, many of the techniques of this work can be extended with little effort to the reliability analysis of any IT system.

In the following sections, we also present a practical application of our techniques. We identify failures in a geographic-scale network of a large ISP for which we are able to study MTBF metrics on the only basis of link utilization logs. A requirement of our analysis is that only failures resulting from simultaneous incidents affecting multiple links have to be considered. A major issue in our analysis is the inherent burstiness of the workload, which makes it difficult to discriminate between workload fluctuation and incidents. In our experience, this problem is worsen by the use of simple statistical means or standard deviations, which can lead to a large errors in the final results because of their high sensitivity to noises in the measured data. We then use the *periodic median* technique for incident identifications, i.e., a non-parametric method described later in the paper, which may provide very good results.

The structure of the present work is as follows. First, in Section 2 we introduce the network failure identification problem in communication networks, as well as some preliminary definitions. In particular, an illustrative figure shows that network incidents can be broadly classified according to four main types, which we call bursts, leaks, heavy bursts and heavy leaks. Next, in Section 3 we describe our incident identification methodology. In particular, we develop in Section 3.1 two techniques for identifying the *expected behavior* (EB) of links required to determine whether an observed deviation is an incident or the result of a mere workload fluctuation. Section 3.2 discusses filtering of the candidate incidents to improve the general quality of the results. In Section 3.3 we present an algorithm to determine network failures from the identified link incidents, and we also summarize MTBF e MTTR formulas. The case study which proves the effectiveness of our methodology is reported in Section 4. Finally, Section 5 concludes the paper.

2. PROBLEM STATEMENT

In this section we give an overview of the considered problem, as well as some required preliminary definitions. From now on, we shall frequently use the following terms: *link* meaning a single Ethernet or ATM channel; *network* meaning any set of interconnected links.

We consider the problem of identifying *network* failures and evaluating both link and network MTBF. We call *failure* a set of almost-simultaneous link incidents which significantly reduce network capacity. Hence, *incident* will denote a single event, while a set of incidents related to a same network problem -that we operationally define as "incidents on different links that are very close in time"- will be considered as a failure. We point out that, since we deal with automatic analysis techniques, we shall ignore in this context the root-causes that originated the failure, i.e., we will classify as failures both a set of unexpected incidents and human-scheduled events (e.g., link maintenance tasks that make some service unavailable). Clearly, whenever such distinction is known to the analyst, it would be rather easy to perform an adequate automatic discrimination among the two classes of failures.

2.1 CLASSIFICATION OF INCIDENTS

In general, the analyst wishes to obtain synthetic information about incidents, e.g., duration and significance of the problem. Failure characteristics are immediately obtained by analyzing the usage data of each link incident that composed the failure. According to this requirement, we consider the following summarizing data:

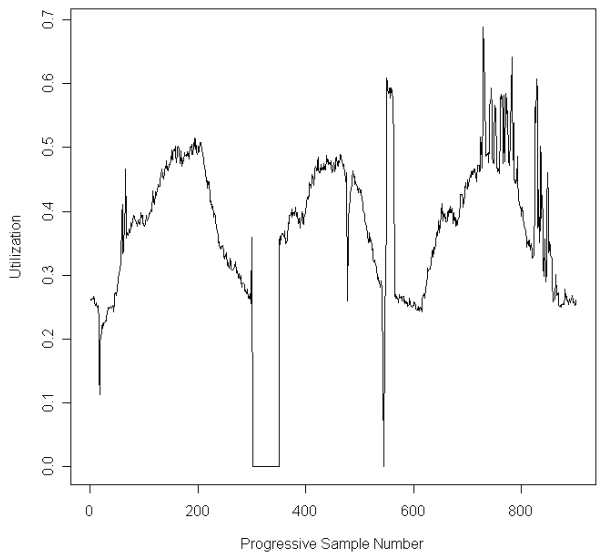

Figure 1. Different types of network incidents

- *Start instant*; i.e., instant at which the link started to behave in a significantly different way from what expected;
- *Duration*, i.e., length of the interval in which the incident lasted;
- *Position of peak value* (i.e., peak within the incident interval);
- *Absolute peak value* (i.e., peak within the incident interval);
- *Peak-to-expected ratio*, that is the ratio between the peak utilization during the incident and the expected utilization at the same instant;
- *Cumulative deviation*, which is the cumulative difference between the observed and the expected utilization.

Some of the above parameters are useful for general analyses. Others can be used effectively to propose a classification of network incidents. In particular, we propose to classify incidents by the following four types:

- *Burst*, i.e., an incident with positive cumulative deviation and with duration less than or equal to a small threshold MAXBURSTDUR>0;
- *Leak*, i.e., similar to a Burst, but with negative cumulative deviation;
- *Heavy Burst*, positive cumulative deviation and duration greater than MAXBURSTDUR;
- *Heavy Leak*, negative cumulative deviation and duration greater than the MAXBURSTDUR.

As said before, the term "heavy" stands for an incident of not-too-short duration that is typically the result of a major network fault interesting several links. A typical example is when a router stops processing packets due to some malfunctions. Clearly, the TCP/IP protocol reacts to this situation by rerouting the unprocessed traffic flow over different links. Ideally, this should be done smoothly, i.e., without resulting in congestions or significant overheads on the newly-loaded links. Unfortunately, such load-balancing may not take place for several reasons such as insufficient overall capacity, or excessive reaction speed by the protocols. Hence, heavy bursts can be experienced in this case.

As an example of the given definitions, consider Figure 1, which shows several types of network incidents over a real link utilization sample taken on a time interval of approximately three days. As we can see, the workload exhibits a number of peaks and leaks which act as perturbations on what we may reasonably expect to be the typical behaviour. Furthermore, an ideal choice of MAXBURSTDUR should allow us to discover that just before the instants 400 and 600 two converges, respectively a leak and a peak converge, took place.

From this small example, it is easy to see that the proposed classification is a natural way to describe

what is immediately evident to a human eye when observing a utilization trace. In the following section we shall define how to automatically infer the existence of incidents in usage data.

3. INCIDENT IDENTIFICATION

In order to be able to estimate network incidents algorithmically, and consequently being able to detect failures automatically, we need a precise definition of incident. Clearly, this in general depends on the application field, i.e., an incident may be equal to an "intrusion" for a security analyst, while for a reliability expert it would be closer to a "HW/SW fault". Nevertheless, in all cases the analyst implicitly refers to a concept of "normal usage", i.e., an incident is a significant deviation from an *expected behavior* (EB). In what follows, we formalize both the EB and the concept of "significant deviation". We sketch the proposed incident identification methodology (which is also illustrated in Figure 2):

1. at the first step, the analyst should extract utilization traces from logs, thus obtaining what we call the *observed behavior* (OB). This data may be either raw or filtered according to some criteria, e.g. missing values may be replaced by suitable values derived from the observed ones.
2. This data is then used by two sub-procedures. The first one, which we believe to be the most important for the quality of the final results, is responsible for estimating the EB from the OB. There are several possible ways to do that, which will be summarized in Section 3.1.
3. The second sub procedure is responsible for identifying the significant deviations of the OB from the EB. In general, this task can be accomplished quite easily by evaluating the deviations

$$Y(t) = |OB(t) - EB(t)|, \qquad (1)$$

where $|\square|$ denotes the absolute value, and by setting a critical deviation threshold DEVTHRES>0 such that a "significant" deviation has to satisfy

$$Y(t) > DEVTHRES. \qquad (2)$$

Unfortunately, this is not always sufficient to correctly identify the most part of the incidents. For instance, consider Figure 3, which shows a real workload for an high-speed link. In this case, the heavy leak may not be identified by the condition (2), since the utilization loss has a small entity, around 5%. Note that the resulting deviation of the traffic flow may be anyway significant, especially affecting performance of low capacity or high utilized links. In order to account for this, we define two additional absolute criteria, i.e.

$$OB(t) > MAXTHRES, \qquad (3)$$

$$OB(t) < MINTHRES. \qquad (4)$$

As we will show in the experimental results, conditions (2)-(4) are enough to identify the most part of network failures.

4. In the final step we refine the set of identified incidents by first characterizing their behavior using the summarizing data described in Section 2.1, and then applying proper filters which in general depend on the analyst objective and will be discussed in Section 3.2.

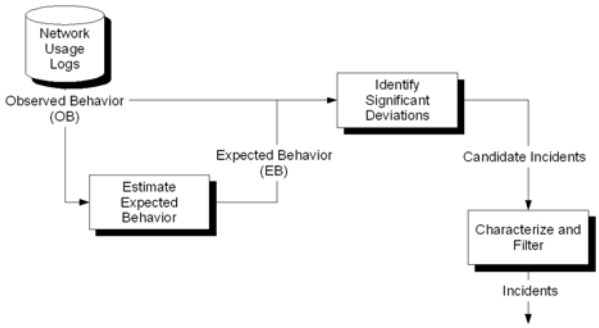

Figure 2. Incident identification methodology

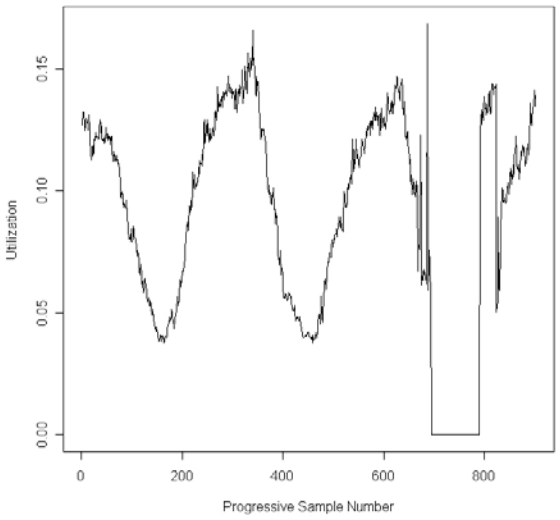

Figure 3. Heavy leak on a high-speed link

We now describe in the following subsections the EB estimation and the incident filtering techniques. Finally, the aggregation of incidents in order to determine

failures and compute their MTBF will be discussed later in Section 3.3.

3.1 EXPECTED BEHAVIOUR ESTIMATES

In this section we define new techniques to determine the EB from the OB. The problem can be specified as follows. The OB is a time series composed by T measured utilizations, labeled 1,2,…,T, which are extracted from log data. We need to determine the time series EB that has the same length of OB, but representing the most-likely behavior of the link in that period. We now propose three techniques we considered for estimating the EB.

3.1.1 SARIMA Model

The acronym SARIMA stands for *seasonal autoregressive integrated moving average*, and it is a stochastic model, fitted from measurements, that is able to forecast the future evolution of a time series even in presence of seasonal data. Confidence bounds for the prediction can be easily generated. Compared to the classic ARIMA models [HAMI94], a SARIMA process allows modeling also the seasonal part of the series. This is extremely useful in network performance analyses, since there is often a strong periodicity in workloads, strongly depending with the day-activity cycle. We also point out that several mathematical applications, as the open source statistical suite R [IHAG96], have functions to work with SARIMA models.

SARIMA models are meant for short-term predictions. Thus, they can be effectively implemented for real-time incident detection. For example, let us consider in Figure 4 a time series similar to the one in Figure 3. The prediction provided by the SARIMA model, depicted in red, gives a good approximation of a reasonable EB. The best model was fitted automatically using the popular Akaike Information Criterion (AIC) [AKA74], and uses over 33 model coefficients. The computational requirement for the fitting was less than 30 seconds. The dataset used was a set of approximately 4000 measured points before the heavy leak during the interval 700-800 in Figure 4. It is possible to note that the quality of the prediction, and especially its burstiness, quickly degrades with the number of steps ahead. Nevertheless, the SARIMA prediction is indeed effective for the identification of the heavy leak, which shows a significantly high deviation from the red values.

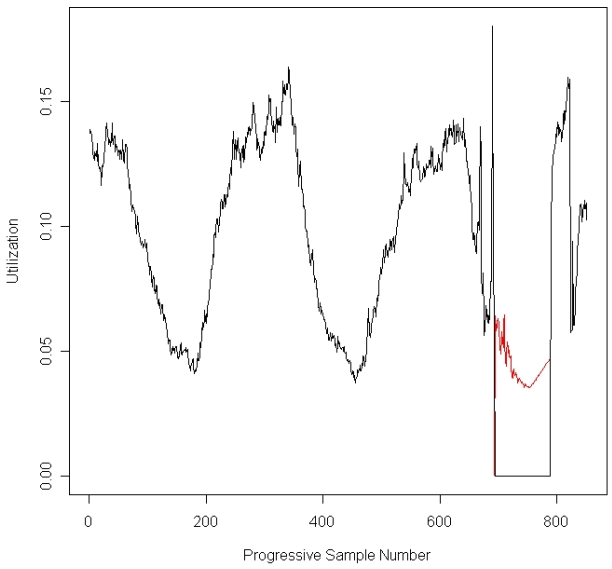

Figure 4. SARIMA EB versus a heavy Leak

Finally, we remark that despite SARIMA models can provide good estimates of the EB, the quality loss for long-term prediction would require the analyst to fit several hundreds models in order to determine the complete EB signal on medium or long periods. Hence, the use of SARIMA models should limited to online EB prediction. A more computationally efficient technique is the *periodic median* that is presented next.

3.1.2 Periodic Median

We now propose the periodic median as a simple, computationally efficient, technique for determining the EB. This technique is fit for data with periodic properties, a frequent situation in real workloads. Let P be the main period of the data (e.g., number of samples in 24 hours), and assume that the OB contains N+1 periods. We define the EB as the signal

$$EB(t) = med(OB(t+nP)),\ n=\{0,\ldots,N\}, \quad (5)$$

where med(□) denotes the median, and so EB(t) is the median value at instant t in the N+1 periods. The periodic median EB is shown in red in Figure 5. As we can see, this is a highly reasonable estimate, which even fits perfectly the third period. Assuming a DEVTHRES of e.g., 5% utilization, we would be able to easily discriminate leaks and bursts from the natural workload burstiness (e.g., the bursts at the top of the second period). Therefore, we conclude that the EB estimate through periodic median can be very effective in all cases, assuming that the data exhibits a constant periodicity with similar statistical characteristics. We show in the section what happens when the workload has regime switches.

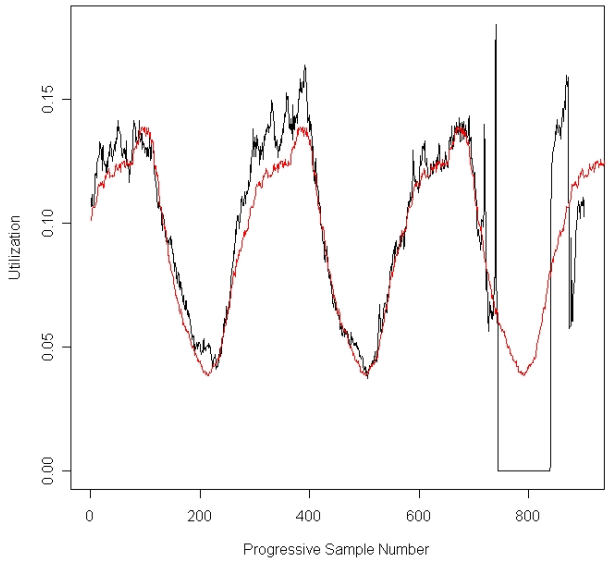

Figure 5. Periodic Median for the same set in Fig 4

3.1.3 Periodic Median with Regime Switch

It is not infrequent to find regime switches in network usage logs. For instance, consider the case of a link whose capacity is upgraded at instant `t0`. What we reasonably expect after the upgrade is a generalized reduction of utilization levels for the link, i.e., the utilization enters a different statistical regime. For our purposes, this has the undesired consequence of yielding wrong results when using the periodic median on data belonging to different regimes. Nevertheless, this case can be efficiently handled by identifying the instant `t0` and then running the periodic median only on the data on the current regime. This can be done very accurately by implementing the regime identification techniques presented in [TAKY06].

3.2 FILTERING CANDIDATE INCIDENTS

In same cases, network managers are interested in identifying only some failures, rather than all possible classes of incidents. For our purposes, we were interested in identifying heavy bursts and leaks. Under this assumption, we need to filter candidate incidents according to their likelihood of being part of a composite event. This can be performed by comparing the candidate incidents obtained for a number of links, and then ruling out all isolated incidents, i.e., problems which affected a single link. In our experience, this kind of filtering turned out to be very effective to isolate only the heavy events.

Another kind of possible filtering requires accounting for the amount of the traffic flows rerouted through the links. We may reasonably expect some kind of conservation law of the amount of the rerouted traffic flow. For example, we may observe that if link A shows a leak of 5% utilization, then the corresponding traffic should result as a burst on other links. It should be anyway observed, that this approach, despite may sound natural, may not be effective for several reasons, among which:

1. it is not always true that the rerouting of the traffic does not affect the amount of traffic already flowing in the newly-loaded links. Since the rerouting imposing new load on the routers, this may decrease the overall quality of service of the network, thus leading customers to reduce their outbound traffic. Hence, for large traffic flows, the rerouted packets may have a feedback on the traffic amount of all network flows which may affect analysis accuracy.
2. When some of the newly-loaded links are able to seamlessly process the new incoming rerouted traffic without significantly increasing their utilization level, it would impossible to distinguish such small deviation from workload burstiness. Hence, it would be very hard to observe any traffic conservation law.
3. Finally, whenever the only monitored metric is utilization, it is not possible to verify a conservation of total utilization unless all network links have the same capacity or we known all link speeds, an information which may hard to find for geographically distributed infrastructures.

3.2 MTBF ESTIMATES

The final output of our analysis is the link and network Mean-Time-Between-Failures (MTBF) reliability metrics. In the context of network reliability, this is simply the average time between failures of a link, or in a wider sense of network service. Clearly, devising the MTBF for links is a straightforward task once that all incidents have been identified. In fact, it is sufficient to study the number of incidents observed in the period under study.

The general case of determining network MTBF is harder, since up to now we have developed techniques for identifying and filtering incidents, and not in general failures. In order to obtain failures from incidents we need some algorithm to map the set of incidents to a set of network failures.

In order to understand the problem, consider the following example table, which may be regarded as the final output of the last step in the chain of Figure 2:

INCIDENT	LINKID	START	END	TYPE
1	A	700	703	BURST
2	A	705	730	CONVERG. LEAK
3	B	732	784	CONVERG. BURST
4	C	680	710	CONVERG. LEAK

A possible interpretation of the table is that an incident on the link C started at instant 680, and then it implied a number of related problems on the links A and B, up to the instant 784. This is somehow intuitive, since the small intervals between the end of an event and the successive one indicate a close relationship. Hence, intuition suggests that the four incidents above should be classified as a single network failure, which originated in C and spread through A and B.

In order to develop a proper algorithm for identifying network failures from the incident table, we need again to introduce a threshold which quantifies the maximum distance between two incidents relating to a same network failure. Let us call this value FAILURETIMEOUT. The general structure of the failure identification algorithm is then as follows:

INPUT: Link Incident Table
OUTPUT: Network Failure Table

1. Sort the Incident Table rows according to the START field, in ascending order
2. Set FAILSTART and FAILDUR equal to the START and DUR fields of the first row
3. **FOR** all table rows starting from the second
4. **IF** START is less than or equal to FAILSTART+DUR+FAILURETIMEOUT
   the current event belongs to the same failure. Set FAILSTART=min(FAILSTART,START) and FAILDUR=min(FAILDUR,START+DUR).
5. **ELSE** store the current failure. Set FAILSTART and FAILDUR equal to START and DUR.
6. **END IF**
7. **END FOR**

Algorithm 1. Failure Identification Algorithm

After that the network failure table is produced, the computation of the network MTBF is straightforward using the formula

$$MTBF=T/N, \quad (6)$$

where N denotes the number of identified failures.
Another reliability index of interest is the Mean Time To Repair (MTTR) which summarizes the average duration of a network failure and can be computed as

$$MTTR=\Sigma\ FAILDUR/N. \quad (7)$$

4. CASE STUDY

In order to show the effectiveness of our approach, we illustrate the results of our approach on the analysis of real usage data. We considered the utilization logs of a geographic-scale composed by hundreds of links with different capacities. Our data was composed by usage samples taken at 5-minutes granularity, over a time period of approximately four weeks excluding Saturdays and Sundays (i.e., ~7000 samples).

The main frequency of the periodic signals was identified automatically using the Fast Fourier Transform (FFT), and was equal to a period of 24 hours. Missing samples were reconstructed using a moving average approach.

The identification of the candidate incidents was performed by a periodic median, with period P equal to 24h. For the considered links, we did not find regime switches.

The candidate incidents were filtered according to their duration (a minimal threshold was used), and their impact of the network, i.e., we were mainly interested at heavy events, thus all incidents related to a single link were filtered out.

The results of our analysis are presented in Figure 6. Each red area represents a network failure. Failures with small durations have been evidenced by red circles instead of red areas.

As we can see, our approach allowed us to mine the existence of three major failures, approximately starting at positions 800, 2300 and 5800.

Furthermore, the simultaneous comparison of multiple time series gave us evidence of several minor failures, which may have been difficult to detect without a multiple comparison. For instance, the first small leak in the second series of Figure 6 would have not probably classified as the result of a network failure to a human eye, but this becomes immediately evident by comparison with the incidents in the other series.

Overall, the result of our analysis is very close to a manual identification, and this proves the effectiveness of the proposed techniques.

The main lines for our future research in the field will be the following:

1. experimental validation of the presented techniques in presence of regime switches which may lead to bad estimates of the EB;
2. comparison of the SARIMA and periodic median estimates will classic window-based filters such as the moving average and the moving median;

3. refinement of the significant deviation identification by accounting the statistical nature of the process, and not only fixed thresholds.

5. CONCLUSIONS

In this paper we discussed the problem of identifying network failures and estimating reliability metrics from network usage logs.

We have described a simple general methodology based on the concept of expected behavior (EB). We discussed three techniques for computing the EB from the observed usage data, showing the effectiveness of a non-parametric technique, the periodic median, to accomplish this task.

We then introduced a simple algorithm to aggregate the identified network incidents into a series of network failures. These allow devising straightforwardly MTBF and MTTR estimates.

Finally, we illustrated the effectiveness of our approach on a real case study based on the utilization logs of a geographic-scale network.

BIBLIOGRAPHY

[AKA74] H. Akaike, “A new look at the statistical model identification”, *IEEE Transactions on Automatic Control*, 19(6):716-723, (1974).

[HAMI94] J.D. Hamilton, “Time Series Analysis”, Princeton University Press, (1994).

[IHAG96] R. Ihaka, R. Gentleman, “R: A language for data analysis and graphics”, *Journal of Computational and Graphical Statistics*, 5(3):299-314, (1996).

[TAKY06] J. Takeuchi, K. Yamanishi, " A Unifying Framework for Detecting Outliers and Change Points from Time Series”, *IEEE Transactions on Knowledge and Data Engineering*, 14(4):482-492, (2006).

[TRIV93] K.S. Trivedi, " Probability and Statistics with Reliability, Queuing and Computer Science Applications”, Prentice-Hall, (1993).

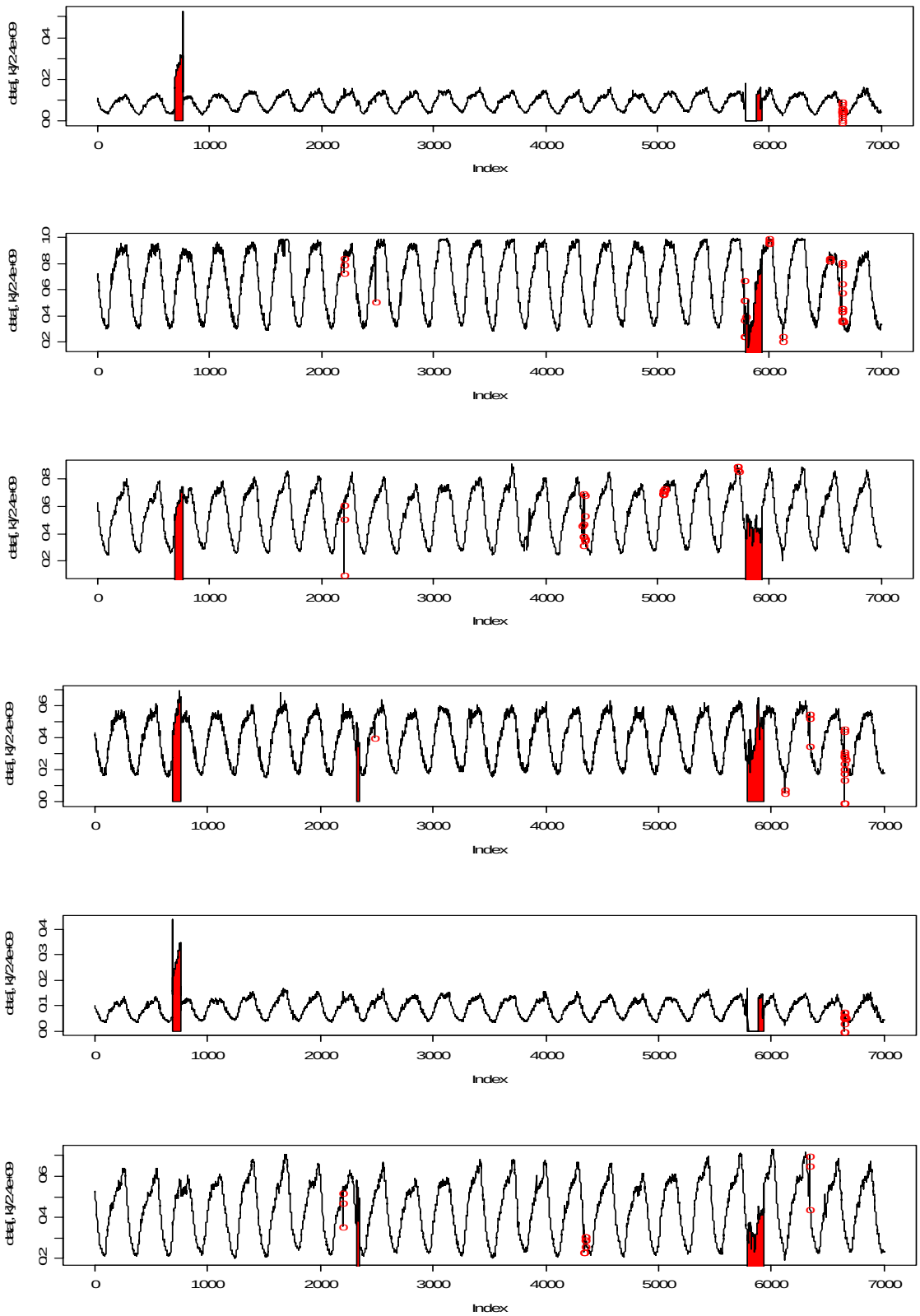

Figure 6. Incident identification methodology.
Each time series represents network traffic on a link.
Red areas and red circles represent network failures.