Adaptive Network/Service Fault Detection in Transaction-Oriented Wide Area Networks

L.L. Ho, D.J. Cavuto, M.Z. Hasan,
& F.E. Feather
Bell Labs, Lucent Technologies
101 Crawfords Corner Road
Holmdel, NJ 07733
USA
{llho, cavuto}@bell-labs.com

S. Papavassiliou, & A.G. Zawadzki
AT&T Labs, AT&T
200 Laurel Avenue South
Middletown, NJ 07748
USA
{spapavassiliou, doose}@att.com

Abstract

Algorithms and online software for automated and adaptive detection of network/service anomalies have been developed and field-tested for transaction-oriented wide area networks (WANs). These transaction networks are integral parts of electronic commerce infrastructures. Our adaptive network/service anomaly detection algorithms are demonstrated in a commercially important production WAN, currently monitored by our recently implemented real-time software system—TRISTAN (Transaction Instantaneous Anomaly Notification). TRISTAN adaptively and proactively detects network/service performance degradations and failures in multiple service-class transaction-oriented networks, where performances of service classes are mutually dependent and correlated, and where external or environmental factors can strongly impact network and service performances. In this paper, we present the architecture, summarize the implemented algorithms, and describe the operation of TRISTAN as deployed in the AT&T Transaction Access Services (TAS) network. TAS is a commercially important, high volume, multiple service classes, hybrid telecom and data WAN that services transaction traffic in the US and neighboring countries. It is demonstrated that TRISTAN detects network/service anomalies in TAS effectively. TRISTAN can automatically and dynamically detect network/service faults, which can easily elude detection by the traditional alarm-based network monitoring systems.

Keywords

Network fault detection, anomaly detection, fault management, transactions oriented networks, electronic commerce, wide area networks.

1. Introduction

Proactive detection of network failures and performance degradations is a key to rapid fault recovery and robust networking, and has been receiving increasing attention lately [1–3]. The past few years have witnessed much progress in path failure detection (including break faults) [4–6], billing fraud detection in telecommunication voice networks [7], anomaly detection in Ethernet [8,9], anomaly and performance change detection in small networks [10,11], and network alarm correlation [12–15]. With the advent and explosive growth of the global Internet and electronic commerce environments, adaptive/automatic network and service anomaly detection in *wide area* data networks and E-commerce infrastructures (e.g., transaction-oriented networks and services) is fast gaining research and practical importance [16,17]. In these cases, being able to proactively detect performance degradations (termed "soft" faults as opposed to the "hard" alarms/failures of networks and their elements [8,9]) and "hard" network faults in wide area networks is becoming crucial for speedy fault recovery and for preventing the onset of network/service failures. As communication infrastructures are evolving into multiplexed and multiple service-class networks, network resource sharing by multiple service-classes correlates the performances of all classes that are supported in logically partitioned networks (e.g., VPNs, ATM networks, and transaction-oriented networks). Hence, performance degradation in one set of service classes can impact negatively the performances of the rest. This again demands anomaly detection of network and service faults [17]. Finally, faulty elements that are outside the jurisdiction of network monitoring systems can degrade the performance of the monitored network proper (see Figure 1). In this case, anomaly detection can infer the presence of these non-monitored failures (i.e., no MIB nor trap information access) from the monitored performance data of the networks, and enable timely fault recovery. Together, these considerations motivate our research in using network performance data to detect network/service anomalies in wide area communication infrastructures. Our research and software apply to fault management of the following networking environments:

- *Wide area networks and E-commerce infrastructures (as opposed to local area networks or small campus networks [8–11]),*
- *Wide area networks where proactive detection and timely recovery of service performance degradations (or "soft" faults) are important (compared with generating and processing "hard" network alarms [12–15]),*
- *Multiple service-class networks where resource sharing is significant and service-class performances are correlated (e.g., in a multiple service-class transaction network, or a virtual private networking infrastructure), and*

- *Networks where the non-managed parts and environmental components (e.g., customer-site equipment, and attached networks) can fail and consequently degrade the performance of the managed network proper.*

To detect network/service faults effectively and automatically, software should be developed such that statistical signatures of network/service anomalies can be recognized algorithmically. This in turn implies that expected network and service performances have to be measured and generated from historical network data as networks function. And that historical performance data and alarm logs (and their problem resolution) of representative wide area data communication and E-commerce networks should be analyzed statistically and visually to draw conclusions that can be implemented in anomaly detection algorithms and software. To this end, we investigated the performance dynamics of a variety of telecom and data networks, and developed methods and algorithms for automatic detection of network/service anomalies from its performance data [17,18].

In this paper, we describe proactive anomaly detection in transaction-oriented WANs by presenting the algorithms, architecture, and application of an on-line software system we developed—the TRISTAN (Transaction Instantaneous Anomaly Notification) system. TRISTAN implements the anomaly detection algorithms we developed [17,18]. We also describe the application of TRISTAN to anomaly detection in a commercially important transaction-oriented wide area network (the AT&T Transaction Access Service network).

TRISTAN is currently being tested as an on-line fault detection system for the commercially important AT&T Transaction Access Services (TAS) network. In this context, the TRISTAN system, as currently implemented, is capable of

- Adaptively sampling the TAS transaction records in real-time with dynamically defined sampling windows and on a per service class basis. This algorithm is partly designed to highlight transactions that have high probability of being anomalous,
- Automatically building dynamic thresholds for all TAS service classes to baseline their individual performances and the overall network performance. These thresholds are updated periodically and automatically to account for the performance evolution of the TAS service classes,
- Detecting network and service anomalies of TAS as dynamically defined violations of the baselined performance characteristics and profiles. In addition to being applicable to faults originated within the TAS network, this also applies to anomaly and faults that may occur outside the

jurisdiction of the TAS monitoring system. Environmental and external failures that impact the performance (or threaten total failure) of TAS are inferred by TRISTAN for timely damage control and fault/performance management, and

- Detecting TAS faults reliably and proactively, as will be illustrated by a concrete example. Some of these faults originated from the non-managed part of the network, and had historically led to TAS service degradations. Being able to detect these faults in the early stage before problem escalation enables early recovery and is a key feature of TRISTAN.

Section 2 concerns network anomaly detection algorithms and processes. Section 3 concerns the AT&T TAS network, and its relevance to network anomaly detection. Section 4 concerns the architecture and implementation of the TRISTAN system, and service-class enabled anomaly detection by TRISTAN. Section 5 provides a real-life example of TAS anomaly detection by TRISTAN, and section 6 provides a summary.

2. Network Anomaly Detection

A network anomaly detector is a real-time software that adaptively analyzes performance data of managed networks to detect "abnormal" changes (relative to some historical baselines or "expected" behavior) in network traffic and performances, which are signatures of soft and hard network faults. The three steps of adaptive network anomaly detection are:

1. *Preferential sampling of transaction records*

   This self-consistently and preferentially samples the network (e.g., transaction records generated by network switches) to detect transactions that have high probabilities for being anomalous. The sampling scheme strikes a balance between sampling frequencies and performance resolution [17].

   Each transaction "*i*" in a transaction-oriented network is characterized by a 3-tuple: (1) starting time of transaction, t_i, (2) duration of transaction, Δt_i, and (3) a service-class identifier. Transaction durations (time-stamped by t_i) are computed to form traffic intensities at discrete time interval for every service class. The traffic intensity of a service class provides a measure of the total number of circuits dedicated to that service class in real-time. For a service class "*s*", its traffic intensity $I_s(T_{n,s})$ at discrete time $T_{n,s}$ (*n* is an integer, where its maximum value N_s determines the total number of daily time intervals) is

$$I_s\left(T_{n,s}\right) = \sum_i \Delta t_i \Big/ \delta T_s \Big|_{T=T_{n,s}} ;T_{n,s} = n \times \delta T_s , n = 0,1,\cdots, N_s .$$

where transactions within a time bin (defined by the sampling time δT_s, which is service-class dependent) for the service class are summed; and $N_s=(24\times 60\times 60)/\delta T_s$ (in this case, the unit of δT_s.is second) is the daily number of time bin of "*s*". The sampling time δT_s is adaptive and dynamically determined from historical transaction records. It is related to the historical "average" and "upper-limit" transaction duration of a service class, and is so determined that transactions with high probability of being anomalous are preferentially highlighted. Depending on the historical transaction pattern of a service class, abnormally long (defined by historical data) transactions are highlighted by its sampling time in the traffic intensities (see Figure 1). From data analysis and verification, this method has been shown to be capable of effectively highlighting potentially anomalous transactions.

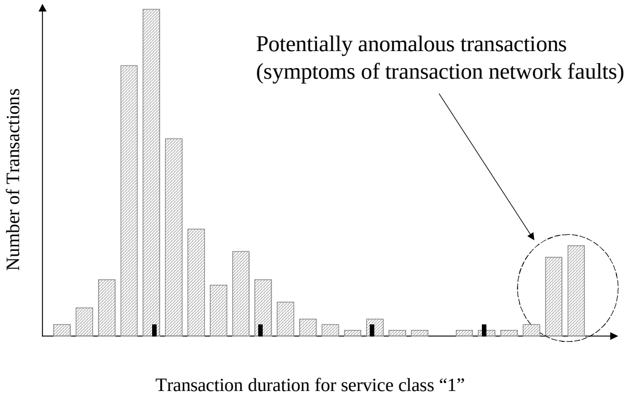


Figure 1: A typical histogram of transaction duration of a service class. Potentially anomalous transactions (as indeed they are) are highlighted

2. *Temporal-based performance thresholds*
   By exploiting the temporal performance regularities of networks, performance thresholds of each TAS service can be classified into 4 classes: weekdays, Saturdays, Sundays, and holidays. Historical data of service classes are used to construct these adaptive thresholds for each TAS service. Expected performances of TAS services are predicted through these thresholds [17,18].

For each of the 4 threshold groups, a set of adaptive thresholds are built to predict the expected performance of TAS services on weekdays, Saturdays, Sundays, and holidays, respectively. Each set of dynamic thresholds (upper and lower thresholds) is composed of a predicted baseline $\tilde{I}_s\left(T_{n,s}\right)$ and tolerance $\tilde{\sigma}_s\left(T_{n,s}\right)$ (note: "~" denotes "predicted") as follows

$$\text{upper\_threshold} = \tilde{I}_s\left(T_{n,s}\right) + 2\tilde{\sigma}_s\left(T_{n,s}\right)\Big|_{\substack{wkdys\\sats\\suns\\holiday}}$$

$$\text{baseline } = \tilde{I}_s\left(T_{n,s}\right)\Big|_{\substack{wkdys\\sats\\suns\\holiday}}$$

$$\text{lower\_threshold} = \tilde{I}_s\left(T_{n,s}\right) - 2\tilde{\sigma}_s\left(T_{n,s}\right)\Big|_{\substack{wkdys\\sats\\suns\\holiday}}$$

The baseline $\tilde{I}_s\left(T_{n,s}\right)$ and tolerance $\tilde{\sigma}_s\left(T_{n,s}\right)$ are computed from historical transaction data through one-dimensional time series analysis and are classified into the "weekday", "Saturday", "Sunday", and "holiday" classes [17,18]. The $\tilde{I}_s\left(T_{n,s}\right)$s represent the predicted "average" traffic intensities of service classes, while the $\tilde{\sigma}_s\left(T_{n,s}\right)$s represent the predicted "average" fluctuations of the corresponding traffic intensities. Both $\tilde{I}_s\left(T_{n,s}\right)$s and $\tilde{\sigma}_s\left(T_{n,s}\right)$ are updated periodically to account for the evolution in network traffic.

3. *Anomaly detection*

Expected performances of TAS services are predicted through the above thresholds, and deviations (in both magnitude and duration, as defined by a set of fault criteria) from the expected are indications of network/service anomalies [17,18].

In anomaly detection, an alarm is sounded that signals the arrival of a network/service anomaly if (1) the measured (in real-time) traffic intensity $I_{s,measured}(T_{n,s})$ at time $T_{n,s}$ deviates from the thresholds by more than a from the predicted baseline, and (2) the previous condition persist and for more than $T_{persist}$, i.e.,

$$\left[\tilde{I}_s\left(T_{n,s}\right) - 2\tilde{\sigma}_s\left(T_{n,s}\right)\right] - a\tilde{I}_s\left(T_{n,s}\right) \geq I_{s,measured}\left(T_{n,s}\right) \text{ or}$$
$$I_{s,measured}\left(T_{n,s}\right) \geq \left[\tilde{I}_s\left(T_{n,s}\right) + 2\tilde{\sigma}_s\left(T_{n,s}\right)\right] + a\tilde{I}_s\left(T_{n,s}\right).$$

> As will be explained in the following sections, the choice of the parameters in the above criterion (*a* and $T_{persist}$) are determined experimentally. Finally, the detected anomaly is mapped to an diagnosis log that identifies the “guilty” service class(es) and the possible cause(s) of anomaly. This information is presented to network operators through a graphic user interface (GUI).

A generic architecture of a transaction anomaly detection system is shown in Figure 2. In this system, network performance data are accumulated on-line by the sampler for analysis. The sampler outputs performance measures (e.g., traffic intensities, or circuit utilization, for service classes in a transaction-oriented network) in which potential anomalous data are highlighted. The historical network performance data output by the sampler are analyzed by the rule generator to build adaptive and dynamic (i.e., temporally based) performance thresholds. These performance thresholds are updated periodically to account for the evolution network traffic. The detector compares real-time network performance data output by the sampler with performance thresholds and predefined fault criteria for anomaly detection. The outputs of the detector are typically sent to a graphic user interface (GUI) to alert network operators of network anomalies and faults, or are sent directly to network control modules for automatic feedback and control (e.g., circuit breaker, rerouting module, etc.).

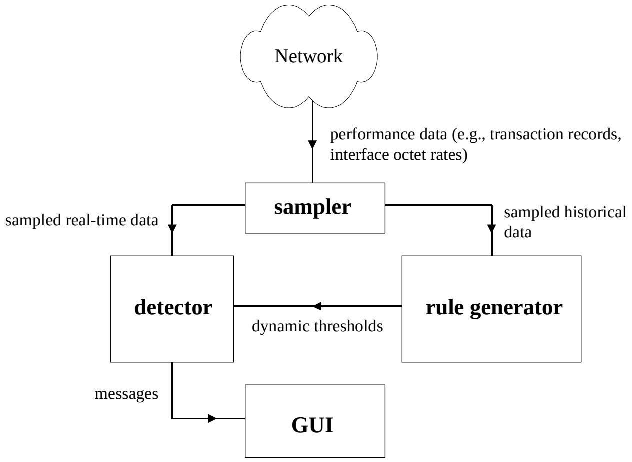


Figure 2: Generic architecture of a network anomaly detection system.

3. The AT&T Transaction Access Service Network

The AT&T Transaction Access Service (TAS) network is a hybrid POTS-and-data wide area network (WAN) that provides ubiquitous dial-to-packet services for carrying short-duration transaction traffic in the United States, Canada, and the Caribbean countries [19]. Average usage of the TAS network amounts to millions of transactions on a non-busy and typical day, and is growing rapidly. The network supports tens of service classes. Typical transactions support point-of-sale applications/services (e.g., credit/debit card authorization and settlement), health care applications, banking and vending applications, and other data-driven sales applications.

The TAS network is selected as the testbed for the TRISTAN system for three reasons. Firstly, TAS services a relatively large number (tens) of mutually dependent service classes, which is a multi-service-class environment where performances of service class are strongly correlated with one another and with the overall network performance. Secondly, the shear traffic volume (averages many millions of transactions per day) supplies a statistically rich data set for anomaly detection. Thirdly, the presence of external devices that are not monitored/managed by TAS (e.g., credit card servers) enables the testing and demonstration of TRISTAN's ability to detect and infer environmental (non-monitored) impact on the network and its service classes.

The physical topology of the TAS Network consists of three major components—the AT&T 800 Network, the TAS nodes (for POTS-to-packet protocol conversion), and the AT&T Packet Service, as illustrated in Figure 3.

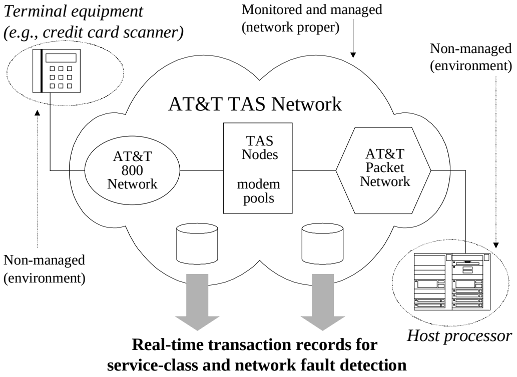


Figure 3. The AT&T TAS Network physical architecture

The central function of the TAS network is to enable transaction-oriented communication between terminal devices (e.g., credit card scanners) scattered across the United States and their designated processing hosts (e.g., credit processing servers). Device access to TAS is accomplished through the AT&T MEGACOM 800 Network, which is terminated at a set of TAS nodes (modems) that act as protocol converters. These nodes use the DNIS (Dialed Number Identification Service) digits provided by the 4ESS switches in the 800 network to establish SVCs in the AT&T packet network. Finally, the packet network is used to complete the connection between the customer devices and their destined host processors. In a typical transaction, call originated in a terminal device is processed by the 4ESS switches in the AT&T network, and is routed to a geographically proximate TAS modem pool. A virtual connection is further set up between the modem and the host processor through a set of packet switches. The result is an end-to-end circuit that connects the E-commerce terminal device and its processor for the duration of the transaction. This circuit is dropped as soon as the transaction is completed.

4. The TRISTAN System

Currently, an on-line network anomaly detection system has been implemented for the AT&T TAS network. This software system is called TRISTAN (Transaction Instantaneous Anomaly Notification). TRISTAN possesses three functional modules: the sampler, the threshold generator, and the detector, in addition to a GUI.

The TRISTAN sampler analyzes the real-time and historical transaction records of TAS to generate traffic intensities on a per service class basis. TAS generates transaction records for every TAS transaction on a 15 minutes and daily basis for delivery to TRISTAN. The 15-min data feeds constitute the real-time performance data while the daily data feeds are used as historical data for threshold generation. In either case, each transaction record is possessed to yield the service-class based traffic intensities. The sampling interval of each service class is unique, adaptive, and dependent on its historical "average" transaction duration. In the current implementation of TRISTAN, the 15-min data feeds are inserted dynamically into a relational database (Oracle). The sampler retrieves records from the database to compute the real-time traffic intensities for all TAS service classes, based on the sampling intervals of the individual service classes. For anomaly detection, these real-time traffic intensities are compared in real-time with the dynamic threshold templates for all TAS service classes. The daily feeds are stored as flat files, which are analyzed by the sampler to yield thresholds and baselines for all service classes

Specifically, for each TAS service, the dynamic thresholds are classified into 4 groups: (1) weekdays, (2) Saturdays, (3) Sundays, and (4) holidays, as explained in Section 2.

In the TRISTAN detector, an alarm is sounded that signals the arrival of a network/service anomaly if the measured (in real-time) traffic intensity $I_{s,measured}(T_{n,s})$ at time $T_{n,s}$ deviates from the thresholds by more than 50% (i.e., $a = 0.5$) of the predicted baseline and for more than 15 minutes, i.e.,

$$\left[\tilde{I}_s(T_{n,s}) - 2\tilde{\sigma}_s(T_{n,s})\right] - 0.5\tilde{I}_s(T_{n,s}) \le I_{s,measured}(T_{n,s}) \quad \text{or}$$
$$I_{s,measured}(T_{n,s}) \ge \left[\tilde{I}_s(T_{n,s}) + 2\tilde{\sigma}_s(T_{n,s})\right] + 0.5\tilde{I}_s(T_{n,s})$$

The choice of the parameters in the above criterion (15 minutes and 0.5) are TAS-specific, and are determined empirically. Through our field testing of TRISTAN, they have been demonstrated to work well in the TAS environment.

The overall architecture of TRISTAN-TAS as currently implemented is shown in Figure 4. TRISTAN is currently implemented on a computing infrastructure composed of a client-server system running NT4.0 and a Sun Ultra 2 workstation running Solaris 2.6. During operation, real-time transaction records are inserted into the Oracle database for comparison with the stored threshold rules that are generated by the rule generator. Rules are computed on the Ultra 2 workstation which analyses the daily transaction feeds. The anomaly detection engines retrieved data from the Oracle database for anomaly detection.

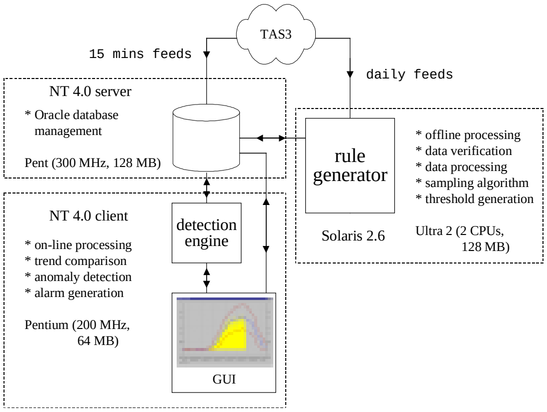


Figure 4: TRISTAN architecture.

The TRISTAN graphic user interface (GUI) consists of (1) a control panel, (2) an alarm log, and (3) a traffic visualizer. The control panel displays information concerning performance of the service classes and the numerical values of their respective dynamic thresholds, in addition to providing a debugging window for database-SQL programming. The alarm log summarizes and classifies the detected anomalies and their severity. The traffic visualizer presents a graphical representation of the service-class based traffic intensities in real-time.

The TRISTAN GUI enables real-time visualization of service-class based traffic intensities. The control panel and traffic visualizer of the TRISTAN GUI is shown in Figure 5.

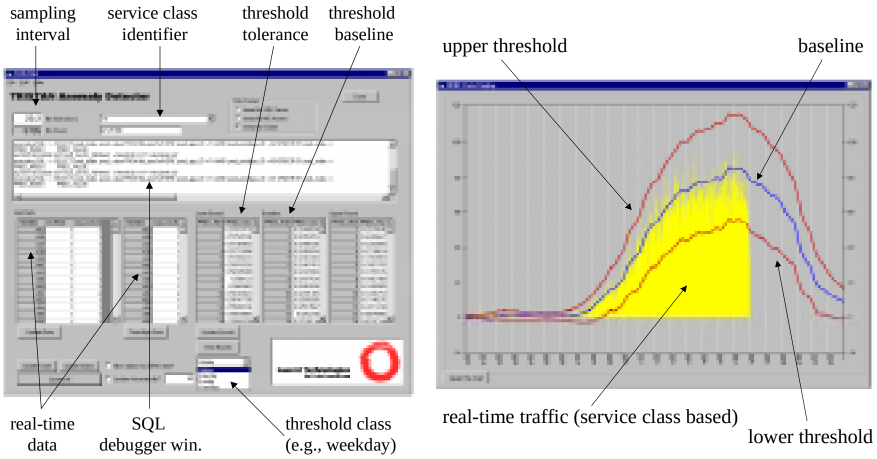


Figure 5: The control console (left) and the alarm log (right) of TRISTAN's GUI.

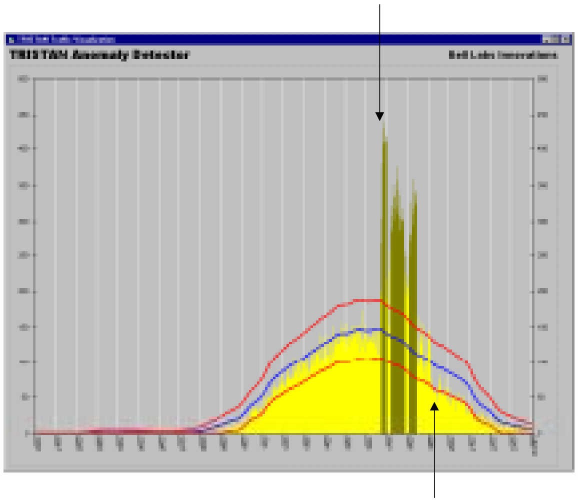
5. Anomaly Detection with TRISTAN: A Real-life Example

One major class of network failures in the AT&T TAS network is due to server failures in the customers' sites (e.g., credit card processing servers). These servers are situated in the customers' premises, and are not monitored by the TAS network management system. These server failures can not be detected directly (e.g., through server alarms or traps), but must instead be inferred from their negative impact on the performance of the TAS network proper. Moreover, these server failures have led to major outages in TAS services, due mainly to the excessive buildup of real and virtual circuits within the TAS network. Thus, server failures originated from one or more service class can potentially degrade the performance and availability of

TAS. On a certain holiday day in late 1997, the transaction servers of a credit card processing service-class crashed in the afternoon, and persisted in a “down” state for more than 2 hours before being completely rebooted. As time progressed, this service class started to tie up network resources unfairly (physical circuits in the telecom network, and virtual circuits in the data network). TRISTAN detected a service-class anomaly in the first 15 minutes (as limited by TAS’s rate of delivering real-time transaction records) of the server failures, and identified the “guilty” service class through mapping the anomaly to the TRISTAN log. This fault escaped detection by conventional alarm-based network management system. Thus, this example illustrates that TRISTAN can proactively and adaptively detect TAS network faults.

The visualization instance of the incident described above is shown in Figure 6. In the figure, the traffic intensity of service class “A” started to deviate significantly from the predicted dynamic thresholds as the server crashed, signifying that network resources were unfairly dominated by this “guilty” service class. This failure, as shown in Figure 6, persisted for about 2 hours and 15 minutes. TRISTAN detected and diagnosed this network fault in the first 15 minutes of the server failure.

onset of server failures



problem resolution

Figure 6. Traffic intensities of a faulty service class (server failures). TRISTAN detected this fault within the first 15 minutes.

6. Conclusions

A software system (TRISTAN), which implements a set of network anomaly detection algorithms, performs automatic and adaptive network anomaly detection for the AT&T Transaction Access Service network is described and analyzed in this paper. It is demonstrated that TRISTAN can detect network anomalies that elude traditional (alarm-based) network management system. Specifically,

1. The AT&T TAS network is a high-volume and multi-service-class hybrid telecom-data wide area network. This provides high statistics for testing and developing a network anomaly detection system. The presence of non-trivial network faults, both inside and outside the network proper (e.g. non-managed server failures), further enables the testing and development of TRISTAN.
2. TRISTAN is demonstrated to be able to detect non-trivial network anomalies originated both within and outside the TAS network. This is attributed to: (1) an adaptive and service-class enabled sampler that highlights TAS transactions that are potentially anomalous, (2) a threshold generator that builds dynamic performance thresholds for TAS service classes, and (3) a detector that performance anomaly detection on-line.

Ongoing and future research and development items include:

1. Combining the TRISTAN anomaly detector with the alarming system (also an anomaly detector) of the data network in AT&T TAS to enable more accurate and efficient detection of a wider range of TAS faults, including the low-level physical faults in the TAS network.
2. Implementing and testing different thresholding schemes in the anomaly detection system to optimize the fault detection probability of TRISTAN.
3. Extending and field-testing the anomaly detection techniques to a wider range of networking environments, such as IP and wireless wide area networks, where service performance degradations and faults can severely impact quality-of-service (QoS) and network availability.

References

[1] A.A. Lazar, W. Wang, R. Deng, “Models and Algorithms for Network Fault Detection and Identification: A Review,” *ICC Singapore*, Nov. 1992.

[2] G. Parulkar, D. Schmidt, E. Kraemer, J. Turner, and A. Kantawala, “An Architecture for Monitoring, Visualization, and Control of Gigabit Networks,” *IEEE Networks*, p.34, Sept/Oct, 1997.

[3] I. Katzela and M. Schwartz, “Schemes for Fault Identification in Communication Networks,” *IEEE/ACM Trans. Networking*, Vol. 3(6), p.753, Dec, 1995.

[4] C. Wang and M. Schwartz, “Fault Diagnosis of Network Connectivity Problems by Probabilistic Reasoning,” *Network Management and Control Volume Two* (Ed. I.T. Frisch, M. Malek, and S.S. Panwar), , p.67, (Plenum Press 1994).

[5] N. Dawes, J. Altoft, and B. Pagurek, “Network Diagnosis by Reasoning in Uncertain Nested Evidence Spaces,” *IEEE Transactions on Communications*, Vol. 43, p.466, 1995.

[6] C. Cortes, L.D. Jackel, W. Chiang, “Limits on Learning Machine Accuracy Imposed by Data Quality,” *Proceedings of NIPS94 - Neural Information Processing Systems: Natural and Synthetic Pagination*, p. 239, (MIT Press 1994).

[7] R.M. Cox, “Detecting Lost Billing Records Using Kalman Filters,” *AT&T Labs Preprint* (submitted), Oct. 1997.

[8] F.E. Feather, D. Siewiorek, and R. Maxion, “Fault Detection in an Ethernet Using Anomaly Signature Matching,” *ACM SIGCOMM'93*, 23(4), 1993.

[9] R. Maxion and F.E. Feather, “A Case Study of Ethernet Anomalies in a Distributed Computing Environment,” *IEEE Transactions on Reliability*, 39(4), Oct 1990.

[10] C. Hood and C. Ji, “Proactive Network Fault Detection,” *IEEE Trans. Reliability*, Vol. 46, No. 3, p.333, 1997.

[11] C. Hood and C. Ji, “Proactive Network Fault Detection,” *Proceeding IEEE INFOCOM*, 1997.

[12] S. Yemini, S. Kliger, E. Mozes, Y. Yemini, and D. Ohsie, “High Speed and Robust Event Corrrelation,” *IEEE Communication Magazine*, May 1996.

[13] S. Katker and M. Paterok, "Fault Isolation and Event Correlation for Integrated Fault Management," *Proceedings of the Fifth IFIP/IEEE International Symposium on Integrated Network Management*, p. 583, 1997.

[14] G. Jakobson and M.D. Weissman, "Alarm Correlation," *IEEE Network*, p. 52, Nov 1993.

[15] M.Z. Hasan, F.E. Feather, L.L. Ho, B. Sugla, "The Conceptual and Software Frameworks of Network Management Event Correlation and Filtering Systems," *Bell Labs Technical Memorandum*, 1998 (to be submitted for external publication).

[16] B.A. Huberman and R.M. Lukose, "Social Dilemmas and Internet Congestion," *Science*, Vol. 277, p. 535, July 1997.

[17] L.L. Ho, F.E. Feather, and M.Z. Hasan, "Automatic Service Anomaly Detection in Wide Area Networks," *Bell Labs Technical Memorandum*, 1998 (an expanded version to be submitted).

[18] L.L. Ho, F.E. Feather, and D.J. Cavuto, "TRISTAN: An Adaptive Network/Service Anomaly Detector for Transaction-Oriented Wide Area Networks," *Proceedings of the 1998 Lucent Network Management Symposium*, June 1998.

[19] E.E. Jerabek, "Transaction Access Service III," *AT&T Technical Services Description (AT&T Proprietary)*, September, 1996.