Darkstar: Using Exploratory Data Mining to Raise the Bar on Network Reliability and Performance

Charles R. Kalmanek, Zihui Ge, Seungjoon Lee, Carsten Lund, Dan Pei,
Joseph Seidel, Jacobus van der Merwe, Jennifer Yates
Networking and Services Research
AT&T Labs
Florham Park, NJ
(crk, gezihui, slee, lund, peidan, spence, kobus, jyates)@research.att.com

Abstract—**Networks have become a critical infrastructure, and performance requirements for network-based applications are becoming increasingly stringent. This trend challenges service providers to raise the bar on the performance and reliability of network services. To achieve this, new network and service management systems are needed that enable providers to continually improve performance, identify issues that are flying under the radar of network operations, and troubleshoot complex issues. This paper presents the Darkstar system, which allows analysts to address these challenges using exploratory data mining and sophisticated correlation tools. We present an overview of key applications that are built on top of the Darkstar system to illustrate the power of the approach.**

Keywords-alarm correlation, exploratory data mining, network management, service management.

I. INTRODUCTION

Networks and network-based services have become an essential part of our lives, the economy, and society. As a result, service disruptions can have a significant impact on people's lives and businesses. To address this, service providers have made significant investments to deliver reliable services - the result being that today's networks are extremely reliable. Network designs incorporate redundant network elements, diverse routing of network facilities, and restoration mechanisms to rapidly recover from failures. Application layer services are designed with redundant configurations that operate in active-active or active-standby mode, to minimize service outages. As performance requirements have become more stringent over time, steps have been taken to reduce the frequency and duration of network and service disruptions. For example, service providers have been deploying the Multi-Protocol Label Switching (MPLS) Fast Re-Route capability, which significantly reduces the time to restore services after a link or node failure as compared with traditional routing protocol re-convergence.

These successes in improved network operations have led to even more demanding expectations for network reliability and performance. Network-based services are mission critical for many businesses. Convergence has caused applications that used to be delivered via dedicated networks to migrate to the Internet Protocol. For example, emergency responders rely on IP-based network services to be able to communicate with each other and to access information in critical situations. In the early days of the Internet, the typical applications were non-real-time applications where packet retransmission and application layer retries would hide underlying transient network disruptions. Today, applications such as online stock trading, online gaming, VoIP and video are much more sensitive to small perturbations in the network than the early uses of the Internet Protocol.

At the same time that application demands have become more stringent, service management in converged networks has become more complex. Large network and service infrastructures are inherently complex, and are becoming more so as their scale increases and as the number of features grows. There are typically thousands or tens of thousands of elements and software systems that must work together for a service to operate correctly. Moreover, network-based services typically must operate continuously and yet are subject to constant change. The elements and systems have complex dependencies that are often not well understood. Since most components of service delivery involve software, they contain software bugs and are subject to configuration errors that can trigger unexpected behavior.

Building and maintaining reliable, high performance IP/MPLS networks and services involves many complex activities. Events must be reliably and rapidly *detected*, *investigated* and *resolved*. These events may be major outages, as are the focus of traditional event management systems. Or they could be performance degradations (e.g., packet loss) or short duration outages ("glitches") which are impacting customer performance. Short duration outages typically disappear even before a network operator can react to them. If the event is a one-off, then there may be little point in investigating further. However, there is a class of network events consisting of recurring conditions, which keep re-appearing and may even get worse over time before eventually turning into serious hard failures. Even when a recurring condition does not result in a hard failure, it can add up to cause significant performance degradation to customers over time. These events must also be investigated and mitigated.

Network and service management have traditionally focused on troubleshooting failures that persist over a long time such as link failures, fiber cuts, and router, server or application failures. In order to troubleshoot such an event, operations staff are trained to gather additional data in real-time, e.g., by running diagnostic tests, or logging into network elements and running commands. However, as customer's applications have become increasingly dependent on and sensitive to network performance, such approaches are, although still necessary, not sufficient for managing overall network health. Service providers need to be constantly vigilant about identifying new signatures which can be used to detect issues, and in trending and analyzing network behavior to understand issues which may have been flying under the radar but which may be causing unnecessary service impact. This requires additional tools than those developed for traditional network management. These new tools must be designed to support exploratory data mining (EDM) – focused on enabling data analysts to better understand network behavior, perform post-mortem analyses, do performance trending, and develop data 'signatures' that can be codified in mainstream tools that are used for reactive or predictive network management.

We view the use of exploratory data mining (EDM) in service management as the next research frontier: one that poses interesting challenges to both systems and analytic researchers. EDM can be characterized by a large volume of data, often of variable quality, and by the fact that, at the outset of a study, the analyst does not always know what he is looking for. Constructing a solid understanding of what was happening (for post-mortem analysis) or what is happening (for ongoing events) and the impact of a given network event or set of events typically involves collating information obtained from a range of different measurements. A single data source rarely gives an adequate perspective – multiple data sources are typically fundamental to the analysis. Both human-guided data exploration and tools-based data analysis thus typically require correlation across multiple data sets. However, telemetry data that may be useful for troubleshooting and analysis may be locked up in network management "silos" due to vendor or organizational constraints. This makes correlation across different data sources extremely challenging and typically a very manual process. The data that exists may use inconsistent naming conventions and time zones, adding to the complexity of correlating across sources. These realities motivate the requirements for the infrastructure needed to support EDM.

In this paper we present such an EDM infrastructure, known as Darkstar, created within AT&T Labs - Research. We present the Darkstar architecture and describe example applications that are enabled by the continually expanding set of data sources it contains.

II. DARKSTAR: INFRASTRUCTURE FOR EXPLORATORY DATA MINING

A. *Data Mining as a Methodology*

A large network or service infrastructure generates an enormous amount of data that is potentially useful for understanding network behavior. Data is gathered from network elements across different network layers – in an IP-based service, this includes (but is not limited to) the lower layer network elements, generic IP-layer network elements (e.g., routers) and service specific network elements (e.g., VoIP equipment or servers). In addition to data from network elements, measurement data generated by specialized passive and active instrumentation is also collected [1]. Data collected includes network element configuration and topology information, logs of network elements faults (up/down events reported via SNMP traps), network element events (syslogs), and control plane events and state information. Performance data is also collected, both locally from the network elements (MIB counters), and across the network / service via active and passive measurements of service health and performance. Traffic measurements (SNMP, Netflow) and workflow logs (TACACS) are also available.

Since our approach to improving network performance and reliability is based on data analysis, the core of our solution is a data repository optimized for analysis and correlation across diverse data sources. We achieve this using a single, logical database that collates massive amounts of data. We collect data from each of the above mentioned data sets and across multiple different networks, covering IP/MPLS enterprise and consumer technologies, IPTV and mobility data services.

B. *Darkstar Architecture*

In order to support exploratory data mining (EDM), we have developed a data warehouse and data mining infrastructure known as *Darkstar*. Darkstar's primary function is a data integration platform, combining data that originates from different sources into a single logical data repository that can be accessed by applications using a single, unified interface. The infrastructure contains both real-time and historical data.

The Darkstar architecture is depicted in Figure 1. Darkstar conceptually maps into a four layer architecture. The bottom two layers consist of the *data management* and *database systems* and collectively form a *data warehouse*. Commonly used tools which enable user applications in data exploration are provided using a *reusable toolset* layer. Finally, an *application* layer houses a variety of EDM applications.

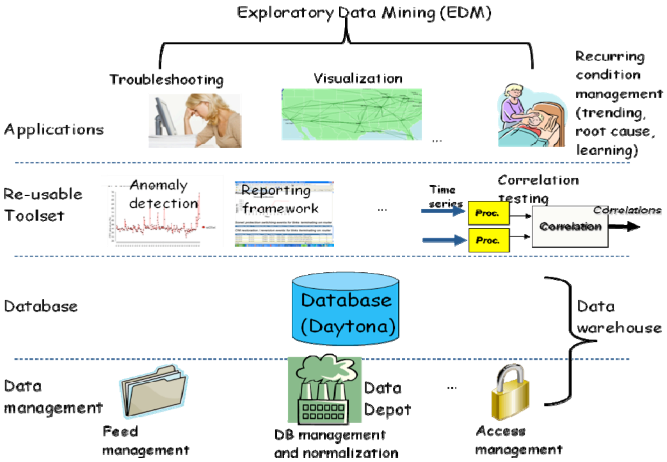


Figure 1: Darkstar architecture.

The heart of the Darkstar infrastructure is its unified database, where all of the data is collated and normalized to facilitate analysis and cross-data set correlation. We chose Daytona™ [2] as the database technology on which to base our infrastructure. Daytona is a flexible Data Base Management System (DBMS) developed within AT&T and used for a large number of data management tasks. Daytona was selected for this application because it has been optimized to be able to ingest massive amounts of data. Daytona also uses data compression, which can reduce the amount of disk needed to store data by as much as an order of magnitude over conventional database technologies.

Data is collected from a wide variety of sources and organizations, most of whom have their own idiosyncrasies concerning data collection. Data is thus obtained by a range of different mechanisms – some of it streamed in real-time (e.g., router syslogs and workflow logs), whilst other data is collected at regular intervals from remote databases or repositories - either pushed to our servers by remote applications or pulled by us from the remote locations.

To scale this effort, Darkstar utilizes tools that simplify and automate data feed management. Feed management can be a complex, labor intensive task, that requires an interface to many different systems, operated by different organizations, with different formats, data transfer protocols, etc. A small research organization cannot afford to dedicate a large amount of staff hours to this task, thus we made the decision early on that automation was critical.

Each feed is defined by metadata that is provisioned when the feed is initially on-boarded into the system. The data feed management module collects the data using a pre-defined mechanism (where we are pulling the data), audits feeds according to prescribed rules, retries if the feed is down or the expected data is incomplete (again, where we are controlling the data feed), and alarms if it is unable to retrieve the data.

Once the data is available in flat files on our server(s), we use a data warehousing system, called DataDepot [3], to automate the database management. DataDepot identifies newly arrived files that are to be loaded, flexibly integrates external scripts to perform data normalization and pre-processing on the data as it is loaded, loads the data into the database, and times out and/or archives old data from the database. Together, the feed management software and DataDepot have allowed Darkstar to scale to nearly one hundred data feeds, and hundreds of database tables containing very diverse network-related data, with new data feeds and tables being added each month. This was accomplished with no dedicated systems admin staff and no full time database administrator. Darkstar also implements a strong access control mechanism, with permissions at the level of individual user and individual data source. This allows a great deal of flexibility, while at the same time ensuring that sensitive data sets are restricted to users that have permission to access them. Since we host a large number of student interns and research visitors, flexible, fine grained access control is a requirement.

As indicated, one of our primary goals was to enable scalable correlation across large numbers of data sets. Data normalization is an essential building-block to achieve this goal. Thus, as data is loaded into our database, timestamps are normalized to a common time zone (Universal Coordinated Time – UTC). Where different systems utilize different naming conventions for a single element like a router, data is normalized to a common key, while retaining the original data element as a secondary key to ensure that data is not lost in the normalization process. For example, IP addresses are converted to router (interface) names, so that data sources which natively refer to routers (interfaces) using IP addresses can be readily correlated with data sources which use router (interface) names without application-specific translations. Complex mappings from layer one facility names all the way to router interface names are also carried out – often involving multiple levels of indirection. This is necessary because lower layer (transport) networks utilize very different naming conventions for describing network circuits compared with routers – the normalization provides a "translator" function so that a given layer one circuit can be equated to the link between a pair of IP routers.

While the overall infrastructure - from data feed to normalized database - represents a fairly significant prototyping effort for our team, the investment is amortized across all of the applications that are built on top of the platform, rather than requiring every researcher or application developer to 'reinvent' data management tools each time.

Above our Darkstar database, we have established a number of common tools that can be used by researchers or developers that are interested in analyzing data or producing reports. To facilitate rapid report generation, we created a set of libraries that utilize the Business Intelligence and Reporting Tools (BIRT) framework [4], which is part of the ECLIPSE open source community. We also created a library of anomaly detection and correlation tools that can be used by application creators.

Applications reside above the tools layer. Since Darkstar is primarily a research platform today, most Darkstar applications are created by researchers, though many of the applications are used by tier 2, 3 and 4 operations analysts[1] to support both historical and real-time data mining applications. Network engineering teams have also recently started developing applications above our Darkstar platform.

A wide range of sophisticated applications have been created above the Darkstar infrastructure including applications that support capacity and risk management (Nperf [1]), network troubleshooting and post-mortem analysis, network topology and event visualization (e.g., LiveRac [5] and topology-based views), identification and analysis of new

[1] Operations personnel are typically organized in response *tiers.* Tiers 1 and 2 are the first line of defense, and are responsible for rapid response to issues that arise in the network. Tiers 3 and 4 have a more analytic, less reactive role, and work on more complex problems. Our researchers often work with Tier 3 and 4 operations personnel.

"signatures" for event detection and performance trending. These capabilities rely on the reusable tool sets available within the infrastructure. For example, troubleshooting applications utilize many of the components – including the reporting engine, correlation tools, anomaly detection, automated learning and visualization.

The following sections describe some of our EDM capabilities, and the applications that use them in more detail. Because of space constraints, we have chosen to emphasize applications related to network troubleshooting. Specifically, we discuss basic data browsing tools in Section III. Section IV presents the G-RCA tool that performs Generic Root Cause Analysis and is described in Section IV. Section V presents the Network Wide Information Correlation and Exploration (NICE) [6] application which supports human-guided data exploration by examining large numbers of pair-wise correlations. NICE can be used to create a set of correlation rules that are used in G-RCA. Section VI discusses how all of these tools work together.

III. Applications to Enable Data Exploration For Network Troubleshooting

One of the challenges in troubleshooting network events is how to rapidly identify the information of interest within the large volume of available data. Troubleshooting is typically performed under immense time pressures, particularly if the analysis is critical to repairing customer service. Scale can also be a daunting challenge, particularly when analyzing a large number of very small events (e.g., packet losses or protocol flaps) to identify potential patterns in root causes and opportunities for improvements.

A. Data Browsing

Obtaining an understanding of what is happening during a given network incident typically involves information collated from a wide range of different viewpoints and network monitoring systems. Troubleshooting network events – particularly for post-mortem analysis – has thus typically involved network operators logging into a suite of different tools to manually collate information in a bid to obtain a good understanding of the symptoms and impacts of the event. This can be a time-consuming process, particularly when the number of disparate systems involved is large and the systems use different conventions to describe time and the location of the events. Investigating a single event could traditionally have required simultaneously pulling data from, say, 5 different systems, mapping IP addresses to router names and/or interface names and/or mapping router interface names down to layer one circuit names so that relevant layer one events can be identified. Examining a single event can take hours, even in simply pulling the relevant data together. If multiple organizations are involved, it can take much longer.

The simplest Darkstar applications simply expose the relevant network data in an integrated fashion to users, to enable faster troubleshooting of network events. Temporal and spatial filtering are used to limit the data being reported to the time frame and network elements or location of interest. Data normalization avoids having operations personnel log into numerous different systems and manually execute complicated mappings between IP addresses and interface names and across layers to identify relevant events. Instead, a single integrated report collates all of the relevant network data at a user's finger tips. We have two primary variants on this – one which enables a user to examine events on individual routers or sets of routers (RouterMiner) and one which looks at events along a path between two locations in a network (PathMiner). Although simple, this capability is immensely powerful in practice. In the rest of this section, we provide a concrete example using PathMiner, which helps reduce operations workload required to troubleshoot a particular class of customer impacting network events.

B. Case Study: PathMiner

Suppose that a large enterprise customer that has purchased a Virtual Private Network (VPN) service contacts the account team with a report that they have experienced brief service "glitches" between specific locations at a number of times over the past two months. The customer provides a list of the affected locations and the specific times when the events are believed to have occurred. The customer account team contacts the service provider's operations team, which is asked to investigate the events to determine the root cause.

Of course, the operations analyst could quickly run a sequence of ping tests and traceroutes to see how the service is performing now and how traffic is currently routed. Unfortunately, since the events occurred in the past, this isn't very useful. Even the path between the ports of interest may not be the same as it was when the events occurred: the events are over. The traditional approach to this problem is for the analyst that is assigned this problem to start an assessment, working with teams that are responsible for all of the relevant network domains, such as layer one operations, the edge router team, and the backbone team. Each of the team members looks through log files on or about the time of the specific events, in order to find significant events that might explain the issue. This manual analysis is a time consuming and error prone process: it can take hours or even days to weeks if multiple individuals are involved in troubleshooting a single event. Data silos make this type of analysis even harder.

PathMiner rapidly identifies and pulls together in a web report all log and performance data relevant to the network elements along the path providing the customer service. To achieve this, PathMiner makes use of the unified view of historical data contained in the Darkstar database, including historical routing information. To use PathMiner, an operator queries PathMiner by specifying a date, time window, the end points of interest (namely the provider edge routers to which the customer is connected), and additional filters to narrow the search response. PathMiner infers the actual network path(s) between the affected endpoints at the time when the problem occurred, including accounting for path re-routing events. PathMiner then collects network event data for each of the routers along each of the different paths during the time interval of interest. By simply presenting all of this data to the Operations personnel in a unified framework, it allows them to focus their analysis to rapidly identify the relevant network events which would have caused the customer concern. This has been demonstrated to reduce investigation time for a network event from potentially up to weeks when multiple teams are involved to a couple of minutes.

The data sets presented by PathMiner include workflow (TACACS) logs (indicating commands that were executed via the router's command line interface), syslogs (a time sequence of logs reported by the router), and OSPF routing protocol messaging (events that correspond to changes in the network topology). In addition, PathMiner interfaces with the RouterMiner application, which extracts more detailed information on a per-router basis, including router memory and CPU usage, high link loads and layer one events during the time interval of interest. PathMiner also incorporates loss and delay measurements reported by active probe servers in major points-of-presences (POPs) [7]. These measurements can identify whether end to end measurements executed within the network also observed the issue being queried.

Let's consider an example of a troubleshooting session using PathMiner. Note that the example relates to a network which uses traditional OSPF routing without MPLS FastReRoute or Traffic Engineering. We consider a customer complaint in which the customer reports brief connectivity issues between two given locations at specific times. The network operator uses PathMiner to troubleshoot the incident – specifying the times and locations provided by the customer. Within minutes he (she) has collated all of the relevant logs, and can rapidly hone in on a link which failed, causing a topology change (OSPF reconvergence event). Through discussion with the customer, it becomes apparent that the customer is running a BGP session across the ISP's network – with extremely aggressive timers. The ISP's short OSPF reconvergence event caused the customer's BGP session to timeout. By setting the customer's BGP timers to a less aggressive setting, the customer network can be made less sensitive to occasional backbone topology changes.

Given the knowledge that was gained from this specific customer event, PathMiner can be augmented to suggest possible root causes for this class of customer outage. Instead of simply reporting all logs for a given path and time interval, human "reasoning" could be mimicked to identify the most likely explanation of the observed symptoms – in this case, the reconvergence event. If PathMiner also had access to customer edge (CE) router configurations, it could go further to automatically pinpoint the source of the problem to be aggressive BGP timers combined with the ISP reconvergence event. This reasoning would extend PathMiner from a simple data browser to an expert system that provides an end-to-end customer troubleshooting tool.

IV. Generic Root Cause Analysis (G-RCA)

PathMiner and related data exploration tools within the Darkstar framework are useful for reducing the time required for human exploration of a network event. However, manual investigation simply cannot scale to effectively examine large numbers of network events. In this section, we present the G-RCA (Generic Root Cause Analysis) tool.

G-RCA is designed to automatically diagnose the most likely explanation, or "root cause", of a given *symptom event* by applying *diagnostic rules* to search through vast amounts of available network data. G-RCA is thus an expert system, mimicking human reasoning to reach conclusions in troubleshooting network events. For example, consider the scenario of troubleshooting a network loss condition across a wide area network. If the potential set of events or conditions that can induce packet loss are known, then diagnostic rules can be created and applied in a bid to automatically isolate the underlying cause of any given loss event.

Figure 2 depicts the high-level architecture of G-RCA. G-RCA obtains both the symptom event(s) of interest and the set of diagnostic time series from the Darkstar database. For a given symptom event, G-RCA uses application specific temporal and spatial rules to identify the relevant diagnostic events that relate to this event (i.e., those which "join" with the symptom event being analyzed). Specifically, G-RCA determines *where* and *when* to look for diagnostic events (based on the location and time of the symptom event), and uses application specific rules to identify diagnostic events of interest (i.e., these define *what* type of events to look for the given application). Once these diagnostic events are identified, G-RCA then applies reasoning logic (rules) to holistically examine all of the different diagnostic events observed for the given symptom (the "joins") to identify the most likely explanation(s) (diagnosis) of the symptom event.

While G-RCA is conceptually simple and appears to be an instance of a traditional rule-based correlation engine, the key challenges in designing the system for large-scale networks and applications lie in capturing the expert knowledge to form the G-RCA rules – both in identifying the diagnostic events and then in reasoning about these to localize the issue. For example, to diagnose packet loss across a network, G-RCA would start by focusing on the route between the two end points between which the loss is being observed during the time interval of interest. While the temporal information might be relatively easily derived from domain knowledge, identifying the location is more challenging. For end to end packet loss, this requires reconstructing the route between the two end points, a capability that also supports PathMiner. All routers, interfaces, line cards and lower layer facilities associated with the route must then be identified, so that G-RCA can search for events associated with them. In this example, the dependency relationship is actually dynamic – it can change as the network topology changes over time. G-RCA incorporates a comprehensive set of spatial models and a broad range of conversion utilities that map from one type of spatial location quantifier (e.g., IP address) to other types (e.g., router and interface name, layer one identifiers, etc.), using available information about the network topology, configuration, protocol and cross-layer dependencies.

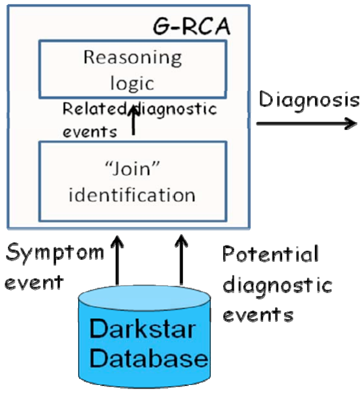


Figure 2. High-level G-RCA architecture.

Once G-RCA knows where and when to look for events, it needs to identify what to look for. Given a symptom event to troubleshoot, a human operator would go through a list of checks based on his or her past experience and domain knowledge. This is encoded in G-RCA within the set of application specific rules that define the diagnostic event types of interest.

Once G-RCA has identified all of the relevant diagnostic events associated with a given symptom event, it can mimic human reasoning to evaluate all of the evidence gathered to derive a concluding diagnosis. The logic for this reasoning is based on application specific rules. For example, if in diagnosing a packet loss event across a network G-RCA identifies diagnostic events denoting a link failure, topology change, interface packet losses and link congestion, the G-RCA reasoning would conclude that there was a failure in the network on a specific link, which caused traffic to re-route triggering link congestion.

The reasoning implemented within G-RCA has a few advanced capabilities that complement the application specific rule reasoning. For example, instead of diagnosing each symptom event in isolation, G-RCA uses machine learning and inference techniques to associate potentially related symptom events together so that it can identify a common root cause that may induce multiple symptom events. G-RCA also includes inference logic to deal with situations in which data that may contain the direct diagnosing evidence is missing, which is undesirable yet unavoidable in any real data collection system. For example, G-RCA can identify likely high load conditions even in the event of missing data by extrapolating from load measurements events either side of the missing measurements.

While G-RCA is ideal for real-time troubleshooting tasks, such as investigating an ongoing loss event that may currently be impacting customers, it can also be effectively used over larger numbers of events to *trend* the root causes of a type of symptom event in the network, such as packet loss events. Such an analysis identifies how symptom events are distributed across underlying "root causes". The analysis is designed to highlight those network impairments which have the biggest impact on the symptom events; these are likely to be the most effective areas to focus operator energy on for providing effective network improvements. For example, if such an analysis were to reveal that a large percentage of network loss was induced by congestion events in a given region, then efforts could be focused on increasing the available network capacity in the region to reduce the frequency and magnitude of such events. If instead the analysis revealed significant loss due to unexpectedly slow reconvergence (re-routing) events, then efforts should be focused on identifying why such events are occurring and how to eliminate or reduce their impact in the future.

We have applied G-RCA to a number of different applications focused on both troubleshooting individual network events and in trending longer term behavior. For example, G-RCA was applied to create a Service Quality Management (SQM) tool for analyzing service impairments observed in an ISP-owned Content Delivery Network (CDN). A CDN is responsible for delivering content (e.g., web pages) that is hosted on CDN servers to end users (consumers).

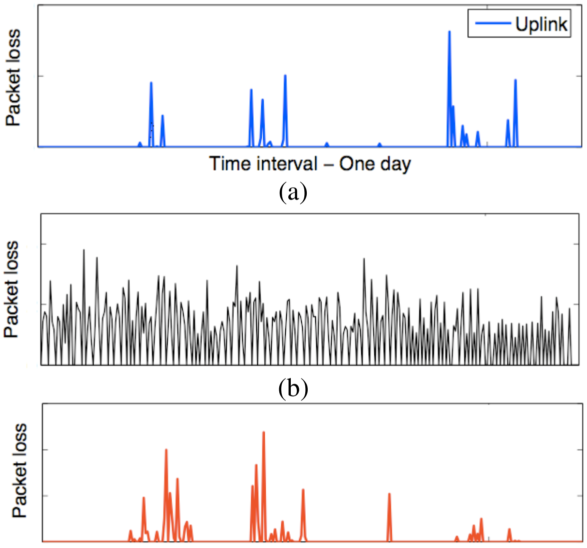
Monitoring tools such as Keynote [8] are used to monitor the performance of the CDN service, and to identify service performance impairments. These service impairments represent our G-RCA symptom events. A given service impairment event is then correlated against known network events and conditions in a bid to identify those which best explain the observed symptom. Of course, SQM applications are typically characterized by the diverse number of network elements involved in providing the service; diagnosis must thus be aware of these relationships and obtain events from all of the relevant network elements. In this case, G-RCA must be aware of the network and service elements involved between the Keynote agent (source) and the CDN server (destination). G-RCA then correlates a given Keynote performance event against diagnostic events that are inferred from within the ISP's infrastructure and beyond. For example, G-RCA considers BGP route changes between a given CDN server and a Keynote agent, intra-domain routing changes that may affect the traffic, congestion on the CDN server or any link along the service path, and packet errors detected along the path as possible diagnostic events.

We initially validated our CDN SQM application using several historical events that had already been diagnosed by the CDN operator. The CDN operators are now using G-RCA to troubleshoot live network events. For example, in one recent event, G-RCA successfully localized an observed service degradation as being the result of traffic being re-routed in response to a peering link failure. The longer path taken by the traffic after the re-route had caused a degradation in service throughput to the Keynote servers.

V. NETWORK-WIDE INFORMATION CORRELATION AND EXPLORATION

One of the challenges with rules based correlation, such as used in G-RCA, is specifying the set of correlations to perform (i.e., the rules). Many of these potential causes of a symptom event are identifiable by domain knowledge. However, domain knowledge is often distributed across a diverse set of individuals and can be surprisingly challenging to capture. Even when basic knowledge is available, translating that to specific logs can still remain an elusive problem. There are an extremely large number of different types of logs – particularly router syslogs – and interpreting each of these messages to determine which are relevant is an excessively painful process, to say the least. Finally, in some cases, network behavior is not consistent with the expectations of the domain experts – either because the domain expert needs to update his/her domain knowledge, or because the network is not behaving as expected (e.g., due to hardware or software bugs). Thus, it is critical to scaling and accurately defining correlation rules that we do not rely purely on domain knowledge, but also use the data to guide us.

The Network-wide Information Correlation and Exploration (NICE) infrastructure [6] was designed to *automatically learn* about potential root causes and impacts associated with a given symptom time series. NICE starts with a set of symptom events, which it aggregates into a *symptom time series*. It then systematically constructs other time series, which represent a broad range of *diagnosing events*. This is illustrated in Figure 3. Figure 3(a) (the upper curve) represents an example symptom time series. This particular example corresponds to a series of packet losses observed on a router *uplink*, a link which connects this router to another router in the ISP backbone. Figures 3(b) and 3(c) represent two example diagnostic time series corresponding to packet losses on customer-facing interfaces on the same router.



(a)

(b)

(c)

Figure 3. Example time series input to NICE. (a) depicts the symptom series; (b) and (c) depict diagnostic series.

NICE takes each diagnostic time series in turn and feeds it into the NICE engine, along with the symptom time series, to test the statistical correlation between them. The high level NICE engine architecture is illustrated in Figure 4. The output of the NICE engine is a correlation score, and a series of "joins" (co-occurrences) between the two time series [6]. If the correlation score is sufficiently high, then we conclude that the two time series are statistically correlated. Thus, by iterating across all diagnostic time series, NICE identifies those diagnostic event series that have *statistically significant correlations* with the symptom series, and presents these results to a human analyst for further analysis. The idea is that these statistically correlated time series are likely root causes or impacts of the symptom series, and are thus worthy of detailed investigation. Note that NICE is particularly focused on recurring conditions, and not on one-off events. Thus, we are looking for the most likely (frequently occurring) explanations of recurring conditions.

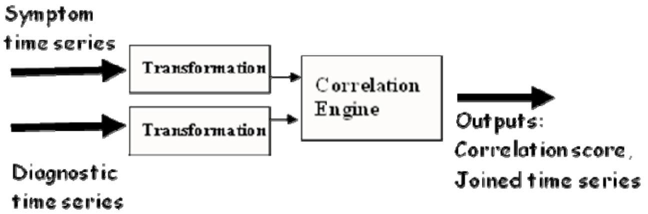


Figure 4: High-level architecture of NICE core engine.

NICE is based on a key assumption that evidence about the root cause of a recurring network condition can be discovered in other network data sources. The intuition behind the statistical correlation analysis is that a single co-occurrence may be a mere coincidence or a one-time incident. However, if co-occurrences keep happening and are statistically correlated, it likely reveals some sort of causal relationship between the time series.

Returning to the example in Figure 3, in applying NICE the goal is to identify those customer-facing interfaces whose packet losses are statistically correlated with the packet losses on the router uplink, and may thus be related to the underlying cause of the recurring uplink losses. The time series in Figure 3(b) represents an interface which is being constantly overloaded by a customer. Although there are many overlaps (joins) when this customer interface and the uplink are both observing packet losses, they are not statistically correlated. In contrast, the packet loss time series in Figure 3(c) does demonstrate statistically significant correlation with the uplink losses. This correlation likely provides a critical insight in diagnosing the uplink recurring packet losses.

There were a number of challenges associated with creating the NICE system [6] - the biggest one being scale. There is an almost limitless number of different time series that can be created from the wealth of available network data sets, and these can be defined at a tremendous number of different spatial locations with potential correlations across all different spatial locations. Simply testing "everything" results in far too many correlations to examine to identify those which may be of interest. Human-derived domain knowledge is fundamental here to identify what is "interesting" versus what is "not" – this does not scale with immense numbers of different time series combinations. Thus, we need to limit the number of correlation tests performed – both by limiting the time series of interest, and the spatial combinations examined. However, the input time series must be defined very carefully – without creating the "right" time series, a critical correlation may be lost. NICE tackles this problem by using *domain knowledge guided data mining*. For example, NICE uses a spatial model similar to that represented in G-RCA, in which network topology, configuration, and protocol and cross-layer dependencies are captured. Thus, correlation tests can be focused in on the area of interest.

Let's return to the example of identifying time series which are statistically correlated with packet losses across a large ISP network. We could test the correlation across all sorts of different time series – for example, in an extreme case we could create time series of each different type of syslog message, workflow log and performance impairment on each different router in the network. However, doing so creates an immense number of time series. Instead, we are most likely interested in time series associated with routers along the route between the two end points associated with a given loss event. We thus focus our analysis there by considering only these time series. By using the spatial model, NICE significantly reduces the number of time series tested, the number of correlation results which need to be inspected afterwards, and the number of potential "false" alarms.

When looking for correlated diagnosing event series, NICE filters events according to domain knowledge specification, so that correlation tests are within the scope of potential impact (e.g., requiring diagnosing events be at the same interface, at the same router, or along the routing path of the observed symptom events). This in some cases can reduce the number of diagnosing event series by several orders of magnitude. A combination of domain knowledge and simple blind creation of time series (e.g., all unique types of syslog messages) is used in guiding the construction of the time series.

NICE incorporates a number of features that aid in analyzing network time series data and that increase its usability [6]. For example, NICE incorporates a clustering algorithm that automatically groups diagnosing events according to their co-occurrence pattern, making it easy to diagnose for the chain of cause and impact, and root cause signatures. NICE applies configurable "padding margins" when computing statistical correlations, allowing data from diverse sources to be correlated even when they use imprecise timestamps or are not perfectly aligned. NICE also uses a circular permutation based significance test for statistical correlation, making it robust to strong auto-correlation structure often exhibited in network event series (for example, packet errors over a link are more likely to occur in bursts than uniformly at random). NICE also supports aggregation such that event series across routers that have the same role (e.g., core routers, edge routers) or the same type (e.g., by vendor, IOS version) can be analyzed collectively. These features are simple, yet powerful building blocks that are necessary when handling network time series data.

The NICE framework is being applied to an increasing range of applications across multiple networks. In general, we are interested in diagnosing a single recurring time series of interest – that time series typically being one indicative of either customer impact, or network health. NICE is then used to identify the root causes and/or impacts of such a recurring condition. NICE has recently been applied to analyzing service impairments, network packet loss events, protocol flaps (see Section VI for a more detailed example), and CPU anomalies. NICE output is used for many things – identifying anomalous network behaviors that require further investigation, troubleshooting anomalous behaviors (e.g. analyzing its extent and scope), verifying that conditions have been successfully eliminated once repair actions have been completed and creating rules for G-RCA. For example, NICE identified strong correlations between CPU anomalies and other events that occur on the same router. CPU cycles are a limited resource on routers, and high CPU utilization can put routing protocols at risk, resulting in protocol flaps that impact customers. NICE was used to discover that invocation of certain commands on routers correlated with short-term high router CPU spikes. Insights like this can be used to constrain the use of such commands, and to drive improvements within the router to limit the impact of such commands.

VI. INTEGRATING EDM CAPABILITIES

The Darkstar infrastructure integrates our data browsing, G-RCA and NICE capabilities to provide an integrated data mining infrastructure. All three capabilities dovetail and are often critical to a single application.

G-RCA is used to troubleshoot individual or sets of events – using rules to identify the most likely underlying causes from the mound of available network data. G-RCA can scale and speed troubleshooting – enabling faster reaction and repair for an individual event, and scaling analysis to large numbers of events.

NICE complements domain knowledge to guide the generation of the rules used within G-RCA – reducing the time to create rules by bypassing the often painful acquisition of domain knowledge that is distributed across network experts, and revealing unexpected behaviors which defy domain knowledge and would thus likely have been flying under the radar. NICE is also used in troubleshooting recurring conditions – for example, honing in on the offending technology which may be associated with the unexpected (often erroneous) network behavior.

Finally, data browsing reports such as RouterMiner and PathMiner allow analysts to manually inspect sample events from G-RCA and NICE – confirming and refining signatures revealed by NICE before they are incorporated as rules into G-RCA, and allowing validation and more detailed analysis of events identified by G-RCA. The data browsing reports also enable deeper analysis into events that "drop out" within G-RCA i.e., that G-RCA is unable to classify according to root cause. As further inspection reveals new signatures, these can be incorporated into G-RCA as new rules, and thus the refinement of the G-RCA capabilities continues.

We illustrate the integrated application of the three core capabilities in a recent example worked with network operations. We focus on the connectivity between a customer and an ISP – specifically, between a customer router (CR) and a provider edge router (PER) where the customer uses the Border Gateway Protocol (BGP) routing protocol to share routes with the ISP over an E-BGP session between the CR and the PER. E-BGP is commonly used, for example, in the case where customers are multi-homed. In the event of a failure of the link between the CR and PER, BGP re-routes traffic onto an alternate route.

The physical connectivity between the CR and PER is provided over metropolitan and access networks as illustrated in Figure 5. These networks in turn may be made up of a variety of layer one and layer two technologies (i.e., transport layer networks). These metro/access networks often have built in mechanisms for rapidly and automatically recovering from failures. Thus, in these situations, failure recovery mechanisms may exist at both the upper layer (through BGP re-routing) and the lower layers. It is highly desirable to ensure that failure recovery is not invoked simultaneously at both layers [9]. This is achieved in routers today using timers – the IP routers are configured with a hold off timer designed to allow the lower layer to attempt to recover from an issue first. If the lower layer restores connectivity within the defined timeout (e.g., 150 milliseconds) then the routers do not react. However, if the lower layer fails to recover from the issue within the defined time interval, then the routers will attempt to restore service.

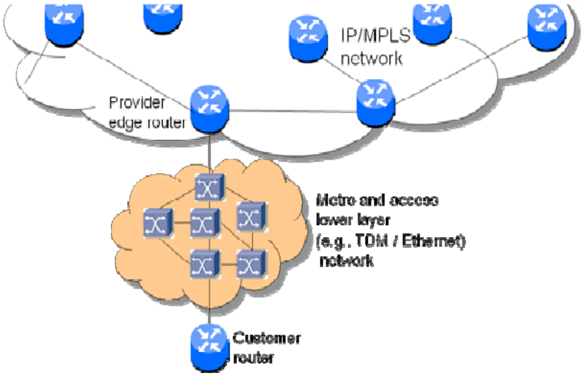


Figure 5. CR to PER connectivity.

The analysis that we will describe here was part of a performance management effort focused on minimizing the number of E-BGP session flaps across the ISP's footprint. In collaboration with network operations and personnel from relevant router vendors, we focused on understanding the frequency and root causes of E-BGP session failures, and examining opportunities for reducing these.

G-RCA formed the heart of this analysis. Through G-RCA, we trended the frequency of E-BGP flaps over time, and identified their root causes. We analyzed over a year's worth of BGP session failures, obtained from router syslog messages stored within the Darkstar warehouse.

Note that one of the challenges in analyzing E-BGP related events is that it relates to interfaces which cross a trust boundary – in this case, one end is owned by the ISP, and the other end is owned by the customer. The CR logs are not typically available for the analysis, especially when customer routers are not managed by the ISP. Thus, our analysis to date only has visibility into the service provider end of the CR-PER link.

Many of the potential causes of BGP session failures were readily available from domain knowledge and were thus directly encoded into G-RCA. For example, router reboots and failures, line card failures and lengthy lower layer (e.g., layer one) outages are all well understood conditions which can cause E-BGP session failures. However, even when the general categories were obvious, it was not always obvious which syslog messages related to these conditions. As a result, NICE was used to capture the specific log messages of interest so that they could be encoded as rules in G-RCA. NICE tests were executed by testing the correlations between E-BGP session failures and the complete set of unique syslog and workflow (TACACS) events. Correlation testing was also executed against SNMP measurements (high CPU, memory, link loads, packet losses) and lower layer network events.

As NICE revealed interesting correlations, we used RouterMiner to delve into the different logs surrounding sample events to validate the precise signatures and rules which should be incorporated into G-RCA. For example, NICE revealed that CPU anomalies and BGP flaps were statistically correlated events. Further examination using RouterMiner to manually inspect sample events revealed that this was due to E-BGP flaps in which large numbers of routes were being exchanged between the ISP and the customer. These flaps and the consequent routing table updates resulted in high CPU conditions. This was thus not a signature that needed to be incorporated into G-RCA and our BGP trending.

Once the relevant rules were in place, G-RCA was used to trend BGP flaps over time, and to characterize their root causes. The vast majority of E-BGP flaps were observed by the PER as being triggered by lower layer events (e.g., physical link failures or flaps). Correlation with ISP data from the ISP's lower layer networks further revealed that most of this activity was coming from outside the ISP's network (most likely customer network related). There were however a wide range of other root causes of E-BGP sessions flaps revealed, including planned maintenance (e.g., router configuration changes on customer interfaces, router reboots), router line card failures (customer-facing failures) and incoming line errors.

One of the most surprising correlations revealed by NICE was between E-BGP flaps and lower layer failure recovery events occurring on a common interface. As discussed earlier, this breaks the principle of layered failure recovery – rapid recovery actions at the lower layer should prevent IP links and BGP session between routers from failing during these events. Thus, domain knowledge would suggest that rapid lower layer failure recovery actions would not be related to (correlated with) E-BGP session flaps. However, NICE correlation testing demonstrated that actual network behavior contradicted this - there was a statistically significant correlation between lower layer events and E-BGP session flaps on the same interface. These occurred more often than could be explained by pure coincidence.

By looking at correlations across different vendor types (lower layer and router), line card types and circuit rates, NICE was integral in identifying the conditions under which the issue occurred, and the hardware involved. The analysis guided troubleshooting efforts, which also required reproduction of the event in the lab and detailed software analysis. As a result of this work and the collaboration between operations, network development staff, and the vendor, the router software was updated by the vendor. Note that this analysis and consequent vendor software repair not only benefited the ISP and its' customers, but also benefited other ISPs and customers using similar hardware – even if they were completely oblivious to the issue.

Once the updated software was deployed in the network, G-RCA and NICE demonstrated that the software repair had indeed had the desired effect – eliminating the unexpected failure mode. This is demonstrated in the output from G-RCA provided in Figure 6 which shows the rate of co-occurrences of E-BGP session flaps and lower layer recovery events on common interfaces. As is clearly evident from this figure, the rate of lower layer recovery triggered E-BGP flaps plummeted to zero after the routers were upgraded.

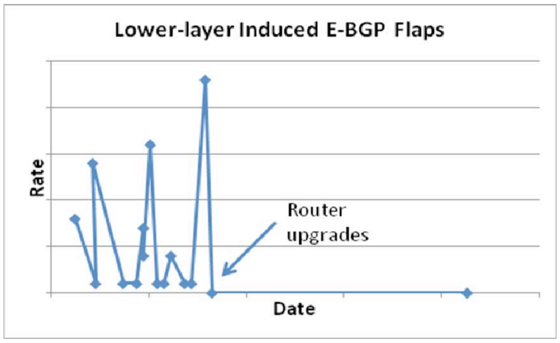


Figure 6. Rate of lower layer recovery actions triggered E-BGP flaps over time.

VII. Conclusions

Exploratory data mining is an essential tool that enables service providers to raise the bar on network and service reliability and performance. This paper describes the Darkstar infrastructure, which provides a highly scalable data integration platform, combining data that originates from different sources into a single data repository that can be accessed by network management applications using a single, unified interface. The Darkstar infrastructure supports a range of data browsing, exploratory data mining and rules-based correlation capabilities that serve as a valuable tool for research and network operations support in a Tier 1 ISP.

We believe that the Darkstar infrastructure represents the next frontier of research in network and service management. Many interesting research challenges remain, including work on network monitoring to augment existing data sources, and further enhancements to include end-to-end service management. Like all large scale data mining problems, the fundamental challenge is one of extracting information from large volumes of data. Data mining is particularly complicated by the scale, complexity and ever changing environment within a large ISP network. Networks are continually evolving, with new features, topology and capacity updates, and technology enhancements. Routers are constantly facing configuration changes, new customers and software upgrades. All of this, in an environment challenged with potential software bugs, hardware failures and external factors which impact traffic loads and routing. On top of this, data mining must contend with data integrity issues – a complex and exciting Research area unto itself. We expect these problems to challenge researchers for many years to come.

Acknowledgments

The authors thank the AT&T network and service operations teams for invaluable collaborations with us, their Research partners, over the years. In particular, we thank Joanne Emmons, Mike Recchia and team for collaborations regarding the cases discussed in this paper.

References

[1] Kalmanek, C, et al. (*eds*), *Guide to Reliable Internet Services and Applications*, Springer London, in press.

[2] R. Greer, "Daytona and the Fourth-Generation Language Cymbal," ACM SIGMOD Conf. on the Management of Data, Vol. 28, No. 2, 1999.

[3] L. Golab, T. Johnson, J. S. Seidel, and V. Shkapenyuk, "Stream Warehousing with Data Depot,“ ACM SIGMOD Conf. on the Management of Data, pp. 847-856, 2009.

[4] http://www.eclipse.org/birt/phoenix

[5] P. McLachlan, T. Munzner, E. Koutsofios and S. North, "LiveRAC: interactive visual exploration of system management time-series data," ACM SIGCHI Conf. on Human Factors, 2008.

[6] A. Mahimkar et al., "Troubleshooting Chronic Conditions in Large IP Networks," ACM Int. Conf. on Emerging Network Experiments and Technologies (CoNEXT) 2008.

[7] L. Ciavottone, A. Morton and G. Ramachandran, "Standardized Active Measurements on a Tier 1 IP Backbone," IEEE Comms. Mag., Vol. 41, June 2003.

[8] http://www.keynote.com

[9] P. Demeester et al., "Resilience in multilayer network," IEEE Comms. Mag., Vol. 37, pp. 70-76, August 1999.