

G-RCA: A Generic Root Cause Analysis Platform for Service Quality Management in Large IP Networks

He Yan¹ Lee Breslau² Zihui Ge² Dan Massey¹ Dan Pei² Jennifer Yates²

¹Colorado State University ²AT&T Labs - Research

ABSTRACT

As IP networks have become the mainstay of an increasingly diverse set of applications ranging from Internet games and streaming videos, to e-commerce and online-banking, and even to mission-critical 911, best effort service is no longer acceptable. This requires a transformation in network management from detecting and replacing individual faulty network elements to managing the service quality as a whole.

In this paper we describe the design and development of a Generic Root Cause Analysis platform (G-RCA) for service quality management (SQM) in large IP networks. G-RCA contains a comprehensive service dependency model that includes network topological and cross-layer relationships, protocol interactions, and control plane dependencies. G-RCA abstracts the RCA process into signature identification for symptom and diagnostic events, temporal and spatial event correlation, and reasoning and inference logic. G-RCA provides a flexible rule specification language that allows operators to quickly customize G-RCA into different RCA tools as new problems need to be investigated. G-RCA is also integrated with the data trending, manual data exploration, and statistical correlation mining capabilities. G-RCA has proven to be a highly effective SQM platform in several different applications and we present results regarding BGP flaps, PIM flaps in Multicast VPN service, and end-to-end throughput drop in CDN service.

1. INTRODUCTION

An increasingly diverse set of applications rely on IP networks. These applications range from entertainment, such as Internet games and streaming videos, to commercial applications, such as e-commerce and online banking, to even some mission-critical applications, such as emergency 911 over VoIP. For many of these applications, best effort service

is no longer an acceptable mode of operation. The networking service offered by Internet Service Providers (ISPs) must maintain ultra-high reliability and performance.

The change of service quality expectations has also transformed the way that ISPs conduct network and performance management. Network operators have traditionally managed networks on the basis of individual network elements, for example, by detecting and replacing faulty network line cards. Today, managing issues related to end-users' service quality has become an increasingly significant part of operators day-to-day work. This work typically involves such tasks as monitoring the loss and delay among different sites of a customer virtual private network (VPN), and identifying ("alarming"), troubleshooting, and fixing any detected performance problems.

As another example, previously network operators primarily focus faults and hard failures. Nowadays, the problems that draw their attention are increasingly transient problems as is often the case with protocol (e.g., BGP) flaps. By their nature, transient problems "repair" themselves; therefore, alarming and responding to each individual problem makes little sense. Instead, examining them – potentially a large number of them – collectively, classifying their root causes and trending them over time can provide operators with critical insights. This information may help in driving the corresponding failure modes out of the network and eventually lead to service improvements.

Moreover, as new services (e.g., multicast VPN), new technologies (e.g., MPLS TE), and new devices (e.g., OC768 line cards) are introduced into ISP networks at a fast rate, ISP operators and hardware vendors often have to learn through experience about service-impacting issues. Should unexpected failure modes or performance impairments occur, network operators need to act quickly to understand the problem, diagnose the root cause(s), and eliminate or mitigate the failure mode to improve service quality.

Given these new challenges, the traditional fault diagnoses and root cause analysis (RCA) systems [10, 12, 26, 27, 3, 4, 2] that network operators have relied on are reaching their limits for the following four reasons.

In particular, first, the narrow view provided by the per network element perspective tends to miss rather complicated **service dependency relationships**. For example, the

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ACM CoNEXT 2010, November 30 – December 3 2010, Philadelphia, USA.

Copyright 2010 ACM 1-4503-0448-1/10/11 ...\$10.00

quality of a VOIP call across the ISP network depends on the status (congestion level, bit error rate, etc.) of the routers and links along the network path carrying the traffic, which is dynamically determined based on the link weights at the time. In another example, the health of a BGP session connecting to a peer router depends on the route processor resource on both routers and the layer-2 line protocol status between them (with complicated timer/protocol interactions), which in turn depends on the condition of the layer-1 (e.g., a SONET ring) network in between. Capturing such service dependency relationships is vital for service quality management (SQM).

Second, Traditional information gathering processes (such as running *traceroute* or invoking *show* command on routers) that are effective at diagnosing problems for large on-going service impacting events are unable to cope with minor and transient service disruptions. RCA for such transient failures should only rely on proactively collected data.

Third, Achieving ultra-high service quality requires going beyond break-and-fix operation and single-event troubleshooting. SQM involves processing and extracting actionable information from a large number of service impacting events in the aggregate. For example, when analyzing sporadic packet losses observed by probing traffic transmitted between different PoPs of an ISP network, one should examine the packet losses over extended period (e.g., a month) and diagnose their root causes. Should link congestion be determined to be the primary root cause, capacity augmentation is needed along the corresponding network path. Alternatively, if packet losses are found to be largely due to intra-domain routing re-convergence, deploying technologies such as MPLS fast reroute becomes a priority.

Finally, in an ever-changing network and service environment, domain knowledge and operator's experience may become insufficient. RCA systems that solely rely on expert input can fail to capture unexpected service dependencies, which unfortunately are hardly unusual in practice due to the various sorts of equipment hardware/software errors or configuration mistakes. An SQM system should allow rapid instantiation of new RCA tasks based on existing expert knowledge as well as flexible data exploration and data mining capability to improve operators' domain knowledge and understanding over time.

In this paper, we introduce our Generic Root Cause Analysis platform (G-RCA) that is designed to bridge the gap between ISP operational needs for service quality management and the state-of-the-art research [10, 12, 26, 27] or commercially available [3, 4, 2] RCA systems. A key feature of G-RCA is a comprehensive service dependency model that includes network topological and cross-layer relationships, protocol interactions, and routing and control plane dependencies. Thus, network operators can look for undesirable network conditions that are potentially **related** to service impacting problems without specifying the details of topology, cross-layer, or routing dependencies. We also ensure that the service dependency relationships in G-RCA can be de-

termined using only data that is proactively collected. For example, network paths are computed from BGP and OSPF route-monitoring data, as opposed to using *traceroute*.

We implemented G-RCA for a tier-1 ISP network. Our design decomposes the RCA process into signature identification for *symptom* and *diagnostic* events, temporal and spatial event correlation, and reasoning and inference logic. Here, symptom events are the type of service problem to be analyzed, and diagnostic events refer to the evidence of a potential root cause taking place. We define a simple yet flexible rule specification language that allows operators to quickly customize G-RCA into different RCA tools as new problems need to be investigated and understood. We integrate into G-RCA data trending, manual data exploration, and statistical correlation mining capabilities that are tailored for service quality management. G-RCA has proven to be a highly effective SQM platform in several different applications. In particular, using the G-RCA platform network operators are able to quickly investigate new service problems, uncover unexpected service impacts, and quantify the scale and trend of different factors contributing to service performance issues using the G-RCA platform.

Our contributions can be summarized as follows.

1. We addressed the need for large scale service quality management in IP networks and services, and designed an abstraction model that hides the complicated service dependency relationship from network operators.
2. We implemented a G-RCA system for an ISP network, making use of data that are already collected from various logging and performance monitoring systems at different network layers in the ISP. We included in our implementation a library of event definitions (for common network problems or failure conditions), network topology and cross-layer conversion utilities, service dependency inference tools, and a library of dependency relationship rules so that building new RCA applications can be as simple as plug-and-play.
3. We collaborated with network operators in applying G-RCA in real-world network operations, and conducted troubleshooting and analysis for a wide range of problems including customer BGP flaps, cross-site VPN PIM session flaps, and CDN service performance issues.
4. We discovered that iteratively applying RCA and statistical correlation tests is an effective way to identify unexpected network behavior and build RCA rules.

The rest of the paper is organized as follows. In Section 2, we first discuss the overall architecture of G-RCA, and we then provide the design details of each component of G-RCA. Section 3 describes how we quickly incorporate RCA applications for BGP flaps, throughput drop in CDN service and PIM flaps in Multicast VPN into G-RCA. Section 4 presents the operational experience that we have learned by

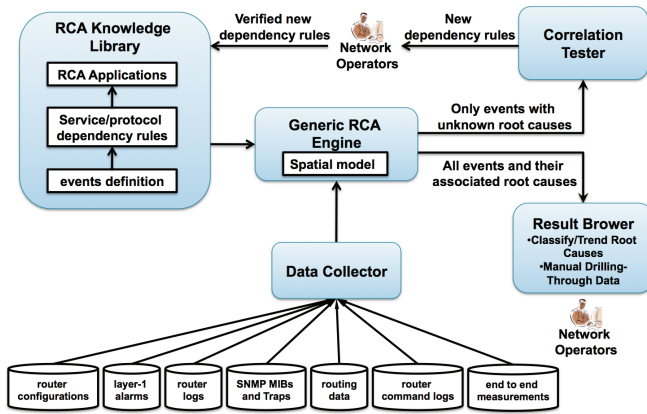


Figure 1: G-RCA Architecture

applying G-RCA in SQM of a tier-1 ISP. Section 5 discusses related work and Section 6 concludes the paper.

2. G-RCA ARCHITECTURE

As described in the Introduction, SQM presents a unique set of challenges for ISP networks. We now present our design of the Generic Root Cause Analysis (G-RCA) platform for SQM. We focus on the following aspects of G-RCA: (1) data (2) service dependency model (3) temporal/spatial correlation (4) reasoning logic and (5) domain knowledge building. The architecture of G-RCA is shown in Figure 1.

2.1 Data

Understanding service quality issues often requires an integrated view of different parts of the network. As mentioned earlier, G-RCA relies on a wide range of *proactively* collected information containing alarms, logs and performance measurement data from various network management systems. As simple as it sounds, there are tremendous instrumentation challenges for data management. Moreover, these data come from many devices and network management systems provided by different vendors, all reporting different statistics, from different time zones, and at varying intervals. The same device may be referenced in different ways by different systems or at different network layers (by a circuit identifier, an IP address, or an interface name). The timestamps can be mixture of local-time (depending on the time-zone of the device), network-time as defined by the service provider, and GMT. To facilitate SQM, one has to look across data sources efficiently. Hence, in G-RCA, the first optimization is on data integration – G-RCA’s Data Collector pulls all the data together, normalizes them so that they can be readily correlated, and stores them in database tables in real-time. The normalization across naming conventions, time zones and identifiers takes place as data is ingested into the Data Collector. This hides the data processing complexity from the remaining G-RCA components and eliminates the need for the operators to be painfully aware of the original data source details when correlating data. The

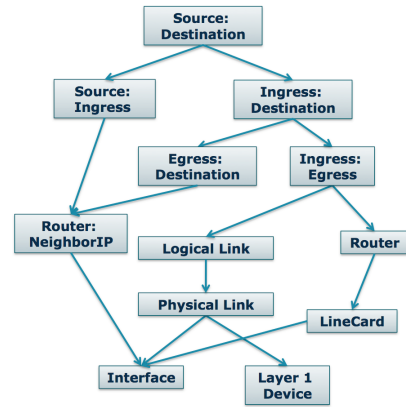


Figure 2: Spatial Model: Location Types and Mapping

data sources in our implementation of G-RCA include layer-1 alarms, router logs, SNMP MIBs and traps, routing data, router command logs, end to end measurements, and router configurations.

Expectedly, raw data are typically difficult to work with. In G-RCA, we introduce a notion of **event** – an event is a *signature* that captures a particular type of network conditions. We associate a *location type* with each event as it provides a key piece of information required for modeling service dependency (in 2.2). Figure 2 shows the location types that can be associated with a single event. A type of event can be extracted from raw input data through a parsing script, a database query, or some more sophisticated processing such as through an anomaly detection program. Specifically, an **event definition** in G-RCA is a tuple consisting of (*event-name, location type, retrieval process, additional descriptive information*), in which the retrieval process points to the actual scripts/queries needed to obtain the matching event instances.

Each **event instance** consists of an (*event-name, event start-time, event end-time, event location, additional info*). For example, the event definition (*link-congestion, interface, myscript*) indicates that the G-RCA Engine will use *myscript* to query SNMP traffic counter data to identify links that are nearly 100% utilized, and output event instances with location type *interface*. A corresponding event instance example is (*link-congestion, 2010-01-01 12:30:00, 2010-01-01 12:35:00, newyork-router1:serial-interface0*).

In order for network operators to quickly analyze new service problems, G-RCA pre-defines and implements a wide range of commonly used event signatures. These are included in the RCA Knowledge Library. For example, various RCA applications running on the IP backbone network may be interested in identifying link congestion events. Furthermore, there can be multiple signatures defined for the same network conditions. For example, in G-RCA Knowledge Library, a link congestion event is defined as either a near 100% link utilization in the SNMP traffic counter or a high number of overflow packets in the SNMP interface MIB. Network operators can pick the event definition that

Event Name	Event Description	Location Type	Data Source
Router reboot	router was rebooted	router	syslog
CPU high (average)	$\geq 80\%$ average utilization in 5-minute intervals	router	SNMP
CPU high (spike)	$\geq 90\%$ average utilization over the past 5 seconds	router	syslog
Interface down	LINK-3-UPDOWN msg	interface	syslog
Interface up	LINK-3-UPDOWN msg	interface	syslog
Interface flap	LINK-3-UPDOWN msg	interface	syslog
Line protocol down	LINEPROTO-5-UPDOWN msg	interface	syslog
Line protocol up	LINEPROTO-5-UPDOWN msg	interface	syslog
Line protocol flap	LINEPROTO-5-UPDOWN msg	interface	syslog
Basic CNI restoration	restoration events in layer-1 optical mesh network	layer-1 device	layer-1 device log
SONET alarm	alarms in the layer-1 SONET network	layer-1 device	layer-1 device log
Link congestion alarm	$\geq 80\%$ link utilization in 5-minute intervals	interface	SNMP
Link loss alarm	≥ 100 corrupted packets in 5-minute intervals	interface	SNMP
OSPF re-convergence event	link weight update in OSPF	interface	OSPF monitor
Router Cost In/Out	Router cost in/out inferred from link weight changes	router	OSPF monitor
Link Cost Out/Down	Link cost out or link down inferred from link weight changes	interface	OSPF monitor
Link Cost In/Up	Link cost in or link up inferred from link weight changes	interface	OSPF monitor
Command to Cost In Links	Command typed by operators to cost in links	interface	TACACS
Command to Cost Out Links	Command typed by operators to cost out links	interface	TACACS
BGP egress change	BGP next hop to some external prefix changed	ingress:destination	BGP monitor
In-network delay increase	delay increase between two PoPs	ingress:egress	performance monitor
In-network loss increase	loss increase between two PoPs	ingress:egress	performance monitor
In-network throughput drop	throughput drop between two PoPs	ingress:egress	performance monitor

Table 1: Common Event Definitions for a Tier-1 ISP’s IP Network

is best suited for the SQM task under investigation. Table 1 lists some common events in G-RCA for the tier-1 ISP network. Note that any event defined in the Knowledge Library can be redefined by an application. For example, the event “link congestion alarm” in web-hosting data-throughput-analysis can be easily redefined as “ $\geq 90\%$ link utilization in the SNMP traffic counter” when needed.

2.2 Service Dependency Model

The key to SQM is understanding the service dependency relationship between a user’s service problem and the underlying network devices and protocols supporting the service. G-RCA uses the model in Figure 2 to capture such dependencies. Though it appears simple, this model actually incorporates topological information (e.g., physical link connecting two different routers), cross-layer dependency (e.g., layer-1 devices supporting layer-3 links), logical and physical device association (which requires model of router configuration), and dynamic routing (e.g., BGP and OSPF routing in determining the service path between source and destination).

The service dependency abstraction is the most powerful component of G-RCA. By specifying the type of service problem (e.g. Ingress:Destination), G-RCA can automatically expand the service dependency to include all network elements that are associated with the service. However, realizing this model in practice is quite challenging. One crucial aspect to the dependency model is that the relationship is time varying – egress points to a destination network can change upon BGP updates; network paths can change as operators modify link weights; logical to physical mappings can change with configuration changes; even

physical connectivity can change over time. Associating the right network elements to a service event at a given time in history requires reconstructing the “network condition” at the time. G-RCA tackles this by implementing a range of sophisticated conversion utilities as follows:

(1) A pair of source of destination, both are outside the ISP is first mapped¹ to “Source:Ingress router” and “Ingress router:Destination”. Then in order to map from “Ingress router:Destination” to “Ingress router:Egress router” and “Egress router:Destination”, G-RCA looks up historical data of BGP tables to find out the longest prefix match and the network egress point for the destination. Note that BGP routing changes are typically not available at all ingress routers and only those changes at the BGP route-reflectors are available. In such case, approximation is needed. The reflectors which feed the ingress router with BGP updates are extracted from the daily archive of router configurations; BGP decision process at the ingress router is emulated based on the BGP route changes from its reflectors as well as the OSPF distance to available egress routers, and one best egress router is picked based on BGP best path selection.

(2) Both “Source:Ingress” and “Egress:Destination” can be mapped to a pair of access router and neighbor’s IP according to router configuration. The mapping from “Router:NeighborIP” to “Interface” can also be acquired by looking at the router configuration. This is particularly useful for diagnosing some protocol (e.g. BGP) events with neighbor IP that typically belongs to a router outside the ISP network.

(3) Given a pair of ingress router to egress router, the logical link or router level path between them can be computed

¹This mapping sometimes needs external mapping information.

via OSPF [20] routing simulation based on network-wide link weights from route-monitoring tools such as OSPFMon [24] (which listens to the flooded messages in OSPF).

(4) A point-to-point logical link can be associated with its attached routers by matching the IP addresses of the logical interface to a /30 network.

(5) A logical link may be mapped to more than one physical link for redundancy and capacity purposes by using techniques such as SONET APS (Automatic Protection Switching)[7] and Multilink PPP bundle [6]. This mapping can be obtained from the router configuration.

(6) G-RCA parses daily router configuration snapshots to infer that a router consists of a set of line-cards, which comprises multiple interfaces.

(7) An external database that keeps track of layer 1 inventory provides G-RCA with the mapping from physical links to all the layer-1 devices in between.

These conversion utilities are specific to the ISP network that we work with. However, we believe similar capability can be established when applying G-RCA to other networks.

2.3 Temporal Spatial Correlation

The most commonly asked question when network operators perform SQM tasks is *what happened in the network at the time that can be related to the service problem?* Breaking this question into more rigid and programable logic, G-RCA defines a temporal and spatial join rule as follows.

The simple concept “at the same time” can be quite entangled in each networking application. Firstly, there are typically various delay timers or expiration timers in various network protocols. Cause and effect rarely follows one another immediately. Secondly, there are always inaccuracy/uncertainty in the timing of network measurements. For example, router CPU measurement in a five-minute interval (via SNMP) may indicate a CPU overload condition within interval, but not any more precisely. G-RCA captures the above by defining a time-window to allow symptom event and diagnostic event to be joined (or “at the same time”).

Specifically, each temporal joining rule consists of six parameters: the left expansion margin X, right margin Y and an expansion anchor option (Start/End, Start/Start, or End/End) for each of the symptom event and diagnostic event. The margin values can be positive or negative in seconds, indicating forward-shift or backward-shift in time. Expansion anchor option indicates the time window expansion from the beginning or the end of the event. G-RCA determines a joint event pair when their expanded time windows overlap.

For example, consider a diagnosis event “Interface flap” (Start/End, X=5, Y=5) to be correlated with a symptom event “eBGP flap” (Start/Start, X=180, Y=5). Here 180 is used to model the cause-effect delay between “eBGP flap” and “Interface flap”. 180-second is the default setting for eBGP hold timer. In other words, “eBGP flap” is likely to occur 180s after an “Interface flap” event takes place. 5-second is used to model the inaccurate time-stamps in syslog messages. For an “eBGP flap” starting at time 1000 and ending

at time 2000, its expanded time interval is [820, 1005]. For an “Interface flap” starting at time 900 and ending at time 901, its expanded time interval is [895, 906]. The two event instances are considered joined since the two time intervals overlap.

For a diagnostic event to be correlated with a symptom event spatially, G-RCA defines the spatial joining rule that consists of three parts: 1) symptom event location type, 2) diagnostic event location type, and 3) *joining level*. The first two follow directly from the event definitions and must be one of the location types specified in Figure 2. The *joining level* is used to link symptom event locations with diagnostic event locations. For example, consider the symptom event of “packet loss on the uplink of an access router”² with location type “Interface”; consider the diagnostic event of “packet loss on an ISP access router customer-facing interface” also with location type “Interface”; with a joining level as “Router”, two event instances are spatially joined only if they take place on the same router. The Generic RCA Engine evaluates the built-in spatial model that ensures the symptom and diagnostic events are related according to the spatial joining rule specified. With this capability, when building a new application from G-RCA, operators are alleviated from the details of routing information, network topologies, router configurations, and cross layer dependency.

The above defines the temporal spatial relationship between one pair of symptom and diagnostic events. For any RCA application, typically many diagnostic signatures are sought after for different types of root causes. We model this using a **diagnosis graph** — an example application is shown in Figure 4. We refer to each edge in the diagnosis graph (the pair of symptom and diagnosis events and their temporal and spatial joining rules) as **diagnosis rules**. Given a diagnostic graph (for a specific SQM application), G-RCA evaluates the time and location conditions and collected data according to the data *retrieval process* in the event definition to determine the presence or the absence of diagnostic signature events.

Similar to the event definition library for frequently used event signatures, G-RCA also includes a library of diagnosis rules. Table 2 lists some of the commonly referenced rules in the ISP network and some others in the case studies in the rest of the paper. These are maintained in G-RCA Knowledge Library in Figure 1.

2.4 Reasoning Logic

Once data are collected regarding the presence or absence of diagnostic signature events, the next step is to determine the root cause of the symptom events based on this “evidence”. This reasoning logic can be implemented in many ways. In particular, G-RCA includes two reasoning engines: rule-based decision-tree-like reasoning and Bayesian inference. Interestingly, in our operational practice, we have found that rule-based reasoning logic is often preferred over its

²An uplinks is the link that connect an access router to a core network router.

Symptom Event	Diagnostic Event
Line protocol down/up/flap	Interface down/up/flap
Interface down/up/flap	SONET alarm
Line protocol down/up/flap	SONET alarm
Interface down/up/flap	Basic service restoration
Line protocol down/up/flap	Basic service restoration
BGP egress change	Interface down/up/flap
BGP egress change	Line protocol down/up/flap
Edge-to-edge delay increase	BGP egress change
Edge-to-edge loss increase	BGP egress change
Edge-to-edge throughput drop	BGP egress change
Edge-to-edge delay increase	Link congestion alarm
Edge-to-edge loss increase	Link congestion alarm
Edge-to-edge throughput drop	Link congestion alarm
Edge-to-edge delay increase	OSPF re-convergence event
Edge-to-edge loss increase	OSPF re-convergence event
Edge-to-edge throughput drop	OSPF re-convergence event
Link loss alarm	Link congestion alarm
Link loss alarm	Line protocol down/up/flap
OSPF re-convergence event	Line protocol down/up/flap
OSPF re-convergence event	Interface down/up/flap
OSPF re-convergence event	Commands to Cost In/Out Links
Link Cost Out/Down	Line protocol down
Link Cost Out/Down	Interface down
Link Cost Out/Down	Command to Cost Out Links
Link Cost In/Up	Line protocol up
Link Cost In/Up	Interface up
Link Cost In/Up	Command to Cost In Links
Link congestion alarm	OSPF re-convergence event

Table 2: Common Diagnosis Rules for a Tier-1 ISP’s Network

more sophisticated counterpart – this is because (1) it is easier to configure, (2) it gives simple and direct association between the diagnosed root cause and the evidence(s) for result interpretation, and (3) it is found to be very effective in most applications that we have explored. However, there are a few cases where Bayesian inference is preferred – for example when the root cause condition is unobservable (e.g., no direct evidence can be collected). Due to space limitation, we only discuss the rule-based reasoning engine in the rest of the paper and leave the details of our Bayesian inference engine and the application experience associated in using Bayesian inference in our technical report[29].

In our rule-based reasoning engine, we allow operators to associate a priority value for each edge in the diagnosis graph (such as in Figure 4). The higher the priority value, the stronger support that the operator believes is the diagnostic event to be the real root cause. G-RCA root cause reasoning engine can simply search through the evidence (as the presence of an diagnostic event) and identify the leaf node with the maximum priority. In the case of a tie between different leaf nodes, all of them are output as joint root causes.

2.5 Domain Knowledge Building

One of the important challenges in SQM is that operators’ domain knowledge and operational experience can be unreliable or incomplete. This implies that the specification of a diagnosis graph for a new SQM application offered by an

operator, especially the initial version, can be quite off, both in accuracy and in completeness. G-RCA must allow operators to validate the result or compare the symptom event with nearly *all* information from different data sources (i.e., irrespective to the diagnostic events) that occur at about the same time and is spatially related to the service problem under investigation. This manual drill-down and data exploration capability is included in G-RCA as Result Browser (in Figure 1). This has proven tremendously useful from operational experience. Operators can often spot the signature of overlooked root causes and add them into the diagnosis graph.

Also included in Result Browser are general root cause classification and trending capability for any given range of symptom event series. As mentioned in the Introduction, this becomes a handy tool for extracting actionable information from aggregate service impairment analyses.

More importantly, when manually looking for new root causes signatures for previously unable- to-diagnose symptom events, operators cannot distinguish the events that are simply co-occurring by chance from the ones having true causal relationship. G-RCA incorporates a statistical Correlation Tester to distinguish the two cases. The idea is that the number of coincidental joins should be bounded when examining the instances of symptom and diagnostic events in bulk. G-RCA implements the statistical correlation algorithm proposed in NICE [19]. In comparison to other canonical statistical test, NICE handles the event autocorrelation structure very well, which is commonly observed in networking event series. Note that operators can also choose to run the statistical correlation *blindly* – in this use, operators can quickly eliminate a large portion of unrelated signature events and focus on the small set of events for further investigation.

Through iteratively using Result Browser and Statistical Correlation Tester, operators can start with incomplete domain knowledge and gradually acquire new knowledge or learn unexpected network behaviors exhibited in the network data, which can then be incorporated into the diagnosis graph.

3. G-RCA APPLICATIONS

The key advantage of G-RCA in SQM is its capability to be rapidly customized into different RCA applications in the ISP’s network. In this section, we use three case studies – end to end throughput management in a CDN service, customer BGP flaps and network PIM flaps in multicast VPN to demonstrate the effectiveness of G-RCA.

3.1 BGP Flaps Root Cause Analysis

In the first case study, we focus on building a RCA tool to understand the root causes of eBGP[1] session flaps between customer routers (CR) and provider edge routers (PER) in a tier-1 ISP.

Customer networks exchange routes with an ISP through the eBGP session - the routes learned from the ISP inform

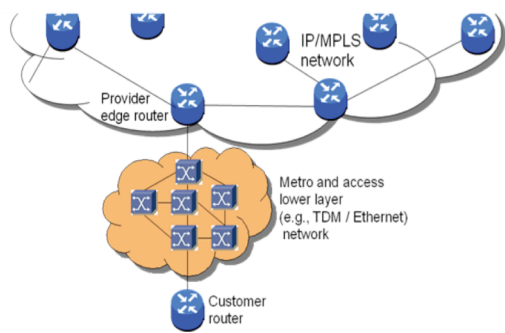


Figure 3: Physical connectivity between CR and PER

the customer network how to route to locations across the Internet and other sites of the same customer; routes shared from the customer network ensure that other sites can reach this site. If a session flaps, all routes are withdrawn and traffic is disrupted. Although relatively short (on the order of a minute), these flaps can disrupt applications. For example, VoIP sessions may be lost and financial transactions may be interrupted. We therefore aim to minimize the number of eBGP session flaps, taking actions to drive avoidable flaps permanently out of the network. The first step to achieving this is to understand the root cause of the flaps - a particularly challenging problem across a trust domain (between customer and provider networks). We achieve this using G-RCA by constructing application specific events and rules.

3.1.1 Application Specific Configuration

We start by constructing our BGP flap-specific events - those events which are not already included in the common event definitions in the RCA Knowledge Library (Table 1). These new application-specific events are illustrated in Table 3. Note that there are only 3 of them, in contrast with 7 other events which we reuse from the Knowledge Library.

After defining the application specific events, we need to add a few application specific diagnosis rules. The complete diagnosis graph is depicted in Figure 4. This combines events and rules taken from the RCA Knowledge Library (Table 2) with BGP application-specific events (shown as gray boxes) and application specific rules (dashed lines).

Finally, we specify priorities for different diagnosis rules for BGP flaps RCA, as depicted via the numbers on the edges in Figure 4. The highest priority is used to determine the most likely root cause among multiple root causes by the Generic RCA Engine. For example, if a BGP flap joins with both a high CPU event and a layer one flap, the layer one flap is identified as the root cause of this BGP flap as it is associated with a higher priority (180) edge.

3.1.2 Results

In order to understand the root causes of BGP flaps in the ISP, we ran the BGP flap RCA tool configured above for 5 provider edge routers in different locations, each of which has several hundred eBGP sessions established with

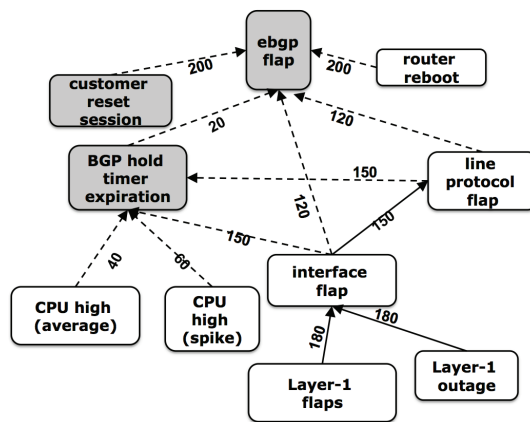


Figure 4: Diagnosis Graph for BGP Flaps Root Cause Analysis

Root Cause	Percentage (%)
router reboot	0.047
customer reset session	0.088
CPU high	0.886
CPU rising (high)	15.32
CPU rising (medium)	4.318
interface flap	29.004
line protocol flap	15.72
eBGP HTE(due to unknow reasons)	18.381
Short Layer-1 Flaps	0.205
Longer Layer-1 Outage	0.428
unknown	15.603

Figure 5: Root Cause Breakdown

customer routers. Figure 5 shows the root cause breakdown generated by the result browser in G-RCA.

One of the most critical insights revealed by applying our BGP flap RCA application to the operational network is in the BGP flaps induced by layer one flaps. The ISP routers are configured so that these short layer one flaps by design should not cause BGP flaps. Yet Figure 5 clearly indicates that they are. When considered in comparison with all BGP flaps, they are a relatively small percentage of events. However, further analysis revealed that for customers who tightly manage their networks, these layer one induced flaps were a far greater percentage of their flaps, and were causing concerns. As this behavior was inconsistent with design, it represented an opportunity for improving service.

After detailed lab replication and close collaboration with the router vendor, it was revealed that the routers were reacting to short layer one events when they should not – a bug in the router software. The router vendor rapidly repaired the code, which was then deployed across the ISP network as a software upgrade. Although the new software was extensively tested in the lab before being rolled out, the operational network is far more complex than can be replicated in a lab environment can replicate, and it was thus necessary to validate that the software repair did indeed have the desired effect in the operational network as well. The G-RCA appli-

Event Name	Event Description	Data Source
eBGP flap	eBGP session goes down and comes up, BGP-5-ADJCHANGE msg.	syslog
Customer reset session	eBGP session is reset by the customer, BGP-5-NOTIFICATION msg.	syslog
eBGP HTE	eBGP hold timer expired, BGP-5-NOTIFICATION msg.	syslog

Table 3: Application Specific Events for BGP Flaps Root Cause Analysis

Event Name	Event Description	Data Source
CDN end to end throughput drop	End to end throughput drop measured by Keynote	Keynote
CDN server issue	CDN server load is high	server logs

Table 4: Application Specific Events for Root Cause Analysis of Throughput Drop in CDN

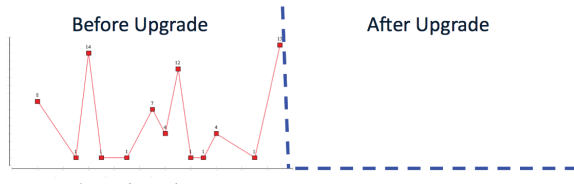


Figure 6: Short Layer-1 Flaps Induced eBGP Flaps

cation was the means through which this was achieved. The same five routers as in Figure 5 were picked as a part of an initial upgrade to test the software fix in the field. G-RCA’s result browser displays the results (Figure 6) regarding the number of layer one flap induced BGP flaps before and after the software upgrade. The graph clearly demonstrates that the software repair was successful as no such flaps were observed after the upgrade. Further analysis as the software upgrade was rolled out across the entire network demonstrated the same result.

This RCA application is now an integral part of the BGP monitoring in our tier 1 ISP. It is used to trend flaps and identify anomalous behavior which requires investigation (e.g., behavioral changes after new software upgrades). It is also used by operations and customer service representatives to provide automatic analysis of specific customer BGP flaps, for rapid responses to customer inquiries about such events.

3.2 Root Cause Analysis for CDN Service Impairments

In this case study we discuss how to build a new RCA application for managing service impairments in the ISP’s Content Delivery Network (CDN) [22]. The ISP operates a CDN service in which static web objects are hosted at several data centers across its network. Through dynamic DNS binding, HTTP requests are directed to the “closest” data centers and served from there. To monitor the end-to-end performance of the CDN service, Keynote [5], a commercial test and measurement infrastructure, periodically sends requests from thousands of computers spanning hundreds of metropolitan areas for a defined list of web objects. Keynote records various performance metrics (e.g., DNS lookup delay, time till first data packet received, avg/max throughput).

The aggregate performance results have become a benchmark in the industry, whilst more fine-grained measurements can be used to detect service impairments. The CDN service management team is tasked with using Keynote and other service measurements to detect service issues and then investigate their root cause.

We focus here on throughput degradations within the CDN. To troubleshoot a sudden throughput impairment in the CDN, operators must determine whether it is caused by network or service element faults, routing changes, congestion, or significant packet error and loss conditions that can impact the delivery of the web objects from the CDN server to the web client. Furthermore these conditions could be in the service layer (e.g., servers which host the content), somewhere in depths of the ISP’s network, or even in the wilds of the Internet. With thousands of routers and servers within the ISP’s responsibility, and lack of data regarding events outside the ISP domain, investigating such end to end performance issues is challenging, to say the least.

The primary challenge is to identify the network and service elements involved in servicing the requests at the precise time of the performance degradation. This is challenging to achieve during a real-time event and practically impossible to manually identify for historical events. However, G-RCA’s spatial model and proactive data collection enables such determination, and is the key to providing the ability to automatically troubleshoot these service issues.

3.2.1 Application Specific Configuration

To create the RCA application for CDN service impairments, we defined the application specific events (Table 4), diagnosis rules (Figure 7) and priorities (not shown). Note that the majority of the events and rules could again be drawn from the RCA Knowledge Library. The most important application-specific event is the “CDN end-to-end throughput drop” inferred from Keynote measurements. This event indicates a decrease in average download throughput and is the input to the RCA application. Each “CDN end-to-end throughput drop” event is associated with a start time and a location, which is defined by a pair of CDN server and client machine (e.g., Keynote agent). In addition to analyze the performance event generated from Keynote measurements, the RCA ap-

The PIM flap RCA application has proved to be extremely useful in classifying root causes of PIM adjacency losses and in guiding operators and engineers to a better understanding of actual MVPN performance in the network, allowing them to focus their effort on those issues that require their attention.

Running the G-RCA PIM application on a day's worth of events required about 1-2 hours. For each day, the application is currently able to identify the root causes for between 60% and 95% of PIM neighbor adjacency changes. We expect that with additional attention to those remaining unclassified events, the G-RCA PIM application will determine root causes for more than 99% of the events.

One of the primary benefits of the G-RCA PIM application was its ability to identify events that could be ignored. For example, a large number of PIM adjacency changes between PERs were found to be due to link flaps between the customer and provider routers, which typically are caused by customer activities. These link flaps did not affect multicast service between the PERs, but they did result in spurious PIM adjacency change syslog messages which could be safely ignored. Along similar lines, another root cause of the PIM adjacency changes was a customer disconnect on the PE router. By identifying those events that did not require investigation or further action, the G-RCA PIM application enabled operators to focus on those events that were potentially performance impacting. In addition, the G-RCA application identified transient events induced by a variety of unexpected causes. As an example, G-RCA found cases in which OSPF routing changes in the ISP's backbone network led to PIM neighbor adjacency changes. This violates the protocol designs and was thus unexpected - PIM adjacency loss should only be induced by outages far longer than typical OSPF convergence events. Through detailed G-RCA analysis and close collaboration with the network operator and relevant router vendors, several different causes of this unexpected protocol interaction have been identified such as various software bugs in devices. Once G-RCA was able to identify the root causes, appropriate solutions to the problems have been identified and put in place to permanently improve customer multicast performance. More importantly, these solutions may also help other ISPs who are not already be aware of these problems.

4. OPERATIONAL EXPERIENCE ON IMPROVING DOMAIN KNOWLEDGE

The main challenge in creating G-RCA applications is identifying the diagnosis rules. Domain knowledge typically provides a solid starting point, but our experience indicated that collating domain knowledge across potentially many domain experts can be surprisingly challenging. Domain knowledge is often distributed across multiple experts - no one expert understands the entire domain. These experts often have trouble thinking of the relevant rules when "put on the spot", or they are so busy fighting issues in the network that it is difficult to obtain their attention for long

enough to obtain the information. In other cases, the network operators domain knowledge may be wrong either because the relationships between events are extremely complex and not well understood, or because the network is not behaving as designed (as in Section 3.1.2). We thus found it surprisingly crucial to provide mechanisms integrated in G-RCA to facilitate diagnosis rule learning.

4.1 Learning Diagnosis Rules via Manual Iterative Analysis

With G-RCA, the individual responsible for creating an RCA application can follow an iterative process to identify new diagnosis rules. For example, in the PIM case, domain experts use data exploratory tools [13] to manually inspect unexplained neighbor adjacency changes to determine root cause(s). Once a new root cause was identified, it was codified in the RCA application, which was then run to identify all those events which could be explained by the existing set of rules and, more importantly, those which remained unexplained. The domain experts would then further sample remaining unexplained PIM flaps searching for new signatures which could be incorporated. The PIM application developer thus continually whittled down the number of unexplained flaps, by iteratively incorporating new rules and examining those which fell outside the scope of the new rules. By using G-RCA's result browser which made individual event analysis easy, the PIM application developer rapidly identified new diagnosis rules for the application and therefore revealed the anomalous behaviors discussed in Section 3.3.2.

4.2 Learning Diagnosis Rules via Statistical Correlation Test

Although the manual iterative analysis was effective in the PIM and other applications, we used a more "intelligent approach to analyzing BGP flaps. We illustrate this here by discussing our experiences in analyzing BGP flaps which were related to high CPU events.

Figure 5 illustrated that a significant portion of BGP flaps occurred at the same time as CPU overload were observed on the router. A naive assumption may be that these BGP flaps were in fact induced by high router CPU load. However, further inspection cast doubt on this assumption.

With the integrated data drilling-through functionality implemented in the result browser of G-RCA, it is easy for operators to explore additional information such as syslog messages and workflow logs that appear on the same router or location as the event being analyzed. Equipped with the powerful GUI, operators revealed via manual drilling-down that not all BGP flaps with a high CPU signature are actually due to CPU overload on PERs. In most cases, the high CPU utilization is likely *caused by* BGP flaps that are triggered from the customer side. Specifically, large amount of routing computation on PER in response to the BGP flaps produces high CPU utilization.

With this cyclic causal relationship - "BGP flap causes

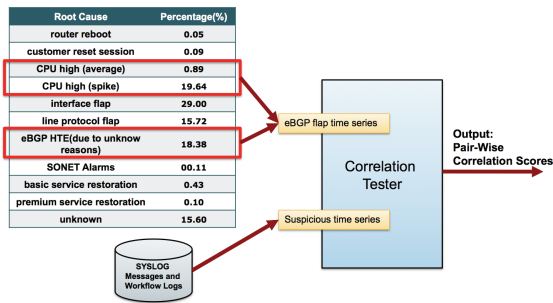


Figure 9: Interaction between Generic RCA Engine and Correlation Tester

CPU overload” and “CPU overload causes BGP session timeout”, evidence based diagnosis systems including our RCA tool hit their limit. We needed further refined signatures such as searching for other potential causes of the high CPU events to identify those which were not BGP flap induced and could thus explain BGP flaps.

Rapid manual inspection of events through G-RCA’s result browser worked well in some situations but our experience demonstrated that it does not work effectively if looking for relatively rare explanations among a sea of events. Instead, we took a different approach (Figure 9); using G-RCA’s correlation tester module to examine the statistical correlation between CPU-related BGP flaps and other types of events on the same PER. Specifically, we created a time series from all CPU-related BGP flaps as defined by our G-RCA application - those BGP flaps associated with BGP hold timer expiries, but where there was no evidence of link failures that could explain the flap, and which joined with one of the high CPU signatures. We then executed a statistical correlation test [19] between this time series and 831 other time-series created from workflow logs, and 2533 time series from syslog messages.

We fed three months worth of data into the correlation tester to analyze the CPU-related BGP flaps. Of the 3361 time series, 80 time series exhibited significant statistical correlation with our CPU-related BGP flaps. A rapid examination of these events by domain experts revealed that many of them were readily explained and/or incorporated into our existing application rules. For example, these CPU-related BGP events were strongly correlated with BGP notifications - a generic message logged for any BGP flap. However, the statistical correlation test did reveal some unexpected correlations. For example, the result revealed that certain provisioning activities (as derived from workflow logs) are strongly correlated with CPU-related BGP flaps. Drilling into individual cases, we identified a small number of incidents where un-related provisioning activities on some routers appear to have caused customer BGP sessions to flap, an unexpected router software behavior. As a result, 10 such incidents were sent to the router vendor for further investigation; the vendor has since implemented software changes to eliminate this issue.

It is worth noting that the pre-filtering of BGP flaps by their root causes as diagnosed by the Generic RCA Engine made a significant difference here. When we fed all BGP flaps to the correlation tester module, the correlation with provisioning activity was no longer statistically significant. By instead focusing on a small subset of the BGP flaps, the correlation “signal” is amplified, revealing the hidden issue. Thus, the interaction between generic RCA engine and the correlation tester is crucial to revealing subtle issues.

5. RELATED WORK

Many existing network management systems such as [10, 12, 26, 27, 3, 4, 2] work on the basis of individual network elements such as routers, line cards and interfaces, while G-RCA focuses on issues related to end-users’ service quality such as throughput degradation among different sites of a customer VPN. In addition, most of the existing network management systems focus on faults and hard failures that require immediate investigation, while G-RCA has its primary focus on classifying and trending the root causes of a large number of historical transient events, which provides operators with critical information that would help in driving the corresponding failure mode out of the network and eventually lead to service improvements.

A large body of recent work has been focusing on root cause analysis of network layer faults without direct evidence from lower layer in large ISPs such as SCORE [17], Shrink [14], and [16]. Shared Risk Link Group(SRLG) was proposed to model the cross-layer dependency, where a group of network layers entities depend on the same physical layer entity. With the concept of SRLG, finding the root cause of network layer faults becomes a minimal set cover problem in a bipartite graph in SCORE and [16]. Shrink enhanced them by incorporating Bayesian network to model inaccurate measurements and SRLG information. While G-RCA is designed for more general root cause analysis problems other than only for network layer faults, G-RCA could actually incorporate SCORE-like algorithms to infer what is happening if there is no direct evidence.

There has been increasing interest in root cause analysis based on inferred diagnosis graph among components. Pinpoint [9] keeps track of client requests to discover the component dependencies and faulty components in a distributed system. [23] employs active probing to infer the diagnosis graph for fault diagnosis in a network of routers and end hosts. Sherlock [8] passively infers the dependencies among the network, server and applications from logs. It focuses on diagnosing end-user problems in the enterprise networks. Recently NetMedic [15] is proposed to perform detailed root cause analysis in small enterprise networks by modeling the network as a diagnosis graph of fine-grained components such as processes and firewall configuration. All of these existing systems have to face a design tradeoff between scalability of analysis and complexity of models. While [9], [23] and Sherlock focuses on scalable analysis with simple models, NetMedic focuses on applying complex/fine-

grained models in small networks. G-RCA is designed to support both large scale analysis and complex models. We simply leverage the existing domain knowledge from network operators and let them specify the initial diagnosis graph, which can be iteratively improved by interactions between G-RCA's generic RCA engine and correlation tester. In G-RCA, automatically learned diagnosis rules have to be checked by network operators before being incorporated into the RCA knowledge library to verify the correctness.

Machine learning and statistical methods have been widely applied in mining relationships among events. NICE [19] proposed a novel statistical correlation approach with circular permutation test for learning correlation between two event time-series. While CORDS [11] employs chi-squared analysis to mine correlations, SPIRIT [21] uses Principal Component Analysis. More sophisticated and computationally expensive techniques such as Hidden Markov Chain [28] and association rule mining [18, 25] have also been proposed to mine relationships among multiple event time-series. Although G-RCA focuses on identifying the root cause of each individual event of interest, these techniques are actually complementary to G-RCA for mining more rules.

6. CONCLUSIONS AND FUTURE WORK

In this paper we described G-RCA, a generic root cause analysis platform for service quality management in large IP networks. G-RCA is an ideal platform for SQM in a “constantly changing” network environment. Firstly, it captures the layered network model in its knowledge library in the form of diagnosis rules. Most of them can be reused by various RCA applications. Its generic RCA engine implements the common logic in various RCA tasks such as temporal/spatial correlation, rule-based reasoning and bayesian inference. In addition, the generic RCA engine also implements a network location model, which models various network locations and the mappings among them. Thanks to the knowledge library and RCA engine, new RCA applications can be quickly incorporated into G-RCA via simple configuration. Secondly, domain knowledge in existing RCA applications can be refined by the interaction between RCA engine and correlation tester, which is important for a dynamic network environment. Thirdly, in order to analyze a large number of service quality issues and classify/trend the root causes of them, it proactively collects all types of data from different sources and normalize them in real-time.

Our work can be extended in several directions. First, we plan to refine the inference algorithm and simplify its configuration to further improve its usability. Second, we want to leverage the interaction between generic RCA engine and correlation tester to systematically complete operators' domain knowledge. Finally, we will work with network operators to extend the G-RCA platform into other networks and services such as cellular data network, IPTV and VoIP.

7. REFERENCES

- [1] A border gateway protocol 4 (bgp-4).
<http://www.ietf.org/rfc/rfc4271.txt>.

- [2] Emc ionix platform.
<http://www.emc.com/products/family/ionix-family.htm>.
- [3] Hp operations center.
https://h10078.www1.hp.com/cda/hpms/display/main/hpms_content.jsp?zn=bto&cp=1-11-15-28_4000_100__.
- [4] Ibm tivoli. <https://www-01.ibm.com/software/tivoli/>.
- [5] Keynote systems, inc. website. <http://www.keynote.com/>.
- [6] Overview of Multilink PPP Bundle.
<http://www.juniper.net/techpubs/software/erx/junose81/swconfig-link/html/mlppp-config2.html>.
- [7] SONET Automatic Protection Switching.
http://www.cisco.com/en/US/tech/tk482/tk606/tsd_technology_support_sub-protocol_home.html.
- [8] P. Bahl, R. Chandra, A. Greenberg, S. Kandula, D. A. Maltz, and M. Zhang. Towards highly reliable enterprise network services via inference of multi-level dependencies. In *SIGCOMM '07: Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 13–24, 2007.
- [9] M. Chen, E. Kiciman, E. Fratkin, A. Fox, and E. Brewer. Pinpoint: Problem determination in large, dynamic internet services. In *Proceedings of the 2002 International Conference on Dependable Systems and Networks*, pages 595–604, 2002.
- [10] P. Corn, R. Dube, A. McMichael, and J. Tsay. An autonomous distributed expert system for switched network maintenance. In *Proceedings of IEEE GLOBECOM88*, pages 1530–1537, 1988.
- [11] I. Ilyas, V. Markl, P. Haas, P. Brown, and A. Abounaga. CORDS: automatic discovery of correlations and soft functional dependencies. In *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, pages 647–658, 2004.
- [12] C. Joseph, J. Kindrick, K. Muralidhar, and T. Toth-Fejel. MAP fault management expert system. *Integrated Network Management I, North-Holland, Amsterdam*, pages 627–636, 1989.
- [13] C. Kalmanek, I. Ge, S. Lee, C. Lund, D. Pei, J. Seidel, J. van der Merwe, and J. Ates. Darkstar: Using exploratory data mining to raise the bar on network reliability and performance. In *Design of Reliable Communication Networks, 2009. DRCN 2009. 7th International Workshop on*, pages 1–10. IEEE, 2009.
- [14] S. Kandula, D. Katabi, and J. Vasseur. Shrink: A tool for failure diagnosis in IP networks. In *Proceedings of the 2005 ACM SIGCOMM workshop on Mining network data*, pages 173–178, 2005.
- [15] S. Kandula, R. Mahajan, P. Verkaik, S. Agarwal, J. Padhye, and P. Bahl. Detailed diagnosis in enterprise networks. In *SIGCOMM '09: Proceedings of the 2009 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 243–254, 2009.
- [16] R. Kompella, J. Yates, A. Greenberg, and A. Snoeren. Detection and localization of network black holes. In *IEEE INFOCOM 2007. 26th IEEE International Conference on Computer Communications*, pages 2180–2188, 2007.
- [17] R. R. Kompella, J. Yates, A. Greenberg, and A. C. Snoeren. Ip fault localization via risk modeling. In *NSDI'05: Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation*, pages 57–70, 2005.
- [18] F. Le, S. Lee, T. Wong, H. Kim, D. Newcomb, F. Le, S. Lee, T. Wong, H. Kim, and D. Newcomb. Minerals: Using Data Mining to Detect Router. In *ACM Sigcomm Workshop on Mining Network Data (MineNet)*, 2006.
- [19] A. Mahimkar, J. Yates, Y. Zhang, A. Shaikh, J. Wang, Z. Ge, and C. Ee. Troubleshooting chronic conditions in large IP networks. In *Proceedings of the 2008 ACM CoNEXT Conference*, 2008.
- [20] J. Moy. RFC2328: OSPF Version 2. 1998.
- [21] S. Papadimitriou, J. Sun, and C. Faloutsos. Streaming pattern discovery in multiple time-series. In *Proceedings of the 31st international conference on Very large data bases*, pages 697–708, 2005.
- [22] M. Pathan, R. Buyya, and A. Vakali. Content Delivery Networks: State of the Art, Insights, and Imperatives. *Content Delivery Networks*, page 1, 2008.
- [23] I. Rish, M. Brodie, and S. Ma. Efficient fault diagnosis using probing. In *AAAI Spring Symposium on Information Refinement and Revision for Decision Making*, 2002.
- [24] A. Shaikh and A. Greenberg. OSPF monitoring: Architecture, design, and deployment experience. In *Proc. USENIX/ACM NSDI*, 2004.
- [25] J. Treinen and R. Thurimella. A framework for the application of association rule mining in large intrusion detection infrastructures. *Lecture Notes in Computer Science*, 4219:1, 2006.
- [26] J. Wright, J. Zielinski, and E. Horton. Expert systems development: the ACE system. *Expert Systems Applications to Telecommunications*, pages 45–72, 1988.
- [27] T. Yamahira, Y. Kiriha, and S. Sakata. Unified fault management scheme for network troubleshooting expert system. *Integrated Network Management, I. North-Holland: Elsevier Science Publishers BV*, 1989.
- [28] K. Yamanishi and Y. Maruyama. Dynamic syslog mining for network failure monitoring. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 499–508, 2005.
- [29] H. Yan. U-RCA: A Unified Root Cause Analysis Platform for Service Quality Management in Large IP Networks. Technical Report 10-103, Colorado State University, 2010.