# Towards Automated Performance Diagnosis in a Large IPTV Network

Ajay Mahimkar[§], Zihui Ge[‡], Aman Shaikh[‡], Jia Wang[‡], Jennifer Yates[‡], Yin Zhang[§], Qi Zhao[‡]
The University of Texas at Austin[§]
{mahimkar,yzhang}@cs.utexas.edu
AT&T Labs – Research [‡]
{gezihui,ashaikh,jiawang,jyates,qzhao}@research.att.com

## ABSTRACT

IPTV is increasingly being deployed and offered as a commercial service to residential broadband customers. Compared with traditional ISP networks, an IPTV distribution network (i) typically adopts a hierarchical instead of mesh-like structure, (ii) imposes more stringent requirements on both reliability and performance, (iii) has different distribution protocols (which make heavy use of IP multicast) and traffic patterns, and (iv) faces more serious scalability challenges in managing millions of network elements. These unique characteristics impose tremendous challenges in the effective management of IPTV network and service.

In this paper, we focus on characterizing and troubleshooting performance issues in one of the largest IPTV networks in North America. We collect a large amount of measurement data from a wide range of sources, including device usage and error logs, user activity logs, video quality alarms, and customer trouble tickets. We develop a novel diagnosis tool called Giza that is specifically tailored to the enormous scale and hierarchical structure of the IPTV network. Giza applies multi-resolution data analysis to quickly detect and localize regions in the IPTV distribution hierarchy that are experiencing serious performance problems. Giza then uses several statistical data mining techniques to troubleshoot the identified problems and diagnose their root causes. Validation against operational experiences demonstrates the effectiveness of Giza in detecting important performance issues and identifying interesting dependencies. The methodology and algorithms in Giza promise to be of great use in IPTV network operations.

## Categories and Subject Descriptors

C.2.3 [**Computer-Communication Networks**]: Network Operations—*Network management*

## General Terms

Management, Performance, Reliability

## Keywords

IPTV, Network Diagnosis


*SIGCOMM'09,* August 17–21, 2009, Barcelona, Spain.


## 1. INTRODUCTION

In the past few years, we have seen a global flurry among telecommunication companies in the rapid roll-out of Internet Protocol Television (IPTV). IPTV encodes live TV streams in a series of IP packets and delivers them to users through residential broadband access networks. There are two key reasons behind the rapid growth of IPTV. First, IPTV allows Internet service providers (ISPs) to strengthen their competitiveness by offering new services such as triple-play (digital voice, TV, and data) and quadruple-play (digital voice, TV, data and wireless). Second, IPTV offers users much greater flexibility and interactivity and opens up opportunities for a broad range of new applications.

Compared with traditional ISP networks, IPTV distribution networks exhibit several unique characteristics. First, IPTV imposes stringent requirements on both performance and reliability. Even a small amount of packet loss and delay could seriously impair the video quality perceived by users (especially those viewing live sport events). Second, IPTV networks have tremendous scale. A large IPTV network can already have millions of residential gateways today, and the number is rapidly growing. Third, IPTV heavily relies on IP multicast protocols. Although multicast technologies have been available for two decades, they start to have wide-scale deployment and use only recently, and so the operational experience of multicast is limited. These characteristics impose significant challenges in the effective management of IPTV networks.

In this paper, we focus on the characterization and troubleshooting of faults and performance impairments in one of the largest commercial IPTV deployments in North America. At the time of writing this paper, the service provider had well over one million subscribers spanning four different time zones. We collected large amounts of diverse measurements both within the residential part of the IPTV network and from the provider network. These measurements ranged from end device usage/error logs, user activity logs, video quality alarms and customer trouble tickets. Mining such a vast and diverse dataset for characterization and troubleshooting is a challenging problem.

**Challenges.** In order to effectively and automatically detect and troubleshoot performance issues in IPTV networks, we need to address the following key challenges.

1. **Large number of network devices.** Scalability is a big challenge in effectively managing and troubleshooting IPTV networks. There are a vast number of network devices (e.g., millions of residential gateways), resulting in a tremendous volume of network data (e.g., event-series) that must be examined for localizing and troubleshooting performance issues. Mining this amount of data is a challenging problem.

2. **Topological and routing models.** Since IPTV uses IP multicast protocols to distribute content to end-users, events in network propagate from the root of the multicast tree towards the

end-users (*i.e.*, leaves in the tree). It is important to take the impact scope of network events into account when troubleshooting a performance problem. Blindly analyzing data without considering the topological and routing models can easily lead to an information "snow" of results and overwhelm the network operations team with false alarms.

3. **Skewed event distribution.** The majority of individual event-series have very small frequency count. This makes it challenging for performing statistical analysis due to insufficient sample size. Thus, there is a need to perform aggregation of events, both in space (across different locations) and in time (over different aggregation intervals).

4. **Discovery of causal dependencies among events.** During performance troubleshooting, network operations team are interested in identifying dependencies between symptom[1] events and other network events. It is challenging to accurately discover causal dependency among different events because of (i) diversity of events, ranging from point events (e.g., router logs) to range events (e.g., 5-minute summarized SNMP data), (ii) inaccurate event timestamps due to measurement artifacts, imperfect clock synchronization, and limited clock resolution, and (iii) distributed event propagation, which may cause an event to be recorded long after when it had impacts.

**Our contributions.** In this paper, we present the first characterization study of performance issues and faults in operational IPTV networks. The study provides interesting insights into the distribution of events, spatio-temporal locality, and time-of-day effects. For fault localization and performance troubleshooting in IPTV networks, we develop Giza, a multi-resolution infrastructure that includes a suite of novel statistical data mining techniques.

1. To cope with a vast number of network devices and network event-series, Giza first applies *hierarchical heavy hitter detection* to identify the spatial locations where the symptom events are dominant. The hierarchy for spatial locations is created using the IPTV multicast tree structure. This greatly reduces the amount of data for subsequent processing. Focusing on the hierarchical heavy hitters also gives sufficient sample points to perform further statistical analysis.

2. Giza applies *statistical event correlation analysis* at heavy hitter locations to identify those event-series that are strongly correlated with the heavy hitter symptom. The list of strongly correlated event-series includes both potential root causes and impacts of the symptom event.

3. Giza applies *statistical lag correlation* and $\ell^1$ *norm minimization* techniques to discover the causal dependencies between events. It constructs a causal dependency graph for each symptom event-series. The graph generated by Giza is sparse and helps network operators to effectively and automatically diagnose symptom events. The discovery process requires minimal domain knowledge.

We evaluate Giza using data collected from an operational IPTV network and service. Our data sources include both the provider network and the customer home networks. We demonstrate that our assumptions about multi-resolution analysis are indeed valid. We also show that the causal discovery algorithm used in Giza outperforms a state-of-art approach known as WISE [28]. In addition, we apply Giza in diagnosing the causes of customer trouble tickets and validate our conclusions against those obtained via operational experiences. Our results demonstrate that Giza has the promise to be an effective tool in identifying and troubleshooting performance problems.

[1] A *symptom event* is an event that is indicative of a network problem and is observable to the network operations team. It is the target event that the network operation team tries to troubleshoot and diagnose.

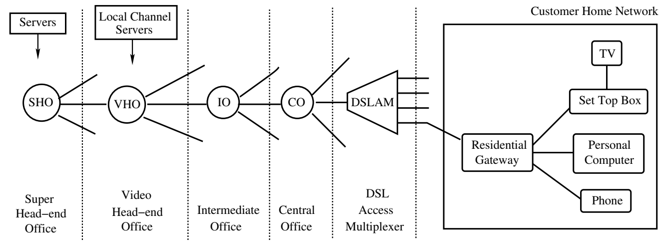


**Figure 1: IPTV service architecture.**

**Paper organization:** The rest of this paper is organized as follows. We provide an overview of a large IPTV service and characterize interesting network activities and performance issues for both the provider network and customer home networks in Section 2. Section 3 presents the system design and detailed description of Giza. We validate Giza, compare it with a previously published solution – WISE, and illustrate the usage of Giza in a case study in Section 4. Section 5 summarizes the related work and Section 6 concludes the paper.

# 2. IPTV NETWORK PERFORMANCE CHARACTERIZATION

In this section, we first present an overview of the IPTV service architecture and the data sets we use in this paper. We then present characterization results and motivate a multi-resolution troubleshooting system in troubleshooting performance issues in IPTV networks.

## 2.1 Overview of IPTV Service

Fig. 1 shows the architecture of how IPTV service is delivered to residential customers by the service provider. In an IPTV system, live TV streams are encoded in a series of IP packets and delivered through the residential broadband access network. The SHO (Super Head-end Office), which is the primary source of national television content, digitally encodes video streams received externally (e.g., via satellite) and transmits them to multiple VHOs (Video Head-end Offices) through a high-speed IP backbone network. The VHOs, each responsible for a metropolitan area, in turn acquire additional local contents (e.g., local news), perform some further processing (e.g., advertisement insertion) and transmit the processed TV streams to end users upon request. Depending on the service provider, these TV streams go through a various number of routers or switches such as intermediate offices (IO), central offices (CO), and digital subscriber line access multiplexer (DSLAM) before reaching a residential home.

Inside a home, an RG (Residential Gateway) serves as a modem and connects to one or more STBs (Set-Top Boxes). It receives and forwards all data, including live TV streams, STB control traffic, VoIP and Internet data traffic, into and out of the subscriber's home. Finally, each STB connects to a TV.

We use the terminology and pyramid hierarchy as shown in Fig. 2 to determine events at different aggregation levels. A DSLAM serves multiple STBs, a CO serves multiple DSLAMs, an IO serves multiple COs, a VHO serves multiple IOs, and finally, an SHO serves content to all VHOs.

It is worthwhile to note that live IPTV streams are delivered from SHO to residential home via native IP multicast in order to save bandwidth consumption in the network. In addition to live TV channels, STBs also support advanced features such as digital video recording (DVR), video on demand (VoD), picture-in-picture (PIP), high definition (HD) channels, choice programming, online gaming and chatting.

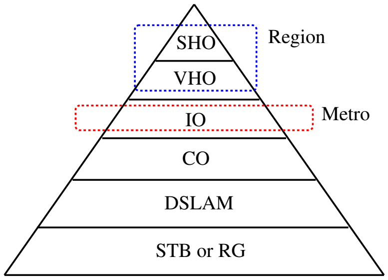


**Figure 2: Pyramid structure in IPTV network.**

## 2.2 Data Sets

We collected a large variety of data from one of the largest commercial IPTV service providers in North America. As of the end of 2008, this IPTV service provider has over one million subscribers (i.e., residential homes), and over two million STBs in use. Our data set consists of four types of data: *customer care call records*, *video quality alarms*, *home network performance/activities*, and *provider network performance/activities*.

**Customer care call records.** We obtained complete customer call records from the IPTV service provider. A customer call can be related to provisioning, billing and accounting, or service disruption. In this paper, we mainly focus on the customer calls regarding service disruptions, each of which results in a trouble ticket associated with a type of the performance issues, the start and end times, the customer account number, etc. The common customer trouble tickets include picture freezing on TV screen, audio and video out of synchronization, DVR black screen, program guide issues, high definition issues and parental control issues.

**Video quality alarms.** We obtained alarms data from the video quality monitors deployed within the IPTV service provider network. The monitors gather statistics such as packet loss, packet delay and outage durations. These statistics are then used as a quality indicator for video.

**Home network performance/activities.** There are several event traces collected from each STB including power state, resets, and crashes. The STB power state logs indicate when the STB was turned on and turned off. This data can be used to identify the active STBs at any point of time. The STB reset logs provide information on when the STB was rebooted by the user. Finally, the software running on an STB may occasionally crash which is also recorded in the log. For each crash, the log contains information about the time and type of the event. In addition to the STB logs, we also obtained the reboot log for each RG, which indicates when the RG was rebooted.

**Provider network performance/activities.** We obtained SNMP MIBs and traps data from every SHO, VHO, IO, and CO. The SNMP data provide performance statistics such as packet and byte counts, packet loss, and CPU utilization. We also obtained Syslogs from devices at every SHO and VHO. These logs provide information about state changes for control plane protocols and error conditions of the devices. Examples include multicast neighbor loss, OSPF adjacency state changes, and BGP session changes.

## 2.3 Characterizing IPTV Performance Issues

We conducted our analysis over three-month data collected in 2008. We present the characterization results focusing on spatial and temporal patterns of various performance related events observed along the paths from the SHO to STBs.

### *2.3.1 Customer Trouble Tickets*

We analyze the trouble tickets that were triggered by customer calling for performance related issues. Based on the nature of the reported performance issues, we classify customer trouble tickets into different categories. Table 1 shows the distribution of number of tickets for the top ten categories. We observe that "live TV video" related performance issues (e.g., video quality, blue screen on TV, picture freezing or no picture on TV) constitute almost half of the trouble tickets. This is not surprising because live TV channels are the basic service offered to customers. When there is any performance issue on the IPTV service, it is likely to be noticed and reported by customers as live TV video related problem. The category "requested information or upgrade" ranks the second on the list. The trouble tickets in this category indicate that the customer was experiencing some performance issues and requested further information. Other top categories range from DVR and remote control related issues to video on demand and parental control related issues.

| Ticket category | Percentage |
|---|---|
| Live TV video | 46.5 |
| Requested information or upgrade | 12.9 |
| Digital video recording (DVR) | 9.6 |
| Remote control | 8.2 |
| Equipment (STB, RG, PC) | 7.7 |
| High definition (HD) | 4.4 |
| Audio | 3.5 |
| Program guide | 1.6 |
| Video on demand (VoD) | 1.6 |
| Parental control | 1.6 |
| Others | 2.4 |

**Table 1: IPTV customer trouble tickets.**

### *2.3.2 Video Quality Alarms*

We analyze the video quality alarms reported by the video monitors. The Media Delivery Index (MDI) is a measurement that is used as a quality indicator for video and streaming media. It comprises of two elements: delay factor (DF) and media loss rate (MLR). The delay factor is the maximum difference between the arrival of a packet and its playback. It indicates the time duration over which a packet stream must be buffered in order to prevent packet loss. The ideal DF score is the packet size. The media loss rate is the number of lost or out-of-order packets within a time interval. The ideal media loss rate is 0%. We analyze one month worth of data and identify that the alarms related to delay factor contribute to the majority (around 79%). Other important video quality alarms include high media loss rate, video stream outages, transport stream outages, high flow bit rates, high transport stream bit rates and synchronization errors.

### *2.3.3 Home Network Performance/Activities*

We characterize a variety of data traces collected from STB, and RG including STB crash, reset, power state and RG reboot. Table 2 shows the distribution of STB crash events. There are mainly four types of crashes: managed, native, out of memory and watch dog reboot. All managed code runs in a protected environment. Some events in this environment are considered fatal and will terminate the Microsoft Mediaroom application, generating a managed crash log in the process. Native crash events occur outside of the protected Microsoft Mediaroom common language run-time (CLR). Examples of native crash events are the crashes in the device drivers, and low level non-application code. The managed and native crash events are vast majority of all the STB crashes. The watch dog reboot occurs when the STB hangs in the low level kernel resulting in the watch dog timer expiry. This type of crash contributes to nearly 20% of all STB crash events. Finally, there is a small percentage of crashes caused by out of memory error. This occurs when the STB native layers run out of memory.

Fig. 3 shows the distribution of the number of simultaneous native STB crash events occurring within a fixed time-bin of five min-

| Crash type | Native | Managed | Watch dog reboot | Out of memory | Others |
|---|---|---|---|---|---|
| **Percentage** | 44.9 | 35.9 | 18.4 | 0.5 | 0.2 |

**Table 2: STB crash events.**

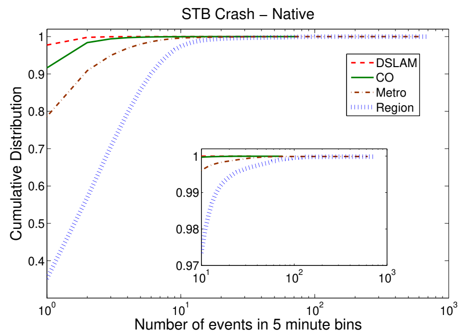


**Figure 3: Number of simultaneous native STB crash events at different spatial aggregation levels.**

| Syslog messages | Percentage |
|---|---|
| Layer-1 alarms (SONET, Ethernet) | 34 |
| IP link flaps | 21.3 |
| SAP (Session Announcement Protocol) state changes | 15 |
| OSPF routing events | 8.5 |
| Configuration changes | 7.4 |
| tmnx events | 7.3 |
| SDP (Session Description Protocol) state changes | 1.5 |
| MPLS (Multiprotocol Label Switching) state changes | 0.7 |
| RSVP (Resource Reservation Protocol) state changes | 0.5 |
| VRRP (Virtual Router Redundancy Protocol) events | 0.4 |
| PIM multicast events | 0.3 |
| BGP (Border Gateway Protocol) events | 0.2 |
| PPP (Point-to-Point Protocol) events | 0.1 |
| Others | 2.8 |

**Table 3: Syslog messages from SHO and VHOs.**

utes. While there are very few simultaneous events occurring for most of the time, there are a few time bins in which a large number of events occurred. This observation holds at all spatial aggregation levels.

### 2.3.4 *Provider Network Performance/Activities*

We analyze SNMP and syslog data collected from the provider network. Table 3 shows the distribution of different types of syslog messages observed on devices in the SHO and VHOs. We focus only on performance related events. We observe that layer-1 alarms and IP link flaps contribute to over 55% of the events. In addition, session announcement protocol (SAP[2]) and session description protocol (SDP) related issues contribute around 16% of events. These protocols are used for multimedia communication sessions and their issues may potentially impact IPTV performance.

Fig. 4 shows the cumulative distribution of inter-arrival times for the top four syslog messages in Table 3. We observe high temporal locality from the figure.

### 2.3.5 *Daily Pattern of Events*

Fig. 5 shows the daily pattern for STB crash, STB resets, STB tuned ON, STB turned OFF, customer trouble tickets and provider network logs. The time is represented in GMT. We observe that there is a lot of activity (STB events and customer trouble tickets) between 00:00 GMT and 04:00 GMT, which is evening prime time in North America, and between 12:00 GMT and 23:59 GMT (mid-night), which is day time in North America. We also observe that there is a relative "quiet" period between 4:00 GMT and 12:00 GMT which is the time during which the customers are sleeping.

[2]Session Announcement Protocol (SAP) is used to broadcast multicast session information. SAP uses SDP (session description protocol) for describing the sessions and the multicast sessions use real-time transport protocol (RTP).

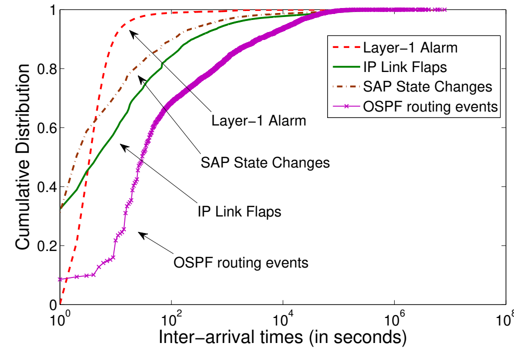


**Figure 4: Cumulative distribution of inter-arrival time of provider network syslog messages at VHO and SHO.**

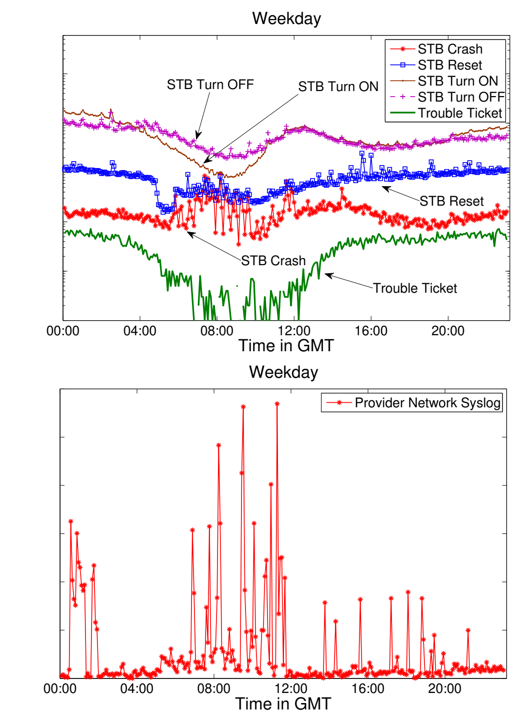


**Figure 5: Daily pattern of STB crash, STB resets, STB turn ON, STB turn OFF, customer trouble tickets and provider network syslogs.**

During this time window, the number of syslog messages at SHOs and VHOs in the provider network can be very high. This can be explained by the network provisioning and maintenance activities. Overall, we find that the more customers watch TV, the more performance issues occur and are reported.

## 2.4 A Case for Multi-resolution Analysis

As we have shown earlier in this section, the IPTV network provides service to about a million residential customers. The network operator needs to identify and troubleshoot performance problems on millions of devices ranging from those in SHO and VHOs inside the provider network to residential gateways and STBs on customer's home network.

One approach to tackle this problem is to identify a few heavy hitter devices, where the performance issues are significant. This is a standard approach applied in IP network troubleshooting where the operation team focuses on a few chronic problems which contribute to a vast majority of the performance issues observed in the

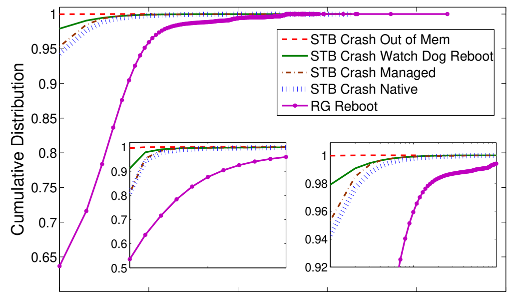


**Figure 6: Distribution of performance events among devices. X-axis is the number of events and not shown for privacy reasons. It starts with event count of one. The embedded plot (left) starts with event count of zero.**

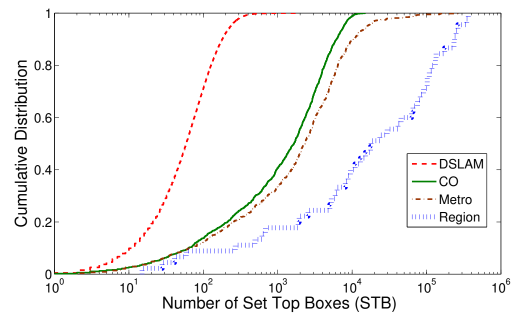


**Figure 7: Skewed distribution of number of STBs and RGs for various spatial aggregation levels.**

network. However, this observation does not hold for the IPTV network. Fig. 6 shows the number of events observed in the IPTV network over a three-month time period. We find that there are a few heavy hitters, but the contribution from these heavy hitters is only a small fraction of the total events. That is, non heavy hitters contribute to vast majority of the events. Therefore, focusing on a few heavy hitters is not sufficient to troubleshoot majority of performance issues in an IPTV network.

In addition, we observe that the occurrence of a given event on an individual device is extremely low. For example, as shown in the embedded plots in Fig. 6, about half of the residential gateways do not have a single reboot event during the three month time period that our study is conducted. Only about 20% of set top boxes experienced native or managed crash events. The watch dog reboot and out of memory crashes are even rarer.

To address the above challenge, we take advantage of the multicast hierarchy which is used for delivery of live IPTV channels and propose to apply multi-resolution analysis by detecting hierarchical heavy hitters across multiple spatial granularities such as DSLAM, CO, Metro and Region. Note that the distribution of the number of set top boxes and residential gateway per spatial aggregation is not uniform (shown in Fig. 7), which indicates that we cannot directly apply the standard hierarchical heavy hitter detection algorithms.

# 3. THE DESIGN OF GIZA

In this section, we present the design of Giza, a multi-resolution data analysis infrastructure for analyzing and troubleshooting performance problems in IPTV networks. Giza includes a suite of statistical techniques for prioritizing various performance issues (i.e., identifying prevailing and chronic performance-impacting conditions), event-correlation detection, dependency-graph reduction, causality discovery and inference.

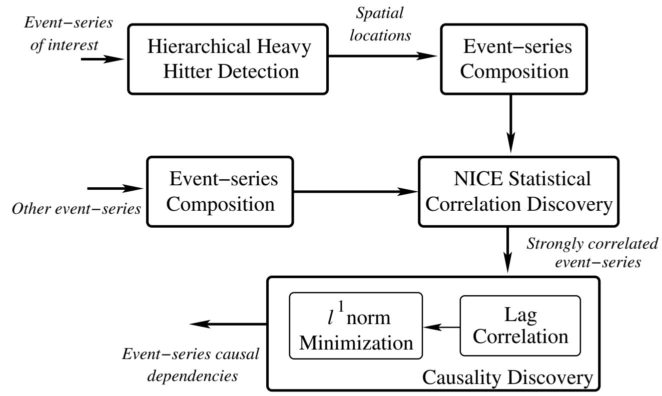


**Figure 8: Architecture of Giza.**

## 3.1 Overview

As mentioned earlier, one of the key challenges in managing IPTV service is its massive scale (particularly in terms of the network edge devices) and hence its overwhelming amount of performance monitoring data such as device usage and error logs, user activity logs, detailed network alarms and customer care tickets. It is very important for network operators to quickly focus on more prevailing and repeating problems and to automate the process of root cause analysis and troubleshooting as much as possible. We design Giza to address such need of IPTV network operators.

Fig. 8 shows the overall architecture of Giza. The inputs to Giza are performance impairment events and the specific time frame of interest. For example, the events could be STB crashes recorded in device logs, or CPU spikes observed at CO devices, or customer complaints recorded in tickets, to name a few. Given the input, Giza first performs a multi-resolution analysis and eliminates areas of network locations that do not have significant observation of the input symptom events. This is achieved by a hierarchical heavy hitter detection component. The locations may be at any aggregation level in the DSLAM, CO, Metro and Region hierarchy. Focusing on the locations where symptom events are dominant greatly reduces the amount of data processing required for later steps. Next, Giza explores a wide range of other event series extracted from various system logs and performance monitoring data and identifies the ones that are correlated with the symptoms using a statistical correlation discovery component. This is done at an appropriate spatial granularity in which the symptom events and the diagnostic events can relate. Furthermore, Giza applies a novel causality discovery approach to deduce the causality graph and identifies the potential root causes. The output of Giza is a causality graph that has tremendous value in the troubleshooting efforts of network operators.

## 3.2 Hierarchical Heavy Hitter Detection

In this subsection, we present the design of our hierarchical heavy hitter detection component. The goal of this component is to identify spatial locations where a given performance impairment is prevalent and recurring, and prune the remaining locations, so as to reduce the complexity for the subsequent phases.

Detecting heavy hitters (i.e., locations that manifest significant occurrences of the symptom network event) in a single dimensional data typically involves setting an appropriate threshold. For example, if crash on STBs is the symptom event, then heavy hitter STBs can be defined to be those which crash more than $k$ times in an hour. However, considering the intrinsic hierarchical structure of an IPTV network, a proper definition of heavy hitter should include not only the event frequency (temporal property) but also the density concentration (spatial property). For example, a DSLAM with

10 STBs that experiences 1000 crashes is more significant than a DSLAM with 1000 STBs having the same number of crashes in the same period. A concentration reported at a lower level in the hierarchy provides more specific information, hence is more valuable, than a heavy hitter reported at a higher level. Bearing these design considerations in mind, we next present our significance test for a hierarchical heavy hitter.

**Significance test.** Given a level of hierarchy in Fig. 2 at which the detailed symptom events are defined, for example, STB level for raw STB crash events, or CO level for router CPU anomalies observed at local distribution office, we need to determine whether any aggregation at a higher level location is significant for the given symptom events. Consider a two-level hierarchy in which the lower layer has $n$ different locations. A specific top layer location has $m$ children locations, which have a total of $c$ events. We need to test whether $c$ is significant. Let $x_i$ be the number of events associated with each of the lower level locations $i$ ($i = 1, 2, ..., n$). We define $e = mean(x_i)$ and $v = variance(x_i)$ as the mean and variance for $x_i$ respectively.

Under the null hypothesis that the $m$ children are drawn independently and uniformly at random among lower level locations, the total number of event count of the $m$ children, $c$, should have mean and variance of $\mathbf{Exp}(c) = m \cdot e$ and $\mathbf{Var}(c) = m \cdot v$. If $m$ is reasonably large, $c$ should approximately be Gaussian distributed according to the central limit theorem (we will validate this in Section 4). It follows that the value of $c$ is considered significant if

$$\frac{(c - m \cdot e)}{\sqrt{m \cdot v}} \geq T,$$

where $T$ is the threshold corresponding to the desired false positive ratio. For standard Gaussian distribution, using values of $T$ as 1.64, 1.96, or 2.33 provides a distribution tail corresponding to 5%, 2.5%, or 1%, respectively.

When $m$ is too small, the central limit theorem no longer applies. In this case, we apply one-side Chebyshev inequality [29] (or Cantelli's inequality) to determine the heavy hitters. Specifically, under the null hypothesis that the $m$ children are drawn identically and independently from the same distribution with mean $e$ and variance $v$, we have

$$Prob(\frac{(c - m \cdot e)}{\sqrt{m \cdot v}} \geq k) \leq \frac{1}{(1 + k^2)}.$$

We can use 4.36, 6.25, or 9.95 as the threshold value of $k$ to achieve an expected false positive ratio of 5%, 2.5%, and 1%, respectively.

When applying heavy hitter detection, we start at the lowest possible spatial level for which the symptom event is applicable, and follow the hierarchy upward. This allows us to detect any significant concentration of symptom events at as precise spatial granularity as possible.

## 3.3 Event-series Formation

Once the heavy hitters are identified, our next step is to understand the performance issue of interest and to troubleshoot for the root cause. We do so through our correlation and causality analysis with other data sources. In this subsection, we describe the construction of the data time series that will feed to the subsequent analysis.

Given, either symptom events or diagnostic events that we hope to find dependency to the symptom ones, we construct a fixed-interval binary time series as described in NICE [23]. A "1" in the time series indicates the presence of the event in the corresponding time interval and "0" indicates absence. We follow the same principle as in [23] in choosing the interval length that takes into account time inaccuracy and the delayed impact due to event propagation or timers.

For an IPTV network, we need to aggregate the low-level event-series at different higher level in the spatial hierarchy. We define two aggregation constructions which will be applied in different scenarios in the paper: (i) *union* and (ii) *concatenation*. In event-series union, we superpose multiple event-series of the same length (e.g., one for each child location in the hierarchy) and apply either element-wise OR or element-wise SUM. In event-series concatenation, we append the multiple event-series one after another to form a longer event-series.

When comparing event-series for correlation and causality analysis, we need to consider the spatial levels of the symptom and diagnostic events. When both event-series are at the same level, we may directly apply pair-wise correlation analysis. An example is to test STB crashes against user channel changes on the same STB. Another example is to test STB crashes against user channel changes within the same DSLAM, in which both events-series are first aggregated (using union) into DSLAM level event-series and then compared against each other. When two event-series are at different spatial levels, we first aggregate (using concatenate) the lower-level event-series into the matching higher-level series, and then replicate the higher-level series multiple times so that it has the same length as the concatenated one, and finally compare the two extended time series. An example of this case is to test for correlation between STB crashes and CPU anomalies on the CO routers with which the STBs are associated. This ensures that the result of our correlation analysis, described next, is meaningful.

## 3.4 Statistical Correlation Detection

Troubleshooting a symptom event often starts with identifying what other events took place at around the same time and might potentially have had an impact on the symptom. Such co-occurrence based approaches, albeit conceptually simple, may catch many events that co-occur merely by coincidence, hence are ineffective due to high false positives. In Giza, we use a statistical correlation based approach for correlation analysis. In particular, we adopt the circular-permutation-based correlation test proposed in NICE [23] for pair-wise correlation test. Comparing to other statistical correlation tests, the advantage of NICE lies in the fact that it takes into account the auto-correlation likely to be present in both symptom and diagnosis event-series; auto-correlation, if present, can have significant influence on correlation score. Next, we briefly describe how NICE works; for more details, please refer to [23].

Let $r_{XY}(t)$ be the Pearson's correlation coefficient between event-series $X$ and the circularly shifted version of event-series $Y$ at lag $t$. For each lag $t \in [0, N)$, where $N$ is the number of samples in each event-series, $r_{XY}(t)$ is defined as

$$r_{XY}(t) = \frac{\sum_{i=1}^{N}(X_i - \mu_X)(Y_{(i+t) \bmod N} - \mu_y)}{(N-1)\sigma_X \sigma_Y},$$

where $\mu_X$ and $\mu_Y$ are the means of $X$ and $Y$, and $\sigma_X$ and $\sigma_Y$ are the standard deviations, respectively.

The circular shifting eliminates the cross-correlation between the two event-series and preserves auto-correlation within each event-series. We can thus use $\{r_{XY}(t)\}$ ($t \in [0, N)$) to establish a baseline for the null hypothesis that two event-series have no significant cross-correlation. In order to test the hypothesis, we apply Fisher's $z$-transform as follows.

$$z(t) = \frac{1}{2} ln \left[\frac{1 + r_{XY}(t)}{1 - r_{XY}(t)}\right].$$

We note that $\{z(t)\}$ is asymptotically Gaussian for sufficiently large $N$. Given this, we define the correlation score as

$$score = \frac{z(0)}{\sigma_z} = \frac{z(0)}{\mathrm{stddev}(\{z(t)\})},$$

where $\mathrm{stddev}(\{z(t)\})$ denotes the sample standard deviation of $z(t)$'s. A correlation score is considered significant if it falls outside of the

$[-2.5, 2.5]$ range. With $z$ asymptotically Gaussian, this yields a low false positive ratio of around 1%.

## 3.5 Causality Discovery

Through the correlation analysis above, we can obtain a list of correlated event-series. The next step is to organize them into a causality graph that provides the causal relationship among different symptom and diagnostic events. We generate a directed edge from event-series $X$ to event-series $Y$ in the causality graph if: (i) $X$ and $Y$ have significant statistical correlation; (ii) $X$ precedes $Y$ in a statistical sense; and (iii) $X$ and $Y$ are not related via other events.

There are many techniques available in the data mining literature to discover statistical causal dependencies, such as linear regression and partial correlations. However, they are not directly applicable in our context due to two main problems: (i) many event-series pairs exhibit strong cross-correlations among each other, which causes regression and partial correlation coefficients to be inaccurate (this is commonly known as the problem of multi-collinearity), (ii) regression when used to distinguish cause and effect produces erroneous edge directionality in noisy data. We address these problems by first identifying edge directionality using a novel statistical lag correlation method, and then applying an edge reduction algorithm to eliminate spurious correlations.

### *3.5.1 Edge Directionality using Lag Correlation*

The key idea is to use timing information to test whether one event-series statistically precedes the other. Given two event-series $X$ and $Y$, we generate samples of the Pearson's correlation coefficient $r_{XY}$ by circularly shifting $Y$ to different lags with respect to $X$, and computing the cross-correlation coefficient between $X$ and the shifted $Y$. By comparing positive lag correlations with negative ones, we identify if $Y$ statistically occurs before $X$ (positive lags dominate over negative lags), or $X$ statistically occurs before $Y$ (negative lags dominate over positive lags), or the directionality is inconclusive (positive and negative lags are comparable).

Focusing on the data in IPTV network in which we are interested, there are two issues which we need to be particularly careful about. First, the timing information in event timestamps is not 100% reliable due to low granularity of periodic polling, imperfect clock synchronization, non-deterministic event propagation and recording delays. This precludes us from using any inference techniques that rely on precise timing such as in Sherlock [4]. Second, many event-series exhibit strong auto-correlation structure, especially at small lags – this smooths out the shape of the cross-correlation graph, making inference difficult.

To solve these two problems, we start with the $z$-scores at different lags in our correlation analysis, which we have already computed in the correlation significance test, and apply the following either of the two heuristics.

**1. Comparing the maximum in a range of positive and negative lag correlations.** If the maximum in the positive lag range $max(z(k_1), .., z(k_2))$ is greater than the maximum in the negative lag range $max(z(-k_3), .., z(-k_4))$, meaning that the correlation score is higher when $Y$ is shifted in the positive direction, we deduce that $Y$ statistically precedes $X$. Similarly, if the maximum in the negative lag range is greater than that in the positive lag range, we deduce that $X$ precedes $Y$ statistically. If the maximum in both ranges are close (within a threshold), then the directionality is inconclusive and we leave the edge unmarked. This metric is useful when there is a strong auto-correlation at small lags.

**2. Statistical change detection between the ranges of positive and negative lag correlations.** Instead of comparing the maximum, we may also compare the mean of the distributions in positive and negative lag ranges. Let *PL* and *NL* denote the sample score sets for positive lags and for negative lags, respectively. Let $\mu_p$ and $\mu_n$ be the mean of the distributions respectively. Then the standard deviations of $\mu_p$ and $\mu_n$ are denoted by $\sigma_p = \frac{1}{\sqrt{k_2-k_1}}\sigma_z$ and $\sigma_n = \frac{1}{\sqrt{k_3-k_4}}\sigma_z$, respectively.

When comparing the means of two distributions, the difference of the means is $\mu_p - \mu_n$ and the variance is the sum of individual variances. Hence, the standard deviation is $\sqrt{\sigma_p^2 + \sigma_n^2}$. We compute the statistical change score as

$$\frac{\mu_p - \mu_n}{\sqrt{\sigma_p^2 + \sigma_n^2}}$$

According to the central limit theorem, the range $[-2.5, 2.5]$ can be used as the score range in which we cannot say statistically which lag dominates with 99% accuracy. If the change score is greater than 2.5, then positive lag dominates. If the change score is less than -2.5, then negative lag dominates.

The above two approaches often find consistent results, as we will see in Section 4.2.

### *3.5.2 Edge Reduction using $\ell^1$ Norm Minimization*

Now that we have obtained the partially directed correlation graph built from statistical lag correlations, our next step is to prune spurious edges if any. A statistical correlation between two event-series $X$ and $Y$ is defined to be spurious if the correlation is actually due to a third variable $Z$. In other words, when the events corresponding to $Z$ are removed, the correlation between $X$ and $Y$ becomes insignificant. For example, if the correlation between packet loss and router CPU anomalies is due to link down, then we can eliminate the edge between packet loss and router CPU anomalies.

The key idea is to apply statistical regression and preserve edges in which the regression coefficients are significant. We use the symptom event-series as the predictee and each diagnostic event-series that has a directed edge towards the symptom as the predictors. A significant regression coefficient indicates that the correlation between two event-series is non-spurious and we keep those edges in the causal graph. On the other hand, an insignificant regression coefficient means that the dependency between two event-series is not strong when other event-series are considered, hence we eliminate the corresponding edge from the causal graph.

One challenge with this edge elimination approach stems from scale since a large number of event-series in the correlation graph means regression coefficients for each predictor would be small and identifying the threshold for significance becomes non-trivial. In practice though, we expect only a few event-series to have significant causal relationship with the symptom. In other words, we expect the vector of regression coefficients to have only a small number of large values. To effectively identify these coefficients, we propose a new method using $\ell^1$ norm minimization with $\ell^1$ regularization which has the ability to find a sparse solution (achieving the approximate $\ell^0$ norm minimization) [15].

The method works as follows. Let $y$ be the vector of predictee event-series. $X_{m\ x\ n}$ is matrix of predictor event-series with $m$ being the length of each event-series and $n$ being the total number of predictors. Note that $X$ is comprised of only those event-series that statistically occur before $y$. We formulate the $\ell^1$ norm minimization problem as:

$$\text{minimize } ||y - \beta X||_1 + \lambda ||\beta||_1$$

where $\beta$ is a vector of regression coefficients and $\lambda \in [0,1]$ is the regularization parameter. We reformulate the above minimization problem into the following equivalent linear programming (LP) problem:

$$\begin{array}{ll} \text{minimize} & \lambda \sum_i u_i + \sum_j v_j \\ \text{subject to} & y = \beta X + z \\ & \mathbf{u} \geq X, \mathbf{u} \geq -X \\ & \mathbf{v} \geq z, \mathbf{v} \geq -z \end{array}$$

**Input:** A list of $k$ event-series

**Output:** Directed causal graph $G = (V, E)$ where $V$ is the set of event-series with $|V| = k$ and an edge $(i, j) \in E$ indicates $i$ is a cause of $j$

**Algorithm:**

1. Initially, $E = \{\}$

Edge directionality using lag correlation

2. $\forall i \in V$
3. $\quad \forall j (j \neq i) \in V$
4. $\quad\quad$ if LagCorr($i$, $j$) is positive significant
$\quad\quad$ /*LagCorr($x$, $y$) is computed by fixing $x$ and shifting $y$ */
5. $\quad\quad\quad$ then $E = E \cup (j, i)$

Edge reduction using $\ell^1$ norm minimization

6. $\forall i \in V$
7. $\quad X = \{j | j \in V$ and $(j, i) \in E\}$
8. $\quad \beta =$ L1NormRegression($X$, $i$)
$\quad$ /* beta: regression coefficients between $i$ and all $j \in X$ */
9. $\quad R = \{j | \beta_{i,j}$ is insignificant $\}$
10. $\quad \forall j \in R$
11. $\quad\quad$ Remove $(i, j)$ from $E$

**Figure 9: Causal discovery algorithm.**

To build the entire causal graph with $N$ event-series, we run LP for each event-series as $y$. The predictors for each $y$ is identified using the lag correlations. We now show the complete causal discovery algorithm in Fig. 9.

# 4. GIZA EXPERIENCES

In this section, we present Giza validation using real data collected from the IPTV network and demonstrate its effectiveness in troubleshooting network issues. First, we demonstrate that our assumption about Gaussianity is valid in hierarchical heavy hitter detection. Second, we show that our causality algorithm that accounts for auto-correlation and multi-collinearity performs better than the state-of-art causality algorithm in WISE [28]. Third, we describe our experiences in applying Giza on diagnosing customer trouble tickets and video quality alarms, and comparing our results with the ground truth information provided by network operators. Finally, we present case study where Giza has been applied to discover the causal graph and previously unknown dependencies in the provider network.

## 4.1 Validating Gaussianity for HHH

We use the Q-Q (quantile-quantile) plot to validate the Gaussian assumption we made in hierarchical heavy hitter detection. Q-Q plot is a graphical method to identify whether there exists a statistical difference between one probability distribution and another. For our validation, we compare the normal distribution constructed from our hypothesis test with the event count distribution at various spatial resolutions. If the two distributions match perfectly, then the Q-Q curves should approximate the straight diagonal line.

Fig. 10 shows one example for a particular type of STB crash events (native crash) at three different spatial resolutions. It can be observed that all three curves largely approximate a straight line with the exception of a few outliers at the distribution tail. We also have plotted Q-Q plot for all the other data sources and have observed similar level of matches. This confirms that the Gaussian approximation due to Central limit theorem works reasonably well in our data. The deviation shown in the tail part in Fig. 10(a) and (b) however suggests that there indeed exists a pattern of spatial concentration – some COs (or Metro's) have observed a higher number of STB crashes that can be explained by simple aggregation variance. Those are the genuine heavy hitters that can be identified through our hierarchical heavy hitter detection scheme. Fig. 10 also shows in dotted line where the thresholds (at 1% from distribution tail) for the hierarchical heavy hitter detection are. All data points to the right of the dotted lines are considered heavy hitters.

| | **Total edges identified** | **Edges correctly identified** |
|---|---|---|
| WISE (Partial Correlation) + Linear Regression | 4903 | 71.9 % |
| $\ell^1$ Norm + Statistical Change Lag Correlation | 1103 | 84.4 % |
| $\ell^1$ Norm + Maximum Lag Correlation | 1125 | 85.3 % |

**Table 4: Comparison of causal discovery algorithms.**

## 4.2 Comparing Causal Discovery Algorithms

Next we show through comparative evaluation that considering multi-collinearity is important for discovering causal dependencies. We compare Giza with WISE [28] which are multi-variate analysis techniques. We do not present comparison with Sherlock [4], Orion [7], or NICE [23] because they only rely on pair-wise correlations. A qualitative comparison of all these techniques is provided in Section 5.

To compare Giza and WISE, we use one-week worth of syslog data aggregated at VHO and SHO resolutions. We consider 80 different VHOs and SHOs in which we construct 1318 different types of event-series including layer-1 alarms (Ethernet, SONET, port errors), protocol state changes (MPLS, OSPF, BGP, PIM, SAP, SDP), link flaps, configuration changes, and CPU activities. To set up ground truth, we have resorted to domain experts and have constructed 482 causal rules (indicated by the presence of edges and their directionality in causal graph). An example rule is "a link down causes OSPF protocol to change state". We consider these rules as a subset of the causal relationships that should be identified by the causal discovery algorithm – the complete set of the causal relationships requires perfect domain knowledge and is nearly impossible to obtain.

Table 4 compares the causal discovery result generated by the three algorithms: (i) WISE partial correlations plus linear regression (a widely used approach in data mining, such as in [9, 10]), (ii) $\ell^1$ norm minimization combined with lag correlation using statistical change detection, and (iii) $\ell^1$ norm minimization combined with lag correlation using maximum in the ranges. The latter two are what we have described in Section 3.5.2. In the cases where we cannot conclusively determine the causal direction of an identified correlation, we construct two directional edges between the pair of events (two different rules). We determine the accuracy of the above algorithms by comparing their results with our subset of ground truth. An edge (out of the 482 rules) is considered a match if both its existence and its directionality have been correctly identified. We observe that either of our approaches significantly outperforms the partial correlation and linear regression approach in accuracy. Since accuracy solely does not reflect the performance of the inference algorithm, (for example, a dummy algorithm that blindly mark all edges in the causal graph would achieve 100% accuracy), we also need to consider the false positives. Since we do not have the complete ground truth, we can use the total number of edges identified as a reference point. We observe that partial correlation and linear regression identifies more than four times of the edges while still achieving around 13% less accuracy compared to our approaches. This demonstrates the strength of our approaches – the high degree of multi-collinearity of the data has been properly accounted for.

The two lag correlation and $\ell^1$ norm minimization based approaches have highly similar performance. We include both in Giza as method for causal dependency discovery for completeness.

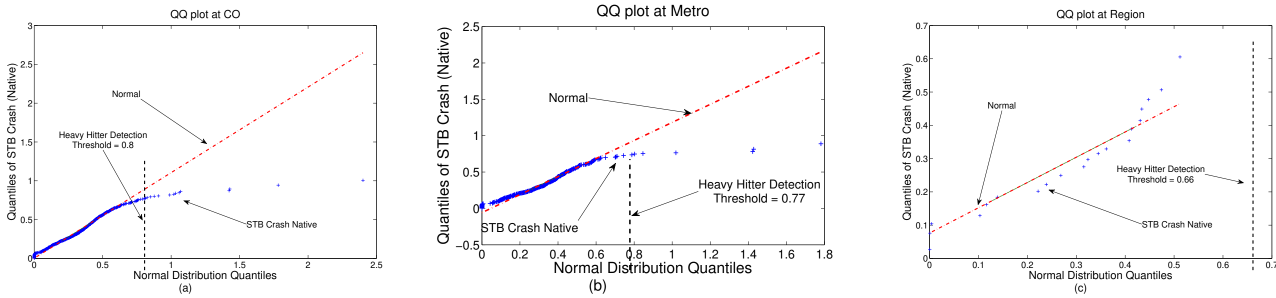


**Figure 10: Q-Q plots at spatial resolutions of CO, Metro and Region for STB crash events of type native.**

## 4.3 Validation by Operational Experiences

We describe our experiences in applying Giza on the data collected from the IPTV network (both the provider network and the customer home network) over one month period in 2008. We demonstrate how can we apply the suite of techniques in Giza to help operations in IPTV network.

In this experiment, we consider the performance issues reported in the customer trouble tickets (Section 4.3.1) and video quality alarms (Section 4.3.2) as our input symptom event-series. These are the most direct measures reflecting the IPTV performance impairments. We apply Giza in diagnosing these symptoms. We find Giza particularly useful in prioritizing these performance issues as it offers a big-picture view on the scale and frequency of the problem. This is an important factor in large-scale network services such as IPTV – conducting detailed diagnosis and troubleshooting of every performance problem would exhaust service and operation resource and become infeasible. Looking at the result, Giza identifies some expected correlation such as home network faults and user interface problems correlating strongly with some trouble tickets. Quite interestingly, a small number of provider network events have also been identified correlating with certain type of trouble tickets. To further understand this, we drill in our investigation and apply Giza to identify causal dependencies among provider network events. Our discovery is presented in Section 4.4.

### *4.3.1 Customer Trouble Tickets*

We focus on customer trouble tickets logged by the IPTV network operators. Each ticket indicates performance related issue reported by a customer and the time it is reported. Each problem reported is categorized into one of the ten types of performance issues: live TV video, audio, digital video recording (DVR), equipment, high definition, video-on-demand (VoD), parental control, program guide, remote control and games. We create a symptom event-series of interest for each of the above types and correlate the defined symptom event-series with other event-series in home networks such as STB crash, STB reset and RG reboots, as well as event-series constructed from the syslogs of the provider network devices.

We first observe a high degree of sparsity for each type of trouble ticket – a small portion of customers have communicated with the IPTV provider about performance issue that results in a ticket created. Moreover, there is little repetition of tickets at each individual customer. We cannot directly apply the correlation and causality analysis for individual customers as it does not provide sufficient statistical significance. Both observation call for a multi-(spatial) resolution heavy hitter detection approach, which allows us to quickly focus on a spatial region in which the given type of trouble ticket is prevalent, applicable for correlation and causality analysis. Fortunately, such capability is handily available in Giza.

**Hierarchical heavy hitter detection results.** Table 5 shows at the four different levels of spatial aggregation and for ten different types of customer trouble tickets, the customer coverage ratio – the percentage of the total number of customers in the identified heavy hitter locations (top table), and the symptom coverage ratio – the percentage of the total number of symptom events in the identified heavy hitter locations (bottom table). We have used the detection thresholds that corresponds to 1% at the distribution tail for Table 5. Note that we use 0 to denote the case in which no heavy hitter can be identified at the given spatial level.

We observe that for the ticket types of DVR, equipment, and remote control, there is hardly any heavy hitters identified, indicated by the extremely low number of customer coverage. This makes sense since these problems should be independent of any network components that the DVR, STB, or remote control associates to, and hence are expected to be distributed evenly at random over spatial locations. For the rest types of customer ticket, there are some level of spatial concentration observed. In these cases, Giza is able to identify a small number of heavy hitter locations. Comparing the symptom coverage ratios to their corresponding customer coverage ratio, we find that those heavy hitter locations indeed experience the symptom problem much more severely – it may be due to a faulty network component, an undesirable protocol condition, etc., at the specific heavy hitter locations. With the help of hierarchical heavy hitter detection in Giza, operators can then quickly focus on these few locations to investigate the root causes in detail. There are sufficient concentration of symptom problems at these locations, to which statistical correlation and causality analysis in the Giza tool suite can then be applied.

**Correlation results.** We now try to discover dependency to the various types of customer ticket using the correlation engine in Giza. We first create a composite time series for each type of trouble tickets at each heavy-hitter location – the value of the time series being the total count of symptom tickets associated in the sub-tree of the heavy-hitter location during the time bin, or in the binary version, the value being the predicate of whether there is at least one symptom ticket in the sub-tree during the time bin. Since customer tickets are entered manually into the system, the time reported on the ticket can be significantly delayed from the time at which performance problem has started (by from few minutes to few hours). In our experiments, we use a time lag of four hours as the correlation margin. That is, we look for co-occurrences between trouble tickets and other event-series within a four-hour window. The correlation algorithm then outputs event-series pairs that manifest a strong statistical correlation.

Fig. 11 illustrates some strong statistical dependencies between different types of customer trouble tickets and various STB Crash, STB Reset and events extracted from provider network syslogs. An edge in the graph indicates the presence of strong statistical correlation between the pair of event series. For example, trouble tickets related to live TV video, digital video recording (DVR), video on demand (VoD), and games have strong statistical correlations with both STB crashes and resets. Some of the correlations can be explained by user's response in trying to resolve the service problem – considering the case when a STB crash causes service interruption

| Resolution | Live TV | Audio | DVR | Equipment | HD | VoD | Parental Control | Program Guide | Remote Control | Games |
|---|---|---|---|---|---|---|---|---|---|---|
| DSLAM | 0.004 | 0.03 | 0.0003 | 0.0001 | 0.02 | 0.03 | 0.03 | 0.001 | 0.009 | 0.08 |
| CO | 0 | 0.21 | 0.0003 | 0.002 | 0.04 | 0.42 | 0.44 | 0.35 | 0 | 0.89 |
| Metro | 0 | 0 | 0.0003 | 0.002 | 0.04 | 0.59 | 0 | 0.39 | 0 | 0.75 |
| Region | 0 | 0 | 0 | 0 | 0 | 0.60 | 0.43 | 0.55 | 0 | 0.19 |
| DSLAM | 0.22 | 0.87 | 0.04 | 0.01 | 0.52 | 3.55 | 3.39 | 0.14 | 0.44 | 60.27 |
| CO | 0 | 1.53 | 0.01 | 0.02 | 0.31 | 3.90 | 3.23 | 3.24 | 0 | 41.09 |
| Metro | 0 | 0 | 0.01 | 0.02 | 0.31 | 4.44 | 0 | 3.38 | 0 | 15.07 |
| Region | 0 | 0 | 0 | 0 | 0 | 3.55 | 2.24 | 4.04 | 0 | 2.74 |

**Table 5: Customer coverage ratio (top) and symptom coverage ratio (bottom) at heavy hitter locations for customer trouble tickets at different spatial levels.**

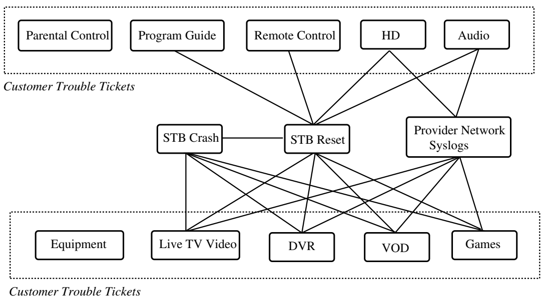


**Figure 11: Dependencies between customer trouble tickets and other events in home network and provider network.**

or performance degradation, customer resets the STB in the hope of clearing the problem; when this fails to work, the customer calls service center, triggering a service ticket created; operators consequently perform certain diagnosis routine remotely, which may produce more STB resets/reboots. While such correlation may be less interesting from operator's point of view, the correlation for each subtype of STB crashes (captured in the crash logs) would offer insight for debugging STB software. Moreover, we also discover some significant correlations of several type of tickets (such as HD, Audio, Live TV video, and Games) to the provider network side events (such as link downs, SAP/SDP state changes, multicast neighbor loss, and MPLS path re-routes) – we will examine the causal graph of these network side events in Section 4.4. Knowing these dependencies allows us to better understand the impact of various network side issue on customer perceived performance. It also helps in identifying signature for network problems, which customer care personnel needs to be informed of, so that they can communicate better to customers complaining about certain type of problems.

**Validation against mitigation action.** Customer care tickets also record the mitigation actions taken in resolving the reported problem. Although there are many different mitigation actions, for simplicity, we classify them into three main categories: user interface related (e.g., adjusting TV or volume settings, replacing remote control), home network related (e.g., residential gateway resets, replacing set top boxes), and provider network related (e.g., maintenance or outage).

Table 6 shows how the mitigation actions for each type of trouble tickets are distributed (in percentage) across different categories. We find the result very much echos our statistical correlation result using Giza. For example, trouble tickets about parental control are typically resolved by explaining the service features to the customer (87% in user interface category). Thus they do not have any significant correlation with either home network events or provider network events. Tickets on remote control problem are least likely to relate to a provider network issue (1.2%). Consistently, Giza reports no correlation between the two. On the other hand, video on demand tickets have many escalation to network care operators (39%), which supports the high correlation score identified in Giza.

| | User Interface | Home Network | Provider Network | Others |
|---|---|---|---|---|
| Live TV Video | 13 | 70 | 5.5 | 11.5 |
| Audio | 10.1 | 69 | 5.5 | 15.4 |
| DVR | 14 | 75 | 4.5 | 6.5 |
| Equipment | 0 | 91 | 6 | 4 |
| High Definition | 17 | 57 | 6 | 20 |
| Video on Demand | 28 | 17.2 | 39 | 15.8 |
| Parental Control | 87 | 8.3 | 4.1 | 0.6 |
| Program Guide | 19.1 | 58 | 17 | 5.9 |
| Remote Control | 70 | 25 | 1.2 | 3.8 |
| Games | 60.4 | 0.4 | 29 | 10.2 |
| Total | 21.1 | 56.3 | 5.3 | 17.7 |

**Table 6: Trouble ticket characterization by problem (row) and mitigation actions (column). Each entry represents a percentage.**

### 4.3.2 *Video Quality Alarms*

In this subsection, we focus on symptom series from alarms generated by the video quality monitors deployed inside the IPTV service provider network. An alarm indicates an impairment in video quality due to problems such as excessive delay factor (DF), media loss rate (MLR), video stream outage, transport stream outage, IP flow bit rate thresholding crossing, transport stream synchronization errors and transport stream bit rate thresholding crossing. The video quality monitors are deployed at VHOs. Each VHO is responsible for a geographical region, which is the highest spatial level defined in Giza. We consider each type of alarms as a symptom event-series and correlate it with other event-series extracted from router syslogs at the VHO and the trouble tickets from customers that are associated with the VHO.

**Correlation results.** We perform the correlation analysis for data collected over one month. The correlation time window is set as 60 seconds. When correlating alarms data with customer tickets, we use a time lag of four hours. We observe strong statistical correlations between video quality alarms and syslog events in the provider network such as configuration changes, SAP port state changes, SDP bind status changes, BGP session downs, PPP / RSVP / SONET link downs, multicast neighbor loss, MPLS path re-routes and layer-1 link flaps. However, most types of the video quality alarms do not statistically correlate with customer trouble tickets. This is partially because the alarms from the monitoring device are too low-level – the intention of the alarms is for monitoring the health of video distribution network as opposed to monitoring customer perceived performance. The alarmed short term packet losses or delay jitters can be automatically repaired by FEC or retransmission of RUDP without introducing interruption on decoding of the video stream, hence have no impact to customers. By looking at the significance of the correlation result, we can easily distinguish the alarms that would produce severe performance impairment of the video stream delivered to customers from those that would not. For example, the alarm on long term (24 hour) excessive media loss rate are likely due to a persistent video feed problem and is identified to be correlated with customer complaints. We have also validated the discovered dependencies of video quality alarms on network events (extracted from router syslogs) with network operators. We will further investigate the causal relationships among the network events in the next section.

| Number of event-series | Pairs to correlate | Strong pair-wise correlations | $\ell^1$ + statistical change lag correlation | $\ell^1$ +max lag correlation |
|---|---|---|---|---|
| 1318 | 867,903 | 3352 | 960 | 972 |

**Table 7: Provider network syslog correlation and causality results.**

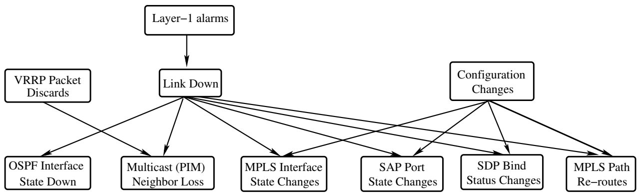


**Figure 12: Causal graph between certain network syslog event-series.**

## 4.4 Case Study: Provider Network Events

As discussed in Section 4.3.1, we observe statistical correlation between certain provider network syslog events (such as link downs, SAP/SDP state changes, multicast neighbor loss, MPLS path re-routes) and different types of trouble tickets. In this section we further investigate statistical dependencies of these events on other provider network events.

The provider network syslog data contains a diverse set of events. Creating a separate time-series based on each unique message type results in the creation of hundreds of individual event-series from syslogs at every VHO or SHO. This provides us with a perfect opportunity to apply our statistical lag correlation and $\ell^1$ norm minimization algorithms, to analyze how well they cope with the scale.

We use one week worth of data to perform the causality analysis on 80 VHO and SHO routers within the provider network. We focus on four symptom event-series that demonstrate strong correlations with trouble tickets: (i) link down, (ii) SAP state changes, (iii) multicast neighbor loss, (iv) MPLS path re-routes.

**Correlation and causality discovery results.** Table 7 shows the correlation and causality discovery results for the provider network syslog data. There are a total of 1318 syslog event-series at several VHO and SHO locations. The total number of pairs to correlate is 867,903 ($= \frac{1318*1317}{2}$), out of which 3352 have strong correlation. For correlation, we perform the analysis for events that occur at the same VHO or SHO. This achieves a reduction of 99.6% compared to the number of pairs for correlation. $\ell^1$ norm minimization along with lag correlation further reduces the number of pairs and achieves around 99.89% reduction.

A part of the causal graph reported by Giza is shown in Fig. 12. As can be seen in the figure, the root cause for link downs is layer-1 alarms. Link downs in turn cause SAP port state changes, OSPF interface state changes, SDP bind status changes, MPLS interface state changes, and multicast neighbor loss. These conclusions are consistent with domain knowledge, and with how the network technologies were designed to operate. The advantage of Giza is that it can discover the causal graph with minimal domain knowledge.

**Findings.** For the week analyzed, MPLS path re-routes had their major root cause as in configuration changes. As would be expected, the strong statistical correlation is observed during the maintenance time-window - the time during which network operations personnel are executing planned maintenance activities. Multicast neighbor losses demonstrated strong correlations with both link downs and configuration changes.

We also observed an interesting previously unknown dependency between multicast neighbor loss and VRRP packet discards. VRRP stands for virtual router redundancy protocol and is used for increased network reliability. It is achieved by advertising a virtual router as a default gateway to the hosts instead of one physical router. Two or more physical routers are configured to act as a virtual router. When the primary physical router is down, the back-up router takes over.

The strong statistical correlation between VRRP packet discards at VHO and SHO and multicast neighbor loss was due to packets looping within the network that causes the multicast protocol to timeout and resulting in neighbor loss. The behavior was more prevalent within the SHO and at VHOs closer to the SHO. Interestingly, when VRRP packet discards have a temporal correlation (or join) with multicast neighbor loss, we do not observe link downs. Thus, the packet looping does not appear to have been caused by link failures. We are currently collaborating with the operations team to analyze this scenario to further understand the behavior.

# 5. RELATED WORK

In this section, we present related work.

**IPTV, P2P, VoD Analysis.** Cha *et al.* [6] present the first large-scale study of user behaviors in IPTV system. They characterize user behaviors such as channel popularity and dynamics, viewing sessions, geographic locality and channel surfing probabilities. Qiu *et al.* [24] develop a model that captures the dynamics of channel popularity in IPTV network. There are many previous measurement studies on VoD [8, 19, 31], P2P IPTV [18, 25].

Our study, on the other hand, focuses on characterizing performance impairments and faults in a large-scale IPTV system. We believe, this is the first characterization study aiming to understand the performance issues in large-scale operational IPTV networks.

**Hierarchical Heavy Hitter (HHH) Detection.** There has been a great deal of work on finding heavy hitters at multiple aggregation points in network traffic data. Their goal is to identify source-destination prefixes at multiple hierarchies that contribute to a large fraction of the total network traffic. Cormode *et al.* [12] was the first to extend the idea of heavy hitters to multiple dimensions. AutoFocus [16] presents several heuristics to detect interesting traffic clusters corresponding to anomalous traffic conditions. [13] presents online algorithms to identify approximate HHHs in one pass. [1, 32] propose algorithms to discover changes in hierarchical summaries.

The key difference of our significance test for detecting hierarchical heavy hitters is our ability to handle the diversity of distribution of spatial components across different aggregation levels.

**Network Troubleshooting using Statistical Analysis.** Recently, there has been an increasing interest in applying statistical analysis for network troubleshooting. The goal is given a symptom problem, identify the set of root-causes that can best explain the symptom. SCORE [22] applies bipartite graph to solve the fault diagnosis problem. Shrink [21] extends this model to deal with probabilistic settings. Sherlock [4] proposes a multi-level graph inference to learn the dependencies in enterprise networks. eXpose [20] learns communication rules in edge networks using spectral graph partitioning that is useful in monitoring and intrusion detection. WISE [28] is a what-if analysis tool that estimates the effects of possible changes to network configuration on service response times.

Yemini *et al.* [30] present an event correlation library that describes faults and the symptoms of faults using a codebook. NetDiagnoser [14] performs fault localization using Boolean tomography. Orion [7] uses delay spike analysis to discovery pair-wise dependencies in network traffic. NICE [23] focuses on troubleshooting undesirable chronic network conditions using Pearson's correlations. NetPrints [2] uses decision-tree learning for troubleshooting home network mis-configurations. [27] is a white paper that discusses recent research efforts at Alcatel Lucent for designing end-to-end diagnosis capabilities in IPTV. Causal modeling is an area of active research, with rich literature in data mining and machine learning [3, 5, 9, 11, 17, 26].

| Property | Sherlock | Orion | NICE | WISE | Giza |
|---|---|---|---|---|---|
| Auto-correlation | X | X | √ | X | √ |
| Multi-variate analysis | X | X | X | √ | √ |
| Multi-collinearity | X | X | X | X | √ |
| Automated edge directionality | √ | √ | X | √ | √ |
| Multi-resolution analysis | X | X | X | X | √ |

**Table 8: Troubleshooting Infrastructure Taxonomy.**

We provide a qualitative comparison of several recently proposed troubleshooting infrastructures in Table 8. As you can see, Sherlock and Orion focuses mainly on pair-wise correlation analysis and aims to automatically discover the directionality of the correlation. NICE addresses the auto-correlation within each event-series and reduces false alarms when discovering the correlation graph. WISE is the first to apply multi-variate correlation techniques. However, WISE does not address auto-correlation and multi-collinearity when discovering causal dependencies. This leads to lower accuracy as we show in Section 4. Giza addresses auto-correlation, goes beyond pair-wise analysis, handles multi-collinearity problem when performing the regression, discovers edge directionality automatically and performs multi-resolution analysis.

# 6. CONCLUSIONS

In this paper, we presented the first characterization study of faults and performance impairments in the infrastructure of a large IPTV service provider in North America. Our analysis spanned routers in the backbone to set top boxes (STB) and residential gateways (RGs) in home networks, hardware and software crashes to video quality impairments. To deal with the scale and heterogeneity of the IPTV network, we proposed and designed a novel multi-resolution data analysis approach termed Giza that enables fast detection and localization of problems. We also proposed novel techniques comprising of statistical lag correlations and $\ell^1$ norm minimization for effective and scalable causal discovery. Our experience with applying Giza in the IPTV network has been very positive. The infrastructure promises to be of immense value to IPTV network operators in automatically detecting and troubleshooting important performance issues.

## Acknowledgement

We thank Aldo Adriazola, Seungjoon Lee, and Sigcomm anonymous reviewers for their feedback. We are also grateful to the network operations and customer support teams of the anonymous IPTV service provider for their kind help on the data collection and the case study analysis. This work was supported in part by NSF grants CNS-0546720, CNS-0615104, and CNS-0627020.

# 7. REFERENCES

[1] D. Agarwal, D. Barman, D. Gunopulos, N. E. Young, F. Korn, and D. Srivastava. Efficient and effective explanation of change in hierarchical summaries. In *ACM KDD*, 2007.

[2] B. Aggarwal, R. Bhagwan, V. N. Padmanabhan, and G. Voelker. NetPrints: Diagnosing home network misconfigurations using shared knowledge. In *NSDI*, 2009.

[3] A. Arnold, Y. Liu, and N. Abe. Temporal causal modeling with graphical granger methods. In *ACM KDD*, pages 66–75, 2007.

[4] 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*, 2007.

[5] W. Buntine. Theory refinement on Bayesian networks. In *Proc. Uncertainty in artificial intelligence*, 1991.

[6] M. Cha, P. Rodriguez, J. Crowcroft, S. Moon, and X. Amatriain. Watching Television over an IP Network. In *ACM IMC*, 2008.

[7] X. Chen, M. Zhang, Z. M. Mao, and P. Bahl. Automating network application dependency discovery: Experiences, limitations, and new solutions. In *OSDI*, 2008.

[8] B. Cheng, L. Stein, H. Jin, and Z. Zhang. Towards cinematic internet video-on-demand. In *ACM EuroSys*, 2008.

[9] P. R. Cohen, L. A. Ballesteros, D. E. Gregory, and R. S. Amant. Regression can build predictive causal models. Technical Report UM-CS-1994-015, 1994.

[10] P. R. Cohen, D. E. Gregory, L. Ballesteros, and R. S. Amant. Two algorithms for inducing structural equation models from data. Technical Report UM-CS-1994-080, 1994.

[11] G. F. Cooper and E. Herskovits. A bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9(4):309–347, 1992.

[12] G. Cormode, F. Korn, S. Muthukrishnan, and D. Srivastava. Finding hierarchical heavy hitters in data streams. In *VLDB*, 2003.

[13] G. Cormode, F. Korn, S. Muthukrishnan, and D. Srivastava. Diamond in the rough: finding hierarchical heavy hitters in multi-dimensional data. In *ACM Sigmod*, 2004.

[14] A. Dhamdhere, R. Teixeira, C. Dovrolis, and C. Diot. Netdiagnoser: troubleshooting network unreachabilities using end-to-end probes and routing data. In *CoNEXT*, 2007.

[15] D.L.Donoho. For most large underdetermined systems of equations, the minimal l1-norm near solution approximates the sparsest near-solution. In *http://www-stat.stanford.edu/donoho/Reports/*, 2004.

[16] C. Estan, S. Savage, and G. Varghese. Automatically inferring patterns of resource consumption in network traffic. In *ACM Sigcomm*, 2003.

[17] C. W. J. Granger. Investigating causal relations by econometric models and cross-spectral methods. In *Econometrica*, 1969.

[18] X. Hei, C. Liang, J. Liang, Y. Liu, and K. W. Ross. A measurement study of a large-scale P2P IPTV system. *IEEE Transaction on Multimedia*, 2007.

[19] Y. Huang, T. Z. Fu, D.-M. Chiu, J. C. Lui, and C. Huang. Challenges, design and analysis of a large-scale P2P-VoD system. In *ACM Sigcomm*, 2008.

[20] S. Kandula, R. Chandra, and D. Katabi. What's going on? learning communication rules in edge networks. In *Sigcomm*, 2008.

[21] S. Kandula, D. Katabi, and J.-P. Vasseur. Shrink: A tool for failure diagnosis in IP networks. In *MineNet*, 2005.

[22] R. R. Kompella, J. Yates, A. Greenberg, and A. C. Snoeren. Detection and localization of network blackholes. In *Infocom*, 2007.

[23] A. Mahimkar, J. Yates, Y. Zhang, A. Shaikh, J. Wang, Z. Ge, and C. T. Ee. Troubleshooting chronic conditions in large IP networks. In *ACM CoNEXT*, 2008.

[24] T. Qiu, Z. Ge, S. Lee, J. Wang, J. Xu, and Q. Zhao. Modeling channel popularity dynamics in a large IPTV system. In *ACM Sigmetrics*, 2009.

[25] T. Silverston and O. Fourmaux. P2P IPTV measurement: a case study of TVants. In *ACM CoNEXT*, 2006.

[26] P. Spirtes, C. N. Glymour, and R. Scheines. Causation, prediction and search. *Lecture Notes in Statistics*, 1993.

[27] K. Sridhar, G. Damm, and H. C. Cankaya. End-to-end diagnostics in IPTV architectures. *Bell Lab. Tech. J.*, 2008.

[28] M. Tariq, A. Zeitoun, V. Valancius, N. Feamster, and M. Ammar. Answering what-if deployment and configuration questions with WISE. In *SIGCOMM*, 2008.

[29] Wikipedia. Chebyshev inequality. `http://en.wikipedia.org/wiki/Chebyshev's_inequality`.

[30] S. A. Yemini, S. Kliger, E. Mozes, Y. Yemini, , and D. Ohsie. High speed and robust event correlation. In *IEEE Comm.*, 1996.

[31] H. Yu, D. Zheng, B. Y. Zhao, and W. Zheng. Understanding user behavior in large-scale video-on-demand systems. *ACM Sigops Operating Systems Review*, 2006.

[32] Y. Zhang, S. Singh, S. Sen, N. Duffield, and C. Lund. Online identification of hierarchical heavy hitters: algorithms, evaluation, and applications. In *ACM IMC*, 2004.