DIMACS Technical Report 2003-18
June 2003

IPSOFACTO: A Visual Correlation Tool for Aggregate Network Traffic Data

by

Flip Korn
AT&T Labs-Research
Florham Park, NJ 02932
flip@research.att.com

S. Muthukrishnan [1,2]
Rutgers University & AT&T Labs
New Brunswick, New Jersey 08903
muthu@cs.rutgers.edu

Yunyue Zhu[3]
New York University
New York, NY 10012
yunyue@cs.nyu.edu

[1]Permanent Member
[2]The work is supported NSF CCR 00-87022 and NSF ITR 0220280.
[3]This work was done while the author was a DIMACS visitor at AT&T Labs.

ABSTRACT

IP network operators collect aggregate traffic statistics on network interfaces via the Simple Network Management Protocol (SNMP). This is part of routine network operations for most ISPs; it involves a large infrastructure with multiple network management stations polling information from all the network elements and collating a real time data feed. This demo will present a tool that manages the live SNMP data feed on a fully operational large ISP at industry scale. The tool primarily serves to study correlations in the network traffic, by providing a rich mix of ad-hoc querying based on a user-friendly correlation interface and as well as canned queries, based on the expertise of the network operators with field experience. The tool is called IPSOFACTO for *IP Stream-Oriented FAst Correlation for Traffic Overseeing*.

1 Introduction

IP network operators collect aggregate traffic statistics on network interfaces via the Simple Network Management Protocol (SNMP). This is part of routine network operations for most ISPs; it involves a large infrastructure with multiple network management stations polling information from all the network elements and collating a real time data feed. First off the pipeline, this data goes to the Network Operations Center (NOC) where personnel monitor and manage events. Further down the pipeline, it is used for troubleshooting and deriving rules that drive the NOC. Still further down the pipeline, it is used for business decisions: forecasting, sizing, etc.

This demo will present a tool that manages the live SNMP data feed on a fully operational large ISP at industry scale. The tool primarily serves to study correlations in the network traffic, by providing a rich mix of ad-hoc querying based on a user-friendly correlation interface and as well as canned queries, based on the expertise of the network operators with field experience. The tool is called IPSOFACTO for *IP Stream-Oriented FAst Correlation for Traffic Overseeing.*

In the academic literature and in practice, the need for understanding correlations in time series data has long been known [6]. Often, this need has been articulated as one of designing methods to detect correlations automatically, or of automatically performing "root cause analysis". It is our experience that the correlations in network phenomena are often complex, and root causes for events are often subtle, but a flexible, browsing tool with the ability to pose sophisticated correlation queries is valuable in the hands of a network analyst. IPSOFACTO is designed to meet this expectation. The demo will show the full functionality of IPSOFACTO on real SNMP data store from a large ISP.

2 IPSOFACTO

IPSOFACTO operates on an IP network data feed of SNMP-polled statistics and is designed for studying correlation between traffic at different network elements, time scales and aggregation levels. In what follows, we first describe the datafeed on which IPSOFACTO acts, and then describe the overall system. Then, we will describe two significant components of the system before providing details about the demo.

2.1 Network Data and the Data Feed

Our datafeed is traffic data polled from a large IP network — such as in an Internet Service Provider (ISP) — using the Simple Network Management Protocol (SNMP). There are typically up to hundreds of network elements configured in a topological hierarchy consisting of routing devices, each containing multiple interfaces which are endpoints for incoming and outgoing links connecting the devices. These links can have high-speed capacities, up to OC-48 (2.4 Gbps) or even OC-192 (9.6 Gbps). There can be multiple links between a given device-pair; traffic on any link is unidirectional but can flow in either direction. The topology

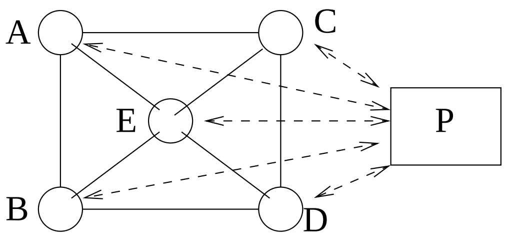

Figure 1: Architecture of an SNMP polling system.

data also contains metadata such as device and interface names, geographic locations, etc. We assume that snapshots of the topology data are obtained from router configuration files and provided on a regular basis (e.g., daily).

The network elements maintain device-level and interface-level statistics. From these statistics it is impossible to determine the type of traffic passing through the link such as the source or destination addresses, or the application of the flows (e.g., ftp/html) for which one needs detailed logs such as Netflow [5]. However, we focus on SNMP-polled data because, unlike detailed traffic logs which are generated only at select routers and tend to be too voluminous to be collected operationally from every element of the network, SNMP polling is ubiquitous (it is supported by nearly every device in today's IP networks), manageable in size, and part of core operations of most ISPs. Among the statistics maintained are cumulative counters of the number of octets (bytes) received and transmitted on each router interface. Figure 1 shows how MIB variable data is collected in the ISP studied in this paper. A single *poller* is shown (P in Figure 1) which periodically requests statistics from each network element (A-E in Figure 1). In reality, there are multiple pollers, both for load balancing as well as for providing redundancy. The data feed contains the difference between successive counts along with a timestamp. For example, the ith tuple would arrive as $(t_i, intID, \Delta V_i)$ where ΔV_i is the traffic volume polled at network interface $intID$ at time t_i.

2.2 Overall System Architecture

Figure 2 shows a schematic diagram of our system. The SNMP data feed is shown on the left. The underlying database is Daytona, which is efficient for dealing with a stream of telecommunications traffic [1]. The system has a graphical user interface for visualizing the network as well as the data and for posing correlation queries and viewing results. This component will be described in some more detail below. The bulk of our work is in supporting various correlation queries, which will also be discussed in more detail below.

Network-polled data feeds have several data quality problems. For example, there are missing polls because the data gathering is typically implemented using UDP, which is an unreliable transport mechanism. The polling intervals may not be synchronized with each other, and different network elements are polled at possibly different times. This leads to

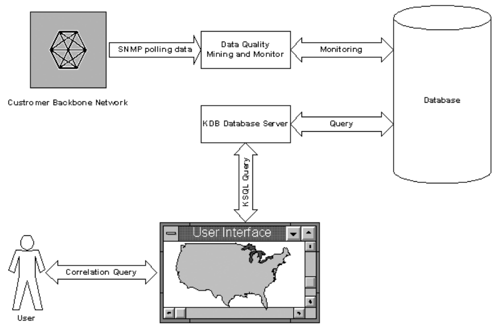

Figure 2: A schematic diagram of IPSOFACTO.

interesting discrepancies; for example, the traffic at a link as measured as outgoing by one of the endpoints may be different from that measured as incoming in the other endpoint. We are working on methods for detecting data quality problems and cleaning them effectively, we plan to submit for publication soon. For the purposes of the demo here, we will assume that such data quality monitoring and cleaning systems are present. This is shown in the top, middle box in Figure 2. Our experience with the system has been that strong interaction between the data quality systems and correlation tool is necessary to ensure that the phenomena that are detected (via correlations, etc) are indeed traffic phenomena and not artifacts in the underlying datafeed. This interaction will be shown in more detail during the demo at the conference.

The front-end user interface is implemented in Java. The user can visualize the traffic of any links in the network at any time and granularity with a few clicks. The user can also use the correlation interface to query for complex analytical results. These queries are translated into KSQL (which is more expressive but not as user-friendly), to be executed by the KDB database server. KSQL is the query language for the KDB database system, which is implemented on top of the K array language [2]. The data structure can be interchanged between K and KSQL so that incorporating special functions into KSQL requires small overhead. KSQL allows functions to be defined over arrays as well as scalars, which makes it efficient for expressing complex correlations.

We made a deliberate design decision to limit the tool to managing only physical network connections and SNMP-traffic data. That is, we did not integrate large scale tabular data such as routing policies, or higher layer information such as service level agreements with other ISPs and customers which may limit some of the links to being only peering links or only private links, etc. IPSOFACTO works on IP traffic datafeed that arrives in multiple streams: the rate and size of this data (few GBs a month) is modest and Daytona, our database system, handles these updates easily. There is currently no need to employ some of the recently discovered streaming methods that work in small space, and tradeoff space for approximations. All our computations are exact. Furthermore, our correlation queries work in fraction of the polling timeperiod of of a few minutes. The tool is currently aimed at analyzing correlations rather than monitoring them, that is, it is an analysis tool and not

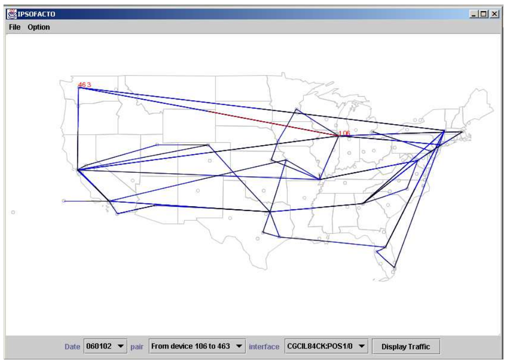

Figure 3: A screen shot of the IPSOFACTO demonstration

an alarming tool. Hence, we do not currently implement techniques for continuous queries. In the future, these methods may become relevant to IPSOFACTO.

2.3 Two Components of IPSOFACTO

We will focus on two significant components: visualization and correlation.

2.3.1 Visualization

IPSOFACTO has a graphical user interface for selecting network elements and links, filtering them on desired aspects, and displaying traffic time series and correlation plots. There are many technical and engineering challenges in developing an effective graphical interface for network data, including how to show colocated routers or multiple links between same pair of routers, how to show the hierarchy in the routers at a physical site (eg., tandem, hub, access or backbone routers), and how to show "satellite" configuration tables associated with the elements.

Figure 3 shows the initial display of IPSOFACTO: an overlay of the network multigraph (routers and links) on top of a map of the United States. IPSOFACTO stores a snapshot of the network topology for each day; when a new day is selected, the topology for that day is loaded and the display is updated. There are menus on the top-left for filtering on link speed or link type to make the display less cluttered. For example, one can choose to display only backbone OC-48s, as shown above. Edges are bichromatic: one side is blue, the other black. The different colors correspond to the different interfaces at the endpoints of the link, since the SNMP data is polled at the interfaces rather than in the middle of the links. Once a link-edge is clicked on, the display confirms this by highlighting the link and identifying the respective device IDs as text on the map; when multiple devices are collocated, an arbitrary one of these device IDs is shown. Collocated links are represented by a single edge, but a drop-down menu can be used to select from a list of links when the representative edge is clicked on. Once (an endpoint of) a single link is selected and the "get traffic" button is clicked on, IPSOFACTO then displays a plot of the traffic rate for the selected interface on the specified date: the outbound number of bytes per second is plotted at each polling time (roughly every 5 minutes) from after midnight the previous day to midnight of the current day. At this point, the user can compute autocorrelations or correlations with other links; the details of this are discussed below.

2.3.2 Correlations

This is the bulk of our technical work. Finding correlations automatically is a formidable task with time-varying effects, and dynamics we do not completely understand. For example, it is difficult to determine the identity let alone the origin of a spike in the traffic. We leave these semantics up to network operators, analysts and other domain experts. Instead, we allow such users to express complex, ad-hoc correlations in a principled way. These queries can be developed and refined interactively. (We have also implemented several canned queries, some of them fairly sophisticated and drawn from the experience of network analysts in the field – for example, what is deemed a traffic "burst" in the NOC for taking actions – and they will be shown in the demo.)

There are three modes of interaction. Each can either correlate pairs of links over the same time period or autocorrelate a link with itself at two different times. In the list below, x and y can refer to the traffic time series vectors on two different links or to the same link at two different times:

1. Specific pair ($\exists x, \exists y$): Given a pair (x,y) of individual links, find their correlation;

2. All elements ($\exists x, \forall y$): Given a single link x, find all links y that correlate;

3. All pairs ($\forall x, \forall y$): Find all pairs of links (x, y) that correlate.

Given some pair (x, y) of time series vectors to correlate, our interface allows the user to specify four independent components of the correlation function $corr(x, y)$ that will be computed on these vectors:

1. **Preprocessing:** scaling, translation, smoothing, aggregation, etc.

   For example, x and y could be mapped to vectors scaled by their respective daily averages $\overline{x}$ and $\overline{y}$.

2. **Temporal band:** determines at which timestamps t_i and t_j to "join" the elements x_i and y_j of respective vectors x and y (i.e., the time window over which the two vectors are composed).

   For example, this could be (i, j) for which $|t_i - t_j| \leq 1$ minute.

3. **Scalar operator:** to be applied on the pairs (x_i, y_j) defined above.

   For example, the operator could be $(x_i - y_j)^2$.

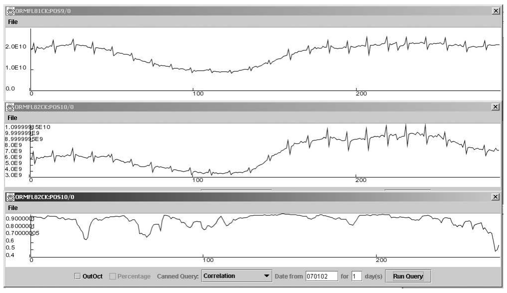

Figure 4: A correlation plot from IPSOFACTO.

4. **Summary:** how to aggregate the output from the scalar operations into an output stream.

   For example, the output could be a vector of sums over one-hour landmark windows over the course of a day.

Putting the above examples together, we get the following:

$$corr(x,y) =$$

$$\left(\sum_{hour1} \sum_{\forall j | t_i - t_j| \leq 1} (\frac{x_i}{\bar{x}} - \frac{y_j}{\bar{y}})^2, \ldots, \sum_{hour24} \sum_{\forall j | t_i - t_j| \leq 1} (\frac{x_i}{\bar{x}} - \frac{y_j}{\bar{y}})^2 \right)$$

As another example, let's consider Pearson's correlation coefficient from statistics:

$$\frac{\frac{1}{w}\sum_{i=1}^{w} x_i y_j - \overline{xy}}{\sqrt{\sum_{i=1}^{w}(x_i - \overline{x})^2}\sqrt{\sum_{i=1}^{w}(y_i - \overline{y})^2}}$$

We can express this using the four components above as follows. First, we scale and shift the elements, e.g., $x_i \leftarrow (x_i - \overline{x})/\sqrt{\sum_{i=1}^{w}(x_i - \overline{x})^2}$. Then for each i, we consider the j for which $|t_i - t_j|$ is the smallest and perform the scalar operation $x_i y_j$. Finally, we aggregate in sliding window averages of length w in each window. Figure 4 shows an example of a correlation plot based on Pearson's coefficient.

3 Demo

The demo will highlight the full functionality of IPSOFACTO. We will have the permission from the client ISP to show trends in the data without divulging crucial business details. We will demonstrate the flexibility and effectiveness of the visualization interface, show the integration of the data quality monitoring and cleaning subsystem, and present the full power of performing correlation analysis using the functionality provided by IPSOFACTO.

Some simple ad-hoc queries are as follows. Here $corr(x,y)$ can be defined by the user as described earlier, or use some of the standard correlation measures.

- What is the correlation between a specified link x and a specified adjacent link y?
- What is the autocorrelation on link x of traffic from this Wednesday compared with that of last Wednesday?
- Given a link x and the devices at its endpoints, which other links between the same device-pair exhibit a similar burst at roughly the same time?
- Find the top-K correlated link pairs (x,y) on 8/28/02.
- Find all links (x,y) that share the same device-pair as endpoints and for which $corr(x, y) < \tau$.

We will also demonstrate the ability of the tool to be interactively used to follow an "event" in the network. For example, say there is a spike of traffic at a router. The tool can be used to look at the neighboring routers and zoom in on links which correlate most significantly with this spike. This can be done by browsing through router by router or link by link, individually or in groups. Our experience has been that this functionality is very helpful for a network analyst to develop viable hypotheses for causes and extent of "events" in the network and do historic correlation searches to test such hypotheses. This proves to be quite effective in practice.

4 Related Work

Software tools exist for general purpose time series analysis. The two most widely used of these tools comes from the statistics community, namely, SAS and S-Plus. Unfortunately, these are standalone tools, without an integrated DBMS backend for processing dynamic data sets, and not directly tailored for the network application of finding correlations, browsing or visualization. Systems for analyzing streams of time series include StatStream [6] and Traderbot [4], but they do not support general correlation analysis like we do and they are not aimed at network traffic databases. While visualization tools such as RRDtool [3] exist for graphing SNMP traffic data, we are not aware of any tool for correlation-based analysis of SNMP data.

5 Acknowledgments

We would like to thank Tamraparni Dasu, Theodore Johnson, Ganga Maguluri, Quynh Nguyen, Matthew Roughan, and Divesh Srivastava for discussions that were helpful in designing IPSOFACTO.

References

[1] Rick Greer. Daytona and the fourth-generation language cymbal. In *SIGMOD 1999, Proceedings ACM SIGMOD International Conference on Management of Data, June 1-3, 1999, Philadelphia, Pennsylvania, USA*, pages 525–526, 1999.

[2] Kx Systems homepage. http://www.kx.com.

[3] RRDtool homepage. http://www.caida.org/tools/utilities/rrdtool/.

[4] Traderbot homepage. http://www.traderbot.com.

[5] Cisco Netflow. http://www.cisco.com/warp/public/732/netflow/index.html.

[6] Yunyue Zhu and Dennis Shasha. Statstream: Statistical monitoring of thousands of data streams in real time. In *VLDB 2002,Proceedings of 28th International Conference on Very Large Data Bases, August 20-23, 2002, Hong Kong, China*, pages 358–369, 2002.