Uncovering Artifacts of Flow Measurement Tools

Ítalo Cunha[1,2] Fernando Silveira[1,2] Ricardo Oliveira[3]
Renata Teixeira[2] Christophe Diot[1]

[1]Thomson [2]UPMC Paris Universitas [3]UCLA

Abstract This paper analyzes the performance of two implementations of J-Flow, the flow measurement tool deployed on most Juniper routers. Our work relies on both controlled experiments and analysis of traces collected at Abilene and GEANT, which provide most of the flow traces used in the research community. We uncover two measurement artifacts in J-Flow traces: a periodic pattern and measurement gaps. We investigate routers' features that trigger these artifacts and show their impact on applications that use flow traces.

1 Introduction

Flow measurement tools summarize traffic going through routers' interfaces. These tools aggregate packets with common characteristics (e.g., IP addresses, ports, and protocol) into flows and compute statistics such as the number of packets and bytes per flow. Both network operators and researchers use flow statistics for tasks that range from capacity planning and traffic matrix estimation [1] to anomaly detection [2,3]. The accuracy of these applications depends ultimately on the accuracy of statistics collected by these flow measurement tools. The first flow measurement tool was NetFlow. It is available on most Cisco routers and has received considerable attention from the research community [4–7]. On the other hand, Juniper's J-Flow has not been studied, even though J-Flow is used to collect statistics in Abilene and GEANT[1] (which are among the few networks that make their data available to the research community).

In this paper, we study Juniper's J-Flow. Our analysis of flow traces from Abilene and GEANT reveals two measurement artifacts—a *periodic pattern* and *measurement gaps*—that appear in all their traces. We use two complementary approaches to identify the cause and study the impact of these artifacts: (1) controlled experiments on a testbed made of Cisco and Juniper routers running NetFlow and two different implementations of J-Flow; and (2) analysis of real flow measurement data. Although it is possible that NetFlow also has artifacts, we leave a specific study for future work. This paper shows that:

- One of J-Flow's implementations exports all flows every minute. These periodic exports alter the duration of flows and create periodic patterns in traffic volume for bin sizes smaller than one minute.

[1] `http://abilene.internet2.edu/` and `http://www.geant2.net/`

– There are periods when routers measure no flows. More than 1.4% of the Abilene and GEANT traces is missing. In some routers, 1% of all 5-minutes time bins lack at least 60% of their data. These gaps are correlated to routing events (i.e., routers miss data in important moments), and create dips in traffic volume.

We believe that this paper is important for all researchers working with J-Flow traces, in particular given the large data sets of publicly available traces from Abilene and GEANT. Even if the artifacts are fixed, they will be present in historical data, and it is important that the research community knows how to work with these data sets.

2 Flow Measurement Tools

J-Flow[2] and NetFlow[3] capture packets as they cross a router's interfaces and aggregate packets with common characteristics (e.g., IP addresses, ports, and protocol) into flows. They are configured per interface and support different aggregation schemes of the measured data. Flow measurement tools maintain a flow cache. Each cache entry counts the number of packets and bytes as well as keeps timestamps of the first and last packets in the flow. Flows are exported to a collector (e.g., a PC) that processes the information for visualization or stores it for later analysis.

There are three main parameters to consider when configuring flow measurements. The *inactive timeout* is the maximum amount of time flows can be in the cache without receiving packets. The *active timeout* is the maximum duration a flow can be in the cache. A flow is exported when it exceeds one of these timeouts. The *sampling rate* controls what fraction of the packets should be inspected and measured on the router's interfaces; other packets are simply ignored. Sampling is necessary because routers lack resources in order to inspect packets at very high speed links. Small timeouts cause flows to be exported more frequently, thereby increasing the accuracy of the statistics, but also increasing storage requirements at the collector. Similarly, when using a high sampling rate, the accuracy of statistics increases, at the cost of the router's CPU overhead.

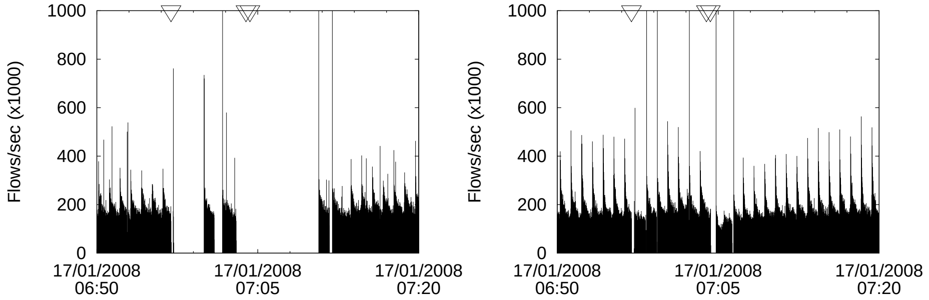
We study the behavior of two different J-Flow implementations: one that shares the route processor (RP for short) with other processes, and the other that uses a dedicated measurement card (DMC).

3 Abilene and GEANT

We analyze flow statistics from Abilene and GEANT. These research backbones collect and make available different types of data sets: flow measurements, BGP updates, and IS-IS messages.

[2] `http://www.juniper.net/products/junos/`

[3] `http://www.cisco.com/go/netflow/`



(a) Milan (M-160, JunOS 8.2)   (b) Frankfurt (T-1600, JunOS 8.5)

Figure 1: Flow Data for GEANT

3.1 Data Description

Flow statistics are collected in all ingress links of both networks by running J-Flow in the route processor (RP). J-Flow RP is more popular than DMC because it does not require a dedicated router port for the measurement card. Abilene currently uses Juniper T-640 routers with JunOS 8.4; and GEANT has either M-series routers with JunOS 8.2 or T-series with JunOS 8.4 or newer. J-Flow is configured with default timeouts (i.e., 15 seconds inactive and 60 seconds active); Abilene and GEANT use packet sampling probabilities of 1/100 and 1/1000, respectively. We aggregate flow statistics from all links of a router. We study data from nine routers from Abilene and 15 from GEANT for four months in four different years: 09/2004, 11/2005, 03/2007, and 01/2008.

3.2 Analysis

Fig. 1 shows the number of flows arriving at each second in two different GEANT routers. This figure shows two unexpected behaviors: a periodic pattern in the number of flows and periods during which no flow is exported.

The periodic pattern repeats every minute and appears in all Abilene and GEANT traces. This pattern is also observed in plots of the number of packets and bytes. We applied a Fourier transform on both Abilene and GEANT data, and we found significant spikes at the frequency corresponding to one minute.

The measurement gaps shown in Fig. 1 are also widespread in Abilene and GEANT data. These gaps occur often and represent between 0.05% and 2.2% of total measurement time depending on the router (i.e., between 21 minutes and 16 hours in one month). They could be explained by lost flow-export packets; however, only 0.0028% (i.e., much less then the gaps) of such packets were lost in the months considered, and even routers with no lost flow-export packets exhibit gaps. We analyze only gaps that are smaller than 10 minutes, and ignore a small number of larger gaps which might be real failures instead of measurement artifacts. While gaps are usually small, with mean gap length ranging from 2.17 to 8.8 seconds, large gaps of one to ten minutes exist, increasing the standard deviation of gap lengths up to 33 seconds.

4 Causes of Artifacts

This section investigates the causes of the periodic pattern and gaps. First, we study whether these artifacts appear in controlled experiments at a router testbed. A testbed gives us full control of the configuration of J-Flow as well as full understanding of the workload. We first describe our experiments and then analyze how the configuration of J-Flow parameters affects the measurements.

4.1 Description of Experiments

We study J-Flow with a testbed composed of two Linux machines connected to a router. One machine replays a packet trace, while the other collects the exported flows. We test a Juniper J-4350 with JunOS 8.0 and both J-Flow implementations (RP and DMC) and, for comparison, we run the same experiments with a Cisco 3825 running IOS 12.4 and NetFlow.

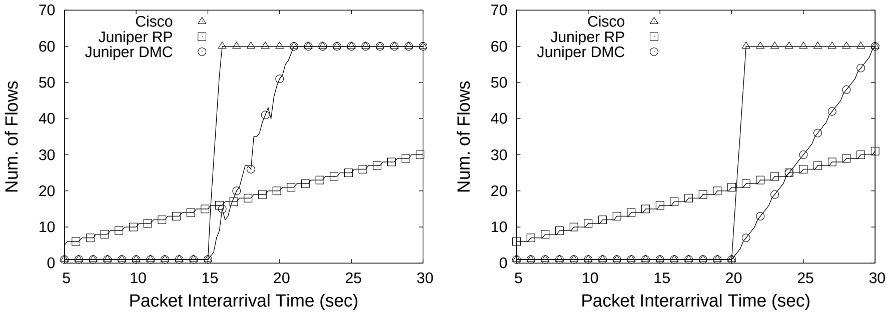
Each experiment proceeds as follows. We configure the router and start replaying the packet trace. Replayed packets are routed to a null interface. Exported flow records are sent to the collector. We also collect SNMP reports on packet and byte counters on the router's interfaces as well as CPU utilization. We analyze flow statistics offline after all experiments have been performed. We turn off all routing protocols and services at the routers and computers to ensure that there is no background traffic in the testbed.

We verify the accuracy of our testbed and configurations by comparing the original packet trace with that received at the collector. This analysis shows that the variance in inter-arrival times between sent and received packets is less than 2.5ms for 99% of packets. No exported flow was lost in our experiments and the SNMP data confirmed the accuracy of the measurements.

We use one synthetic and one real packet trace. The synthetic trace is designed to test whether timeouts occur as configured. It contains simultaneous flows, each with a fixed packet inter-arrival time. Inter-arrival times range from 5 to 31 seconds in steps of 0.2 seconds (i.e., 130 flows). We also replay a packet trace collected at a 100Mbit Ethernet interface between WIDE and its upstream provider[4]. We use one direction of the link in a 3-hours trace starting 14:00 on 09/Jan/07. This subset has 9402 pkts/s and 54.16 Mbits/s on average.

We denote the inactive timeout by I in seconds and the active timeout by A in minutes. Routers are configured to inspect all packets (i.e., no sampling). For the synthetic trace, the cache never overflows and all flow exports are due to timeouts only (e.g., no TCP RST/FIN packets). Results for the synthetic load are the mean over five experiments and have standard deviations under 3%.

[4] http://mawi.wide.ad.jp/mawi/samplepoint-F/



(a) $I = 15$ seconds (b) $I = 20$ seconds

Figure 2: Inactive Timeout Characterization

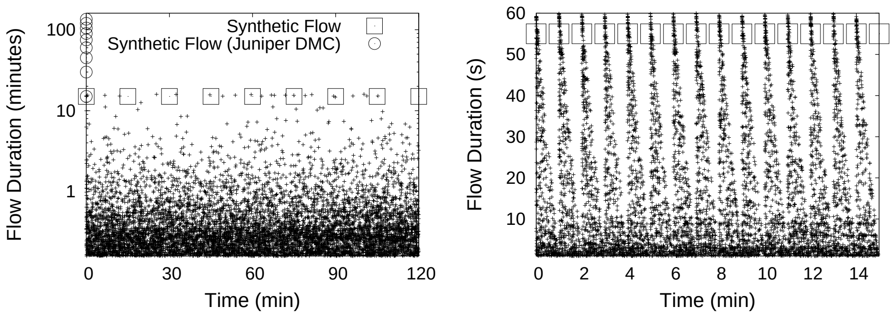
We vary the configuration of J-Flow and NetFlow as well as the characteristics of the traces. We quantify the *precision* of each timeout as the difference between the timeout value (i.e., I and A) and measured flow durations.

4.2 Periodic Patterns

Inactive Timeout. We characterize the precision of the inactive timeout using a synthetic trace where each flow has 60 packets. We vary I between 15 and 30 seconds and fix $A = 30$ minutes. Fig. 2 shows the number of flows exported for this trace. The x-axis presents the packet inter-arrival time and the y-axis the number of flows with a given inter-arrival time. The expected behavior of this graph is that all flows with inter-arrival times less than I should be exported in one flow record (i.e., $y = 1$), whereas flows with inter-arrival times greater than I should be exported in 60 flows (i.e., one flow per packet as after each packet, the flow created in the cache would timeout before the arrival of the next packet).

NetFlow follows the expected behavior. Its inactive timeout has a precision around one second: flows with packet inter-arrival times between I and $I+1$ seconds are split in a variable number of flows. It is also conservative: no flows with inter-arrival times smaller than I are split. J-Flow's DMC implementation follows the same trend, but with worse precision. In addition, its precision decreases as the value of I increases. When $I = 15$ seconds, J-Flow DMC has around five seconds of precision. The precision decreases to ten seconds for $I = 20$.

Finally, a comparison of results in Figs. 2-a and 2-b shows that J-Flow's RP implementation does not consider the inactive timeout. Its behavior is exactly the same irrespective of the timeout value. In fact, the number of exported flows is proportional to the flow's duration (i.e., 60 packets × inter-arrival time) as J-Flow RP exports flows periodically every minute. We detail this behavior next.



(a) Cisco (b) Juniper RP

Figure 3: Active Timeout ($A = 15$ min)

Active Timeout. This section studies the precision of the active timeout. We run experiments with the synthetic and WIDE traces. We vary A from 15 to 30 minutes and fix $I = 15$ seconds.

Fig. 3-a shows the WIDE trace as captured by NetFlow using dots. The x-axis shows the time a flow started, and the y-axis shows the flow's duration. The WIDE trace contains many small flows. We also show as squares the results of a synthetic flow to serve as reference. This flow has one packet every five seconds and lasts for the entire experiment. Hence, given $A = 15$, we have one square every 15 minutes with flow duration of 15 minutes. In the WIDE trace, only 0.07% of flows are exported due to the active timeout. NetFlow's active timeout has a precision around two minutes (i.e., durations range from A to $A + 2$).

J-Flow DMC's behavior (not shown) is similar to the one in Fig. 3-a. However, DMC reports inaccurate start times for flows exported due to the active timeout. To show this, we plot the synthetic flow in Fig. 3-a as large circles. Instead of falling on top of the squares, the circles are all aligned at $x = 0$, but with different durations. When DMC exports a flow due to the expiration of the active timeout it resets all the counters related to this flow, except for the start time. Hence, subsequent exports of the same flow always have the same start time and different end time, which makes the flows longer at each export.

Fig. 3-b presents the same results of Fig. 3-a when using J-Flow's RP implementation. The contrast between these two figures is striking. RP introduces a clear periodic pattern, which is similar to the pattern seen in Abilene's and GEANT's data (Fig. 1). This periodic pattern is not due to cache overflow, as it also impacts the synthetic flow. Instead of exporting the synthetic flow every 15 minutes, RP exports it every minute (as shown by the squares). Moreover, an active timeout of one minute would not create this periodicity. We conjecture that this periodic pattern is due to a flush of the flow cache every minute, when all flows are exported. This explains the sawtooth shape of Figs. 1 and 3-b. This periodic flush of the cache leads to higher memory consumption at the router [8]. For instance, the WIDE trace has an average number of active flows (i.e., memory consumption) 28% higher in J-Flow RP than in NetFlow, because the inactive timeout is ignored and short flows (specially those with one sampled packet) are kept in the cache until the next flush. Finally, frequent flushes could impose unanticipated load on the network and the collector.

Summary. Our controlled experiments show that J-Flow's RP implementation, used by both Abilene and GEANT, creates a periodic pattern in exports because it flushes the flow cache every minute. We also observe that J-Flow's DMC implementation logs inaccurate start time for flows exported due to the expiration of the active timeout. We investigate measurement gaps next.

4.3 Gaps

We did not observe measurement gaps in our testbed. This is mainly because of light load, as we could not stress the routers with only two PCs. We also lack heavy load traces obtained with NetFlow or J-Flow DMC, so we leave the evaluation of these implementations for future work. We analyze only J-Flow RP using traces from Abilene and GEANT.

As J-Flow RP shares the CPU with other higher-priority routing processes, we check if routing messages are related to gaps. We use Abilene and GEANT traces to collect IS-IS updates that signal changes in the link state data base (LSDB) of each router, i.e. if an internal link fails or recovers, an IS-IS update is propagated to all routers inside the network. The instants when such messages happen are represented by the upside down triangles in Fig. 1. We observe that the starting point of the gaps in the figure coincide with IS-IS activity.

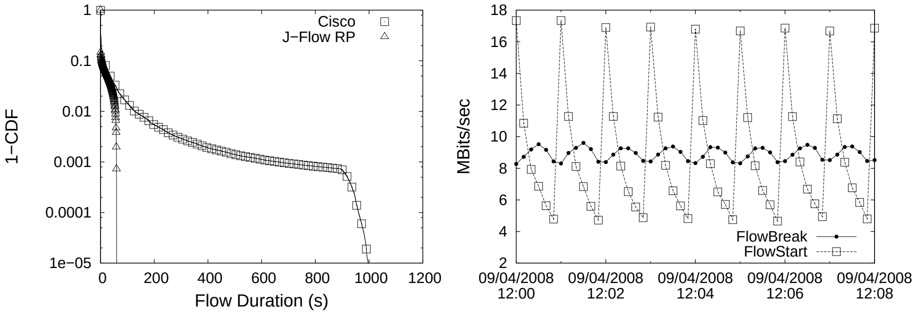
We checked with Juniper, and they have recently addressed this in Problem Report 277942. They detected that the routing daemon could lock the routing table for extended periods of time, causing the J-Flow process to block. They updated the software, making the routing daemon unlock the routing table in a timely fashion to avoid starving the J-Flow process.

5 Impact on Applications

In this section, we show how periodic cache flushes and measurement gaps can influence the result of network management tasks such as anomaly detection. We quantify this impact on the Abilene and GEANT traces, and discuss alternatives to deal with the problem.

5.1 Periodic Cache Flushes

Cache flushes have a direct impact on the distributions of flow sizes and flow durations. In J-Flow RP, the flow duration distribution will clearly have a truncation point at one minute, as shown in Fig. 4-a for the WIDE trace. We do not show the flow durations for J-Flow DMC, since this implementation does not



(a) Flow Duration

(b) Total Volume

Figure 4: Impact of Cache Flushes

reset the time stamp of the start of the flows after the active timeout expires. The distribution of flow sizes (not shown), which is the basis of applications such as heavy hitter detection, follows the same truncated behavior.

Cache flushes also cause periodic patterns in traffic volume over time. This behavior is only visible when traffic is binned in intervals smaller than one minute. We consider two ways of allocating bytes to bins. *FlowStart* adds all packets (or bytes) in a flow to the bin where the flow started, whereas *FlowBreak* assumes that packets in a flow are equally spaced across the duration of the flow and spreads them on the corresponding bins. Fig. 4-b shows the traffic volume for bins of ten seconds using GEANT data. FlowStart has the same periodic behavior as of Fig. 3-b, as more flows start at the beginning of each minute after the cache flush. FlowBreak is more stable, but also shows a periodic pattern. These periodic patterns are relevant to applications using flow statistics at fine-grained time granularities. The periodic behavior in Fig. 4-b is less visible if bins are larger (more aggregation) or if traffic volume is small (more variation). Bins larger than one minute are not significantly impacted by cache flushes.

Most researchers and tools already post-process flow measurements, joining exported flows with the same signature to calculate their duration. Wallerich et al. [6] presents and evaluates an effective method for this task. We point out that tools working at finer time-scales could also take advantage of the periodic exports of a flow to estimate its transfer rate at 1-minute intervals, instead of calculating an average over larger intervals.

5.2 Gaps

Large measurement gaps can significantly impact the volume of traffic in a bin. Fig. 5 shows the distribution for the amount of missing data per 5-minutes bin on January 2008 for different GEANT routers. For some routers, 1% of their bins (i.e., 90 bins over a month) have more than two minutes of missing data. These gaps would impact both anomaly detection [2,3] and traffic matrix estimation [1].

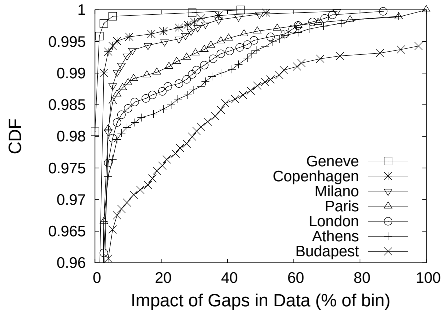


Figure 5: Impact of Gaps on 5-min Bins

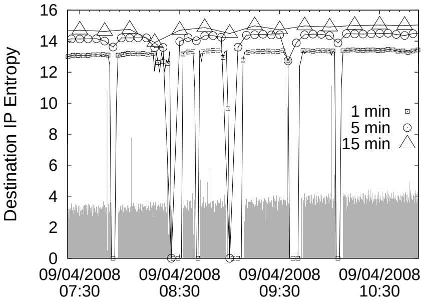


Figure 6: Impact of Gaps on Entropy

Lakhina et al. [3] proposed using the entropy of traffic feature distributions (e.g., IP addresses and ports) to find traffic anomalies. Fig. 6 shows the impact of measurement gaps on the entropy of destination IPs using different bin sizes. The behavior of other entropies (e.g., ports) are similar. The flow measurement, with gaps, is plotted in gray. Gaps have small impact on entropy because they do not impact directly the distribution of IP addresses and ports. On one hand, small bins that contain no data because of large gaps have an entropy value of zero; on the other hand, 15-minutes bins are completely oblivious to the gaps.

The gaps have two opposing effects on anomaly detection. First, a gap may be easily confused with a link failure, and detection methods can trigger alarms which are just exposing a measurement artifact instead of a traffic change. Second, the gaps are often caused by routing events, which may lead to traffic shifts that anomaly methods should be able to uncover. Since data is missing right after a routing change, it may be impossible to analyze the anomaly and quantify its impact on applications.

New routers, like Frankfurt and Geneva (Juniper T-1600 with JunOS 8.5), experience only few and short gaps, as shown in Figs. 1-b and 5, respectively. Although the impact of gaps may become less important as software and hardware are upgraded, these artifacts are present in large amounts of data collected in the past and should be taken into account by researchers and tools (e.g., check SNMP packet counters to differentiate an outage from a measurement gap).

6 Related Work

Although Cisco NetFlow has received a lot of attention [4–7], we are the first to study the correctness and accuracy of Juniper J-Flow and how its measurement artifacts impact applications. The accuracy of assuming constant throughput and packet sizes when joining flows (i.e., FlowBreak) was analyzed by Wallerich et. al. [6]; the authors found that this is a good approximation for heavy-hitters while still reasonable for small flows. Previous work have also proposed improved methodologies for flow measurement [5,8]. They are based on the idea of dynamically setting the configuration (e.g., sampling rate) depending on the available resources and the workload on the router. These works are orthogonal to ours: while a better measurement methodology is desirable, J-Flow and NetFlow are the currently deployed tools and large data sets are collected using them.

7 Conclusion

We have identified two measurement artifacts that impact most publicly available traces used by the Internet research community to study new traffic analysis and engineering techniques. At large time scales (15 minutes and above), measurement gaps generally go unnoticed. This is critical as these measurement gaps can include an anomaly as they are often correlated to major routing events that could trigger large traffic shifts. At small time scales, measurement gaps are detectable, but cache flushes create periodic patterns in traffic volume. The research community needs to be aware of these artifacts in order to take them into account when using Abilene and GEANT traces to validate their research work. These artifacts could also impact ISPs that use flow measurement tools for traffic engineering and billing.

Acknowledgements

We thank Maurizio Molina for his comments and help in understanding measurement artifacts in GEANT data. We also thank Olivier Fourmeaux and Florian Le Goff for giving us access to the LIP6 routers. We thank Jennifer Rexford, Zied Ben-Houidi, and Augustin Chaintreau for their comments on the paper.

References

1. Zhao, Q., Ge, Z., Wang, J., Xu, J.: Robust Traffic Matrix Estimation with Imperfect Information: Making use of Multiple Data Sources. SIGMETRICS Perform. Eval. Rev. **34**(1) (2006) 133–144
2. Barford, P., Kline, J., Plonka, D., Ron, A.: A Signal Analysis of Network Traffic Anomalies. In: Proc. ACM IMW, Marseille, France (Nov. 2002)
3. Lakhina, A., Crovella, M., Diot, C.: Mining Anomalies Using Traffic Feature Distributions. In: Proc. ACM SIGCOMM, Philadelphia, PA (Aug. 2005)
4. Sommer, R., Feldmann, A.: NetFlow: Information Loss or Win? In: Proc. ACM IMW, Marseille, France (Nov. 2002)
5. Estan, C., Keys, K., Moore, D., Varghese, G.: Building a Better NetFlow. In: Proc. of ACM SIGCOMM, Portland, OR (Aug. 2004)
6. Wallerich, J., Dreger, H., Feldmann, A., Krishnamurthy, B., Willinger, W.: A Methodology for Studying Persistency Aspects of Internet Flows. SIGCOMM Comput. Commun. Rev. **35**(2) (2005) 23–36
7. Choi, B.Y., Bhattacharyya, S.: Observations on Cisco Sampled NetFlow. SIGMETRICS Perform. Eval. Rev. **33**(3) (2005) 18–23
8. Kompella, R., Estan, C.: The Power of Slicing in Internet Flow Measurement. In: Proc. ACM IMC, Berkeley, CA (Oct. 2005)