Noname manuscript No.
(will be inserted by the editor)

Flow-Based Network Management: A Report from the IRTF NMRG Workshop

Ricardo de O. Schmidt · Ramin Sadre · Luuk Hendriks

the date of receipt and acceptance should be inserted later

Abstract This is the report on the Workshop on Flow-Based Network Management, held within the 37th IRTF NMRG meeting, during IETF 93, on 24th July 2015, in Prague, Czech Republic. Following the tradition of the IRTF NMRG, the workshop focused on technologies, developments, and challenges of using flow-level traffic measurements for network management.

Keywords NMRG · Flows · NetFlow · IPFIX · OpenFlow

1 Introduction

Flow-based approaches are used in various areas of network management today, such as link monitoring, accounting, and security. It is a tradition for the IRTF NMRG (Network Management Research Group) to organize a day workshop focusing on flow-based network management, with the goal to provide a forum for academia and industry to present and discuss the latest developments and experiences in this area. Previous editions of the NMRG workshop have had their reports also published within this journal. In 2009 [5], the workshop brought up discussions on technologies to capture and analyze flow data, and on the effects of sampling and aggregation strategies on the accuracy of results. In 2011 [2], the workshop raised discussions on the standardization of the IPFIX protocol and challenges on its adoption, potential applications of NetFlow/IPFIX, and how NetFlow/IPFIX was being used in practice. The workshop organizers also initiated a special issue on recent advances and future trends in flow-based approaches [7].

R. de O. Schmidt · L. Hendriks
University of Twente, Enschede, The Netherlands
E-mail: {r.schmidt,luuk.hendriks}@utwente.nl

R. Sadre
Université Catholique de Louvain, Louvain-la-Neuve, Belgium
E-mail: ramin.sadre@uclouvain.be

Differently from previous editions, which focused only on NetFlow/IPFIX, in this edition of the workshop we have broadened the workshop scope to also include other flow-based technologies that allow for traffic measurements, such as OpenFlow and other technologies and strategies introduced with Software-Defined Networking (SDN). In total, 32 people attended the workshop, which was composed of 9 presentations. This report is a summary of the presentations along with their main conclusions, addressing the following topics:

- Developments on flow-based measurements for constrained devices, very high-speed links, and storage and processing of large amounts of flow data.
- Usage of flows in network security.
- Flows measurement in OpenFlow-enabled networks.

The slides from the presentations are publicly available at the repository website of the IETF 93 meeting[1].

2 Challenges of flow-level traffic measurements

2.1 TinyIPFIX for WSN

Wireless Sensor Networks (WSN) have been widely researched and deployed in the context of large scale distributed measurements. Typically, WSNs are composed by a large number of smart meters locally gathering data on their surrounding environment, and sending the data to a centralized location for further processing. These smart meters are resource limited (*i.e.,* low-cost and low-power hardware) and demand for efficient communication protocols to transmit metering data. A push-based protocol is, therefore, the most suitable solution.

IPFIX [6] has been developed based on the successful Cisco NetFlow for traffic flow measurements. IPFIX is an IETF standard and defines a push protocol with template-based design. However, IPFIX as proposed is not a suitable protocol for using within WSNs due to the overhead of additional headers (*i.e.,* 20 Bytes extra). Moreover, sensors have limitations on the maximum transmission unit of 102 bytes from the IEEE 802.15.4 standard.

TinyIPFIX, proposed by Corina Schmitt and Burkhard Stiller from the University of Zürich, uses compression techniques to reduce the overhead of traditional IPFIX. In its aggressive compression mode, TinyIPFIX can reduce 85% of the IPFIX overhead, from 20 to 3 bytes. TinyIPFIX is currently implemented for TinyOS and it has been successfully deployed in multiple sensors, namely the IRIS sensors from Crossbow Inc., the TelosB from Advantic Sys., and the OPAL from APDM Wearable Tech.

[1] https://datatracker.ietf.org/meeting/93/materials.html

2.2 Software-Defined Monitoring

We have been seeing that network monitoring is shifting to application layer (*i.e.,* L7) processing. We have also been seeing the ever increasing traffic demands and the recent deployments of 100 Gbps networks. The CPUs of commodity multi-core platforms are limited to process few thousands of packets per second (basic NetFlow monitoring) and statistics are computed up to transport layer (TCP and UDP). Processing of application layer information, however, is more complex and demands more resources. A complete hardware-based implementation of monitoring (if possible, because it is still under research) would lack flexibility.

Software Defined Monitoring (SDM), initially proposed in [3] and here presented by Lukáš Kekely, Viktor Puš and Jan Kořenek from CESNET, is a trade-off between software and hardware monitoring. Although SDM borrows the general idea of the uprising Software-Defined Networking (SDN) paradigm, these two do not overlap. Roughly, SDM is implemented in two modules, namely the firmware and the software modules. The first packet of a unknown flow is sent by the firmware to the SDM software, which will then decide how the monitoring for that specific flow will be performed. If the flow is of interest and further processing is needed on the flow packets, the firmware will be required to send all matching packets to the SDM software. Otherwise, basic flow accounting for that flow will happen in hardware.

The SDM monitoring system uses the FPGA card (*Field Programmable Gate Array* – SDM firmware) to offload and accelerate the processing of the bulk traffic, while a commodity multi-core CPU (SDM software) performs the detailed analysis of application layer headers for the traffic of interest. By combining CPU and software plugins with FPGA processing, SDM is able to achieve software-like flexibility on monitoring traffic with the performance of a dedicated hardware box.

2.3 Data storage and processing using big data

Monitoring high-speed networks generates a massive amount of flow data. For example, at CESNET's network (Czech Republic's NREN) approximately 250 GB of flow data is monitored per day. The storage and post-processing (*i.e.,* interactive manipulation) of such amount of data is a challenging task. This problem aggravates when using the monitored flow data for, *e.g.,* security operations that require fast data retrieval with low latency. One potential solution is to use open-source platforms for big data processing. However, these open-source platforms present certain limitations due to the specific format of flow data.

Experiments on using big data solutions for flow processing have been done by Martin Žadník, from CESNET (presented by Viktor Puš). These experiments used a Hadoop cluster of 24 slaves and 3 master nodes with 128 GB RAM each and total disk capacity of 1 PB. A comparison using simple queries showed that MapReduce operations in text (CSV) and binary formats, and queries in Hive (a SQL interface into Hadoop) were meaningfully faster than those made in Pig (a functional interface that stores data in CSV format). Vertica, a column-based database, has also been tested. Using a cluster of 3 nodes with 4 GB RAM each, queries were performed significantly faster with Vertica.

3 Flow measurements for network security

3.1 Botnet detection

Botnets are a significant threat to the security and integrity of the Internet. They provide an infrastructure for various kinds of attacks, either directed against the users of the infected Bot machines (e.g., data theft) or against other parties (e.g., SPAM or Distributed DoS attacks). In a botnet, the infected hosts are under the command of a bot master that communicates with them through one or more command-and-control servers.

In his presentation, speaker Christian Dietz from the Universität der Bundeswehr München identified, among others, heterogeneous environments as an important challenge toward the successful detection of botnets from flow data. Standardized and normalized descriptors for attack behavior and measurement noise are needed, so that descriptors can be stored and exchanged among involved parties in a heterogeneous environment.

As an example, a normalized descriptor of the temporal network behavior of a bot is proposed. The normalization allows to exchange the descriptor between the local intrusion detection systems of cooperative heterogeneous networks. If appropriately implemented, efficient algorithms (such as K-nearest, Bloom filters, etc.) would enable local detection systems to efficiently search the shared data for behavior similar to their observations.

3.2 Fingerprinting and classification

Christian Hammerschmidt from the University of Luxemburg gave a presentation on the usage of Automaton models for fingerprinting and classification of flow data. In his work in progress, he proposes to learn finite state machines from NetFlow data. The resulting machines describe the sequence of observed flow data in a network and can be used to predict the statistics of the next flows or to classify flows for activity or intrusion identification. When seeing NetFlow data as a (regular) language to be produced by an automaton, one challenge is the large state-space, which has to be reduced by clustering to obtain few representatives or by discretization (binning).

Initial results from experiments with time-aggregated flow data look promising. Future research will comprise a comparison of the performance of the approach with other fingerprinting solutions and the application of more expressive automaton models, such as state machines that specify symbol probabilities.

3.3 Distributed anomaly detection

In a small network, network-based anomaly detection can be achieved by monitoring network traffic at one single point of observation, e.g., a router. However, such a simple approach is not possible anymore for large networks with multiple ingress points, where monitoring at one of those points only provides a partial view of the network. Researchers have proposed different IDS (*Intrusion Detection System*) designs to handle the latter scenario [1]. In a centralized design, all observation points send their data to a central point where analysis and intrusion detection is performed. In a distributed design, multiple local detection systems are deployed next to the observations points. Those local systems communicate with other systems or with a central server.

The presentation by Carlos García Cordero (TU Darmstadt) on joint work with Andreas Vöst and Jochen Kögel (IsarNet) addresses the problem of learning in a distributed system. In an anomaly-based IDS, the goal of learning is to produce a normality model that can be used to detect anomalous behavior. In order to avoid centralized learning (which would require to collect all data at a central point), the authors successfully follow a distributed approach where each local system learns its model independently from the other local systems. The result is a set of models that can be used to create ensembles of learners. The authors are also interested in techniques to merge the individually learned models into a single global model without the overhead of centralized learning. Although such techniques exist, they are model and learner specific.

3.4 IPv6 security landscape

With the increasing adoption of IPv6 within networks and the Internet, and the eventual replacement of IPv4 in the long term, security research in the area of IPv6 is a necessity. While many theoritical works have pointed out weaknesses and vulnerabilities in IPv6 on both LAN and WAN scale, real-world measurements have been limited to observing traffic reaching dark-nets (i.e., address space that is advertized and routed, but does not contain any real hosts). This approach enables easy classification of malign traffic (as everything reaching the dark-net can be considered malicious), but the amount of traffic is low as there are no hosts to be a target of interest. Luuk Hendriks from the University of Twente aims at analyzing IPv6 traffic in production networks that have been IPv6-enabled for several years, e.g., NREN and university networks, to determine which threats are actually active in the wild.

The main goal of this research is determining to what extent new security measures are a necessity, or whether approaches from the IPv4 era suffice. With the growing number of devices connected to the Internet, including non-traditional machines (i.e. the Internet of Things), we can not rely on threat detection on end hosts. Flow-based solutions enables for scalable approaches, to observe entire networks from a single vantage point. Leveraging this allows for analysis on a large scale and creating a realistic view of the threats active on networks. Attack tool suites are analyzed for flow-level characteristics, forming the bases of detection algorithms for these threats.

Threats that focus on the application layer of the OSI model are likely to occur in an IPv6 network, and approaches to counter these threats will be very similar to those of IPv4 networks. The network layer itself is where new concepts are introduced, and therefore the more interesting area of research. To do flow-based analysis, the use of IPFIX and the possibility of custom Information Elements is vital, enabling to include new protocol fields in detection algorithms. The actual measurements will be performed by distributing a piece of open source software, performing detection based on flow-based input, and creating privacy preserving reports on which classes of threats are observed. This provides scalable detection for the networks of the vantage point, and over time give an overview of the status of security in an IPv6 Internet.

4 Flow-level traffic measurements in OpenFlow-enabled networks

4.1 Monitoring, visualization and configuration of OpenFlow-based SDN

While the SDN paradigm reduces, or in some cases eliminates, traditional network management problems, it comes with new challenges. The introduction of the controller concept is an example of that: while enabling network operators to do configuration using high level languages, the necessary communication between the controller and the forwarding devices introduces previously non-existing resource consumption and possible effects on the forwarding performance. Pedro Heleno Isolani (UFRGS) quantified the overheads imposed by OpenFlow messages on the control channel, and furthermore proposes an interactive approach to SDN management based on monitoring, visualization and configuration.

Using Mininet, a network composed of 230 hosts, 11 switches, 2 content servers and 1 controller was emulated, and both the control channel load and the resource usage were analyzed. This showed that 99.70% of the control traffic could be ascribed to the four OpenFlow messages for Modify-State (to add or modify flow tables entries), Read-State (to retrieve statistics from the switch), Send-Packet (to send packets to a specific swith port) and Packet-In (to send unmatched packets from the switch to the controller). Furthermore, a clear relation between controller channel load and the configured idle timeouts was observed. Longer timeout values cause more control channel load, as responses to statistics request contain information for a greater number of flow entries. In addition, longer timeout values result in less messages of types Modify-State and Packet-In.

The second (and partly based on the former) contribution of this research was the proposal of an SDN management approach composed of monitoring, visualization and configuration managers. Besides the architectural overview a prototype GUI was presented, showing interactive graphs visualising control channel loads, network topologies and configuration parameters, among other information. The researchers emphasize the fact that implementations of control channel handlers differ and that standardisation of interfaces for all layers, i.e. forwarding devices, controllers and applications, may foster the development of SDN-based solutions.

4.2 Flow-based management in NFV

Within the Network Function Virtualization Research Group (NFVRG) an open discussion is ongoing, trying to address challenges in flow-based network management. The draft related to this, `draft-unify-nfvrg-devops` [4], contains implications which were presented and explained by Catalin Meirosu from Ericsson Research. This document describes a set of principals relevant from a DevOps perspective in managing SDN infrastructures, and also the challenges that come with it such as development of tools, interfaces and protocols. Catalin, active in the UNIFY project, specifically focussed on two challenge areas in his presentation, namely challenges of stability, and observability, in SDN infrastructures.

As controllers and agents in such an SDN infrastructure have limited capacity to process flow requests, admission of flows is not always possible in a pro-active way. Furthermore, every flow has similar overhead, whether it is a big (*elephant*) or small (*mice*) flow, eliminating any possible gain otherwise obtained via prioritizing one or the other. Ultimately, flow table congestion can occur, naturally affecting the stability of the network. Using aggregated flows (using wildcards in the OpenFlow matching fields) will increase scalability, at the cost of visibility. This directly affects the observability: while monitoring aggregated flows for a subnet and observing e.g. packet loss, it is impossible to derive between which two individual hosts this packet loss actually occurred. A possible solution in the draft uses more complex flow entries in the flow table on the ingress node, to select a more granular sub-flow from the aggregated flow.

Another challenge is the interaction between the routing application and the monitoring application on the controller. To serve sound information, the monitoring application needs to receive context information from the routing app, e.g., path changes in the network. The routing application uses that information again to decide what path changes to apply. Such a solution emphasizes the need for a consistent deployment, where every forwarding device is able to provide the statistics needed by the monitoring application. The draft proposes monitoring annotations in order to forward graph information to the controller, using separate probes proving the necessary information. Placement of these probes and their update frequencies form another topic of research, as are the specific measurements they should provide. The draft describes examples for statistical counters, i.e., mean and variance values, providing more insight than traditional, simple counter values.

5 Summary

The IRTF NMRG Workshop series has been considered a valuable place to discuss new ideas and recent advances on flow-level traffic measurements. Unlike previous editions of the workshop that focused on NetFlow/IPFIX technologies, in this edition the scope of the program has been widened to other flow-based approaches for network management. We have seen, therefore, presentations ranging from the use of IPFIX in the constrained networking scenarios of WSN, to security-related solutions using flows, and to the most recent SDN and OpenFlow technologies. This edition of the workshop enabled remarkable discussions and interactions between people already engaged in the IRTF NMRG and also first-time comers. All the presentations are available at the website of the IETF 93 meeting.

Acknowledgements This report was partially supported by the Self-managing Anycast Networks for DNS project (http://www.sand-project.nl) and by the EU FP7 Mobile Cloud Networking project (#318109).

References

1. Axelsson, S.: Intrusion detection systems: A survey and taxonomy. Tech. rep., Chalmers University of Technology, Sweden (2000)
2. Drago, I., Barbosa, R.R.R., Sadre, R., Pras, A., Schönwälder, J.: Report of the Second Workshop on the Usage of NetFlow/IPFIX in Network Management. Journal of Network and Systems Management 19(2), 298–304 (2011)
3. Kekely, L., Puš, V., Kořenek, J.: Software Defined Monitoring of Application Protocols. In: Proceedings of the IEEE INFOCOM. pp. 1725–1733 (2014)
4. Meirosu, C., Manzalini, A., Kim, J., Steinert, R., Sharma, S., Marchetto, G., Papafili, I.: DevOps for Software-Defined Telecom Infrastructures. Internet-draft (2015)
5. Pras, A., Sadre, R., Sperotto, A., Fioreze, T., Hausheer, D., Schönwälder, J.: Using NetFlow/IPFIX for Network Management. Journal of Network and Systems Management 17(4), 482–487 (2009)
6. Quittek, J., Zseby, T., Claise, B., Zander, S.: Requirements for IP Flow Information Export (IPFIX). RFC 3917 (2004)
7. Sadre, R., Sperotto, A., Hofstede, R., Brownlee, N. (eds.): Special Issue of the International Journal of Network Management: Flow-based Approaches in Network Management: Recent Advances and Future Trends, vol. 24(4) (2014)

Author Biographies

Ricardo de O. Schmidt is a Postdoctoral Researcher at the Design and Analysis of Communication Systems (DACS) group, of the University of Twente, Netherlands. He received a PhD degree from the University of Twente in 2014. He also received a M.Sc. degree in Computer Science from the Federal University of Pernambuco in 2010, and a B.Sc. degree in Computer Science from the University of Passo Fundo in 2007, both in Brazil. His research interests include network management, traffic measurements, DNS, routing and addressing.

Ramin Sadre is an Assistant Professor at the Université Catholique de Louvain, Belgium. He received a Ph.D. degree from the University of Twente for his thesis titled “Decomposition Based Analysis of Queuing Networks”. His research interests include traffic modeling, the design and analytical performance evaluation of communication systems, and the design of network intrusion detection systems.

Luuk Hendriks is a PhD candidate at the Design and Analysis of Communication Systems (DACS) group, of the University of Twente, Netherlands. He received a MSc degree from the University of Twente in 2014. His research interests include IPv6 and network security.