Next Generation Application-Aware Flow Monitoring

Petr Velan, Pavel Čeleda

Next Generation Application-Aware Flow Monitoring

Petr Velan and Pavel Čeleda

Institute of Computer Science, Masaryk University
Botanická 68a, Brno, Czech Republic
{velan|celeda}@ics.muni.cz

Abstract. Deep packet inspection (DPI) and IP flow monitoring are frequently used network monitoring approaches. Although the DPI provides application visibility, detailed examination of every packet is computationally intensive. The IP flow monitoring achieves high performance by processing only packet headers, but provides less details about the traffic itself. Application-aware flow monitoring is proposed as an attempt to combine DPI accuracy and IP flow monitoring performance. However, the impacts, benefits and disadvantages of application flow monitoring have not been studied in detail yet. The work proposed in this paper attempts to rectify this lack of research. We also propose a next generation flow measurement for application monitoring. The flows will represent events within the application protocol, e.g., web page download, instead of packet stream. Finally, we will investigate the performance of different approaches to application classification and application parsing with a computational complexity in mind.

Keywords: flow, network measurement, application monitoring, IPFIX

1 Introduction

The number of different applications communicating over the Internet is ever increasing and so is the need for application-aware network monitoring. However, building network monitoring systems is always a compromise between accuracy and performance. The more information processed, the more accurate the monitoring system is. However, thorough examination of the traffic is computationally expensive [14,18].

Application flow monitoring is a network monitoring approach created to exploit the benefits of deep packet inspection (DPI). Integration of the DPI into flow monitoring allows for information aggregation, which provides better performance than the DPI alone. However, the impacts, benefits and disadvantages of application flow monitoring have not been studied in detail yet. Therefore, we will research the impact of application flow monitoring on flow exporters. Then we will propose improvements to the application flow monitoring that will help to cope with any challenges discovered during the research. We also believe that it is possible to utilize a newly acquired application information to improve the

quality of the flows. Based on experiences gained during the research, we will propose a next generation flow measurement for application monitoring. The flows will match events within the application protocol, e.g. web page download, instead of packet stream. This approach will provide more context to monitored traffic, which will improve network security, the quality of service and the quality of experience. Finally, to address the monitoring of high-speed networks, we will investigate the performance of different approaches to application classification and application parsing with a computational complexity in mind.

The rest of the paper describes the motivation for our research and introduces the proposed research in more detail.

2 Motivation

This section describes problems of the current generation of application flow monitoring and identifies areas of research that should be addressed.

The NetFlow protocol originally designed by Cisco has been used for almost 20 years now [1]. The first version widely deployed was NetFlow version 5 [5]. The protocol was designed to provide information about network traffic up to the transport layer. The fixed message format was found to be inadequate and NetFlow version 9 [7] was introduced. However, even this protocol limits the data that can be transferred to several simple fixed-length types. Modern measurements often require a definition of new elements, possibly of variable length. Based on NetFlow v9, the IETF defined IP Flow Information Export protocol [9] (IPFIX). This protocol allows us to define private organization-specific elements with complex types and of variable length. The IPFIX protocol is essential for application flow monitoring since it allows us to transfer application information in a standardized manner. Cisco is also using the IPFIX protocol and proposed a specification of application information export [8]. Based on their work, new IPFIX information elements have been added to [16].

The current generation of application flow monitoring uses the same principles as the first generation of NetFlow. Each flow is created as an aggregation of packets with the same key elements [23]. These elements are taken from link, network and transport layers. The information about application protocols are added as new elements to existing flow records, as shown by [6]. Little research has explored the possibilities of using the application information to improve the measurement itself. In [10] we have shown that the information from IPv6 tunnels can be used to create flows with finer granularity. We expect that the information from application protocols can be utilized in a similar manner to create more detailed flow records. There are other unexplored possibilities for improvements in application flow monitoring. The measurement process can be optimized based on observed application, which might lead to performance and quality improvements.

The application-aware flow monitoring inspects the network traffic in more detail than IP flow monitoring. Providing more detailed information about the traffic has a negative impact on the monitoring process performance. Standard

flow cache [19] was not designed for application data. Therefore, methods to cope with the extensive amount of data gained from application monitoring need to be researched. The amount of data from measurements might cause an overload of the service network. We believe that possibilities of IPFIX data stream compression should be examined to find solution similar to [3].

The IP flow monitoring is designed to focus on communication consisting of packet streams. This approach is certainly useful for network management [4,11]. However, users are more focused on applications than the network itself. To comply with this trend, content providers need to ensure so called Quality of Experience, which reflects the user's subjective experience with a service. Individual flows might be generated as a result of single *event*. An event might be user opening a website or a server performing a planned synchronization. Information about several flows being part of one event is lost in the current application flow monitoring paradigm. This is a shortcoming that should be addressed.

One of the benefits of application flow monitoring over DPI is its processing speed. However, adding protocol analyzers for new applications degrades the performance. We have described this problem using HTTP protocol as an example in [25]. The authors of [21] propose an automated way to construct application parsers and also analyze their performance. However, the absence of a ground truth and methodology for comparing performance of application protocol analyzers makes comparison of different approaches difficult. Traffic traces used to evaluate anomaly detection methods are freely available and widely used [15]. Similar traces must be provided to create comparable conditions for the evaluation of application protocol analyzers, together with a suitable methodology.

3 Proposed Research and Approach

The aim of our work is to research a next generation flow monitoring system. The new system will be based mainly on the application layer instead of the network layer. To aid this research, we will investigate new approaches in application flow monitoring to make the measurement process more accurate and scalable. We have found that various services such as intrusion detection systems, quality of service and quality of experience measurements are significantly limited by the data provided by flow exporters. Providing high quality data will lead to improvements in all of these areas. The next generation flow monitoring will also enable the development of new methods for network security and management. The main research questions are as follows:

1. *How can information from multiple packet streams be aggregated to single application event and how can we utilize application events to design the next generation flow monitoring?*

Instead of working with flows based on packet streams, the next generation flow monitoring system will be based on events. One event may encompass more than one packet stream. We believe, that this architecture will allow the processing of more complex events than the collection of individual packet streams.

2. *What are the impacts of application protocol measurement on flow exporters?*

To research the next generation flow monitoring system, we need to understand how the addition of application layer information affects the flow exporters. We have already encountered several issues when adding application information to basic flow records. The extracted application data need to be stored for each flow record, which causes the flow records to grow significantly. The result is a large memory consumption of the flow cache, which in turn causes ineffective caching and performance degradation. We will measure the effects of large flow records and propose an alternative approach to solve this problem.

3. *How can application protocol information be used to improve flow measurement quality?*

Using information from the application layer can improve the flow measurement quality and efficiency. We believe that we can utilize application specific information to tailor the flow exporter for distinct protocols. For example, a DNS request is usually sent using only one packet, so the resulting flow record can be exported almost immediately. However, some applications, e.g., video streaming, generate large number of packets over a period of time [13], therefore the inactive timeout should be longer. By analyzing the behavior of major application protocols we can make the flow cache management more effective. If the short flows are exported sooner, the memory requirements of the flow cache might be decreased, which would improve overall performance. Research on flow cache timeouts without considering application protocol information was done in [22].

4. *What are the limits of application protocol measurement on high-speed networks?*

To evaluate the proposed next generation flow monitoring system, we need to build a prototype of a next generation flow exporter. We will use this exporter to compare the results of the new system with existing flow monitoring solutions [2,20]. Since the flow monitoring is frequently used for high-speed network monitoring [12,17,24], we need to design our flow exporter to handle such speeds. The processing of application protocols makes this task even more challenging, since each packet needs to be analyzed more thoroughly to gain the necessary information.

4 Conclusions

In this paper, we presented our goals for future research on application flow monitoring. We aim to analyze the impact of application protocol measurement on flow monitoring. Our contribution includes a study of the quality of generated flow data and of flow monitoring performance. We also propose the next generation application flow monitoring where the flows are merged into network events.

Acknowledgments. This material is based upon work supported by Cybernetic Proving Ground project (VG20132015103) funded by the Ministry of the Interior of the Czech Republic.

References

1. Brownlee, N.: Flow-Based Measurement: IPFIX Development and Deployment. IEICE Transactions on Communications 94(8), 2190–2198 (2011)
2. Network Situational Awareness group at CERT, C.M.U.: Yet Another Flowmeter (2014), http://tools.netsa.cert.org/yaf/, [cited 2014-01-18]
3. Chen, S., Ranjan, S., Nucci, A.: IPzip: A Stream-Aware IP Compression Algorithm. In: Data Compression Conference, 2008. DCC 2008. pp. 182–191 (March 2008)
4. Chen, T., Hu, L.: Internet Performance Monitoring. Proceedings of the IEEE 90(9), 1592–1603 (Sep 2002)
5. Cisco: NetFlow Export Datagram Format (2014), http://www.cisco.com/en/US/docs/net_mgmt/netflow_collection_engine/3.6/user/guide/format.html, [cited 2014-01-18]
6. Cisco: Network Based Application Recognition (NBAR) (2014), http://www.cisco.com/en/US/products/ps6616/products_ios_protocol_group_home.html, [cited 2014-01-18]
7. Claise, B.: Cisco Systems NetFlow Services Export Version 9. RFC 3954 (Informational) (Oct 2004), http://www.ietf.org/rfc/rfc3954.txt
8. Claise, B., Aitken, P., Ben-Dvora, N.: Cisco Systems Export of Application Information in IP Flow Information Export (IPFIX). RFC 6759 (Informational) (Nov 2012), http://www.ietf.org/rfc/rfc6759.txt
9. Claise, B., Trammell, B., Aitken, P.: Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information. RFC 7011 (INTERNET STANDARD) (Sep 2013), http://www.ietf.org/rfc/rfc7011.txt
10. Elich, M., Velan, P., Jirsík, T., Čeleda, P.: An Investigation Into Teredo and 6to4 Transition Mechanisms: Traffic Analysis. In: Damla Turgut, Nils Aschenbruck, Jens Tölle (ed.) 38th Annual IEEE Conference on Local Computer Networks (LCN 2013). pp. 1046–1052. Sydney, Australia (2013)
11. Estan, C., Keys, K., Moore, D., Varghese, G.: Building a Better NetFlow. In: Proceedings of the 2004 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications. pp. 245–256. SIGCOMM '04, ACM, New York, NY, USA (2004), http://doi.acm.org/10.1145/1015467.1015495
12. Estan, C., Varghese, G., Fisk, M.: Bitmap Algorithms for Counting Active Flows on High-speed Links. IEEE/ACM Trans. Netw. 14(5), 925–937 (Oct 2006), http://dx.doi.org/10.1109/TNET.2006.882836
13. Fioreze, T., Oude Wolbers, M., van de Meent, R., Pras, A.: Finding Elephant flows for optical networks. In: Integrated Network Management, 2007. IM '07. 10th IFIP/IEEE International Symposium on. pp. 627–640 (2007)
14. Gao, M., Zhang, K., Lu, J.: Efficient packet matching for gigabit network intrusion detection using TCAMs. In: Advanced Information Networking and Applications, 2006. AINA 2006. 20th International Conference on. vol. 1, pp. 6 pp.– (2006)
15. Gogoi, P., Bhuyan, M., Bhattacharyya, D., Kalita, J.: Packet and Flow Based Network Intrusion Dataset. In: Parashar, M., Kaushik, D., Rana, O., Samtaney, R., Yang, Y., Zomaya, A. (eds.) Contemporary Computing, Communications in Computer and Information Science, vol. 306, pp. 322–334. Springer Berlin Heidelberg (2012), http://dx.doi.org/10.1007/978-3-642-32129-0_34

16. IANA: IP Flow Information Export (IPFIX) Entities (2014), http://www.iana.org/assignments/ipfix, [cited 2014-04-07]
17. Iannaccone, G., Diot, C., Graham, I., McKeown, N.: Monitoring Very High Speed Links. In: Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement. pp. 267–271. IMW '01, ACM, New York, NY, USA (2001), http://doi.acm.org/10.1145/505202.505235
18. Lai, H., Cai, S., Huang, H., Xie, J., Li, H.: A Parallel Intrusion Detection System for High-Speed Networks. In: Jakobsson, M., Yung, M., Zhou, J. (eds.) Applied Cryptography and Network Security, Lecture Notes in Computer Science, vol. 3089, pp. 439–451. Springer Berlin Heidelberg (2004), http://dx.doi.org/10.1007/978-3-540-24852-1_32
19. Muenz, G., Claise, B., Aitken, P.: Configuration Data Model for the IP Flow Information Export (IPFIX) and Packet Sampling (PSAMP) Protocols. RFC 6728 (Proposed Standard) (Oct 2012), http://www.ietf.org/rfc/rfc6728.txt
20. ntop: nProbe (2014), http://www.ntop.org/products/nprobe/, [cited 2014-01-18]
21. Pang, R., Paxson, V., Sommer, R., Peterson, L.: Binpac: A Yacc for Writing Application Protocol Parsers. In: Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement. pp. 289–300. IMC '06, ACM, New York, NY, USA (2006), http://doi.acm.org/10.1145/1177080.1177119
22. Quan, L., Heidemann, J.: On the Characteristics and Reasons of Long-lived Internet Flows. In: Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement. pp. 444–450. IMC '10, ACM, New York, NY, USA (2010), http://doi.acm.org/10.1145/1879141.1879198
23. Sadasivan, G., Brownlee, N., Claise, B., Quittek, J.: Architecture for IP Flow Information Export. RFC 5470 (Informational) (Mar 2009), http://www.ietf.org/rfc/rfc5470.txt, updated by RFC 6183
24. Schuehler, D., Lockwood, J.: A Modular System for FPGA-Based TCP Flow Processing in High-Speed Networks. In: Becker, J., Platzner, M., Vernalde, S. (eds.) Field Programmable Logic and Application, Lecture Notes in Computer Science, vol. 3203, pp. 301–310. Springer Berlin Heidelberg (2004), http://dx.doi.org/10.1007/978-3-540-30117-2_32
25. Velan, P., Jirsík, T., Čeleda, P.: Design and Evaluation of HTTP Protocol Parsers for IPFIX Measurement. In: Bauschert, T. (ed.) Advances in Communication Networking, Lecture Notes in Computer Science, Vol. 8115. pp. 136–147. Springer Berlin / Heidelberg, Heidelberg (2013)