Distributed Architecture for Real-Time Traffic Analysis

Cristian Morariu and Burkhard Stiller

Department of Informatics, University of Zürich
CH-8050, Zürich, Switzerland
{morariu,stiller}@ifi.unizh.ch

Abstract. Traditional real-time IP traffic analysis applied on todays' high-speed network links suffers from the lack of scalability. Although sampling proves to be a promising approach, there are application scenarios foreseen, in which decisions cannot be based on sampled data, *e.g.*, for usage-based charging or intrusion detection systems. Moreover, traditional traffic analysis mechanisms do not map the traffic observed in the network to a particular user, but rather to a particular end-node, which may have been shared by several users. Thus, DARTA (Distributed Architecture for Real-time Traffic Analysis) develops a model for distributed IP traffic analysis and introduces new mechanisms for three different aspects in IP traffic monitoring: (a) a framework enabling the development of distributed traffic analysis applications, (b) a distributed packet capture mechanism, (c) an user-based IP traffic accounting for mapping IP traffic to individual users.

1 Introduction

Since the first days of Internet, the traffic carried by network operators increased year by year. Studies have shown that the yearly increase of traffic observed in several large network operators during the last decade ranged between 50% and 100% [5], [6], [8] with steep increases in the last years for the mobile Internet traffic segment. Moreover, [1] sees this trend to continue at least until 2012, when the Internet traffic will grow to approximately 75 times larger than the Internet traffic of 2002. Traffic increase not only impacts the routing and switching infrastructure of an operator, but also his metering, monitoring, and accounting infrastructure which are vital operations for a modern network. If the network traffic increased about 50%-100% every year in the last decade, the memory access speeds only improved about 7-9% [7] per year during the same period. As a result, today, network operators either a) reduce the traffic they inspect by using sampling or aggregation, which reduces the accuracy of analysis applications or b) use hardware specialized in some specific traffic monitoring or analysis tasks, which is usually very expensive and less flexible than a software traffic analysis application.

2 Motivation

Solving traffic monitoring problems by distributing data to several nodes was proposed several times for solving specific problems. Although already existing proposals show that distribution may improve performance of traffic monitoring and analysis applications running in high-speed traffic environments, each of those solutions was designed and tuned for a specific problem. The first aspect which this thesis investigates is ***how to design a generic and flexible architecture of a distributed system which enables distributed traffic analysis***. The results of this work resulted in SCRIPT, which is a both, a framework for building distributed traffic analysis applications and an im-

B. Stiller and F. De Turck (Eds.): AIMS 2010, LNCS 6155, pp. 171–174, 2010.

plemented prototype platform. The second aspect is ***how can the performance of traffic monitoring applications which run on off-the-shelve PCs be improved?*** Many traffic monitoring applications were built for the Linux operating system and make use of the *libpcap* of *libpcap-PFRING* libraries to access the packets on the network link. If was observed that at high packet rates these libraries cause the operating system to spend most of its resources on capturing packets, while leaving less resources for the monitoring application, thus, causing an overload of the system which eventually leads to dropped packets. DiCAP, developed in this thesis allows several PCs running the same monitoring application to share the monitoring workload by splitting the observed packets between themselves. Finally, the third problem is ***how can network traffic be mapped to individual users or even processes and applications***? The traditional way to address this problem is to assume that an IP address is used by a single user at a time and have a mapping between IP address-to-user mapping all times. A problem arises when the end systems are multi-user capable and several users run at the same time network applications (*e.g.* background bittorrent applications). In this case an IP-to-user mapping is not possible anymore, as two consecutive packets from the same IP address may be produced by applications of two different users; the solution is LINUBIA.

3 DARTA Approach

The distributed traffic monitoring and analysis model proposed is not intended to address a single particular problem, but to cover a larger area of problems of high speed network monitoring. Figure 1 shows the proposed architecture for distributed IP traffic monitoring and analysis. It shows a layered architecture including a metering layer, a monitoring and analysis layer, and a presentation layer. The distributed metering layer includes one or more metering systems which are responsible with extracting the relevant data from the observed traffic. In order to cover also the user-based IP accounting problem this layer includes a model for general packet capture and processing, and another model for user-based IP traffic accounting.

The second layer shown in Figure 1 represents the traffic analysis task which is also distributed. IP traffic data is exchanged between the first and the second layer as IPFIX records and uses internal mechanisms to forward these records to analysis application instances that may use them. Finally, the third layer represents a presentation system which includes interfaces that allow a human administrator to visualize the results of the analysis process. As the presentation system is dependent on the analysis application it was not cover by the thesis, but an API is described which allows a presentation system to be *hooked* in the distributed analysis system.

Figure 2 shows a general model for distributed traffic analysis. In a network there are multiple *Network Components* (such as routers, switches, links, services, etc) that need to be monitored. The operation of these components is observed and measured by one or more *Meters*. One meter can measure more than a single component, for example it could measure traffic aggregated from several routers. At the same time a network component can be metered by multiple meters, for example one meter doing packet-level measurements, and a second doing flow-level measurements. The metered data, once it is produced, needs to be sent (or exported) to one or more *Data Collectors.* These data collectors may perform limited pre-processing tasks, such as aggregation, anonymiza-

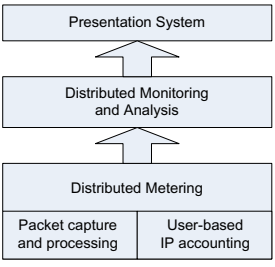

Fig.1. Distributed IP Traffic Metering and Analysis Architecture

tion, filtering, or encapsulation, which prepare the data to be used by traffic analysis applications, before feeding the received data to a *Traffic Processing Platform.* The encapsulation process is of particular importance as its task is to switch the format of the received metered data (*e.g.* SNMP, NetFlow v5, Diameter, IPDR, proprietary protocols) to IPFIX which is used by the traffic processing platform. The traffic processing platform consists of one or more *Processing Units.* Each processing unit runs one or more *Traffic Analysis Applications.* It is the task of the traffic processing platform to feed each piece of metering data to the right analysis application instance. The results of the traffic analysis applications are fed to a *Presentation* component which presents them to a *User* or *Administrator*. The presentation component also maintains a relation with the underlying traffic processing platform which allows it to access different traffic application instances.

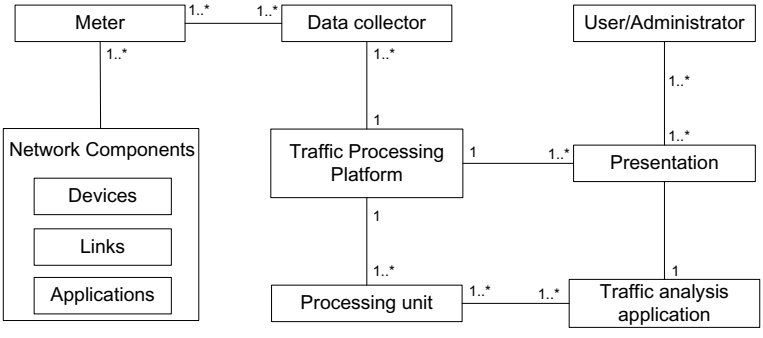

Fig. 2. Generic Model for Distributed Traffic Analysis

4 Prototype and Evaluation

Three different distributed mechanisms have been developed in order to validate the distributed traffic metering and analysis model. The first mechanism named DiCAP [2] allows distributed packet capture using a libpcap-based application on a high-packet rate link. As its evaluation shows, DiCAP significantly improves (up to 10 times) the amount of packets that can be processed with four machines in parallel. The second metering mechanism named Linubia allows per-user traffic accounting in Linux end-hosts. It works for both IPv4 and IPv6 and as the evaluation in [3] shows it only introduces a very small overhead in processing a packet.

The third mechanism developed in this thesis is SCRIPT [4] a framework which can be used to build and deploy distributed traffic analysis applications. SCRIPT is generic enough to support any type of traffic analysis application, as long as it uses IPFIX to transport traffic data. Several prototypes for different traffic analysis applications have been implemented and the evaluation shows that SCRIPT fairly distributes workload among different traffic analysis nodes, and that increase in traffic can be addressed by adding new traffic analysis nodes.

5 Concluding Remarks

DARTA solves major challenges of IP traffic metering and analysis in high speed networks, therefore, the newly designed set of distributed mechanisms for handling traffic data can be used by future network management infrastructures. A generic distracted traffic analysis framework (SCRIPT) has been designed and prototypically implemented. Besides, two different distributed metering mechanisms (DiCAP) for capturing traffic on high-packet rates, and Linubia for mapping IP traffic to individual users on Linux end-hosts have been developed. As the evaluation of these mechanisms shows they increase the amount of traffic that can be handled by analysis applications by combining computational and storage resources from multiple devices. As the evaluation of Linubia shows, retrieving granular metering information (such as the user or process which generated a packet) from Linux end-devices is feasible as it only introduces limited overhead.

Acknowledgements

This work was supported in part by the Cisco University Research Program Fund, the SNF DaSAHIT project, and the IST NoE EMANICS .

References

[1] Cisco Systems: Hyperconnectivity and the Approaching Zettabyte Era (June 2009)
[2] Morariu, C., Stiller, B.: DiCAP: Distributed Packet Capturing Architecture for High-Speed Network Links. In: 33rd Annual IEEE Conference on Local Computer Networks (LCN), Montreal, Canada (October 2008)
[3] Morariu, C., Feier, M., Stiller, B.: LINUBIA: A Linux-supported User-Based IP Accounting. In: Clemm, A., Granville, L.Z., Stadler, R. (eds.) DSOM 2007. LNCS, vol. 4785, pp. 229–241. Springer, Heidelberg (2007)
[4] Morariu, C., Racz, P., Stiller, B.: SCRIPT: A Framework for Scalable Real-time IP Flow Record Analysis. In: 12th IEEE/IFIP Network Operations and Management Symposium (NOMS 2010), April 2010. IEEE, Osaka (2010)
[5] Minnesota Internet Traffic Studies (MINTS), `http://www.dtc.umn.edu/mints/home.php` (Last accessed: February 2010)
[6] Odlyzko, A.M.: Internet Traffic Growth: Sources and Implications. In: Proceedings of SPIE, August 2003, vol. 5247, pp. 1–15 (2003)
[7] Patterson, D.A., Hennessy, J.L.: Computer Organization and Design, 4th edn. Morgan Kaufmann, San Francisco (2008)
[8] Roberts, L.G.: Beyond Moore's Law: Internet Growth Trends. IEEE Computer Magazine (January 2000)