Lehrstuhl für Netzarchitekturen und Netzdienste
Fakultät für Informatik
Technische Universität München

Antrittsvortrag zur Bachelor arbeit

Leistungsanalyse der Funktionen von Middleboxes

Name:	Simon **Sternsdorf**
Betreuer:	Florian Wohlfart
Aufgabensteller:	Prof. Dr.-Ing. Georg Carle
Beginn:	04/2017
Ende:	08/2017

Topic

In this Bachelor's Thesis we test different implementations of middleboxes. We will set up different middleboxes in our testbed and then measure the performance of packet-processing. Our advanced goal is to identify different bottlenecks in modern middlebox implementations. As a common middlebox we will quantify NAT-boxes.

The intended field of application is very broad. NAT-Boxes are used in big ISP networks as well as in home-routers. ISPs often use middleboxes in their setups to cache or even modify user data. End-user routers mostly use NAT to give all devices in their network an private IP address as IPv4 addresses are expensive and most End-users only get one from their provider. NAT-boxes also play an important role in cellular networks, where ISPs use them to give private IP addresses to mobile devices.

Our goal is to split up an NAT implementation in different smaller parts and to measure them separately. We hope to get statements about the performance of variant parts of NAT-Boxes. Also we want to create a more general model of an NAT-Box out of the collected data.

Approach

First we will research good implementations of middleboxes. We want to split up NAT in different components to measure them seperately. This way we can get an better understanding of which part of the NAT is actually responsible for how much of the time we need to forward a packet. We will first implement a stateless firewall to measure how much time it takes to parse a packet. Afterwards we will implement an state-full firewall to measure the costs of holding a state. And finally we will test an full NAT implementation. There we will measure the extra costs of rewriting the IP address and the port in a packet.

For the beginning we want to test our general idea with iptables, as iptables is easy to set up and you can split up the 3 different parts of an NAT we want to test there. The performance of iptables will probably be very mediocre but it is well-suited as a starting point.

To generate traffic we will use Moongen, an high-speed traffic generator. Moongen send out packets to our testserver and also measure different metrics when they come back. Here we will experiment with different packet sizes, Ipv4 vs IPv6, different ports, IP addresses and more. Our goal is to get good traffic, traffic where we can see anomalies in our measured data.

Moongen can also function as an firewall implementation. We will test Moongen to see if it has differences in his performance compared to other implementations.

We aim to measure the packet rate at which the middlebox can process the packets, the power consumption of the server under maximum load, the RAM usage or the Latency.

To make these tests replicable we will automate the procedure completely with different shell and python scripts. The goal here would be to have a script with different input parameters. This way we can get reproducible results.

We will then evaluate the data. Our goal here is to find a correlation between different benchmark data and the performance of our middlebox. Also we want to identify which part of the NAT-Box is responsilbe for which delay. A secondary goal would be to find the right models to explain our results.

Previous work

The thesis is thematically similar to the Bachelor's Thesis *A Model for Performance Prediction in PC-based Packet Processing Systems* [1] by Dominik Scholz. He tested the general performance of the Linux Network Stack and generated a performance prediction model for some use cases. He did not directly test middlebox implementations and especially no NAT implementations. So we can follow up his thesis and use similar methods to test middleboxes in our testbed.

To help use with the implemenation of the middlebox we can use different APIs for modular development of middleboxes. An example would be mOS, described in *mOS: A Reusable Networking Stack for Flow Monitoring Middleboxes* [2]. Such an reusable networking stack will be useful when setting up different middleboxes in the testbed.

To generate good traffic we will orient ourselfs towards the traffic generated in *Implementation and Performance Analysis of Firewall on Open vSwitch* [**?**]. In this work they tested firewall performance, so very similar to our use case.

If we want to use a middlebox with deep packet inspection BlindBox might come in handy, described in *BlindBox: Deep Packet Inspection over Encrypted Traffic* [3]. BlindBox builds on the ClickOS framework , described in *ClickOS and the Art of Network Function Virtualization* [**?**].

Time Management

Tasks	**Start Date**	**End Date**
Research about middlebox implementations, induction in the testbed	15-April	15-May
Testing of middlebox implementations, data collection	15-May	15-June
Evaluate the data, find models for characteristic effects	1-June	1-July
Write the thesis and print it	1-July	15-August

Literatur

[1] D. Scholz, "A model for performance prediction in pc-based packet processing systems," 2014.

[2] Y. M. Muhammad Jamshed and D. Kim, "mos: A reusable networking stack for flow monitoring middleboxes."

[3] J. Shah, "Implementation and performance analysis of firewall on open vswitch," 2015.

[4] C. L. Justine Sherry and R. A. Popa, "Blindbox: Deep packet inspection over encrypted traffic," 2015.

[5] M. A. e. a. Joao Martins, "Clickos and the art of network function virtualization," 2014.