Transparent Cloud Access Performance Augmentation via an MPTCP-LISP Connection Proxy

Yacine Benchaïb
Sorbonne Universités, UPMC
Univ Paris 06, UMR 7606,
LIP6, F-75005, Paris, France
yacine.benchaib@upmc.fr

Stefano Secci
Sorbonne Universités, UPMC
Univ Paris 06, UMR 7606,
LIP6, F-75005, Paris, France
stefano.secci@upmc.fr

Chi-Dung Phung
Sorbonne Universités, UPMC
Univ Paris 06, UMR 7606,
LIP6, F-75005, Paris, France
chi-dung.phung@upmc.fr

ABSTRACT

The use by a growing number of users of Cloud-based services requires an adaptation of the network technologies used to access them. We propose to combine two novel protocols at the state of the art at Cloud access middle-boxes to better profit from spare unused network path diversity. The first protocol, Multipath TCP, allows creating multiple TCP/IP subflows, as much as needed. The second, the Locator/Identifier Separation Protocol (LISP), can be used to route the subflows on different wide-area network paths, possibly disjoint, and also allows native support for seamless virtual machine migrations. In this paper we specify how we can combine these two protocols to increase the bandwidth available to access applications run in multi-homed data-centers. We describe how these protocols can be integrated into a Cloud access middle-box. By means of a combined MPTCP-LISP access proxy, the acceleration is transparent to the user terminal that does not necessitate any upgrade. We provide the detailed system-level architecture based on open source code, and we document results from preliminary experimentations on one of two targeted use-cases. The evaluations conducted show that the overhead generated by our solution remains moderate despite the various system-level steps required to translate incoming TCP packets into MPTCP-LISP packets then routed over different IP paths.

Categories and Subject Descriptors

C.2 [**Computer Systems Organization**]: Computer Communication Networks; C.2.2 [**Computer Systems Organization**]: Computer Communication Networks—*Network Protocols*

General Terms

Cloud access protocol, data-center networking

1. INTRODUCTION

A promising way to improve cloud access performance is to design bandwidth aggregation techniques at edge networks. The edge networks involved in cloud access are, obviously, the user's access network and the data-center network. Data-center networks are commonly multi-homed with multiple transit providers to increase reliability. Moreover, many cloud providers geographically distribute their data-center fabric over multiple distant sites [1], and therefore could offer multiple network entry points to the users. The technology challenge in this context is twofold: (i) how to take profit from unused network path diversity between the user network and the data-center network, and (ii) how to let single applications transparently taking profit from bandwidth aggregation over the available paths, without modifying the user's terminal for the sake of scalability.

In a previous work [2] it was investigated the usage of two novel protocols standardized by the Internet Engineering Task Force (IETF) in the recent years: the Multipath Transport Control Protocol (MPTCP) [3], an incrementally deployable extension of TCP that creates sub-connections (called MPTCP subflows), each identified by a different TCP port-based and IP address-based 4-tuple; the Locator/Identifier Separation Protocol (LISP) [4], an IP-to-IP encapsulation protocol architecture including a distributed control-plane (similar to the Domain Name System one) to map destination IP address to routing locator (RLOC) IP address. In [2], it is proposed to modify the MPTCP implementation at endpoints to allow user's terminal to query the LISP mapping system and get the number of RLOCs of the server, and hence to open as much subflows as the number of RLOCs, gambling on a high level of disjointness between the corresponding Internet paths and hence decreasing the risk of passing through bottleneck and traffic shaping middle-boxes. The solution was coined Augmented MPTCP (A-MPTCP).

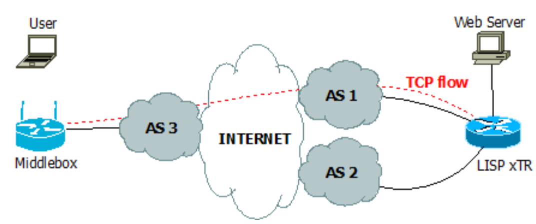

Figure 1: A-MPTCP middle-box use-cases

Even if the intuition in [2] proved to be technically viable, with important achievable throughput gains in multi-homed data-centers situations, it has two major impediments: the requirement to deploy a custom MPTCP version implemented at user's terminal operating system, and the requirement that the user's IP endpoint is a LISP network end-point identifier. The solution presented in this paper aims to be a more scalable A-MPTCP alternative. It lets the user running TCP connections as usual, and without imposing a LISP IP addressing to the user endpoint. This is possible using an A-MPTCP connection proxy middle-box at the access network gateway that translates TCP connections into MPTCP connections, natting the IP source information, while routing the MPTCP subflows on different IP-level paths by means of LISP encapsulation. The reference cloud access use case we address is represented in Fig. 1, with the data-center network that has multiple entry points (in the figure it is supposed to be multi-homed with two Autonomous System, AS, providers; it could also be multi-homed with the same provider but at different geographically distributed sites). The assumption is that the data-center providers opens to cloud users' networks the access to a LISP mapping system (in fact, a map-resolver server), exposing RLOCs of server IP addresses.

2. SYSTEM ARCHITECTURE

The A-MPTCP middle-box system architecture relies on the usage of a connection proxy to operate the TCP-to-MPTCP translation. The proxy terminates transparently (without notifying the source) incoming TCP connections to then transparently re-open the connection toward the destination, using another IP address that belongs to a LISP network. By installing MPTCP at the kernel level of the A-MPTCP node, the reopened connection can be an MPTCP connection. Before exiting the A-MPTCP node, MPTCP packets are by default routed to a custom LISP software router that routes different subflows to different RLOCs by means of LISP encapsulated MPTCP packets. When creating the MPTCP connection, the MPTCP subflow discovery module can be customized to open as many subflows as RLOCs [2], minimizing the signaling overhead and simplifying the routing operation. Fig. 2 provides an illustration of

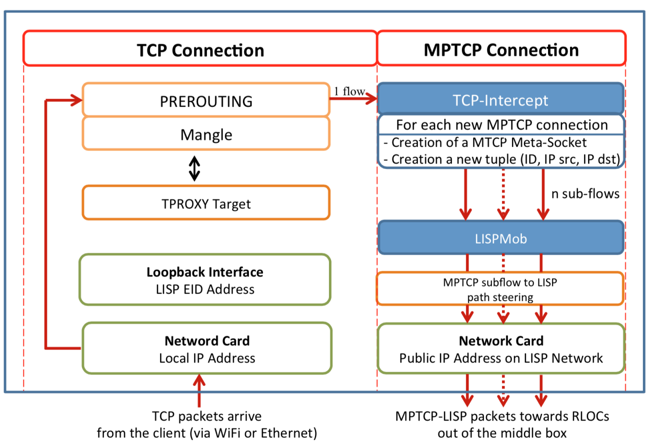

Figure 2: Inside the A-MPTCP middle-box

the system architecture. The software we used to implement the A-MPTCP middle-box are as follows:

- TCP-Intercept[1], a transparent TCP connection proxy that uses the Linux firewall TProxy target[2], and that is able to translate a TCP connection to an MPCTP connection.
- LISPmob[3] is a LISP software ingress/egress tunnelling router (ITR/ETR, or xTR) available for Linux that, upon proper customization for explicit routing, absolves the LISP encapsulation function. It implements the control-plane functions to interact with mapping servers, resolvers and data-center xTRs.
- the open source MPTCP Linux kernel implementation[4] for both the A-MPTCP middle-box and the server.

3. EVALUATION AND FURTHER WORK

To evaluate the system overhead, we compare the net bandwidth between two local nodes communicating via a Linux node under three situations: a node with Network Address Translation (NAT) only, a node with TCP-Intercept only and a node with the full A-MPTCP functionalities. The results on Figure 3 show that the decrease in bandwidth performance is of roughly 18%. Given that NAT and TCP-Intercept solutions have similar performance, this decrease is essentially due to LISP encapsulation. Indeed, LISPmob data-plane runs in the user space. We believe this small gap can be largely reduced by running the LISP data-plane operations at the kernel level, for example by replacing LISPmob with OpenVSwitch[5] (having LISP data-plane functions), or with the FreeBSD-based OpenLISP[6] jointly with a FreeBSD-based MPTCP implementation[7]. Further work is also need to integrate LISP Traffic Engineering extension in the A-MPTCP middle-box for advanced routing.

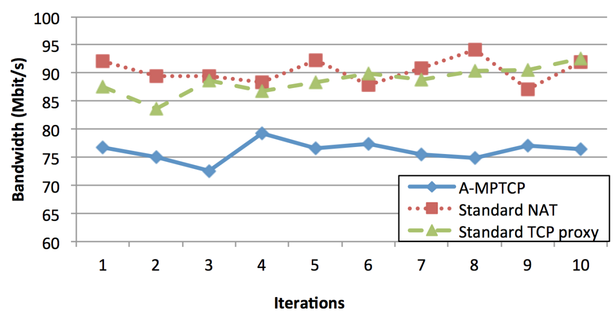

Figure 3: A-MPTCP bandwidth overhead measures.

Acknowledgment

This work was partially funded by the European Institute of Technology (EIT), ICT-Labs Future Networking Services action line, the FUI 15 project RAVIR, and the ANR LISP-Lab project (grant nb: ANR-13-INFR-009).

4. REFERENCES

[1] S. Secci and S. Murugesan. Cloud networks: Enhancing performance and resiliency. *Computer*, 47(10):82–85, Oct 2014.

[2] M. Coudron, S. Secci, G. Pujolle, P. Raad, and P. Gallard. Cross-layer cooperation to boost multipath tcp performance in cloud networks. In *IEEE CLOUDNET 2013*, Nov 2013.

[3] Sébastien Barré, Christoph Paasch, and Olivier Bonaventure. Multipath tcp: From theory to practice. In *NETWORKING 2011*, volume 6640 of *LNCS*, pages 444–457. Springer, 2011.

[4] David Meyer and Darrel Lewis. The Locator/ID Separation Protocol (LISP). RFC 6830, 2013.

[1] https://github.com/VRT-onderzoek-en-innovatie/tcp-intercept
[2] https://www.kernel.org/doc/Documentation/networking/tproxy.txt
[3] http://lispmob.org
[4] http://www.multipath-tcp.org
[5] http://www.openvswitch.org
[6] https://github.com/lip6-lisp
[7] http://caia.swin.edu.au/urp/newtcp/mptcp