Experiences with MPTCP in an intercontinental multipathed OpenFlow network

Ronald van der Pol*, Michael Bredel†, Artur Barczyk†

*SURFnet
Radboudkwartier 273
3511 CK Utrecht, The Netherlands
Email: Ronald.vanderPol@SURFnet.nl

†California Institute of Technology, c/o CERN
1211 Geneva, Switzerland
Email: michael.bredel@cern.ch, Artur.Barczyk@cern.ch

KEYWORDS

MPTCP, OpenFlow, Floodlight, multipath, big data

ABSTRACT

In this paper we present our experiences with multipath TCP (MPTCP) in an intercontinental multipathed OpenFlow testbed. We believe that multipathing will become an important network concept in the next few years. One of the major reasons is that data sets in e-science are increasing exponentially in size. To transfer these huge data sets we need to make efficient use of all available network capacity. This means using multiple paths simultaneously wherever possible. To get experience with this concept we set up a testbed in which we transferred data between two servers. The testbed consisted of OpenFlow switches and multiple link disjoint paths between the servers. An OpenFlow application provisioned multiple paths between the servers and MPTCP was used on the servers to simultaneously send traffic across all those paths. Figure 1 shows the testbed that was

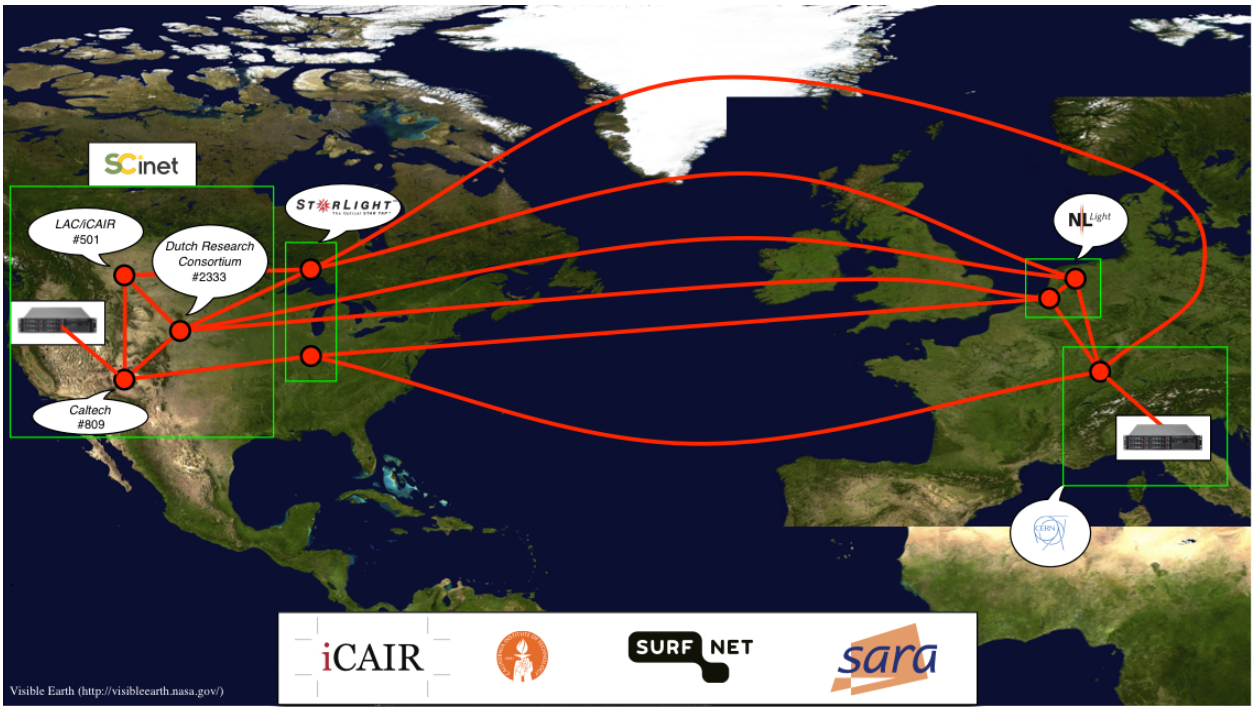

Fig. 1. SC12 Multipath OpenFlow Network Topology

used for the SC12 demonstration in Salt Lake City in November 2012.

Multipathing can be done at L3 with Equal Cost Multipath (ECMP) routing or at L2 with protocols like TRILL (IETF RFC 5556) [1] or IEEE 802.1aq (Shortest Path Bridging - P802.1aq-2012 [2]). In all these cases load balancing across the paths is done based on flows by calculating a hash (based on e.g. Ethernet addresses, IP addresses and TCP/UDP port numbers) of the packets. Each packet of such a flow follows the same path through the network, which prevents out of order delivery within a flow. When the traffic has many different flows the traffic will be evenly spread across the various paths. But when there are only a few flows, which is typically the case in large data e-science applications, this is not the case. Another disadvantage of hashing is that usually all links get the same percentage of the hash values and therefore all the paths need to have the same capacity.

Multipath TCP is a new approach towards efficient load balancing. Instead of letting the network do the load balancing by using hashes and ECMP, MPTCP is doing the load balancing in the end nodes as part of the TCP process. Multipath TCP (MPTCP) is described in RFC 6182 [3] and the 'TCP Extensions for Multipath Operation with Multiple Addresses' internet draft [4]. MPTCP is an active working group in the IETF. Figure 2 shows how MPTCP works. In a MPTCP enabled kernel

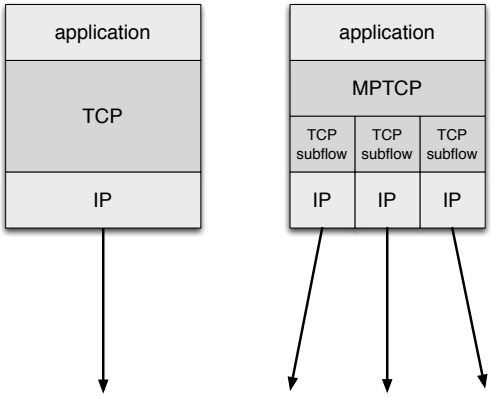

Fig. 2. Traditional TCP vs Multipath TCP

the TCP component is split in a MPTCP component and TCP subflow components for each interface. The MPTCP component receives a byte stream from the application (MPTCP uses an unmodified socket API and TCP semantics northbound, so applications do not need to be adapted). The MPTCP component splits the byte stream into multiple segments which are handed to the TCP subflow components. Each subflow behaves as a normal TCP flow to the network. MPTCP can handle paths of different bandwidth because there is a congestion control mechanism across the subflows. This congestion control mechanism takes care that traffic on a congested path is moved to a path with less congestion. So it adapts the load balancing according to the load of other traffic on the network.

The MPTCP component implements three functions. It takes care of path management by detecting and using multiple paths to a destination. Packet scheduling splits the byte stream received from the application in multiple segments and transmits these segments on one of the available subflows. These segments are numbered, so that the receiving MPTCP component can put the segments in the correct order and reconstruct the original byte stream. Finally there is congestion control across the subflows. This function spreads the load over the subflows. When a subflow becomes congested, traffic is moved to a subflow that is less congested. This function also takes care of retransmissions on another subflow when one of the subflows fails.

In 2012 two demonstrations were given, one in October in Chicago during the GLIF meeting and one in November in Salt Lake City during SC12. During the GLIF meeting we showed streaming from Geneva to Chicago over multiple 10GE paths. On our servers we used the Linux MPTCP implementation of the IP networking lab of the Université de Louvain in Belgium [7]. Both servers had two 10GE NICs each. On these physical interfaces we configured two MAC VLAN virtual interfaces so that we could give each virtual interface its own MAC address. In our testbed the various paths through the network are set up by provisioning forwarding entries on the OpenFlow switches. Each of the four virtual interfaces was mapped to its own path through the OpenFlow network and each path had it own IP subnet assigned to it. The OpenFlow forwarding entries matched on destination MAC and IP address. There was no routing in our testbed, so four subflows (paths) could be used by MPTCP. During the GLIF demonstration we were able to reach an aggregated end-to-end throughput of around 3 Gbit/s. Later analysis showed that we used an MPTCP segment size of only 1400 bytes, which was probably one of the reasons for the relatively low throughput. During SC12, we showed OLiMPS and MPTCP by streaming from Geneva to Salt Lake City as part of the SCInet Research Sandbox. Figure 1 shows the topology used during SC12. Figure 3 shows the live monitoring website at SC12. We have also streamed between Geneva and Amsterdam and by using four paths we were able to reach an aggregated end-to-end

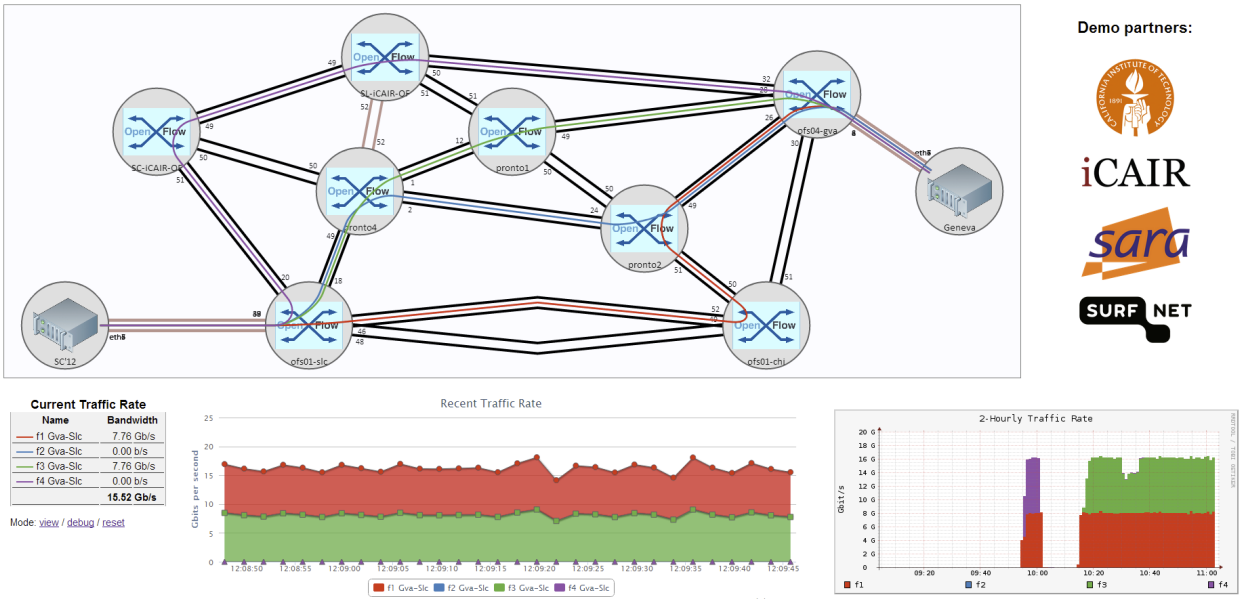

Fig. 3. Live monitoring website at SC12

throughput of around 13 Gbit/s. Each of the four subflows is mapped to one of the four paths and has its own colour in the graphs (see figure 4). When the stream was started all four subflows were used. After a while only the red and green subflows

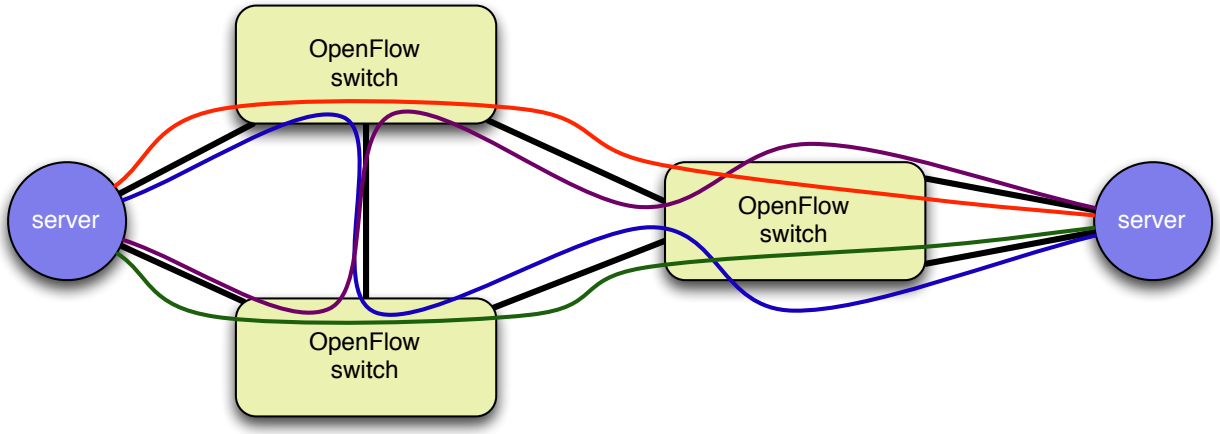

Fig. 4. Four paths between Geneva and Amsterdam

were used. These are the only two link disjoint paths between the servers, so it makes sense that MPTCP would eventually use only these paths. Figure 5 shows how initially all flows are used and after some time only the two link disjoint paths. Figure 6 show the subflow usage after some time. In the first half of 2013 we will continue this work in several areas:

- MPTCP throughput measurements between Geneva, Amsterdam and Chicago
- MPTCP throughput on links of different bandwidth (e.g. 1GE and 10GE)
- Continue the development of the OLiMPS OpenFlow application
- MPTCP and multipathing experiments on the Géant OpenFlow testbed
- Demo at Joint Techs in Hawaii
- Demo at TNC2013

The results of these activities will be presented at TNC2013.

ACKNOWLEDGEMENTS

The work of SURFnet is made possible by the GigaPort program of the Dutch Government and the Géant3 project of the European Commission. Caltech research in this area is supported by the DOE Office of Advanced Scientific Computing

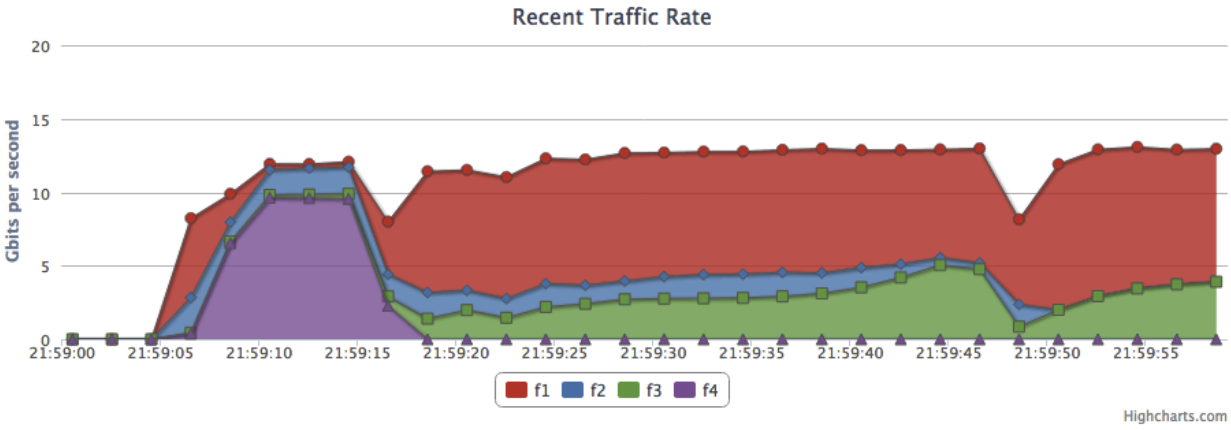

Fig. 5. Streaming between Geneva and Amsterdam, initial phase

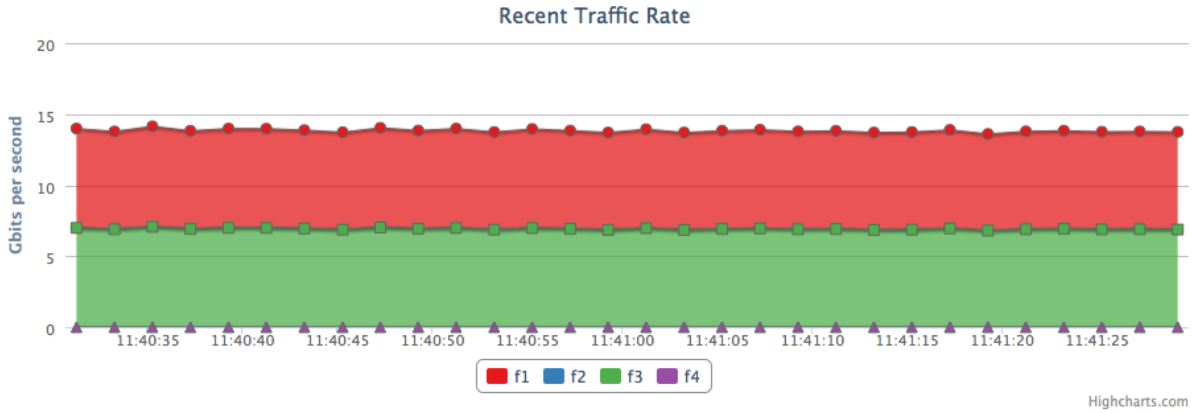

Fig. 6. Streaming between Geneva and Amsterdam, stable phase

Research (OASCR).

REFERENCES

[1] J. Touch and R. Perlman, Transparent Interconnection of Lots of Links (TRILL): Problem and Applicability Statement, RFC 5556, May 2009
[2] IEEE Approved Draft Standard for Local and Metropolitan Area Networks: Bridges and Virtual Bridged Local Area Networks - Amendment 9: Shortest Path Bridging, IEEE 802.1aq-2012, January 2012
[3] A. Ford, C. Raicu, M. Handley, S. Barre and J. Iyengar, Architectural Guidelines for Multipath TCP Development, RFC 6182, March 2011
[4] A. Ford, C. Raiciu, M. Handley, O. Bonaventure, TCP Extensions for Multipath Operation with Multiple Addresses, draft-ietf-mptcp-multiaddressed-08, November 2012
[5] Floodlight OpenFlow Controller
http://floodlight.openflowhub.org/
[6] IEEE Standard for Local and Metropolitan Area Networks - Station and Media Access Control Connectivity Discovery, IEEE 802.1AB-2009, ISBN 978-0-7381-6038-2, September 2009
[7] Multipath TCP - Linux Kernel implementation
http://mptcp.info.ucl.ac.be/

VITAE

Ronald van der Pol is a network researcher at SURFnet. He has been working in the field of Education and Research Networks for more than twenty years. His former employers include the VU University in Amsterdam, NLnet Labs and SARA. His current interests are in new network technologies and how these can be applied to next generation networking. In recent years he worked on monitoring and management of optical transport networks, carrier Ethernet, end-to-end performance of demanding applications and OpenFlow. He holds masters degrees in both Physics and Computer Science and is co-author of several IPv6 RFCs.

Michael Bredel is a network research engineer at Caltech based at CERN. He received a Diploma of Electrical Engineering from the Technische Universität Darmstadt, Germany and a Ph.D. from the Leibniz Universität Hannover, Germany. His current

research activities are related to network measurements, network management, software defined networks and OpenFlow. His particular interests are new technologies that can improve the movement of Big Data, like for the Large Hadron Collider.

Artur Barczyk is working since 2007 in the Caltech HEP Networking group as a senior research engineer. He is the technical team lead in US LHCNet, the transatlantic, mission-oriented network for LHC data. Artur holds a PhD in Particle Physics.