MultipathTCP

Multipath TCP - MPTCP

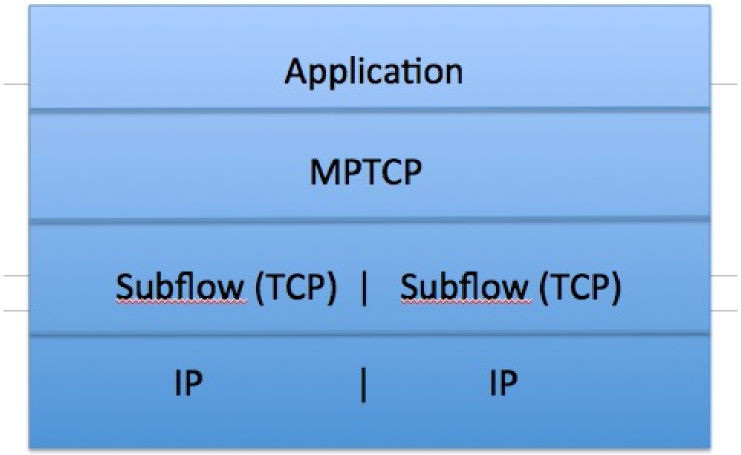
MPTCP is an extension to regular TCP that enables a transport connection to operate across multiple paths simultaneously.

It is important to underline that it is an extension of an existing protocol. It is not a new protocol. MPTCP adds multipath capabilities to TCP. It doesn’t change anything in TCP itself and wherever TCP works, it will still work even with the new additions.This condition guarantees that MPTCP is backwards compatible and that can always fall back to regular TCP when something goes wrong. The benefits of having a multipath transport are different. It increases the resilience of the connectivity and it improves the throughtput by means of a better usage of the available resources. The main assumption for MPTCP to work is that the end-hosts are multihomed.

MPTCP in the Networking Stack

The key behind the ability of MPTCP of effectively implementing the concept of Multipath is decoupling TCP from IP.

MPTCP operates at the transport layer and aims to be transparent to both higher and lower layers. It is a set of additional features on top of standard TCP.



How MPTCP works

Multipath TCP enables the hosts to use different paths with different IP addresses to exchange packets belonging to the MPTCP connection.A Multipath TCP connection appears like a normal TCP connection to an application as well as to the network layer each MPTCP subflows looks like a regular TCP flow whose segments just carry a new TCP option type. Multipath TCP manages the creation, removal and utilization of these subflows to send data. The number of subflows that are managed within a Multipath TCP connection is not fixed and it can fluctuate during the lifetime of the Multipath TCP connection.

MPTCP starts a new connection using the same approach and signaling as for initiating a normal TCP connection. The only difference is the presence of a new option in the SYN, SYN/ACK and ACK packets. The MP_CAPABLE option allows the hosts to exchange some information (keys) that is used to authenticate the establishment of additional subflow. Every new additional subflow begins like a normal TCP connection, but this time the SYN, SYN/ACK and ACK packets will carry the MP_JOIN option. This option contains a token generated from the key that is used to identify which MPTCP connection the subflow is joining.

As said before, the number of subflows can fluctuate during the lifetime of the connection. To ensure reliable delivery of data over subflows that may appear and disappear, MPTCP uses something called Data Sequence Number (DSN) to number all data sent over the MPTCP connection. The "Data Sequence Signal” (DSS) option allows mapping the subflow sequence space to the data sequence space. In this way, data can be retransmitted on different subflows in the event of failure. The DSS can also carry a connection-level acknowledgement (the "Data ACK") for the received DSN. With MPTCP, all subflows share the same receive buffer and advertise the same receive window. There are two levels of acknowledgement in MPTCP. Regular TCP acknowledgements are used on each subflow to acknowledge the reception of the segments sent over the subflow independently of their DSN. In addition, there are connection-level acknowledgements for the data sequence space. These acknowledgements track the advancement of the bytestream and slide the receiving window.

When there are no more data to send, a "Data FIN" is used as part of a DSS. Basically it has the same semantics and behavior as a regular TCP FIN, but at the connection level. Once all the data has been successfully received at the other side, this message is acknowledged at the connection level with a DATA ACK.

Something to keep in mind

Because of its backwards-compatibility the protocol must function through the majority of existing middleboxes such as NATs, firewalls and proxies. As consequence of this it has to be kept in mind that the segments sent on the wire will most likely arrive modified at the destination and TCP options may be removed or duplicated. This constraint has heavily influenced the design of MPTCP and its ability to fall back to standard TCP when something goes wrong.

Related work

Quentin De Coninck and Olivier Bonaventure have proposed multipath extensions for the QUIC protocol in the corresponding IETF WG.

References

- RFC 6824, *TCP Extensions for Multipath Operation with Multiple Addresses*, A. Ford, C. Raiciu, M. Handley, O. Bonaventure, January 2013
- draft-ietf-mptcp-rfc6824bis-09, *TCP Extensions for Multipath Operation with Multiple Addresses*, Alan Ford, Costin Raiciu, Mark Handley, Olivier Bonaventure, Christoph Paasch, July 2017
- RFC 6182, *Architectural Guidelines for Multipath TCP Development*, A. Ford, C. Raiciu, M. Handley, S. Barre, J. Iyengar, March 2011
- Multipath TCP Website, last accessed September 2015
- Multipath TCP Blog, last accessed September 2015
- draft-ietf-mptcp-attacks-04, *Analysis of MPTCP residual threats and possible fixes*, Marcelo Bagnulo, Christoph Paasch, Fernando Gont, Olivier Bonaventure, Costin Raiciu, March 2015
- draft-ietf-mptcp-experience-07, *Use Cases and Operational Experience with Multipath TCP*, Olivier Bonaventure, Christoph Paasch, Gregory Detal, October 2016
- draft-dreibholz-mptcp-nornet-experience-03, *MPTCP Experiences in the NorNet Testbed*, Thomas Dreibholz, Simone Ferlin, Ozgu Alay, Ahmed Elmokashfi, Ioana Alexandrina Livadariu, Xing Zhou, March 2017
- draft-hesmans-mptcp-socket-02, *A socket API to control Multipath TCP*, Benjamin Hesmans, Olivier Bonaventure, July 2017
- draft-zuo-banana-mptcp-integration-00, *Integration of Bonding Tunnels and MPTCP*, Nicolai Leymann, Cornelius Heidemann, Geng Liang, Jing Zuo, Lianshu Zheng, June 2017
- draft-xu-mptcp-prmp-04, *Multipath Transmission Control Protocol (MPTCP) Partial Reliability Extension*, Changqiao Xu, Hui Huang, Hongke Zhang, Chunshan Xiong, Lei Zhu, June 2017
- draft-song-mptcp-owl-02, *One Way Latency Considerations for MPTCP*, Fei Song, Hongke Zhang, H Anthony Chan, Anni Wei, March 2017
- draft-bonaventure-mptcp-converters-01, *0-RTT TCP converters*, Olivier Bonaventure, Mohamed Boucadair, Bart Peirens, July 2017
- draft-duchene-mptcp-load-balancing-01, *Multipath TCP Load Balancing*, Fabien Duchene, Vladimir Olteanu, Olivier Bonaventure, Costin Raiciu, Alan Ford, July 2017
- draft-xu-mptcp-momp-04, *A Message-Oriented Extension to Multipath Transmission Control Protocol (MPTCP)*, Changqiao Xu, Jiuren Qin, Hongke Zhang, Chunshan Xiong, Lei Zhu, June 2017
- draft-boucadair-mptcp-dhc-07, *DHCP Options for Network-Assisted Multipath TCP (MPTCP)*, Mohamed Boucadair, Christian Jacquenet, Tirumaleswar Reddy, April 2017
- draft-boucadair-mptcp-radius-04, *RADIUS Extensions for Network-Assisted Multipath TCP (MPTCP)*, Mohamed Boucadair, Christian Jacquenet, April 2017
- draft-samar-mptcp-socketapi-02, *Extended Socket APIs to control subflow priority in Multipath TCP*, Samar Shailendra, Hemant Kumar Rath, Arpan Pal, Abhijit Mondal, January 2019

– AlessandraScicchitano - 2013-08-14
– SimonLeinen - 2014-12-29 - 2019-09-10