Making Multipath TCP friendlier to Load Balancers and Anycast

Fabien Duchêne <fabien.duchene@uclouvain.be>
Olivier Bonaventure <olivier.bonaventure@uclouvain.be>
Université Catholique de Louvain

ICNP 2017 - Toronto - October 2017
Without Multipath TCP

Address 1

Address 2

WiFi

4G/LTE
Multipath TCP: one pipe to rule them all
Multipath TCP in a nutshell
Multipath TCP in a nutshell

A1

SYN+MP_CAPABLE

SYN+ACK+MP_CAPABLE

A2
Multipath TCP in a nutshell

A1

SYN+MP_CAPABLE

SYN+ACK+MP_CAPABLE

SYN+MP_JOIN

A2

6
Multipath TCP in a nutshell
TCP and Load Balancers
TCP and Load Balancers

A1

SYN

HASH(src_ip,dst_ip,src_port,dst_port,TCP) = 10.0.0.1
TCP and Load Balancers

HASH(src_ip,dst_ip,src_port,dst_port,TCP) = 10.0.0.1
TCP and Load Balancers

A1

HASH(src_ip,dst_ip,src_port,dst_port,TCP) = 10.0.0.1
TCP and Load Balancers: **Direct Server Return**
The problem: Multipath TCP and Load Balancers

HASH(A1:src_ip,dst_ip,A1:src_port,dst_port,TCP) = 10.0.0.1
The problem: Multipath TCP and Load Balancers
The problem: Multipath TCP and Load Balancers

\[
\text{HASH}(A2:src\_ip, dst\_ip, A2:src\_port, dst\_port, TCP) = 10.0.0.2
\]
The problem: Multipath TCP and Load Balancers

\[
\text{HASH}(A2:\text{src}_i\text{p}, \text{dst}_i\text{p}, A2:\text{src}_p\text{ort}, \text{dst}_p\text{ort}, \text{TCP}) = 10.0.0.2
\]
Existing solutions

- This problem has been slowing down the deployment of Multipath TCP
- Several solutions [1,2] already proposed but these solutions requires
  - Stateful load balancing [1]
  - Modifications of the load balancers [1,2]

- Multipath TCP was designed to work in the “current” Internet

Our solution: objectives

- Allow Multipath TCP to work with unmodified load balancers
- Minimize the importance of the load balancer

A solution that scales
Our solution
Our solution

\[
\text{HASH}(\text{src_ip}, \text{dst_ip}, \text{src_port}, \text{dst_port}, \text{TCP}) = 10.0.0.1
\]
Our solution
Our solution

SYN+MP_JOIN

SYN+MP_CAPABLE

SYN+ACK+MP_CAPABLE

1.2.3.4

10.0.0.1

5.6.7.8
Our solution

SYN+MP_JOIN

1.2.3.4

10.0.0.1

5.6.7.8
Our solution: advertising the address

SYN+MP_CAPABLE

SYN+ACK+MP_CAPABLE

ADD_ADDR(5.6.7.8)

10.0.0.1

5.6.7.8
Our solution: advertising the address

- SYN+MP_CAPABLE
- SYN+ACK+MP_CAPABLE
- ADD_ADDR(5.6.7.8)
- 10.0.0.1
- 5.6.7.8
- SYN+MP_JOIN
Our solution: advertising the address

SYN+MP_JOIN

SYN+MP_CAPABLE

SYN+ACK+MP_CAPABLE

ADD_ADDR(5.6.7.8)

BACKUP_MODE

10.0.0.1

5.6.7.8
Our solution: beyond **Direct Server Return**
Implementation

- Implementation done in the Linux Kernel
- 3 parts:
  - Avoiding MP_JOIN on load balanced addresses
  - Address advertisement reliability
  - Load balancing path manager

- +- 600 lines of code
Application: Layer-4 load balancer

100 Mbps link

1 Gbps link
Layer-4 load balancer: results

Transfer rates with 0.0% loss and 0.0ms delay

- Regular TCP
- Multipath TCP

942Mbits/s
~90Mbits/s
Layer-4 load balancer: results

- MPTCP isn’t significantly affected by the loss
- The latency affects only the connection establishment
- Transfer rates (10MB):
  - TCP : 16Mbits/sec
  - MPTCP : 803Mbits/sec
Application: Anycast

Anycast addr.
1.2.3.4
Application: Anycast

Anycast addr.
1.2.3.4
Application: Anycast

Anycast addr.
1.2.3.4
Application: Anycast

- 1Gbps link
- 10 Gbps link

Anycast addr. 1.2.3.4

1.2.3.4
5.6.7.0/24

1.2.3.4
8.9.10.0/24

1.2.3.4
11.12.13.0/24
Anycast: results

- ECMP pool of 3 servers
- Every 10 sec.: remove a server for 5 sec.
- 3 servers: 2800Mbit/s
- 2 servers: 1900Mit/s
- Spikes in RST when a server is removed and when it’s re-added
Anycast: results

- No drop in Bandwidth
- No RST sent

Multipath TCP can be deployed to support anycast services.
Conclusion

- Multipath TCP is now on millions of iPhones
- Multipath TCP’s deployment has been hindered on servers

- Our solution
  - Works with unmodified load balancers
  - Puts the load balancer off-path
  - Enables the use of Anycast addresses
Thank you!

Fabien Duchêne <fabien.duchene@uclouvain.be>
Backup slide : Security

- Concerns about exposing the server directly
- Possible solutions :
  - A firewall (on the server or in the network)
  - Generate a specific IPv6 address per client
  - Block everything but MP_JOINS on public address
Our solution: beyond Direct Server Return

NAT

Direct Server Return

Multipath TCP
A fair comparison? Fair enough!