Multipath TCP Address Advertisement

Follow-up

Fabien Duchêne <fabien.duchene@uclouvain.be>
Olivier Bonaventure <olivier.bonaventure@uclouvain.be>

Integrated into RFC6824bis

- "NO JOIN" flag in MP_CAPABLE : "do not connect back to this address"
- "Echo" flag in ADD_ADDR : making ADD_ADDR reliable

Implementation status

- The "Echo" flag in ADD_ADDR is implemented
  - "Agressive" retransmission
  - Still need some tuning
  - Pull request coming soon

- The "NO JOIN" is still under development

Make before or after break

Adding a "Backup" bit in the ADD_ADDR
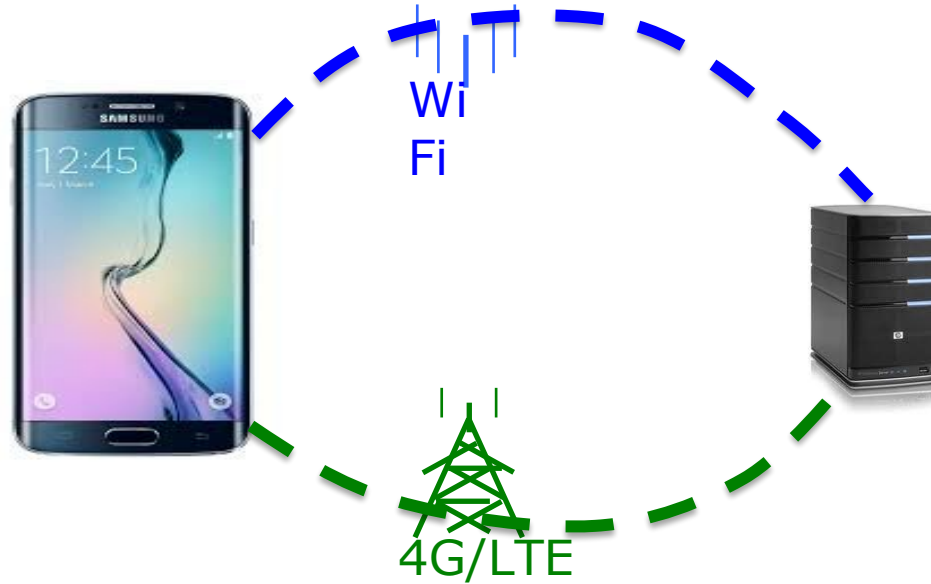
Motivations for make after break

      Reducing the number of subflows

      Energy utilisation and radio resources on smartphones
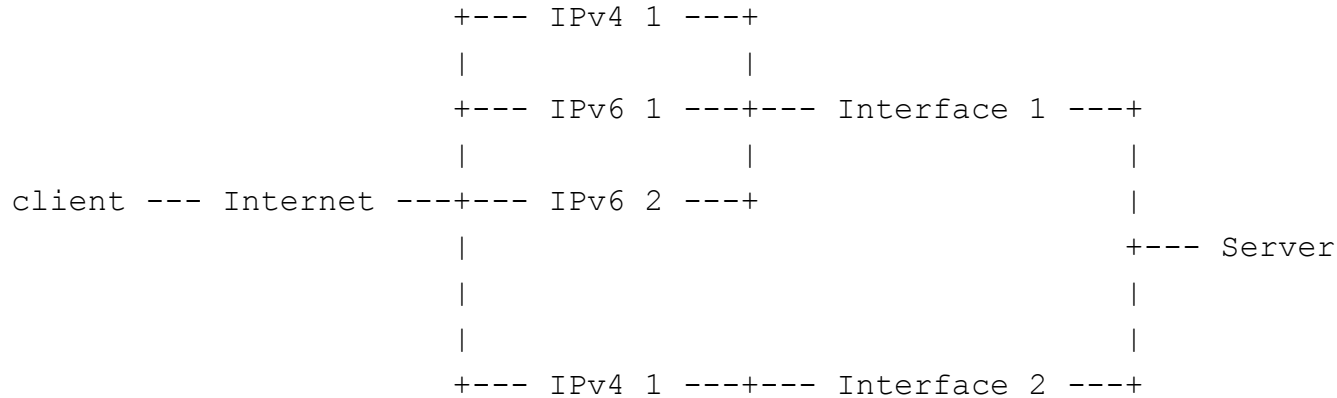
Received support from Christoph Paasch.

Priorities

The scheduler could schedule according to specific priorities.

Path Diversity (Communities)

An interface can have multiple addresses (dual-stack, multiple interfaces,...)

```
                        +--- IPv4 1 ---+
                        |              |
                        +--- IPv6 1 ---+--- Interface 1 ---+
                        |              |                   |
   client --- Internet ---+--- IPv6 2 ---+                   |
                        |                        +--- Server
                        |                                   |
                        |                                   |
                        +--- IPv4 1 ---+--- Interface 2 ---+
```

Multipath TCP Load Balancing

Fabien Duchêne <fabien.duchene@uclouvain.be>
Vladimir Olteanu <vladimir.olteanu@cs.pub.ro>
Olivier Bonaventure <olivier.bonaventure@uclouvain.be>
Costin Raicu <costin.raiciu@cs.pub.ro>

General idea

A guide to help implementers/network admins

Describe the existing solutions

Compare the existing solutions

    eg: stack modification required, public address required,....

Per-server addresses

Use two types of addresses:

- One address behind the load balancer (announced in the DNS)
- One/Several addresses directly connected (not announced)

The first address is used to initiate the first subflow, the others are used for the subsequent subflows

Uses **draft-duchene-mptcp-add-addr-00**

Embedding Extra Information in Packets

Getting the client to embed connection or server-identifying information in the packets.

A stateless load-balancer could use this information.

2 proposals:

- embed server ID in secondary subflows' server port
- embed server token in TCP timestamp

Needs a modification of the load balancers and the server's stack.

Next steps

- New solution: Application Layer Authentication
- Security considerations

- More input from the industry?