Tuning Multipath TCP for Interactive Applications on Smartphones

Quentin De Coninck, Olivier Bonaventure

May 16, 2018

Université catholique de Louvain

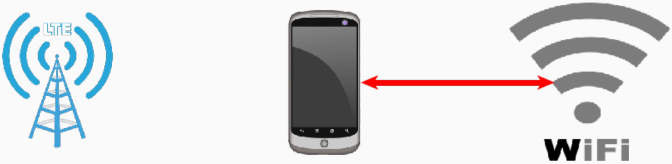
IFIP Networking 2018

UCL
Université catholique de Louvain

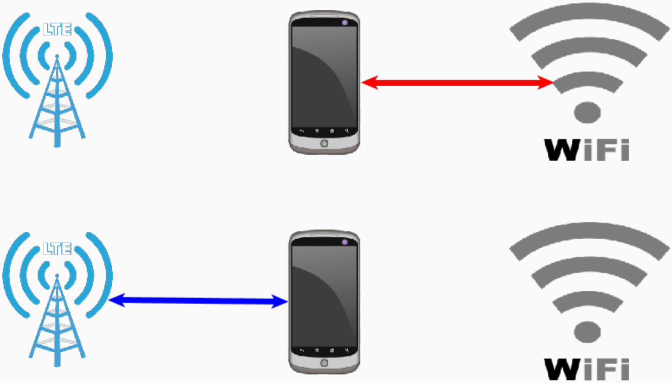
fnrs
LA LIBERTÉ DE CHERCHER

Smartphones are Multi-Homed Devices

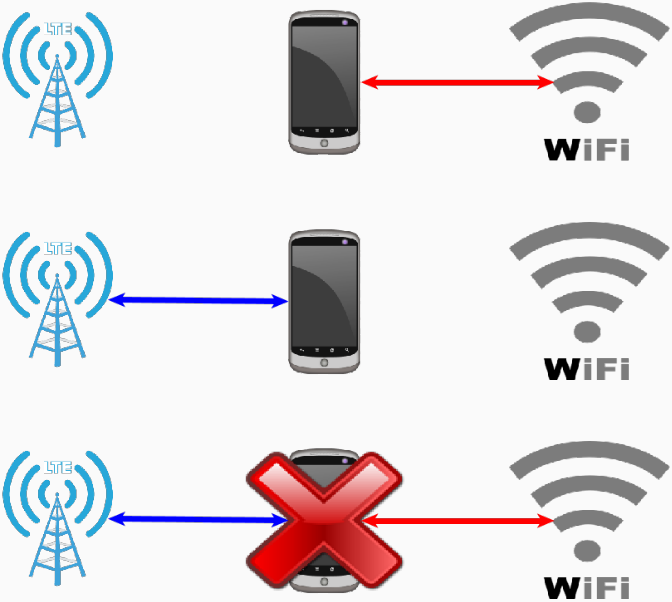
What is wrong today with TCP?



What is wrong today with TCP?



What is wrong today with TCP?

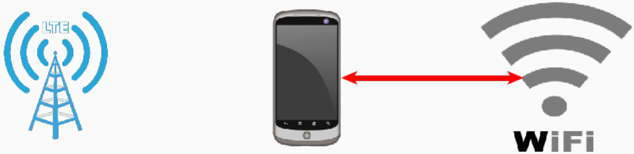


Multipath TCP on Smartphones

- Multipath TCP = TCP connections using several paths
    - Bandwidth aggregation, network resiliency

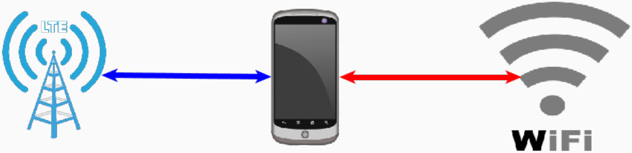
Multipath TCP on Smartphones

- Multipath TCP = TCP connections using several paths
  - Bandwidth aggregation, network resiliency
- Start Multipath TCP connection on `default` interface



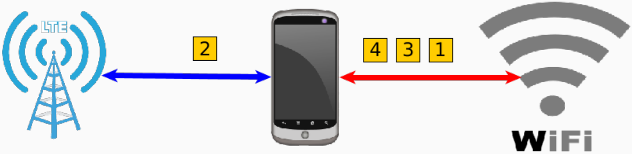
Multipath TCP on Smartphones

- Multipath TCP = TCP connections using several paths
  - Bandwidth aggregation, network resiliency
- Start Multipath TCP connection on `default` interface
- Create additional paths



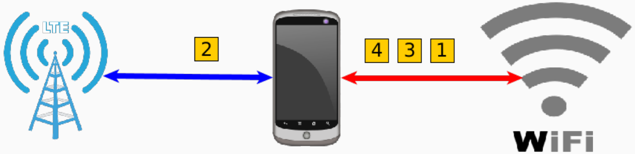
Multipath TCP on Smartphones

- Multipath TCP = TCP connections using several paths
  - Bandwidth aggregation, network resiliency
- Start Multipath TCP connection on `default` interface
- Create additional paths
- Spread packets over available paths
  - Selection done by the packet sender



Multipath TCP on Smartphones

- Multipath TCP = TCP connections using several paths
    - Bandwidth aggregation, network resiliency
- Start Multipath TCP connection on `default` interface
- Create additional paths
- Spread packets over available paths
    - Selection done by the packet sender



Siri, iOS11

SOCKS Proxy

Imperfections with Multipath TCP on Smartphones

MultiMob

Evaluation with Real Users

Conclusion

Imperfections with Multipath TCP on Smartphones

Smartphone Traffic $\neq$ Bulk Transfers

Linux implementation of Multipath TCP

- Mainly bandwidth aggregation driven
- But most of the connections are (very) short in bytes
  - Most are **latency-sensitive**
  - Growing importance of interactive applications
    - Apple Siri, Google Now, Amazon Alexa,...
  - Multipath = **network resiliency under mobility**

Multipath TCP Cellular Subflow Usage

"Cost less than"

Multipath TCP Cellular Subflow Usage

"Cost less than"

Multipath TCP Cellular Subflow Usage

$
WiFi
<
"Cost less than"
$
LTE

Multipath TCP Cellular Subflow Usage

$
WiFi
<
"Cost less than"
$
LTE

Smartphone = data consumers

- Servers take most of the scheduling decisions
    - Prefer **lowest perceived latency** path by default
- Ability to define cellular as **backup** path
    - Use non-backup paths unless they all failed

Multipath TCP Cellular Subflow Usage

WiFi & cellular path creations as soon as possible

- *make-before-break* for fast network handover
- But most cellular subflows do not see **any** data
  - WiFi is often sufficient
    - Short connections, low application push rate, no mobility...
  - **Network + energy waste**
    - Cellular can consume as much as a screen 100% on

MultiMob

Imperfections with Multipath TCP on Smartphones

MultiMob

- Giving Control to the Client
- Break-Before-Make
- Immediate Reinjections

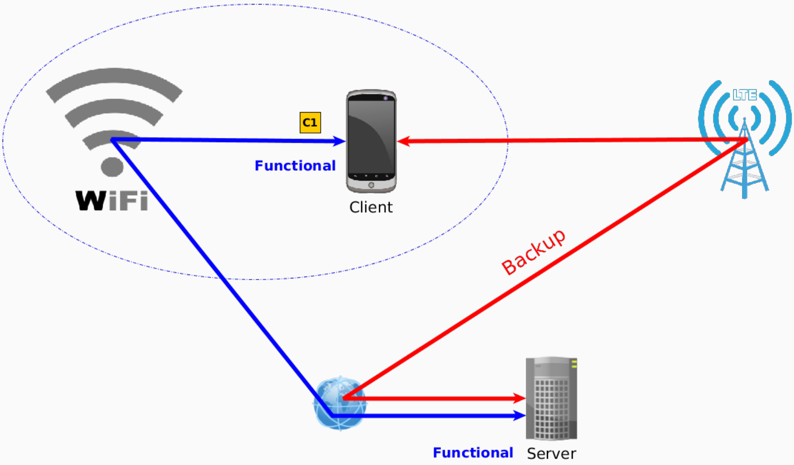
Evaluation with Real Users

Conclusion

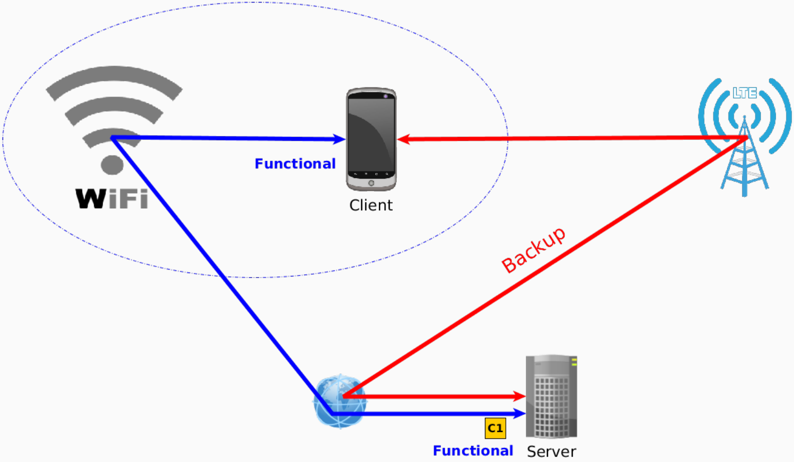
MultiMob

Giving Control to the Client

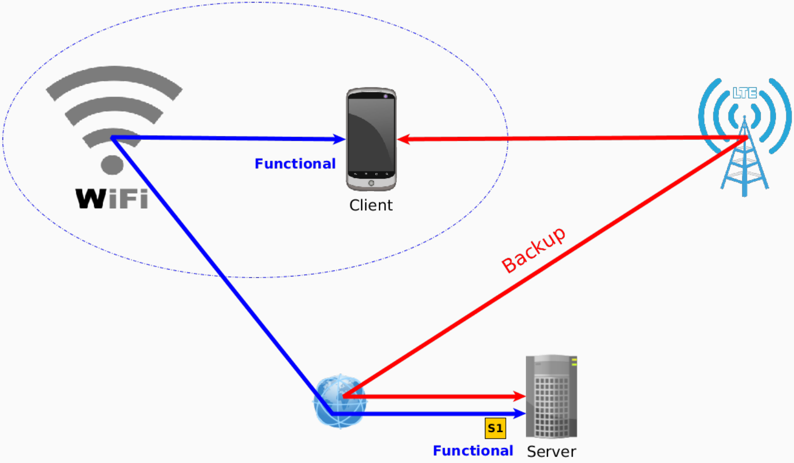
An Illustrative Example - Request/Response Traffic



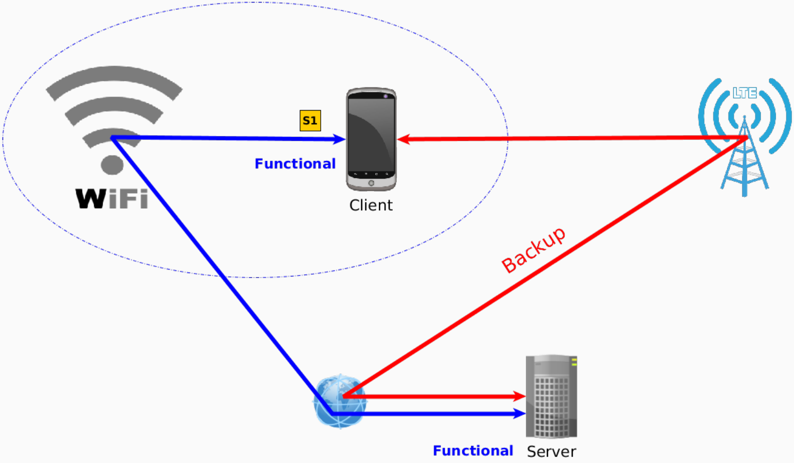
An Illustrative Example - Request/Response Traffic



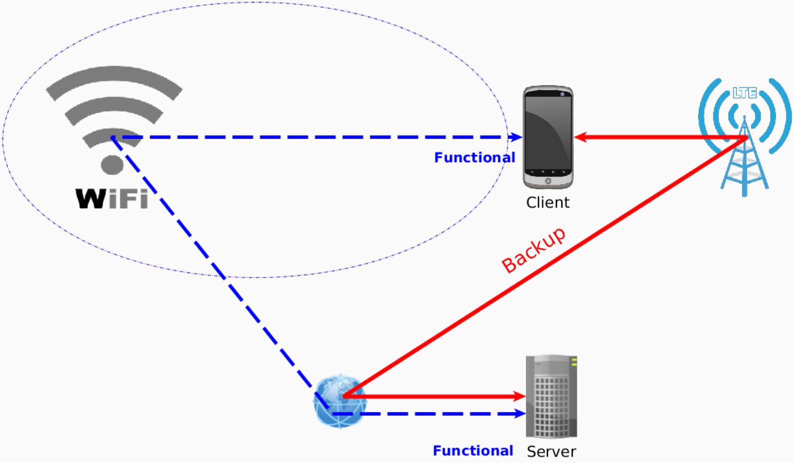
An Illustrative Example - Request/Response Traffic



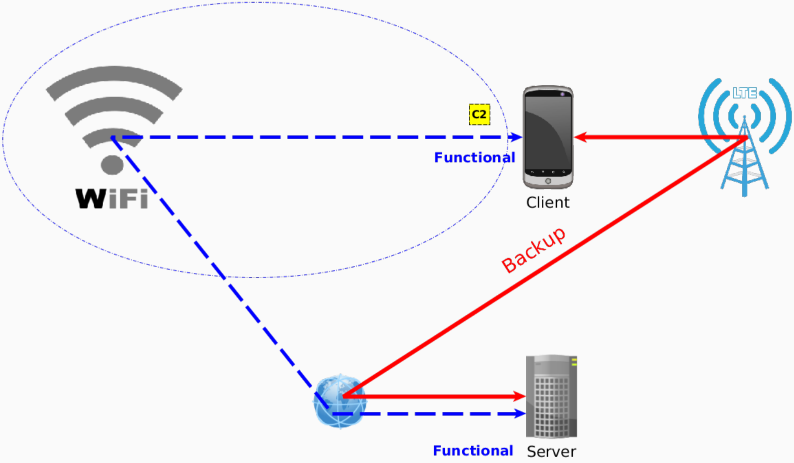
An Illustrative Example - Request/Response Traffic



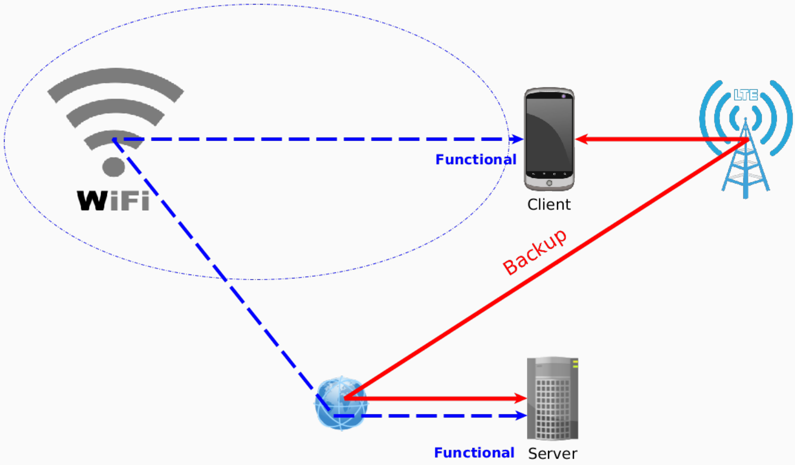
An Illustrative Example - Request/Response Traffic



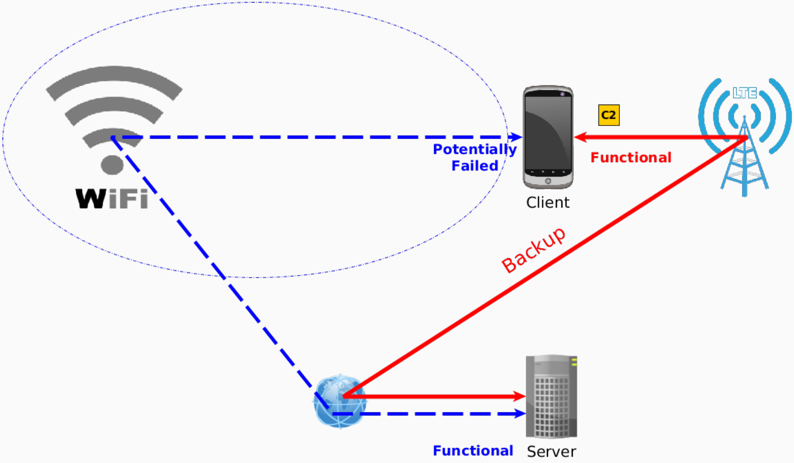
An Illustrative Example - Request/Response Traffic



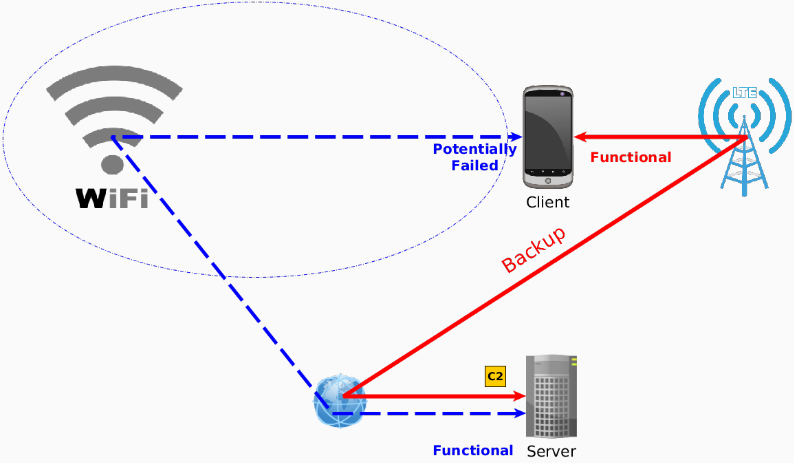
An Illustrative Example - Request/Response Traffic



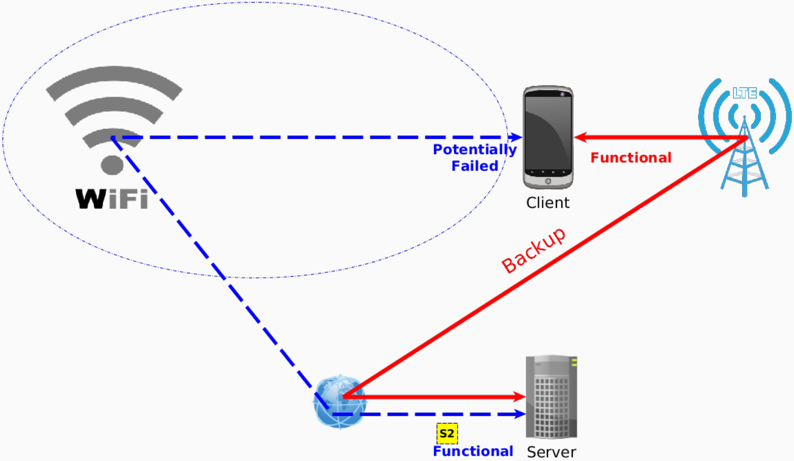
An Illustrative Example - Request/Response Traffic



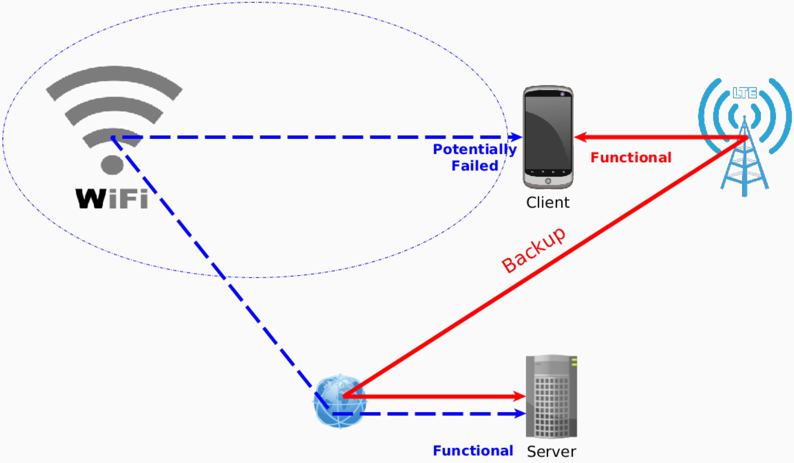
An Illustrative Example - Request/Response Traffic



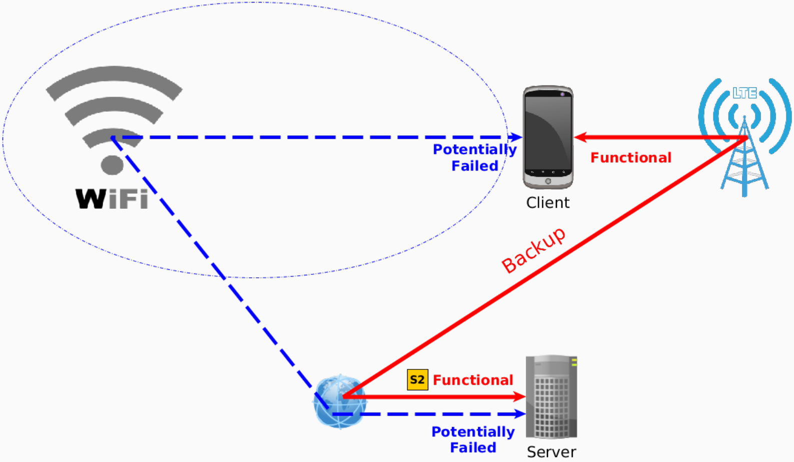
An Illustrative Example - Request/Response Traffic



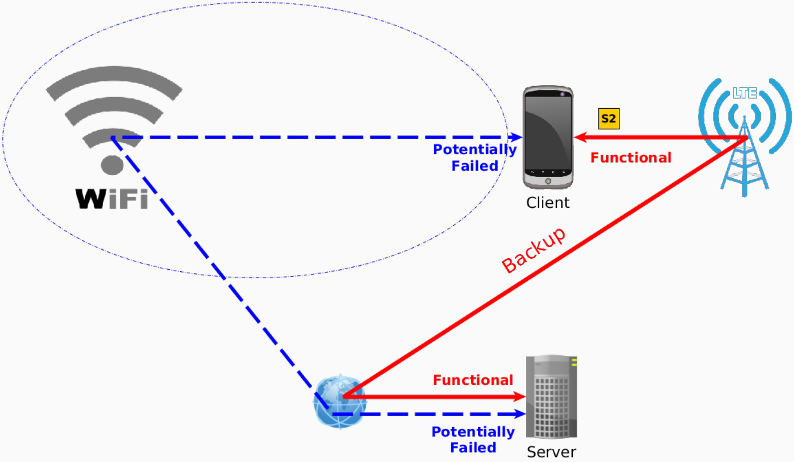
An Illustrative Example - Request/Response Traffic



An Illustrative Example - Request/Response Traffic



An Illustrative Example - Request/Response Traffic

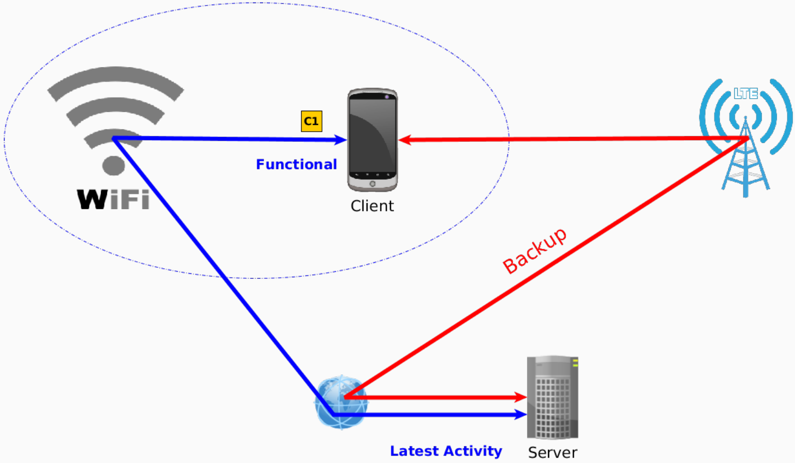


MultiMob Server Scheduler

→ Let the server follow the path used by client

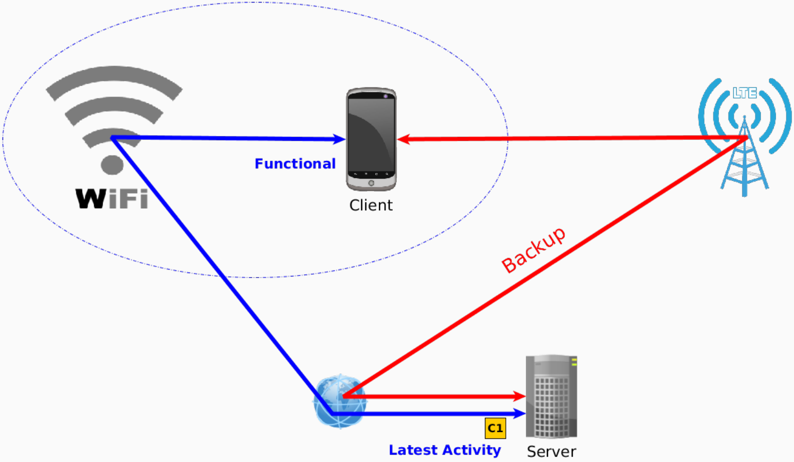
MultiMob Server Scheduler

→ Let the server follow the path used by client



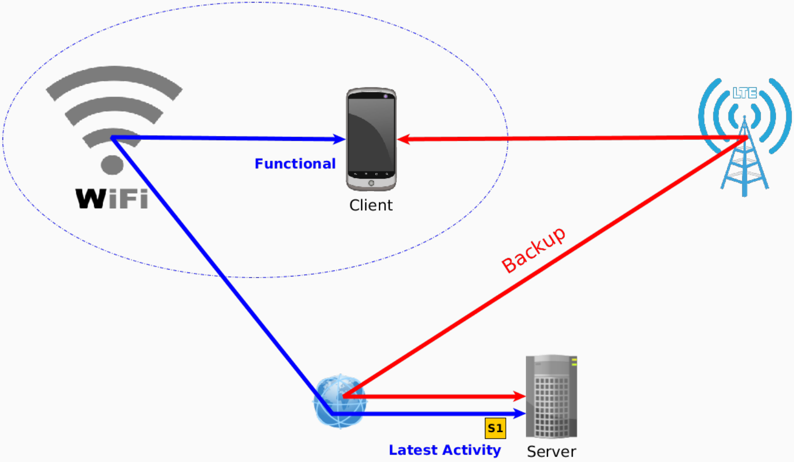
MultiMob Server Scheduler

→ Let the server follow the path used by client



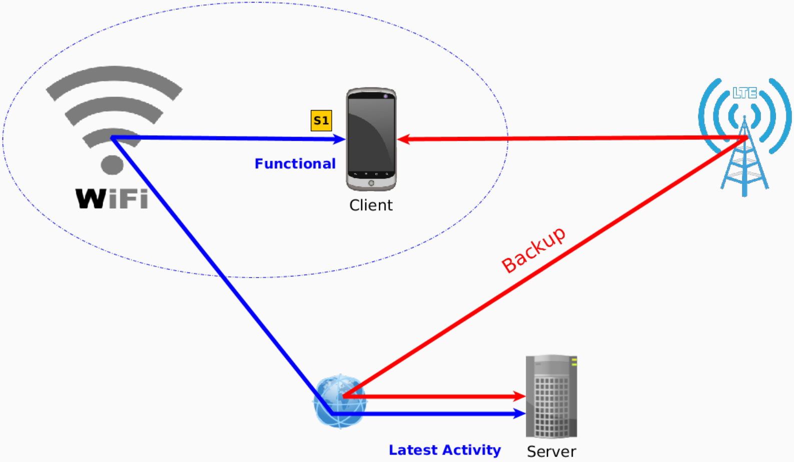
MultiMob Server Scheduler

→ Let the server follow the path used by client



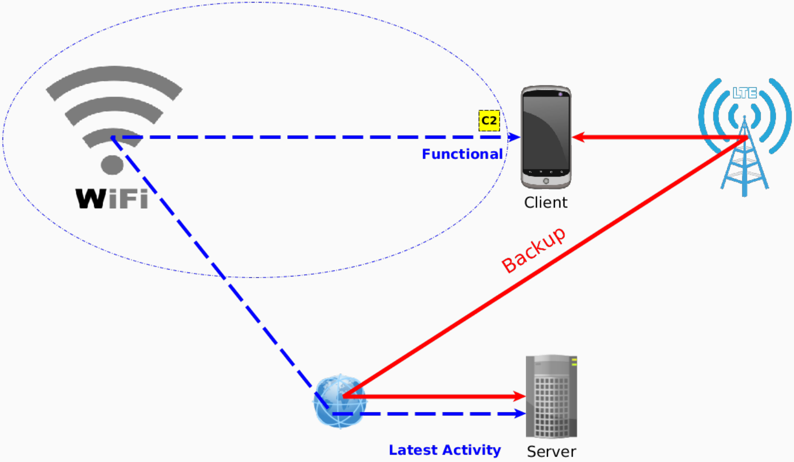
MultiMob Server Scheduler

→ Let the server follow the path used by client



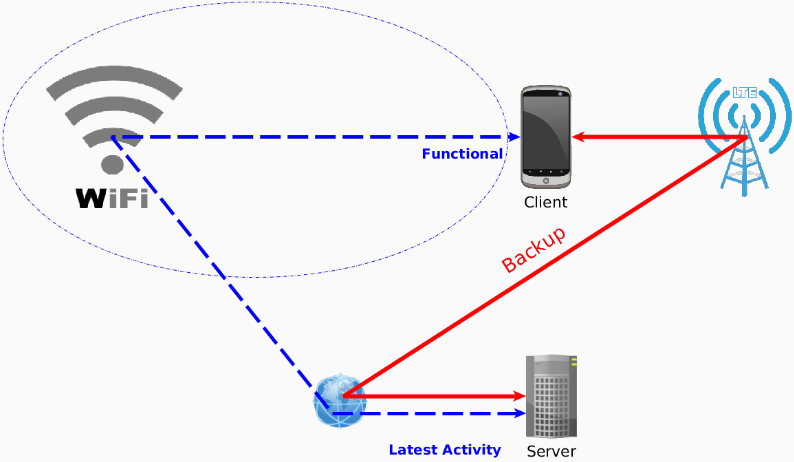
MultiMob Server Scheduler

→ Let the server follow the path used by client



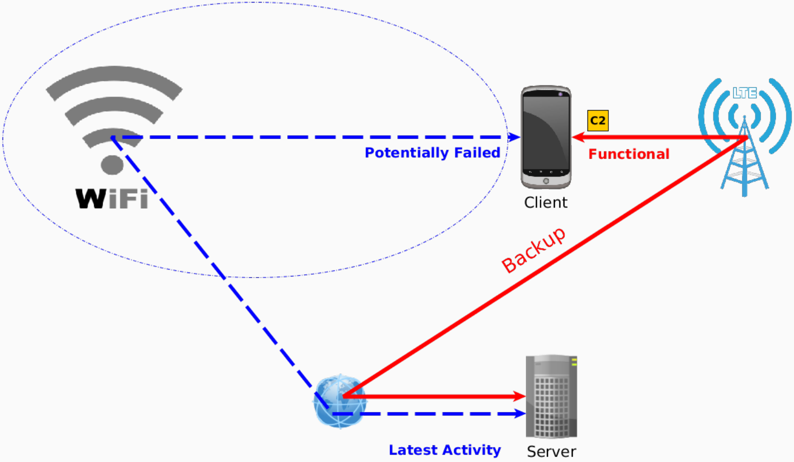
MultiMob Server Scheduler

→ Let the server follow the path used by client



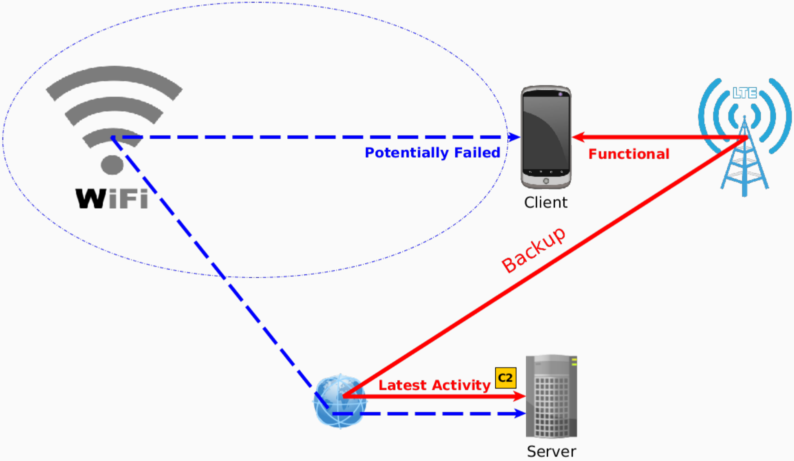
MultiMob Server Scheduler

→ Let the server follow the path used by client



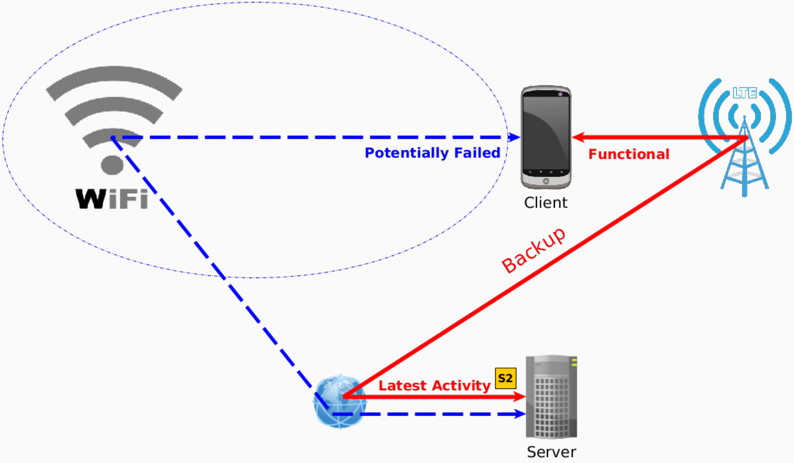
MultiMob Server Scheduler

→ Let the server follow the path used by client



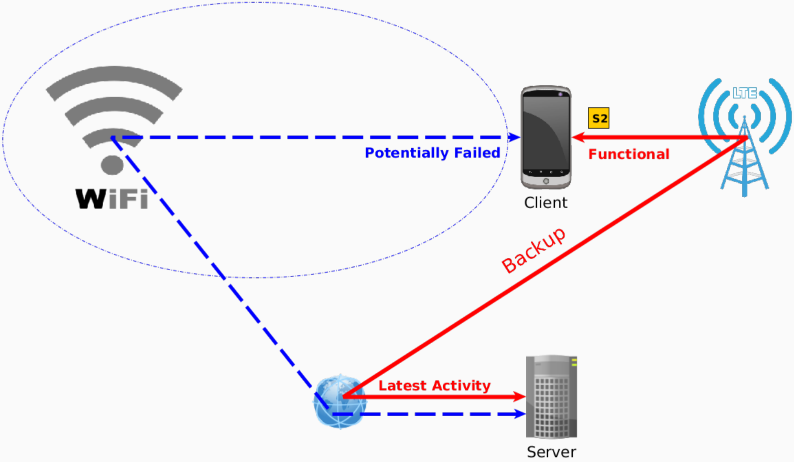
MultiMob Server Scheduler

→ Let the server follow the path used by client



MultiMob Server Scheduler

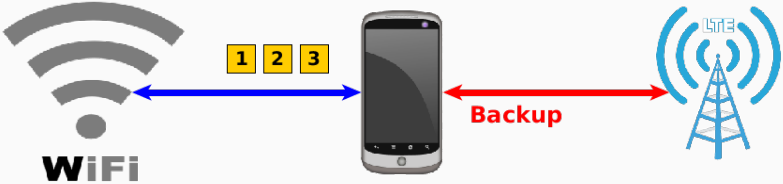
→ Let the server follow the path used by client



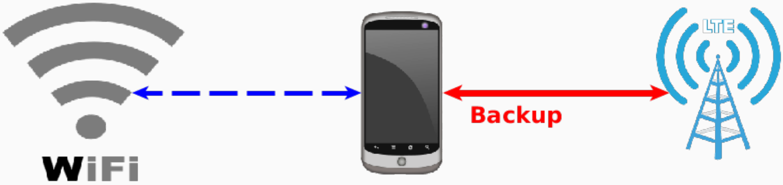
MultiMob

Break-Before-Make

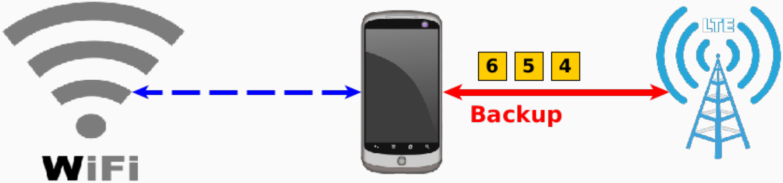
Current Make-Before-Break Approach



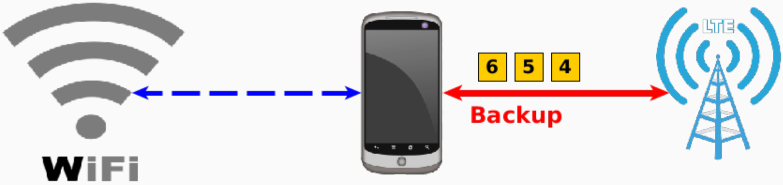
Current Make-Before-Break Approach



Current Make-Before-Break Approach



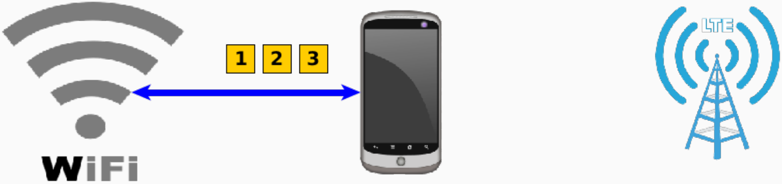
Current Make-Before-Break Approach



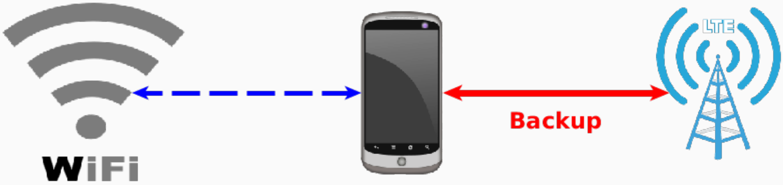
→ Minimizes amount of data sent over cellular

- Yet lot of energy wasted by LTE usage...

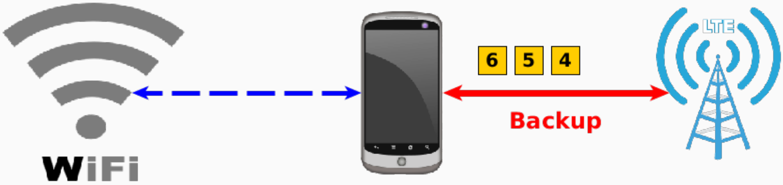
Proposed Break-Before-Make Approach



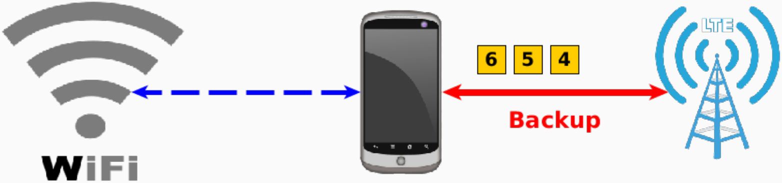
Proposed Break-Before-Make Approach



Proposed Break-Before-Make Approach



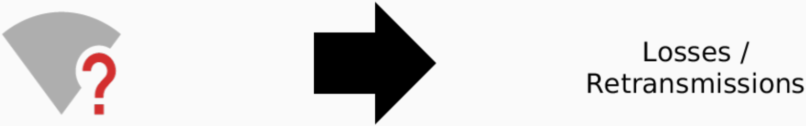
Proposed Break-Before-Make Approach



→ No LTE usage if cellular path not needed!

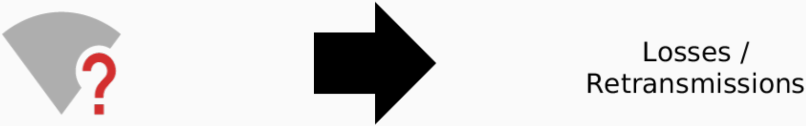
- But need to quickly detect bad wireless networks

Quick Failure Detection



Losses /
Retransmissions

Quick Failure Detection



Losses /
Retransmissions

Proposing an **in-kernel Multipath TCP oracle**

- Periodically compute statistics about *netpaths*
  - Netpath = (IP_{src},IP_{dst}, Net interf.)
- Trigger backup creation upon excessive losses/retransmissions
  - + prevents primary use when bad

IP_{src}	IP_{dst}	Net interf.	TCP sfs	TCP stats
1.2.3.4	4.5.6.7	WiFi	[tp_1, tp_3]	sloss 2%, ...
2.3.4.5	5.6.7.8	Cellular	[tp_2]	sloss 0%, ...
2.3.4.5	4.5.6.7	Cellular	[tp_4, tp_5]	sloss 15%, ...

Table 1: Oracle monitoring table example.

MultiMob

Immediate Reinjections

Limiting Handover Delay

The backup cellular path creation is delayed

- Nice from a energy consumption point of view...
- ...but incurs larger app perceived latency in mobility cases

Furthermore, additional Multipath TCP path creation takes time...

Improving Establishment of Additional Subflows

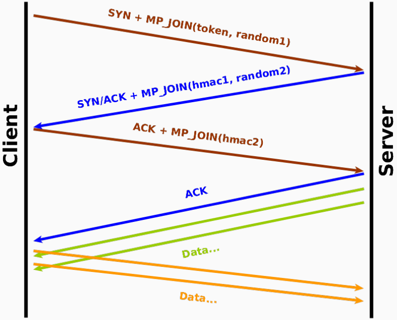

Client
Server
SYN + MP_JOIN(token, random1)
SYN/ACK + MP_JOIN(hmac1, random2)
ACK + MP_JOIN(hmac2)
ACK
Data...
Data...

Figure 1: Normal `JOIN`.

Improving Establishment of Additional Subflows

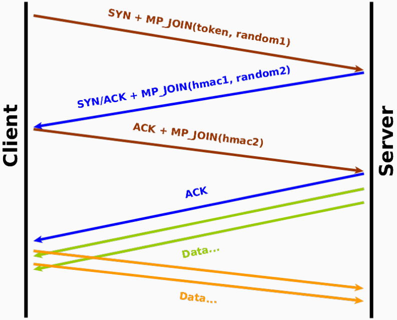

Client
Server
SYN + MP_JOIN(token, random1)
SYN/ACK + MP_JOIN(hmac1, random2)
ACK + MP_JOIN(hmac2)
ACK
Data...
Data...

Figure 1: Normal JOIN.

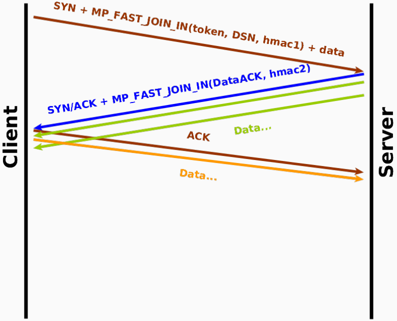

Client
Server
SYN + MP_FAST_JOIN_IN(token, DSN, hmac1) + data
SYN/ACK + MP_FAST_JOIN_IN(DataACK, hmac2)
Data...
ACK
Data...

Figure 2: Fast JOIN with data.

Evaluation with Real Users

Methodology

Half-dozen of smartphone users

- Running Android 6.0.1
- Two sets of users
  - Running vanilla Multipath TCP (with/without backup)
  - Running MultiMob

Performing interactive traffic measurements

- Light bursty request/response traffic, see paper for details
  - Observe delay between request sent and response received
- Running 80 s once motion is detected
- Only consider tests with both WiFi/LTE online at beginning

Interactive Traffic - Maximal Request Delay over Tests

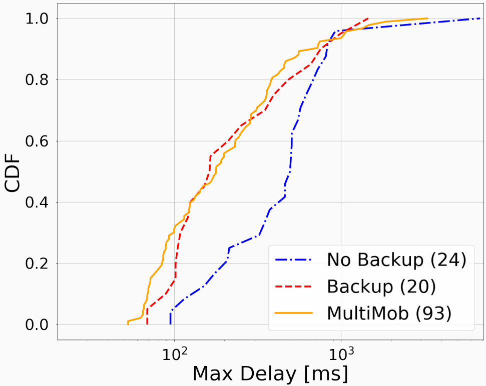


Figure 3: Maximal delays.

- MultiMob does not perform worse than Vanilla Multipath TCP
- Using multiple paths is not always beneficial

Interactive Traffic - Energy Consumption ("static" tests)

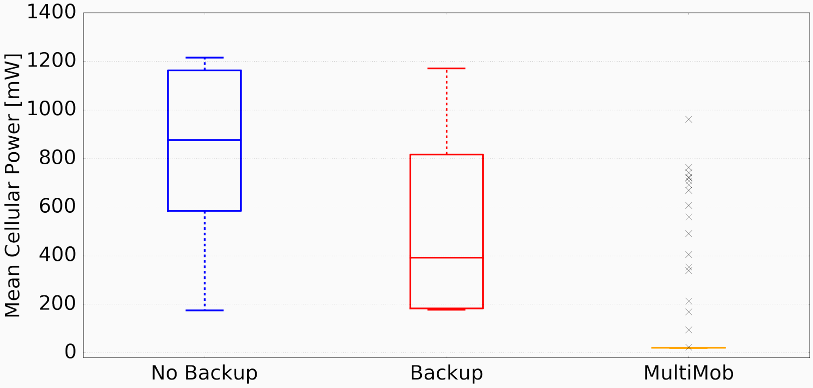


Figure 4: Estimated mean cellular power, WiFi not lost.

- **MultiMob consumes much less cellular energy!**

Conclusion

Conclusion

- Vanilla Multipath TCP not efficient from energy viewpoint
- MultiMob: Multipath TCP tuned for smartphones
    - Keep similar performances
    - Lower energy consumption
    - Lower LTE radio resource usage
- MultiMob is available

`http://multipath-tcp.org/multimob`

Thanks for your attention!

Feel free to ask questions!

`http://multipath-tcp.org/multimob`

Coping with Corner Cases - "Single" Bulk Download

- Client detects bad networks with lost sent packets
- Client creates backup paths once bad network detected
- One connection experiences issues, others react

Coping with Corner Cases - "Single" Bulk Download

- Client detects bad networks with lost sent packets
- Client creates backup paths once bad network detected
- One connection experiences issues, others react

→ What about single bulk download without background traffic?

- By default, server does not create paths
    - NAT, firewalls,...

Receive Timer

Exchange additional information at connection level

- Send RTO_{sender} to peer
- `MP_IDLE` bit in DSS to indicate no more data to send (now)

Allows to setup a **receive timer**

- Reset to RTO_{sender} if activity without `MP_IDLE`
- Stop if received data with `MP_IDLE`

If receive timer fires, create backup path

Assessing Receive Timer

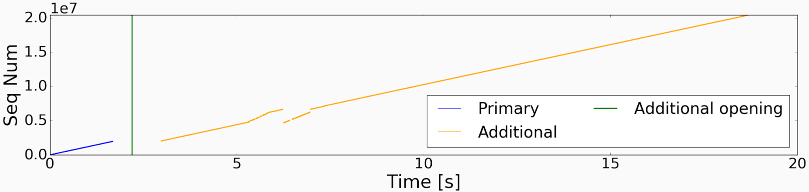


Figure 5: Time-sequence graph of server to client flow from client side perspective for a HTTP GET of 20 MB. Primary has 100% losses at 1.5 s.

- Client opens additional path after inactivity period
- Once established and data detected as lost, server continues the connection on the additional path