A proposal for MPTCP Robust session Establishment (MPTCP RobE)

enable full multipath capability for MPTCP

Markus Amend, Eckard Bogenfeld, Andreas Matz

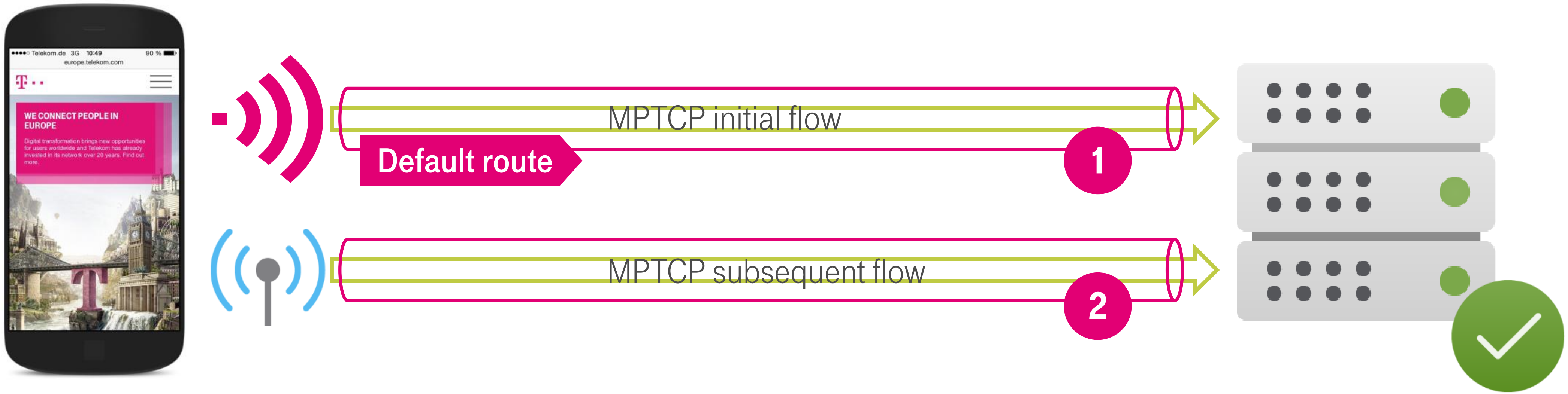
18 July 2017

LIFE IS FOR SHARING.

Agenda

01	The role of the initial flow in MPTCP
02	MPTCP RobE idea
03	MPTCP RobE proposals & criteria
04	Experimental investigation pref. approach
05	Results and Discussion
06	Conclusion and Future work

THE ROLE OF THE INITIAL FLOW IN MPTCP
MULTIPATH TCP CONNECTION ESTABLISHMENT (RFC6824)

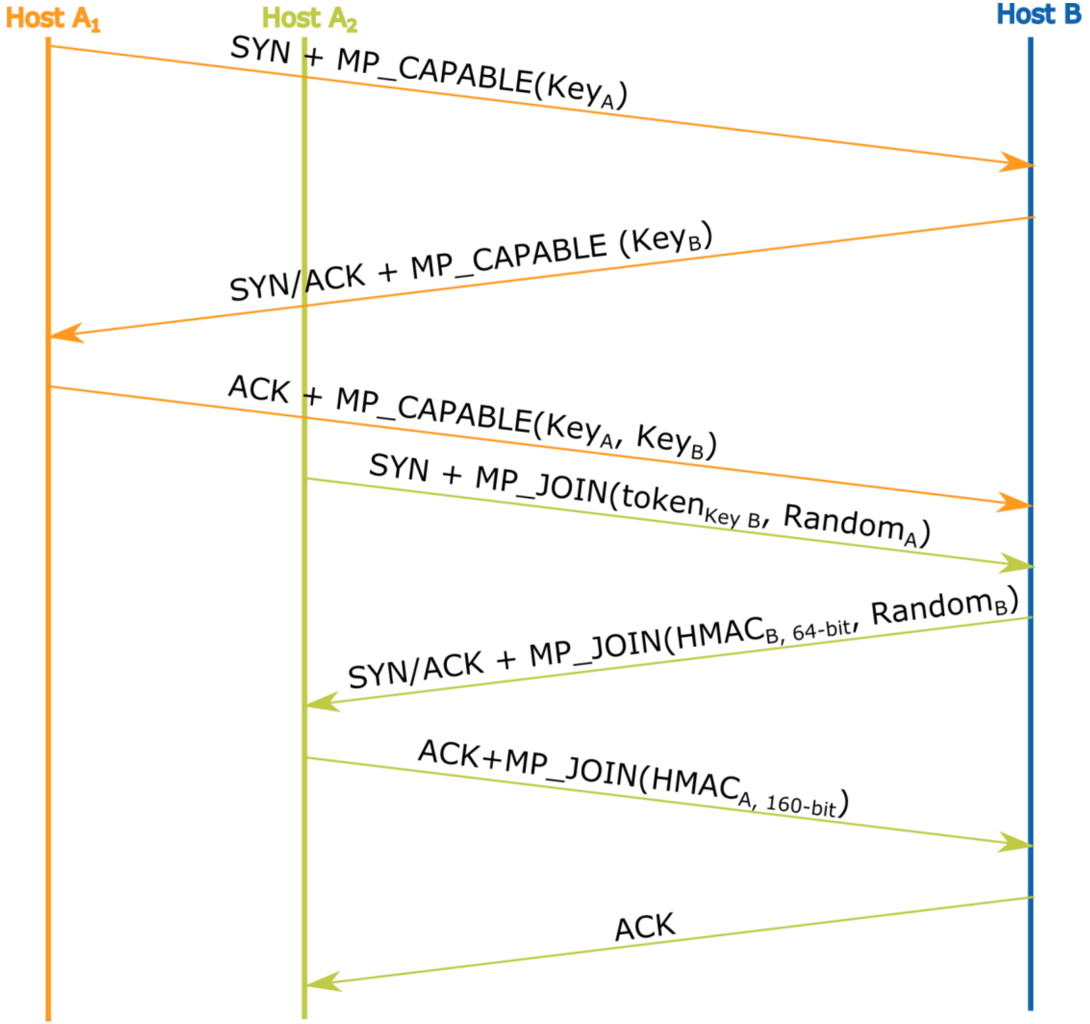


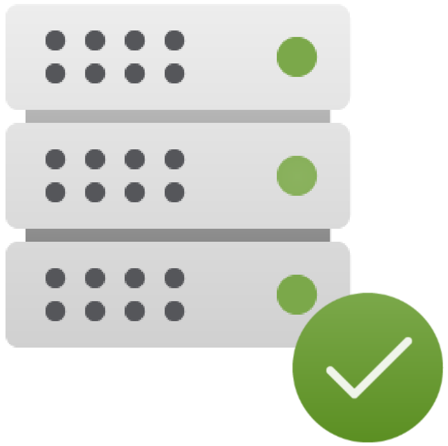
With 2 → Resilience and possibly bandwidth aggregation

Multipath benefit is first enabled now

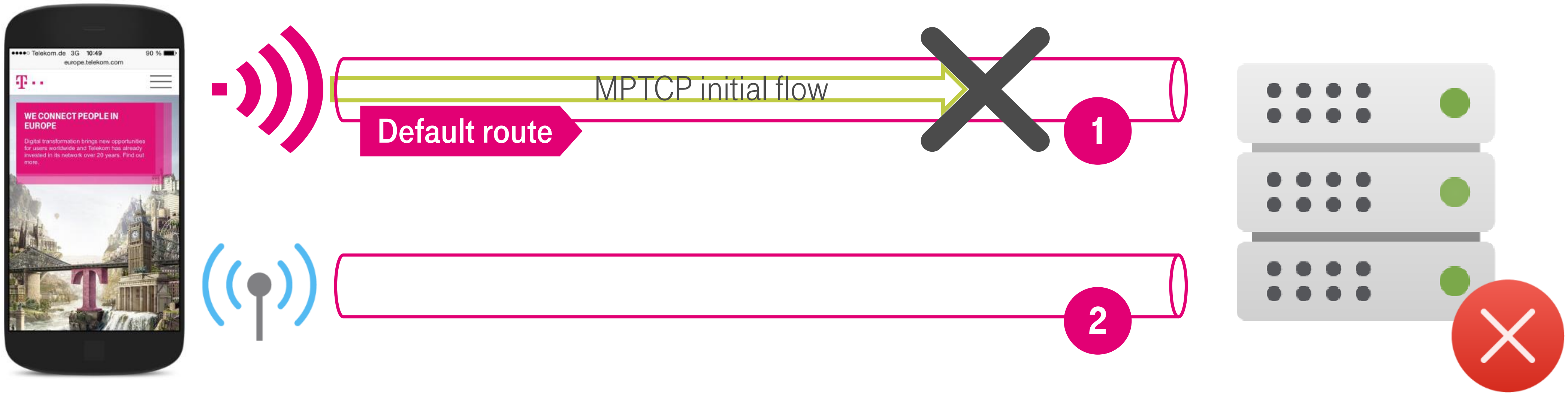
THE ROLE OF THE INITIAL FLOW IN MPTCP

MULTIPATH TCP CONNECTION ESTABLISHMENT (RFC6824)





THE ROLE OF THE INITIAL FLOW IN MPTCP
MOTIVATION



If the initial flow cannot be established, there is no connectivity!

Even if an other working path is available.

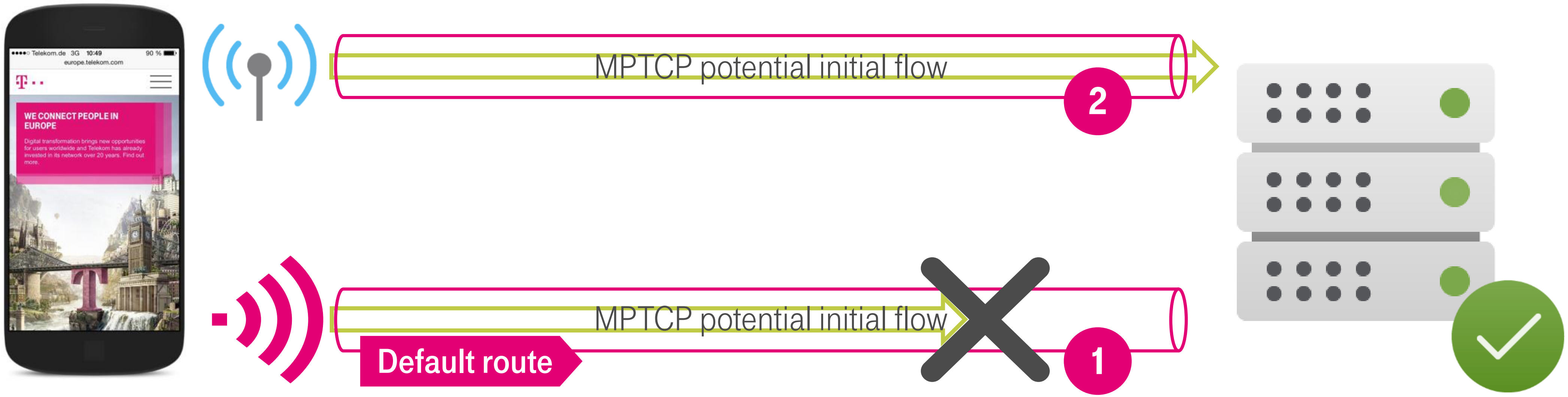
MPTCP RobE IDEA

"If there is at least one functional path,

a connection must be possible"

MPTCP RobE IDEA

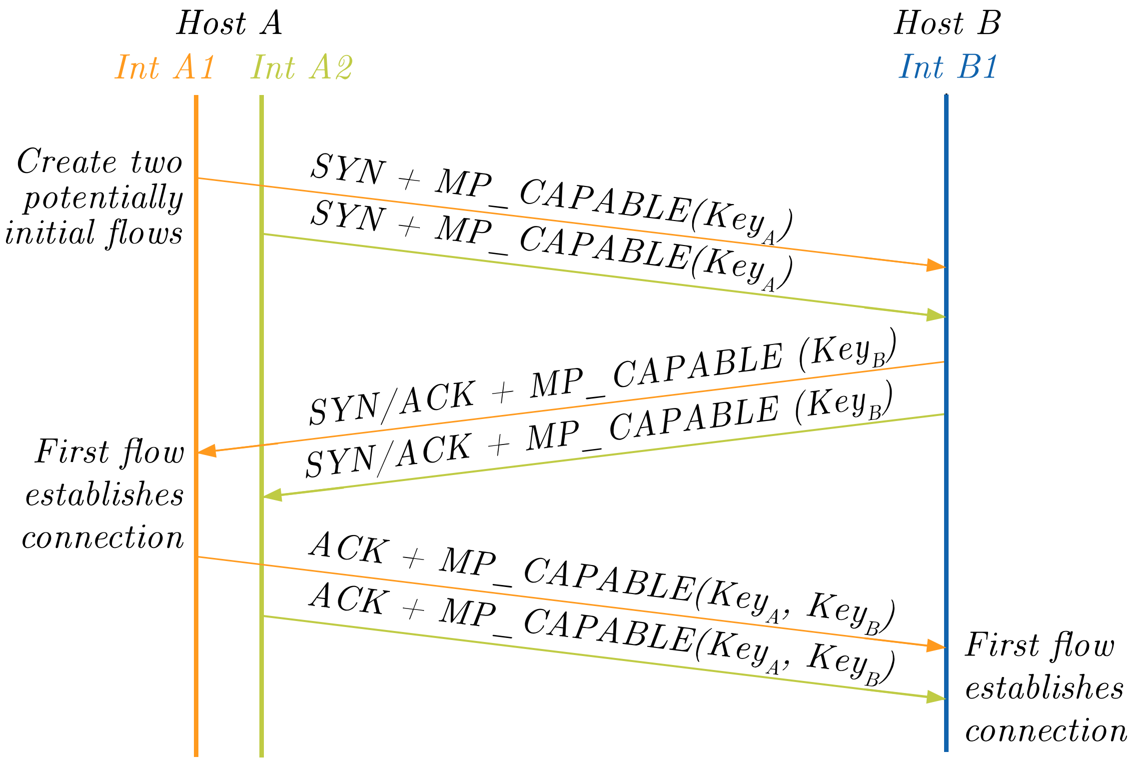
INTRODUCE POTENTIAL INITIAL FLOWS



Any path can be used to establish a connection!

MPTCP RobE PROPOSALS & CRITERIA

1. DOWNGRADE POTENTIAL INITIAL FLOWS

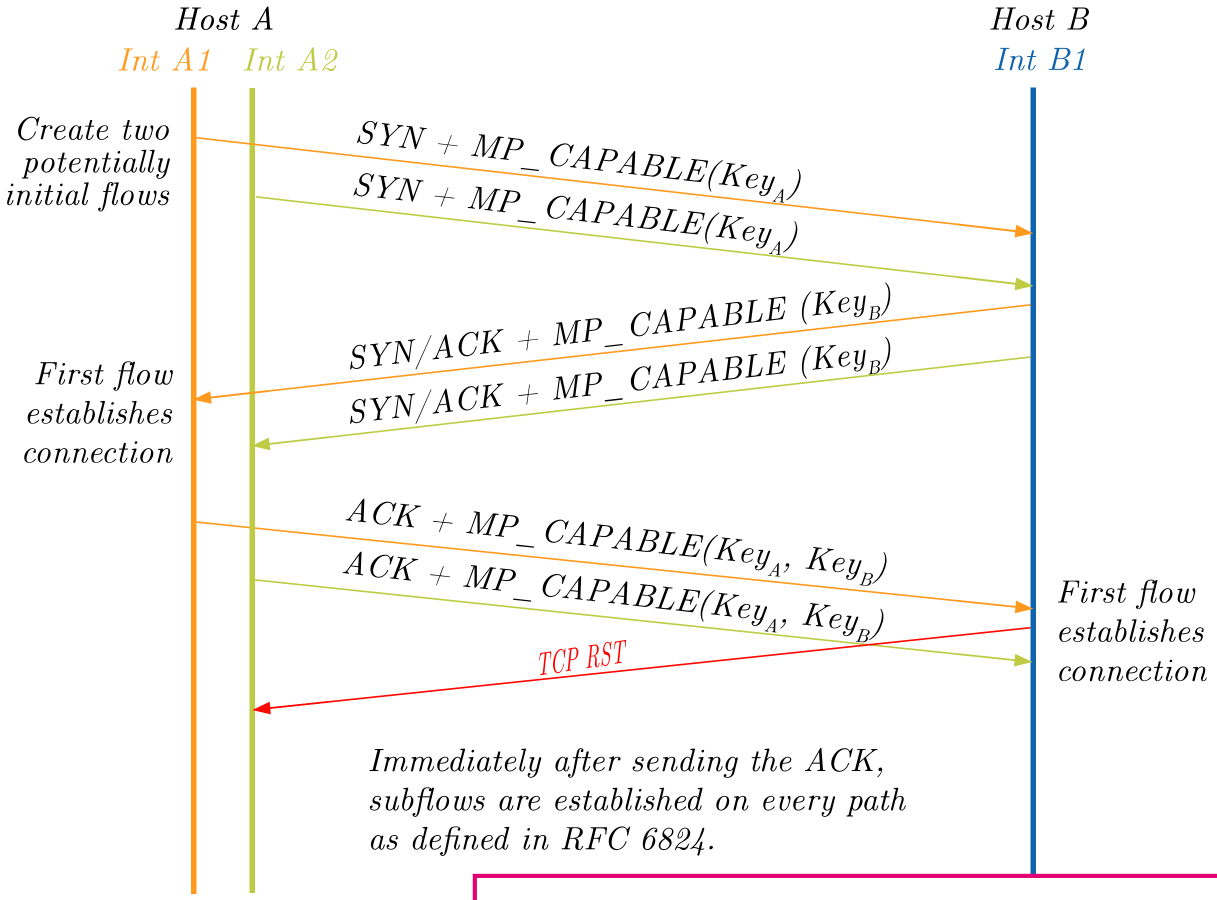


- Two *potentially initial flows* are established over the two available paths
- The first flow that returns establishes the connection on both endpoints
  - This resembles an **initial flow**
- The second flow will be attached to the existing end-to-end connection
  - This flow is *downgraded* and now acts like a **subsequent flow**

Guarantees robustness and overall latency reduction without any network overhead

MPTCP RobE PROPOSALS & CRITERIA

2. BREAK BEFORE MAKE

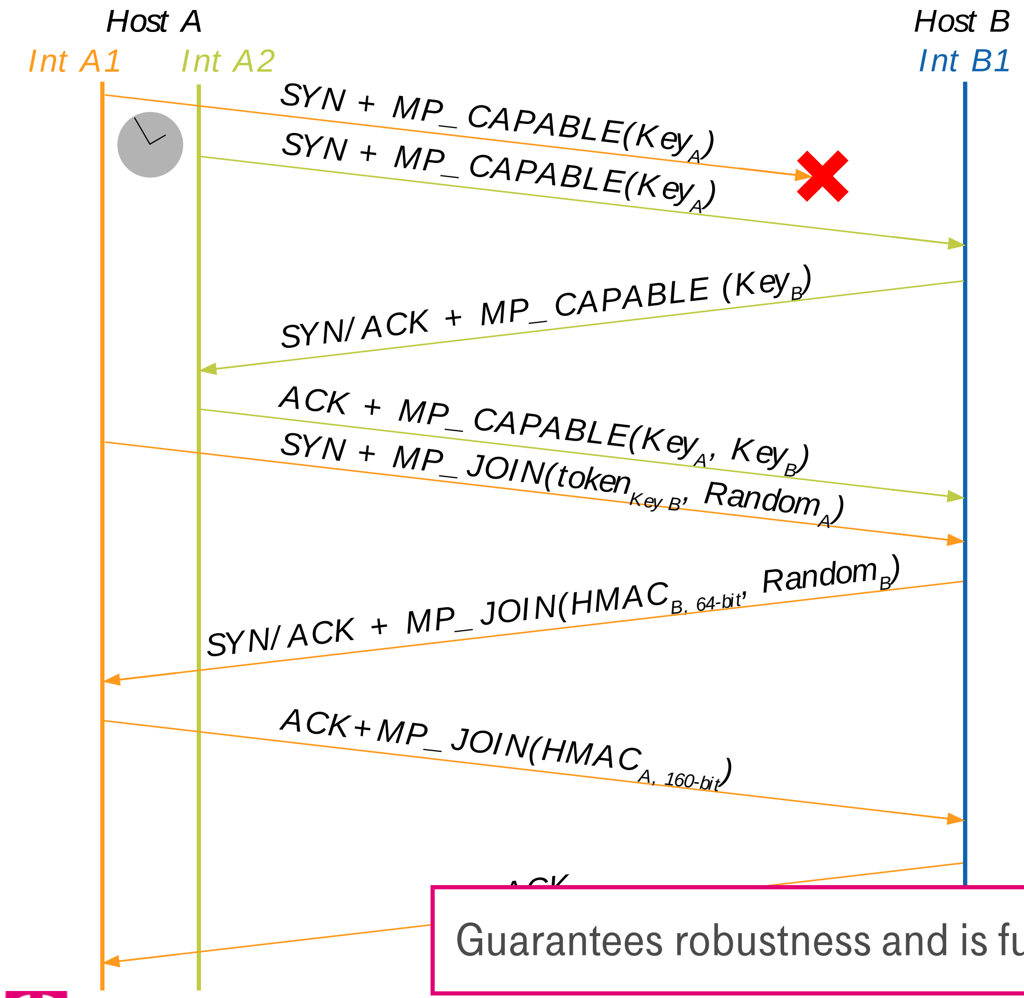


- Two *potentially initial flows* are established over the two available paths
- The first flow that returns establishes the connection on both endpoints
- As soon as one flow is fully established, Host B resets all other flows
- Additional flows are created as described in RFC 6824

Guarantees robustness and some latency reduction, but cause additional network overhead

MPTCP RobE PROPOSALS & CRITERIA

3. THE TIMER SOLUTION



- The SYN retransmission timer is modified
  → If the initial path is defective, the client will retry on another path, like Happy Eyeballs (RFC 6555)
- After the initial flow is successfully established, subsequent flows can be created as defined in RFC 6824
- The SYN/ACK of the first flow might arrive after the second flow is fully established
  - The first SYN/ACK can be dropped
  - Or the first flow can be downgraded (as in proposal 1)

Guarantees robustness and is fully standard compliant, but less efficient

MPTCP RobE PROPOSALS & CRITERIA
COMPARISON

„Downgrade“

Pro:

Most efficient in terms of

- Robustness
- Overall latency reduction
- Network overhead

Con:

- Needs sender & rec. modification
- Possibly some standard extension
- Most challenging

„ Break before make“

Pro:

Efficient in terms of

- Robustness
- Initial flow latency reduction

Con:

- Needs sender & rec. modification
- Possibly some standard extension
- Challenging
- Netw. overhead (1. RST, 2. MP_JOIN)

„Timer“

Pro:

Efficient in terms of

- Robustness
- Implementation (only sender)
- Full standard compliant

Con:

- Less efficient
  - Latency
  - Netw. Overhead
- Possibly latency increase

MPTCP RobE PROPOSALS & CRITERIA

CRITERIA & SELECTION

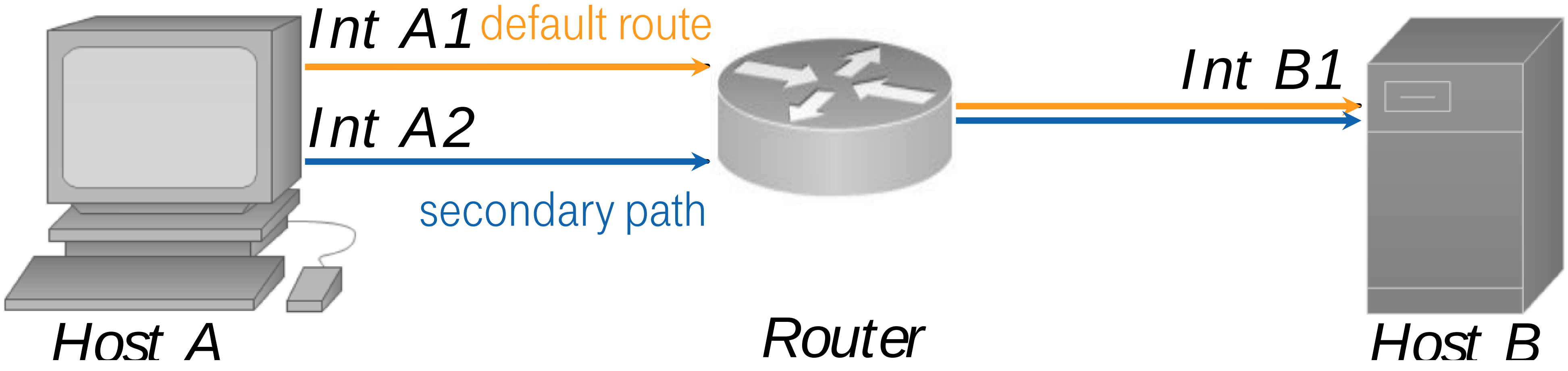
	Proposal 1 (Downgrade)	Proposal 2 (Break before make)	Proposal 3 (Timer)
Robustness	✓	✓	✓
Netw. overhead minimized	✓	✗	✓
Latency: increase/reduction	✓ / ✓	✓ / ✓✗	✗ / ✗
Standard compliance	✓	✓	✓

Note:

- Robustness: If there is at least one functional path, a connection must be possible
- Overhead and latency: The solution should not introduce excessive amounts of overhead and latency compared to standard MPTCP
- Standard compliance: MTPCP RobE should use and integrate with existing standards and needs only, if required, minor adaption.

EXPERIMENTAL INVESTIGATION PREF. APPROACH

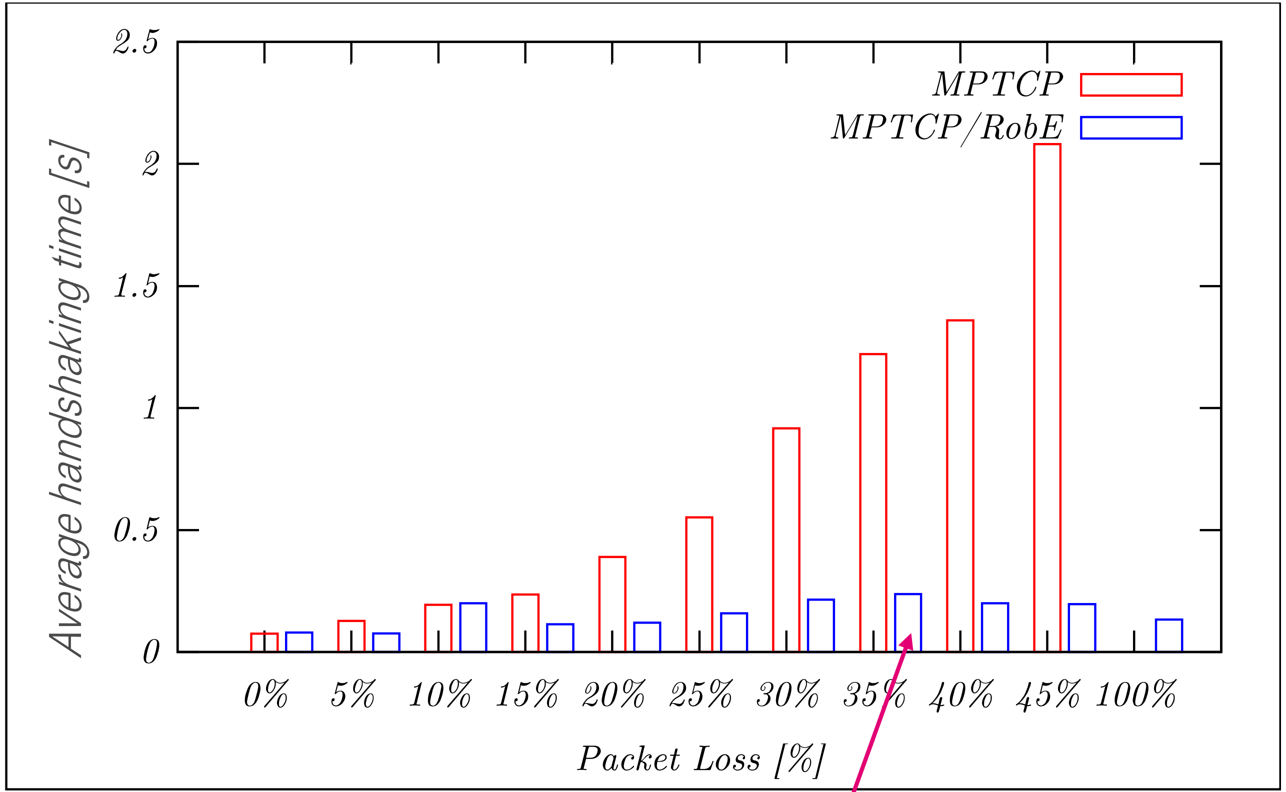
EXPERIMENT SETUP – LAB



Objective: Evaluation of robustness and latency gains with MPTCP RobE

EXPERIMENTAL INVESTIGATION PREF. APPROACH

EXPERIMENT 1 – ROBUSTNESS



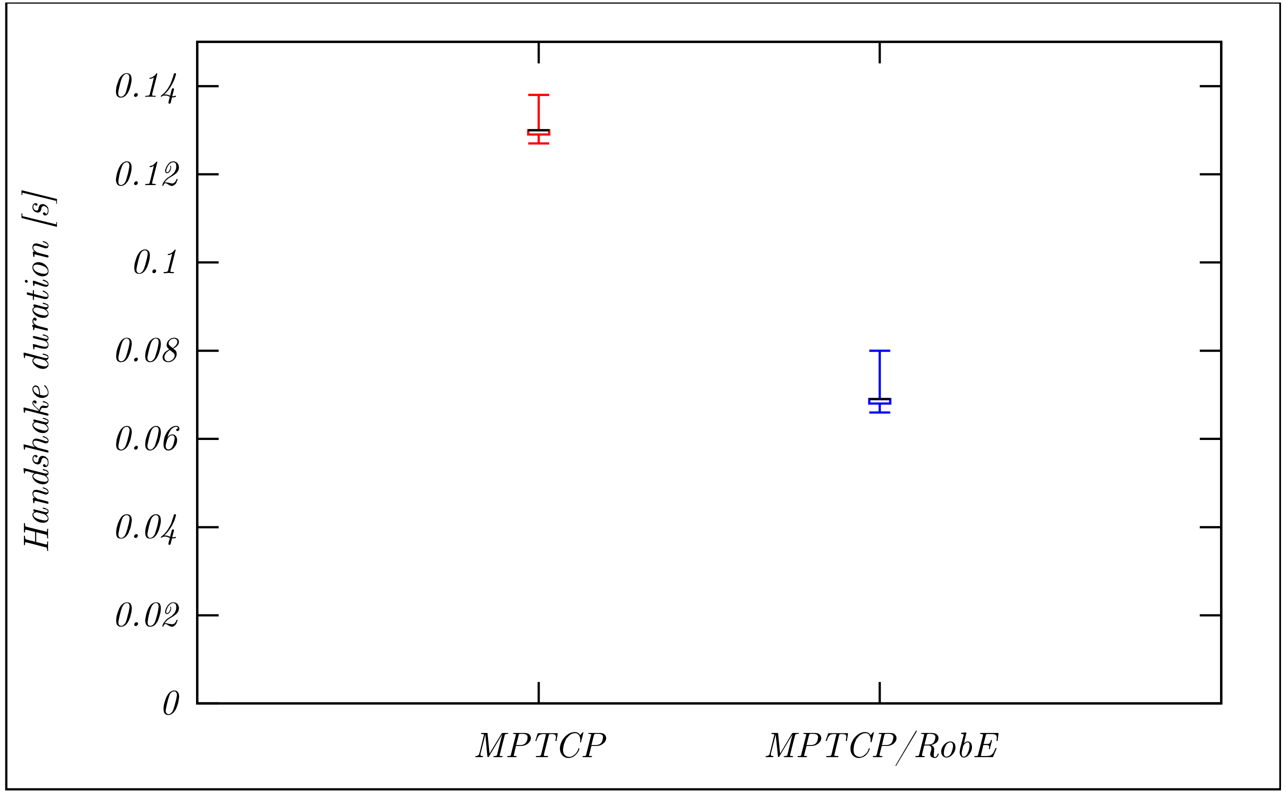
RobE should stay stable over time, here it is not fully the case because impl. and setup is still imperfect. But it shows a clear indication compared to MPTCP

- **Path setup**
  - Variable packet loss rate on default route
  - No packet loss on secondary path
  - Same latency on both path
- **TCP Retransmission Timeout (TCP_RTO):** If a SYN is lost, the client will retry after some time
  - **TCP_RTO >> RTT** in most cases
- **"Exponential Back-Off"** mechanism
  - TCP_RTO increases exponentially with every failed attempt
- **MPTCP:** The average loading time increases exponentially
  - No connection possible for 100% loss

No increase in handshaking times with MPTCP RobE

EXPERIMENTAL INVESTIGATION PREF. APPROACH

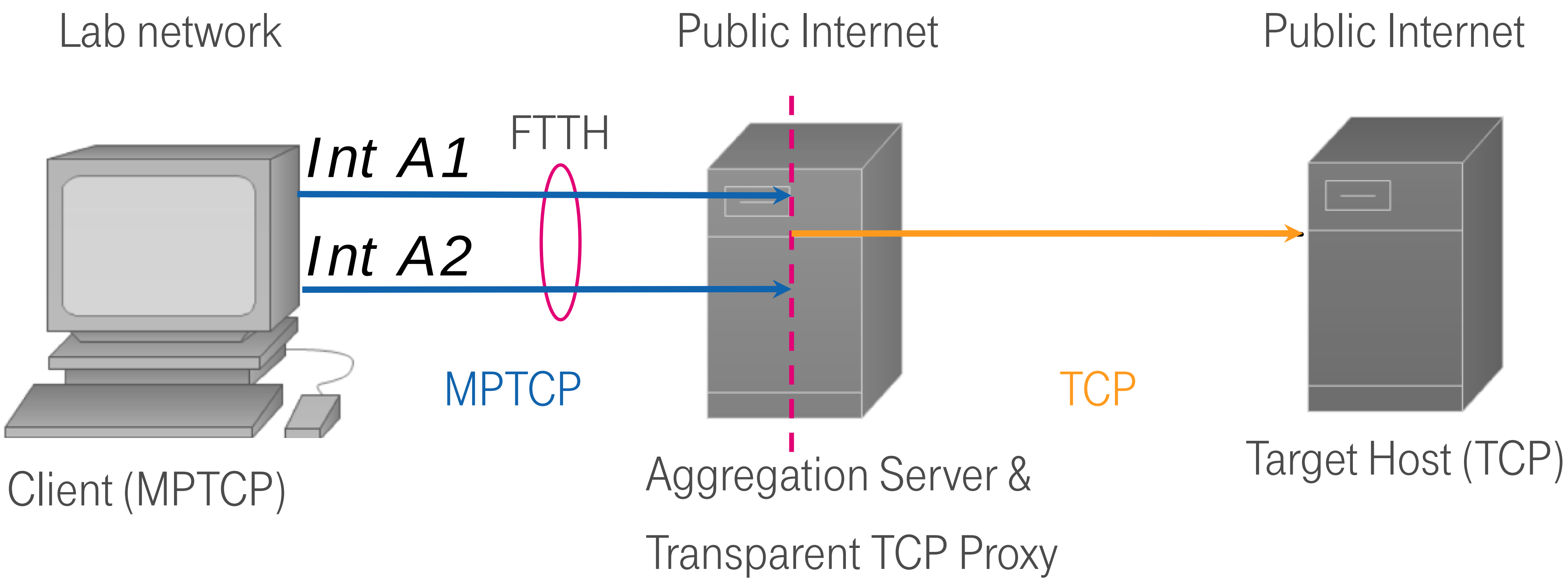
EXPERIMENT 2.1 – LATENCY



- The default route does not have the lowest latency
  - Default route: 40ms one-way latency
  - Secondary path: 20ms one-way latency
  - Simulates an unloaded mobile (LTE) and a fixed access (DSL/WiFi) link
- Handshake duration: approximately **1,5 * RTT**
- **MPTCP:** the default route determines the handshake duration

MPTCP RobE: the handshake duration is determined by the quickest path available

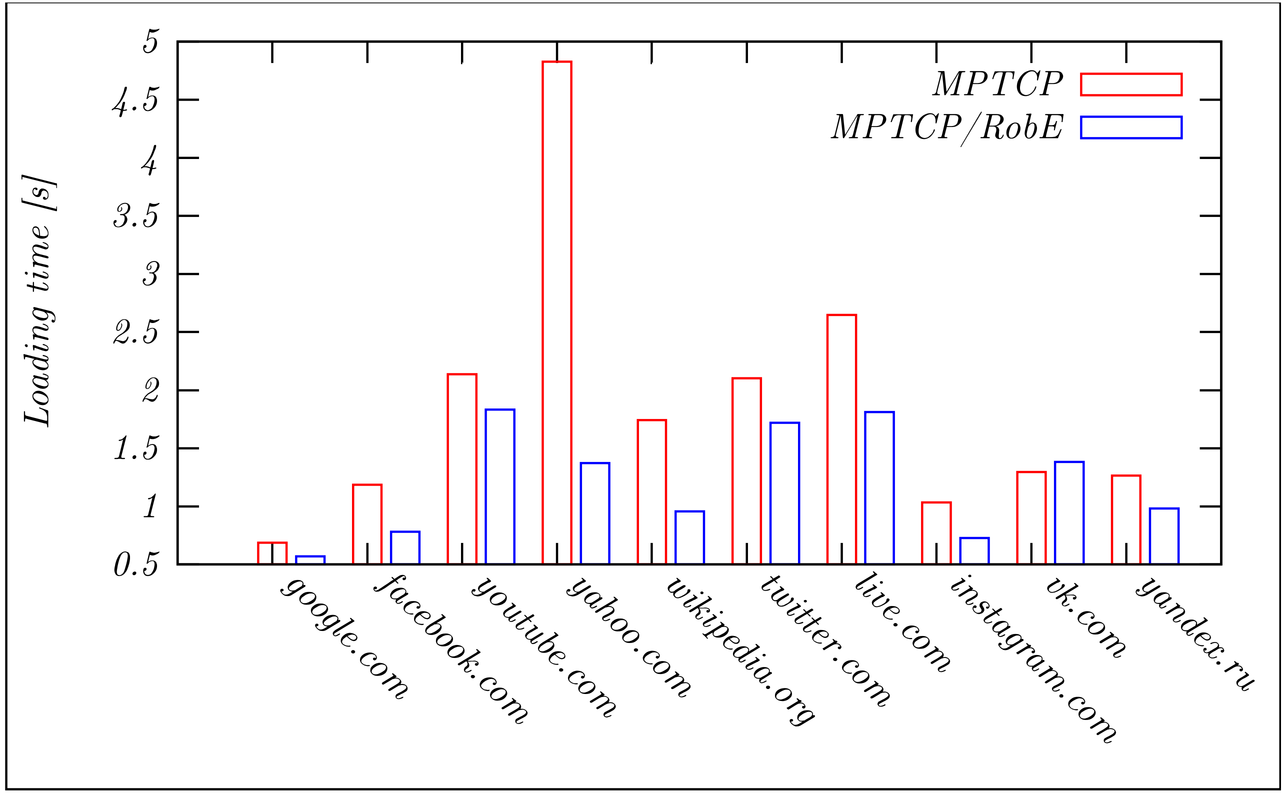
EXPERIMENTAL INVESTIGATION PREF. APPROACH
EXPERIMENT SETUP – REAL-WORLD



Objective: Evaluation of loading time improvements with MPTCP RobE

EXPERIMENTAL INVESTIGATION PREF. APPROACH

EXPERIMENT 2.2 – REAL-WORLD



- The default route does not have the lowest latency
  - Default route: 40ms one-way latency
  - Secondary path: 20ms one-way latency
- The top ten most popular websites were downloaded in an automatic procedure and filtering ads.
- Websites consist of many resources from different hosts and locations
  - Many short TCP connections necessary
  - MPTCP receives additional latency each time

The MPTCP RobE prototype benefits from a quicker secondary path in two ways*:

1. First connectivity latency is reduced
2. BW aggregation starts earlier

*The result shows a first good indication of the speed boost gained by MPTCP RobE, which will be still more accelerated in future by an optimized implementation of MPTCP RobE

CONCLUSION AND FUTURE WORK

GENERAL FACTS & DISCUSSION

- MPTCP RobE can protect MPTCP against network outages during connection establishment
- It can improve the user experience in terms of reliability and latency
- Under most circumstances, loading times can be shortened by having max. throughput earlier available
- First “Downgrade” reference implementation is done (based on MPTCP v0.90)

- Is there a need for robust establishment?
- Where should it take place, application or MPTCP layer?
- Want we benefit from robustness AND latency reduction?
- Which approach fits best in future?
- How to integrate MPTCP RobE into MPTCP standard and/or implementation?
- Develop or improve existing reference implementation and make it public available.

CONCLUSION AND FUTURE WORK
DETAILED DISCUSSION

- **“Downgrade” approach still needs to solve the following standard and impl. relevant points:**
  - Duplicating SYN request introduce
    - additional processing overhead on receiver side to check for “duplication” (also applies to “Brake before make”)
    - misuse by accident or by intention of KeyA which is already in use
      - can be mitigated by allowing new KeyA requests only during a time frame until first flow is established
      - And /or using remaining 4Bytes in MP_CAPABLE to (see RFC6824, Appendix A) to indicate identity
    - Address ID (RFC6824, 2.2, 2.3, 2.7 ...) negotiation for potential initial flows
- RobE support negotiation
- Fallback mechanism
- RFC6824bis is missing KeyA in the SYN ☹

- **Other approaches**
  - IETF 97 NICT proposal (https://www.ietf.org/proceedings/97/slides/slides-97-mptcp-a-proposal-for-improving-mptcp-initialization-00.pdf)
  - Exploit Happy Eyeballs (RFC6555) approach for possible application only solution

Thank you very much for your attention

If there are any questions, please feel free to ask.

Markus Amend
markus.amend@telekom.de