MULTIPATH TRANSPORT CHALLENGES AND SOLUTIONS

Michael Scharf, Thomas-Rolf Banniza

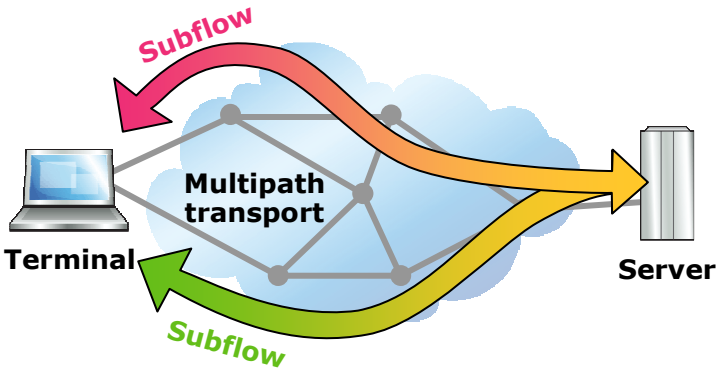
October 2011

Supported by the German-Lab project NETCOMP funded by the German Federal Ministry of Education and Research (BMBF).

MOTIVATION
TCP-BASED MULTIPATH TRANSPORT

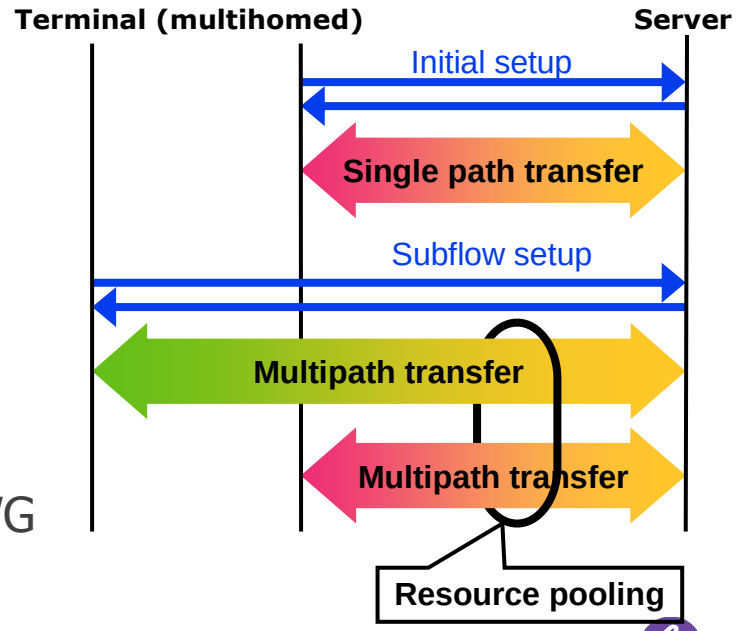
- **Challenge**: Multipath transport in the Internet
  - Aggregation of capacity for terminals with more than one interface (e. g., LTE+Wifi)
  - Extension of Transmission Control Protocol (TCP) to deal with multiple subflows

  ➔ **"Network MIMO"** is a missing feature



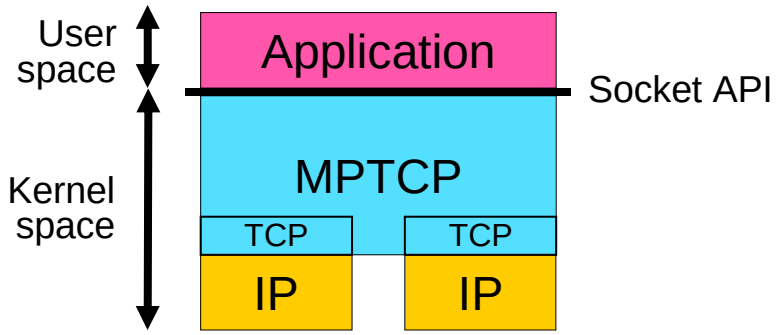
- **Potential solution**: Multipath TCP
  - New TCP-based multipath transport protocol with several subflows
  - Resource pooling of different paths
  - Assumption: At least one multi-addressed endpoint (multi-homed device)
  - Backward compatible to current TCP, in particular same socket API

  ➔ **Ongoing IETF standardization** in MPTCP WG



A MULTIPATH TRANSPORT SHIM LAYER
MULTIPATH TRANSPORT DESIGN SPACE

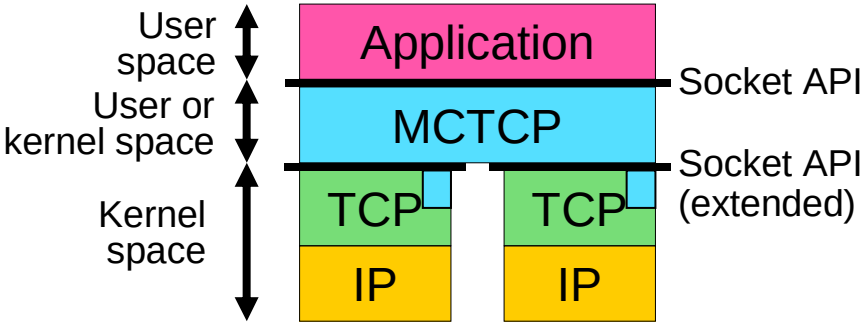
Option encoding



- MPTCP: IETF solution
- TCP extension adding multipath transport capability for unmodified applications
- New TCP options, consuming large part of scarce TCP option space
- Tight integration in TCP/IP stack

➔ draft-ietf-mptcp-multiaddressed

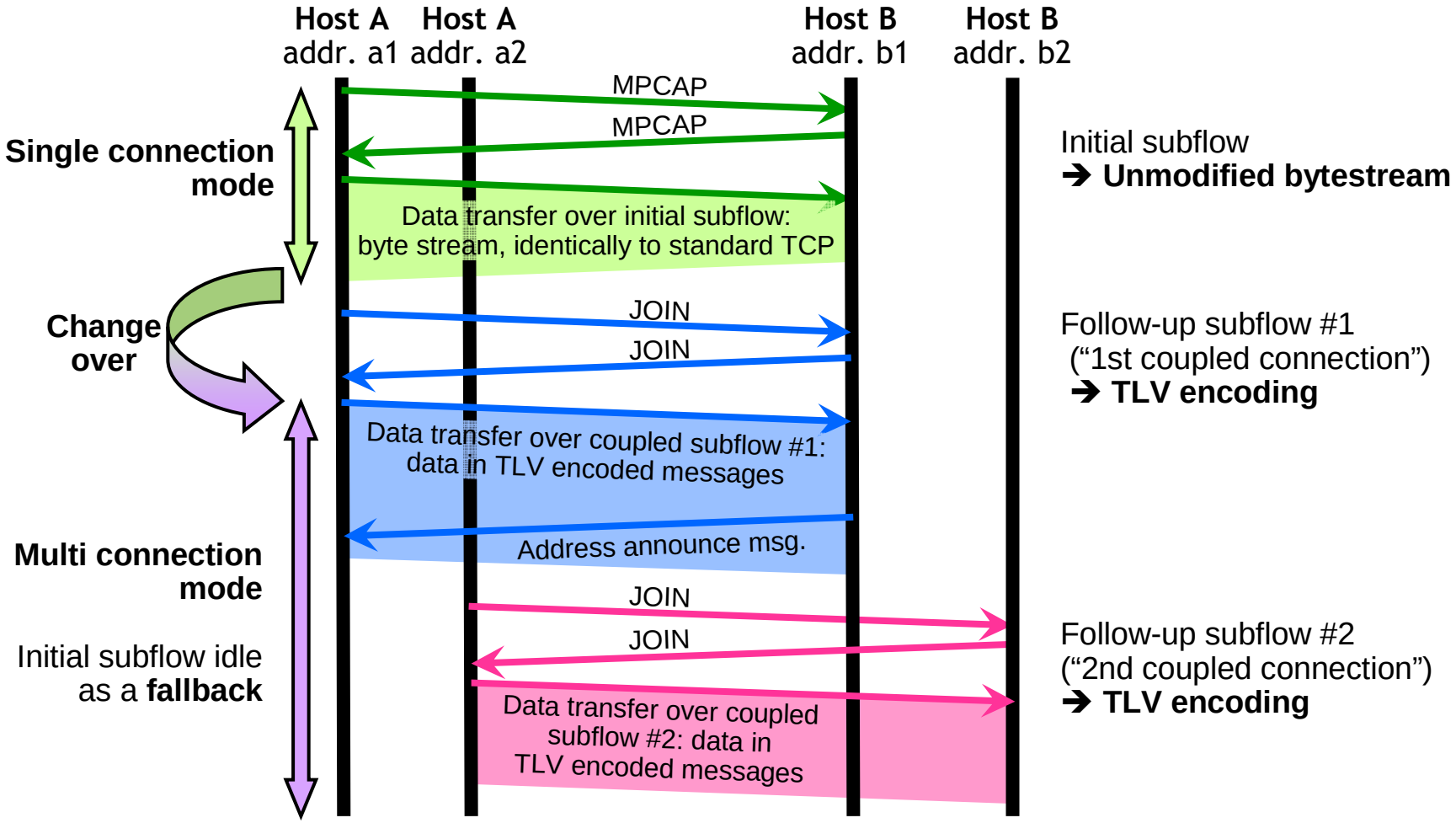
Shim layer



- MCTCP: Multi-Connection TCP
- Hybrid approach combining a minimal TCP extension and an app protocol
- Simple and extensible type-length-value (TLV) encoding whenever possible
- Shim-layer implementation possible

➔ draft-scharf-mptcp-mctcp

A MULTIPATH TRANSPORT SHIM LAYER
MPTCP MESSAGE SEQUENCE DIAGRAM

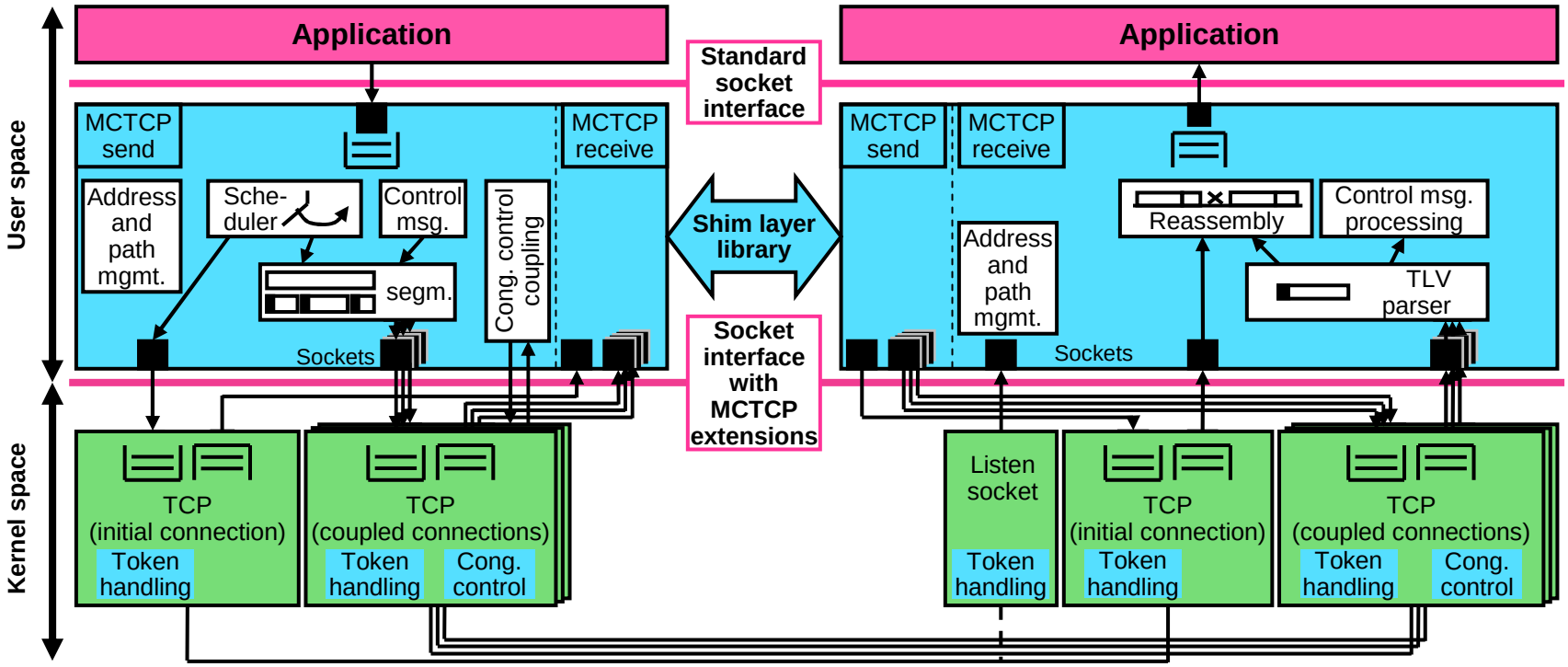


➔ **Payload encoding** outside initial subflow (used also for fallback)

A MULTIPATH TRANSPORT SHIM LAYER
COMPARISON OF MPTCP AND MCTCP

	Option encoding (MPTCP)	Shim layer (MCTCP)
Signaling connection setup	**TCP options**	**TCP options**
Signaling during session	**TCP options**	**TCP payload**
Extensibility	**Difficult**	**Simple**
Dropped/striped options in SYNs	**Fallback**	**Fallback**
Dopped/striped options outside SYNs	**Fallback or failure**	**Not affected**
Protocol helpers (e. g., NAPT)	**Simple**	**Difficult**

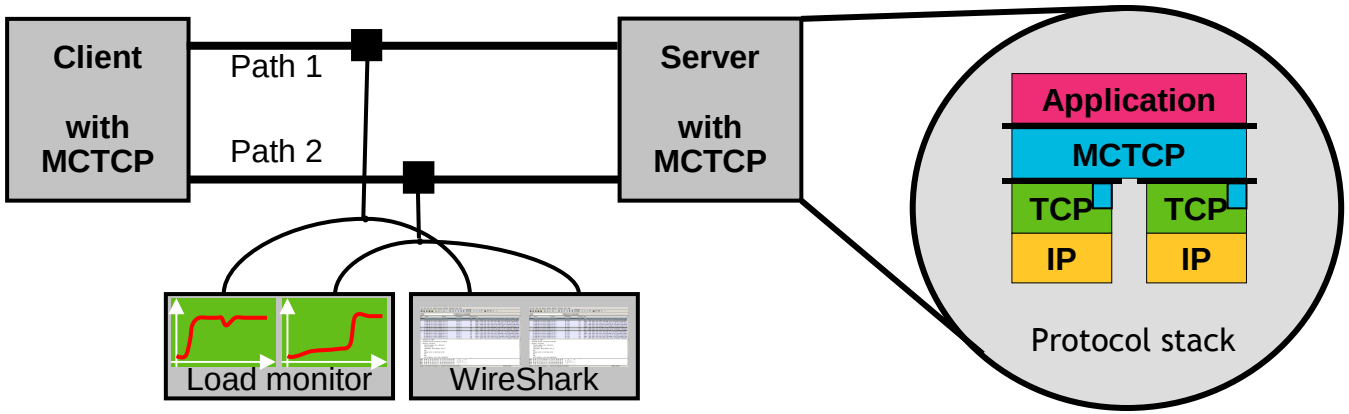
LINUX IMPLEMENTATION
COMPLEXITY AND LESSONS LEARNT



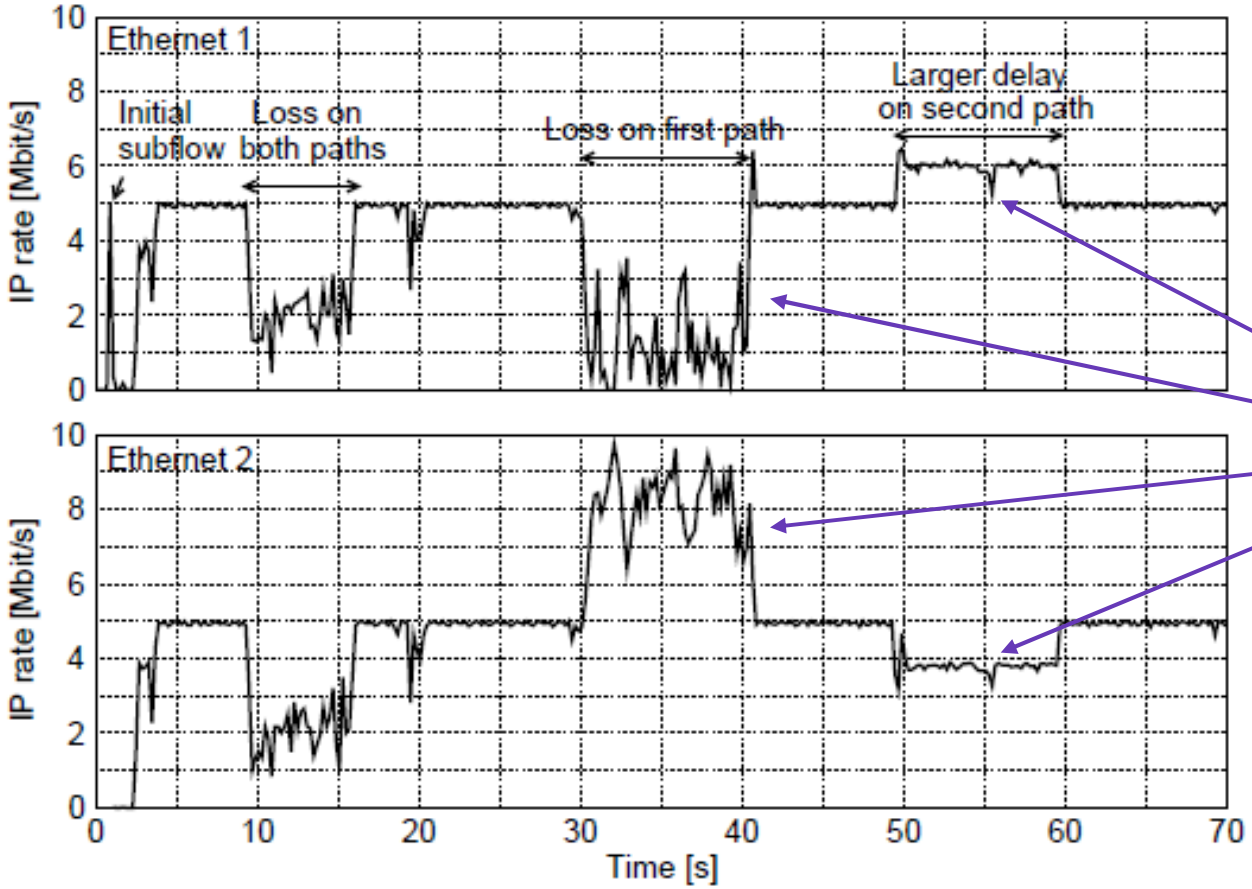
- Minimal **Linux 2.6.32 kernel patch**: 800 lines of code
- **Shim layer library**: 5000 lines of code (baseline protocol)
- **Challenges** for shim approach: Memory copies, flow control, blocking calls

PERFORMANCE RESULTS
EVALUATION METHODOLOGY

- Testbed setup
  - Two computers with Linux Ubuntu 10.04 running patched kernel
  - Two disjoint 10 Mbit/s Ethernet links, optionally also Wireless Local Area (WLAN)
  - Delay and packet loss by Linux "netem" tool
- TCP stack parameters
  - CUBIC congestion control, unless mentioned otherwise
  - Socket buffer size set to 262144B to prevent limitations by the TCP flow control
  - Other TCP stack parameters are set to the default values
- Applications
  - Simple client and server programs written in C
  - Also tests with a real video streaming application (VLC player) and a HTTP server



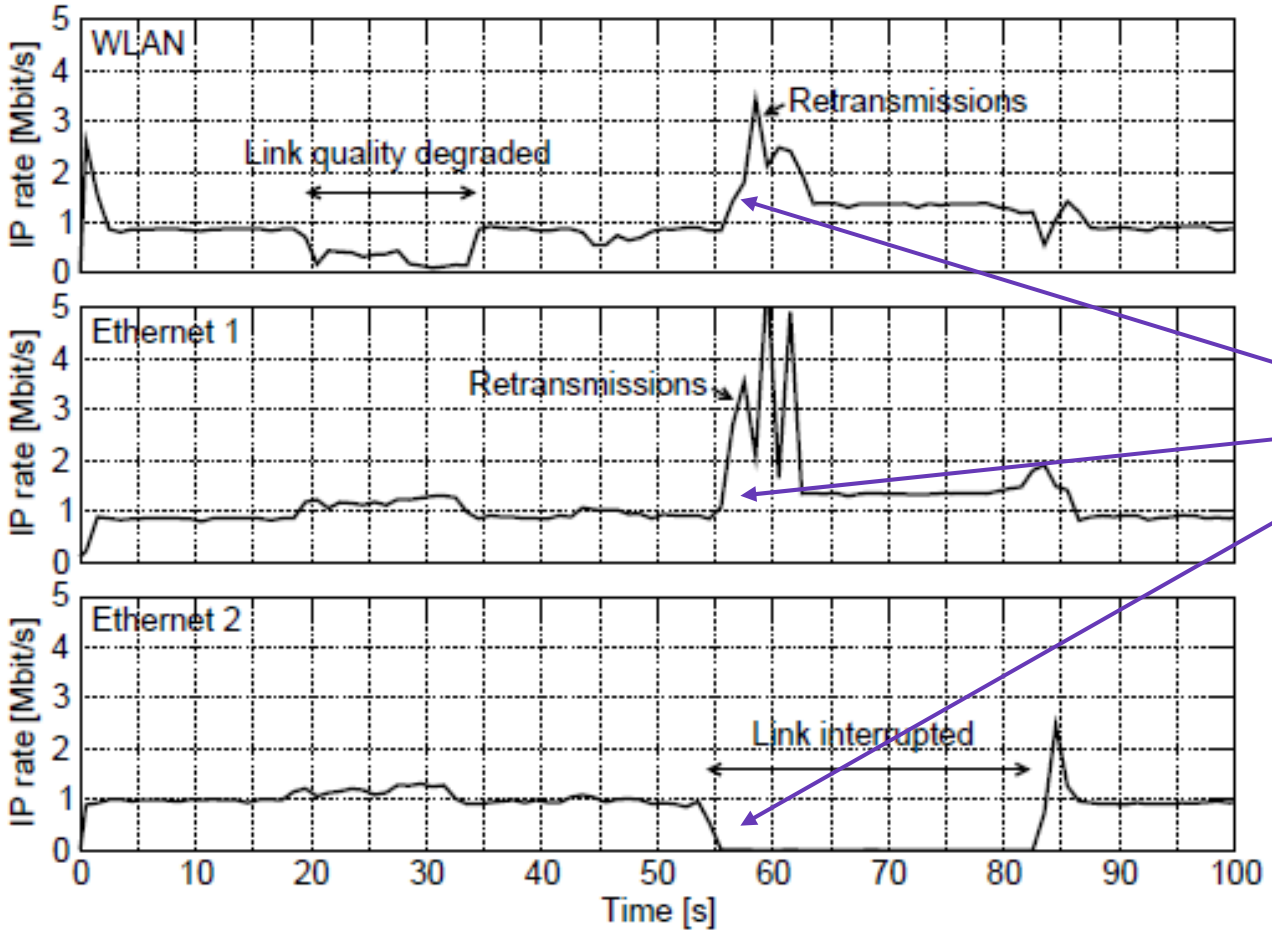
PERFORMANCE RESULTS
EFFICIENCY OF RESOURCE POOLING



Shifting of the load to better path

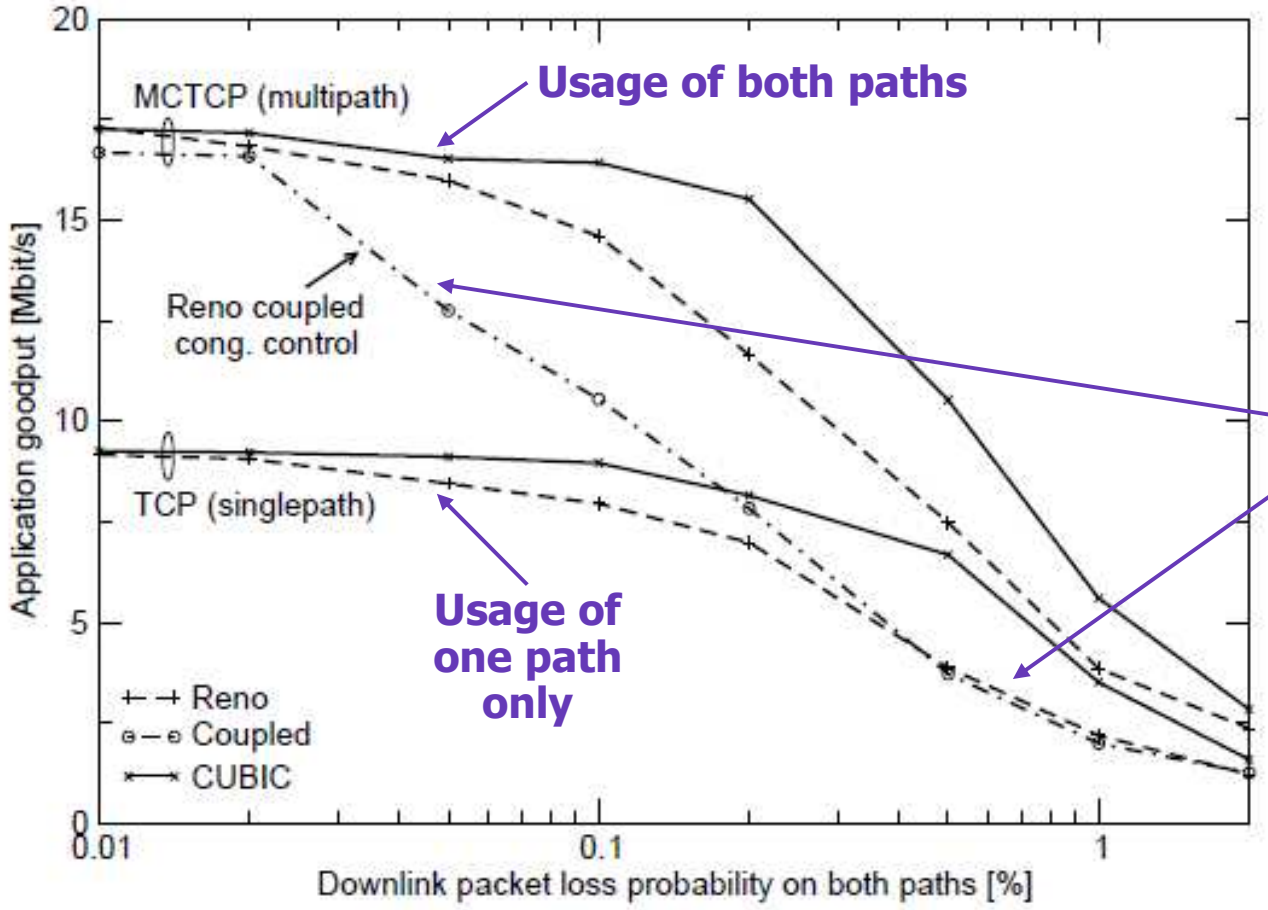
➔ MCTCP **dynamically pools the available bandwidth** of several paths

PERFORMANCE RESULTS
DEALING WITH LINK OUTAGE



➔ **Compensation of path failures** transparent to applications

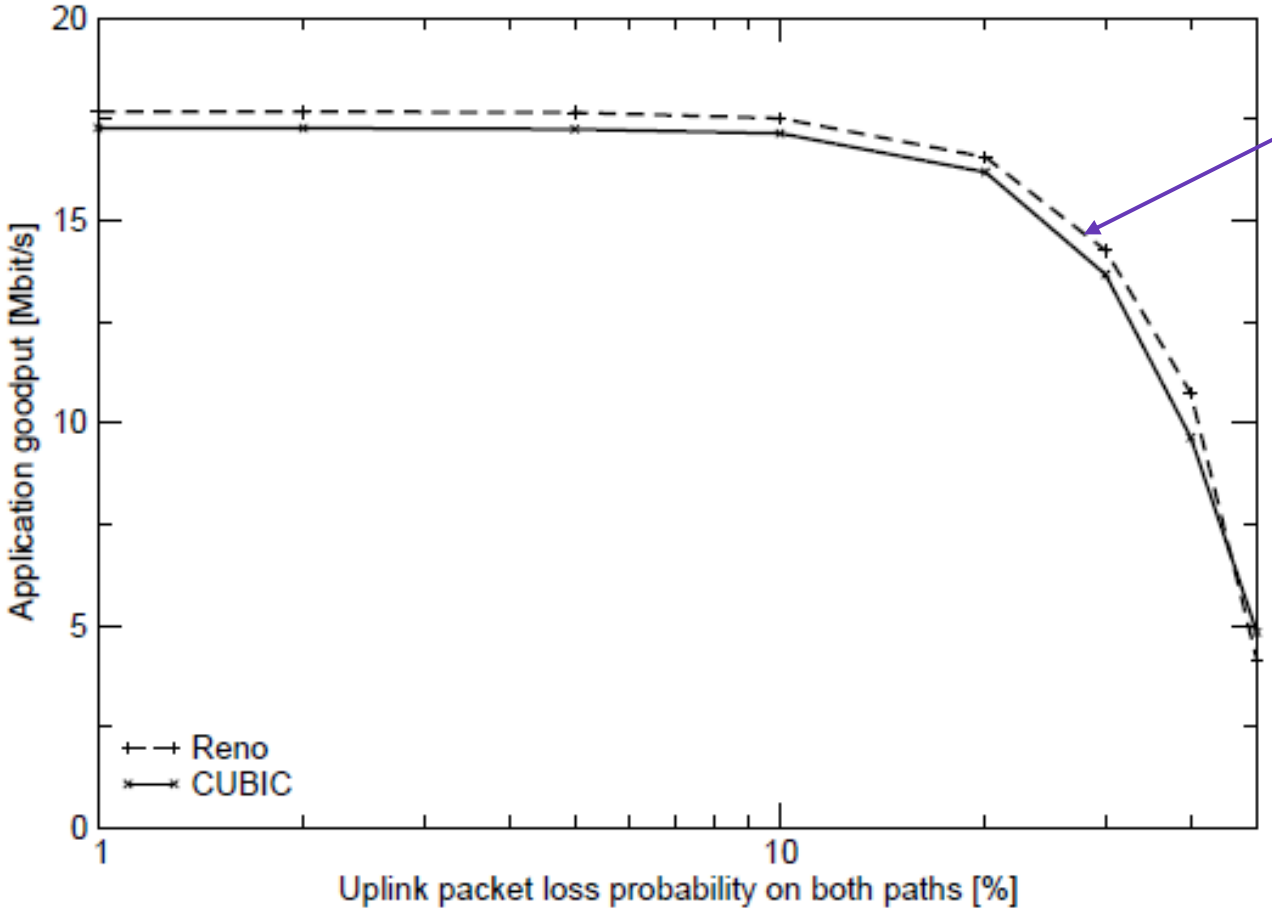
PERFORMANCE RESULTS
PERFORMANCE OVER CONGESTED LINKS



Coupled congestion control: Fall-back to singlepath TCP for high loss rate

➔ Possible **support of a coupled congestion control** for fairness

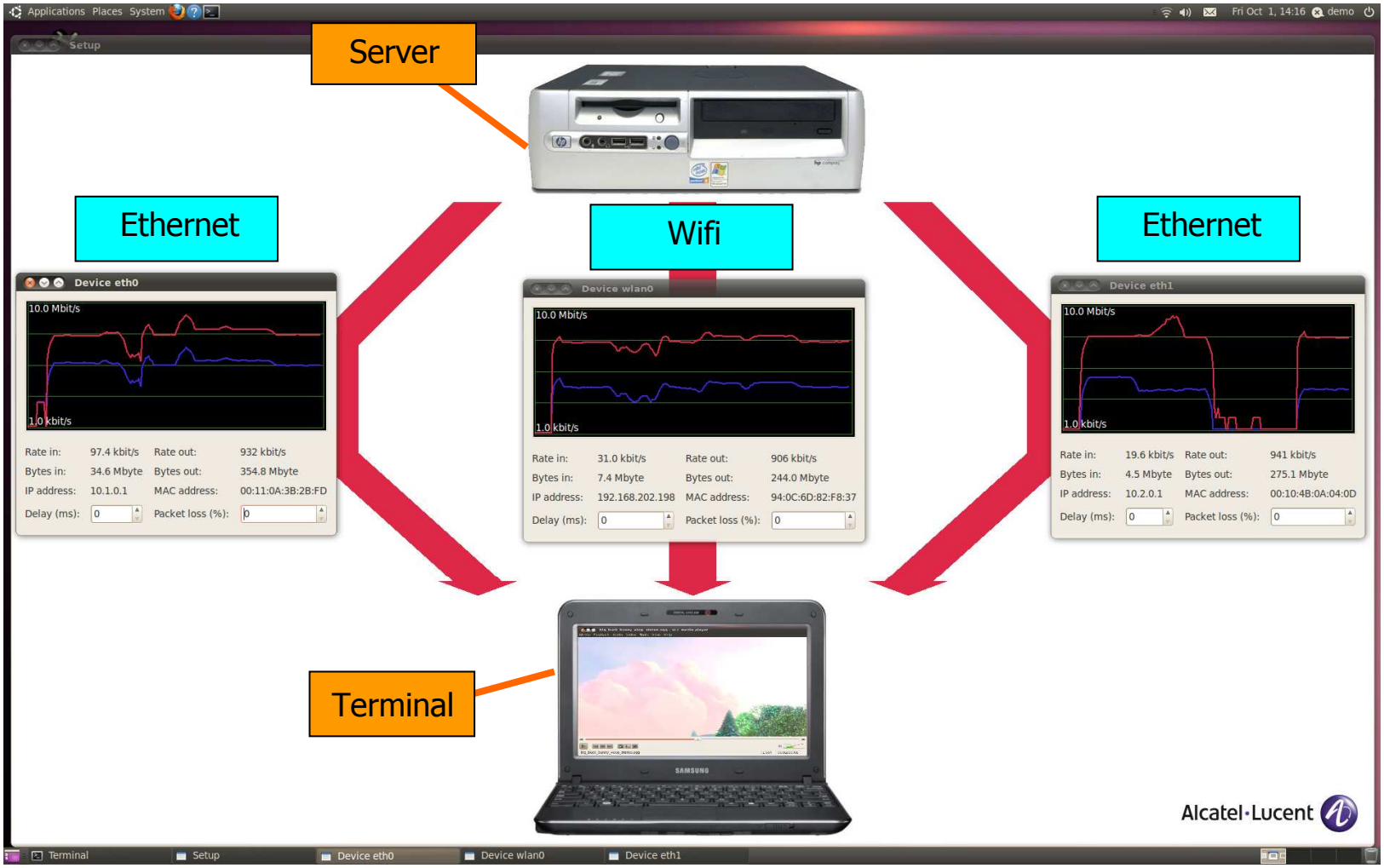
PERFORMANCE RESULTS
CONGESTED REVERSE PATH



Impact only for loss packet rates larger than 10%

➔ **No significant impact by reverse path congestion** even with DATA ACKs

PERFORMANCE RESULTS
BELL LABS OPEN DAY DEMO 2010



➔ Multipath HTTP-based video **streaming without interruptions**

MULTIPATH CHALLENGES
MULTIPATH: QUO VADIS?

- Multipath use cases
  - Multi-interfaced wireless devices: Vendor support?
  - Data centers: Benefit?
  - What else? Benefit only for several roughly equal paths...
- Protocol design issues
  - MPTCP specification is still a moving target
  - Additional complexity (checksums etc.), TCP option space limitation
  - Currently one major MPTCP implementation only
- Fundamental questions
  - Multiple addresses ≠ multiple paths
  - Simple, robust algorithms for many degrees of freedom

MULTIPATH CHALLENGES
IMPACT ON CAPACITY SHARING?

- Question: Would multipath fundamentally affect capacity sharing?
  (assuming widespread deployment)
- Well, not necessarily...
  - Multiple paths are seldom
  - No benefit for short flows, bulk data download does it today (e. g., P2P)
  - Even without congestion control coupling still somehow similar to single-path TCP
  - Capacity sharing is mostly a network task
    - Scheduling policies
    - Per-subscriber queues
    - Deep packet inspection (DPI) in border/edge routers
    - ...

CONCLUSION AND OUTLOOK
MULTIPATH CHALLENGES AND SOLUTIONS

Conclusion

- Multi-Connection TCP (MCTCP): A multipath transport shim layer
  - Simple, extensible and robust
  - Alternative to a Multipath TCP solution only using option encoding
- Evaluation of MCTCP
  - Limited implementation complexity
  - Efficient resource pooling of several paths, including congestion control coupling
- Multipath transport: Quo vadis?

Outlook

- Performance comparison with other multipath transport protocols
- More complex setups (e. g., larger topologies, German-Lab platform)

Reference: M. Scharf, T.-R. Banniza, "MCTCP: A Multipath Transport Shim Layer", Proc. IEEE Globecom, 2011

www.alcatel-lucent.com