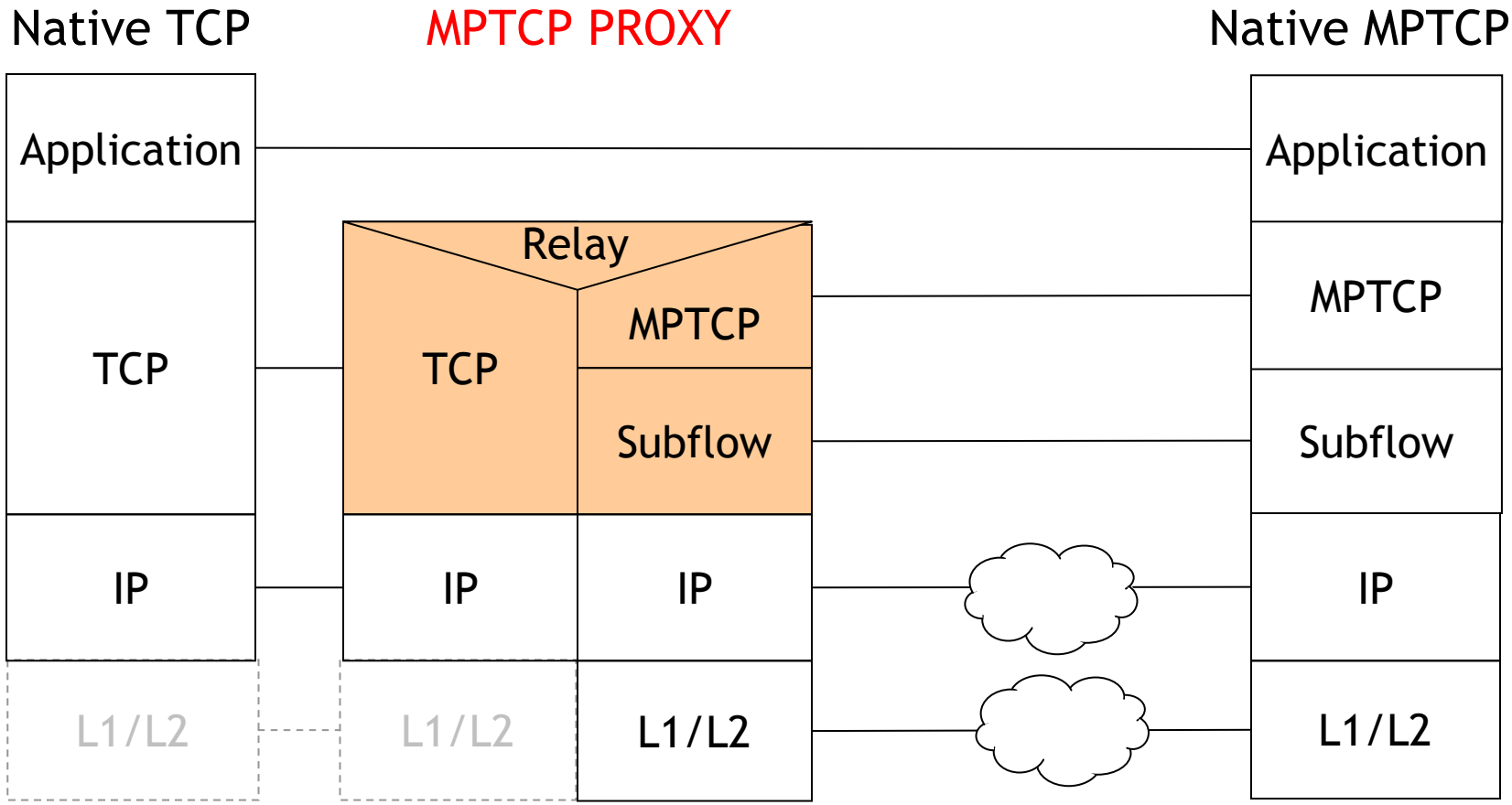
MPTCP PROXY

- Implementation Update
- Interoperability
- Code Release

Georg Hampel, Anil Rana – Bell Labs/Alcatel-Lucent



Realization of MPTCP PROXY:

- Sequential packet processing (Packet Filter)
- No split connection.
- No stream assembly. No segment buffering.

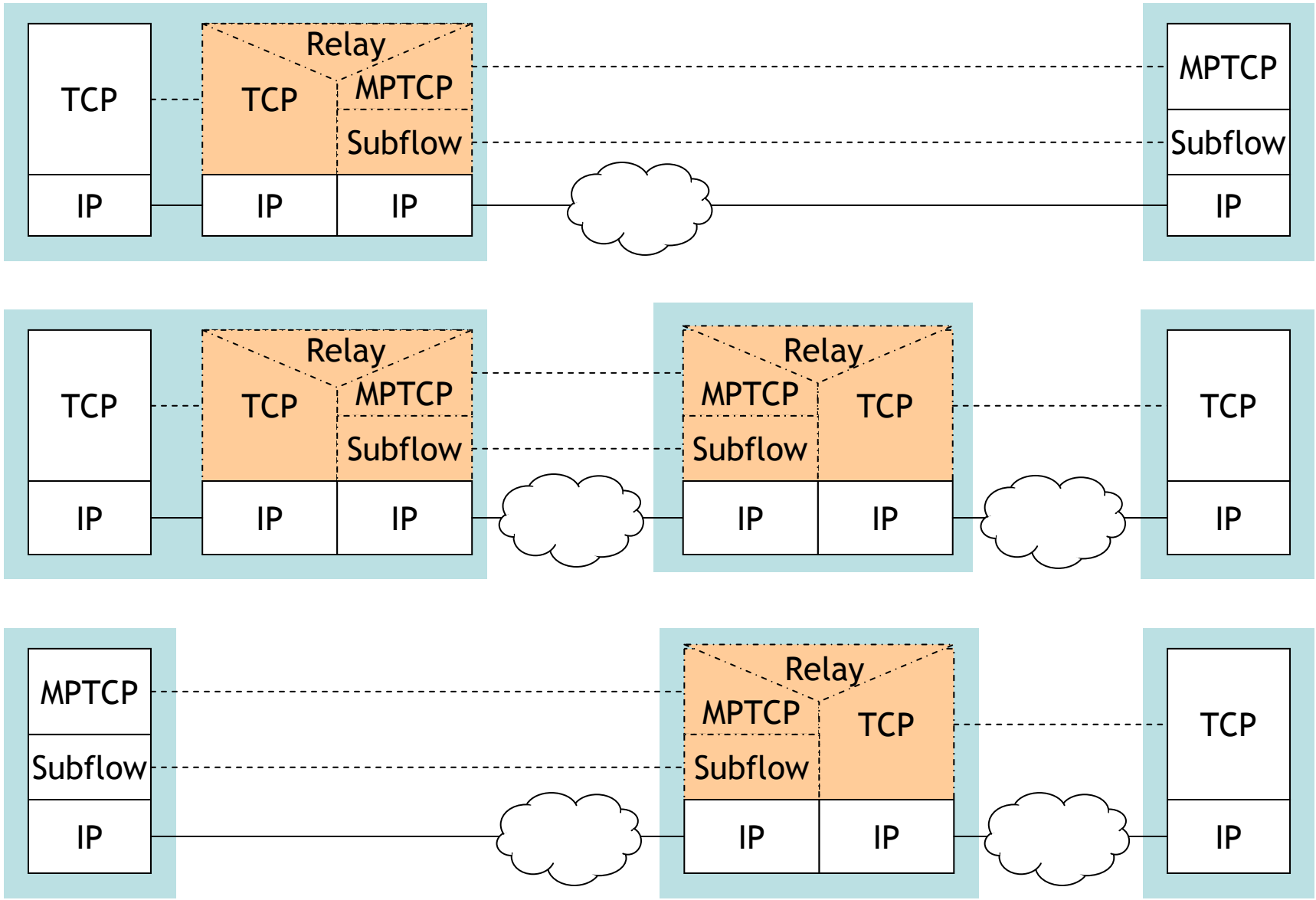
MPTCP Proxy Objectives

- Mobility between untrusted domains
- Cellular traffic offload to untrusted domains

- Exploit MPTCP's middlebox compliance

- Focus: Incremental deployment
  - Mobile: Easy upgrade & cross-platform portability
  - Network: Compliance with gateway processing

Deployment Options



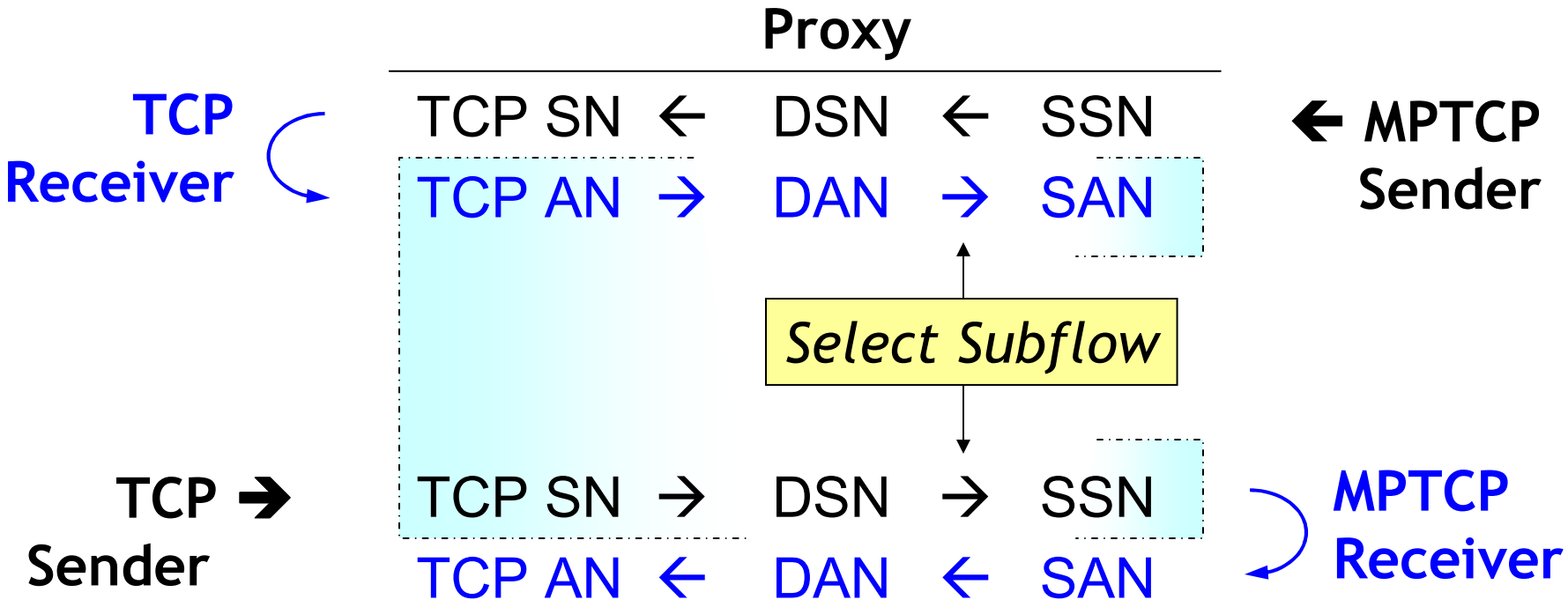
MPTCP Proxy: How it Works.

- Multiple simultaneous subflows per connection
  - Proxy MPTCP Sender: Uses *one* subflow at a time
  - Proxy MPTCP Receiver: Multipath-capable
- MBB handover → Multipath operation for ~RTT
  - New data on *new* subflow
  - In-flight data & retransmissions of old data on *old* subflow
- BBM handover → Single-path operation
  - Immediate cut of old subflow
  - Creates & switches to new subflow, recovers lost data
- Mobility needs only **one flow & congestion control**

MPTCP Proxy

Algorithmic Challenges

MPTCP Proxy: Packet Splitting



- During seamless handover
- When peer does multipath transmission

MPTCP Proxy: ACK Replication

MPTCP Sender → SFL1, DSN=1, SSN=50 → TCP SN=11

SFL2, DSN=2, SSN=60 → TCP SN=12

SFL1, DAN=3, SAN = 51 ← TCP AN=13 ← **TCP Receiver**

SFL2, DAN=3, SAN = 61 ←

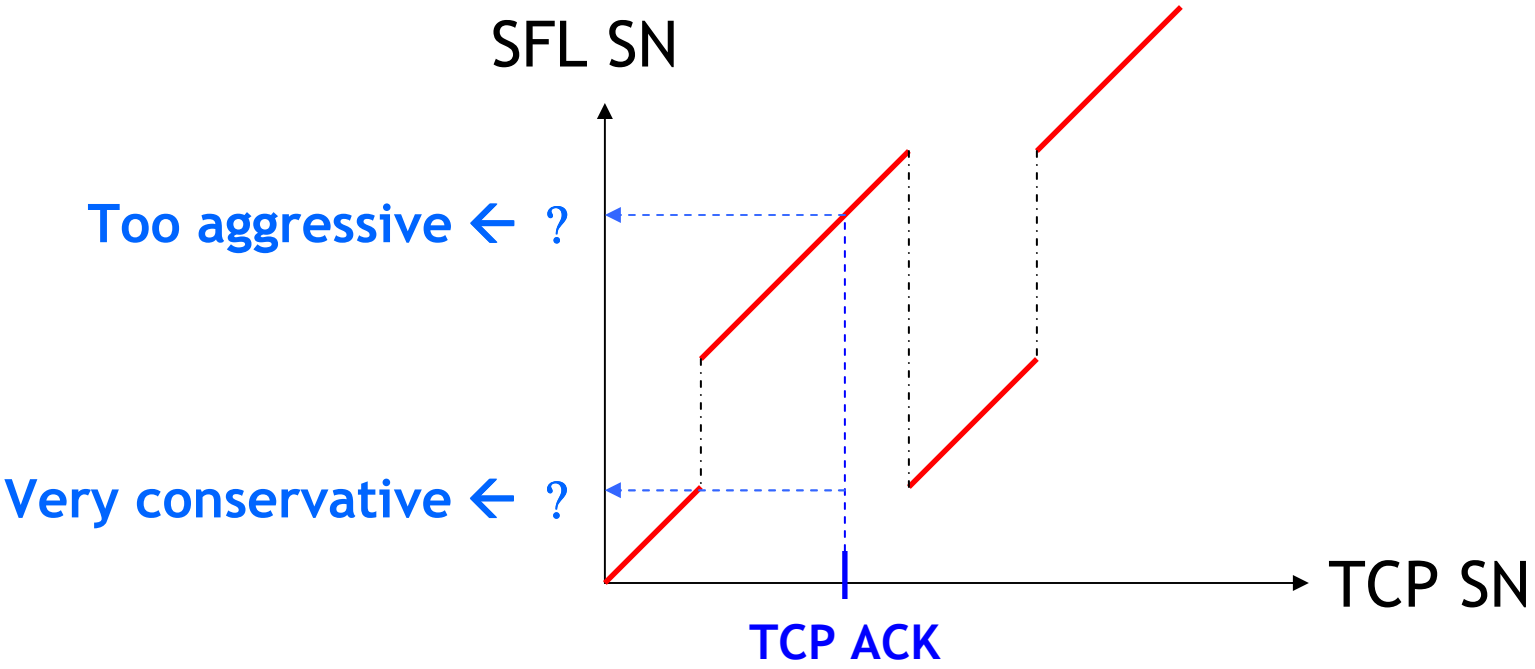
MPTCP Sender → SFL1, DSN=3, SSN=51 → TCP SN=13

SFL1, DAN=4, SAN = 52 ← TCP AN=14 ← **TCP Receiver**

~~SFL2, DAN=4, SAN = 62 ←~~

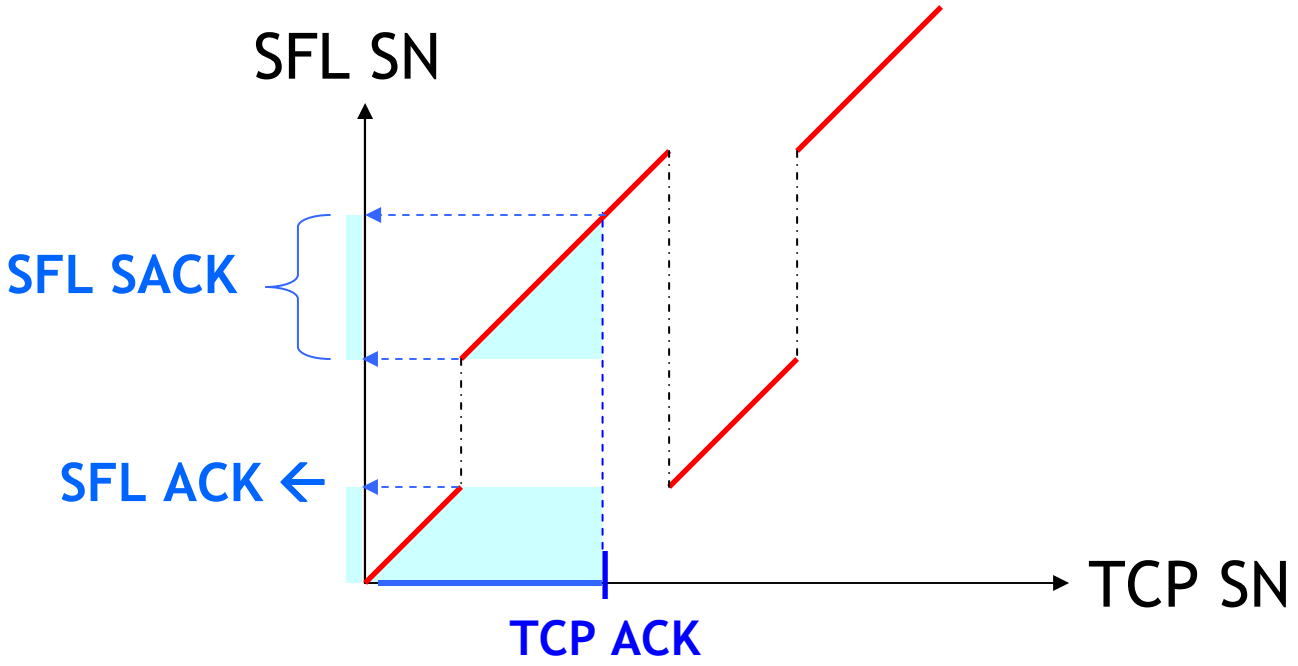
- Keep track SSN⇔DSN⇔TSN mapping history

Non-Monotonous Mappings



- TCP → SFL: Monotonize mappings. Impossible after break!
- SFL → TCP: Data ACK Nmb (DSS) may not be provided!
- There is no Data SACK! How to translate SFL SACK to TCP?

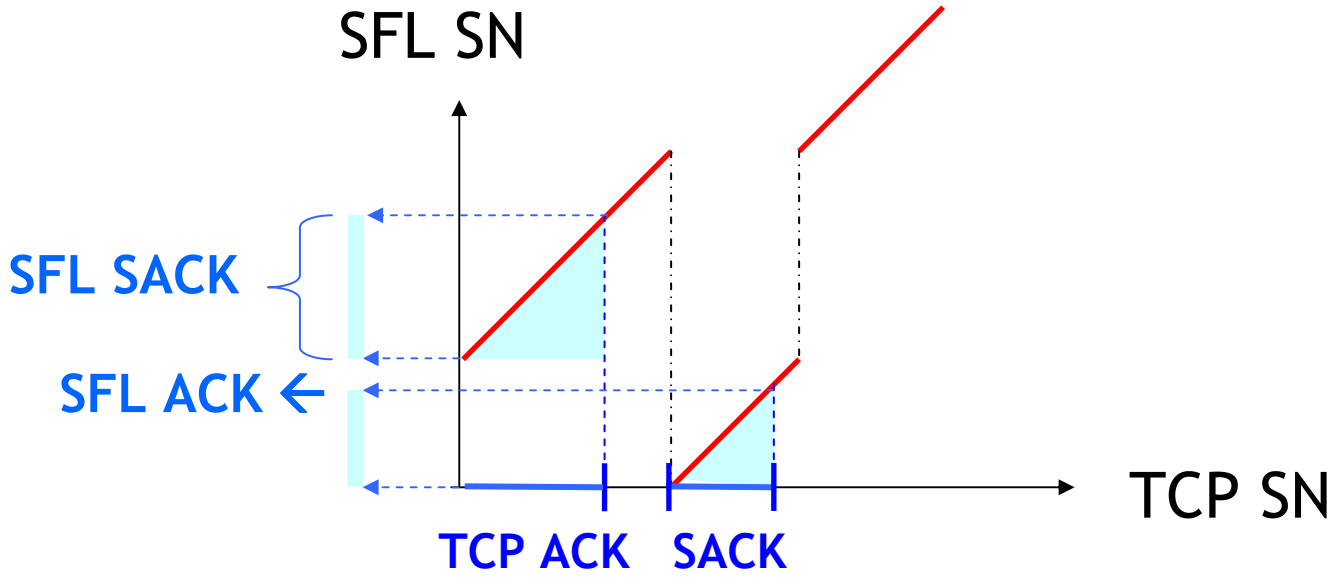
Sequence Space Projection



- Project ACKed SEQ space from TCP → SFL or SFL → TCP
- Requires inference: SFL ACK → Data ACK

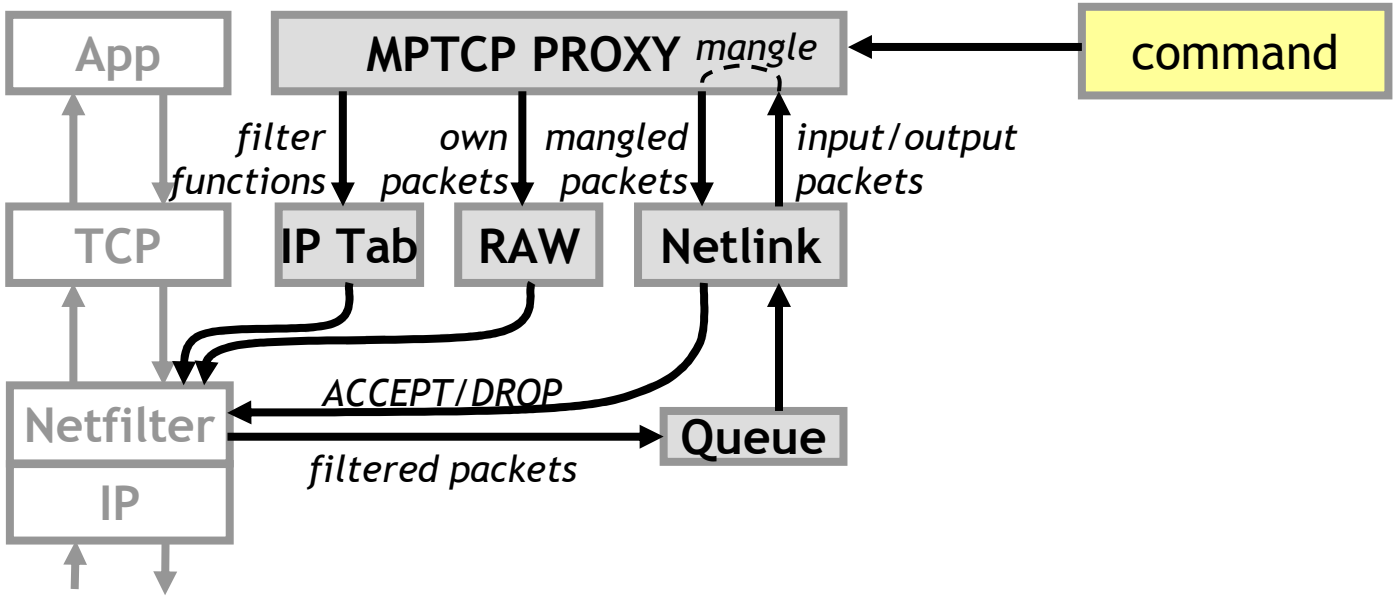
  Legitimate outside proxy multipath transmission !

Selective ACK after Break



- TCP ACK → SFL SACK Block
- TCP SACK Block → SFL ACK

MPTCP Proxy: Implementation



- Linux >= 2.6.14
- Packet filter: Netfilter + Netlink
- Prototype: Userspace, IPtables
- Command-line interface

MPTCP Proxy: Feature Support

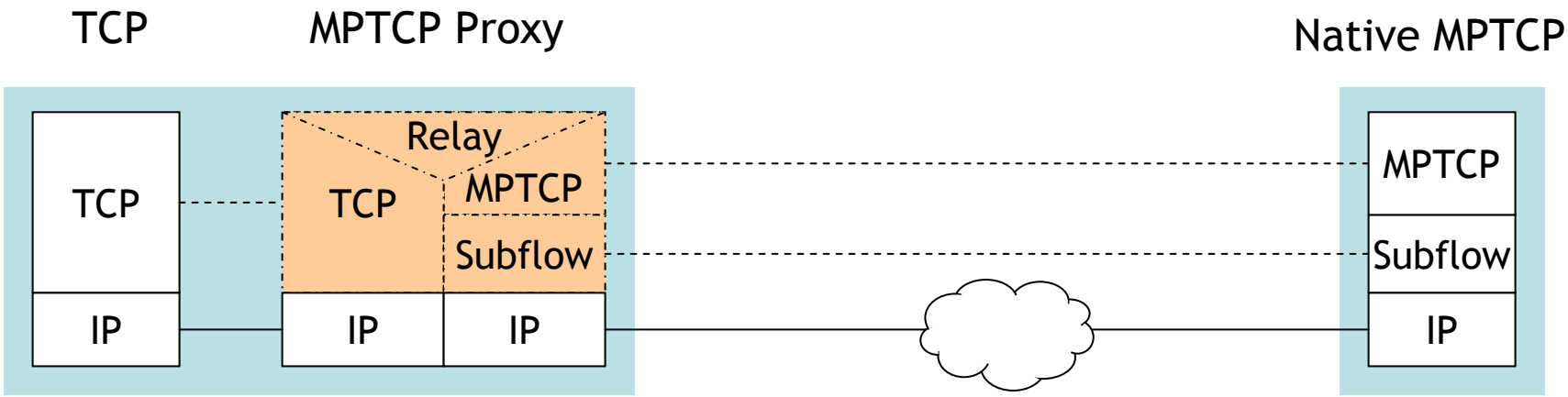
- All features of MPTCP design doc…

  Except:

  - IPv6
  - 8 octet DSN
  - ADD_ADDRESS
  - DSS Checksums
  - MP_FAIL

- Command line features:
  - Establish/tear down of subflows
  - MBB: Seamless handover between established subflows
  - BBM: Breaks subflow and establishes new subflow on new interface

Interoperability Tests



- Native MPTCP: Olivier Bonaventure's group
- Support by Christoph Paasch

Interoperability: Basic Trials

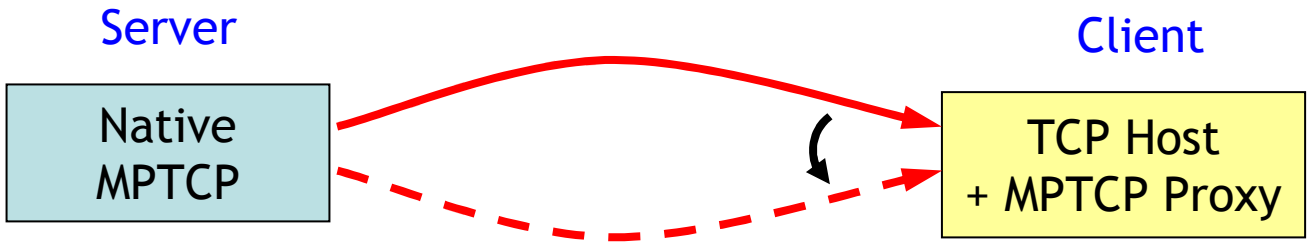
- Establishment of connections ✓
- Adding (multiple) subflows ✓
- Closing subflow(s) ✓
- Closing connections ✓
- Fast closing of connections ✓

Done for:

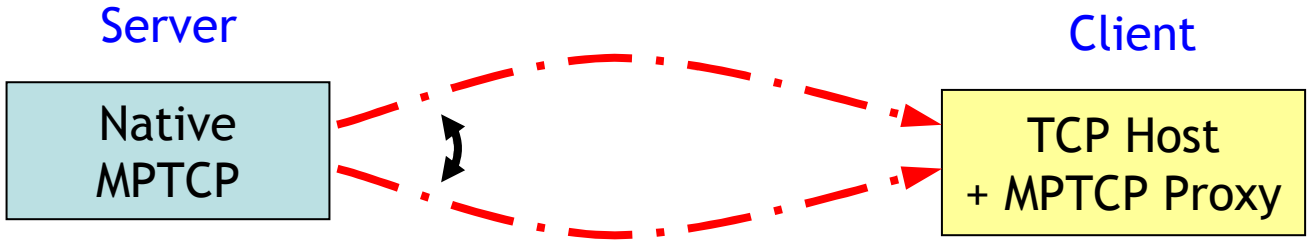
MPTCP Proxy ⇔ Native MPTCP

MPTCP Proxy ⇔ MPTCP Proxy

Interoperability: Dynamics

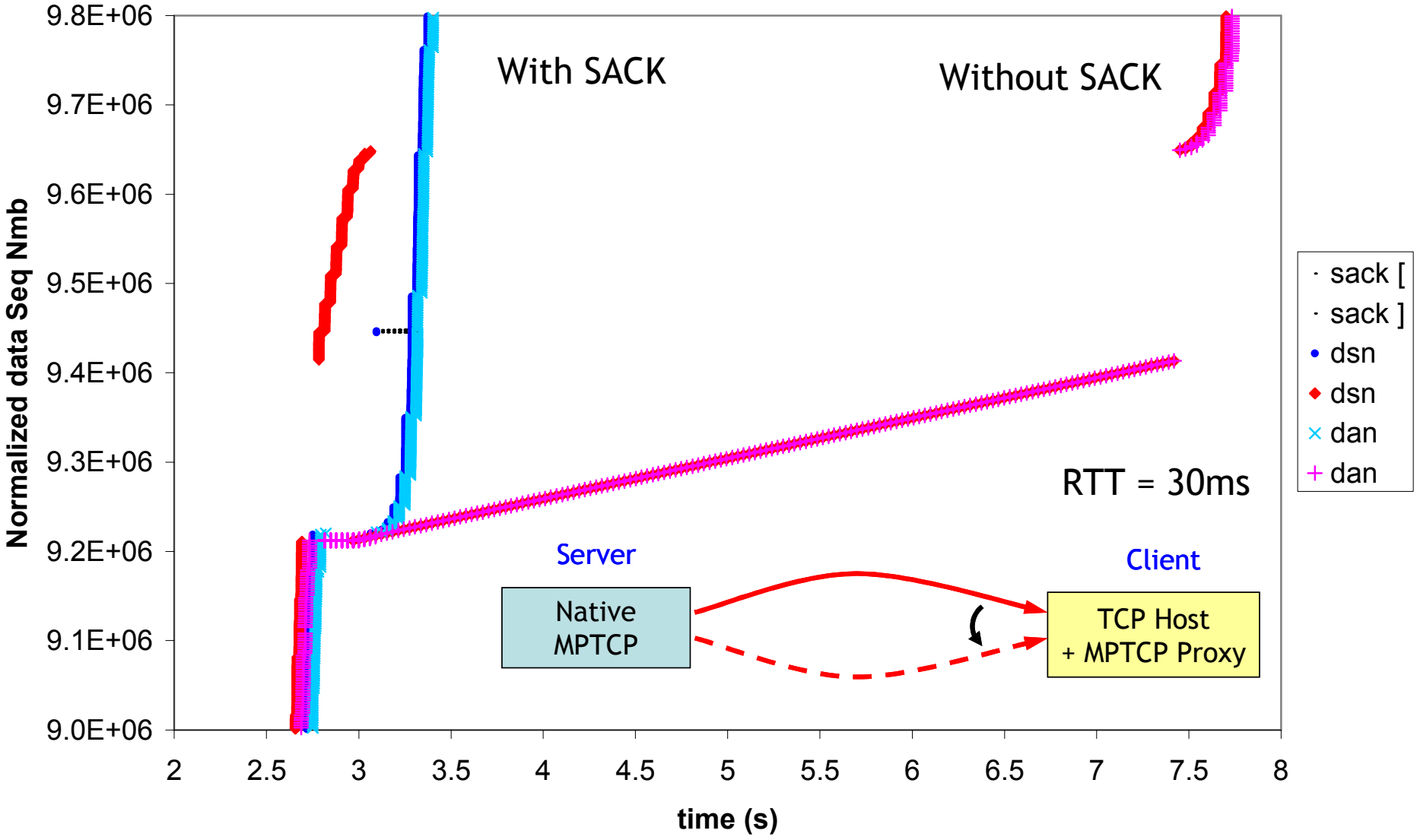


- MPTCP Proxy switches: MP_PRIO (3B, B=0) + MP_PRIO (3B, B=1) ✓
- MPTCP Proxy breaks: REMOVE_ADDR, SFL RST ✓

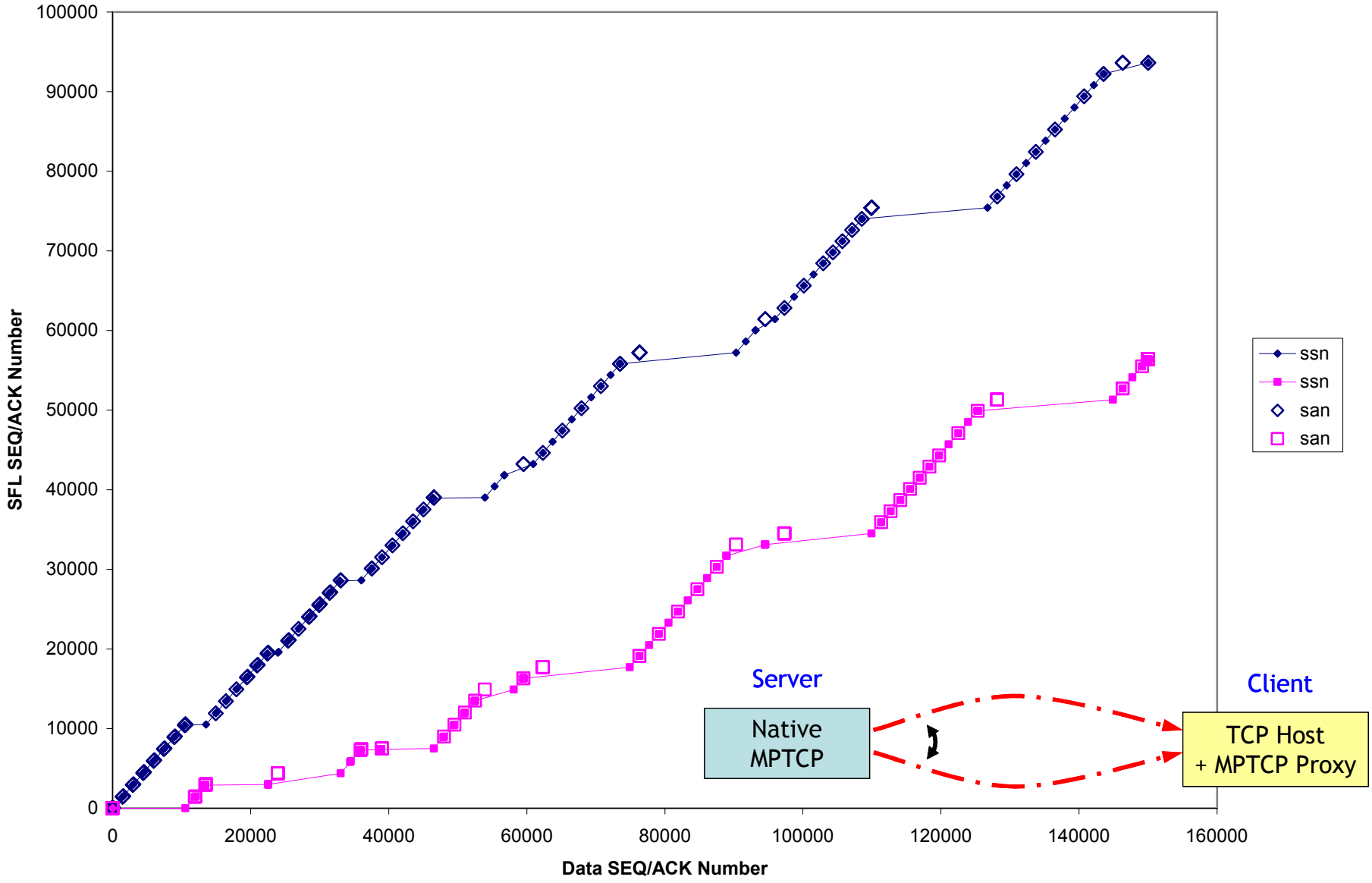


- Native MPTCP: multipath operation ✓

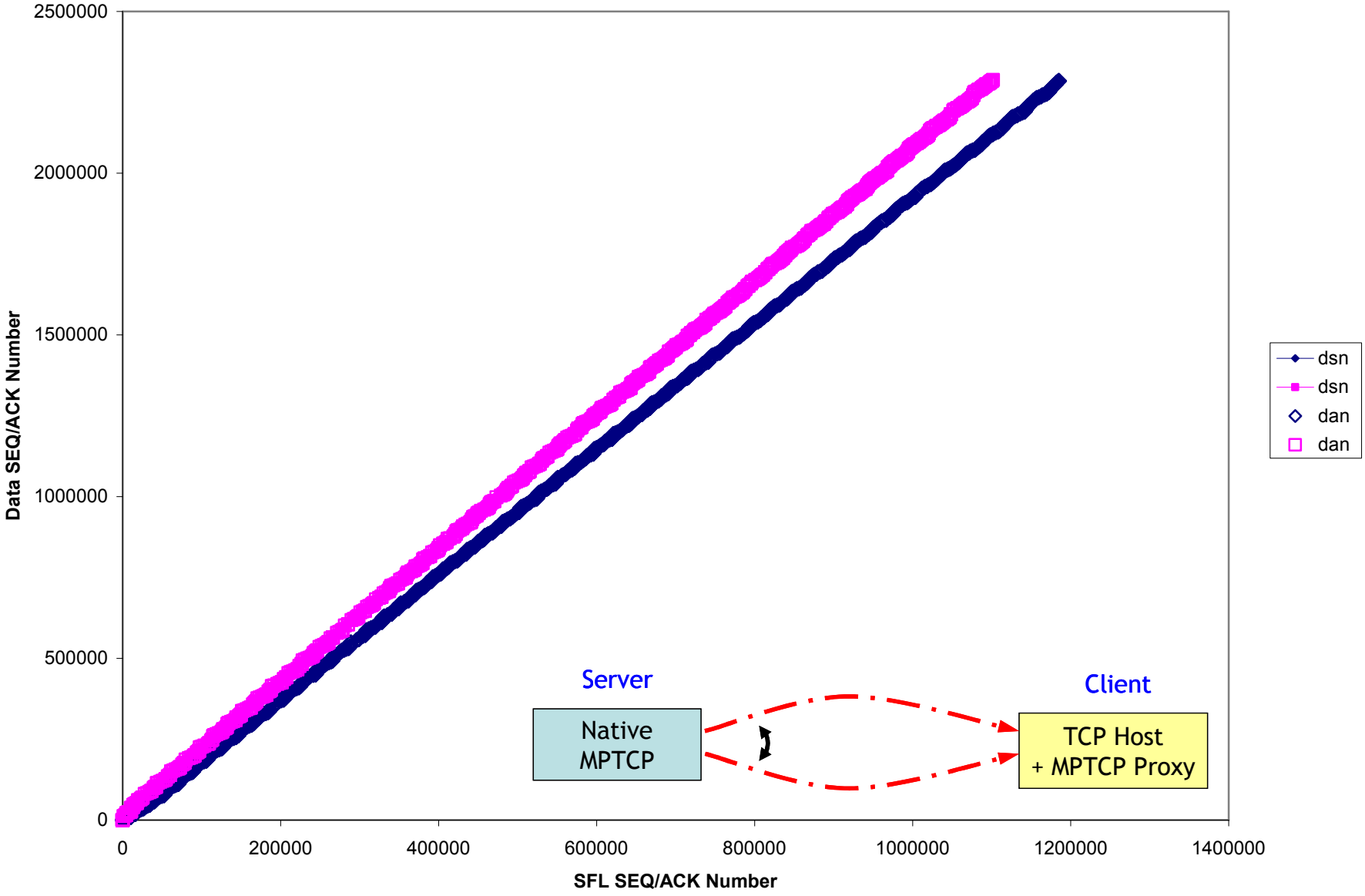
Native MPTCP + server → MPTCP Proxy + client
MPTCP Proxy Switches



Native MPTCP + server → MPTCP Proxy + client
Native MPTCP does multipath



Native MPTCP + server → MPTCP Proxy + client
Native MPTCP does multipath



Code Release

- MPTCP Proxy Vs 0.9: Released on Oct 26, 2012
- Installation & Operation Guide
- http://open-innovation.alcatel-lucent.com/projects/mptcp-proxy

Next Steps

- New signaling features for proxy support
  - Implicit proxy
  - Explicit proxy
  - Anchor features

draft_hampel_mptcp_proxies_anchors_00