Experience with Multipath TCP

Olivier Bonaventure

Christoph Paasch

Known Multipath TCP implementations

- Opensource
  - Linux kernel
    - http://www.multipath-tcp.org
  - Apple
    - http://opensource.apple.com/source/xnu/xnu-2422.1.72/bsd/netinet/mptcp.c
  - FreeBSD
    - http://caia.swin.edu.au/urp/newtcp/mptcp/tools.html
- Closed source
  - Citrix Netscaler

The feedback in this draft comes from the Linux implementation. Feedback from others is more than welcome

Users of Multipath TCP on Linux

Outline

- Middlebox interference
- Use cases
- Congestion control
- Subflow management
- Packet schedulers
- Interactions with DNS

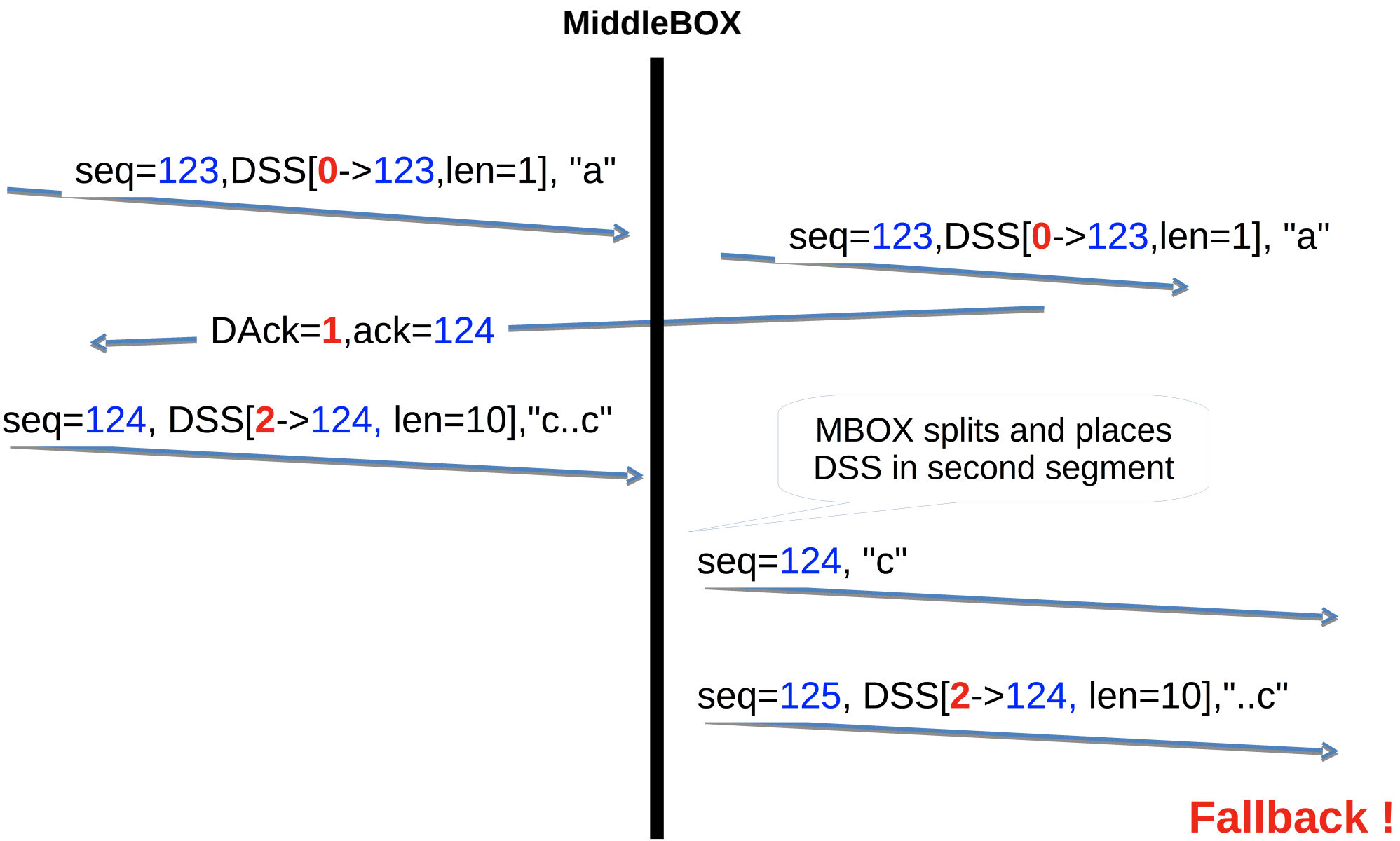
Middelbox interference

#	Middlebox-function	Successfull?	Fallback?
1	NAT	Y (Yes)	N (No)
2	Remove opt. SYN	Y	Y
3	Remove opt. Data	Y	Y
4	Sequence number rand.	Y	N
	Segment Splitting		
5	Opt. both segment	Y	N
6	Opt. first segment	Y	N
7	Opt. second segment	Y	Y
8	Opt. second segment*	N	-
9	Coalesce	Y	Y

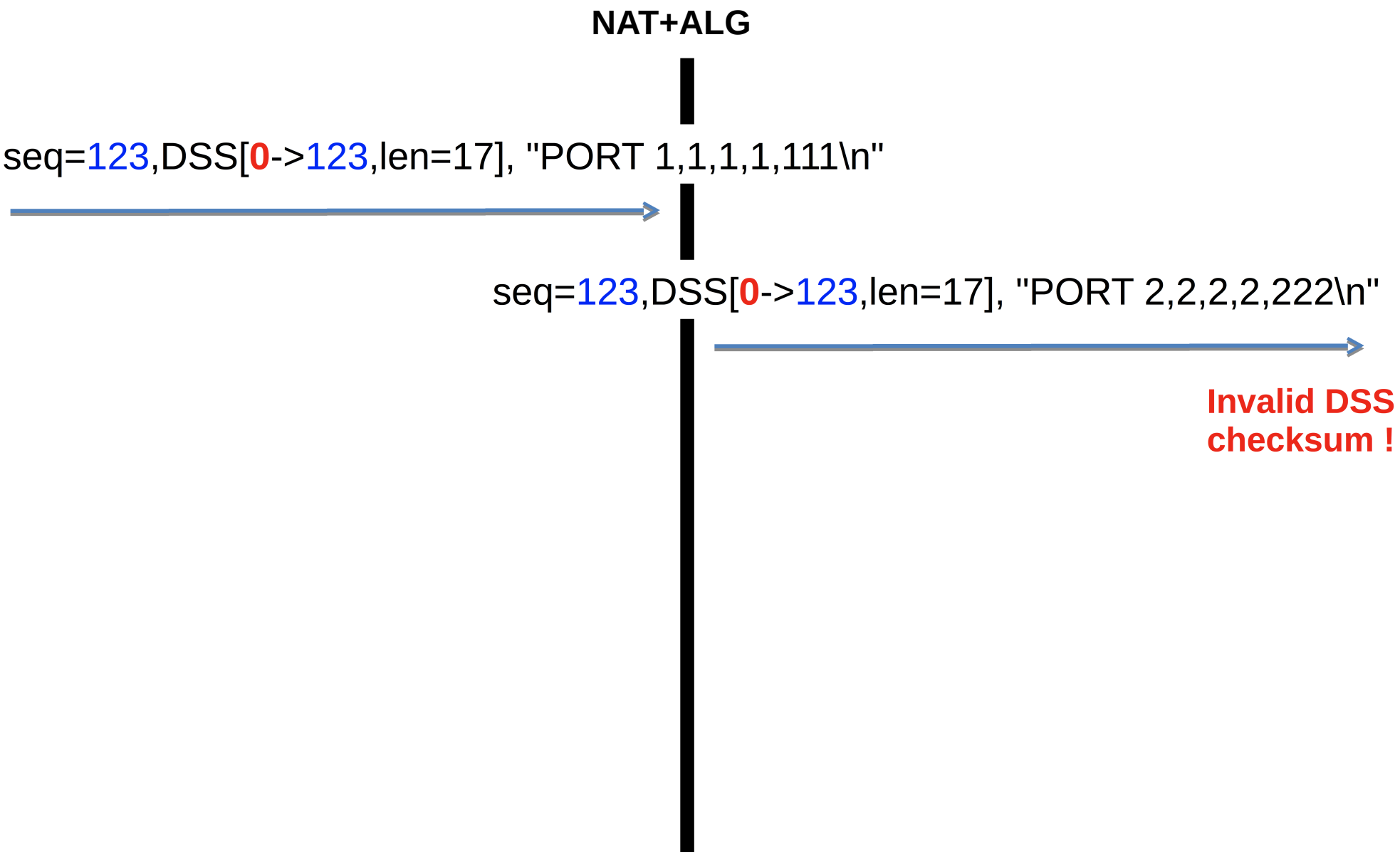
Table 1: MPTCP works across all common middleboxes, by either falling back to regular TCP, or thanks to a built-in support for this type of middlebox.

B. Hesmans, et al. , “Are TCP Extensions Middlebox-proof?. CoNEXT workshop HotMiddlebox, December 2013. ACM.

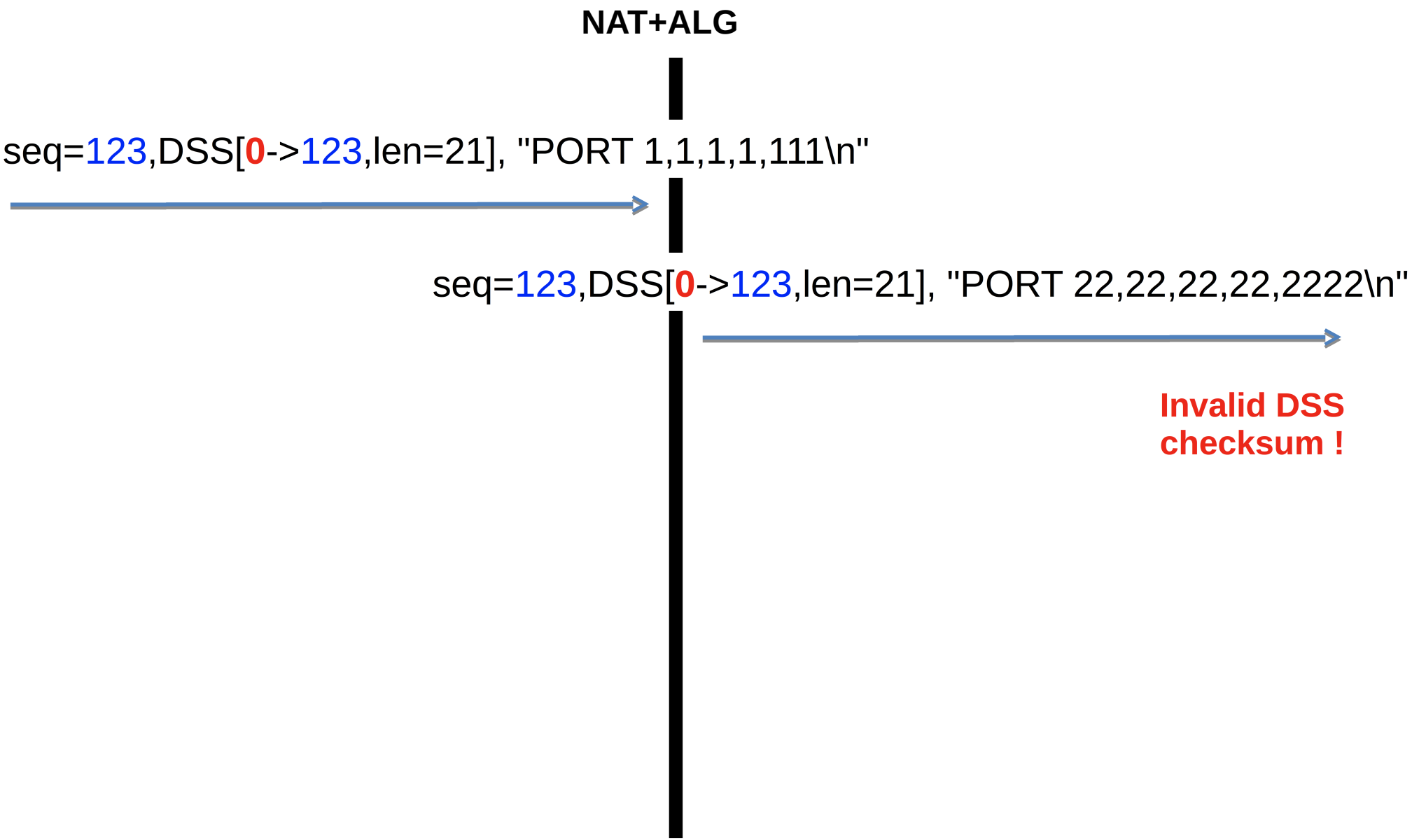
Segment splitting
Unlikely corner case



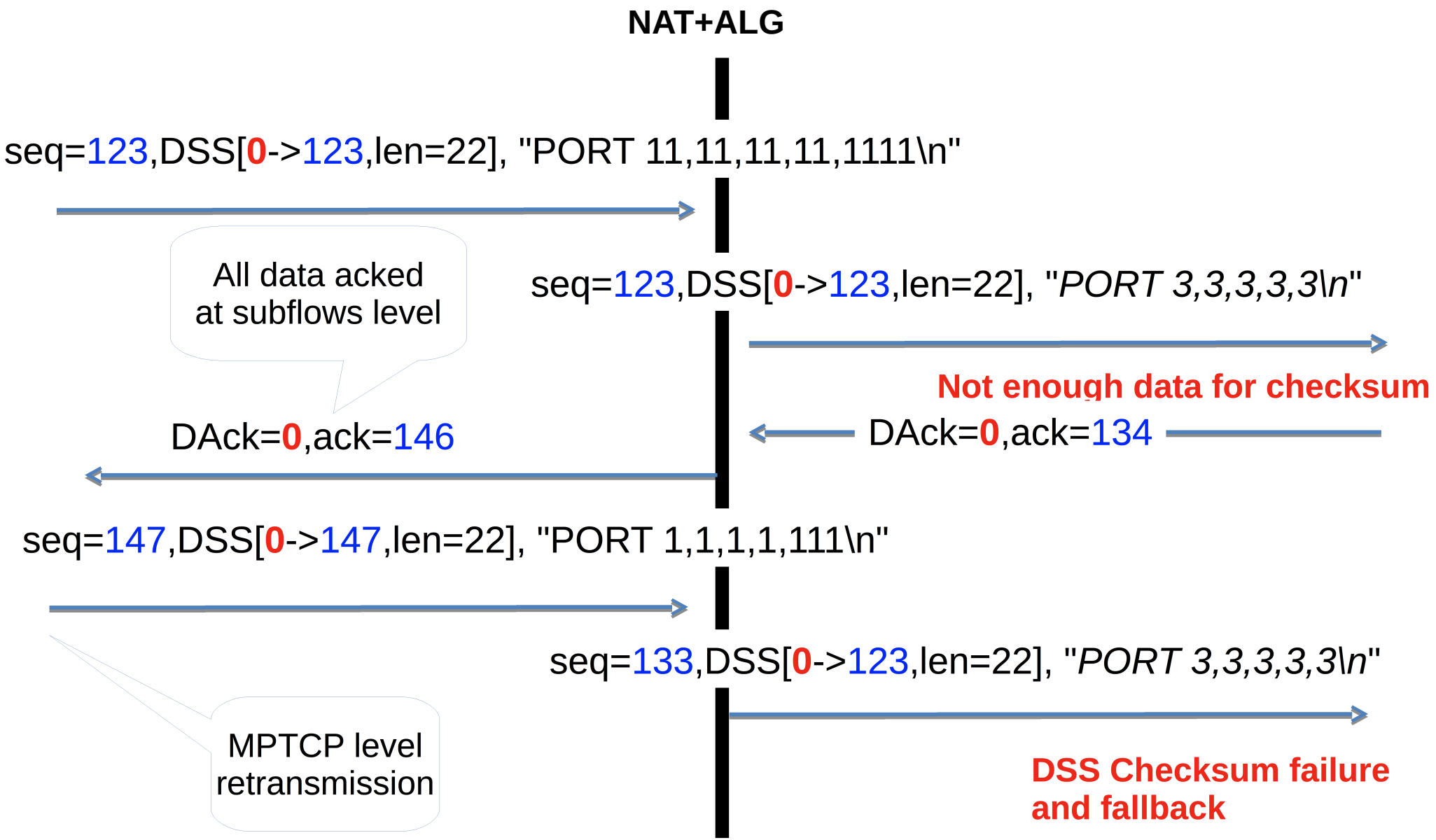
Multipath TCP and FTP ALG translated PORT same length



Multipath TCP and FTP ALG translated PORT longer



Multipath TCP and FTP ALG translated PORT shorter



Outline

- Middlebox interference
- Use cases
- Congestion control
- Subflow management
- Packet schedulers
- Interactions with DNS

Datacenter use case

- Objective
  - Enable Multipath TCP to use several load-balanced paths between a pair of single-homed hosts
    - ECMP is widely used inside datacenters
    - ECMP is also widely used in ISP networks
  - Simulations and measurements show that this approach has many performance benefits
  - RFC6824 assumes *that one or both hosts are multihomed and multiaddressed*
  - **Experience shows that Multipath TCP should not be restricted to multihomed/multiaddressed hosts**

Multipath TCP in datacenters

- The ndiffport path manager
  - Implemented in the Linux kernel
- Operation
  - N subflows, differing by their source port on the client side are established for each Multipath TCP connection
    - Number of subflows is currently static, it would be better to have some interactions with networkor server to determine the number of subflows to use to reach a given destination

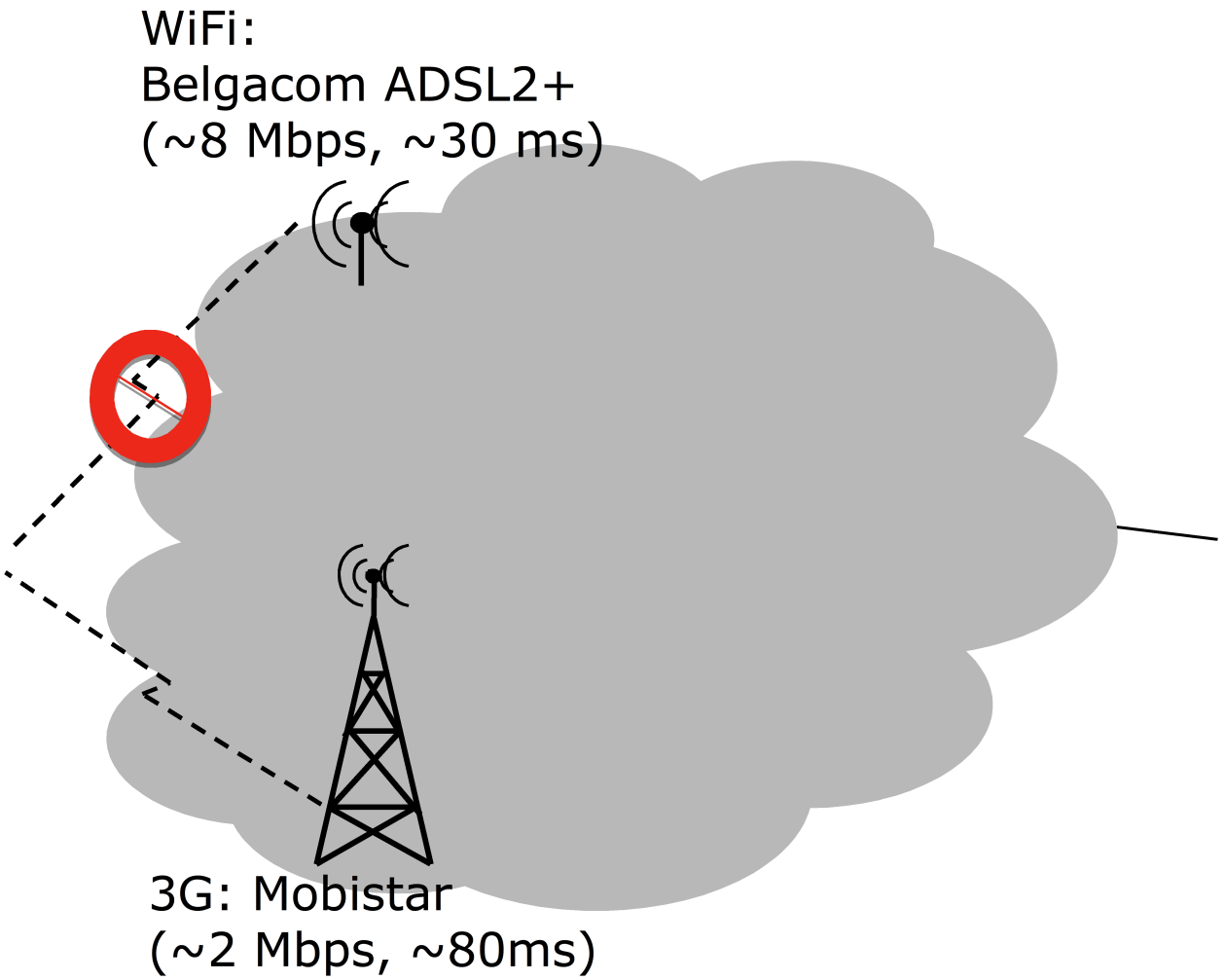
Multipath TCP on mobile devices

- Multipath TCP is used by SIRI application on ios7 devices
  - Unfortunately, no operational feedback has been received about this large deployment
- Multipath TCP has been ported to Android smartphones and used for some experiments
  - Contact us is you have experience with these smartphones

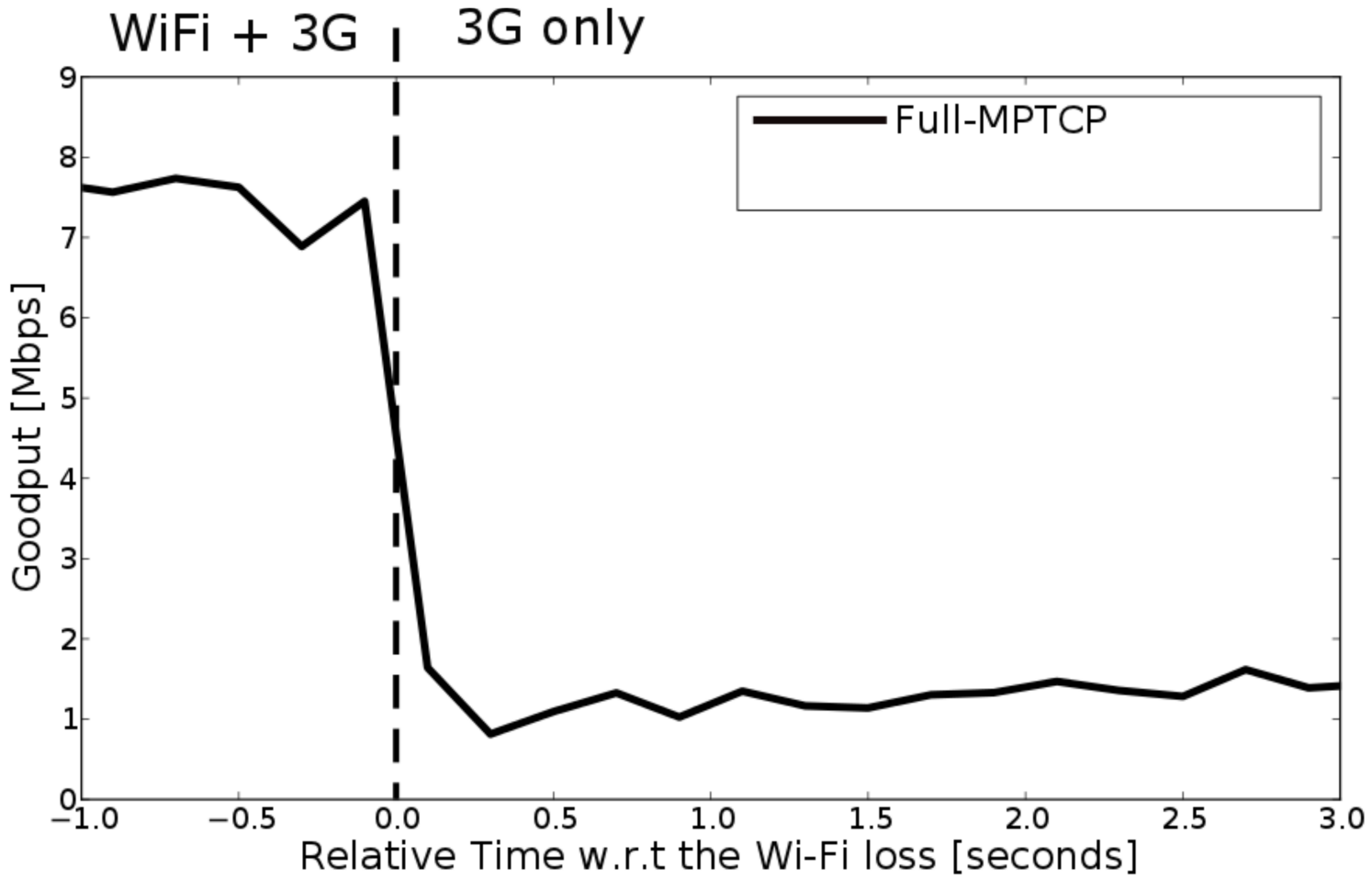
Handover modes

- Defines how xG and WiFi interfaces are used
- Full
  - Both interfaces used at the same time
- Backup-mode
  - Subflows are created on both interfaces, but data only flows on one of them
- Single-path mode
  - Only one interface is used at a time. Subflows are not preestablished over the backup interface

Evaluation scenario

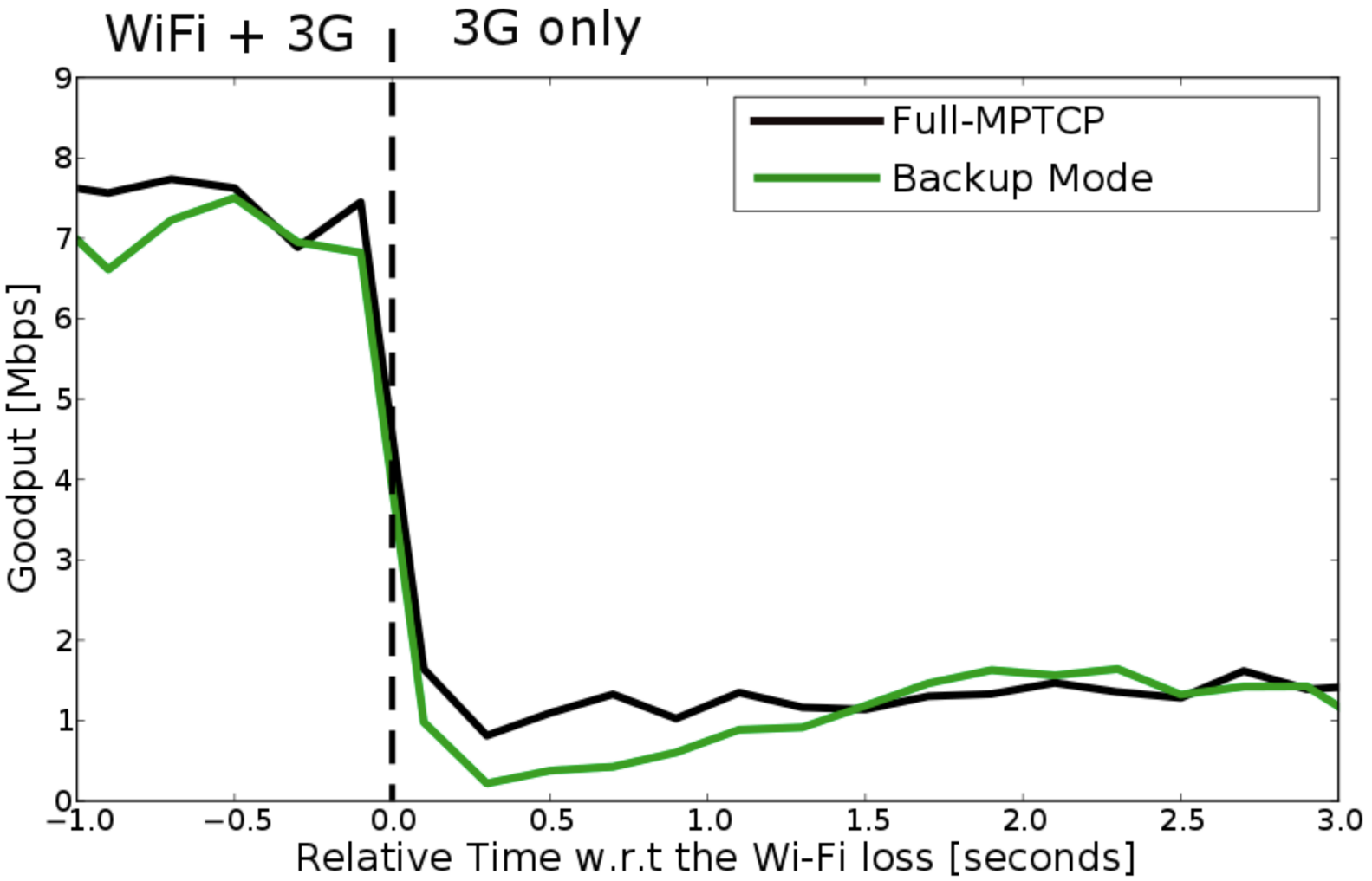


Recovery after failure



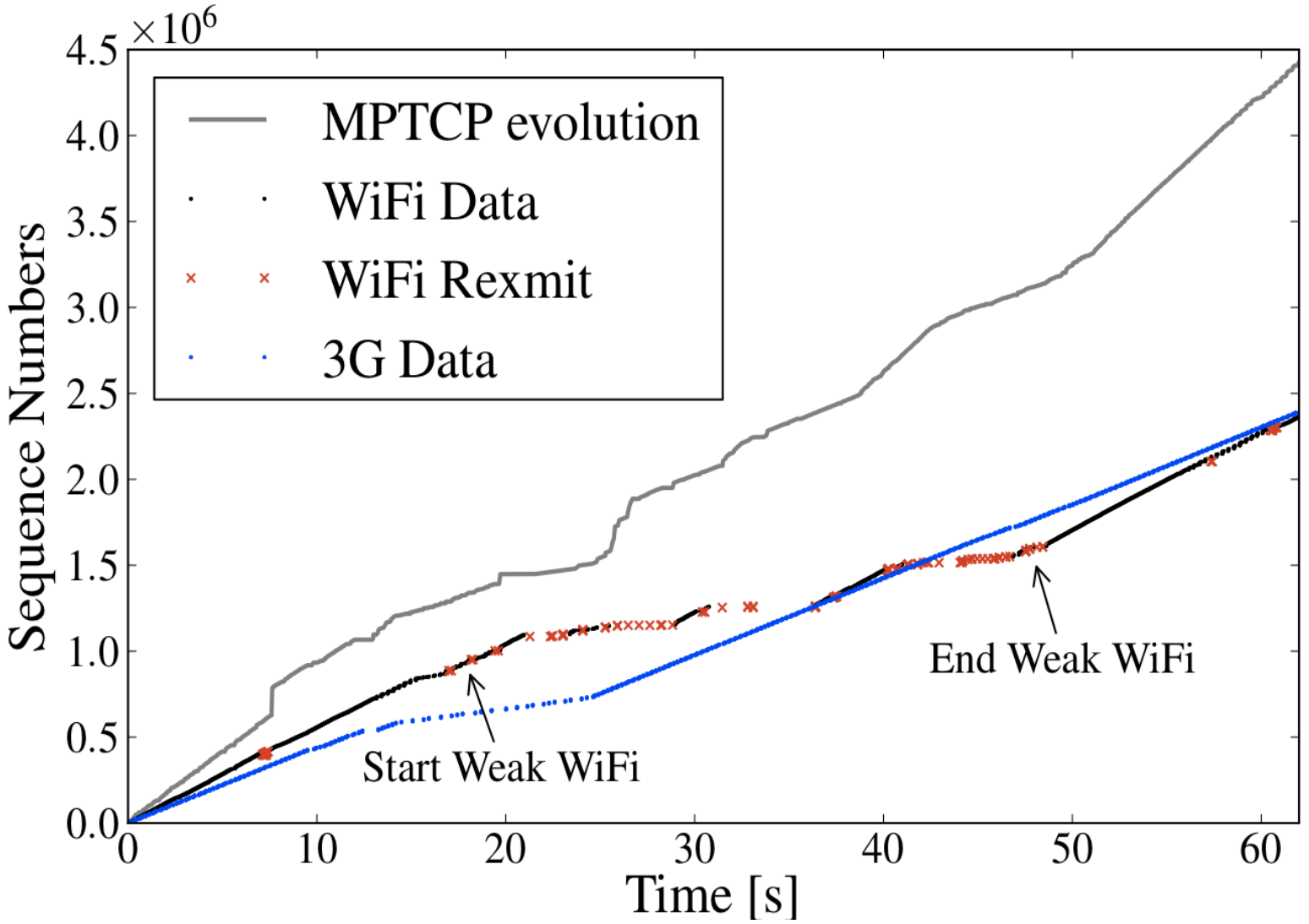
C. Paasch, et al. , “Exploring mobile/WiFi handover with multipath TCP,” presented at the CellNet '12: Proceedings of the 2012 ACM SIGCOMM workshop on Cellular networks: operations, challenges, and future design, 2012.

Recovery after failure



C. Paasch, et al. , “Exploring mobile/WiFi handover with multipath TCP,” presented at the CellNet '12: Proceedings of the 2012 ACM SIGCOMM workshop on Cellular networks: operations, challenges, and future design, 2012.

3G/WiFi handover

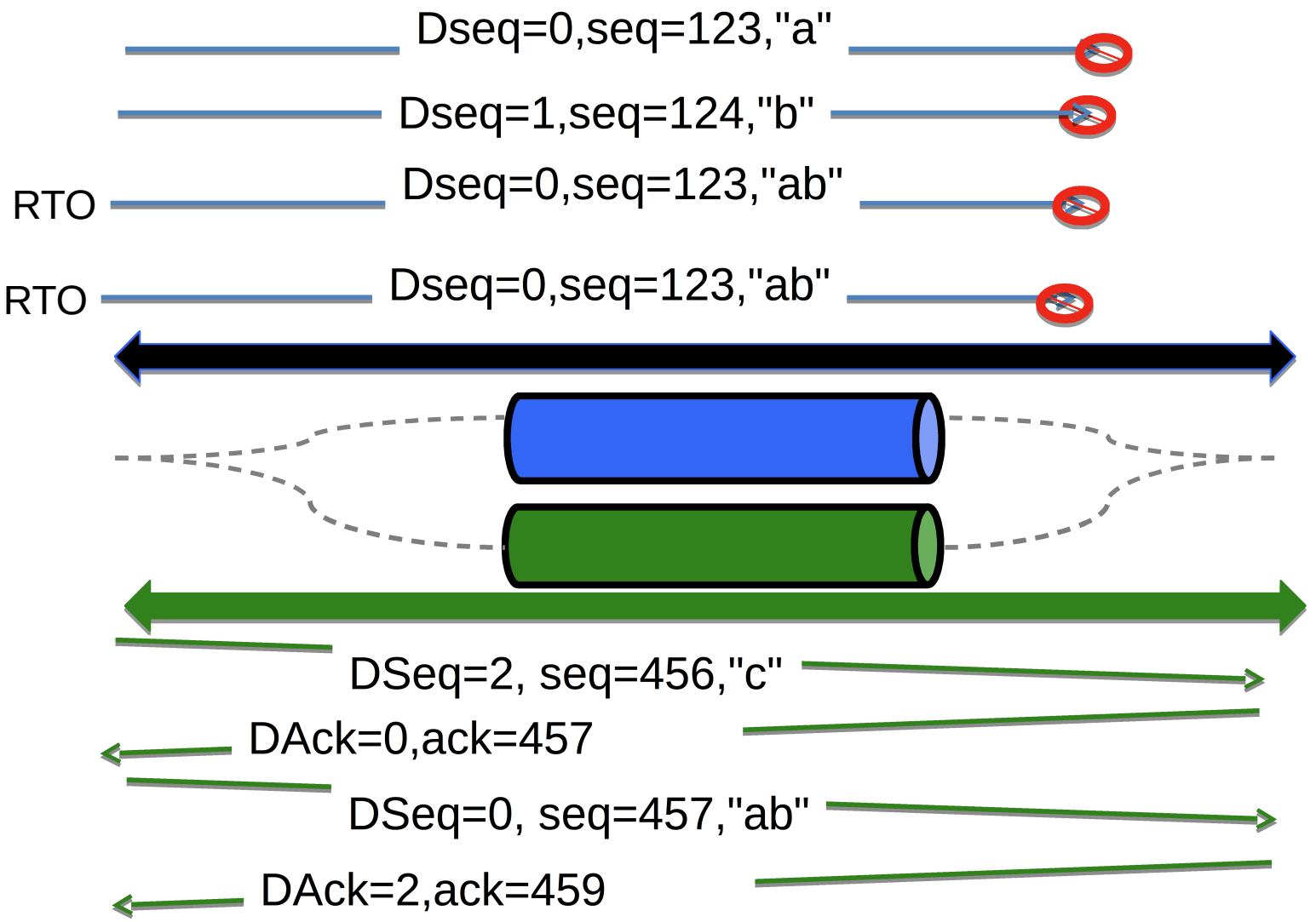


C. Paasch, O. Bonaventure, “Multipath TCP”, Communications of the ACM, Aplri 2014
http://queue.acm.org/detail.cfm?id=2591369

Issues with mobile use case

- Reliability of the RM_ADDR option
  - A loss of the RM_ADDR option that follows a handover has a negative impact on performance since MPTCP continues to use bad subflow
- Transmission of data over lossy links like poor SNR or congested WiFI
  - Can negatively impact MPTCP performance

The problem with lossy links



Towards a solution

- Allow a Multipath TCP host to terminate a subflow (with a RST) if
  - Too many data have been retransmitted unsucessfully
  - Data transmitted over this subflow has already been acked (at MPTCP level) on another subflow
    - These retransmissions are only useful to cope with middleboxes that require in-sequence data

RFC6824, 3.3.6, "*However, additional research is required to understand the heuristics of how and when to reset underperforming subflows*."

Outline

- Middlebox interference
- Use cases
- Congestion control
- Subflow management
- Packet schedulers
- Interactions with DNS

Congestion control

- Multipath TCP has triggered several new congestion control schemes
  - RFC6356
  - OLIA
  - Delay-based congestion control
  - …
- The chosen congestion control scheme has clearly an impact on the performance of Multipath TCP, but IETF is unlikely to standadize multiple congestion control schemes

Outline

- Middlebox interference
- Use cases
- Congestion control
- Subflow management
- Packet schedulers
- Interactions with DNS

Subflow management

- Which host may create the subflows ?
  - According to RFC6824, both hosts are equivalent and any host can create subflows
  - On existing implementations, only the client creates the subflows
    - Main motivation is that client is often behind a NAT/firewall that will block subflows established by servers

Subflow managers

- Current subflow managers in Linux
  - “Fullmesh”
    - Client creates a subflow from each IP address of the client to each IP address of the server
      - Works well when server is single-homed and client multi-homed
      - Does not always work well when client and servers are multihomed
        - results in Nc x Ns subflow if client has Nc addresses and server Ns
  - Ndiffports
    - Assumes that client and servers are single-homed
    - Client creates N subflows with different source ports to server
      - Difficult to know how many subflows should be used without network knowledge
      - With ECMP, selecting the source port that results in different paths is not simple

The destination port

- Which destination port should be used for the second subflow ?
  - Same as the initial subflow
    - All subflows of a Multipath TCP connection have same destination port
    - Best approach for firewalls, and servers
    - Should become a MUST in RFC when server does not advertise a different port with the ADD_ADDR option
  - Another destination port than the initial subflow
    - Requested by some users to circumvent some traffic shaping middleboxes
    - AFAIK not yet implemented
    - Should only be used if server advertises a different port number in ADD_ADDR

Subflow policies

- How to specify the subflow management policies ?
  - Currently coded in the MPTCP implementation
  - In the future, what kind of interface would sysadmin and developers expect to manage subflows ?
    - An API with socket options ?
    - Configuration files ?
    - Information from an SDN controller ?
    - Should the IETF standardize such an interface ?

Closing MPTCP and subflows

- Applications try to avoid applications try to avoid Time-Wait state by deferring the closure of the connection until the peer has sent a FIN
- Should work with MPTCP as well
  - Application should only do passive close after reception of subflow FIN and not after reception of DATA_FIN
    - Assumes that server will do FIN on all subflows after having issued DATA_FIN

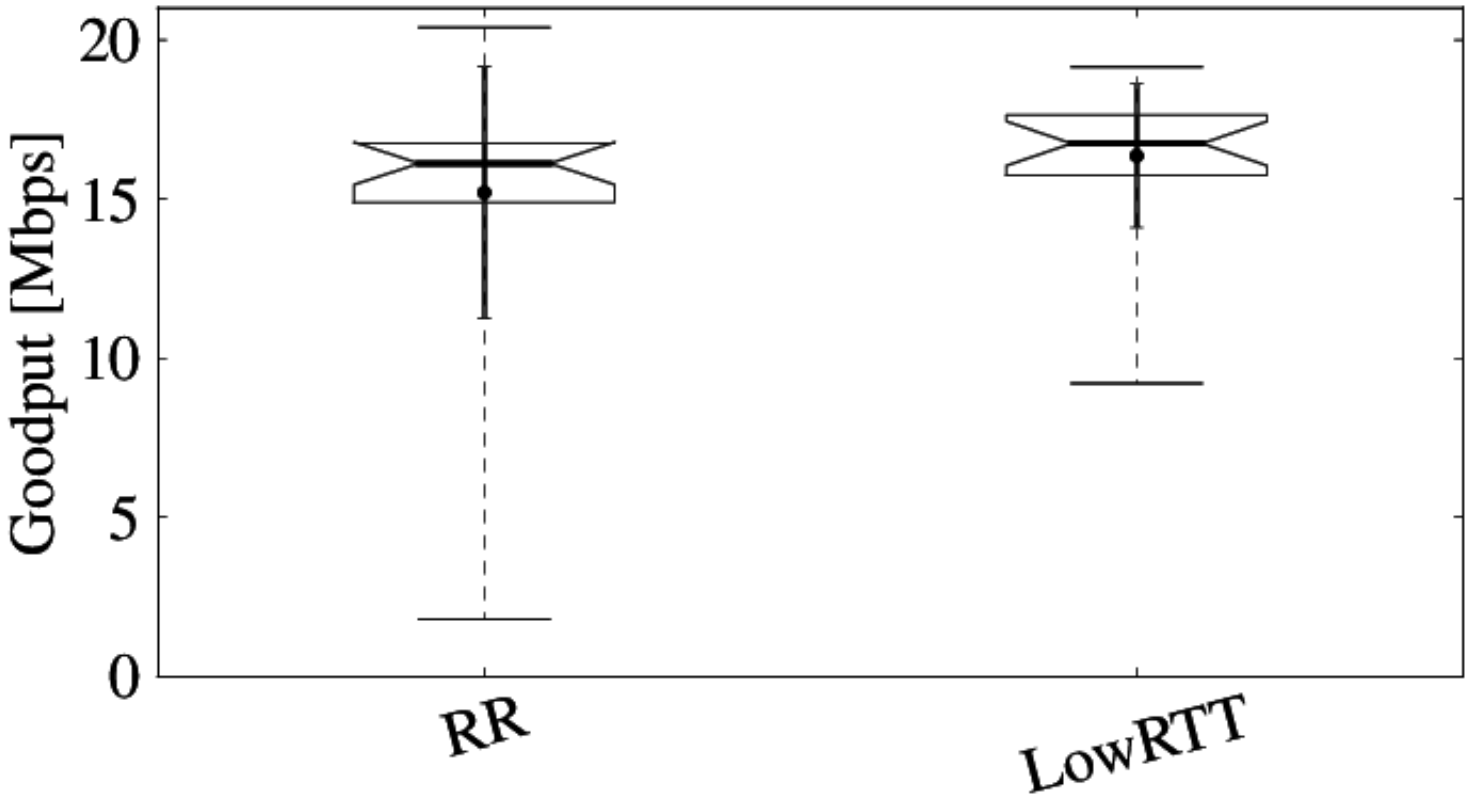
Outline

- Middlebox interference
- Use cases
- Congestion control
- Subflow management
- Packet schedulers
- Interactions with DNS

Multipath TCP schedulers

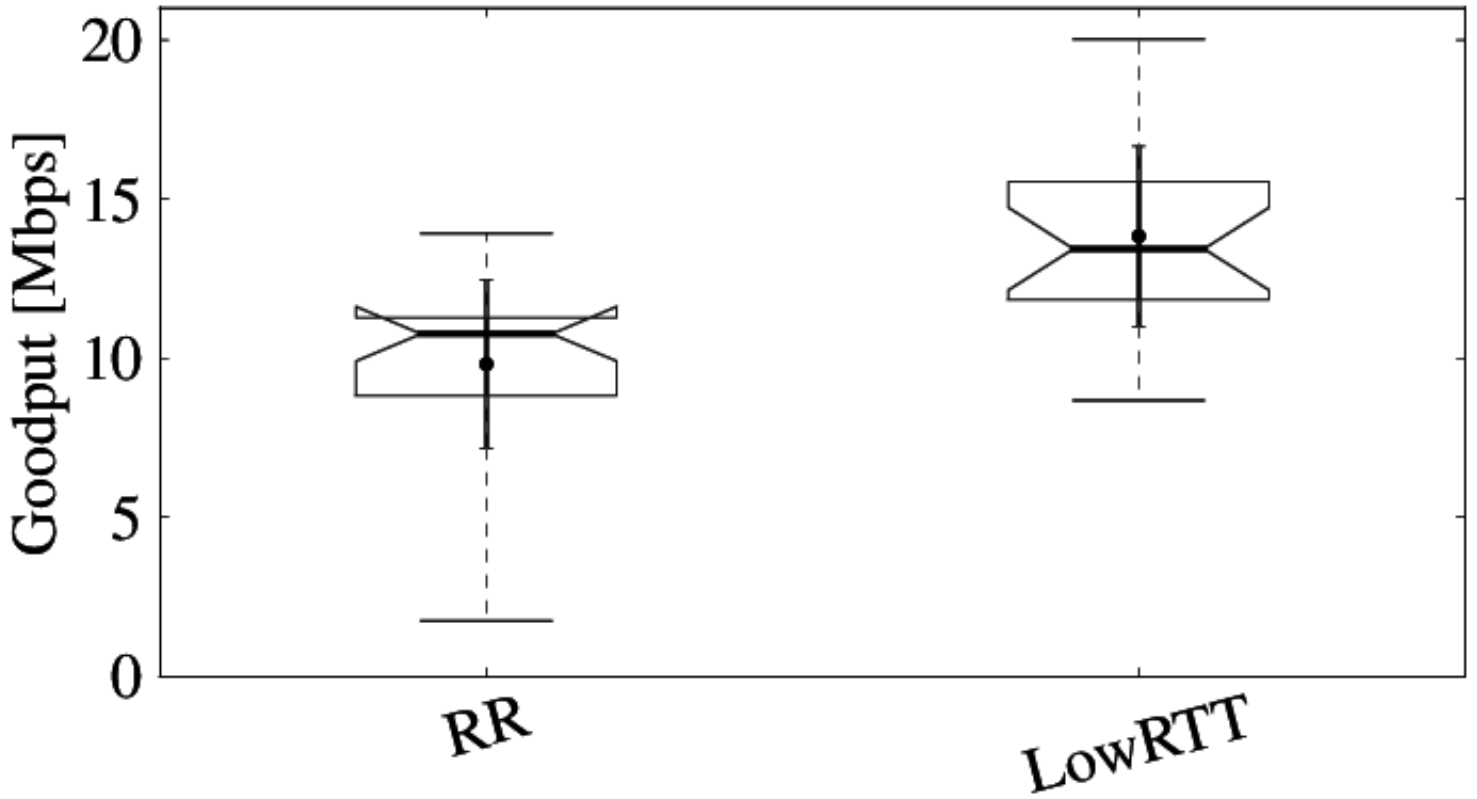
- How to schedule packets over different subflows ?
- Modular scheduler framework in Linux
  - Lowest rtt first
    - Subflow with lowest rtt is usually the one with the highest performance
  - Round-Robin

Measurements on Nornet
(16MB receive buffer)



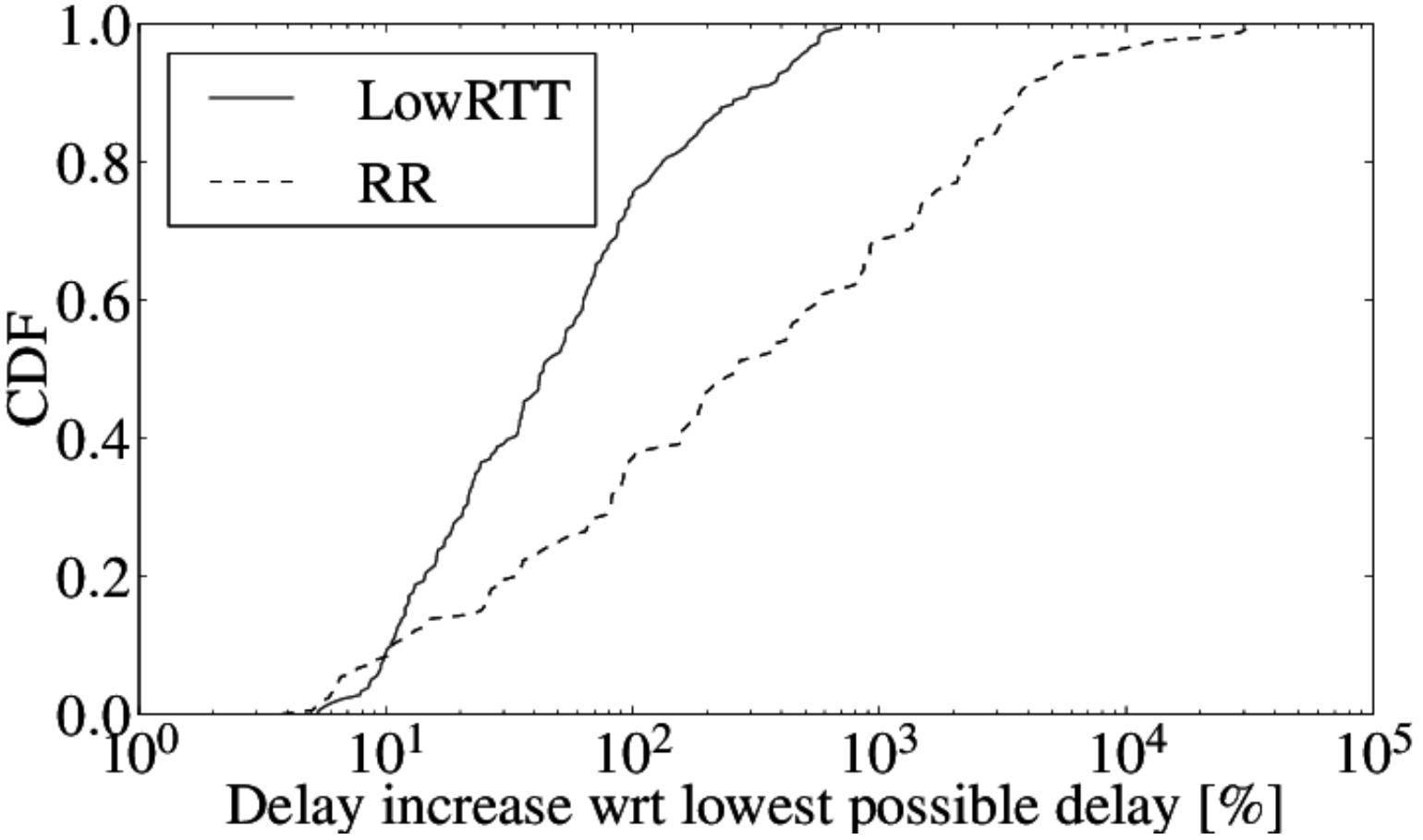
Christoph Paasch, Simone Ferlin, Ozgu Alay and Olivier Bonaventure. Experimental Evaluation of Multipath TCP Schedulers. ACM SIGCOMM Capacity Sharing Workshop (CSWS), 2014. ACM.

Measurements on Nornet
(2MB receive buffer)



Christoph Paasch, Simone Ferlin, Ozgu Alay and Olivier Bonaventure. Experimental Evaluation of Multipath TCP Schedulers. ACM SIGCOMM Capacity Sharing Workshop (CSWS), 2014. ACM.

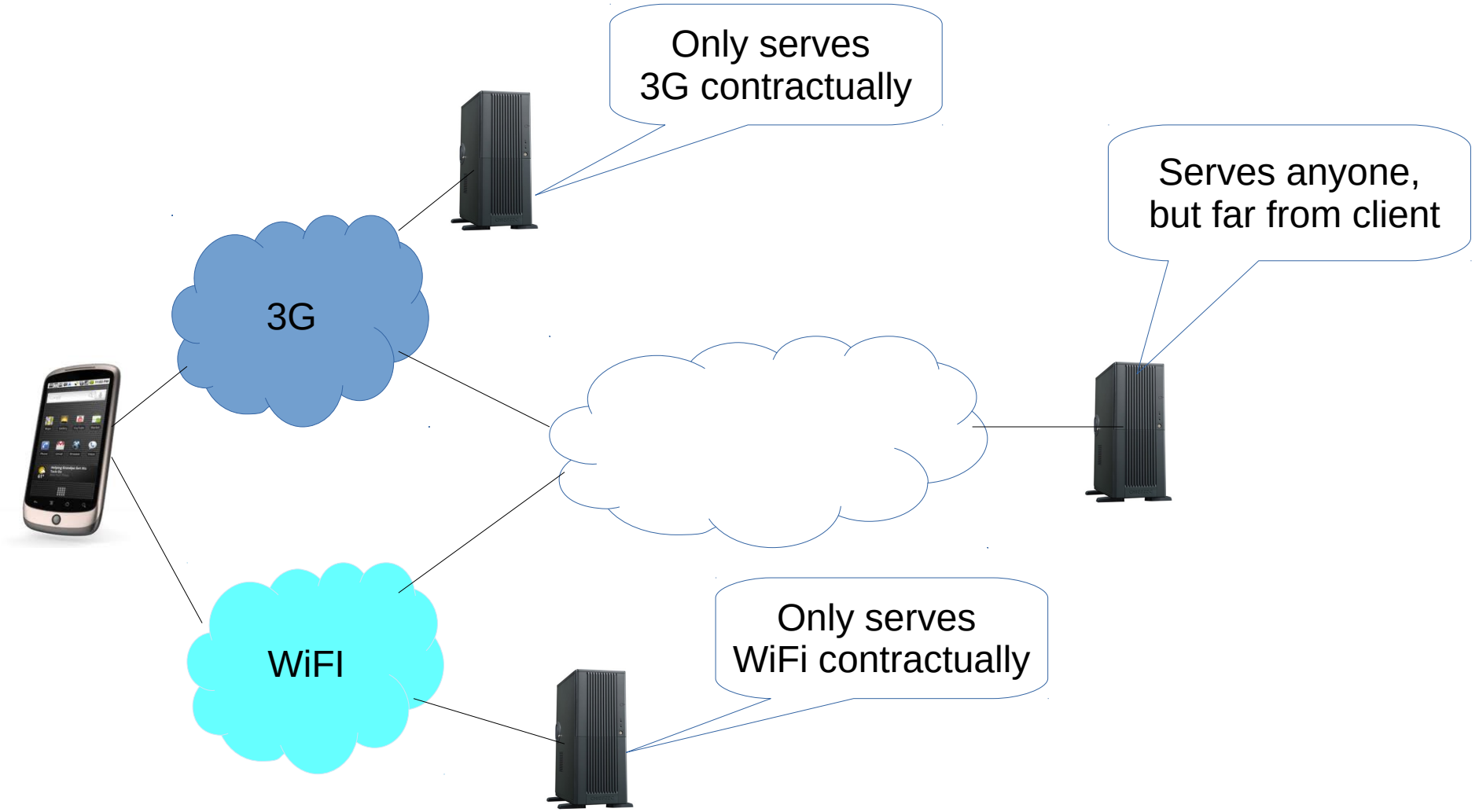
How does scheduler impacts e2e delay ?



Outline

- Middlebox interference
- Use cases
- Congestion control
- Subflow management
- Packet schedulers
- Interactions with DNS

CDNs and Multipath TCP



Conclusion

- Multipath TCP is being used
  - SIRI on iOS devices
  - Multipath TCP on Linux hosts
- Operational experience shows that the protocol specified in RFC6824 is already solid
  - Performance gains remain possible
    - Packet scheduling, path management
  - Should IETF specify subflow management