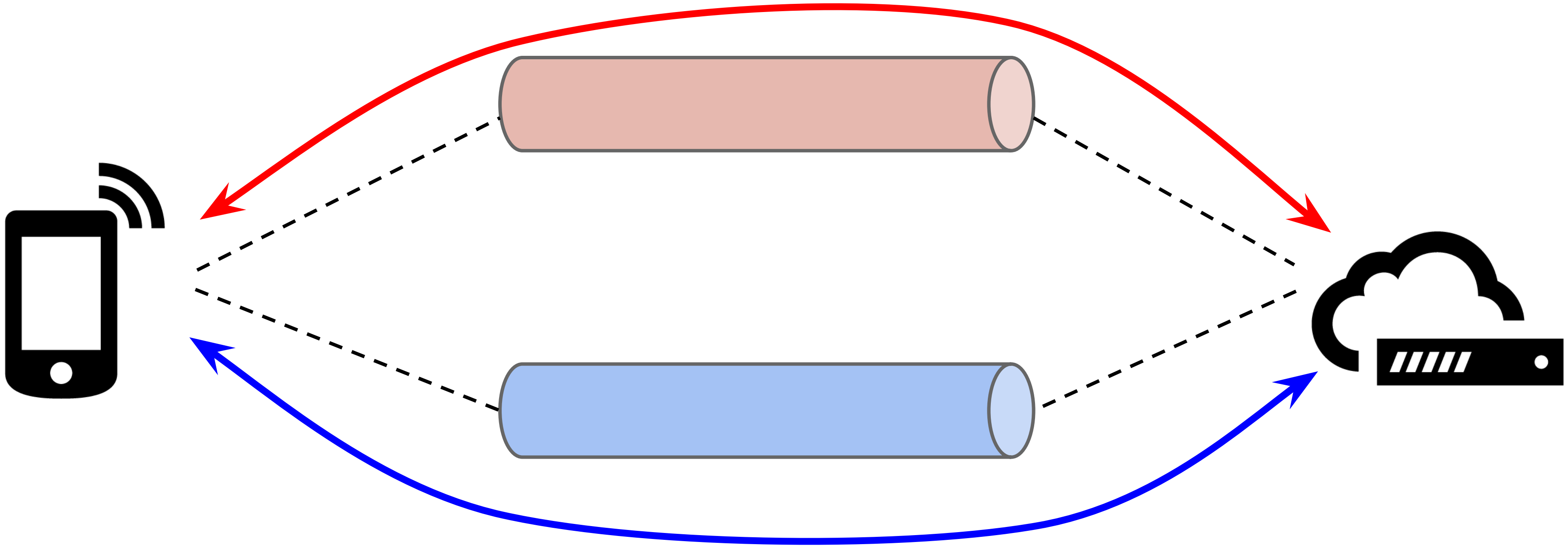
Experimental Evaluation of Multipath TCP Schedulers

Christoph Paasch[1], Simone Ferlin[2],
Özgü Alay[2] and Olivier Bonaventure[1]

[1]ICTEAM, UCLouvain, Belgium
[2]Simula Research Laboratory, Fornebu, Norway

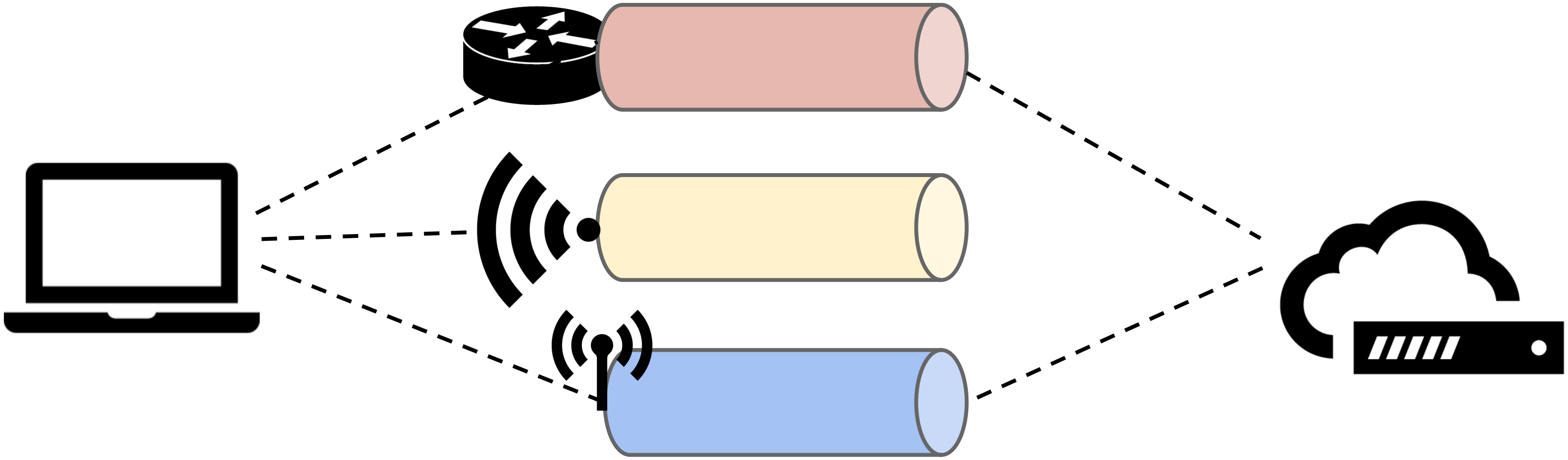
Multipath TCP



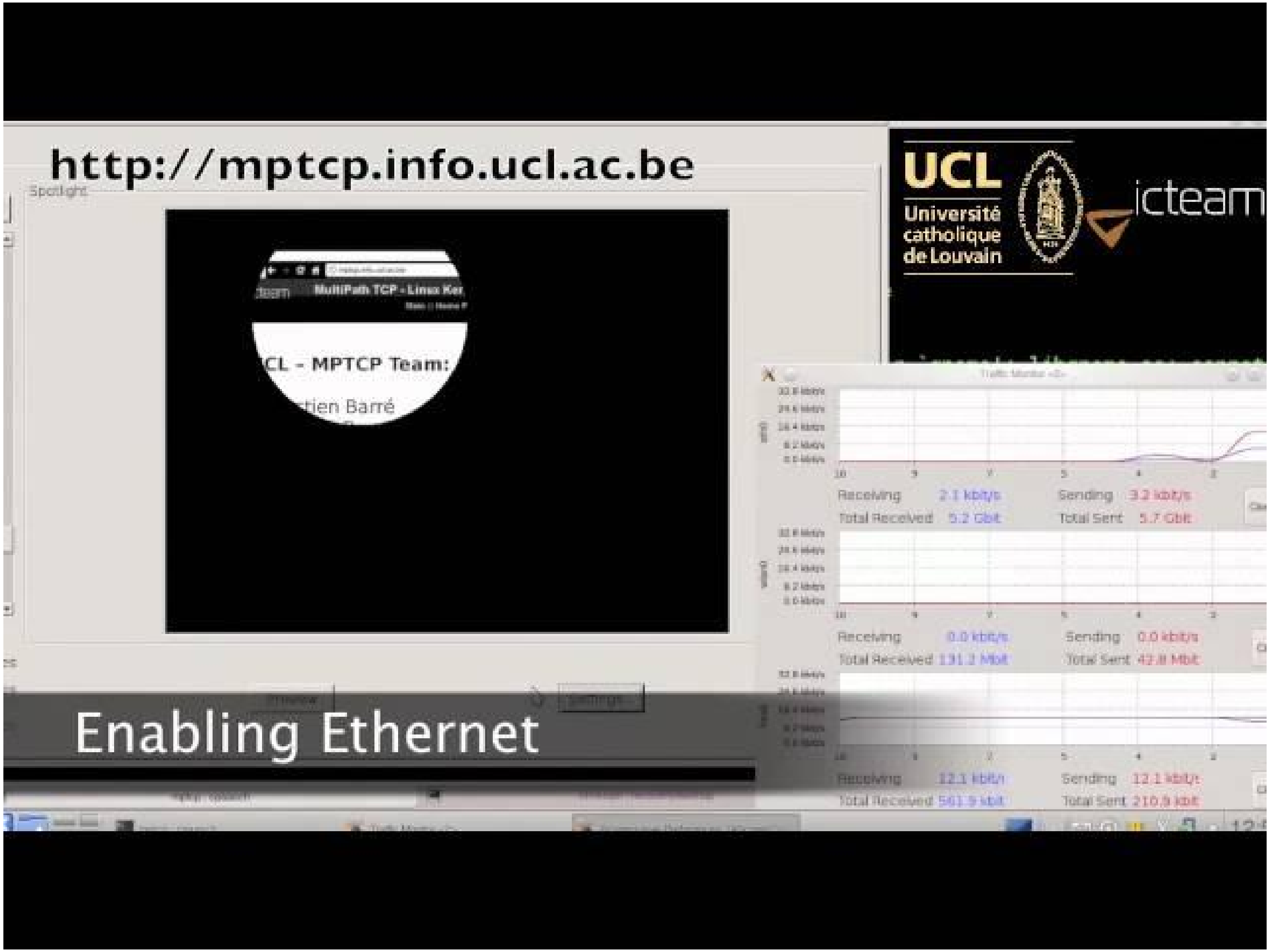
Resource pooling

Increased resilience to failures

Multipath TCP - Example



Multipath TCP - Example



http://mptcp.info.ucl.ac.be

UCL Université catholique de Louvain

icteam

MultiPath TCP - Linux Ker

CL - MPTCP Team:

stien Barré

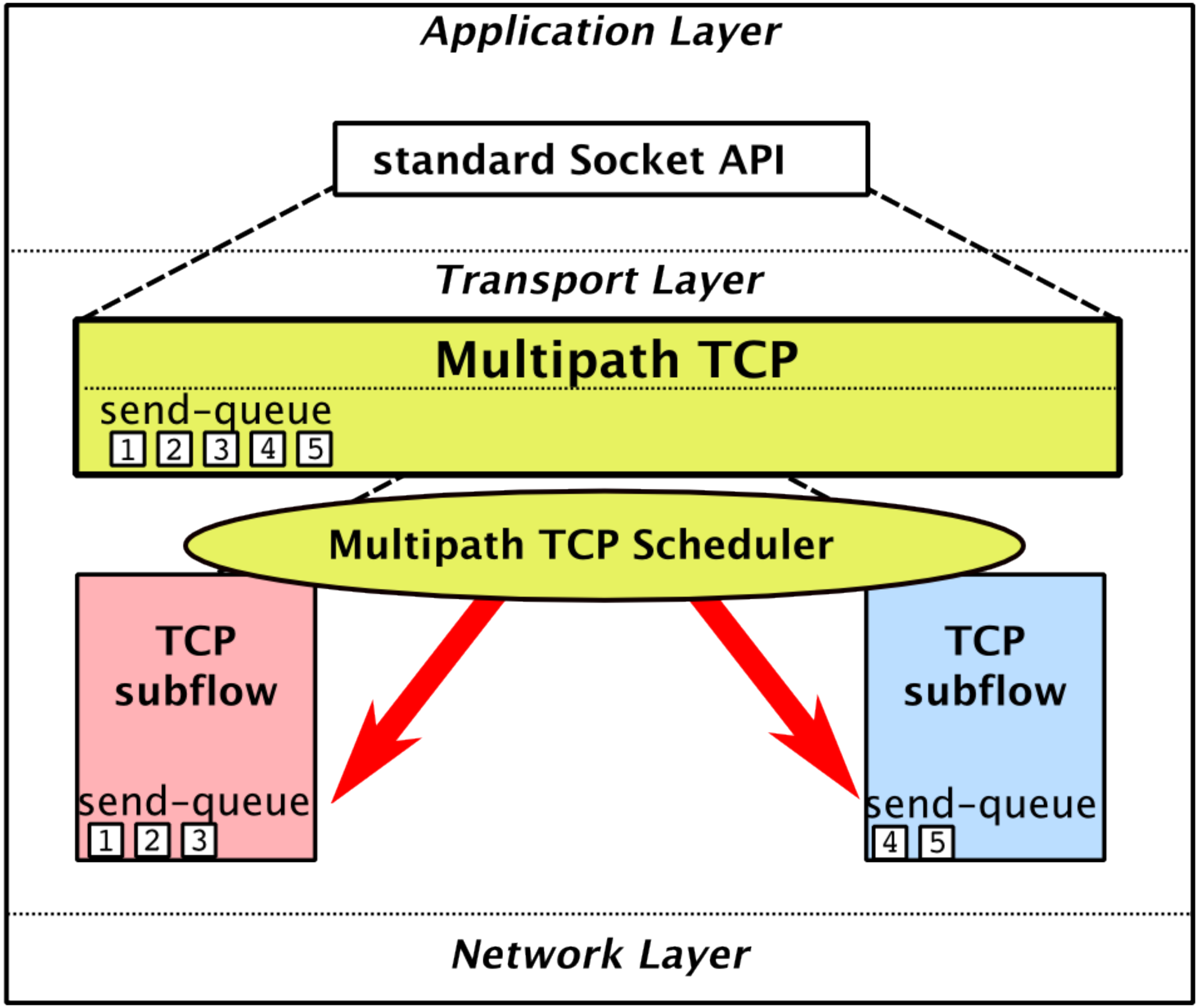
Receiving 2.1 kbit/s Sending 3.2 kbit/s
Total Received 5.2 Gbit Total Sent 5.7 Gbit

Receiving 0.0 kbit/s Sending 0.0 kbit/s
Total Received 131.2 Mbit Total Sent 42.8 Mbit

Receiving 12.1 kbit/s Sending 12.1 kbit/s
Total Received 561.9 kbit Total Sent 210.9 kbit

Enabling Ethernet

Multipath TCP Scheduler

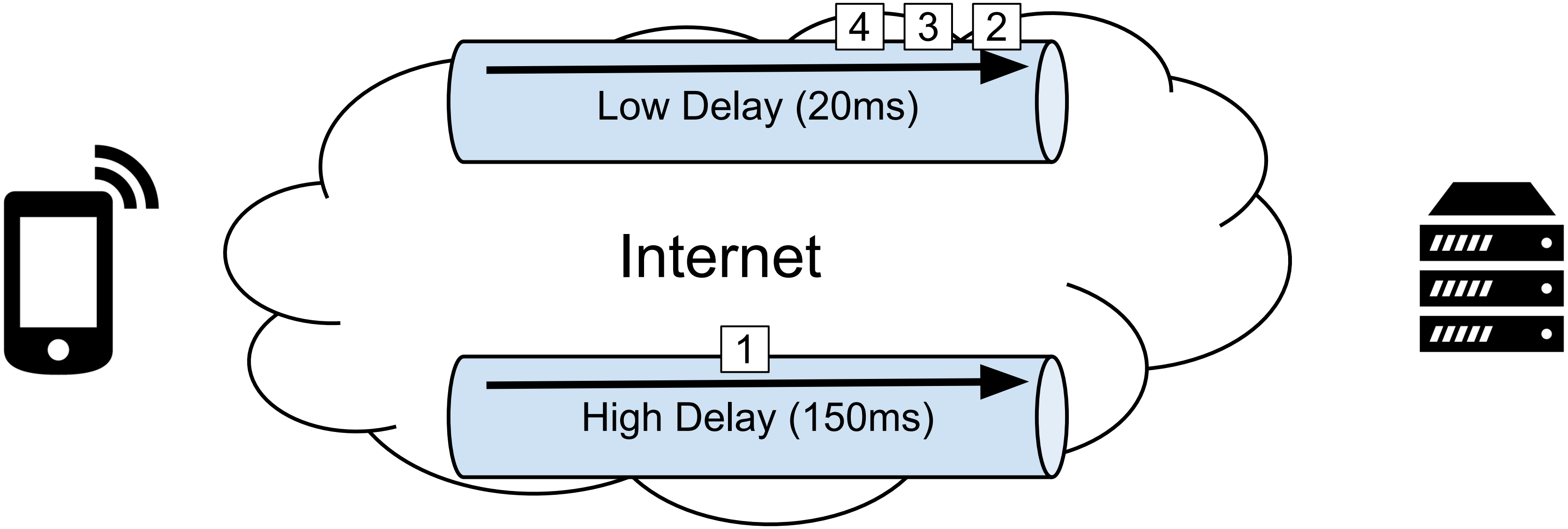


What opportunities lie in Multipath TCP Schedulers?

- What does it **influence**?
- How to **implement** it?
- How to **evaluate** it?

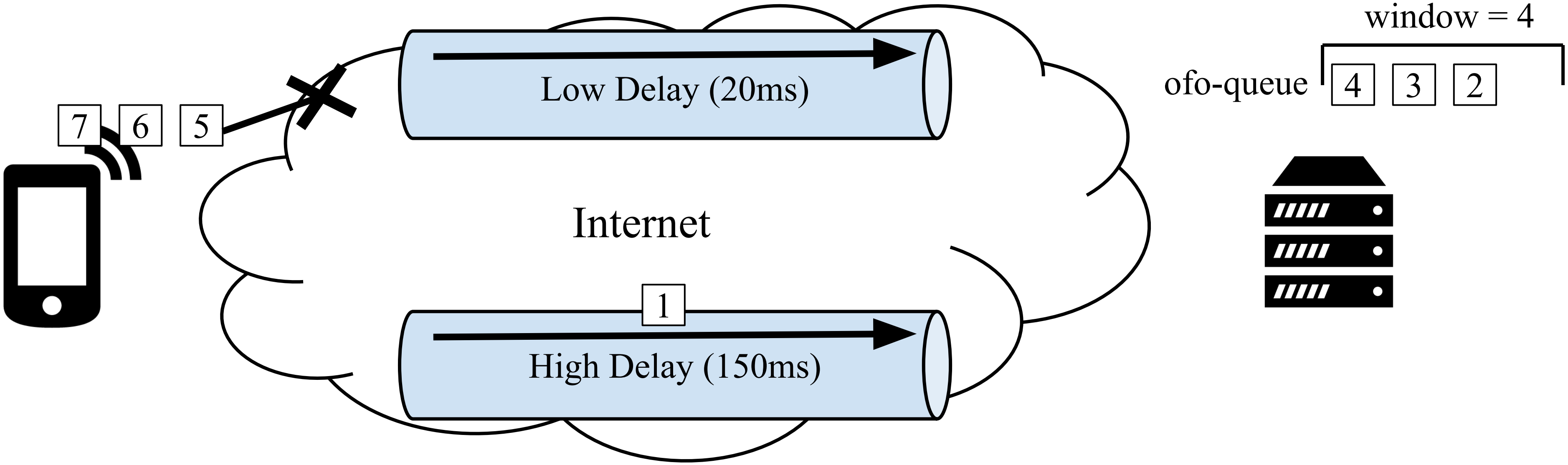
What does the scheduler influences?

Head-of-Line Blocking



- Session blocked due to #1
- High application-level delay
- Burstiness

Receive-window limitations



- Unused capacity on low-delay path
- Overall, reduced goodput

A pluggable scheduler framework

Pluggable Scheduler Framework

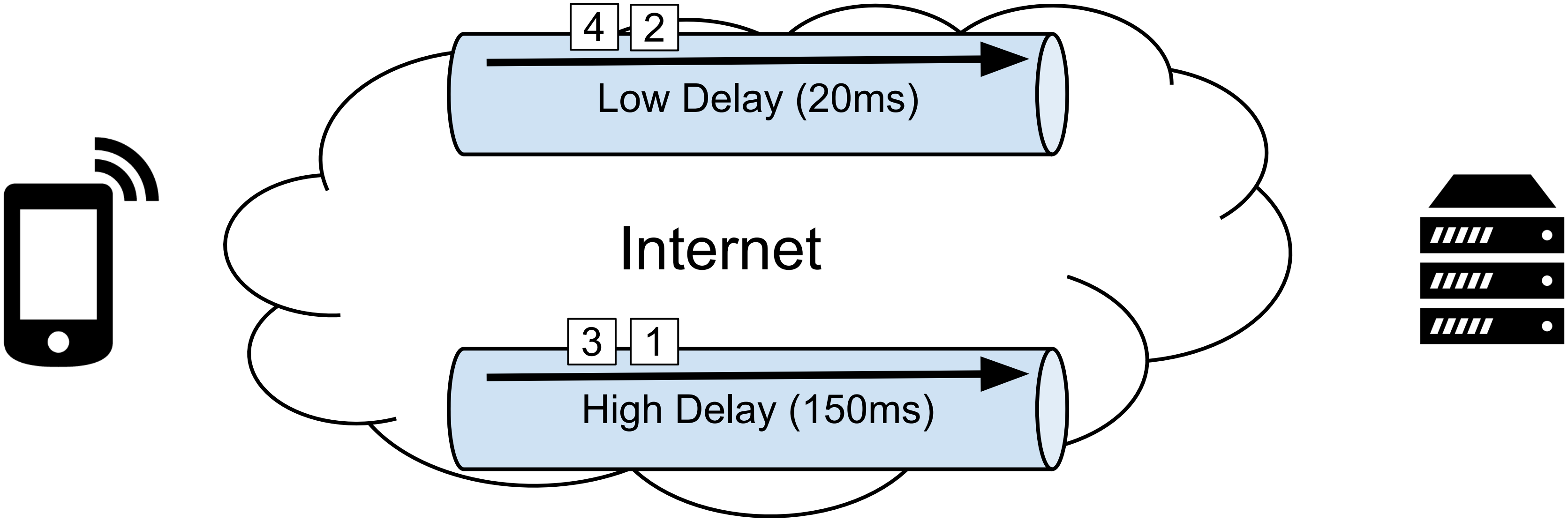
- Scheduling was a static decision
- Pluggable Scheduler Framework
    - Per MPTCP-session
    - loadable modules

```
while (subflow = MPTCP->sched->get_subflow()) != NULL do
    while (data = MPTCP->sched->get_data(subflow)) != NULL do
        send_data(subflow, data);
```

```
user@home:~$ sysctl -w net.mptcp.mptcp_scheduler=’roundrobin’
```

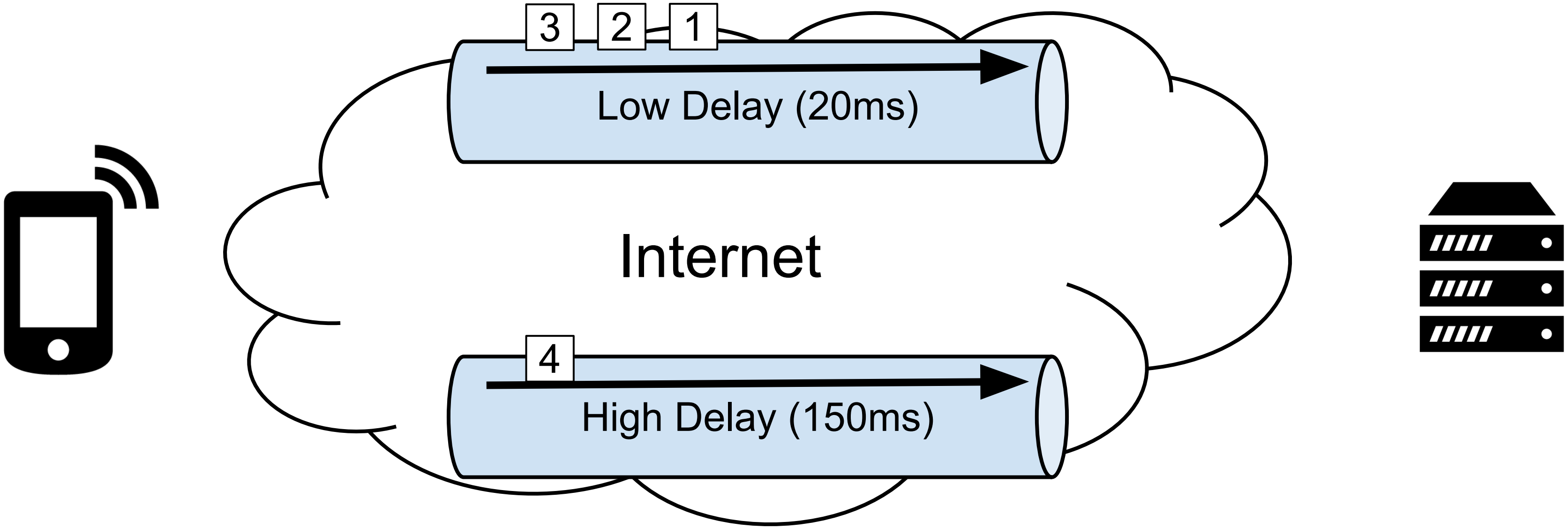
Pluggable Scheduler Framework

- Round-robin scheduler *(RR)*



Pluggable Scheduler Framework

- Round-robin scheduler *(RR)*
- Lowest-Delay-First *(LowRTT)*



Pluggable Scheduler Framework

- Round-robin scheduler *(RR)*
- Lowest-Delay-First *(LowRTT)*

Extensions:

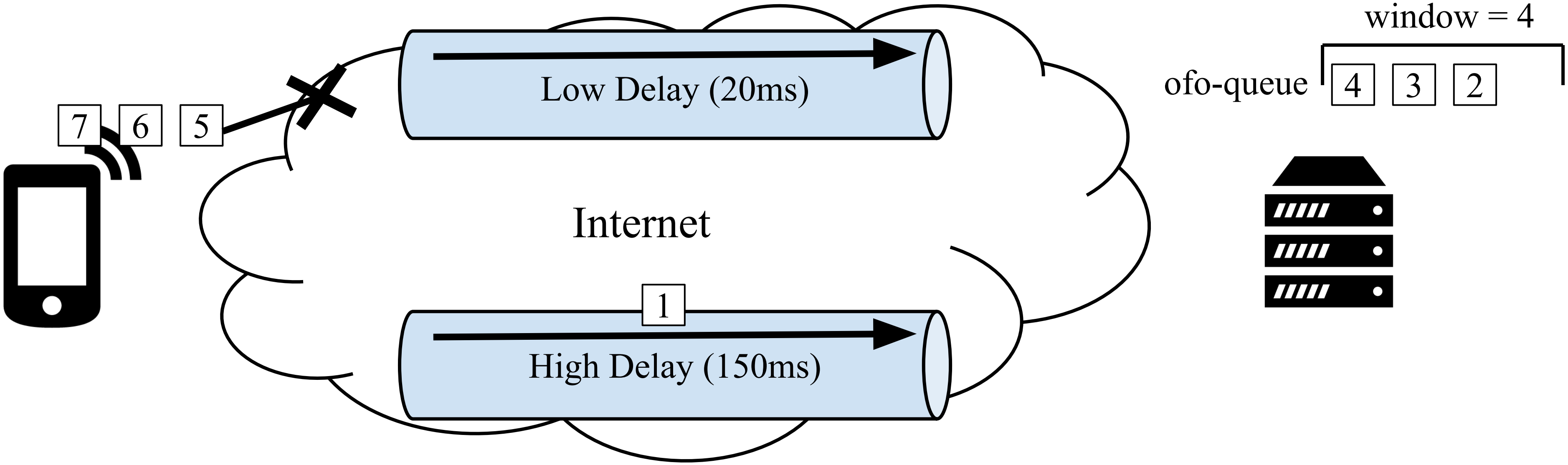
Retransmission and Penalization *(RP)*

Bufferbloat mitigation *(BM)*

Pluggable Scheduler Framework

Extensions:

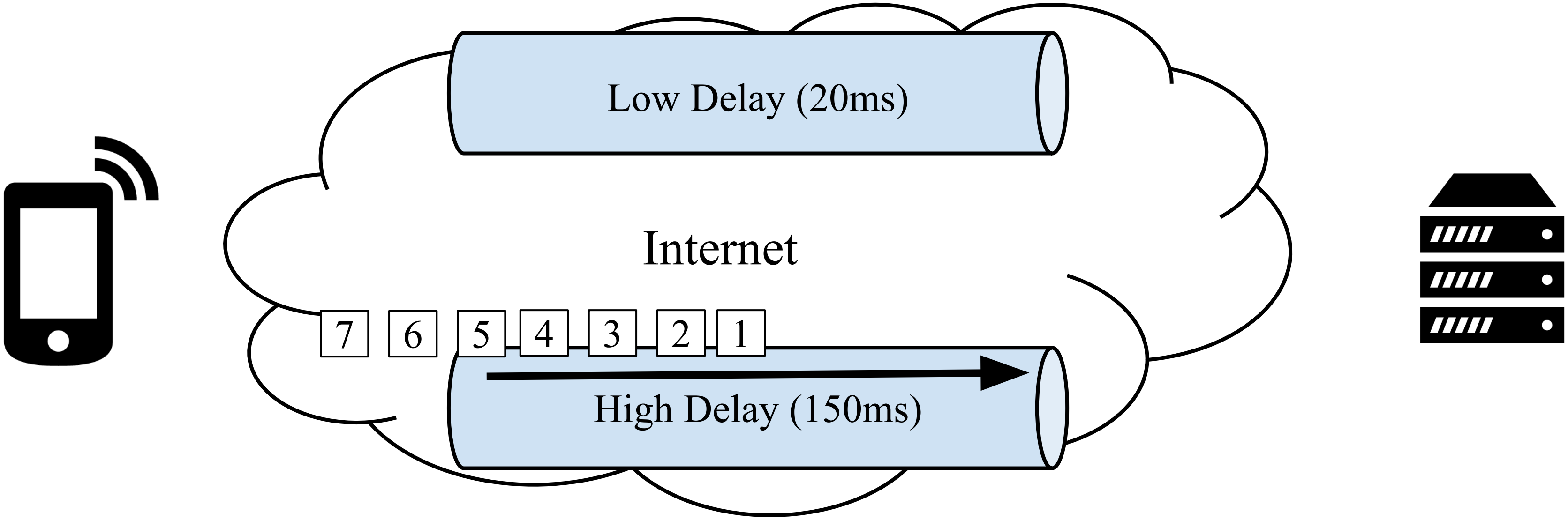
Retransmission and Penalization *(RP)*



Pluggable Scheduler Framework

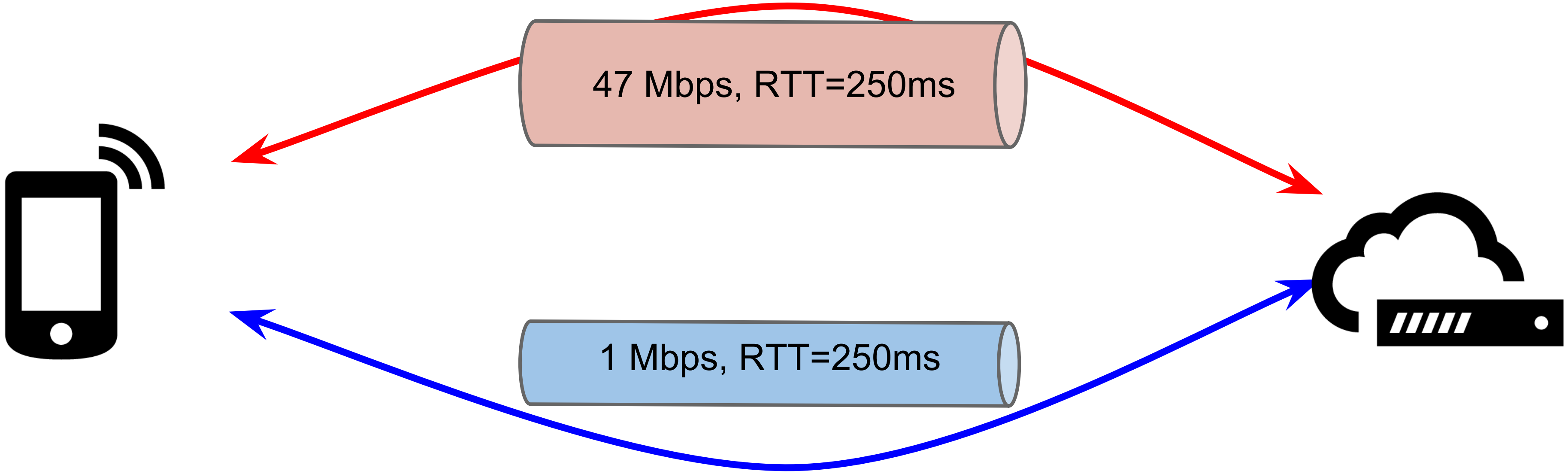
Extensions:

Bufferbloat mitigation *(BM)*



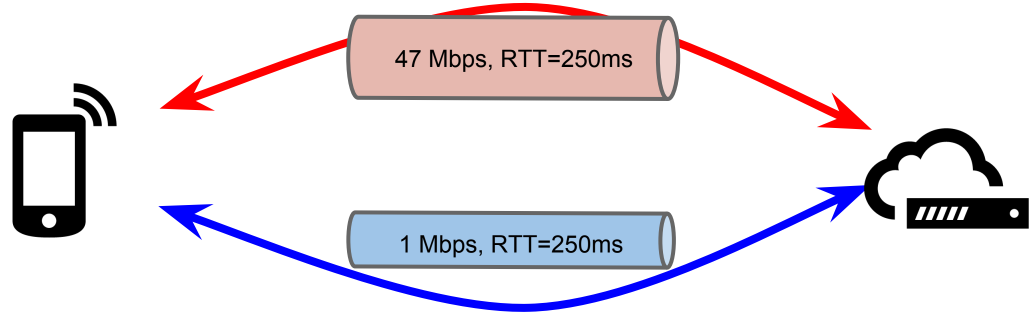
Evaluating Schedulers

Resource Pooling



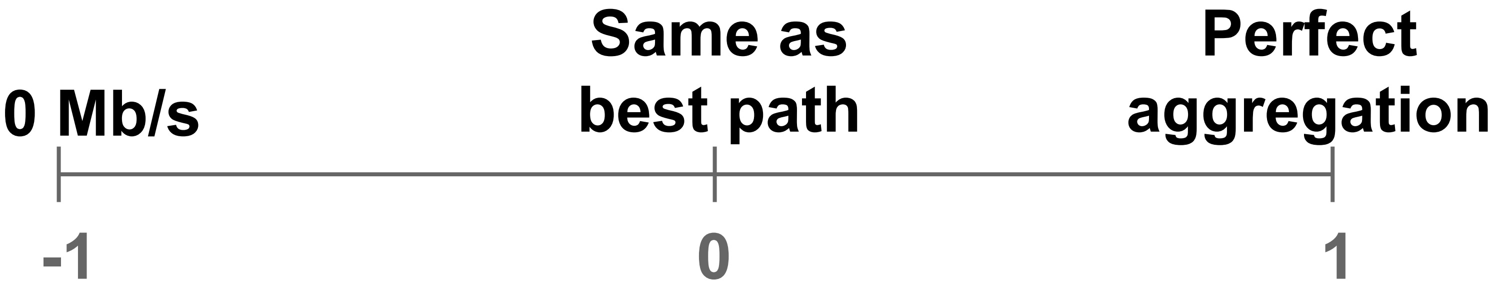
LowRTT : 11 Mbps
LowRTT + RP : 46 Mbps

Resource Pooling - normalization



47 Mbps, RTT=250ms

1 Mbps, RTT=250ms



0 Mb/s

Same as best path

Perfect aggregation

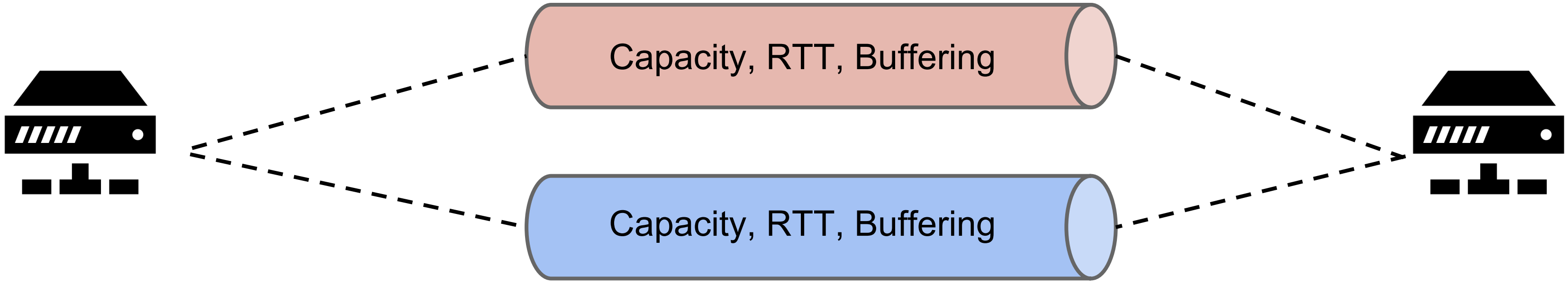
-1

0

1

“Multipath Aggregation of Heterogeneous Access Networks”. D. Kaspar. Phd Thesis. University of Oslo. 2011.

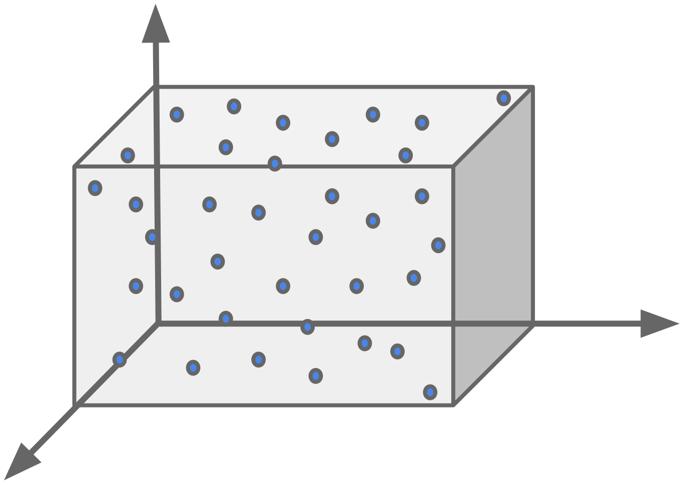
Mininet evaluation



	Low-BDP	**High-BDP**
Capacity	0.1 to 100 Mbps	0.1 to 100 Mbps
RTT	0 to 50 ms	0 to 400 ms
Buffering	0 to 100 ms	0 to 2000 ms

Mininet evaluation

Emulated environment

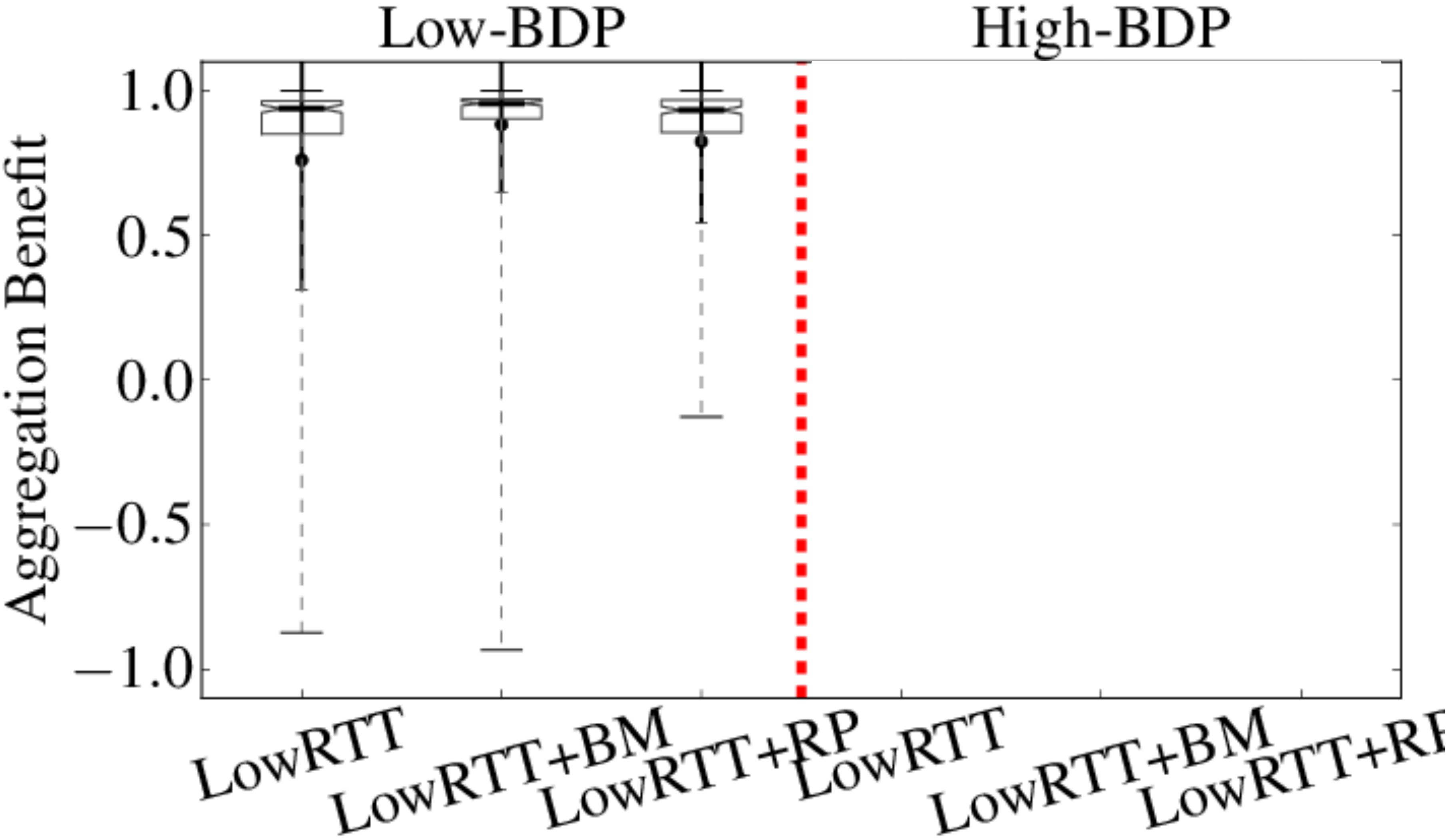


~400 experiments

“Experimental Design” -approach

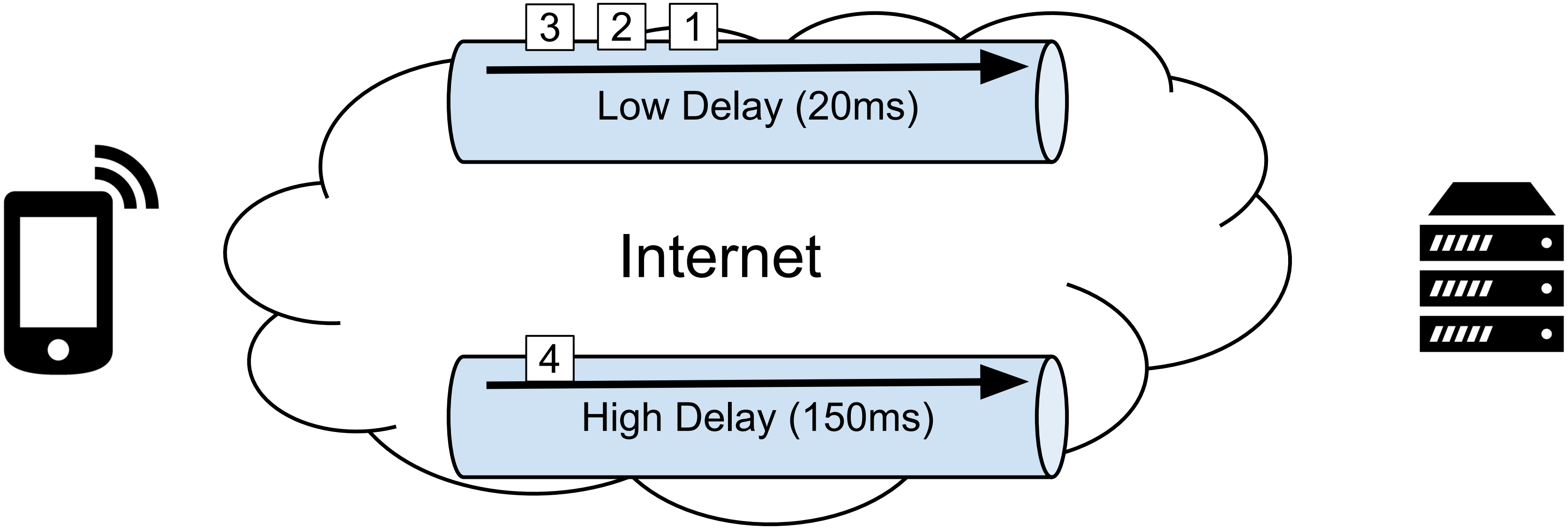
“On the Benefits of Applying Experimental Design to improve Multipath TCP”. C. Paasch, R. Khalili and O. Bonaventure. CoNEXT 2013.

Mininet evaluation

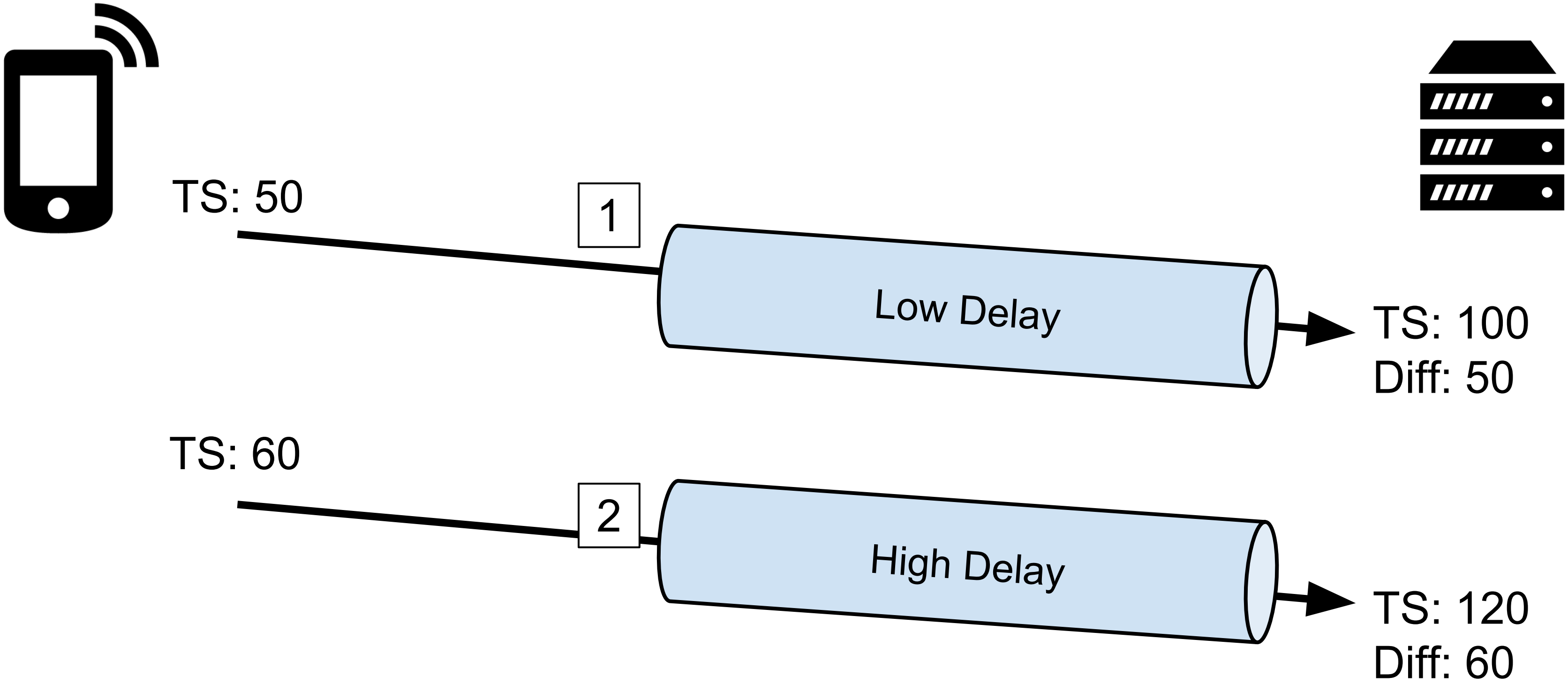


Measuring application-delay

- Custom application, sending at constant rate
- Blocks of 8KB
- Measuring application-delay

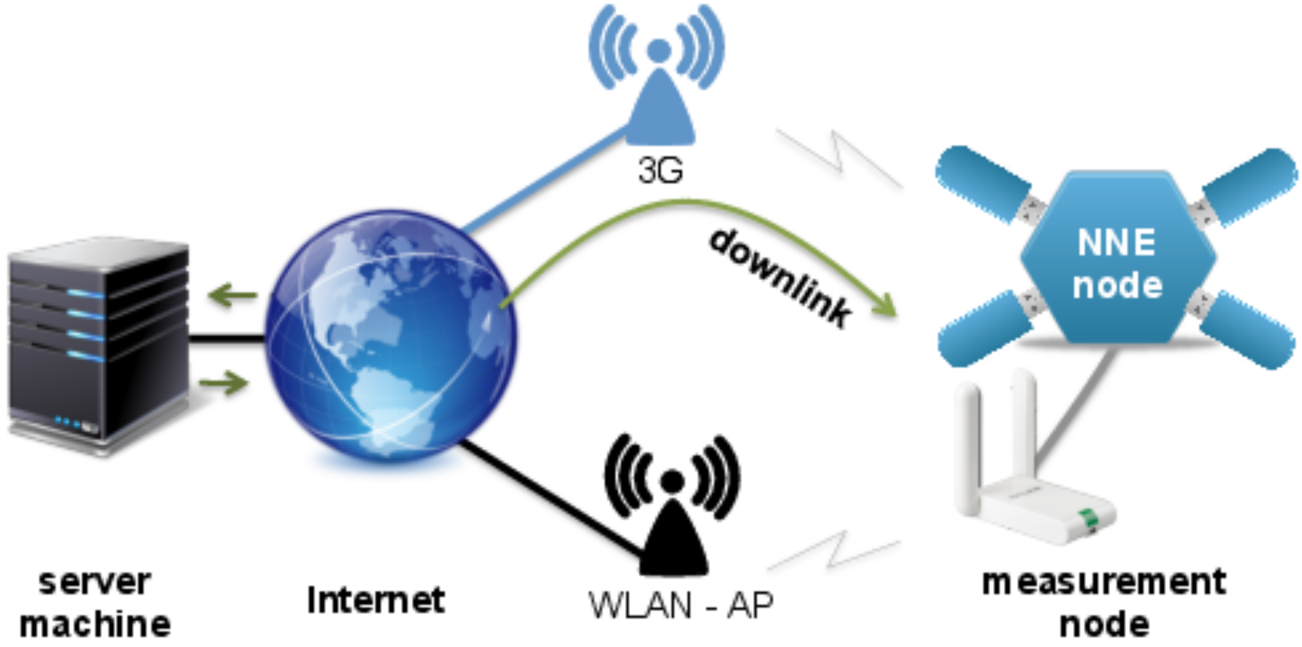


Measuring application-delay

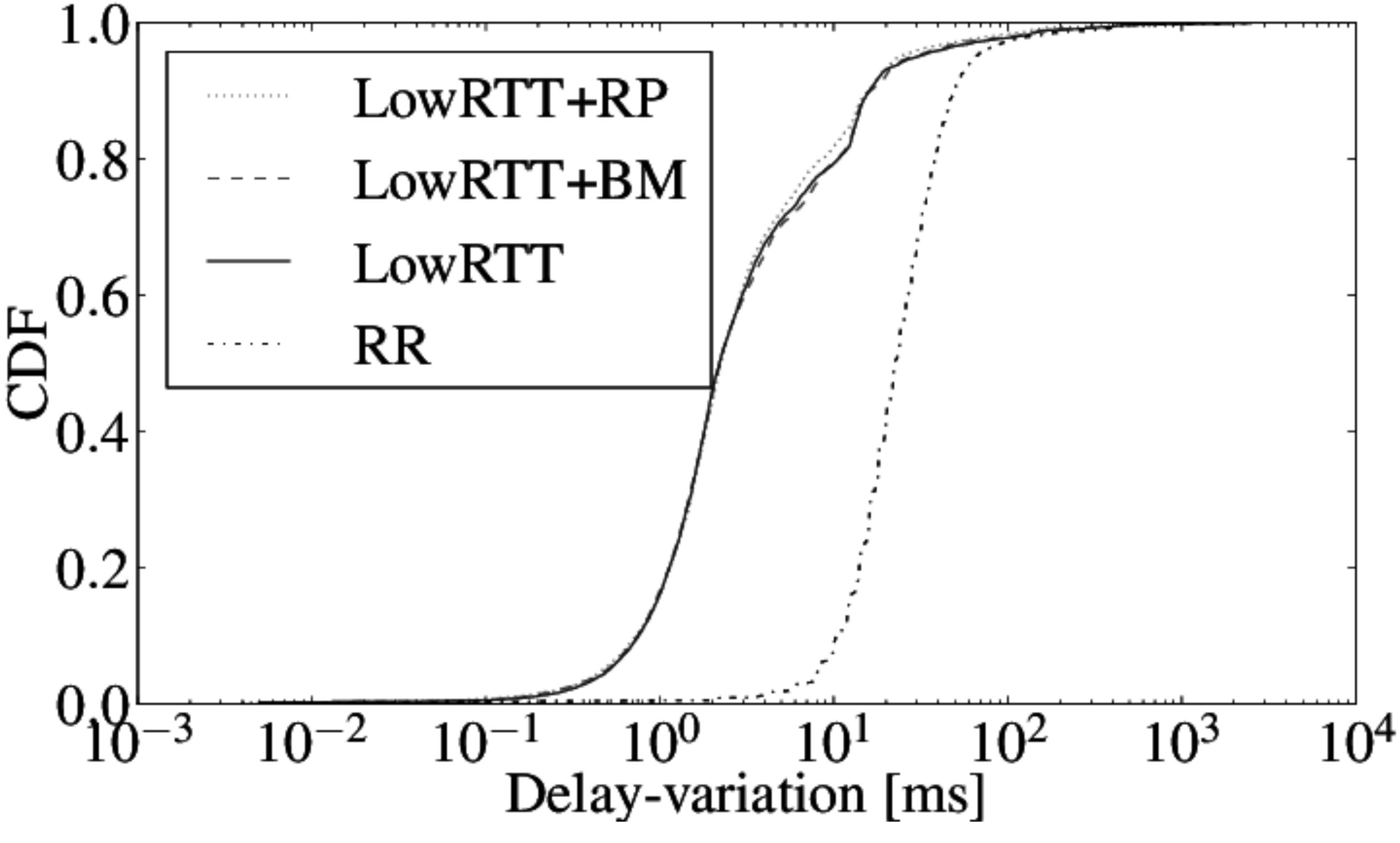


Delay-Variance: 60 - 50 = 10ms

NorNet testbed



Application-limited flows (500Kbps)



Conclusion

Conclusion

- Scheduling adds a **new dimension** with new problems and opportunities

- **Pluggable scheduler** for easy switching

- **No “perfect” scheduler** (yet)