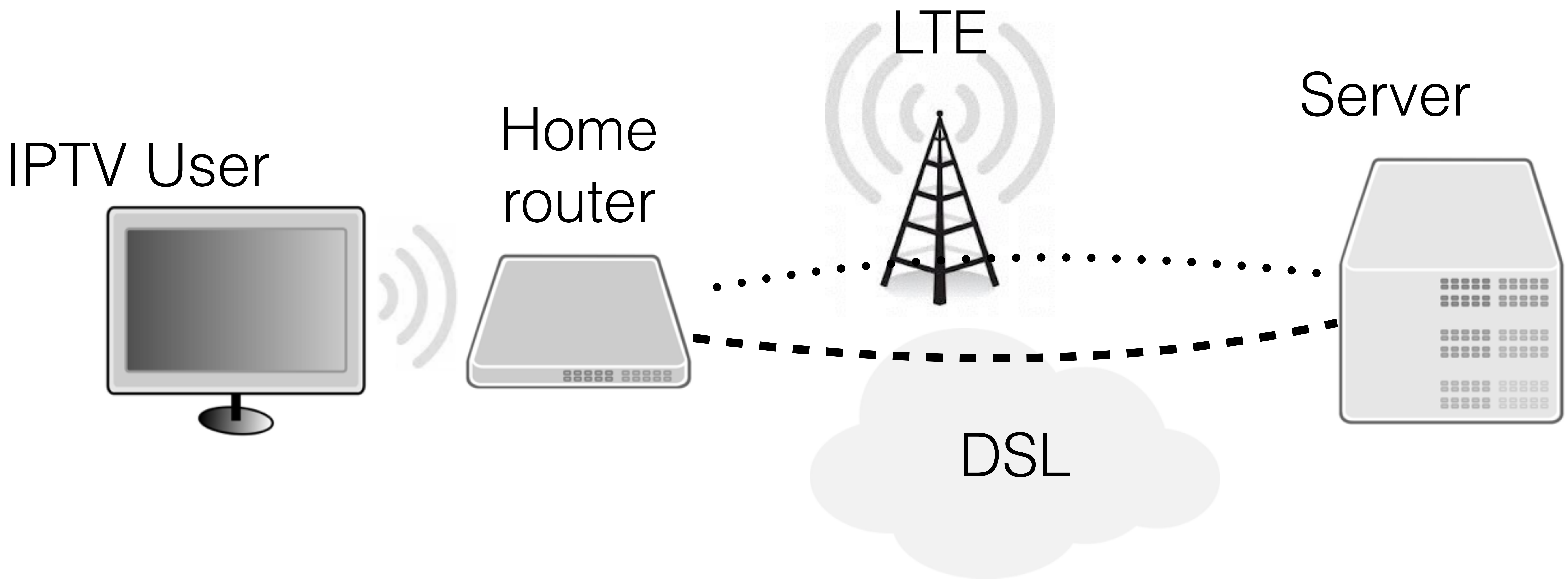
Multipath bonding at Layer 3

Maciej Bednarek (ETH Zurich), Mirja Kühlewind (ETH Zurich),
Guillermo Barrenetxea Kobas (Swisscom), **Brian Trammell (ETH Zurich)**

November 17 2016 - BANANA BoF, IETF 97 서울 (Seoul)

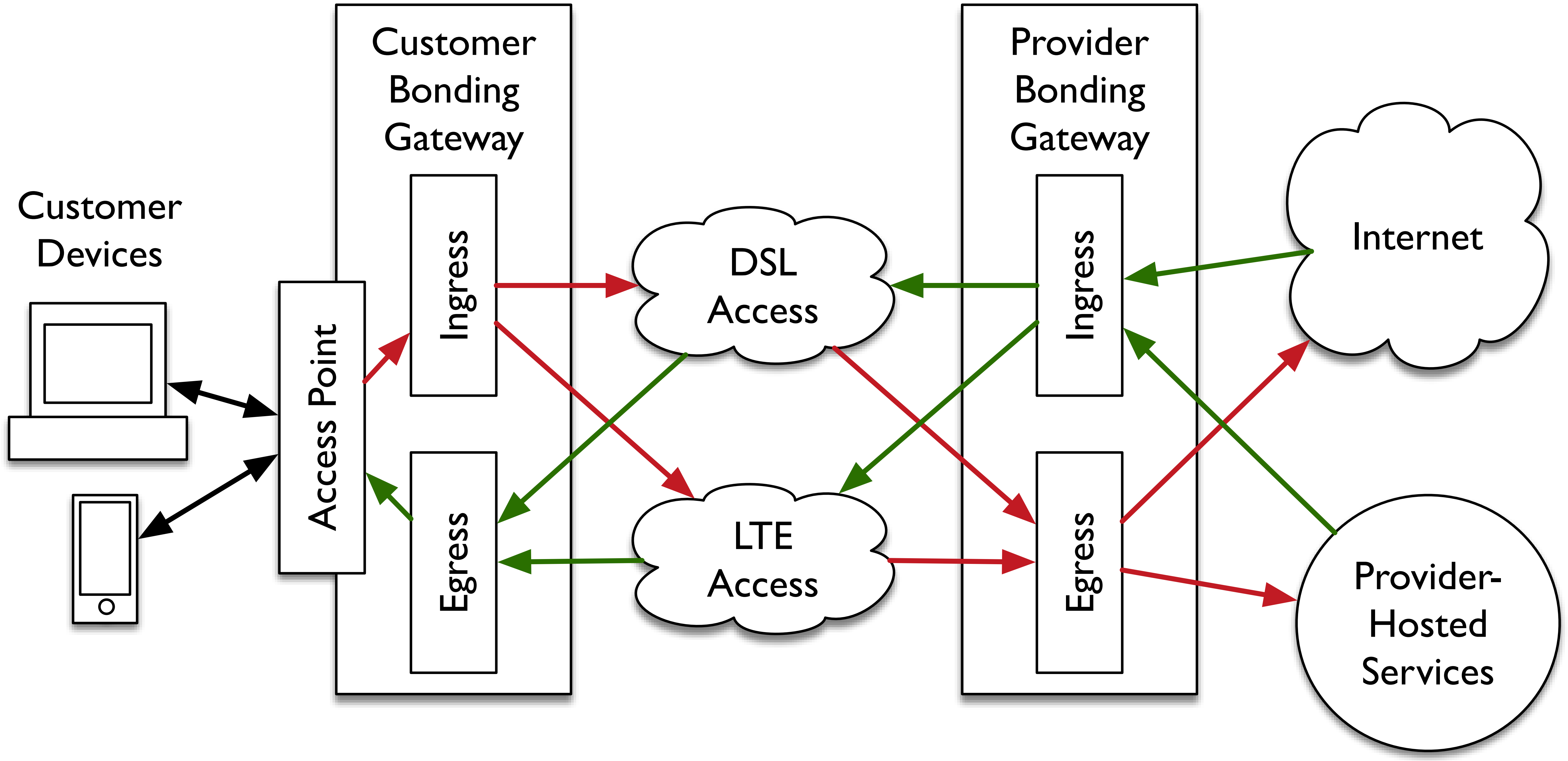
measurement and architecture for a middleboxed internet

Motivation: Aggregation of DSL and mobile capacity



- DSL capacity is not sufficient to e.g. serve HD video service
- MPTCP proxy only suitable for TCP traffic
- Paper at ANRW ’16: Multipath Bonding at Layer 3

Bonding Architecture:
Costumer and Provider Bonding Gateways



- **Ingress:** accepts traffic, schedules transmission & adds SEQ#
- **Egress:** takes traffic from bonding interface, re-orders & strips SEQ#, sends loss report to ingress

Scheduling Algorithm:

Adaptive Weight Increment (AWI)

Goal: fill fixed link first, use mobile link for excess traffic demand only

$$w_{mobile} \mathrel{+}= k * \frac{pkt_{lost}}{pkt_{sent}} * w_{fixed}$$

control parameter

AWI using Weighted Round Robin (WRR)

- fixed weight for fixed line: $w_{fixed} = 50$
- dynamic calculation for mobile line (initially $w_{mobile} = 0$):

Scheduling Algorithm:

Initial Weight Increment (IWI)

Goal: react quickly when congestion is arising

If $\boldsymbol{w_{mobile}} = 0$ & loss is reported:

increases $\boldsymbol{w_{mobile}}$ by the number of lost packets

Note: $\boldsymbol{w_{mobile}}$ is clamped to a maximum value $\boldsymbol{w_{mobilemax}} = 50$

Scheduling Algorithm:

Delayed Weight Decrement (DWD)

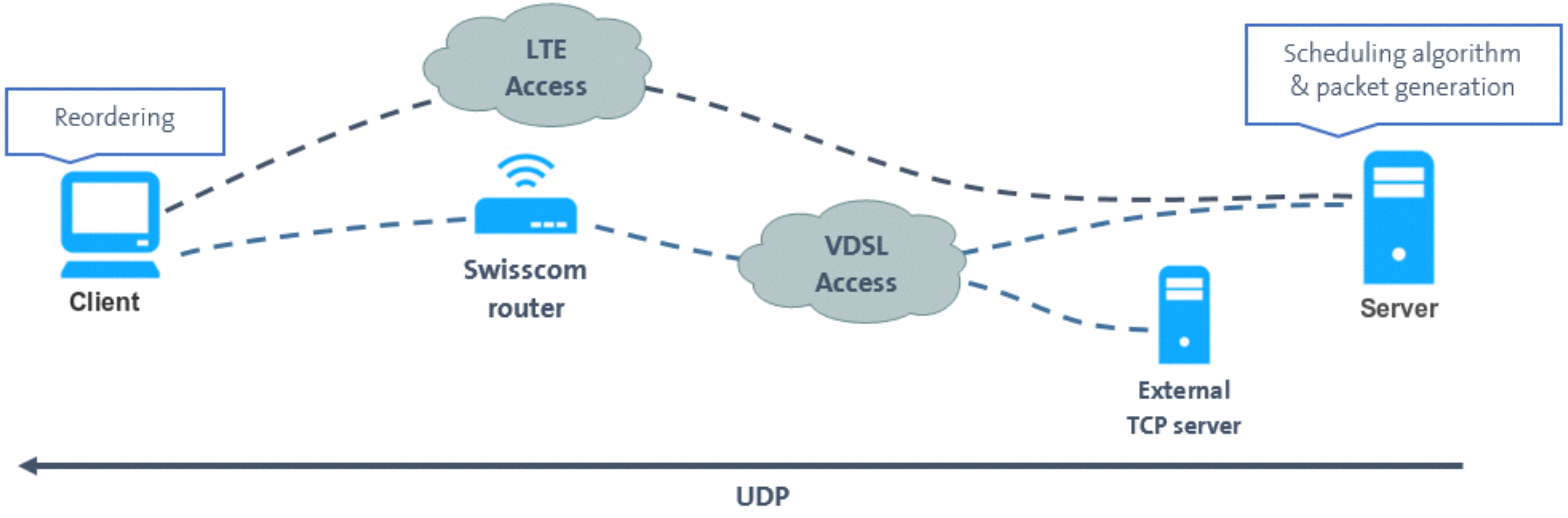
Goal: shift load back to the fixed line without inducing loss by shifting the load too quickly

If no loss reported for $\boldsymbol{T_{dwd}}$:

decrement $\boldsymbol{w_{mobile}}$ by one for each interval $\boldsymbol{T_{report}} = 50ms$

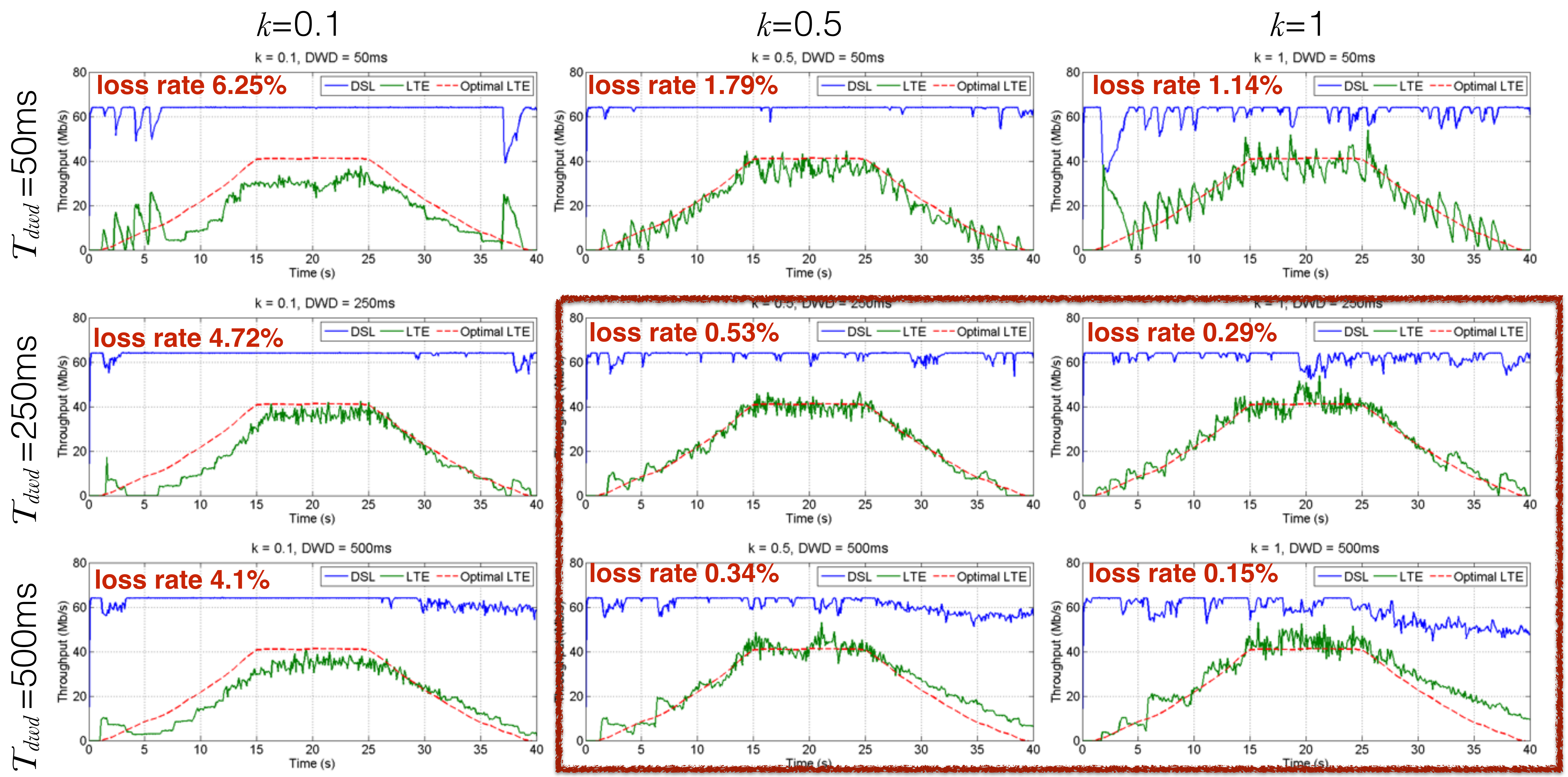
Note: We investigate different values for $\boldsymbol{T_{dwd}}$ but it must be a multiple of $\boldsymbol{T_{report}}$ (as loss reports are only received every $\boldsymbol{T_{report}}$ milliseconds)

Evaluation: Experimental setup



- Two Linux Debian Wheezy machines (client & server)
- 1492 bytes UDP packets (28 bytes UDP/IPv4 header, 4 bytes for SEQ#, and 1460 bytes of dummy payload)
- TCP cross traffic: file transfer from a public server (cdimage.debian.org) with 50ms to client
- DSL link is shaped to a maximum rate of 64 Mb/s and stable 13ms delay (measured)
- Swisscom's Huawei E3276s LTE stick with about 60Mb/s (and variable delay of 25 - 45ms)

Evaluation: Results for a single flow

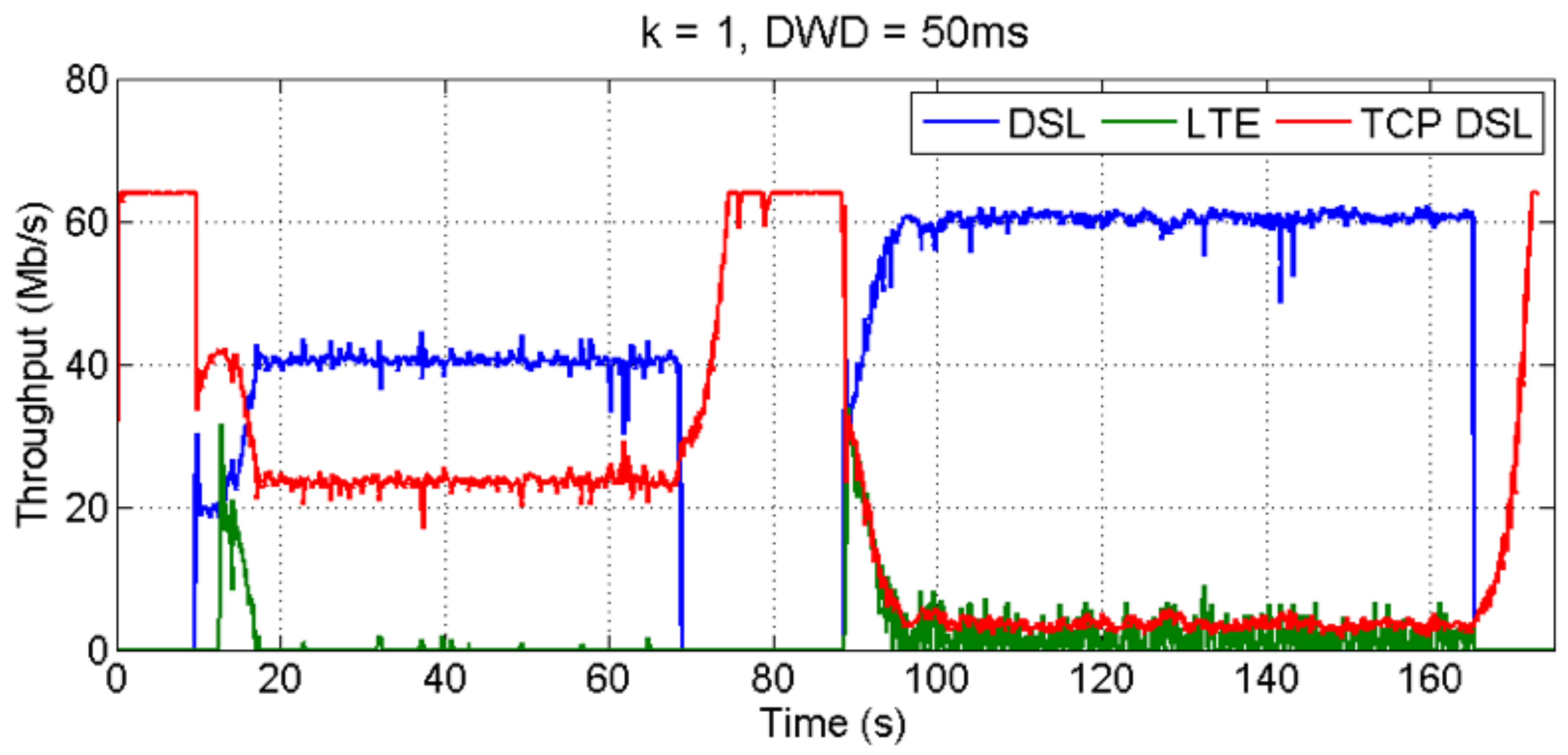


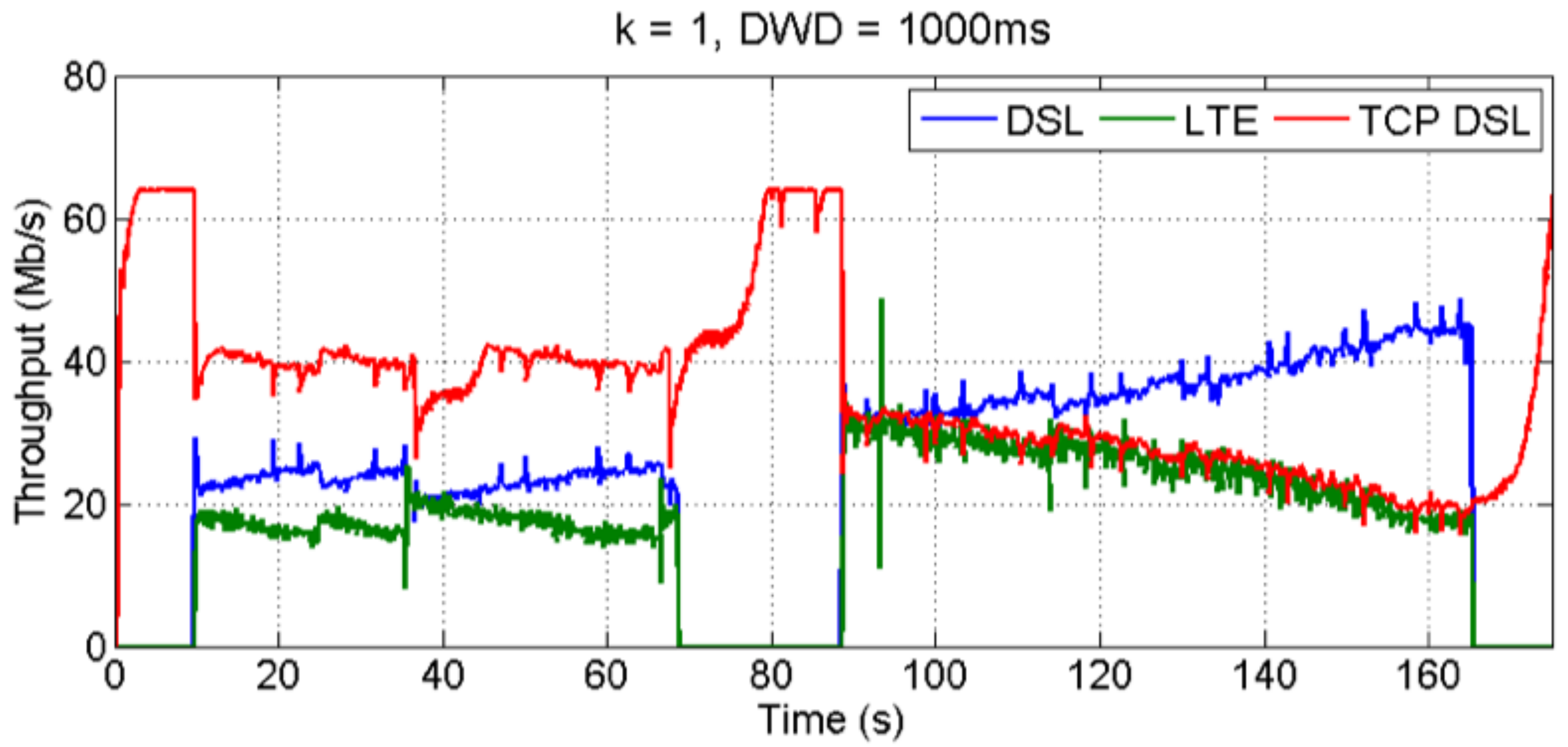
➡ k and T_{dwd} provide trade-off between aggressiveness and responsiveness

Evaluation: Results with TCP cross traffic

- $T_{dwd} = 50ms$: TCP flow only gets spare capacity
- $T_{dwd} = 1000ms$: UDP traffic permanently shifted to mobile link

➡ Operator can decide how TCP-friendly the algorithm should be





Conclusion

- **Goal:** Aggregation of DSL and mobile capacity for excess traffic
- Layer 3 bonding solution
  - Ingress: Packet mangling and scheduling that adapts w_{mobile} dynamically
  - Egress: Re-ordering buffer
- Evaluation of parameters k and T_{dwd} for trade-off aggressiveness/responsiveness tradeoff
- **Future Work and Potential for Standardization**
  - Interoperation with MPTCP proxies (deployed and proposed)
  - Standardize reordering support at egress
    - Apply Generic Routing Encapsulation (GRE), use Sequence Number and Key fields?
    - RuRo (reordering insensitivity bit) to disable reordering based on transport tolerance.
  - Standardized measurement feedback loop