The Impact of Delay Variations on TCP Performance

Michael Scharf
Institut für Kommunikationsnetze und Rechnersysteme
Universität Stuttgart
scharf@ikr.uni-stuttgart.de

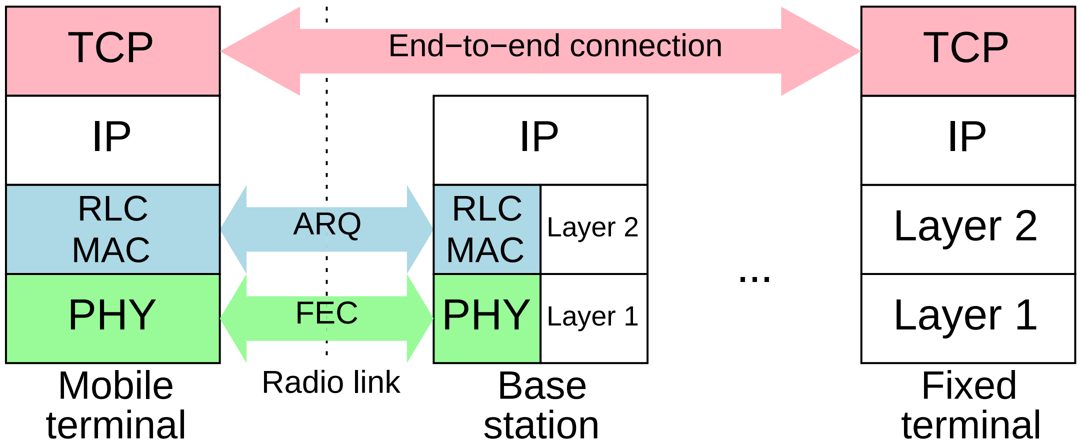
ITG FG 5.2.1 Workshop, Hamburg
20. Feb. 2004

Outline

- Characteristics of mobile networks
- Impact of delay variations on TCP
- Modeling the RTT estimation in TCP
- Model validation
- Optimization approaches
- Conclusions and future work

Characteristics of mobile networks

Network architecture (simplified)



TCP... Transmission Control Protoc
IP... Internet Protocol
RLC... Radio Link Control
MAC... Medium Access Control
PHY... Physical Layer
ARQ... Automatic Repeat Request
FEC... Forward Error Correction

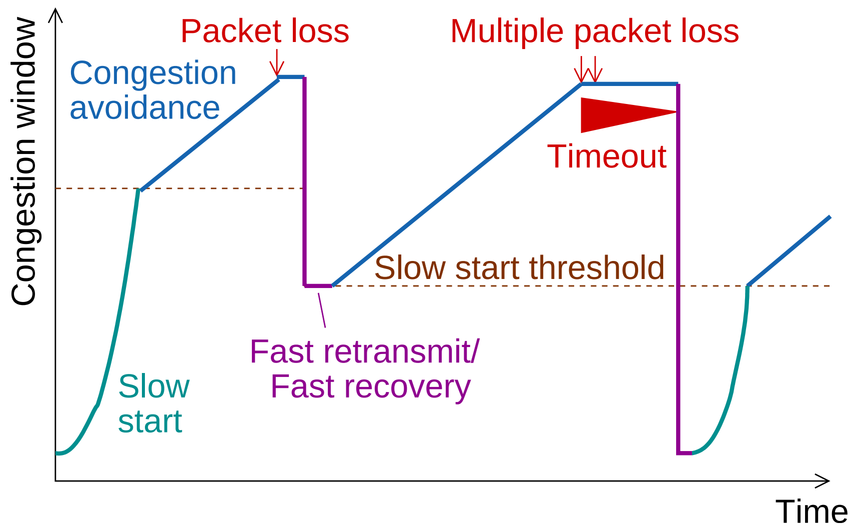
- **Few packet losses due to highly persistent link layer**
- **Rather low bandwidth, but high latency**
- **Variable delays and significant jitter**
  - Retransmission of radio blocks (ARQ mechanism)
  - Cell handovers and link outages
  - Radio resource preemption by voice traffic

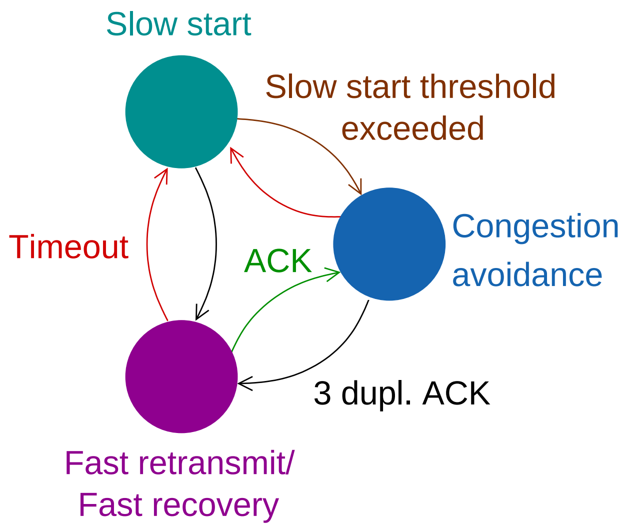
➥ **Impact of delay variations on TCP performance?**

Impact of delay variations on TCP (1)

Transmission Control Protocol (TCP)

- **Window-based, reliable transport protocol**
  - Receiver advertised window
  - Congestion window
- **Error recovery**
  - Fast retransmit/Fast recovery mechanism
  - Retransmission timeouts
- **Different variants: TCP Reno, NewReno, SACK, ...**





Impact of delay variations on TCP (2)

Potential effects

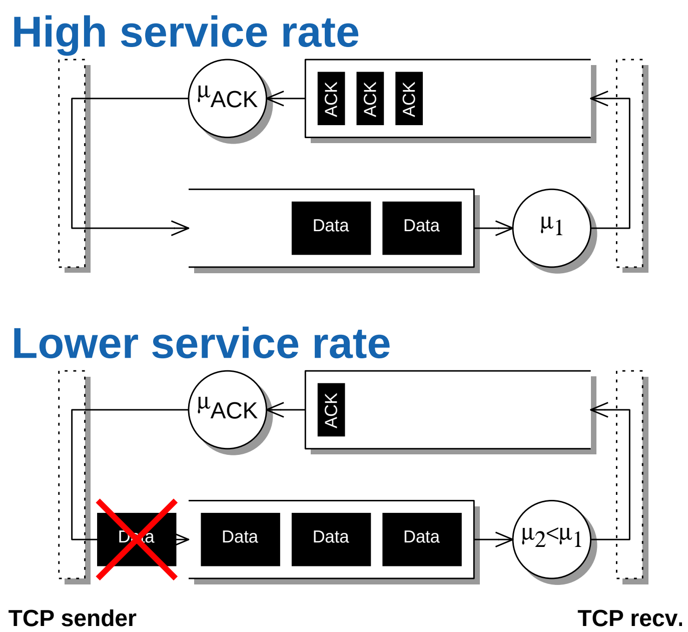
- **Spurious TCP timeouts = timeouts even though no packets are lost**
  - *Go-back-N* mechanism unnecessarily retransmits segments
  - Needless reduction of congestion window
  - ➥ TCP may waste bandwidth or underutilize the available resources

Impact of delay variations on TCP (2)

Potential effects

- ❐ **Spurious TCP timeouts = timeouts even though no packets are lost**
  - ❍ *Go-back-N* mechanism unnecessarily retransmits segments
  - ❍ Needless reduction of congestion window
  - ➥ TCP may waste bandwidth or underutilize the available resources

- ❐ **Mis-estimation of available network capacity by TCP**
  - ❍ Buffer overflows if over-estimated
  - ❍ Poor throughput if under-estimated



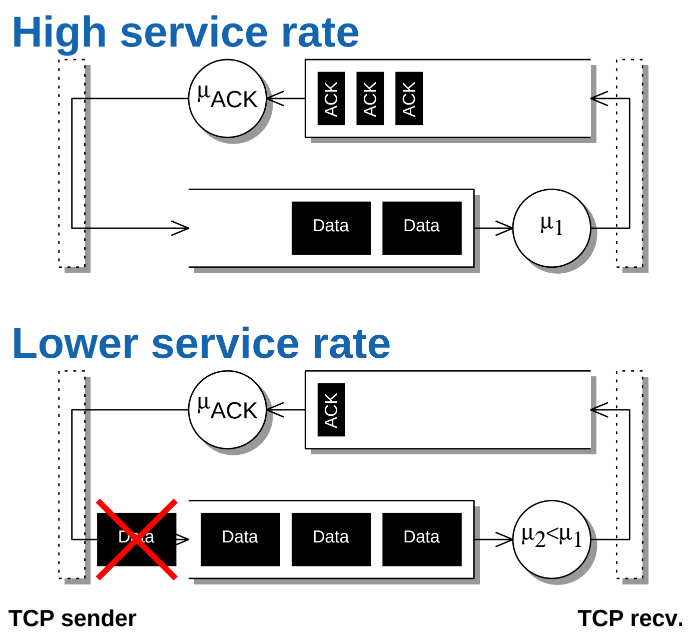
Impact of delay variations on TCP (2)

Potential effects

- **Spurious TCP timeouts = timeouts even though no packets are lost**
  - *Go-back-N* mechanism unnecessarily retransmits segments
  - Needless reduction of congestion window
  - ➥ TCP may waste bandwidth or underutilize the available resources

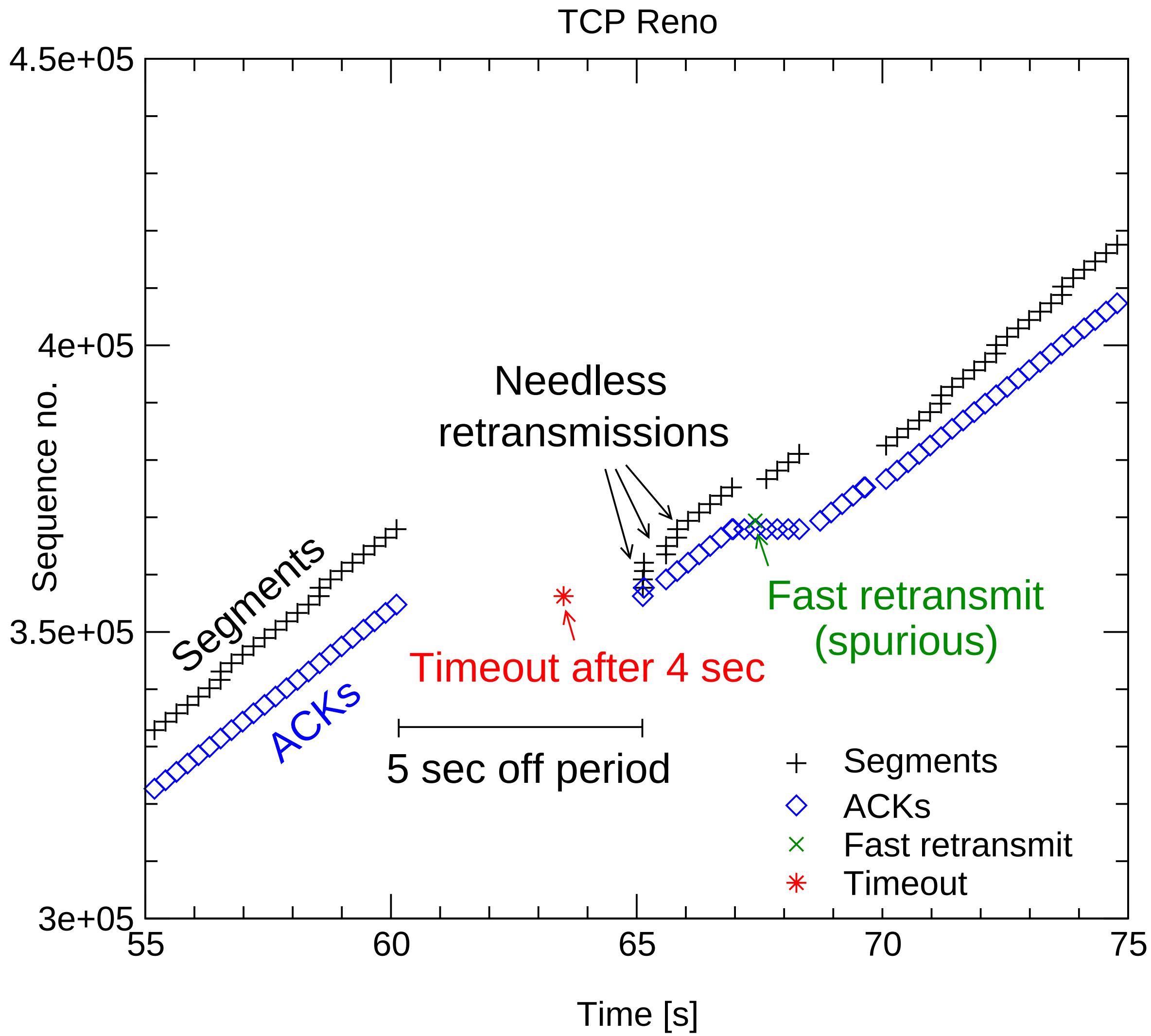
- **Mis-estimation of available network capacity by TCP**
  - Buffer overflows if over-estimated
  - Poor throughput if under-estimated

- **Packet reordering**
  - May trigger spurious fast retransmits
  - Usually prevented by reliable link layer



Impact of delay variations on TCP (3)

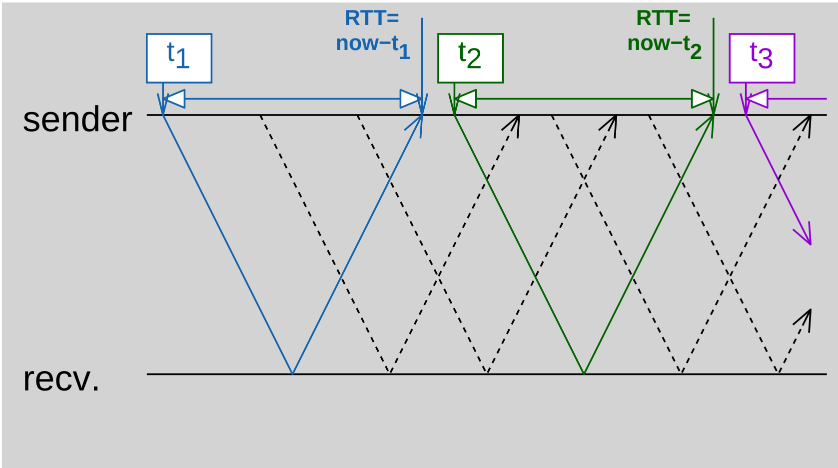
Spurious timeouts - An example



Modeling the RTT estimation in TCP (1)

RTT sampling by TCP

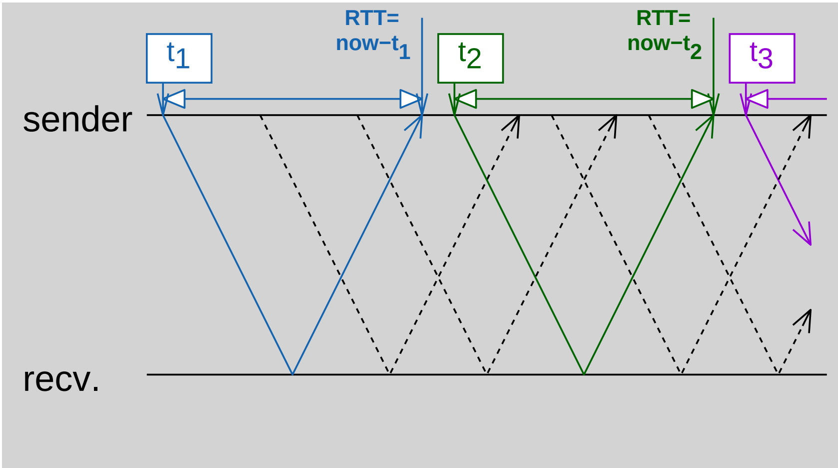
Default method:



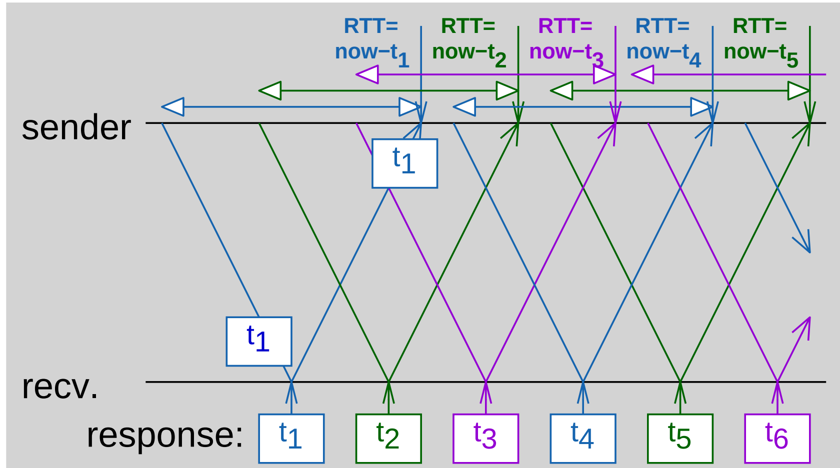
- ❒ **One measurement per RTT**
- ❒ **Low sampling rate**
- ❒ **No measurement on retransmitted segments (Karn's algorithm)**

RTT sampling by TCP

Default method:



Timestamps (RFC 1323):



- ❒ **One measurement per RTT**
- ❒ **Low sampling rate**
- ❒ **No measurement on retransmitted segments (Karn's algorithm)**

- ❒ **Additional 12 byte TCP header option**
- ❒ **Samples with every new ACK**
- ❒ **Measurements on retransmitted segments**

Modeling the RTT estimation in TCP (2)

RTO calculation

- **Tradeoff necessary**
  - Frequent spurious timeouts if value too small (aggressive timer)
  - Long idle times if value too large (conservative timer)
- **Algorithm according to RFC 2988:**

$$\text{RTTVAR: } v(n) = \frac{3}{4} \cdot v(n-1) + \frac{1}{4} \cdot |s(n-1) - x(n)|$$

$$\text{SRTT: } s(n) = \frac{7}{8} \cdot s(n-1) + \frac{1}{8} \cdot x(n)$$

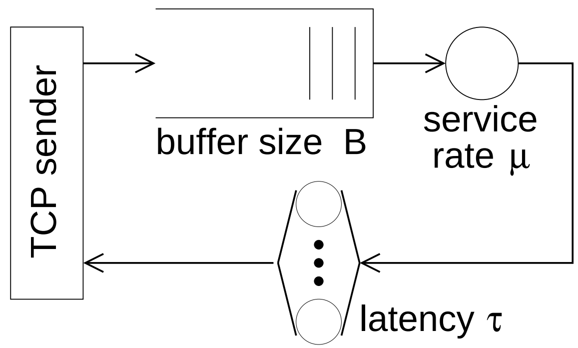
$$\text{RTO: } R(n) = \max(s(n) + 4 \cdot v(n), m)$$

- **Recommendation for minimum:** m=1 sec
- **Coarse timer granularity in many protocol stacks**

Modeling the RTT estimation in TCP (3)

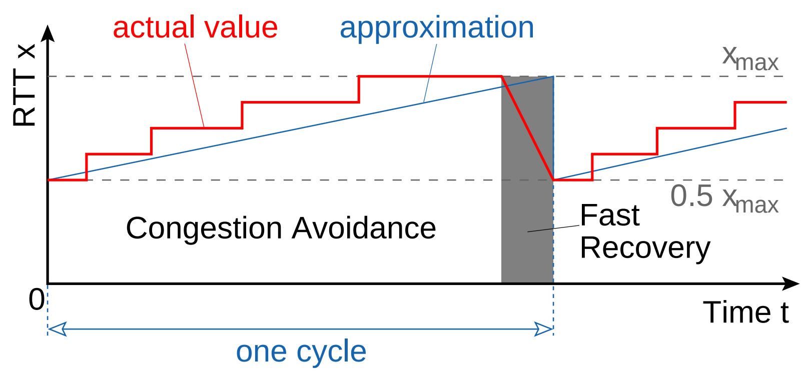
Assumptions

- **Single bottleneck: Radio link**
- **Main parameters**
  - Service rate μ
  - Latency τ
  - Buffer size $B > \mu \cdot \tau$

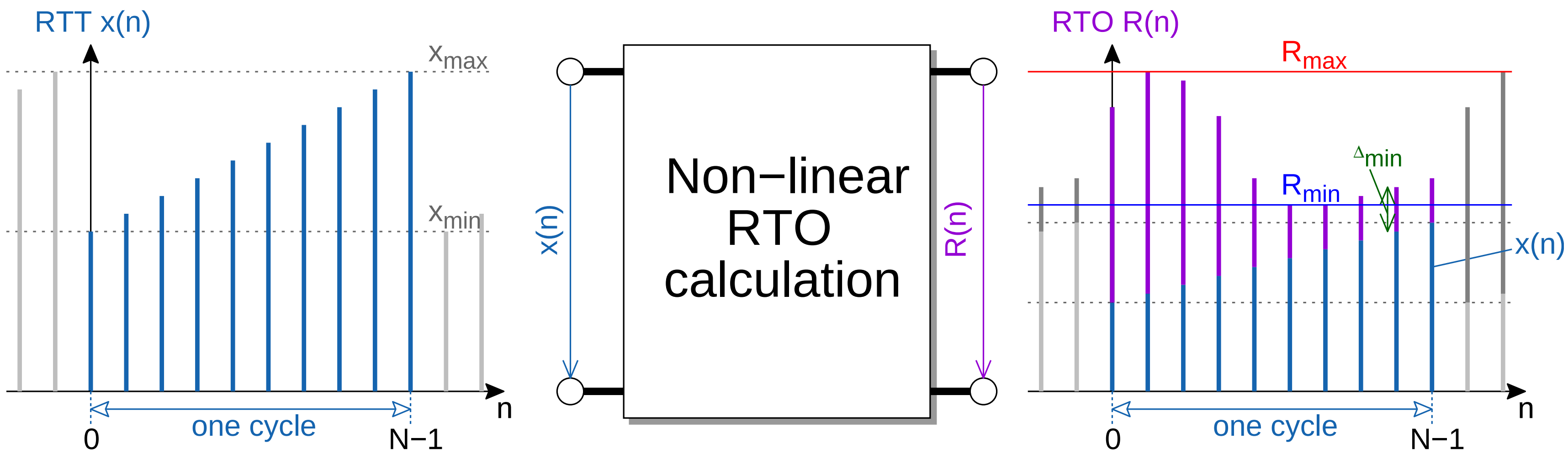


- **TCP bulk data transfer**
  - Single TCP connection over radio link
  - Greedy source
- **Not limited by receiver advertised window**

➥ **Typical saw-tooth behavior with *cycles***



Modeling approach



- ❒ **RTO calculation is a non-linear filter**
- ❒ **Approximation: Linear input function** $x(n)$
- ❒ **Number of samples per cycle:**

$$N = f(\mu, \tau, B, \text{sampling method})$$

➥ **RTO duration** $R(n)$ **can be determined analytically**

Modeling the RTT estimation in TCP (5)

RTT sampling rate

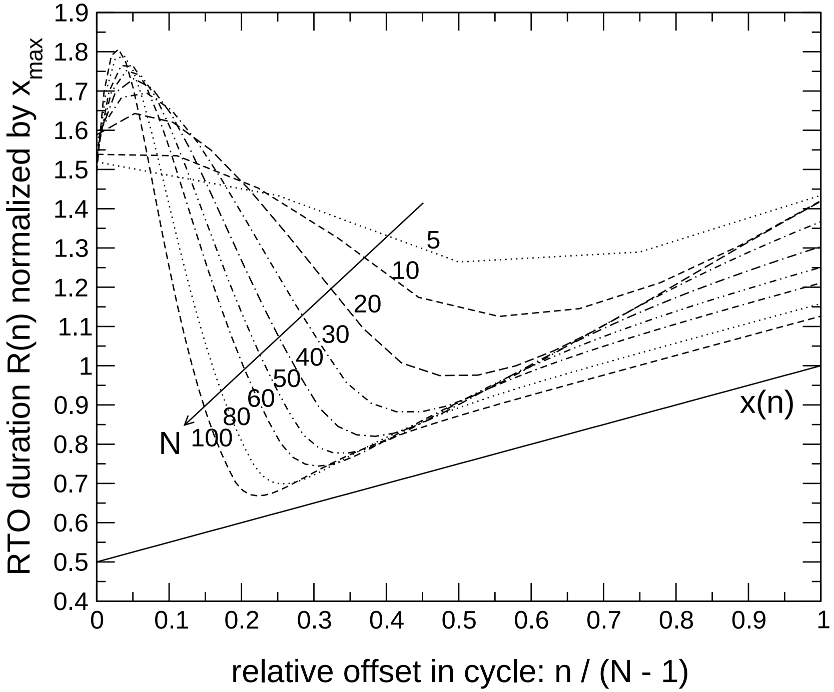
- **Path capacity:** $C = \lfloor B + 1 + \mu \cdot \tau \rfloor$
- **Delayed Acknowledgements mechanism:** b **segments per ACK**
- **Samples per cycle for default measurements**

$$N = b\left\lceil \frac{1}{2}(C+1) + 2 \right\rceil$$

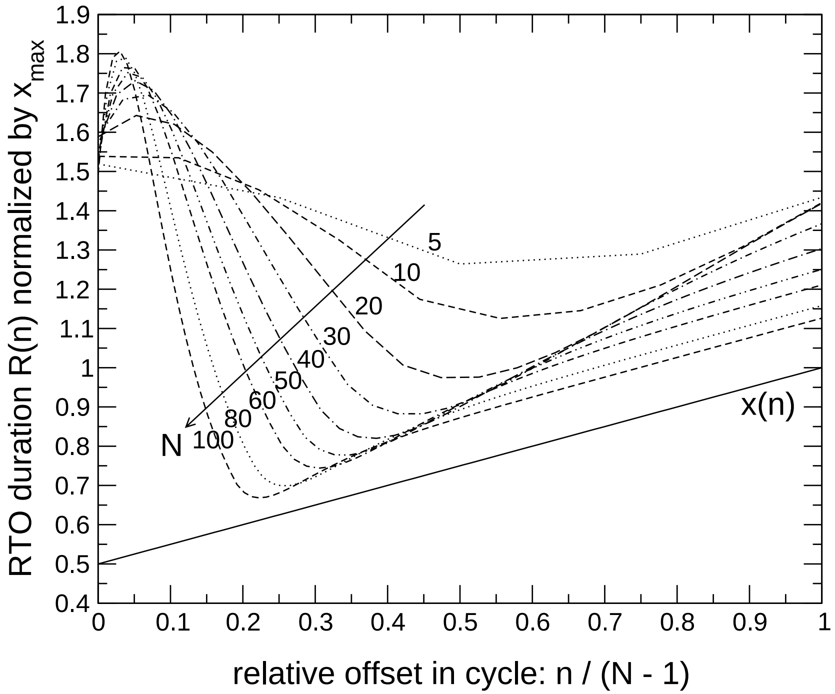
- **Samples per cycle for timestamps**

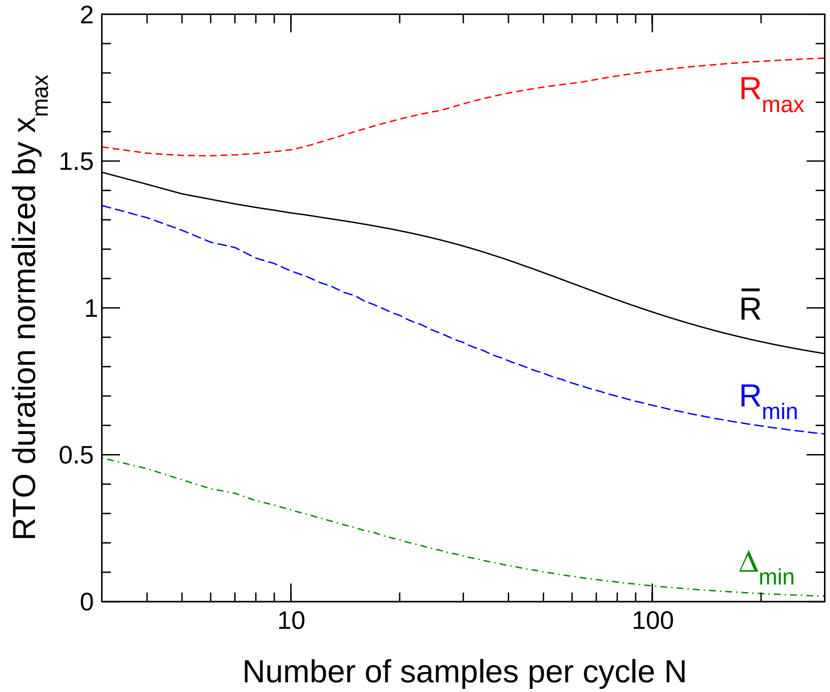
$$N = \left\lceil \frac{3}{8}(C+1)^2 + \frac{C}{b} + 1 \right\rceil$$

Numerical results



Numerical results

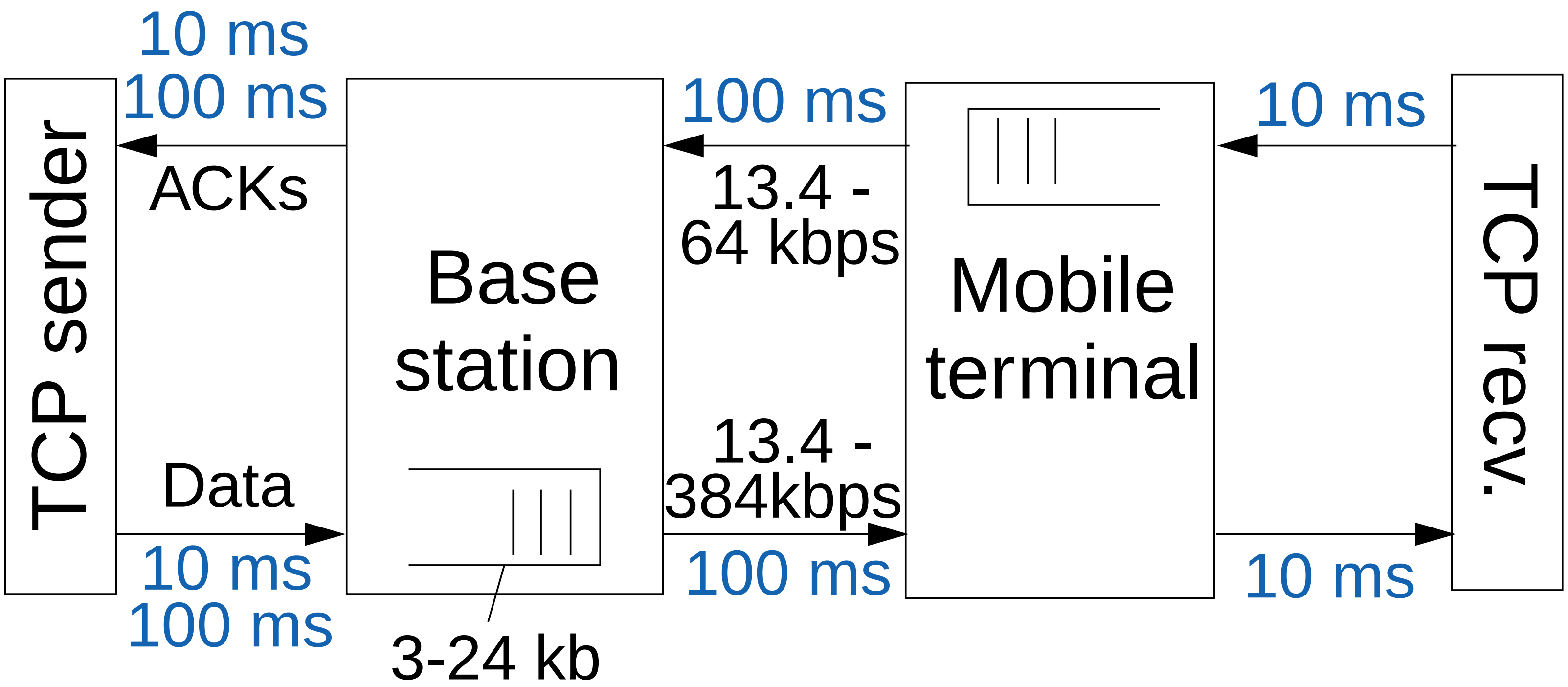




- ❒ **RTO characteristics highly depends on sampling rate**
- ❒ **Algorithm more aggressive if timestamps are used (*N* » 10)**
- ➥ **RFC 2988 and timestamps do not harmonize well**

Model validation (1)

Simulation setup

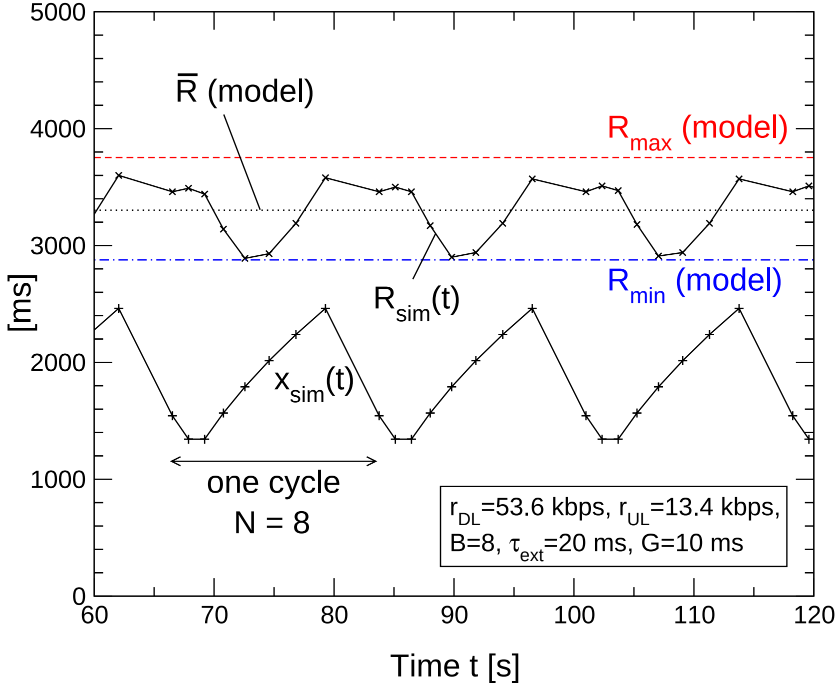


- **Various network configurations (GPRS, UMTS)**
- **Parametrization of RTT estimator**
  - With and w/o timestamps
  - Granularity 10 ms or 200 ms

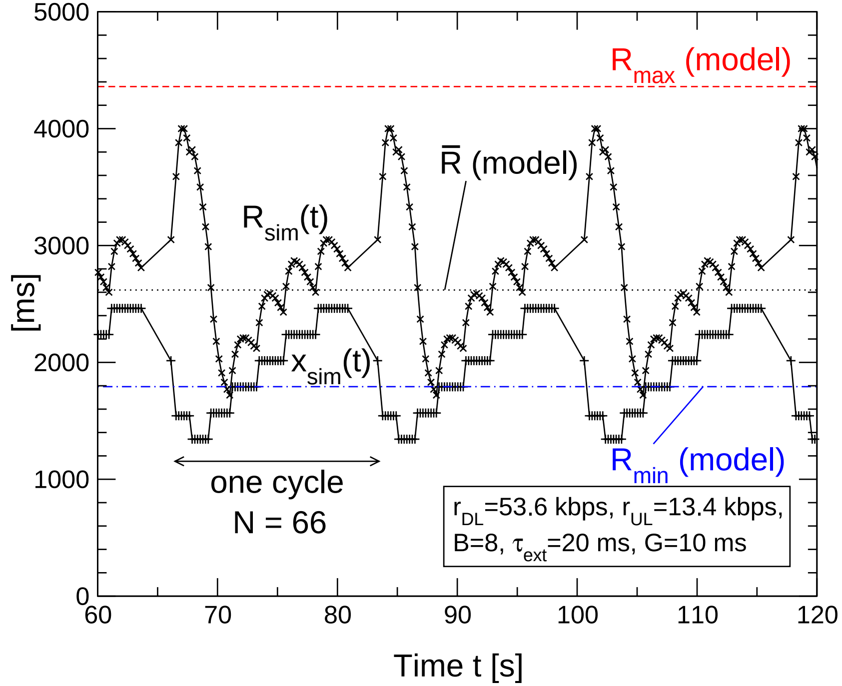
Model validation (2)

Selected examples

Without timestamps



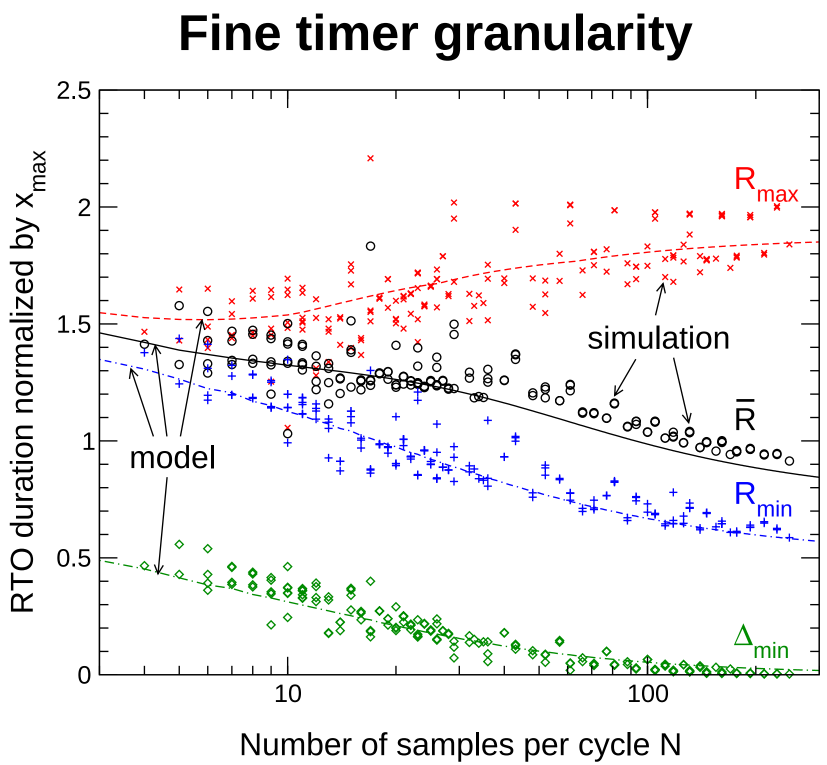
With timestamps



➥ **Model matches quite well to simulation results**

Model validation (3)

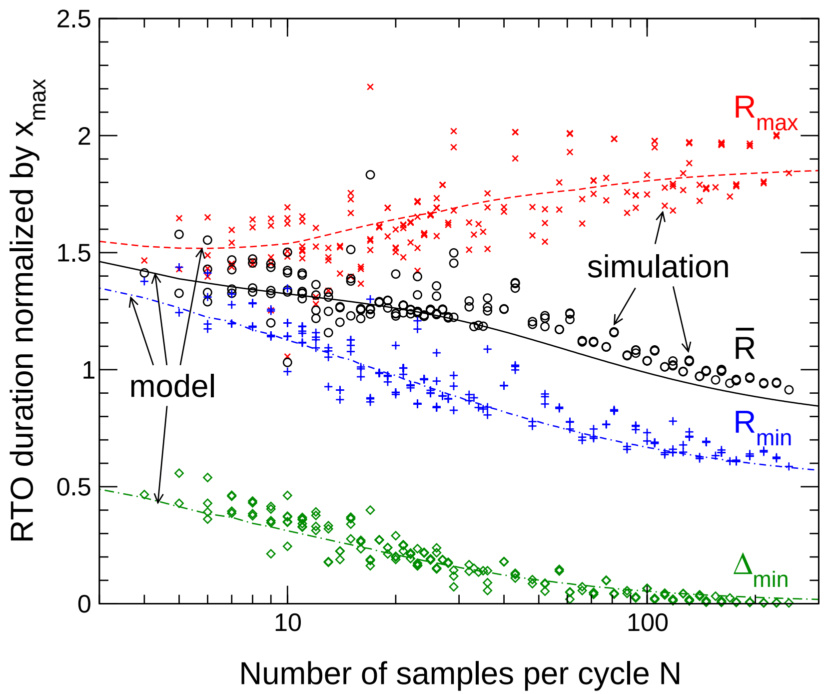
Different GPRS and UMTS scenarios



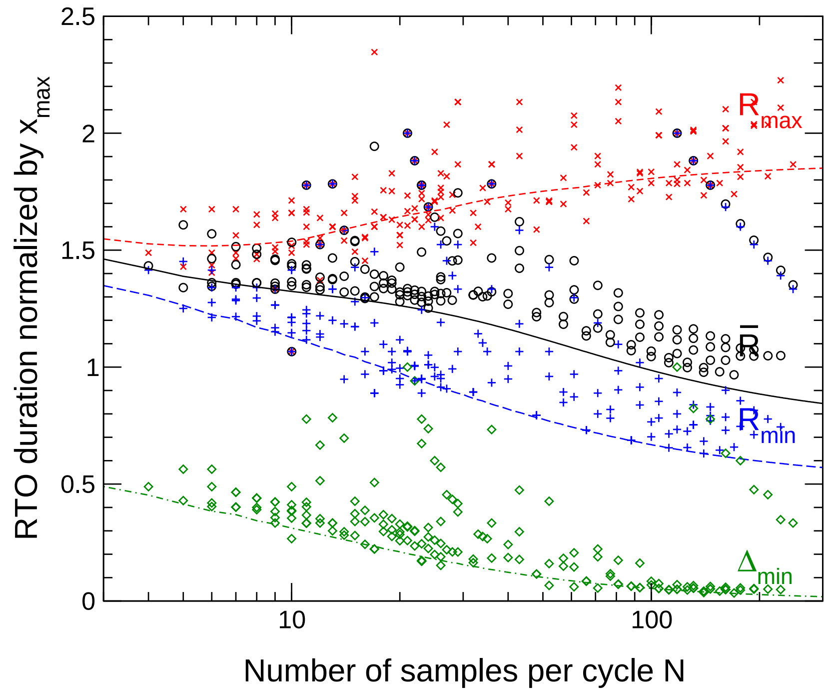
Model validation (3)

Different GPRS and UMTS scenarios

Fine timer granularity



Coarse timer granularity

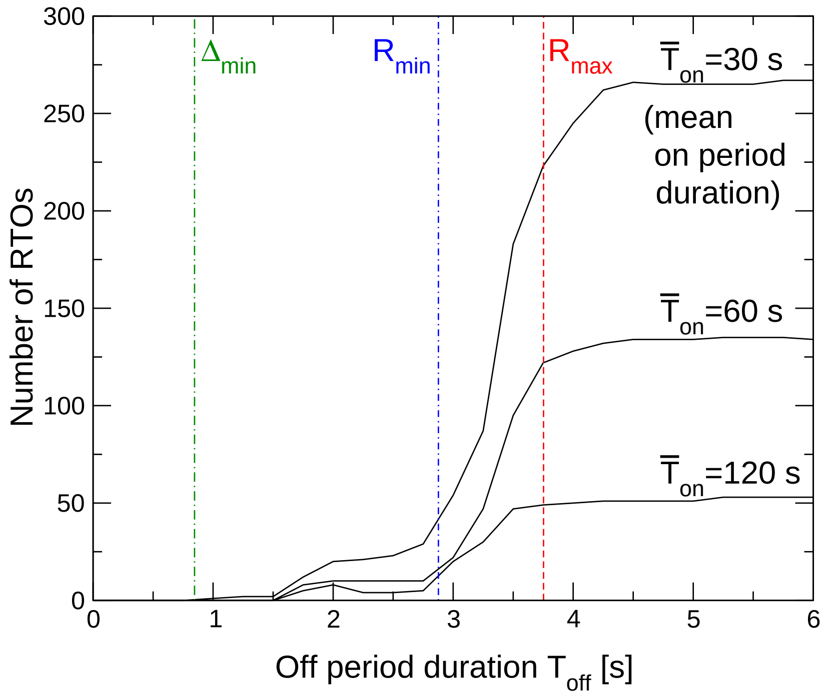


➥ **Discrepancies for coarse timer granularity**

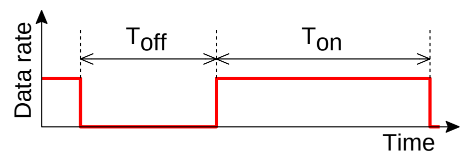
Model validation (4)

Performance degradation by spurious timeouts

Timeout probability



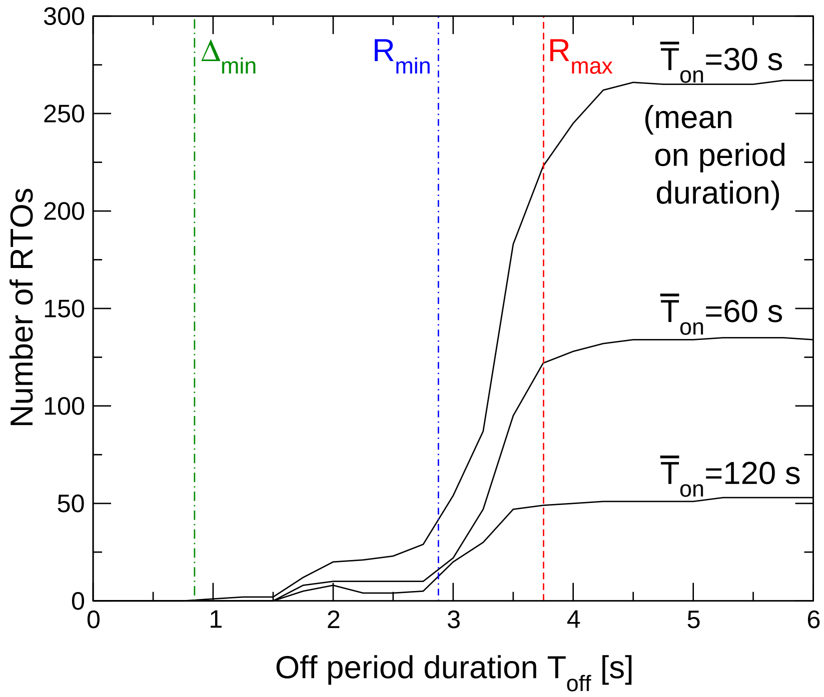
➥ **Spurious timeouts are triggered by off periods of duration** $T_{off} > R_{max}$



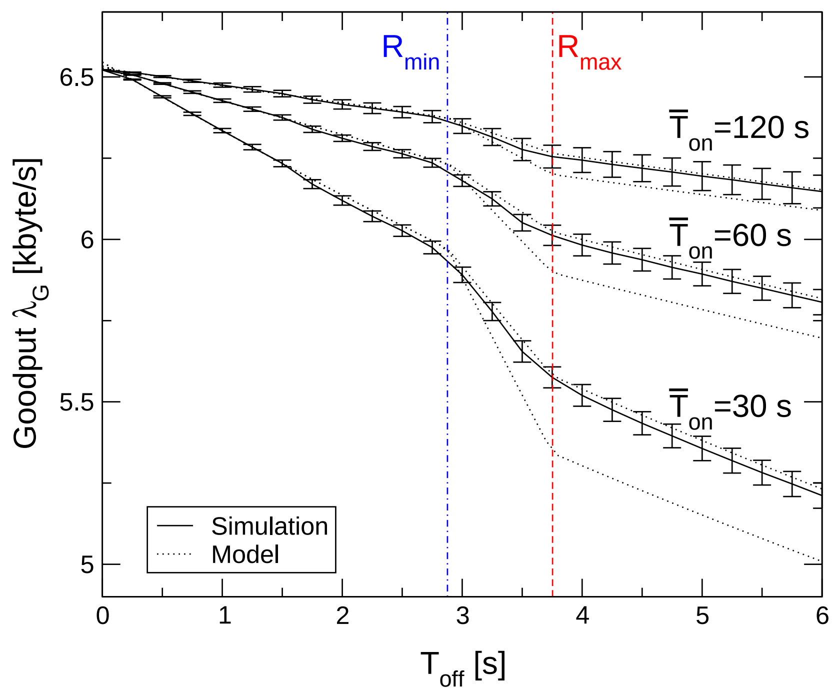
Model validation (4)

Performance degradation by spurious timeouts

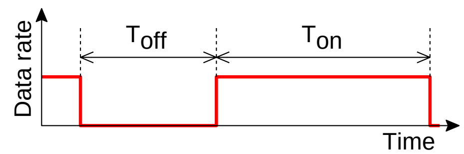
Timeout probability



Impact on goodput



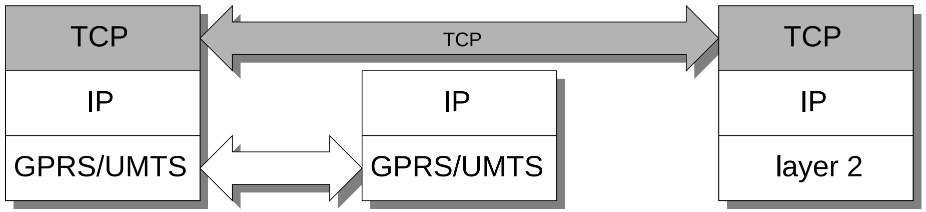
- ➥ **Spurious timeouts are triggered by off periods of duration $T_{off} > R_{max}$**
- ➥ **Impact only significant in case of frequent delay variations**



Optimization approaches

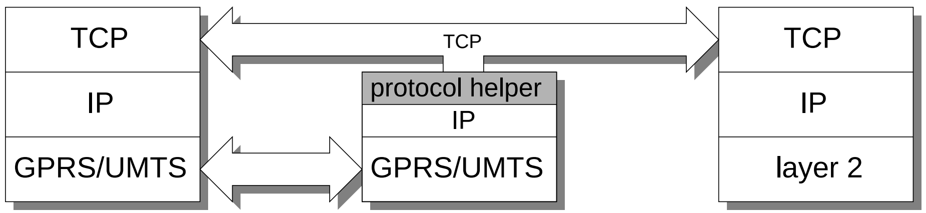
Architectures

Modification of TCP algorithms



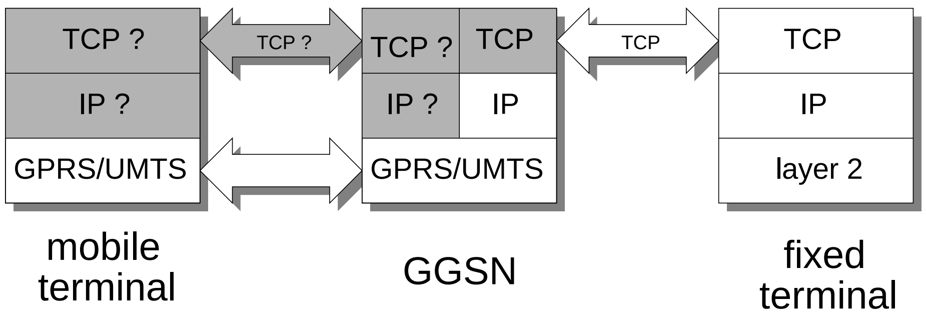
- Eifel Algorithm
- F−RTO
- D−SACK
- Freeze TCP

Protocol helper / protocol booster



- Spurious packet filter
- ACK Regulator
- ACK Buffering
- M−TCP
- Active queue management

Split connection



- numerous approaches
  example: WAP

Conclusions and future work

Conclusions

- ❐ **Mobile networks are characterized by variable delays**
  - ❍ Spurious TCP timeouts degrade performance
  - ❍ Sensitivity of TCP to delay variations depends on RTT estimator
- ❐ **Model for RTT estimation of TCP**
  - ❍ RTO duration as a function of path parameters
  - ❍ Significant impact of RTT sampling rate
  - ❍ Timestamps and RFC 2988 do not harmonize well
- ❐ **Quantification of risk of spurious timeouts**
- ➥ **Only delay variations of the order of seconds are critical**

Future work

- ❐ **Simulations with more sophisticated UMTS radio link models**
- ❐ **Verification by measurements**

The Impact of Delay Variations on TCP Performance

Michael Scharf
Institut für Kommunikationsnetze und Rechnersysteme
Universität Stuttgart
scharf@ikr.uni-stuttgart.de

ITG FG 5.2.1 Workshop, Hamburg
20. Feb. 2004

Optimization approaches (cont.)

- **Modification of TCP algorithms**
  - Sender-based
    - Eifel Algorithm (RFC 3522)
    - F-RTO (P. Sarolahti et. al.)
    - D-SACK (RFC 2883)
  - Receiver-based
    - Freeze TCP (Goff et. al.)
- **Protocol helper / protocol booster**
  - Filtering of segments
    - Spurious packet filter (Schüler et. al.)
    - Active queue management (Sågfors et. al.)
  - Manipulation of acknowledgments
    - ACK Regularor (Chan&Ramjee)
    - ACK Buffering (Huang&Chang)
    - M-TCP (Brown&Singh)
- **numerous split connection approaches**

Optimization approaches (cont.)

Eifel algorithm

- ❐ **Eliminates retransmission ambiguity by timestamps**
- ❐ **Can detect spurious timeouts**
- ❐ **Improved reaction to spurious timeouts**
  - ❍ No go-back-N
  - ❍ Congestion window restored
  - ❍ RTT estimation reinitialized