Context-aware Scheduling in Radio Access Networks

Capacity Sharing Workshop

Matthias Kaschub, Magnus Proebster, Thomas Werthmann, Christian Blankenhorn
matthias.kaschub@ikr.uni-stuttgart.de
Stuttgart, 2011-10-13

Universität Stuttgart
Institute of Communication Networks
and Computer Engineering (IKR)
Prof. Dr.-Ing. Andreas Kirstädter

Content

Motivation

Approach

- Achitecture
- Transactions
- Signalling

Evaluation

- Simulation Model
- Implementation

Conclusion & Outlook

Work partially sponsored by:

– BMBF Access 2.0 Project (ATOB Cluster)
– Bilateral Cooperation with Alcatel Lucent Bell Labs Stuttgart

Motivation

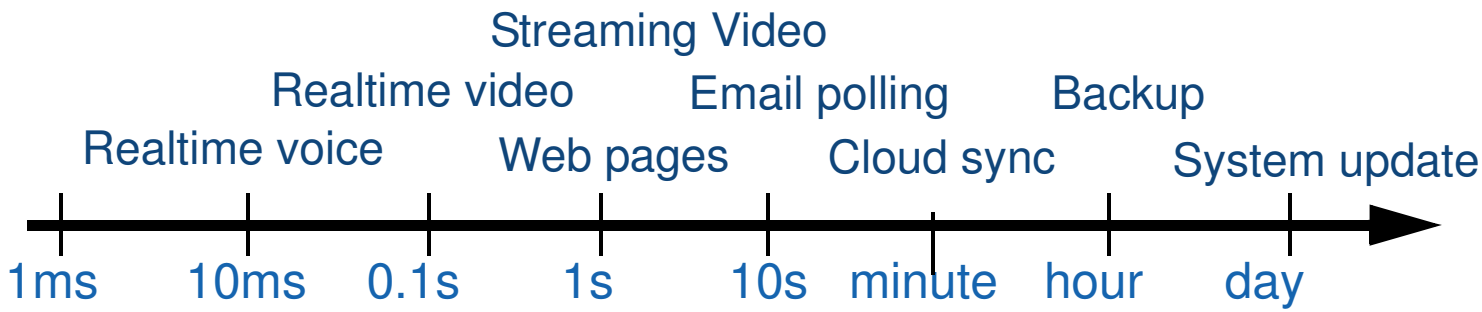
Current situation of (mobile) access networks

Sytem design

- Star-shaped
- Headend (BS) controls UL & DL
- Low aggregation: 1 .. 100 users

Traffic

- Heterogenious
- Bursty



User experience

- Depends on peak rate (and latency)
- Impaired already at low average utilization (3% .. 30%) **<< 100%**

Motivation

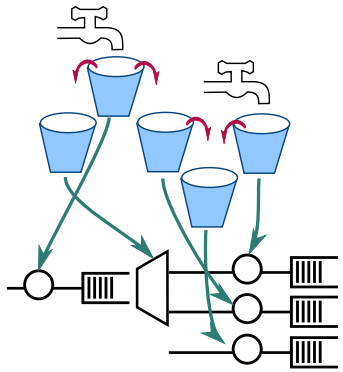
Approaches for improvement of situation

General

- Prioritizing urgent traffic
- Delaying traffic with relaxed requirements

→ Increases peak rate for the sensitive traffic

→ Allows higher average utilization of the network



Existing approaches

- Several "QoS" approaches exist
- Cooperation: Requires everyone on the internet

→ None has gained significant deployment

Our approach

- Aim: solve the capacity-sharing problem on the access link
- Cooperation: Only one operator + devices of his customers
- Not based on single packets

Access networks

Where do our degrees of freedom come from?

1.) Traffic

Some traffic can handle extra delays

→ Delay as a resource

2.) Aggregation of traffic with different requirements

- Dedicated line (e.g. DSL)
  - multiple applications (one user)
- Shared medium (e.g. DOCSIS, PON, WIFI, WiMAX, LTE)
  - *multiple applications .....*
  - applications of multiple customers

3.) Capacity variation

Radio access network (e.g. WiMAX, LTE)

→ Schedule prefereably when channel is good

Approach

Example: Web page

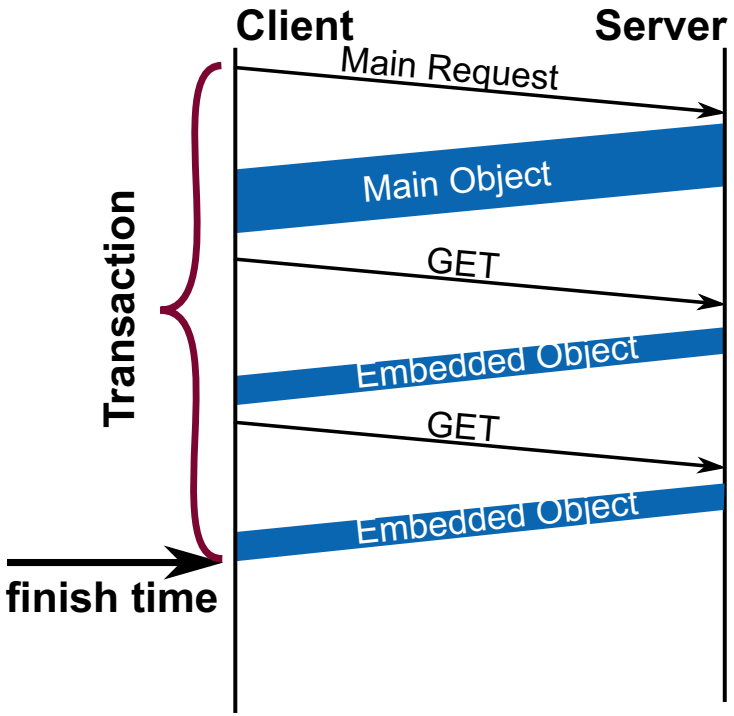
Definitions

Transaction

is all traffic that leads to a user-observeable result

Requirement

is a formal description of the users' expectation



Example

- Transaction: Web page with all embedded objects
- Requirement: display everything in 1s ("finish time")

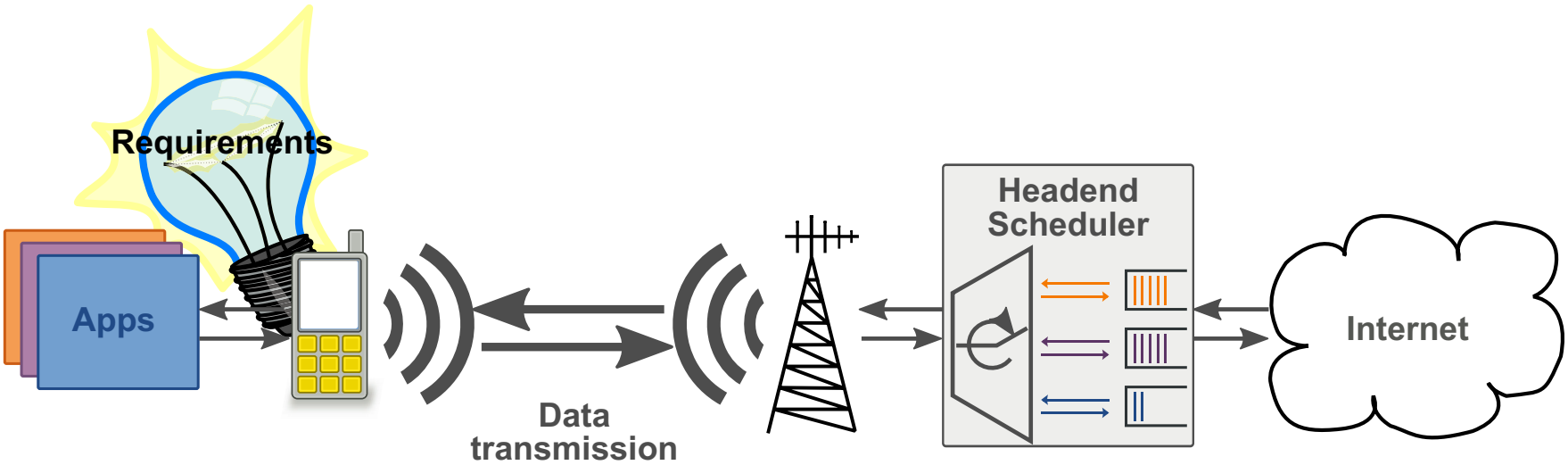
Characteristics

- Transaction
  - Consist of multipe connections, bursts, chunks
  - Connections might be reused (HTTP/1.1)
- User experience depends on when the last packet is delivered

→ **Approach tries to improve the Quality of Transaction**

Architecture

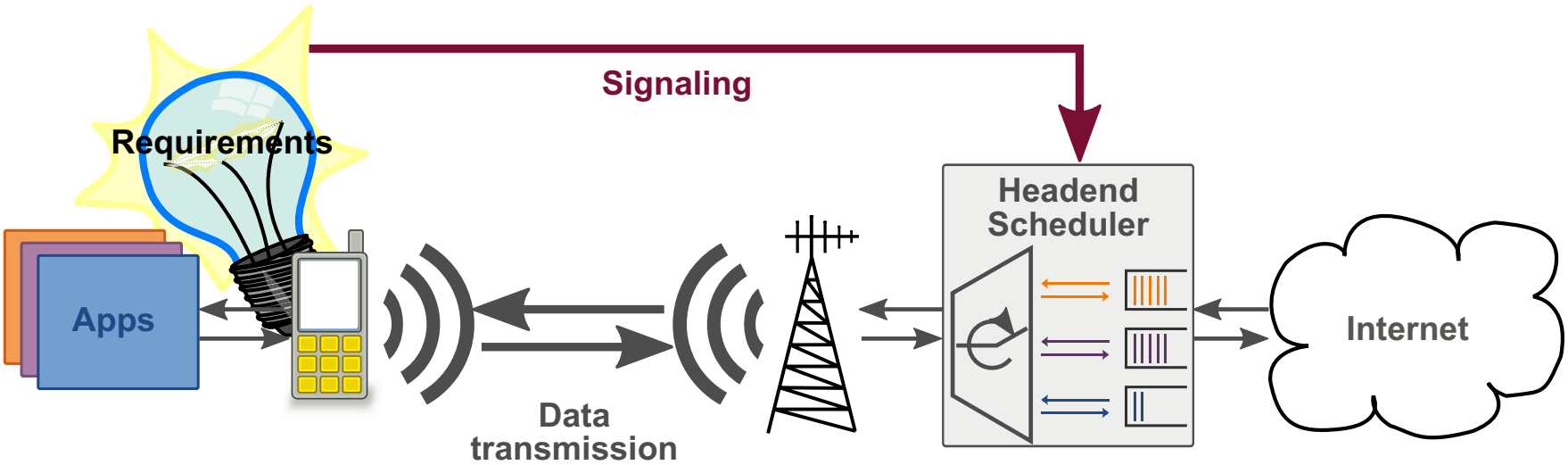
Overview



- User's device: Knowledge about transactions and requiements
- Headend: Scheduler, per transaction (access!)

Architecture

Signaling

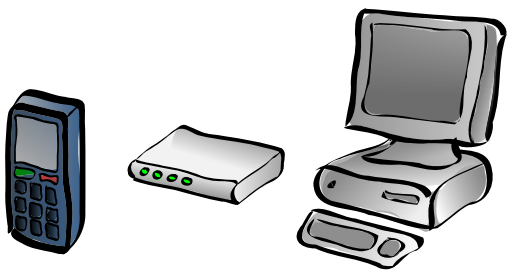


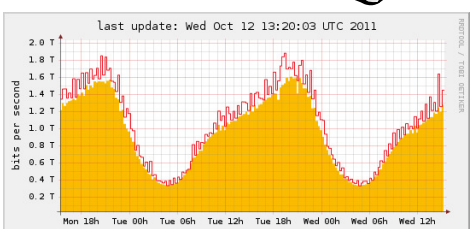
- User's device: Knowledge about transactions and requiements
- Headend: Scheduler, per transaction (access networks)
- Signaling: Unidirectional, from user to headend

Signaling

Where does the information come from

- **User**
  - explicit feedback
  - preferences, configuration
- **Applications**
  - type of application, transaction, priority, ...
  - activity (foreground tab?)
  - size of transaction (often estimation)
- **Plattform**
  - event source (click, timer)
  - parallel or interactive activity
  - sensible defaults for application values
- **Device / operating system**
  - screensaver, device orientation, proximity sensor
  - foreground / background
- **Network**
  - Current and future network load





Signaling

Protocol

General

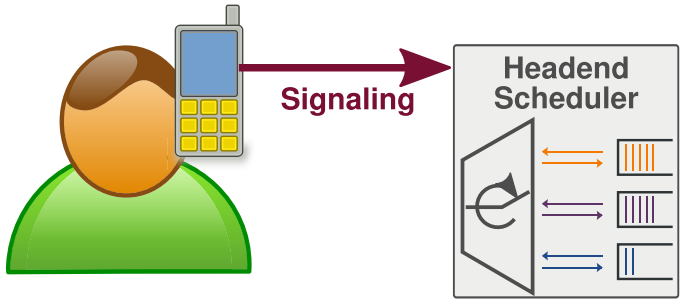
- From user to headend
- Contains: transaction description & requirements

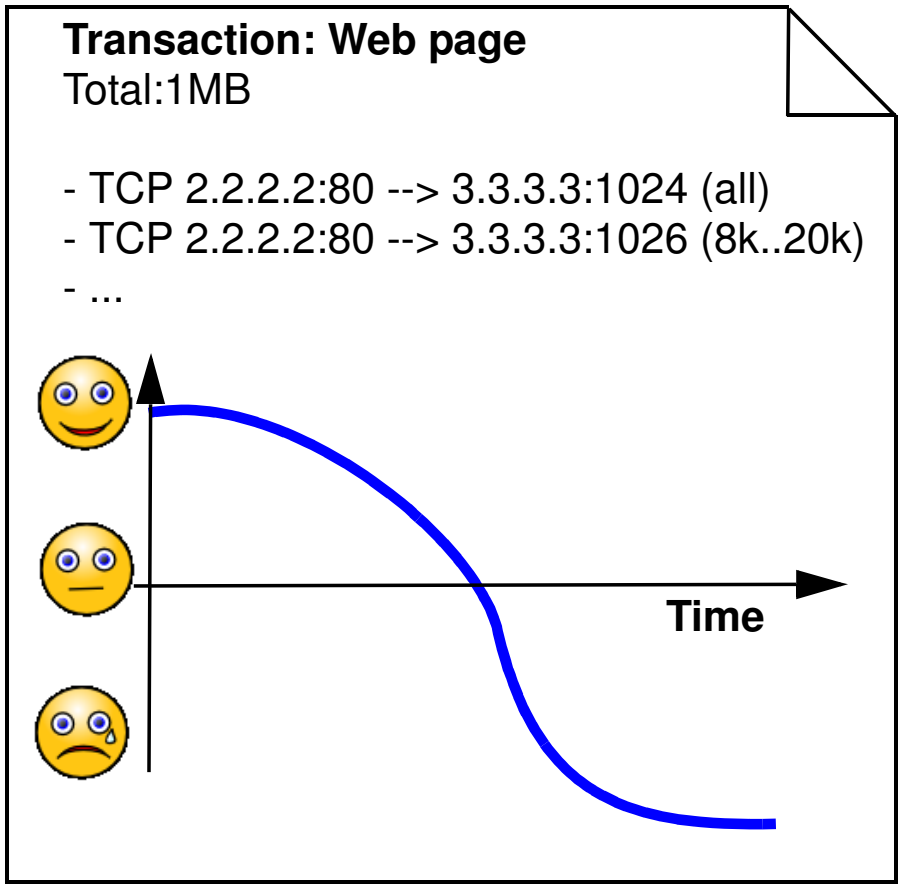
Transaction

- List of transport level connections (e.g. IP 5-tuple)
- Maybe only part of a connection
- Amount of data (for scheduler)

Requirement

- What is the user expecting
- Value (utility) of this transaction depending on finish time





Types of traffic

Traffic types and characteristics change over time

→ We search for common invariants

Realtime transaction

- Example: VoIP, Fußball-Bundesliga
- Requirement: Each packet has to be delivered before its deadline
- User experience: Depends on how often the **deadline** is violated

Streaming transactions

- Example: Youtube, VoD
- Requirement: receiver can buffer as long the average bitrate is sufficient
- User experience: whether **required bitrate** was met at all times (playout curve)

Finish time transactions

- Example: Web pages
- Requirement: "best effort"
- User experience: depends on when the **last packet** has been delivered

Evaluation

Simulation

Szenario

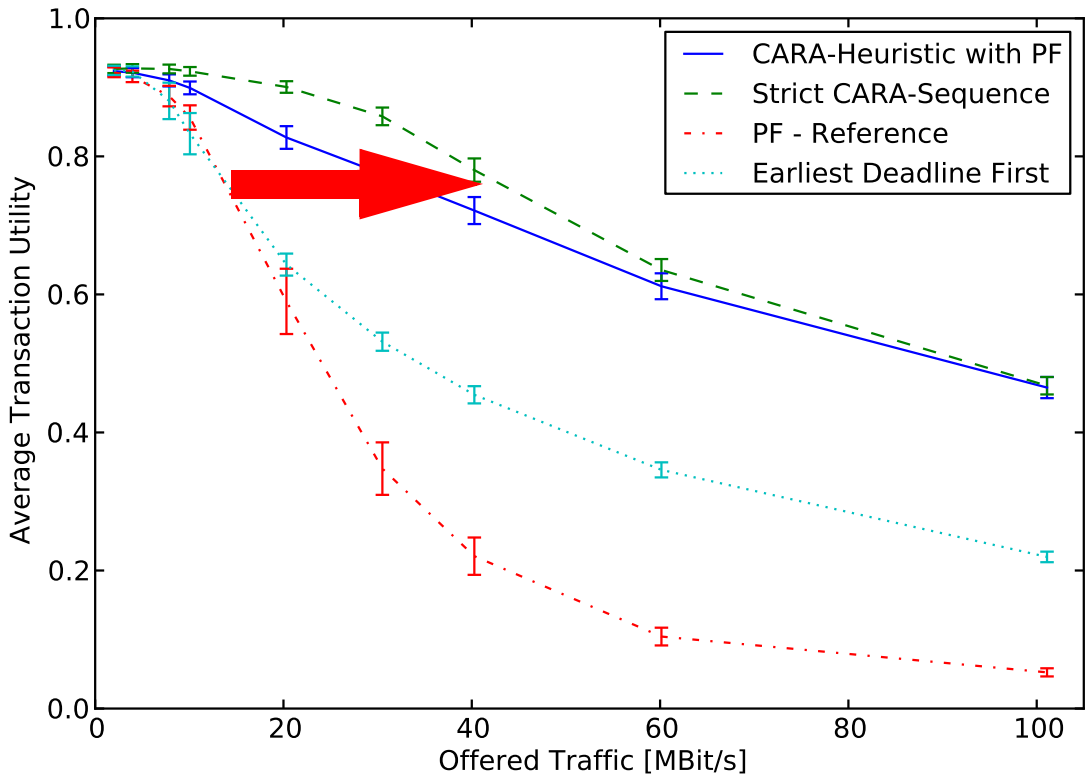
- Mobile access network, system level simulation
- Simple traffic model (3GPP Web Model)
- Direct & Combination with common Proportional Fair Scheduler

Preliminary results

- Improves finish times by reduced interleaving
- Handles >100% more traffic at the same Utility level

➨ High potential of schedulers with new transaction framework!

Published on ICC2011, Kyoto, June 2011



Evaluation

Implementation

Map-viewer for OpenStreetMap

Modifications:

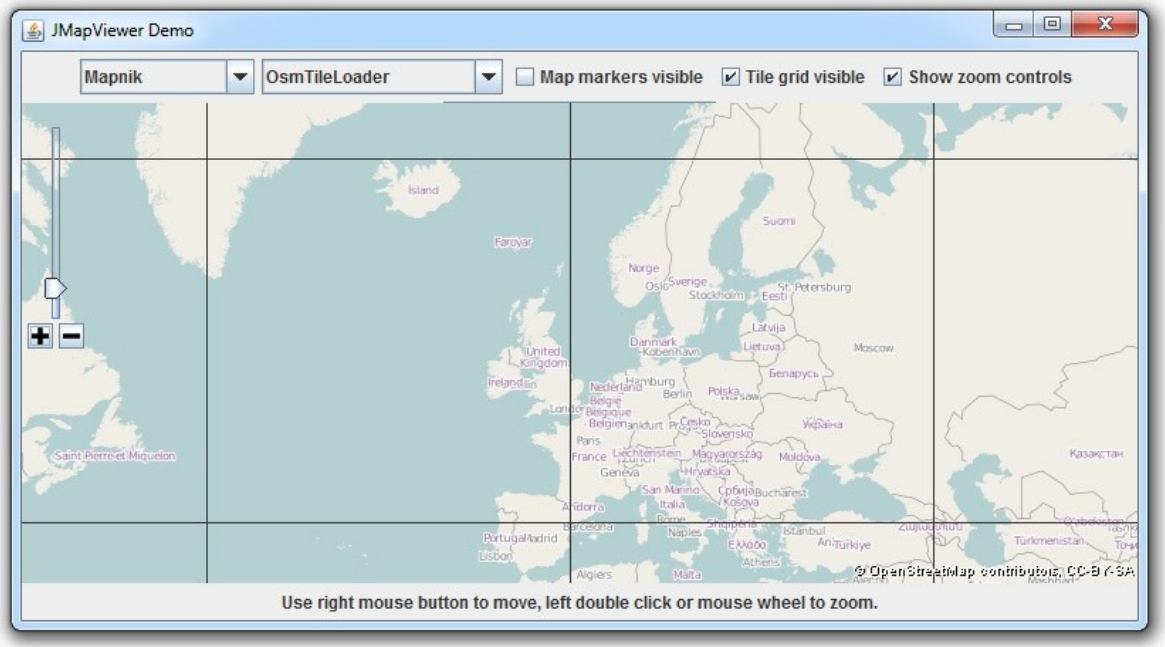
- Signal importance of each tile
- Importance depends on distance from center
- Student project

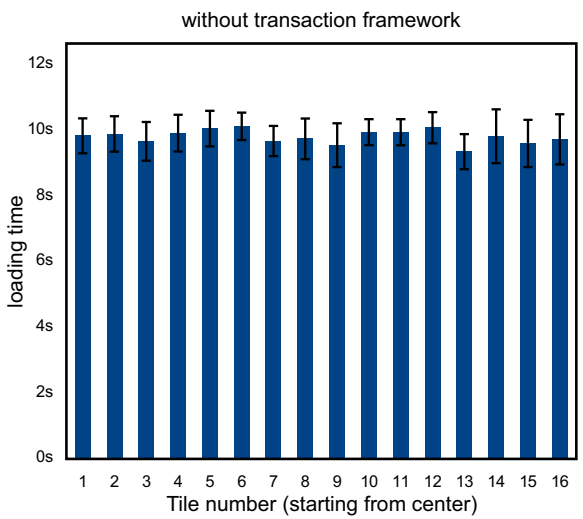
Headend

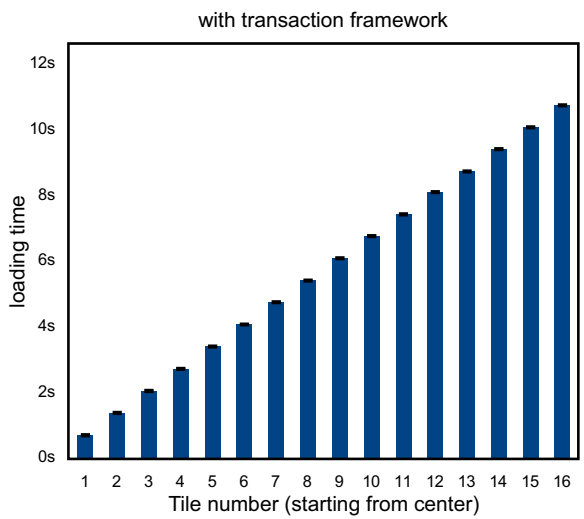
- TCP-Proxy to avoid TCP effects
- Simple scheduling algorithm
- Student project

Result

- Works as expected
- Center tiles load first







Student project: Kasten Schöck: "Verkehrspriorisierung in IP-Netzen mittels Anwendungswissen", 2011

Traffic Modeling

Traffic models are crucial for the evaluation of such approaches

- Performance depends on heterogenious mix
- More delay-insensitive Traffic -> more gain

Current evaluations

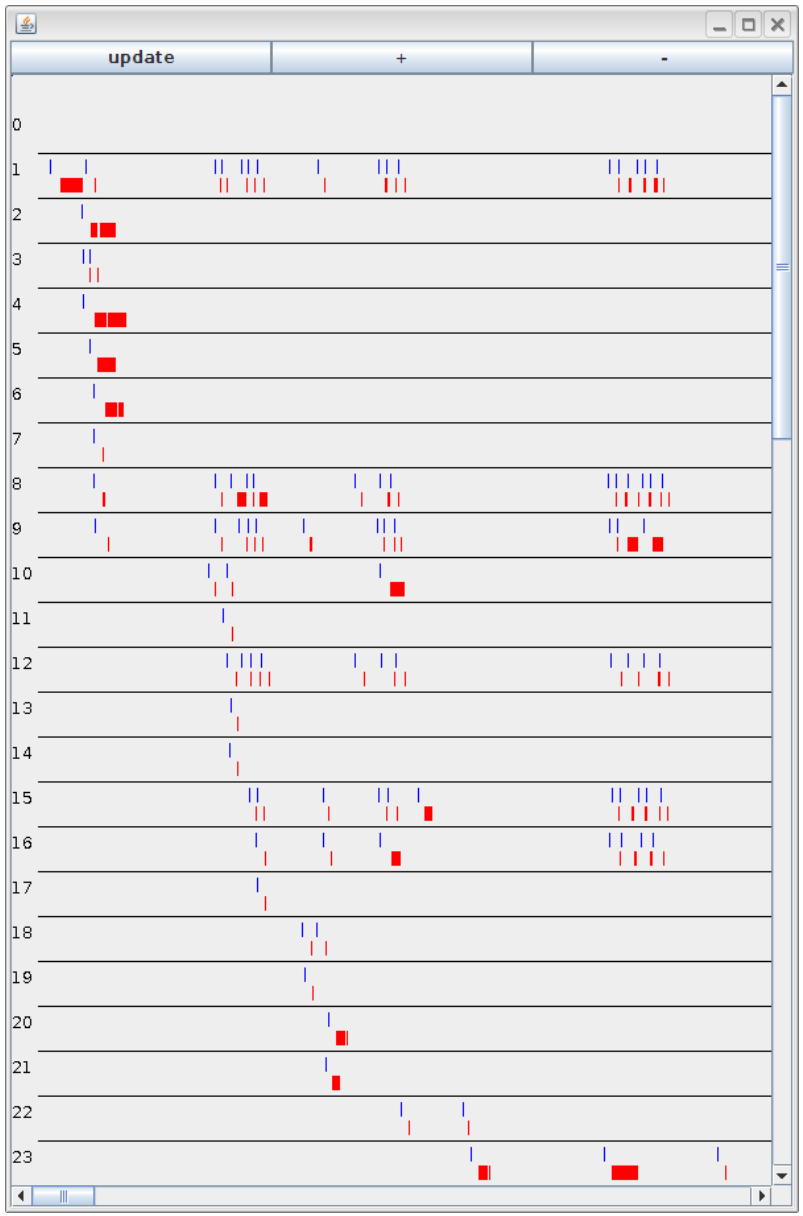
Simple models (3GPP Web Model)

Requirements

- Unaggregated traffic
- Model of user & application behaviour
- Mix of applications
- Including users' expectations

Current activities

- Lab measurement, identifying transactions
- Identifying invariant patterns
- Creating models



Conclusion and Outlook

Approach for better capacity-sharing in (mobile) access networks

- Involved entities: one operator and (some of) his customers
- Based on transactions
- User (or his apps or his plattform) signal the requirements to headend
- Headend (scheduler) prioritizes urgent transactions

Evaluation

- High gains with simple & synthetic traffic models
- Probalbly even higher gains with more heterogenious traffic mix

Next steps

- Traffic models
- Modifying more applications
- Modifying Android plattform