Measurements and Models of a 3G Network's GGSN

Florian Metzger

University of Vienna
Faculty of Computer Science

DOC Forschungsseminar
2013/11/13

Outline

Relation to Thesis

Thesis will consist of 2.5 parts:

1. Investigation of TCP-based video streaming techniques
   - Protocol survey and classification
   - Deriving a model
   - Measurements with the model
2. Evaluation of a 3G core network
   - Investigation and evaluation of the control plane
   - **Modeling and simulating load**
3. Measuring video streaming in a 3G network

Presentation based on MMB'14 submission *"A PDP Context Load Model and Virtualization Gain for a Mobile Network's GGSN"*

Motivation

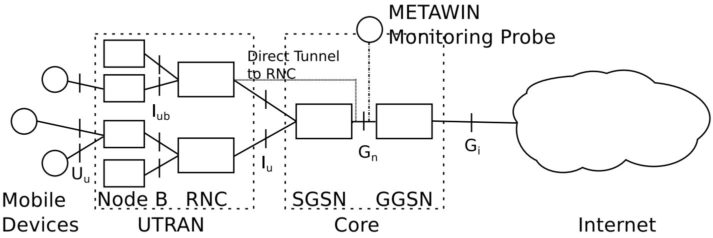
Mobile network planning and dimensioning today

- solely based on expected user traffic
- good algorithms and tools for placing radio towers and planning radio propagation
- core network and control plane usually not given much consideration

Our approach

- presents models for a GGSN in the core network
- models fed with data from a real network
- simulates the model

GTP Tunnels



Tunneling concept

- Any user traffic in a 3G net is encapsulated into tunnels
- GTP protocol between SGSN and GGSN
- Tunnel state (PDP Context) held at nodes
- Tunnels created, deleted, and updated based in reaction to user
- GGSN involved in all GTP signaling

The Dataset

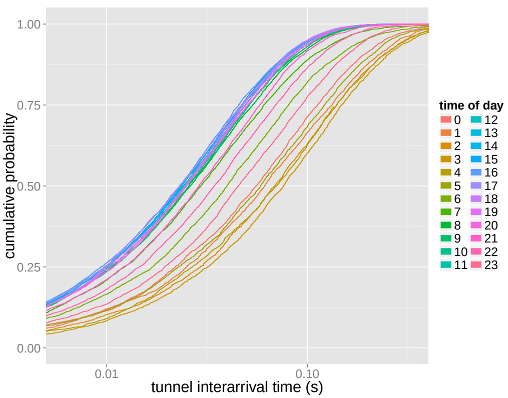
- Passive measurements in the core network of a operator (METAWIN)
- One week long, anonymized (April 2011)
- 2.2Bn user traffic records, 410M GTP tunnel management messages
- Specific properties make distribution modeling hard

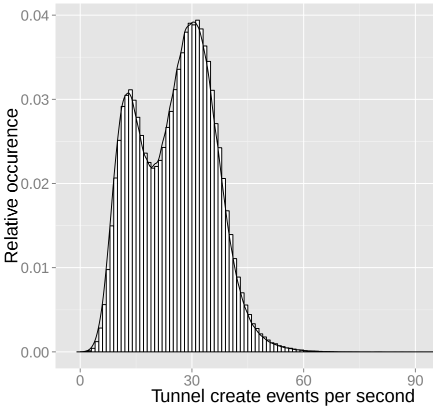
Evaluated properties important for the model

- arrival rate of new tunnels
- duration of tunnels
- and diurnal influences on them

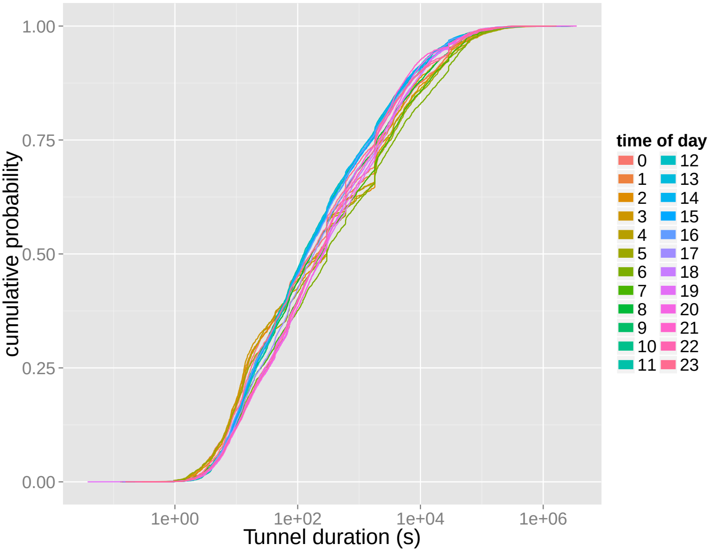
Arrival Process

Interarrival Time of Successful Tunnel Requests

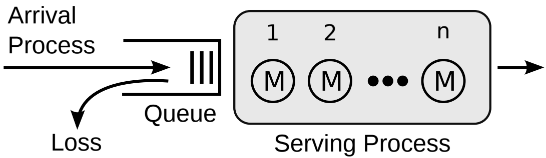




Tunnel Durations



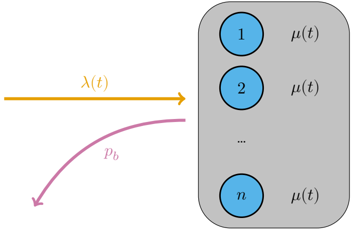
Queuing Models



Described by Kendall's Notation $A/S/c/q$

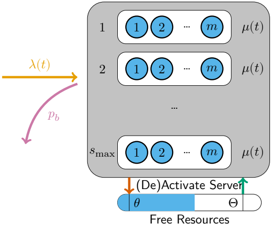
- Distribution of the arrival process A
- Distribution of the serving time S
- Number of Servers c
- Queue Length q
  - $q = \infty$ no loss will occur
  - 0 loss/blocking system, no queue
- Evaluate
  - Average queue length and server occupation
  - Blocking probability

Traditional GGSN Model



- Poisson tunnel arrival process with rate $\lambda(t)$, adjusted for the time of day
- GGSN server can serve n tunnels in parallel (limited by state / processing overhead)
- each with a serving time (tunnel duration) of $\mu(t)$
- If server is full, reject new tunnel requests
- Non-stationary Erlang loss model $M_t/G/n/0$ with blocking probability p_b

Virtualized GGSN Model



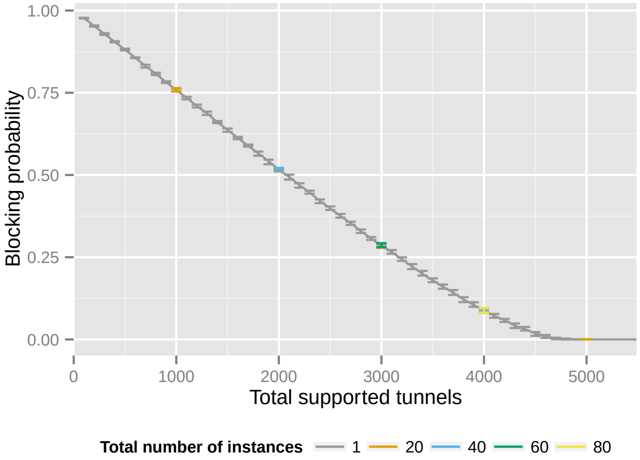
- Same arrival and serving time process, no queue
- Hypervisor distributes tunnels and starts virtualized GGSN instances on demand
- Up to s_{max} instances with capacity of m each
- Instance count kept near actual system load (energy efficiency)
- Additional blocking, when new instances are not switched on fast enough

Simulating the Model

- $M_t/G/n/0$ model, no exact mathematical solution available
- Use queuing simulation instead of stationary analysis
- SimPy3 based discrete event simulation[1]
- Arrival process with normalized exponential distributions fitted to dataset, four time of day slots
- Tunnel durations fitted with rational functions
- Scenario variables: n and s_max
- Evaluate and compare both models based on
  - Blocking probability
  - resource and instance usage

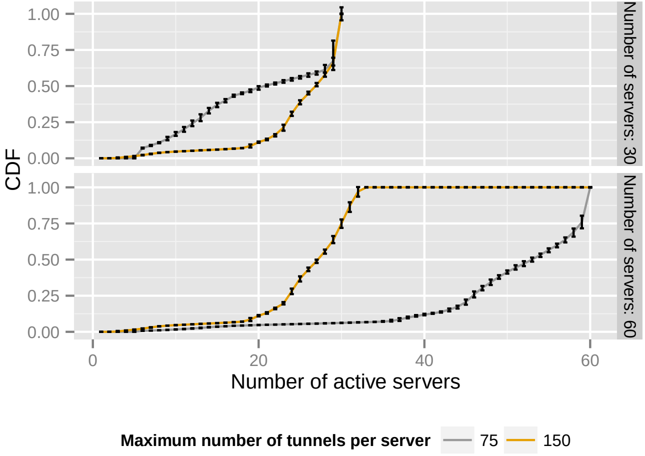
[1] `https://github.com/fmetzger/ggsn-simulation`

Blocking Probability



- Traditional and virtualized scale equally with supported tunnels

Virtualized GGSN Resource Usage



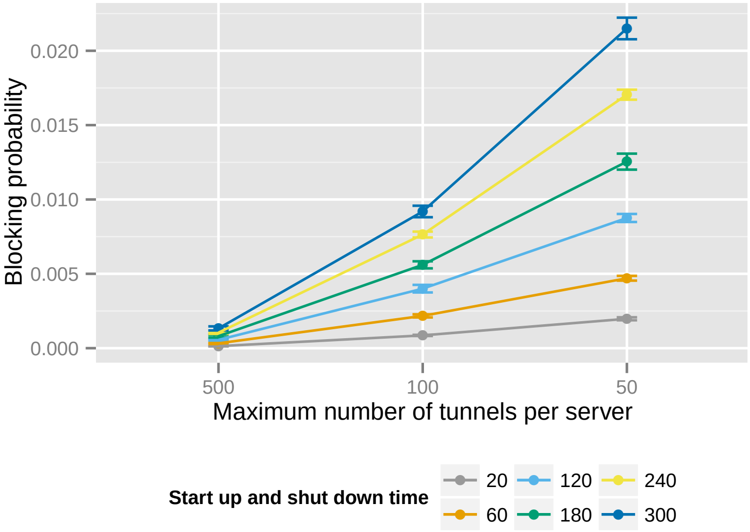
- System scales both up (tunnels/instance) and out (instances)

Conclusion

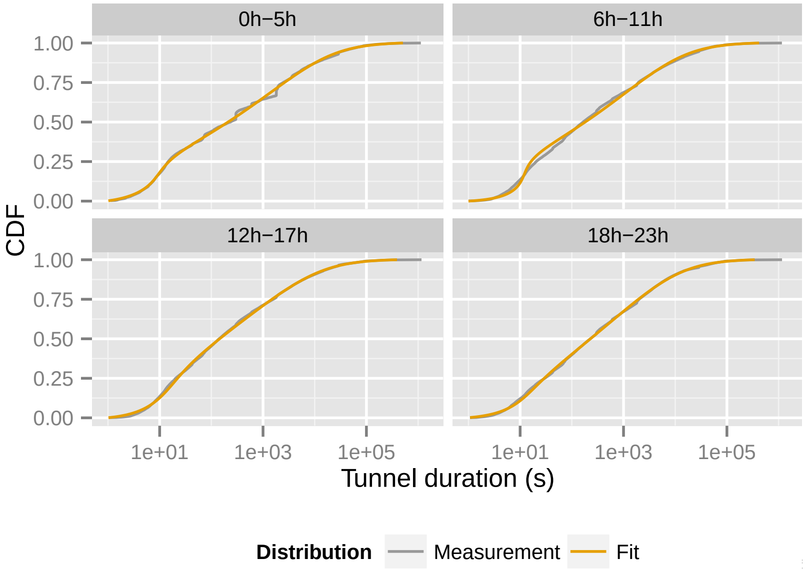
- Investigated dataset for tunnel properties
  - Non-stationary poisson-arrivals
  - Tunnel duration with general distribution fitted to a rational function
- Two novel models for tunnel load at a mobile core network's GGSN
  - Traditional GGSN representing today's makeup
  - Virtualized GGSN proposal with improved scalability and efficiency
- Simulative evaluation of the model
- Enable mobile network dimensioning based on tunnel blocking rate instead of only user traffic volume

Thanks!

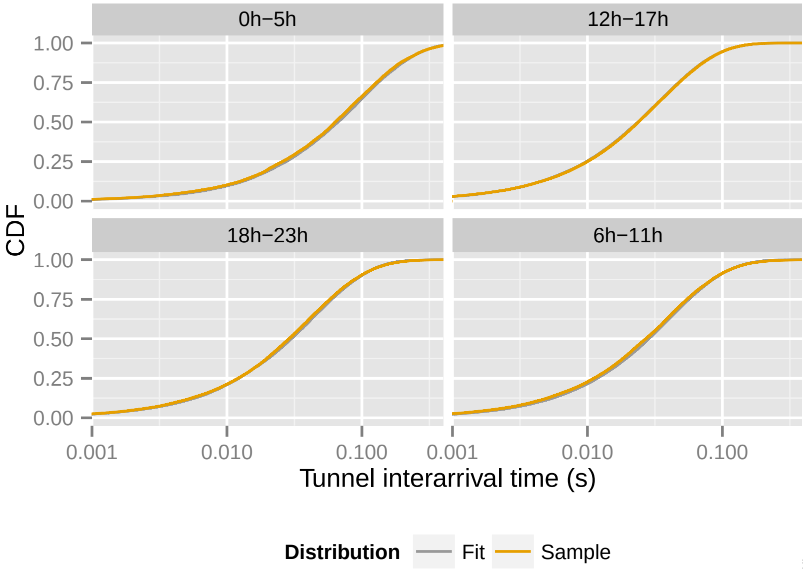
Questions?


0.020
0.015
0.010
0.005
0.000
Blocking probability
500
100
50
Maximum number of tunnels per server
Start up and shut down time
20
60
120
180
240
300

Exponential Arrival Process Fits



Serving Time Rational Functions Fit



Scaling Up or Out with a Virtualized GGSN