GTP-based Load Model and Virtualization Gain for a Mobile Network's GGSN

Florian Metzger
University of Vienna, Austria
Future Communication Research Group
florian.metzger@univie.ac.at

Christian Schwartz and Tobias Hoßfeld
University of Würzburg, Germany
Chair of Communication Networks
christian.schwartz,tobias.hossfeld@informatik.uni-wuerzburg.de

***Abstract*—Multiple outages in major mobile networks have been reported in the recent past. In fixed and datacenter networks such capacity problems are solved by scaling out, i.e. purchasing additional hardware. In mobile networks this is not as easily possible as network components are usually sold as sealed middleboxes. With the advances in server performance and SDN it has been suggested to virtualize these boxes. This also opens up opportunities to dimension according to current load and save energy by switching off parts of the infrastructure.**

Such suggestions immediately raise questions on the cost of virtualization. To answer this, we introduce models for both a traditional as well as a virtualized GGSN. In addition, we provide distributions for the load experienced at the GGSN based on network measurements. With this at hand, we study the influence of different dimensioning parameters on important performance metrics, with special consideration for the impact of provisioning new instances for the virtual GGSN.

I. Introduction

With the increased importance of smart phones, mobile networks are currently experiencing rapid growth. Compared to a fixed access provider additional aspects have to be taken into account when dimensioning a mobile network. First and most prominent is the planning of radio access cells — their coverage, frequency selection, and backhaul, i.e., the connection to the operator's network. Radio network planning research and tools readily exist to help solve this problem [1]. Albeit of equal importance, there is much less public knowledge and research on the second aspect in setting up the mobile network: setting up and dimensioning the core network. Consisting of a large number of specialized network nodes in need of careful tuning to each other, correctly putting together the core is no small feat. The reason for this is the large number of services incorporated into the protocol stack — e.g., authentication, accounting, or monitoring — and the amount of state, that needs to be held and signaled throughout the network, coming with it.

One major metric to consider in this core dimensioning is the number of supported tunnels, i.e., connections to the Internet, of the Gateway GPRS Support Node (GGSN). The GGSN's performance depends on factors like customers to serve, applications in the network, user behavior and devices used. During dimensioning, these factors are either unknown or subject to change as user behavior evolves. But these network components are sold as static middleboxes and cannot not be easily extended with of-the-shelf hardware in order to account for new requirements. The newly introduced concept of Network Function Virtualization (NFV) [2] suggests to harness technologies from cloud computing in the network. This would allow network operators to scale out, i.e., using additional low performance machines, instead of scaling up, which requires them to replace existing hardware with more powerful components.

The contribution of this work is threefold. First, we introduce models for both a traditional GGSN as well as a virtual GGSN using NFV. Second, we provide distributions for GPRS Tunneling Protocol (GTP) tunnel interarrival times and durations. Finally, we study performance trade-offs when using a virtual GGSN, discussing different options to consider when using a virtual GGSN.

This paper is structured as follows. Section II gives a brief explanation of the involved 3G infrastructure and general behavior of mobile networks. An overview of the related work is also included. We present our two models in Section III. Afterwards, Section IV consists of a short description of the dataset that was used and the relevant evaluations and conclusions drawn from the data. Afterwards, we evaluate the numerical results and implications of the queuing simulation in Section V. The paper concludes in Section VI.

II. Background

This section provides the necessary background on topics important for the remainder of the paper including General Packet Radio Service (GPRS) basics and related work.

A. GPRS & UMTS Fundamentals

UMTS is specified by the Third Generation Partnership Project (3GPP), with relevant parts for this investigation found in Technical Specification (TS) 23.060 [3], which defines the network's basic aspects involving GPRS protocols and its system architecture, and TS 29.060 [4], describing the specifics of GTP across the Gn and Gp interfaces.

The Serving GPRS Support Node (SGSN) and the GGSN are the main components in the core's packet switched domain. The SGSN serves mainly as mobility anchor, the GGSN represents the gateway to the public Internet and is responsible for most connection and transmission related management. All user traffic between these nodes is encapsulated in a tunnel and managed with explicit GTP signaling.

Tunnel state is kept in the SGSN and GGSN as Packet Data Protocol (PDP) Context data structures. These contain various information, such as the device's IP address, International Mobile Subscriber Identity (IMSI), and a tunnel identifier. Usually, any user-plane IP traffic is transported within a primary "best effort" tunnel. The GTP signaling, responsible for the context management interactions, contains procedures for managing data paths, Mobile Station (MS) locations, mobility, and, of course, tunnels. Of relevancy to this paper are the tunnel management request/response message pairs involved in the maintenance of PDP Contexts.

B. Related Work

This work is a continuation of our previous evaluations conducted in [5], [6]. Besides these, there is to our knowledge no other directly preceding literature to this paper's novel models. Still, efforts have been made to investigate the special properties of mobile networks and its traffic. These include attempts to infer control plane behavior through active measurements at the mobile device or synthetic traces and investigations of user traffic characteristics by means of real 3G core network traces. The authors of [7] discuss cross-layer interactions in mobile cellular networks and the consequences for device energy consumption and radio channel allocation efficiency.

Looking at the multitude of radio network control state machines, we find in [8] some simple yet effective application layer methods investigating transitions of these state machines.This is further elaborated on by [9] in order to analyze the radio signaling load and thus power efficiency from several different applications.

Having access to core network datasets, the authors of [10], [11] both take the approach of looking at high-level user traffic characteristics, focusing on temporal and spatial variations of user traffic volume and investigating the influence of different devices on this metric. Additional user flow and session traffic metrics are being studied in [12] with the conclusion that, in comparison to wired traffic, short flows are occurring more frequently. In 2006, a core network measurement study of various user traffic related patterns was conducted, providing an initial insight into PDP context activity and durations [13].

III. Model

In this section we provide a model for a traditional GGSN and discuss a model for a virtual GGSN using NFV. In NFV [2] static network middleboxes are replaced by commodity hardware. The tasks solved by the original middleboxes are then handled by dedicated software. The generic queuing theoretic model is based on observations drawn from the measurement set provided in Sec. IV. As such, any properties outside these observations are not reflected.

While internally a *traditional* GGSN may consist of multiple individual servers, it acts as a monolithic entity from an outside point of view. Therefore, idle portions of it can neither be deactivated nor reused for other purposes. This first model is based on this monolithic idea.

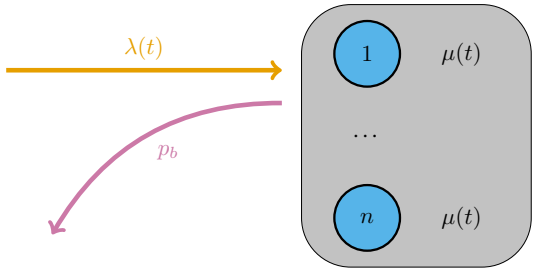

Fig. 1: Model of a Traditional GGSN

The queuing theory equivalent to this model is displayed in Figure 1. New tunnels requests arrive according to a Poisson distribution with a rate of $\lambda(t)$ at the GGSN, which has a maximum tunnel capacity of c_c. When the capacity is reached, blocking will occur and newly incoming tunnels are rejected. Traditionally, GGSNs can be expected to be overdimensioned in such a way, that this rarely happens. If an incoming tunnel request is accepted, one of the GGSN's serving units will be occupied for the tunnel's duration of $\mu(t)$. The duration is assumed to be of an arbitrary non-Markovian service time distribution. Together this results in a non-stationary Erlang loss model, or $M(t)/G/c_c/0$.

In order to give QoS guarantees the network operator is interested in the system's blocking probability p_B, which we consider to be a key metric of our model. Additionally, the previously described diurnal patterns can also be modeled by adjusting the arrival and serving process distributions for each time of day. This alternatively also allows just to investigate the busy hour and thus the system's peak load.

A. GGSN using Network Function Virtualization

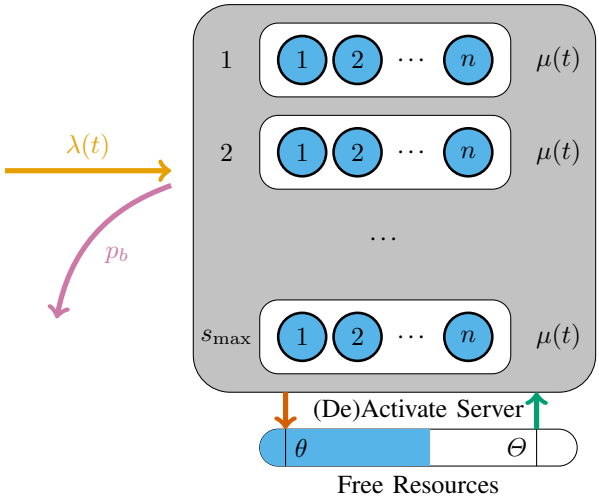

Fig. 2: Model of a GGSN using Network Function Virtualization

In the second model, we introduce concepts from NFV, i.e., the idea to replace middleboxes with commodity hardware. This allows us to realize benefits from cloud computing, as we are now able to scale out instead of up. The assumptions

of the Markov arrival process $\lambda(t)$ and the serving time distributions $\mu(t)$ are carried over. However, instead of one server processing every tunnel, this model assumes that there are up to s_{max} virtualized servers s_i. Each of these can be much smaller than the traditional GGSN, having a tunnel serving capacity of $c_i \ll c_c$ and a total system capacity of $c_{max} = s_{max} \times c_i$.

To increase efficiency all but a small portion of the server instances can be initially turned off. Only when a certain condition is reached, a new one needs to be provisioned. For example, one could always hold one instance in reserve for upcoming requests and provision as soon as the reserver gets used. Similar rules should apply in the shutdown of servers and should form a hysteresis together with the boot condition.

If these conditions are not carefully selected and are in tune with the expected boot time of an instance, additional blocking could occur. Despite not having reached its maximum capacity, this system will still reject tunnel requests during the provisioning phase when no tunnel slots are free. This could be remedied by a request queue. However, this makes the system more complex without providing real benefit, as mobile devices usually just repeat their attempts when the request is taking too long.

To place incoming tunnel state on one of the available servers and manage the servers a load balancer or hypervisor is required. To ensure, that the system can scale down to its actual needs, the balancer should place tunnels on servers, that are the fullest, keeping the reserve free. It may even migrate tunnel state from almost empty servers away so that these can be shut down, when certain conditions are fulfilled. Keeping instance close to their capacity should also have no impact on the performance a mobile device associated to a specific tunnel experiences. Adequate strategies for both load balancing and migration will be considered in future work.

IV. The Data

In order to evaluate the newly introduced models we use data collected from a nation-wide mobile operator. This allows for precise core network evaluations and the creation of statistical fits for the observed processes. In this section we first describe the dataset used for the evaluation and afterwards we derive the random variables required for our models.

A. Dataset Description

All data was collected by the Measurement and Traffic Analysis in Wireless Networks (METAWIN) monitoring system [14] with measurement probes located at the Gn interface within the core network, enabling broad access to signaling. For this investigation we exclusively use GTP protocol data which was fully anonymized to meet privacy requirements.

The dataset used in our evaluation is a week-long trace from the third week of April 2011. It contains 410 million GTP tunnel management transactions, and was tapped at one of the operator's GGSN locations, handling about half of the operator's total traffic volume in this period.

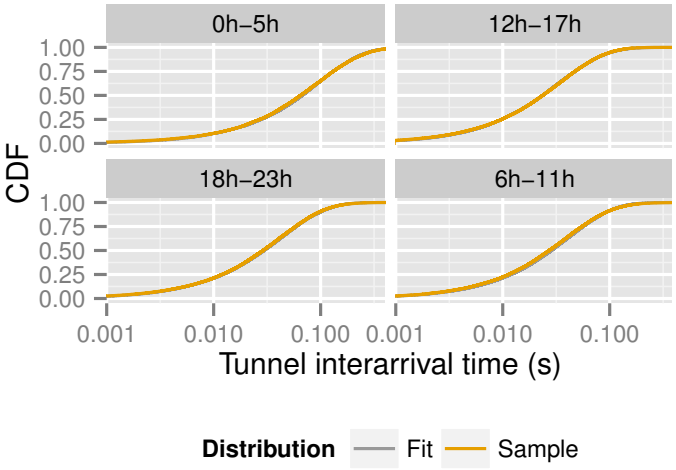

Fig. 3: Empirical and exponentially fitted CDFs of the tunnel interarrival duration by time of day. CDFs are overlapping as the coefficient of determination is close to 1.

B. Statistical Evaluation

Using this dataset, we can now compare the processes in our proposed models with the empirical distributions from actual data. First, we take a look at the tunnel interarrival time in Figure 3. The arrivals show a strong diurnal effect, closely resembling patterns also present in user plane traffic. In the data we see a decline of arrivals, i.e., longer interarrivals, late in the night and during the early morning hours with a peak rate in the afternoon and early evening. To represent this time-of-day dependence in the model, the measurement was split into the four time slots displayed in the figure. Each slot was then fitted with an exponential distribution with the rates λ given in Table I for the four time slots. The fitted functions match the empirical data, with some deviation present at the left tail but overall with a correlation coefficient approaching positive 1. Next, we consider the

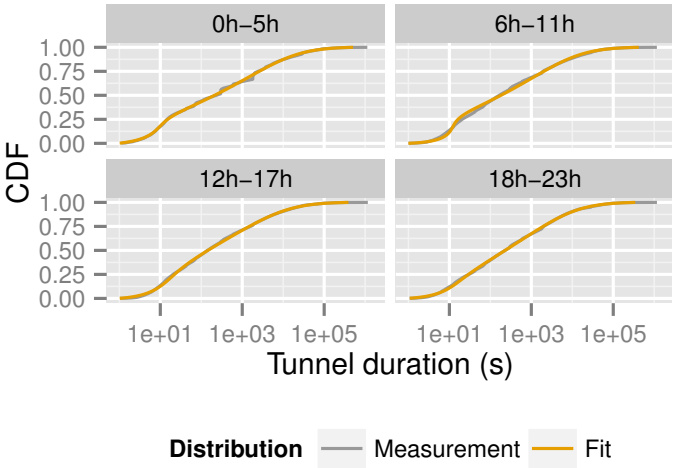

Fig. 4: Empirical and fitted CDFs of the tunnel duration by time of day with fitted rational functions.

duration the PDP Context state, which accompanies any GTP

tunnel, is held at the GGSN. Fig. 4 shows the tunnel durations split up for the time of day. There is once again a slight diurnal effect present, albeit with shifted peaks. Longer tunnels tend to occur at night, shorter tunnels during midday. For the model a distribution fit of the tunnel duration was also desired. However, none of the basic probability distributions (including exponential, gamma, and Weibull distributions) fit the tunnel duration well enough. We assume, that this can be attributed to the correlation of the tunnel duration to a large number of external factors, including user behavior, network-specific timers and procedures. All of which introduce artifacts and make it difficult to fit any distribution against. To get some kind of approximate mathematical representation of this distribution regardless, we attempted a rational function fit using Eureqa [15]. Table I additionally displays the generated functions which were fitted to the inverse CDF[1]. Both the CDFs in Fig. 4 as well as the Pearson correlation coefficients confirm the goodness of the fitted functions.

TABLE I: Parameters for the exponentially distributed inter-arrival times and corresponding Pearson correlation coefficients as well as the inverse functions fitted to the empirical duration distribution and correlation coefficients of the fit.

Time of Day	λ	R_{arr}	Inverse Fitted Duration Function	R_{dur}
0h-5h	10.67	0.99	$0.91 - 60.61y - 3498.78y^3 - \frac{110.70y + 2289.94y^3}{y - 1.00}$	0.99
6h-11h	24.53	0.99	$1 + 117.48y - 368.64y^2 - \frac{1720.13y^4}{y - 1.00}$	0.99
12h-17h	29.25	0.99	$0.95 + 69.49y + \frac{81146.10y^3 + 1.08 \times 10^6 y^5}{805 - 802.01y}$	0.99
18h-23h	23.49	0.98	$0.91 + 82.05y - \frac{2936.93y^4}{1.94y - 1.95}$	0.99

V. Numerical Evaluation

We implement[2] the models introduced in Sec. III using a Discrete Event Simulation (DES) with the SimPy [16] package as foundation. To be in line with the measurement data we consider a simulation time of seven days for all simulation scenarios, with a transient phase of 60 minutes accounted for. Ten replications of each scenario were performed. All error bars given in this section show the 5% and 95% quantiles of all replications.

In Sec. V-A we use the measurements introduced in Sec. IV in order to dimension a traditional GGSN as a baseline for all further studies. Based on these results, in Sec. III-A we examine the effects of NFV by scaling *out* instead of up in Sec. V-B through a virtual GGSN model. Finally, we arrive at a more realistic version of the virtual GGSN by taking the start up and shut down times into account in Sec. V-C.

A. Traditional GGSN

With the help of the interarrival times and duration of tunnels we study the traditional GGSN model previously introduced. Whilst our measurements provided us with information on the frequency of new tunnels and the duration they remain active, we have no reliable information on the number of active tunnels the GGSN can support. Thus, in a first step, we dimension the GGSN in such a way that a suitable blocking probability p_B can be achieved.

TABLE II: Manipulation check for the experimental factors based on one-way ANOVA.

	$F(2, 1275)$	η_p^2	p	Cohen's f^2	Cohen's $\hat{\omega}^2$
blocking probability					
maxTunnels	15601.53	0.993	< 0.001	26.73	0.96
maxInstances	10218.17	0.986	< 0.001	1.06	0.51
startstopDuration	0.86	0.003	0.482	0.00	0.00
mean tunnel count					
maxTunnels	20448.34	0.994	< 0.001	27.71	0.96
maxInstances	13348.25	0.989	< 0.001	1.06	0.51
startstopDuration	2.87	0.009	0.022	0.00	0.00

To obtain a baseline dimensioning, we perform a simulation study that considers the impact of an increasing load offer on the blocking probability. We find that for the normalized interarrival time no blocking is occurring if we allow for more than 5000 parallel tunnels. Thus, we consider the range of 4000 to 5000 parallel tunnels to be of special interest for the remainder of the study.

B. Virtual GGSN

In order to study the feasibility of the virtual GGSN approach discussed in Sec. III-A, we compare the performance indicators of the virtual GGSN with that of a traditional GGSN. To this end, the virtual GGSN is simulated in varying configurations. The number of servers and supported tunnels per server is chosen in such a way that the results can be compared with those obtained from our study of the traditional GGSN.

In the virtual GGSN model, servers are activated and deactivated on demand, while in the traditional GGSN model, the single server is always on. Generally, deactivating server instances reduces energy consumption and frees up inactive servers for other use. Thus, the number of active servers is a relevant performance metric. In order to analyze the influence of the different model parameters on the performance metrics, we perform a one-way ANOVA analysis with the results collected in Tab. II. High values for η_p^2 and Cohen's f^2 [17] indicate that the main influence for both the blocking probability and mean number of tunnels is the maximum number of tunnels n and servers $S_{\max}$, i.e., the total number of possible concurrent tunnels in the system. In Fig. 5 the Culmulative Distribution Function (CDF) of the number of active servers for four different virtual GGSN configurations is displayed. We observe, that increasing the number of supported tunnels per server allows a larger percentage of servers to be shut down or used for other tasks. This demonstrates the capability to scale the virtualized model in two dimensions quite well. Next, we take a look at the blocking probability of the virtual

[1]The inverse CDF was chosen as target to be able to directly use them in the random number generator in our simulation.

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

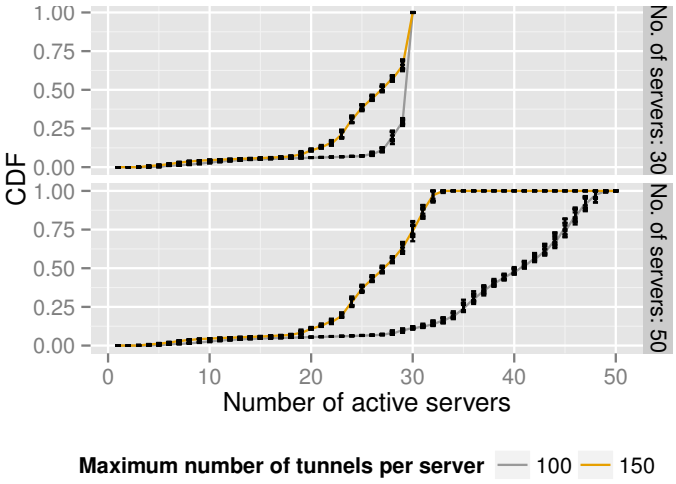

Fig. 5: Impact of the maximum number of tunnels and servers on number of active servers in the virtual GGSN model.

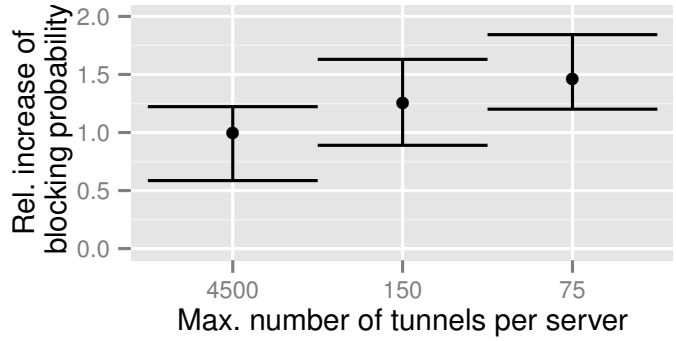

Fig. 6: Relative increase of blocking probability compared to the traditional GGSN; 4500 maximum tunnels per server being on a single server, 150 on 30, and 75 on 60 servers.

GGSN system in Fig. 6 and compare it to the results from the traditional GGSN model dimensioned for 4500 concurrent tunnels. We observe that the blocking probability increases by a factor of 1.48 — albeit at a still very low absolute scale — if the capacity of each server is set to 75, i.e. 1⁄60 of the original server capacity, while 27 of all 60 servers can be turned of or used for other purposes at 50% of the time. We conclude, that choosing more powerful servers decreases the blocking probability but reduces the potential to disable servers.

C. Impact of startup and shutdown times

In this section, we first consider the impact of different boot and shut down times, for example if fast flash storage is used, on resource utilization and blocking probabilities. Afterwards, the influence of varying server start and stop times on a fixed combination of maximum tunnels and servers in the system is examined. Fig. 7 shows scenarios with 40 and 100 number of virtual GGSN instances surmounting to a total tunnel capacity between 1000 to 5000. We study the impact of selecting different tunnel capacities per virtual instance as well as start up and shut down times on the blocking probability and mean resource utilization. We observe that by increasing the number of servers, i.e., scaling out, the blocking probability can be decreased, while maintaining a relatively low mean resource utilization. In addition to the previous effects, we notice that a higher start up and shut down time causes a slight increase in blocking probability for servers with low tunnel capacity. To study this behavior in more detail, we focus on a specific scenario in Fig. 8, where 5000 total tunnels should be supported by the system. In order to achieve this goal, we consider three types of instances, with the server capacity varying between 50 and 500. In each case we change the start up and shut down time between 1 and 5 minutes. Lower server capacities combined with higher start up and shut down times increase the blocking probability. This is can in part be attributed to the simplistic instance start up threshold mechanism used in the model, which does not take the additional capacity gained by activating an additional server into account. If smaller instances are to be used, for example because they are cheaper than large instances, start up delay should be kept minimal or an appropriate instance management strategy has to be chosen.

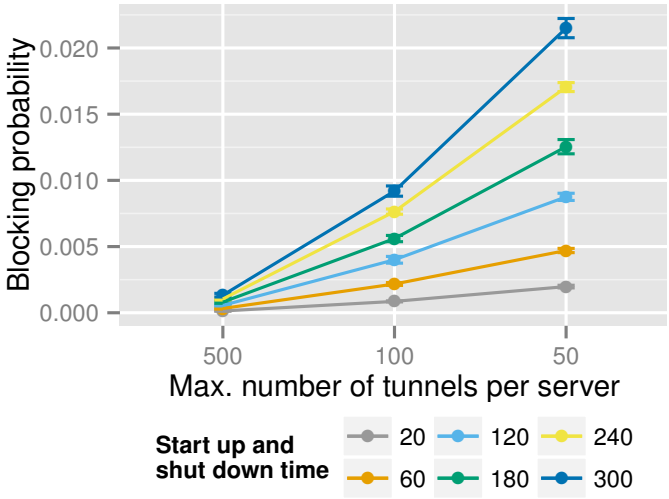

Fig. 8: Influence of start up and shut down time on blocking probability with regard to different numbers of servers.

VI. Conclusion

In this paper we investigated models and trade-offs for virtualizing components of the mobile core network. We first discussed a novel approach to mobile core network load modeling based on the control plane load at the GGSN. The non-stationary Erlang loss model $M(t)/G/c_c/0$ is based on the currently implemented state of the network architecture and backed by an evaluation of actual data. This can serve as a baseline reference to plan and dimension mobile network accordingly, not just based on expected user traffic as traditionally. To improve scaling in the future, we proposed a new and virtualizable approach for GGSNs. We presented random variables to model load in a GGSN based on measurement data from the network of a nation-wide mobile service provider

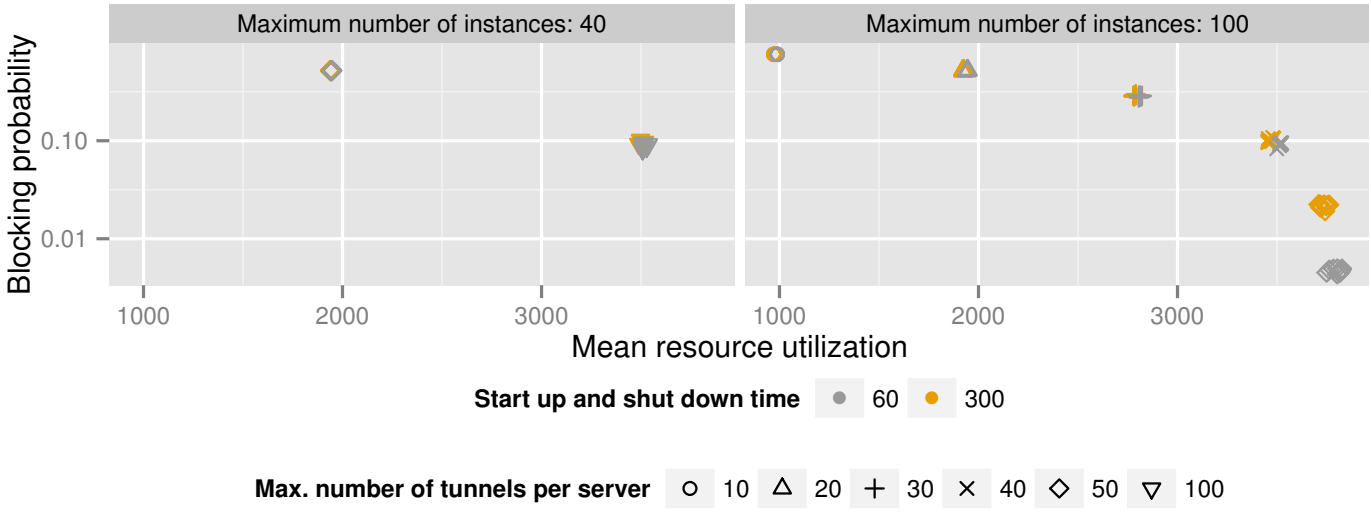

Fig. 7: Trade-off between blocking probability and mean resource utilization with regard to maximum number of servers, maximum number of tunnels per server, and start up and shut down time.

and made them available for reuse. Finally, we evaluate the model using a queuing simulation. We have shown, that the system's blocking probability is roughly equal to the single-server model but in addition achieves large efficiency gains, even when subjected to rudimentary provisioning conditions and long boot times. The model also has the ability to very easily scale out one's infrastructure by simply adding more small servers, reducing operational overhead. Implementing this model in an actual network might need considerable future effort and adaptation of existing infrastructure, protocols and standards. But if done correctly it could lead to new GGSN-as-a-Service business models, removing the need to provide and operate large amounts of infrastructure for rare cases of peak load.

Acknowledgements

This work was partly funded by the Deutsche Forschungsgemeinschaft (DFG) under Grants HO 4770/1-1, TR257/31-1, and TR 257/41-1.

References

[1] K. Tutschku, "Demand-based radio network planning of cellular mobile communication systems," in 17th *Conf. on Computer Communications*, 1998.

[2] NFV Industry Specification Group in ETSI, *Network Functions Virtualisation – An Introduction, Benefits, Enablers, Challenges & Call for Action*, SDN & OpenFlow World Congress, Whitepaper, Oct. 2013.

[3] 3GPP, *3GPP TS 23.060 General Packet Radio Service (GPRS); Service description; Stage 2*, 2012.

[4] ——, *3GPP TS 29.060 GPRS Tunnelling Protocol (GTP) across the Gn and Gp interface*, 2012.

[5] F. Metzger *et al.*, "Research Report On Signaling Load and Tunnel Management in a 3G Core Network," 2012.

[6] F. Metzger *et al.*, "Exploratory Analysis of a GGSN's PDP Context Signaling Load," *Journal of Computer Networks and Communications*, Feb. 2014.

[7] F. Qian *et al.*, "Profiling Resource Usage for Mobile Applications: A Cross-Layer Approach," in 9th *Conf. on Mobile Systems, Applications, and Services*, 2011.

[8] P. Perala *et al.*, "Theory and Practice of RRC State Transitions in UMTS Networks," in *GLOBECOM Workshops*, 2009.

[9] C. Schwartz *et al.*, "Angry Apps: The Impact of Network Timer Selection on Power Consumption, Signalling Load, and Web QoE," *Journal of Computer Networks and Communications*, 2013.

[10] M. Shafiq *et al.*, "Characterizing and Modeling Internet Traffic Dynamics of Cellular Devices," in *SIGMETRICS*, 2011.

[11] U. Paul *et al.*, "Understanding traffic dynamics in cellular data networks," in 30th *Conf. on Computer Communications*, 2011.

[12] Y. Zhang and A. Årvidsson, "Understanding the characteristics of cellular data traffic," *SIGCOMM Comput. Commun. Rev.*, vol. 42, Sep. 2012.

[13] P. Svoboda *et al.*, "Composition of GPRS, UMTS Traffic: Snapshots from a Live Network," in 4th *Workshop on Internet Performance, Simulation, Monitoring and Measurement*, 2006.

[14] F. Ricciato, "Introduction to the DARWIN Project," Accessed: 06-Jul-2011. [Online]. Available: www.ftw.at/~ricciato/darwin/.

[15] S. M. and H. Lipson, "Distilling Free-Form Natural Laws from Experimental Data," *Science*, vol. 324, 2009.

[16] *Simpy*. [Online]. Available: simpy.readthedocs.org/.

[17] P. D. Ellis, *The Essential Guide to Effect Sizes: Statistical Power, Meta-analysis, and the Interpretation of Research Results*. Cambridge University Press, 2010.