Network-wide measurement of GPRS bandwidth and latency

Bernd Pfitzinger[1], Tommy Baumann[2], Andreas Emde[2], Torsten Gründer[3], Dragan Macos[4], and Thomas Jestädt[1]

[1]Toll Collect GmbH, Linkstraße 4, 10785 Berlin, Germany.
[2]Andato GmbH & Co. KG, Ehrenbergstraße 11, 98693 Ilmenau, Germany.
[3]GRÜNDER Consulting GmbH, Am Krautgarten 6, 86926 Greifenberg, Germany.
[4]Beuth Hochschule für Technik Berlin, Luxemburger Str. 10, 13353 Berlin, Germany.

Abstract

Complementing a recently collected large data set on the TCP connection termination latency in GPRS networks we analyze server-side log data generated in a large scale automatic toll system to observe the network bandwidth. After a recent architectural change the on-board units (OBUs) record GPS tracks and transmit track data to the central system for processing rather than transmitting the toll data after local processing. The bandwidth in upload direction is estimated from the server-side log entries and corrected for the network latency. The data collected allows comparing the performance of seven types of OBUs in three GPRS networks over time.

While the three networks differ in the average bandwidth offered, the biggest performance impact is the OBU type where modems with the same specification yield different upload rates. In addition we update the GPRS network latency data by fitting two statistical distributions, improving markedly on the prior results.

1. Introduction

Network access is one important factor for the performance of mobile applications. The technical performance of a network is typically given by the bandwidth available for a given user and the latency, i.e. the minimum time to transmit a single data packet. "Internet-of-things" applications are also very cost sensitive, both regarding the cost of the hardware and the cost of network access subscription. While many new network protocols are proposed to address these challenges [1], near-ubiquitous access depends at present on GSM networks and their successors.

Unfortunately, mobile network access is known to be a challenging environment for the TCP/IP protocol – high latency, packet loss and rapidly changing network capacity impact the protocol efficiency [2].

In this article we take a real-world application – the German automatic toll system – to determine the bandwidth and latency of different types of OBUs in three national GSM networks. The next section summarizes the (scarce) literature on GSM 2G network bandwidth and Section 3 the recent architectural changes to the toll system. Where previously the thick-client was sensitive to the network latency, the thin client also depends on the bandwidth and we analyze server-side logs to calculate the bandwidth in upload direction (Section 4). Section 5 discusses the empirical data collected on the network bandwidth and Section 6 updates the data set on the network latency.

2. Literature review

The bandwidth achieved depends on many factors: GSM offers eight time slots that can be assigned to either GPRS data transmission or GSM telephony [3]. A given GPRS time slot transmits data only in one direction (either upload or download) and GPRS modems offer to combine several time slots to increase the bandwidth available for the user. In that way typically up to 5 time slots are taken up by one modem with one or two time slots in the upload direction.

The *channel bandwidth* for using one time slot further depends on the level of error correction. Modems are required to implement four different GPRS coding schemes (CS-1 to CS-4 offering a net bandwidth of 8, 12, 14.4 and 20 kbps respectively, [3]). However, networks are only required to offer CS-1 but typically offer at least CS-2. When offered, the higher coding schemes only apply under favorable radio conditions i.e. the modem is stationary near a base station. To our knowledge,

URI: https://hdl.handle.net/10125/60189
ISBN: 978-0-9981331-2-6

no public information is available on the coding schemes used in the three German GPRS networks.

Looking at the literature, large-scale network performance measurements are rare, especially in the rather antiquated 2G networks: The Netradar app [4] has collected 57 286 network speed measurements between 2012 and early 2015 in GPRS networks – the remaining 96.2 % of the measurements concern the more modern network types EDGE, HSPA, HSPA+ and LTE (see Figure 2.4 in [5], the crowd-sourced initiative has in the meantime apparently transitioned to a commercial entity). For GPRS [5] lists a median upload speed of 8 kbps with a maximum of 24 kbps.

Yet the *empirical bandwidth* is better described as a distribution rather than a single number. Accessing the GPRS core network [6] collected more than 12 million samples on GPRS connections in seven different (and unnamed) countries. For a (unspecified) sample of connections with more than 100 kB transferred, [6] gives a histogram with peaks at multiples of the CS-2 channel bandwidth (Figure 9 in [6]). Packet losses and idiosyncrasies of the TCP/IP protocol prevent reaching the full channel bandwidth.

3. Revisiting the German automatic toll system

Since 2005 the German truck toll on motorways and selected federal trunk roads is operational. The vast majority of the approx. €4.3 bn truck toll collected annually is automatically processed by one of the more than 1 100 000 OBUs deployed in heavy goods vehicles (HGVs).

3.1 Thick-Client

The thick client approach is to calculate the tolls due by the OBU, given the current position and the up-to-date geo and tariff data. The – rather brief – result is periodically transmitted to the central system for further processing and billing.

Even though the toll road network is large – with an extent of about 15 000 km in each direction – the road network topology is fairly simple: Motorways have few exits where trucks can enter or leave the toll road network and U-turns are neither allowed nor possible. As a consequence it is possible to create a rather compact map representation of the toll road network that can be transmitted to the OBUs for local toll calculation. Together with infrequent changes to the road network topology, few updates to the map and tariff data suffice.

Toll data transmission is infrequent and typically very brief, i. e. the network bandwidth is in this case of no concern. If at all, the network latency determines the duration of the toll data transmission. It this context where we use server-side log data to determine the latency (see Section 6 for an update).

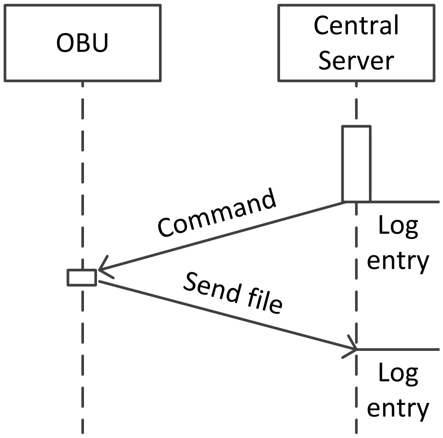

Figure 1: In the thin-client approach the transmission of recorded journey data is initiated by the central system.

3.2 Thin client

As of 1 July 2018, motorways and *all* federal trunk roads in Germany are subject to tolls for HGVs. The toll road network grows to encompass around 52 000 km (again in each direction). The additional toll roads are no longer similar to motorways: Trucks can enter and leave the toll road network almost arbitrarily, U-turns are (technically) possible and some of the roads are within cities restricting the accuracy of GNSS positioning. It is also expected, that changes to the toll road topology become much more frequent.

To address this extension the German automatic toll system transitioned to a thin-client architecture, where the tolls are calculated by the central system. The OBUs are simplified to only record journey data and vehicle characteristics that are in turn encrypted and stored in files for later transmission to the central system. The OBUs contact the central system periodically and depending on the current state the central system may send a command to initiate the upload of an existing file.

Upon receipt the file is decrypted and further processed, i.e. the tolls due are calculated and billed, the original file residing on the OBU is deleted.

While the overall process flow remains unchanged – GNSS positioning followed by toll calculation and billing – shifting the toll calculation to the central system incurs additional network traffic and stringent requirements regarding the data privacy and protection.

Since the computing power is much larger in the data center, the toll road topology and its update frequency is no longer a practical concern.

Since late 2017 the OBUs were gradually migrated to the thin client architecture and by July 1st 2018 any remaining thick-client OBUs became nonoperational. At present the OBU fleet contains OBUs from 7 different types: Two current OBU generations and the other five dating back more than a decade. These two groups differ in their thin-client configurations as to the file size used to record and transmit the journey data. The current OBU generations use 25 kB files, all other generations 50 kB.

All OBU generations are equipped with a GSM modem for the data transmission, for economical reasons limited to use only the GPRS (2G) network (i.e. without the EDGE extension). At least the current OBU generations use class-10 modems that offer the use of up to two time slots in upload direction if the capacity is provided by the network.

4. Data extraction

Whenever an OBU is powered on it periodically contacts the central system to exchange status information. As usual, each connection starts with the authentication through a RADIUS server before a TCP/IP connection between the OBU and the central system is established. Upon connection termination, the RADIUS server writes a log entry containing the number of bytes exchanged.

In the thin-client approach the OBU initiates a TCP/IP connection with a central server. Once the relevant status information is exchanged and processed, the central server can decide that the OBU should transfer a file containing recorded journey data (see Figure 1). To that extent the central server sends a command to the OBU accompanied by a server-side log entry.

Since the specific file requested already exists at that time, the OBU will immediately transfer the file. Once the upload is complete, the central server logs the data transfer and proceeds e.g. with sending a receipt to the OBU.

We take these log entries to calculate the time needed to transfer a file. The application level protocols allow in rare cases for partially filled files and we use the RADIUS log entries to exclude such transmissions from our analysis. To analyze the data we extract the log data for typically 12 hours, sort the events chronologically and try to exclude spurious or inconsistent log entries by evaluating log entries concerning the communication session and the command issued by the server. However, this introduces a systematic error since connections cannot be evaluated if they are open at the end of the exported time range. In effect the (few) long running transmissions will therefore be under-estimated at the end of the extracted data set.

5. Empirical bandwidth in GPRS networks

The upload bandwidth is of interest in the example of the German automatic toll system, e.g. determining the number of parallel sessions held by the central server. However, to our knowledge no data exists on the network-wide GPRS bandwidth. Therefore we analyze the server-side log data to calculate the file upload time (Section 5.1) and after correcting for the network latency calculate the upload bandwidth (Section 5.2).

5.1 File upload time

With the migration of the OBU fleet to the thin client architecture we started collecting data on the file upload time. Figure 2 gives the probability density function (PDF) of the aggregated data (encompassing more than 194 million events for the current OBU generations and 34 million events for the older OBU generations).

Looking at Figure 2 the minimum upload time is 6.3 s for the small files and 18.8 s for the large files including the round-trip time of the command issued by the central server. The older OBU generations apparently take three times as long to transmit twice the data as the current OBU generations. The main reason is the capability of the modems used: Class-10 modems offer to use up to two time-slots in the upload direction, the older modems only allow using one time-slot.

The log-log plot of the data emphasizes the long tail of the distribution: Successful uploads are recorded with transmission times of up to 20 minutes (when a server-side application timeout closes any remaining session). The events shown in Figure 2 beyond that timeout result from incomplete log data filtering which has since been improved.

5.2 Upload bandwidth

Taking the file upload time it is trivial to compute the upload bandwidth since the data has already been filtered to include only full journey data files. However, the average round-trip time – typically between one and two seconds – included in the upload time distorts the bandwidth for short transmissions. Therefore we take the network and OBU specific average round-trip time (from the data set first described in [7] and since then doubled in size) to correct the file upload time.

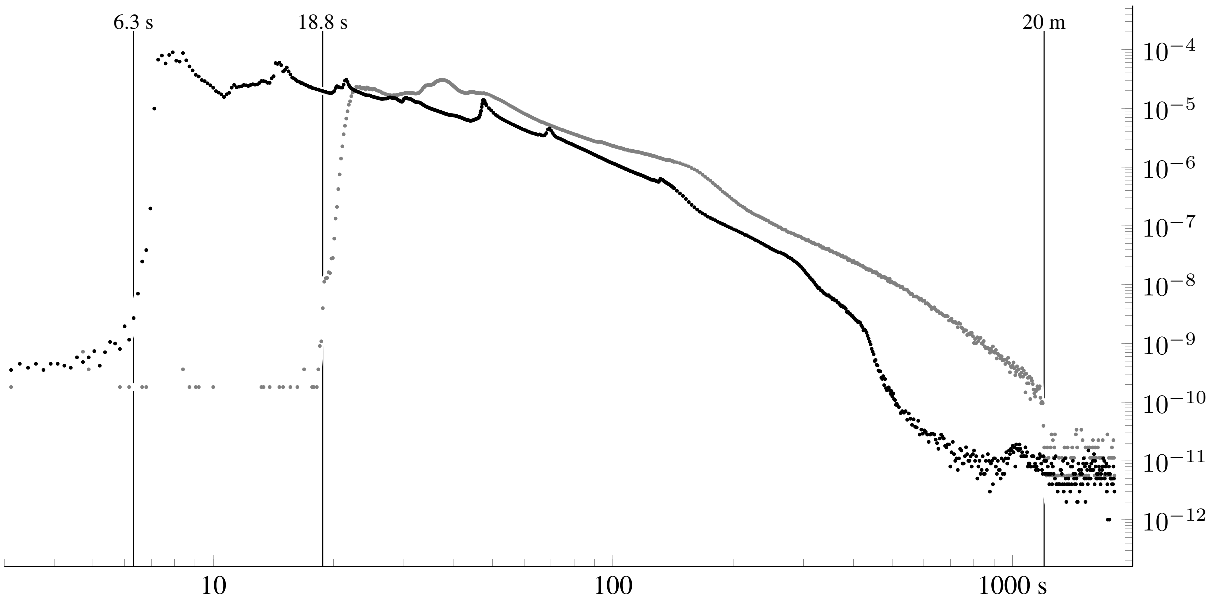

Figure 2: Empirical probability density function (normalized to 1ms time resolution) for the data upload transmission time of the current OBU generations (black, small file size) and the older OBU generations (gray, twice the file size transmitted with slower modems). Empirical data using 160 ms, 640 ms, 2560 ms and 5120 ms binning for the regions below 47s, 141s, up to and beyond 512s.

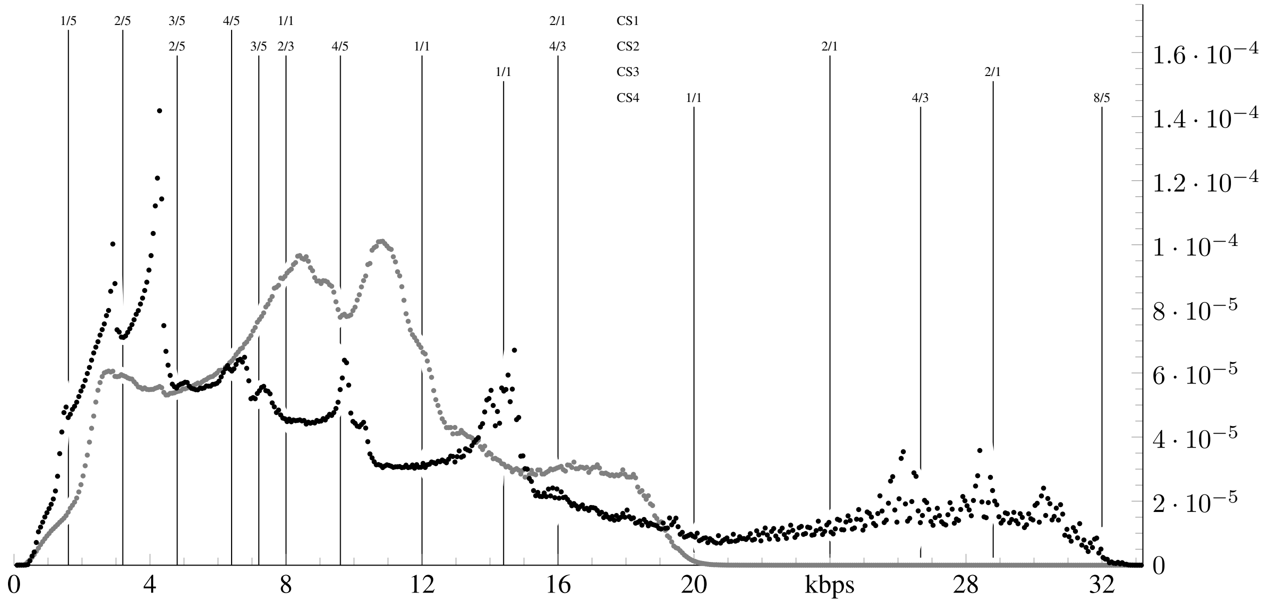

Figure 3: Empirical probability density function (normalized to 1 bps bandwidth resolution using 64 bps binning) for the data upload of the current OBU generations (black) and the older OBU generations (gray), data set as in Figure 2 but corrected for the average network round-trip time. The lines denote (fractions of) the GPRS net channel bandwidths for the coding schemes CS1 to CS4 using up to two time slots.

The result is summarized in Figure 3: The PDF of the upload bandwidth for the two groups of OBU types. The linear scale in Figure 3 emphasizes the most dominant bandwidths (rather than the long tail behavior of the tramission duration). To aid the interpretation we have added typical multiples of the GPRS channel bandwidth that can be achieved using up to two time-slots with one of the four coding schemes. Note, that both groups of OBUs use different operating systems and file sizes, i.e. both PDFs should not be compared directly.

Looking at the older OBU types (gray dots in Figure 3) it is obvious, that the modems used offer only one time-slot for uploads – the maximum bandwidth achieved is just short of the 20 kbps provided by the CS-4 coding scheme. In fact, we observe that all German GPRS networks offer all four coding schemes. As expected, the net bandwidth is somewhat lower than the theoretical channel bandwidth with further peaks close to the values expected for the CS-2 and CS-1 coding schemes. At the low end, few file uploads proceed at a speed of less than $1/5$ the CS-1 channel bandwidth.

The current OBU generations (black dots in Figure 3) can take advantage of up to two time slots for uploads, about 17% of all file uploads achieve a bandwidth $>$ 20 kbps i. e. use two time slots. These transmissions typically use the CS-3 and CS-4 coding schemes.

For slower transmissions the PDF differs markedly from the behavior seen with the older OBU generations: Distinct peaks occur at fractions of the single time-slot channel bandwidth with the slow coding schemes CS-1 and CS-2 (e.g. at $1/5$, $2/5$ and $3/5$ the CS-1 channel bandwidth). One reason could be the network utilization yet this is not visible for the older OBU generations in the same networks. Most probably the TCP/IP implementation and specific workload do not allow to achieve the single channel bandwidth provided by the network.

5.3 Variation over time

Figure 4 gives the variation of the average and median upload bandwidth over the week. For this analysis we aggregate the data by the hour of the week and compute the average and median (in the example for the current OBU generations). During the day the bandwidth varies by more than 10% reaching the highest value after midnight and the lowest value at noon with a noticeable speed-up on Saturday and Sunday.

An artifact of the data extraction discussed in Section 4 is visible in Figure 4 on every weekday morning: The cut in the data export is typically each day at a similar hour. At that point in time the long running connections are not evaluated, overestimating the bandwidths achieved.

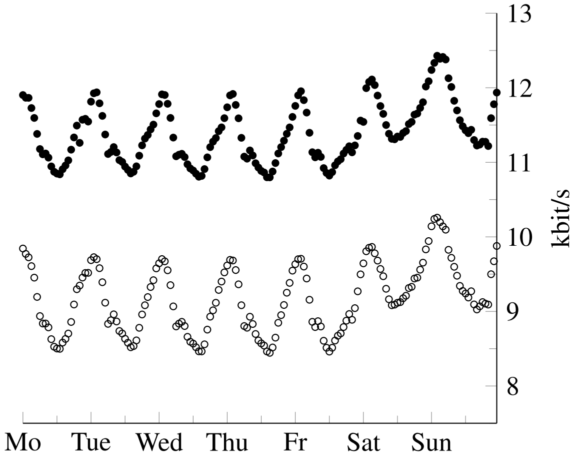

Figure 4: Average (top) and median (bottom) upload bandwidth over the week for the current OBU generations.

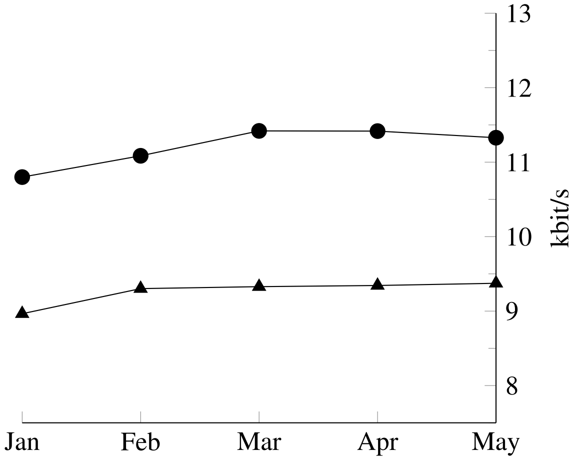

Figure 5: Average upload bandwidth per month for the current OBU generations (top, small files) and the older OBU generations (bottom, twice the file size and slower modems).

Looking at a time scale of one month (Figure 5) the average upload bandwidth remains almost unchanged in the few months available for analysis. Especially early 2018 the migration to the thin-client architecture was still ongoing, severely limiting the statistics.

6. Network latency

The thick-client OBU was characterized by short data transmissions to the central system which we used to collect data on the connection termination latency [7]. Since then the data set has doubled from some 300 million data points to almost 700 million with a slightly more emphasized long-tail of the probability density function (see Figure 6). The larger data set expands the long-tail well beyond the time-out at 70 s – which is still visible but much less pronounced than in [7].

In combination with a 500 ms binning an additional peak becomes visible at 96 s, the continuation of the exponential back-off already seen in [7]. Following [8] (chapter 21) we interpret the peak at 6.9 s as the initial re-transmission timeout followed by the exponential back-off algorithm with peaks at 10.8 s, 23.0 s, 47.5 s and 96.2 s. At the far end in Figure 6 another time-out occurs (at 120 s): The number of events drops from above 100 (per 500 ms bin) to below 1.

Looking back at the statistical distributions fitted to the data the double Pareto lognormal distribution (dPln) and the Generalized Beta distribution of the second kind (GB2) gave the best results [9]. Since then we added two distributions – the four-parameter α-stable distribution (numerical implementation given in [10]) and the five-parameter generalized hyperbolic (GH) distribution (equations 1.1 and 1.2 in [11]).

The α-stable distribution and the GH distribution are defined for $x \in \mathbb{R}$. Since the latency is non-negative we adjust the PDF accordingly (division by $1 - \text{CDF}(x = 0)$) and fit these distributions to the large data set. The computation follows the description given in [9], the small data set is fitted between 1170 ms and 20 000 ms and the large data set is fitted between 1170 ms and 60 000 ms (with a scale factor of 99.988%).

The GH distribution uses – among others – the modified Bessel function $K_\nu(x)$. The computation of the GH distribution is straight forward using the confluent hypergeometric function $U(a, b, x)$ (provided by the GSL library [12] and using its relation to $K_\nu(x)$ see [13], Section 10.39). The numerical calculation may again break down for large x or certain choices of parameters. Therefore we repeat the calculations with higher precision using the `Arb` C library for arbitrary-precision ball arithmetic [14] which implements the hypergeometric function [15]. For the computation we start with 128 bit precision and increase the precision up to 4096 bits trying to achieve at least 64 bits of relative precision for the computed result.

Table 1: Results of the parameter fitting for the large data set: Fitness Δ and model parameters. The parameter definition follows [9] for the GB2 and dPln, [10] for the α-stable and [11] for the GH distributions.

model	parameters
GB2	α = 10.7304, β = 340.120
Δ = 13.460	p = 0.174479, q = 0.136890
dPln	α = 1.46890, β = 1.86338
Δ = 13.427	τ = 0.152647, ν = 5.83541
α-stable	α = 1.10266, β = 0.560759
Δ = 13.498	σ = 152.694, μ = 317.924
GH	λ = -0.94940
Δ = 12.360	α = 0.0024072, β = 0.0023809
	δ = 229.251, μ = 274.186

Neither CDF nor PDF of the α-stable distribution exist in a closed form. Each value must be computed by numerical integration, we use the program given in [10].

6.1 Parameter fitting

To fit the distributions we keep the computation and weight factor as in [9] and add the use of a genetic algorithm [16] in conjunction with the multi-dimensional simplex-algorithm used previously. Applying the genetic algorithm to the small data set leaves the best fit unchanged for the dPln distribution and improves the fit for the GB2 distribution (from $\Delta = 13.979$ to $\Delta = 13.419$).

This remains almost unchanged for the large data set (Table 1): The dPln-, GB2- and α-stable distributions fit the data equally well when using our weighted fitness factor. The five parameter GH distribution improves the fitness considerably – at the cost of an additional parameter. Looking at the CDF (Figure 7) the α-stable and GH distributions follow the bulk of the CDF much better, where the α-stable distribution overestimates the long tail of the data (dashed red line in Figure 6) in contrast to the GH-distribution.

All distributions give a similar asymptotic behavior. The power-law asymptotic of the dPln-, GB2- and α-stable distributions yield a leading exponent of -2.4689, -2.4689 and -2.11 respectively (when taking only the leading term of the expansion of the stable distribution [17]).

The asymptotic behavior of the GH-distribution is 'semi-heavy': $gh(x, \lambda, \alpha, \beta, \delta) \propto |x|^{\lambda-1}\text{exp}((\beta - \alpha)x)$ for $x \to \infty$, see equation 1.19 in [18]. The smaller

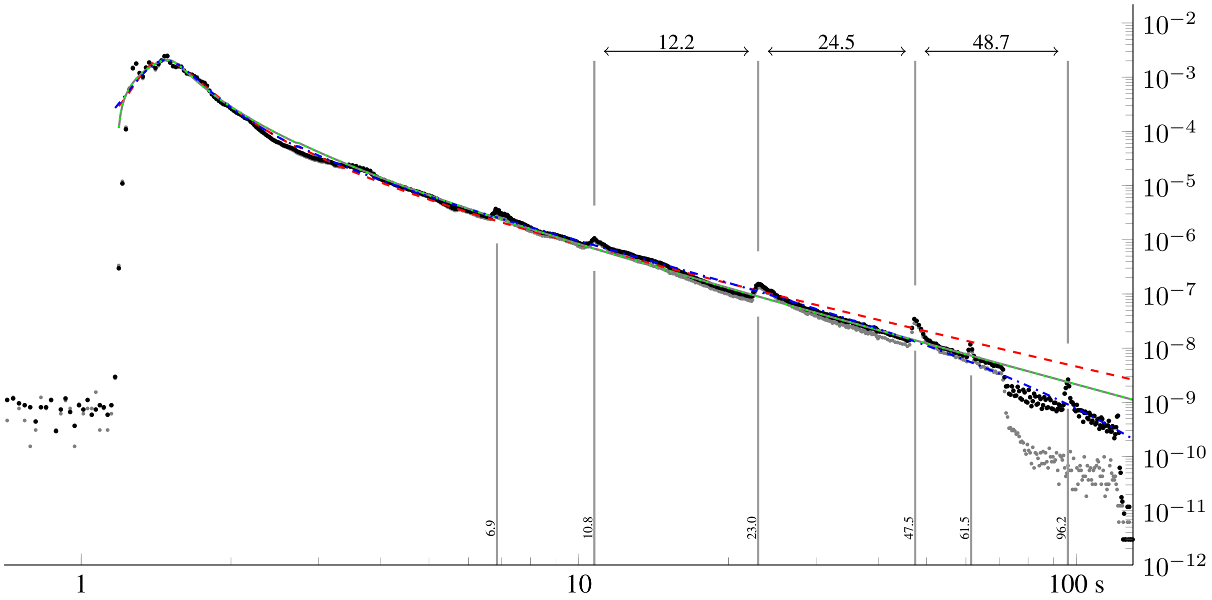

Figure 6: Empirical PDF (normalized to 1ms time resolution) for the TCP connection termination latency in 2G mobile networks (data set from [7] in gray, expanded data set in black). Empirical data using 20ms, 100ms and 500 ms binning for the regions below 8s, up to and beyond 40s. An additional peak attributed to the exponential back-off is visible at 96s in the expanded data set. In comparison the best parameter fits of the α-stable (dashed red), GH- (dash-dotted blue), GB2- (gray) and dPln-distribution (dotted green) are plotted as lines – the latter two being essentially indistinguishable.

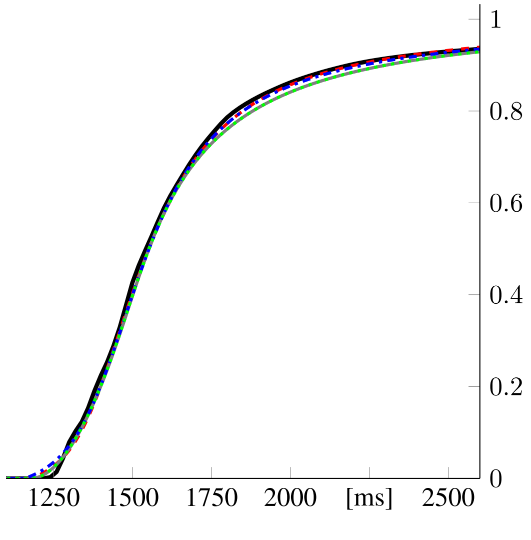

Figure 7: Empirical CDF of the network latency (black line) compared to the best parameter fits of different statistical functions (lines as in Figure 6).

leading exponent of -1.949 is combined with an exponential fall-off that is only visible for large x (see Figure 6) since its coefficient is very small: $\beta - \alpha \approx -2.62$e-5. In fact this suggests that the special case of the GH distribution where $\alpha = \beta$ (the generalized hyperbolic skew Student's t-distribution, equation 8 in [19]) might also be applicable.

7. Summary and outlook

We collected a large data set on network-wide bandwidth and latency measurements in GPRS networks. The GSM channel bandwidths offered by the four different coding schemes can be observed in all German mobile networks. However, the network performance depends significantly on the OBU type suggesting that sometimes the TCP/IP implementation and the specific workload are at odds with each other. Even with a channel bandwidth of at least 8 kbps we observed successful transmissions proceeding at less than 150 bit/s. Further work is needed to investigate the low bandwidth behavior of the current OBU generations – either through observations in a laboratory setting or using TCP/IP and GPRS simulations.

The data set regarding the network latency doubled in size and we updated the statistical models used to describe the empirical data.

8. Acknowledgments

We thank the reviewer for pointing out the α-stable distribution as an additional candidate for parameter fitting redoubling our efforts on finding an appropriate statistical model. One of the authors was funded by the German Federal Ministry of Education and Research (BMBF), grant ”SimDesign, 01|S14026B”.

References

[1] U. Raza, P. Kulkarni, and M. Sooriyabandara, “Low power wide area networks: An overview,” *IEEE Communications Surveys Tutorials*, vol. 19, no. 2, pp. 855–873, 2017. DOI: 10.1109/COMST.2017.2652320.

[2] G. Xylomenos, G. C. Polyzos, P. Mahonen, and M. Saaranen, “TCP performance issues over wireless links,” *IEEE Communications Magazine*, vol. 39, no. 4, pp. 52–58, Apr. 2001, ISSN: 0163-6804. DOI: 10.1109/35.917504.

[3] R. Kalden, I. Meirick, and M. Meyer, “Wireless internet access based on GPRS,” *IEEE Personal Communications*, vol. 7, no. 2, pp. 8–18, Apr. 2000, ISSN: 1070-9916. DOI: 10.1109/98.839328.

[4] S. Sonntag, J. Manner, and L. Schulte, “Netradar - measuring the wireless world,” in *2013 11th International Symposium and Workshops on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt)*, May 2013, pp. 29–34, ISBN: 978-3-901882-54-8.

[5] S. Sonntag, “No link left behind,” PhD thesis, Aalto University, 2016, ISBN: 978-952-60-6652-3 (electronic). [Online]. Available: http://urn.fi/URN:ISBN:978-952-60-6652-3.

[6] P. Benko, G. Malicsko, and A. Veres, “A large-scale, passive analysis of end-to-end TCP performance over GPRS,” in *IEEE INFOCOM 2004*, vol. 3, 2004, pp. 1882–1892. DOI: 10.1109/INFCOM.2004.1354598.

[7] B. Pfitzinger, T. Baumann, A. Emde, D. Macos, and T. Jestädt, “Network-wide measurement of TCP RTT in 2G networks,” in *Proceedings of the 51st Hawaii International Conference on System Sciences (HICSS)*, Jan. 2018, pp. 5818–5824, ISBN: 978-0-9981331-1-9. DOI: 10125/50618.

[8] W. R. Stevens, *TCP/IP Illustrated, Volume 1*. Boston, MA, USA: Addison-Wesley, 1995, ISBN: 978-0201633467.

[9] B. Pfitzinger, T. Baumann, A. Emde, D. Macos, and T. Jestädt, “Modeling the GPRS network latency with a double Pareto-lognormal or a generalized Beta distribution,” in *Proceedings of the 51st Hawaii International Conference on System Sciences (HICSS)*, Jan. 2018, pp. 5825–5830, ISBN: 978-0-9981331-1-9. DOI: 10125/50619.

[10] J. Royuela-del-Val, F. Simmross-Wattenberg, and C. Alberola-López, “Libstable: Fast, parallel, and high-precision computation of α-stable distributions in R, C/C++, and MATLAB,” *Journal of Statistical Software, Articles*, vol. 78, no. 1, pp. 1–25, 2017, ISSN: 1548-7660. DOI: 10.18637/jss.v078.i01.

[11] O. Barndorff-Nielsen, “Hyperbolic distributions and distributions on hyperbolae,” *Scandinavian Journal of Statistics*, vol. 5, no. 3, pp. 151–157, 1978, ISSN: 03036898.

[12] B. Gough, *GNU Scientific Library Reference Manual*, 3rd. Network Theory Ltd., 2009, ISBN: 9780954612078. [Online]. Available: https://www.gnu.org/s/gsl/manual/gsl-ref.pdf.

[13] *NIST digital library of mathematical functions*, http://dlmf.nist.gov/, Release 1.0.16 of 2017-09-18, F. W. J. Olver, A. B. Olde Daalhuis, D. W. Lozier, B. I. Schneider, R. F. Boisvert, C. W. Clark, B. R. Miller and B. V. Saunders, eds. [Online]. Available: http://dlmf.nist.gov/ (visited on 11/03/2017).

[14] F. Johansson, “Arb: Efficient arbitrary-precision midpoint-radius interval arithmetic,” *IEEE Transactions on Computers*, vol. 66, pp. 1281–1292, 8 Aug. 2017. DOI: 10.1109/TC.2017.2690633.

[15] F. Johansson, “Computing hypergeometric functions rigorously,” 2016. arXiv: 1606.06977. [Online]. Available: http://arxiv.org/abs/1606.06977 (visited on 05/11/2018).

[16] M. Wall, “GAlib: A C++ library of genetic algorithm components,” PhD thesis, Mechanical Engineering Department, Massachusetts Institute of Technology, 1996. [Online]. Available: http://hdl.handle.net/1721.1/10259.

[17] H. Bergström, “On some expansions of stable distribution functions,” *Arkiv för Matematik*, vol. 2, no. 4, pp. 375–378, Oct. 1952, ISSN: 1871-2487. DOI: 10.1007/BF02591503.

[18] K. Prause, “The generalized hyperbolic model: Estimation, financial derivatives, and risk measures,” PhD thesis, Mathematische Fakultät, Universität Freiburg, 1999. [Online]. Available: http://nbn-resolving.de/urn:nbn:de:bsz:25-opus-155.

[19] K. Aas and I. H. Haff, “The generalized hyperbolic skew Student’s t-distribution,” *Journal of Financial Econometrics*, vol. 4, no. 2, pp. 275–309, Mar. 2006. DOI: 10.1093/jjfinec/nbj006.