

Passive monitoring and geo-based prediction of mobile network vehicle-to-server communication

Josef Schmid, Philipp Heß and Alfred Höß
Ostbayerische Technische Hochschule
Amberg-Weiden
92224 Amberg, Germany
Email: {j.schmid, ph.hess, a.hoess}@oth-aw.de

Björn W. Schuller^{1,2}
¹ZD.B Chair of Embedded Intelligence for Health Care & Wellbeing
University of Augsburg, 86159 Augsburg, Germany
²Department of Computing, Imperial College London, UK
Email: schuller@ieee.org

Abstract—Predicting mobile network parameters while driving is a challenge. The high dynamics and the mobility of the clients lead to spontaneous changes in the communication quality. There are different known approaches to predict the throughput of such mobile network connections. The paper presents a novel overview for the performance of different geographical prediction methods based on the same dataset. To generate the dataset, a tool for monitoring a vehicle-to-server communication in a passive way is presented, too. It does not only capture high level parameters like TCP throughput and round trip times, but also mobile network parameters, like RSSI or RSRQ. In addition, the data is recorded in real-time and augmented with GPS coordinates to analyze them location-dependent.

Index Terms—passive monitoring, vehicle-to-everything communication, throughput prediction, connectivity map, cellular network

I. INTRODUCTION

Within the automotive sector, several manufactures started working on automated driving. They all use onboard sensors to detect the local environment. Due to the fact that the range of on-board sensors is limited and not sufficient to guarantee a safe driving process at higher speeds, research is carried out on the subject of vehicle-to-everything (V2X) communication [1]. There are three common communication standards. IEEE 802.11p, which is focusing on vehicle-to-vehicle communication, vehicle-to-road-side-unit communication using small relay stations next to the street and finally, communication based on the mobile networks for vehicle-to-server connections. This paper focuses on communication based on mobile networks. For many scenarios it is crucial to guarantee that data is sent reliably. Therefore, a deeper analysis of the communication is necessary. Typically, the throughput (TP) and the Round-Trip-Time (RTT) are two indicators to determine the performance of a communication, but also mobile network parameters like the signal strength or the signal-to-interference plus noise ratio (SINR) are areas of interest. To measure TP or RTT, it is possible to inject traffic to the network and to analyze it. The mobile network is a shared medium with limited resources. By injecting additional data for the measuring process, the network congests and gets slowed down. To avoid loading the network, within this paper, a tool is presented to collect these network parameters passively. On one side, long-term evolution (LTE) related parameters get recorded and on the

other side, TCP related metrics are calculated. The recorded dataset is used to analyze the performance of different geographical based prediction algorithms. These methods are able to predict the throughput of a specific location along the route of the driving vehicle. This allows an application, e. g., adaptive video stream players [2], [3], to react on the varying bandwidth. In section II, the related work is discussed, followed by a detailed explanation of the measured mobile network parameters and the calculation of TP and RTT in section III. Section IV introduces the measurement setup and provides an overview of the architecture of the measuring tool. Our measurement campaign and the test area is described in section V. In section VI, the different types of geo-based prediction methods get described followed by an evaluation of the performance for each approach. Finally, we conclude our paper and present an outlook to future work.

II. RELATED WORK

This section is structured into two parts. The first part shows the related work regarding measuring data in a passive way and the second part reflects the current state of science for map based mobile throughput prediction.

For monitoring the behavior of a network, we have to distinguish between active and passive measurement methods. In [4] and [5] active measurements are used to get the round trip time or the throughput of a network. Such a measurement method leads to the problem that additional network traffic is generated. To avoid this problem, passive measurement methods are applied.

In [6], a technique called SYN-ACK (SA) estimation is introduced. This method is based on the packets exchanged during TCP's three way handshake in the beginning of a connection. With this technique it is possible to measure quite accurately RTT values for most of the TCP connections.

A second technique called Slow-Start (SS) estimation in [6] is based on the congestion control mechanism and relies on the assumption, that the time period between two packet bursts is equal to the connection's RTT. To get a precise measurement, it is important that five consecutive segments have the maximum segment size (MSS). As a consequence, this technique cannot be used for flows that only transfer small segments.

In [7] a third passive measurement technique is worked out. The FIN-ACK measures the RTT at the end of a TCP connection. This algorithm is quite similar to the SA technology in [6]. Both deliver an exact value for the RTT, but they have also the same disadvantages. These algorithms can just measure one RTT value per connection and since the segments do not contain any data, this can cause a smaller RTT.

The method described in [8] leverages the timestamp option within TCP. The authors determined that in 76.4 percent of the Alexa Global 500 list the TCP timestamp option is supported. The accuracy of the algorithm depends on the granularity of the TCP timestamps. These are different, depending on the implementation on sender side. They detected that over 90 percent of the implementations use a granularity bigger than 10 ms. Due to this constraint, this method is not useful to measure RTTs precisely.

In [9], different passive measurement approaches for measuring bandwidth and latency are investigated. The analysis focuses on mobile networks, mainly 3G. Within this paper, a tool is presented which implements the SYN-ACK algorithm and the FIN-ACK algorithm to calculate the RTT at runtime. Additionally, this software is able to determine the throughput and mobile network related parameters. The data of the tool shall be used to predict the throughput of a mobile network connection.

In the past, different methods have been presented to predict the throughput of a TCP connection. For the dynamic vehicle driving scenario, it turns out that location-based prediction overtakes predicting with previous throughput data from the actual driving circle [11], [12] and there are different ways these data can be used. In [13–15] approaches are deployed, which count all measured points nearby the actual location. Another approach is to match all data points to the road and split the track in segments. This can be done with segments defined by the Open Street Map data [16] or by predefining them with a fixed length as shown in [2], [11], [17], [18]. There is also the possibility of matching the data to a grid, which is more simple than building segments. Apart from these approaches there are also other methods like predicting the throughput with machine learning based algorithms leveraging LTE low level parameters [10]. Our recorded dataset also contains these parameters, but an evaluation is not part of this paper.

III. NETWORK PARAMETERS

To determine the quality of a network, characteristic parameters need to be analyzed. The communication is based on layers and they are interfering with each other. For a deeper analysis, we want to investigate metrics from the physical layer as well as from the transportation layer. Within our testbed, the communication is based on mobile networks, namely LTE. On the transport layer, the TCP protocol is used. Our measurement is taken on the client side. The following section describes the metrics which are collected from the measurement tool.

For judging the physical mobile connection quality, LTE parameters shown in Table I are gathered.

Table I: Captured LTE low-level parameters

Acronym	Definition
CNT	Cellular Network Type
RSSI	Receive Signal Strength Indicator
RSRP	Reference Signal Receiving Power
RSRQ	Reference Signal Receiving Quality
SINR	Signal to Interference Noise Ratio
ASU	Arbitrary Strength Unit
Cell Id	Cell Identifier
P. Cell Id	Physical Cell Identifier
LAC	Location Area Code
TAC	Tracking Area Code
MNC	Mobile Network Code
MCC	Mobile Country Code
ARFCN	Absolute Radio Frequency Channel Number
PLMN	Public Land Mobile Network

The behavior of TCP on LTE shall be determined by evaluating TCP communication throughput and round trip times. To measure these parameters, a passive measurement method based on sniffing the available traffic of a connection is utilized.

1) *Passive measurement of RTT*: The RTT is an important metric in determining the behavior of a network connection. It describes the period of time it takes to send a packet to a receiver and to get the corresponding acknowledgement to this packet. One common method to measure the RTT is done via injecting an ICMP message into the network. Due to the lack in the network neutrality, it may happen that this message does not get routed directly through the network. Likewise, the measured RTT with ICMP could be higher than for TCP.

To measure the round trip time within a TCP connection in a passive way, several methods are described in section II. For our measurement tool, different algorithms will be used. Based on [6], the RTT is calculated utilizing the three way handshake of a TCP connection. First, the RTT is measured during the establishment of the communication. Additionally, the RTT is measured with the FIN-ACK method from [7]. This algorithm is based on the connection closing procedure of TCP.

2) *Throughput measurement*: In general, throughput (TP) is the amount of data sent over a channel within a defined time frame [19]. To measure the throughput of the actual TCP flow, we can simply count the payload of each TCP segment and divide it by the transfer time. Before a TCP flow achieves its steady-state throughput, a sufficient amount of bytes has to be transmitted. This behavior is caused by the slow-start phase to avoid congestions. To determine the maximum-throughput of a connection, it is important that the flow finished the slow-start phase. To detect the end of the slow-start, different measurements were conducted. By downloading files of different sizes and analyzing the average throughput, we figured that, only flows larger than 800 kB can leave the slow-start phase. This result is confirmed by [20]. They detect that flows larger than 1 MB can leave the slow-start phase. So, to measure the maximum throughput, only flows of this size have to be evaluated.

IV. IMPLEMENTATION OF THE MEASUREMENT TOOL

Within our work, we develop an application to monitor a vehicle-to-server communication based on the mobile network. The main focus is to provide an architecture to easily extend the functionality of the software. The following section describes our hardware setup and our software architecture.

A. Hardware

To avoid running out of computing performance we used an in-vehicle computer, equipped with an Intel i7-5650U and 8 GB of RAM, as a measuring platform. It is equipped with an MC7304 LTE module from Sierra Wireless which is able to handle all available mobile network frequencies utilized in the EU. The module supports a data rate up to 100 MBit/s. To reach a high alliterative of LTE, we use external antennas which are mounted on the roof of the vehicle. These antennas provide an additional gain of +12 dB*i*. For measuring the position, we are using a uBlocks UBX-G6010 GPS receiver.

B. Software

Our tool is implemented in C++ and uses the *Boost*-, the *libpcap*-, the *libqmi* and the *libgps* libraries. Figure 1 illustrates the main architecture. The application is structured into three main parts:

- **Input:** The tool sniffs data from the LTE modem, the actual network interface and a GPS device. For accessing information from the LTE modem, the QMI-protocol is used. The sniffing is done with *libpcap* and the single packets for a connection are stored within a list. The GPS data is extracted from the GPS device with the help of *libgps*.
- **Processing:** Within the processing block, first, the data are aggregated. Then, the network parameters are calculated in a passive way. In this step, the previously described measurement methods are utilized.
- **Output:** It is possible to save the measurement results within a SQLite-Database, CSV-File or to write the results into a named pipe as a JSON object. This is useful for processing the data in a next step. For instance, we want to push the measurement results instantly to a server to analyze them.

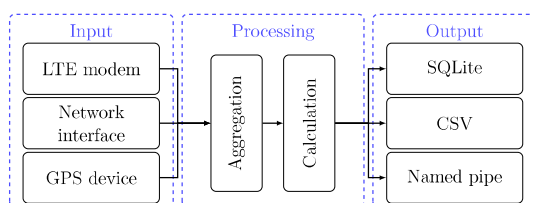


Figure 1: Block diagram of the measurement tool. It collects TCP, mobile network and GPS parameters and store the data locally

The main focus of the application is to sniff, to analyze, and to provide the measurement results in a suitable format. A flexible configuration of the tool allows to enable or disable single

measurements and to adjust the frequency of the measurement. We monitor the network with a rate of 3.33 Hz.

V. MEASUREMENT CAMPAIGN

We conducted a comprehensive measurement campaign using our measurement tool and the hardware described in section IV-A. To generate network traffic during the campaign, the data transfer is simulated with a script. It periodically downloads a 4 MB file from a server, which is located at OTH Amberg-Weiden. To keep the server load small, a method which constraints the maximum simultaneous downloading clients to two is implemented. The server connection supports 150 Mbit/s. Our mobile data plan supports 100 MBit/s, such that it is guaranteed that the measurement server is not the bottleneck of the connection. Figure 2 illustrates the measuring process. The LTE related data are collected cyclically. Mea-

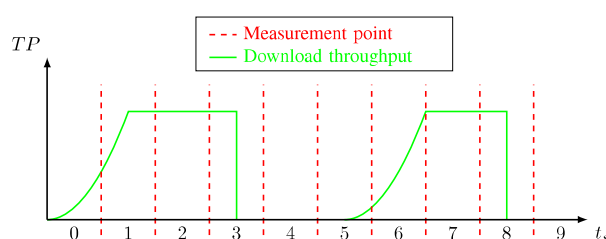


Figure 2: Illustration of the TCP and LTE parameter measurement process. A simplified trend of the throughput is shown in green. The red dashed lines indicate the measurement points.

surement points are indicated with red dashed lines in the figure. In our case, measurements are taken with a frequency of 3.33 Hz. The green line shows the trend of the throughput in a simplified way. In the beginning of the data transfer, the throughput is growing, due to the slow-start phase. In interval 1, the steady-state of the throughput is reached. Since we want to predict the maximum available throughput, it is important to delete the values for interval 0 and 1. In interval 3, the throughput measurement stops, when the data transfer is finished. So for the illustrated throughput trend in the figure 2, we get four valid throughput values, for interval 2, 3, 7 and 8.

For retracting the data, a test round is defined around Amberg, Germany. It covers different scenarios like urban, interurban and motorway roads. Figure 3 illustrates the test round. The circuit has approximately 28 km and is evenly split into three parts. Each part represents a specific scenario. In the urban scenario, the speed limit is 50 km/h and the way of driving is determined by stop and go traffic. The population density and consequently the amount of mobile devices is higher, but the 4G network coverage is fully developed within this area. The interurban scenario exhibits a speed limit of 100 km/h and shows a better traffic flow, with less intersections and traffic lights. The last scenario is the motorway. In Germany, there is no speed limit, but within our measurement campaign, the limit is set to the advisory speed of 130 km/h. On the motorway, the traffic flow was very constant during our test



Figure 3: Defined test area around Amberg, Germany. The circuit contains urban, interurban and motorway parts in one track.

drives and the 4G network coverage is also fully equipped. The measurement campaign was conducted in November 2017 within a week and consists of 46 rounds. The whole dataset is stored in a PostgreSQL database and contains 136 000 data points with 35 features.

VI. IMPLEMENTING DIFFERENT CONNECTIVITY MAPS

Forecasting the Quality of Service (QoS) of a mobile connection is helpful for several applications. To implement different prediction methods and algorithm, a generic framework, based on Python, is developed. Here, our dataset is used to implement different Connectivity Maps (CM). A CM is a geographical based approach to provide the network status for specific positions and areas. Therefore, the CM aggregates the data from our dataset based on the GPS position of the measurement. Basically, a CM consists of two parts. A geo-database contains the actual dataset and provides the structure of the road network. Additionally, the geo-database supplies a map data structure for aggregating the measurements to it. We use the PostGIS extension for PostgreSQL to setup such a geo-database. The second part of the CM is a visualization to show the results within a real map. It consists of two layers. The first layer is the road map and in our work, it is based on the free available OpenStreetMaps (OSM). The second layer defines the segments with the corresponding network condition for this section. We present three different types of CMs, for predicting the throughput of a connection. Before developing the CM, it is necessary to pre-process our dataset; for example, due to the inaccuracy of GPS, the data points need to get map matched to the correct road. We developed a map matching with the help of the PostGIS functionality. By defining a threshold of 5 m around the road, the data points get matched to the closest road. For a CM, this simple map matching algorithm is sufficient.

A. Connectivity Map based on OSM segments (CM-OSM)

The first approach for setting up a CM is based on OSM segments. The roads within OSM maps are composed of different segments as depicted in figure 4. The length of the single segments is fluctuating and depends on how the user

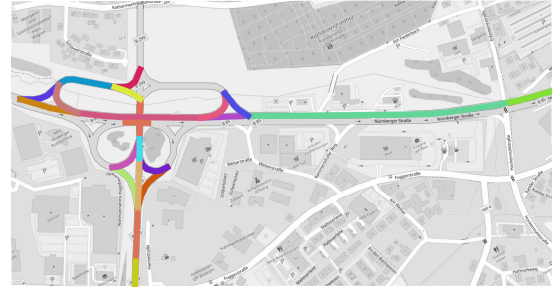


Figure 4: Exemplary OSM segments on the test circuit. It shows the variance of the segment length.

drew the map. For each segment, the throughput measurements get aggregated using the median. Since the OSM segment structure already exists, this type of CM is easy to generate and it is no problem to scale it on a larger area. Our test round consists of 188 segments with an average length of 148.93 m.

B. Connectivity Map based on equidistant segments (CM-Manual)

To address the issue of different segment length from the CM based on OSM segments, a second method is implemented. Instead of using the OSM segments, the circuit is split into equidistant road sections with a length of 500 m. The structure for our circuit consists of 56 segments. Since our test round is slightly longer than 28 km, with this type of CM the information for a short road section is lost. This is also one disadvantage for this kind of CM. The process of generating the equidistant segment is difficult and it is hard to scale this method to a bigger area. The throughput data gets aggregated for each segment, again.

C. Connectivity Map based on a grid (CM-Grid)

The third approach is to map the data to a grid instead of matching it to segments. Therefore, a grid layer is generated. Each single tile is quadratic and covers an area of 0.25 km². The throughput data is aggregated for each tile. So, with this kind of CM, it is possible to predict the throughput for each single tile. Here, it is not necessary to conduct a map matching, since the tiles are overlapping the roads. Compared to the aforementioned version of generating equidistant segments, the grid based method is easy to generate and also provides an equidistant predicting range.

VII. EVALUATION OF CONNECTIVITY MAPS

To compare the performance of different CMs, the next step is to validate every single model. For validation, the predicted value of each CM type is compared with four randomly chosen rounds from the dataset. For each model, the root mean square relative error (RMSRE) as shown in [21] is selected as defined in equation 1, where \hat{X}_i is the predicted and X_i the measured throughput value at the time i .

$$RMSRE = \sqrt{\frac{1}{n} \sum_{i=1}^n \left(\frac{\hat{X}_i - X_i}{\min(\hat{X}_i, X_i)} \right)^2} \quad (1)$$

A. CM-OSM

For the CM based on OSM segments, the RMSRE has a range from 54.12% to 67.38%. The error deviation of the relative error for each validation round is visualized in figure 5. All rounds show outliers up to 300%.

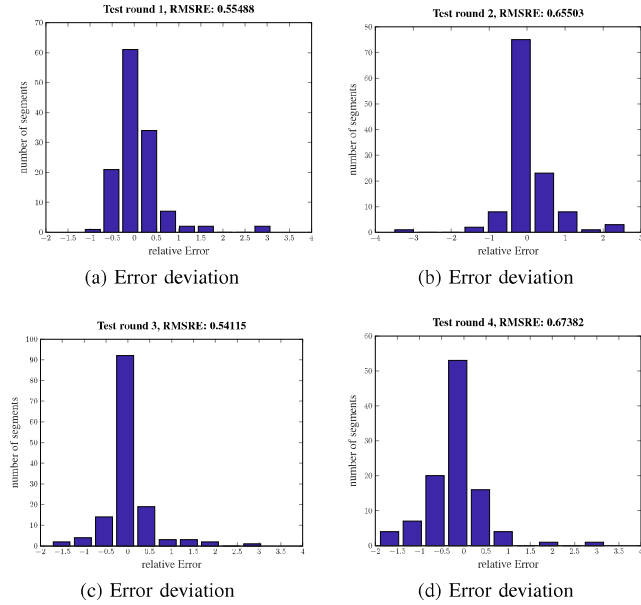


Figure 5: CM-OSM: Error deviation of validation round 1-4 for the download throughput using a OSM segment based CM

Particularly, noticeable is that the amount of outliers over all test rounds is similar. By investigating these segments, it is possible to elaborate coherences between them. It can be shown that the length of these bad segments is always smaller than 150m. Better results can be achieved for segments which are longer.

B. CM-Manual

The CM based on equidistant segments addresses the problem of having too short and different segment sizes. The RMSRE has a range from 32.54% to 50.53%. The error deviation is shown in figure 6. Most of the values can be predicted with an error smaller than 10%, but there are still outliers up to 150%. It can be seen that all outliers are located within the urban and interurban scenario, but not on the motorway. It can be supposed that predicting the throughput within areas with a higher population density is harder.

C. CM-Grid

The last CM is based on a grid. Figure 7 shows the deviation of the relative errors for predicting the throughput. The RMSRE has a range from 31.41% to 48,40%. There are again outliers up to 150%. It can be noticed that those outliers are on one hand located in urban and interurban areas and on the other hand those tiles just contain a short part of the defined test round. Due to the segmentation of the tiles,

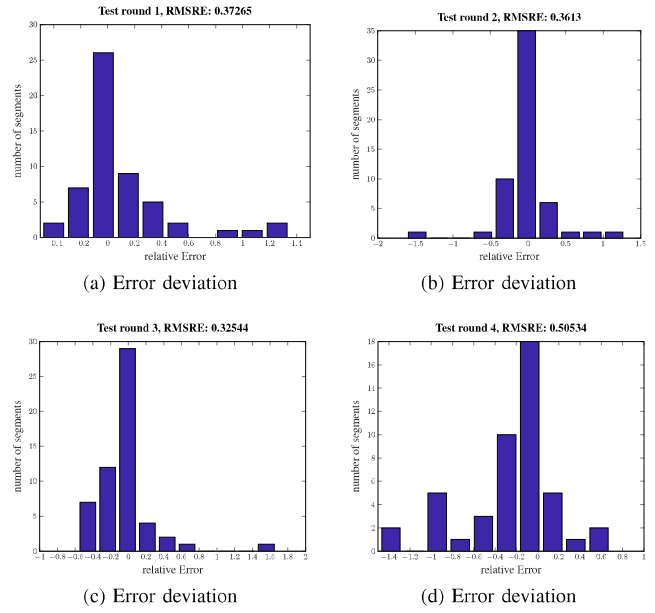


Figure 6: CM-Manual: Error deviation of validation round 1-4 for the download throughput using a fixed length segment based CM

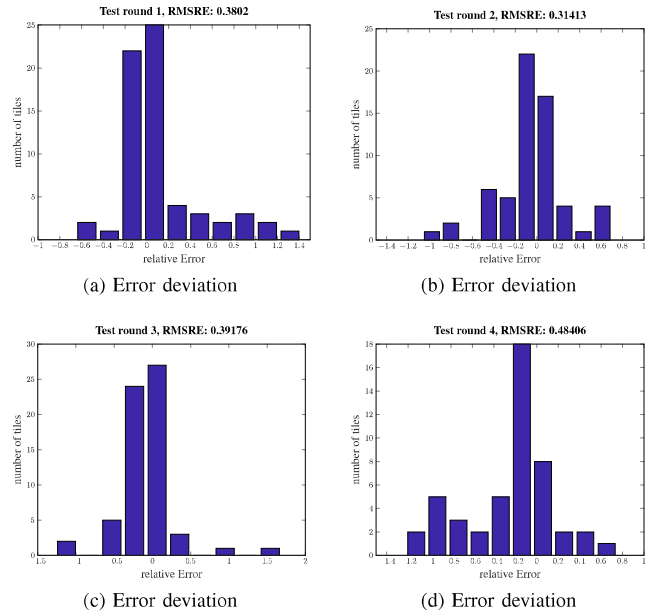


Figure 7: CM-Grid: Error deviation of validation round 1-4 for the download throughput using a grid based CM

it is inevitable that some of them just contain a short section of the test round. With a grid based CM one has equidistant tiles, but this does not guarantee that the road segments have the same length.

Summarizing, with the CM based on equidistant segments,

the best results for predicting the throughput can be achieved. The grid based approach also performs well and due to its simpler implementation, this solution has more advantages. The OSM based version achieves no satisfactory results. All the results are summarized in Table II. Also notable is that

Table II: Performance of the different Connectivity Maps

	Min. RE [%]	Max. RE [%]	RMSRE [%]
CM-OSM	0.07	355.03	60.62
1. Round	0.24	310.18	55.49
2. Round	0.58	355.03	65.50
3. Round	0.07	308.99	54.12
4. Round	0.90	320.09	67.38
CM-Manual	0.08	169.93	39.12
1. Round	0.16	134.81	37.27
2. Round	0.08	162.07	36.13
3. Round	0.38	169.93	32.54
4. Round	1.44	148.75	50.53
CM-Grid	0.15	168.94	39.25
1. Round	0.35	141.90	38.02
2. Round	0.15	106.54	31.41
3. Round	1.46	168.94	39.18
4. Round	1.43	122.37	48.40

the forth validation round always performs the worst. By analyzing the meta information of it, it can be shown that the elapsed time for this round is the longest. This indicates a high traffic density and confirms the assumption that predicting the throughput for a higher population density is more difficult.

VIII. CONCLUSION

In this paper, we presented a tool for passive monitoring a vehicle-to-server communication based on a 4G mobile network. In addition to the LTE parameters, we utilized a SYN-ACK and FIN-ACK method for measuring RTT. With the help of the tool, a comprehensive measurement campaign is conducted. By utilizing this dataset, different geographical based Connectivity Maps for predicting the throughput are developed. More precisely, a CM based on OSM segments, a CM based on equidistant segments and a grid based CM is developed. To compare the performance of each method, the deviation of the relative errors for each validation dataset and the RMSRE is calculated. It is shown that a CM based on equidistant segments performs best with an RMSRE of 39.12%. Our recorded dataset contains also network parameters like RSRQ or RSSI. So in the future, more algorithms for predicting the throughput shall be implemented and analyzed. For this purpose, history based algorithms and methods based on machine learning will be implemented. The goal is to provide a novel and comprehensive overview for different throughput prediction approaches.

REFERENCES

[1] A. Rathore, "State-of-the-Art Self Driving Cars," in *International Journal of Conceptions on Computing and Information Technology*, vol. 4, pp. 1-5, January 2016.

[2] A. Bokani, "Location-Based Adaptation for DASH in Vehicular Environment," presented at the *Proceedings of the 2014 CoNEXT on Student Workshop*, 2014, pp. 21-23.

[3] G. Zhong and A. Bokani, "A Geo-Adaptive JavaScript DASH Player," presented at the *Proceedings of the 2014 Workshop on Design, Quality and Deployment of Adaptive Video Streaming*, 2014, pp. 39-40.

[4] N. V. Mnisi et al., "Active Throughput Estimation Using RTT of Differing ICMP Packet Size," in *Third International Conference on Broadband Communications, Information Technology Biomedical Applications*, Gauteng, 2008, pp. 480-485.

[5] M. Jain and C. Dovrolis, "End-to-end available bandwidth: measurement methodology, dynamics, and relation with TCP throughput," in *IEEE/ACM Transactions on Networking*, vol. 11, no. 4, pp. 537-549, Aug. 2003.

[6] H. Jiang and C. Dovrolis, "Passive Estimation of TCP Round-Trip Times," in *ACM SIGCOMM Computer Communication Review*, p vol. 32, pp. 75-88, July 2002.

[7] A. Moosbrugger and P. Dorfinger, "Passive RTT measurement during connection close," in *18th International Conference on Software, Telecommunications and Computer Networks*, Split, Dubrovnik, 2010, pp. 392-396.

[8] B. Veal et al., "New Methods for Passive Estimation of TCP Round-Trip Times," in *Passive and Active Network measurement Conference*, p. 121-134, 2005.

[9] T. Pögel, J. Lübke and L. Wolf, "Passive client-based bandwidth and latency measurements in cellular networks," in *2012 Proceedings IEEE INFOCOM Workshops*, Orlando, FL, 2012, pp. 37-42.

[10] A. Samba, Y. Busnel, A. Blanc, P. Dooze and G. Simon, "Instantaneous Throughput Prediction in Cellular Networks: Which Information is Needed?," in *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, Lisbon, Portugal, 2017, pp. 624-627.

[11] J. Yao, S. S. Kanhere, and M. Hassan, "Improving QoS in High-Speed Mobility Using Bandwidth Maps," *IEEE Transactions on Mobile Computing*, vol. 11, no. 4, pp. 603-617, Jan. 2012.

[12] P. Deshpande, X. Hou, and S. R. Das, "Performance Comparison of 3G and Metro-scale WiFi for Vehicular Network Access," in *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement*, New York, NY, USA, 2010, pp. 301-307.

[13] I. D. D. Curcio, V. K. M. Vadakital and M. M. Hannuksela, "Geo-predictive Real-time Media Delivery in Mobile Environment," in *Proceedings of the 3rd Workshop on Mobile Video Delivery*, New York, NY, USA, 2010, pp. 3-8.

[14] K. Evensen et al., "Mobile video streaming using location-based network prediction and transparent handover," presented at the *Proceedings of the 21st international workshop on Network and operating systems support for digital audio and video*, 2011, pp. 21-26.

[15] B. Taani and R. Zimmermann, "Spatio-Temporal Analysis of Bandwidth Maps for Geo-Predictive Video Streaming in Mobile Environments," presented at the *Proceedings of the 2016 ACM on Multimedia Conference*, 2016, pp. 888-897.

[16] T. Pögel and L. Wolf, "Optimization of Vehicular Applications and Communication Properties with Connectivity Maps," in *2015 IEEE 40th Local Computer Networks Conference Workshops (LCN Workshops)*, Clearwater Beach, FL, 2015, pp. 870-877.

[17] J. Yao, S. S. Kanhere, and M. Hassan, "Using Bandwidth-road Maps for Improving Vehicular Internet Access," in *Proceedings of the 2Nd International Conference on COMMunication Systems and NETWORKS*, Piscataway, NJ, USA, 2010, S. 460-461.

[18] H. Riiser, T. Endestad, P. Vigmostad, C. Griwodz, and P. Halvorsen, "Video streaming using a location-based bandwidth-lookup service for bitrate planning," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 8, no. 3, Jan. 2012.

[19] M. Mathis and M. Allman, "A Framework for Defining Empirical Bulk Transfer Capacity Metrics," in *RFC 3148*, July 2001, pp 1-3.

[20] A. Gerber, J. Pang, O. Spatscheck and S Venkataraman, "Speed Testing without Speed Tests: Estimating Achievable Download Speed from Passive Measurements," in *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement*, Melbourne, Australia, 2010, pp. 424-430.

[21] Q. He, C. Dovrolis, and M. Ammar, "On the predictability of large transfer TCP throughput," in *ACM SIGCOMM Computer Communication Review*, 2005, vol. 35, pp. 145-156.