Title	Back to the future: Throughput prediction for cellular networks using radio KPIs
Author(s)	Raca, Darijo; Zahran, Ahmed H.; Sreenan, Cormac J.; Sinha, Rakesh K.; Halepovic, Emir; Jana, Rittwik; Gopalakrishnan, Vijay
Publication date	2017-10-16
Original citation	Raca, D., Zahran, A. H., Sreenan, C. J., Sinha, R. K., Halepovic, E., Jana, R. and Gopalakrishnan, V. (2017) 'Back to the Future: Throughput Prediction For Cellular Networks using Radio KPIs', HotWireless '17 Proceedings of the 4th ACM Workshop on Hot Topics in Wireless, Snowbird, Utah, USA, 16 October, New York: ACM, pp. 37-41. doi:10.1145/3127882.3127892
Type of publication	Article (peer-reviewed) Conference item
Link to publisher's version	http://dx.doi.org/10.1145/3127882.3127892 Access to the full text of the published version may require a subscription.
Rights	**© 2017 Association for Computing Machinery. This is the author's version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in HotWireless '17 Proceedings of the 4th ACM Workshop on Hot Topics in Wireless, http://dx.doi.org/10.1145/3127882.3127892**
Item downloaded from	http://hdl.handle.net/10468/4934

Back to the Future: Throughput Prediction For Cellular Networks using Radio KPIs

Darijo Raca, Ahmed H. Zahran, Cormac J. Sreenan
University College Cork
{d.raca,a.zahran,cjs}@cs.ucc.ie

Rakesh K. Sinha, Emir Halepovic, Rittwik Jana, Vijay Gopalakrishnan
AT&T Labs – Research
{sinha,emir,rjana,gvijay}@research.att.com

ABSTRACT

The availability of reliable predictions for cellular throughput would offer a fundamental change in the way applications are designed and operated. Numerous cellular applications, including video streaming and VoIP, embed logic that attempts to estimate achievable throughput and adapt their behaviour accordingly. We believe that providing applications with reliable predictions several seconds into the future would enable profoundly better adaptation decisions and dramatically benefit demanding applications like mobile virtual and augmented reality. The question we pose and seek to address is whether such reliable predictions are possible. We conduct a preliminary study of throughput prediction in a cellular environment using statistical machine learning techniques. An accurate prediction can be very challenging in large scale cellular environments because they are characterized by highly fluctuating channel conditions. Using simulations and real-world experiments, we study how prediction error varies as a function of prediction horizon, and granularity of available data. In particular, our simulation experiments show that the prediction error for mobile devices can be reduced significantly by combining measurements from the network with measurements from the end device. Our results indicate that it is possible to accurately predict achievable throughput up to 8 sec in the future where 50^{th} percentile of all errors are less than 15% for mobile and 2% for static devices.

KEYWORDS

Celluar network; throughput guidance; machine learning

1 INTRODUCTION

The achievable throughput for devices in cellular networks can fluctuate by an order of magnitude over a span of a few seconds for a variety of reasons. There can be rapid changes in the underlying radio channel conditions and system load as devices move and new devices enter and existing devices leave the network. Scheduling algorithms used in cellular networks cause burstiness in the cellular channel. In an end-to-end flow, data to a mobile device is regulated by protocols that operate at very short time scales (e.g. radio retransmissions at millisecond level) as well as congestion control protocols (e.g. TCP) at much larger time scales (hundreds of milliseconds to seconds) that are challenged by channel variations [12].

Some applications, including video streaming and VoIP, use embedded logic to adapt their behavior, such as encoding bitrate, based on local estimates of perceived throughput. Evidence, however, suggests that these applications would make significantly better adaptation decisions if they were provided with accurate Throughput Guidance (TG) [13]. We believe that TG would also dramatically benefit demanding applications like virtual and augmented reality. It would also open up new opportunities for sophisticated energy management on mobile devices, based on intelligent scheduling techniques matched to communication needs.

There exists recent work that shows promising results in terms of sub-second predictions of cellular throughput based on selected radio metrics [5, 8, 9]. Since radio conditions greatly impact the achievable throughput, there has been an effort to understand the impact more deeply and use Radio Access Network (RAN) data for TG [5, 10], focusing on prediction horizons of under one second. The IETF has also recently examined protocol changes to inform servers using TCP of the available throughput in cellular networks [2]. The horizon of all these predictions, however, is insufficient for applications video.

In contrast, in this paper, we seek to examine the limits of prediction; specifically, whether reliable forecasts are feasible several seconds into the future. We examine the challenges in producing accurate TG in cellular LTE networks. We explore the complex multi-dimensional parameter space, including different measurement data sources (device or network Key Performance Indicators (KPI)), feature engineering, mobile or static cases, various cell load indicators, cell sector clustering, prediction look-ahead horizons, number and time-granularity of past data samples. Our efforts focus only on the RAN KPIs, omitting geolocation or diurnal patterns. We conjecture that Machine Learning (ML) techniques are ideally suited to understand this 'non-linear' multi-dimensional space. We highlight some of the challenges in throughput prediction in the said context and provide some key takeaways.

Using a combination of simulations and real network data, we attempt to predict up to 8 seconds into the future, thus going beyond the state of the art which has mainly focused on sub-second prediction. Our preliminary results indicate that it is possible to accurately predict throughput over such future horizons; 50^{th} percentile of all prediction errors were less than 15% and 2% for mobile and static devices respectively. Further, our results show that that prediction error can be reduced significantly by combining measurements from the network with measurements from the end device. With this level of accuracy, we believe that deployment of TG capability would have a disruptive effect on the design and operation of both existing as well as emerging mobile applications.

, ,
2017. ACM ISBN 978-1-4503-0000-0.
https://doi.org/10.1145/0000

2 MULTI-DIMENSIONAL PARAMETER SPACE

Providing throughput guidance to each mobile device in a real-world cellular network is a complex systems challenge. For the predictions to be useful, the response has to be timely. Given that the typical round-trip time in LTE networks is a few tens of milliseconds, the time budget available to compute a response will typically be in the order of tens to hundreds of milliseconds. While this in itself may not be hard, scaling this to potentially millions of devices requesting for such predictions is a huge challenge.

There are many dimensions to how one can create a prediction model and the choices here offer different trade-offs. E.g., very fine-grain data can improve the accuracy of the prediction but there is a cost to gathering data at such a high frequency, and we would like to find the optimal price-performance sweet spot. Other trade-offs are even harder to guess without comprehensive evaluation. For example, we can create a separate model for each device or we can create a single model for all devices or something in-between. The first option (one model per device) is attractive because it is capturing the unique characteristics of the associated user behavior. On the other hand, combining multiple devices in a single model makes it more robust and less prone to over-fitting. Below we outline some of these design factors.

Length and frequency of past history: Instead of using just the current KPIs, we explore using historical samples of these KPIs. Having multiple samples makes us immune to short-term variations plus there is a tantalizing possibility of being able to capture 'trends' in KPIs. Also, capturing these trends plays a key role in indirectly inferring a scheduler that runs in eNB, as achievable throughput is not only dependent on radio conditions. Other users competing for resources from the same eNB will have a significant impact on predicted throughput. Scheduler implementation is vendor specific and for our experiments, represents a black box. A natural question is how much and how frequently should we collect these samples. Higher frequency translates into higher operator cost. With length of history, there may be a point beyond which it no longer represents the current or future conditions. A related question is how to summarize historical information. E.g., for each KPI, we can use all samples, take exponential weighted average, or take a few key metric such as median and mean.

Length of horizon: Different applications may need throughput guidance of different time lengths. E.g., a video rate adaptation algorithm trying to decide on the quality of an 8 second chunk would want to know the available throughput for the next 8 seconds whereas other applications may benefit more by throughput predictions in the very short or long term. Do we need fundamentally different techniques for these different types of predictions or can a single algorithm (with some tuning of parameters) be applied to all the cases.

How many devices per prediction model: As outlined earlier, we can either create one prediction model per device or create a model for a group of devices. With the former, we have the challenge of dealing with new devices where we do not have enough of a learning history to be able to make a prediction. With the latter, we need to find the right way to group the devices – by eNodeB, by eNodeB vendor, edge vs. center, static vs mobile, or something else.

Network Information: Can providing network related information help in throughput prediction? In device prediction model, a device does not have direct access to information such as the number of users connected to the same cell, the load on the cell, and channel conditions of other users. These features give a more direct assessment of eNB scheduler. Adding this information as new features can improve prediction accuracy.

Most previous research on throughput prediction learns some of this indirectly by observing history. E.g., if an eNodeB region is the environment we are trying to deduce, historical observations enable us to (indirectly) learn scheduling algorithm and connected users behaviour. Directly accessing that information can lower the amount of history we need, while also improving the prediction accuracy.

Algorithm Selection: In our work, we use machine learning algorithms for throughput prediction. We are exploring various ML techniques such as ensemble methods (Random Forest, Ada Boost, Gradient Boosting), Support Vector Machines (SVM), Gaussian Process (GP), Neural Networks and others. To compare these algorithms, we optimize each algorithm by tuning all their parameters via grid-search techniques on medium-scale horizons.

Error Analysis: Through our experiments, we get a better understanding of situations where prediction is particularly challenging. From our initial study, we observed that accuracy decreases when a user moves from one base station to another (**handover scenario**). A few possible hypothesis include the use of "invalid" history samples (user moving to new cell uses a history samples from the old cell), change of cell environment (number of active users, load), edge channel conditions.

3 METHODOLOGY

We conduct a large scale simulation using the ns-3 framework[1] as well as real world field experiments to validate some of the simulation results.

3.1 Simulation Setup

Using ns-3, we create a typical seven-cell hexagonal layout configuration with three sectors per cell, with 100 users randomly positioned across a cluster. Half of the users are downloading with maximum speed, and remaining users are uploading content. In addition, all users are moving with the constant speed of 50 mph (we use Gauss-Markov mobility model). We fix the inter cell-site distance to 500 m and cell transmission power to 44 dBm. The path loss model is log-distance, and carrier bandwidth is 10 MHz (50RBs).

3.2 Field Experiment Setup

We conduct static and mobile experiments in a real cellular network of a U.S. mobile operator using a laptop with an LTE dongle and QXDM[2] diagnostic software for collecting data and radio-level information. Each test run consists of downloading a large file, repeated under static and high-speed mobile conditions (between 25 and 60 mph). Run duration of collected traces is 35 minutes per trace.

[1] https://www.nsnam.org/
[2] https://www.qualcomm.com/

3.3 Data Preparation

We process logs from ns-3 for all simulations and QXDM log files for all field experiments in a similar fashion. We first time-align the data of all measurements by binning the data into five sampling intervals: 250ms, 500ms, 1s, 2s, and 4s. For each bin, we average the collected measurements. Here we only present results for 250ms sampling interval.

We consider various durations for data history and predicted future horizon in our evaluation. We define the horizon as the average value for throughput over given number of seconds in the future. For data history, a number of samples corresponding to the past duration are used as individual features.

3.3.1 Device and Network Prediction Features. We collect the following device-side metrics from both real and simulated data: SINR, CQI, PDCP delay, PDCP throughput, TBLER, RSRP, RSRQ, HARQ, and Application Throughput. In addition, we collect following network level metrics (simulation only): Competing Throughput (average throughput of all clients connected to the same base station), Competing CQIs (CCQI: average CQI of all clients connected to the same base station), Competing SINR (similar to CCQI, this value is average SINR for all clients at the same base station), Cell Load (number of users connected to the same base station).

3.4 Evaluation Metrics

To evaluate our methodology, we compare predicted throughput obtained from our trained model with the actual measured throughput. We compute the absolute prediction error (APE) as the ratio of absolute residual error and actual throughput, where the residual error is the difference between actual and predicted throughput. We show the 25, 50, and 75 percentiles and mean value in a box plot. The R^2 score is a measure of the goodness of the model compared to a naïve model.

4 RESULTS

4.1 Real-Data

In this section, we explore the performance of prediction algorithms using real traces collected from a major US mobile operator. We start by exploring whether ML techniques such as Random Forest (RF) with user KPI data offer any real benefits over prediction methods that rely only on past samples of throughput values. For ease of notation, Fx denotes a future horizon of x seconds. Similarly, Py denotes using y seconds of historical data for this prediction. So PyFx means that we are predicting average throughput for next x seconds using historical data from last y seconds.

4.1.1 Static Case. Future horizons: Fig. 1 shows the APE for different future horizons. Exponential weighted moving average (EWMA) and Autoregressive Integrated Moving Average (ARIMA) make their predictions based only on past samples of throughput values whereas RF creates a model out of user KPIs listed in section 3.3.1. It shows that RF technique provides the best performance for all future horizons, followed by ARIMA, then EWMA. RF reduces the 75th percentile APE by 21.5% and 5% with respect to EWMA and ARIMA, respectively. The accuracy of both RF and ARIMA improves noticeably as the prediction horizon increases but RF still outperforms ARIMA. Specifically, for an 8-second future horizon (F8), the 75th percentile APE drops to 9% and 14% for RF and ARIMA, respectively. Such improvement is due to the reduced variability observed in longer throughput samples averaged over larger interval.

We reran this experiment by reducing sampling frequency. While the APE went up with less frequent sampling, RF still outperformed ARIMA and EWMA. This shows that there is value in using Machine learning techniques with KPIs.

Comparison of ML algorithms: Generally, in most cases, RF outperforms SVM and GP with GP performing the worst. We discuss the comparative performance of SVM and RF in detail for the mobile case (Fig. 3a and Fig. 3b) but do not report any results for GP algorithm because of its significantly higher APE values.

4.1.2 Mobile Case. Similar to the static case, we start by investigating performance for EWMA, ARIMA and ML approaches. Fig. 2 compares APEs for three techniques. For the shorter horizon values (0.5s) ARIMA slightly outperforms RF in terms of 75th percentile APE. As we extend our future horizon, RF gains more ground, resulting in 18% for 75th percentile compared to 40% and 50% for the ARIMA and EWMA, respectively. Increasing the sampling interval leads to higher APE overall, similar to the static case.

Fig. 3a and 3b show comparison along three dimensions. First, they compare RF and SVM algorithms for multiple combinations of (future) horizon and past history. RF outperforms SVM in most cases. Compared with static case, the overall APE increases 10-20% depending on the chosen algorithm, which is intuitive, as the impact of environment change (including channel and number of users) is more pronounced.

Second, Fig. 3a compares different lengths of history for a fixed horizon. Increasing how far do we look into the past has no significant impact on APE until we cross certain threshold (4s). This threshold is half of the horizon duration, indicating that we need at least that amount to start inferring a trend. In the future work, we'll explore this assumption in greater detail. Comparing case P0.25F8 and P8F8 we observe a decrease of the 75th percentile of 5% on average for both, SVM and RF algorithms.

Finally Fig. 3b shows the overall trend as we fix history but predict further and further into the future. The 75th percentile of APE drops by 50% and 25%, from P2F0.5 to P2F8, for SVM and RF respectively. Again, smoothing out throughput variations helps in reducing overall APE.

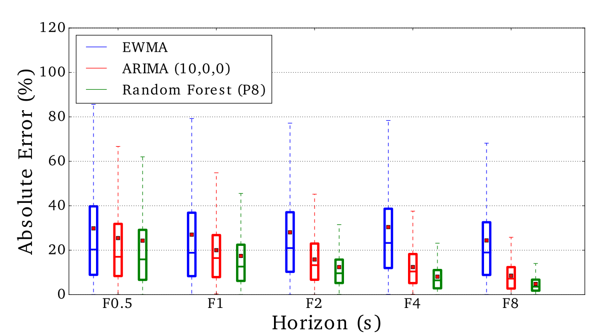


Figure 1: Comparison between different approaches for prediction of future throughput value (real data, *static* case

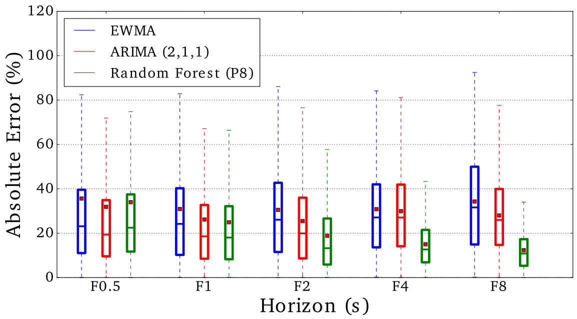
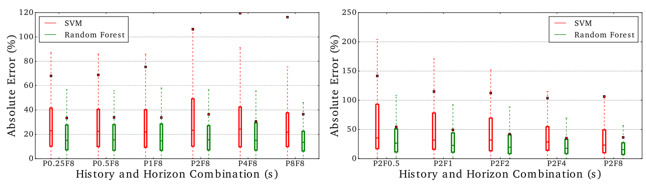


Figure 2: Comparison between different approaches for prediction of future throughput value (real data, *mobile* case)



(a) Fixed horizon (8s) and various history span

(b) Fixed history (2s) and various horizon values

Figure 3: Comparison between SVM and RF for particular arrangements (real data, mobile case)

4.2 Simulation data

In this section, we explore how network-related information can help to boost the prediction accuracy of the network throughput.

We start our analysis with pure static case, and evaluate the impact of integrating network information on the APE. We conclude that integrating network information would not introduce any significant benefits in a *fully* static environment due to its high predictability. We omit the figures due to space limitation.

However adding the network information in the prediction engine for the mobile case results in a noticeable reduction in the APE for all history-horizon combinations. Fig. 4 illustrates this by showing the actual throughput together with the predicted time series with device-based and device+network predictors. As an example, P8F8 APE drops by 15% after considering the additional network information. The benefits of adding network information varies among scenarios. For shorter horizons, it appears that the incremental gains are less significant. However for short history, adding network information results in less pronounced variations in predicted throughput and "faster" response to an abrupt change of throughput, compared to device only prediction. This result confirms our assumption that shorter history limitations can be alleviated with network information.

5 RELATED WORK

Existing solutions present several shortcomings. There are different strategies or heuristics that have been applied depending on the prediction horizon. For example, some studies have proposed to perform active measurements by sending a sequence of short data packets to estimate the current throughput, Round Trip Time (RTT)

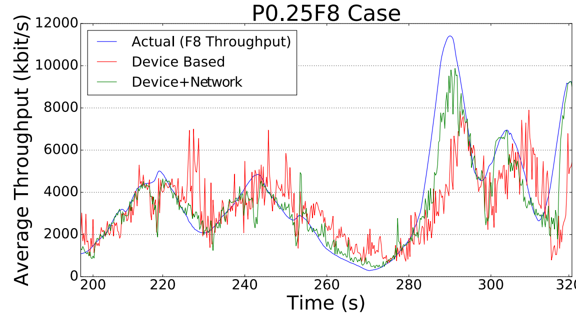


Figure 4: Time-series prediction for device based and device+network approach (short history)

and the packet loss rate [3]. This approach requires exchanging data before making a decision. Other studies rely only on a device's instantaneous radio channel quality indicator (CQI) and base its future decision on that estimate [9]. However, these are targeted for very short time intervals of the order of 500 msec.

Xu et al. propose a framework for forecasting packet loss, delay, and throughput [11]. Based on cellular traces, throughput prediction is obtained for 0.5 second based on a history of 20 seconds using a time-series with regression trees. CQIC [5] proposes another time-series based solution for predicting throughput up to 0.5 sec into the future. They use a product of CQI and discontinuous transmission ratio (DTX) to bit rate mapping function for throughput prediction. The 90th percentile for the average absolute error is less than 30%. Our prediction model has similar error values, but note that all the ML algorithms are tuned for larger horizons as we focus on medium time-scale. Sayeed et al. use an auto-regressive ARIMA based time-series model taking very specific parameters such as SINR and MCS as inputs to first predict the number of received bits per PRB and then translate that to effective throughput [8]. Their experiments are evaluated for a stationary device under different channel configurations. Note that network behaviour over long periods (hours, days, months) has been explored as well, with different types of data and techniques [7].

All these techniques rely on very specific selected device and network parameters for a short prediction horizon of 500 millisecond. However, certain applications, like video streaming, relies on prediction horizon of a couple of seconds. It is not at all clear if these techniques provide the best mechanism for prediction under a wide variety of device and network conditions for a range of different prediction horizons. Throughput prediction is inherently a 'non-linear' problem. We conjecture that throughput prediction for 'all' horizons can only be learnt using sophisticated ML techniques that model this complex multi-dimensional parameter space.

There are also proposals that are very application specific (e.g. video streaming). For example, Zou et al [13] propose an algorithm for HTTP adaptive streaming that relies on accurate forecast of average throughput. Their solution leads to significant improvement in video QoE compared to other state-of-the-art approaches [1, 4]. In a similar vein, Mangla et al. design an adaptation algorithm that takes prediction errors into account when making a decision for the next chunk [6]. Some solutions look for patterns of similarity between sessions to predict what QoE will the new session have, where similarity is determined through coarse-grained geographic and network features, not precise network performance measurements [9].

6 DISCUSSION

While we have provided some preliminary answers to cellular throughput prediction using ML, there are several open questions that need to be explored.

Feasibility (scalability) of having eNode prediction model vs. regional prediction model.

In theory, we could have the prediction based on data from each eNB separately. However, the architecture of present day cellular networks where user traffic tunnels via a packet core makes this infeasible. Clustering groups of cells into regions also results in less overhead of handling mobile users that move between cells.

Are ML algorithms robust enough to learn this time-varying non-linear multi-dimensional space or is our prediction accuracy biased towards collected traces?

We conducted initial experiments in the simulation environment, where we first wanted to explore how much value do they add. We did a very controlled experiment in the simulation environment by trying different reporting frequency to see what is the right trade-off point in "difficulty of reporting data" vs. prediction accuracy. There are many degrees of freedom that need to be explored carefully such that we can perform prediction along multiple time horizons accurately.

We are assuming that we can separate dataset based on mobility pattern (static vs. moving). *Does the network have this information or do we need more clever approach to know this?*

Classifying users based on mobility is not trivial. There can be multiple layers of classification. How do we know that we are not under fitting or over fitting the learning models?

How accurately can we make a prediction for users who have just entered a new cell or have recently connected to the network?

While we seek to build prediction region to minimize mobility of users from one region to another, it is still possible that such transitions do happen. Even assuming we carry over the models from one region to another, there will still be a brief period where we do not have accurate predictions in the new region. As part of our future work, we plan to explore approaches such as maintaining an average throughput for each prediction region (esp. cell-edge) and briefly use this as an estimate of future throughput.

Our initial results look promising. But maximum error (90th percentile) is still too high indicating a presence of outliers. Identification of situation that causes these spikes in error is needed. Further exploration of data abstraction and tuning of machine learning algorithms need to be further explored.

7 CONCLUSION

We posed the question *how well can cellular throughput be predicted?*, the answer to which can be useful for numerous applications that make decisions based on expected network performance over a future time horizon. To answer this question, we set a task to conduct a thorough quantitative study in which we systematically applied ML techniques to holistically examine this multi-dimensional parameter space. Using real and simulated network data, we want to explore the feasibility and effectiveness of TG. Our initial findings suggest that a predictor using machine learning can significantly reduce the prediction error (especially for mobile users) compared to a naive predictor. As part of future work, we plan to further investigate the multi-dimensional space, focusing more on feature engineering, as we feel that better abstraction of feature history values can capture trend and variation in data more accurately.

ACKNOWLEDGEMENTS

The authors acknowledge the support of Science Foundation Ireland (SFI) under Research Grant 13/IA/1892.

REFERENCES

[1] Te-Yuan Huang, Ramesh Johari, Nick McKeown, Matthew Trunnell, and Mark Watson. 2014. A Buffer-based Approach to Rate Adaptation: Evidence from a Large Video Streaming Service. In *Proceedings of the 2014 ACM Conference on SIGCOMM (SIGCOMM '14)*. 187–198.

[2] A. Jain, A. Terzis, H. Flinck, N. Sprecher, S. Arunachalam, and K. Smith. 2015. Mobile Throughput Guidance Inband Signaling Protocol. (2015). https://tools.ietf.org/html/draft-flinck-mobile-throughput-guidance-02

[3] Manish Jain and Constantinos Dovrolis. 2003. End-to-end Available Bandwidth: Measurement Methodology, Dynamics, and Relation with TCP Throughput. *IEEE/ACM Trans. Networking* 11, 4 (August 2003), 537–549.

[4] J. Jiang, V. Sekar, and H. Zhang. 2014. Improving Fairness, Efficiency, and Stability in HTTP-Based Adaptive Video Streaming With Festive. *IEEE/ACM Transactions on Networking* 22, 1 (Feb 2014), 326–340.

[5] Feng Lu, Hao Du, Ankur Jain, Geoffrey M. Voelker, Alex C. Snoeren, and Andreas Terzis. 2015. CQIC: Revisiting Cross-Layer Congestion Control for Cellular Networks. In *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications (HotMobile '15)*. 45–50.

[6] Tarun Mangla, Nawanol Theera-Ampornpunt, Mostafa Ammar, Ellen Zegura, and Saurabh Bagchi. 2016. Video Through a Crystal Ball: Effect of Bandwidth Prediction Quality on Adaptive Streaming in Mobile Environments. In *Proceedings of the 8th International Workshop on Mobile Video (MoVid '16)*. Article 1, 6 pages.

[7] Ashkan Nikravesh, David R. Choffnes, Ethan Katz-Bassett, Z. Morley Mao, and Matt Welsh. 2014. *Mobile Network Performance from User Devices: A Longitudinal, Multidimensional Analysis.* In Proceedings of the 2016 Passive and Active Measurement Conference (PAM '16), 12-22.

[8] Z. Sayeed, E. Grinshpun, D. Faucher, and S. Sharma. 2015. Long-term application-level wireless link quality prediction. In *2015 36th IEEE Sarnoff Symposium*. 40–45.

[9] Yi Sun, Xiaoqi Yin, Junchen Jiang, Vyas Sekar, Fuyuan Lin, Nanshu Wang, Tao Liu, and Bruno Sinopoli. 2016. CS2P: Improving Video Bitrate Selection and Adaptation with Data-Driven Throughput Prediction. In *Proceedings of the 2016 ACM SIGCOMM Conference (SIGCOMM '16)*. 272–285.

[10] Xiufeng Xie, Xinyu Zhang, Swarun Kumar, and Li Erran Li. 2015. piStream: Physical Layer Informed Adaptive Video Streaming over LTE. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking (MobiCom '15)*. 413–425.

[11] Qiang Xu, Sanjeev Mehrotra, Zhuoqing Mao, and Jin Li. 2013. PROTEUS: Network Performance Forecast for Real-time, Interactive Mobile Applications. In *Proceeding of the 11th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys '13)*. 347–360.

[12] Yasir Zaki, Thomas Pötsch, Jay Chen, Lakshminarayanan Subramanian, and Carmelita Görg. 2015. Adaptive Congestion Control for Unpredictable Cellular Networks. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication (SIGCOMM '15)*. 509–522.

[13] Xuan Kelvin Zou, Jeffrey Erman, Vijay Gopalakrishnan, Emir Halepovic, Rittwik Jana, Xin Jin, Jennifer Rexford, and Rakesh K. Sinha. 2015. Can Accurate Predictions Improve Video Streaming in Cellular Networks?. In *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications (HotMobile '15)*. 57–62.