CLOUD ANALYTICS FOR SHORT-TERM LTE METRIC PREDICTION - CAPACITY, CLOUD FRAMEWORK AND PERFORMANCE

Zulfiquar Sayeed
Bell Labs, Nokia
zulfiquar.sayeed@nokia.com

Abstract

The knowledge of a future link quality for a user equipment (UE) in wireless networks may be used by the application and TCP layer so that user's experience is optimized. The metrics that are relevant are available at the core of the radio link and the application or TCP layers have no knowledge of the wireless metrics. With a signaling protocol to the application/TCP layer and a prediction mechanism of the future link quality the applications will be able to avoid buffer overflows and/or congestion. In this paper we identify metrics that are suitable for application/TCP control, and analyze the performance of the prediction of the metrics. We show that the prediction of wireless metrics can be made with low error (3-10% MAPE) even with high cloud latency. This is a significant result as it states that predictions in the cloud of even short term LTE metrics are possible and that the predictions are fairly accurate to influence the proper operation of applications/TCP in wireless environments. We also show results on how short-term prediction increases the system capacity of a wireless system.

Keywords: Machine Learning, Predictive Analytics, System capacity; Data Analytics; Regression; Forecasting; Functional Regression; LTE; Latency; Wireless.

1. INTRODUCTION

Much work has been done in channel and Signal-to-Interference and Noise Ratio (SINR) prediction Ekman (2002), Avidor et. Al (2006), Duel-Hallen (2006), Ling (2006), Mathar (2013). However, the prediction of SINR or the channel coefficients may be able to predict the fundamental operation of the mobile's performance - but the actual throughput is a complex output of the channel conditions, other users in the cell, the base station scheduling, other algorithms and their parameters. The channel prediction may be too far removed from the actual metric of interest - which is the throughput in our case. The ultimate metric is the Modulation Coding Scheme (MCS) of the UE, and by extension, all UE's in the system. This higher order wireless metric prediction and its use in rate-control is what is novel in this paper. Prediction requires complex operations which the base stations may not economically perform due to hardware limitations. We need to devise an architecture that can perform the analysis away from the base station hardware and still be able to predict reasonably well despite the latency involved in cloud computing. The prediction framework, methodology and the signaling is what we claim to be novel.

Note that the TBS is determined by the number of physical resource blocks (PRBs) and the modulation and coding scheme (MCS) as defined in a lookup table in 3GPP TS 36.213. The difficulty of predicting the TBS with a good accuracy comes from two reasons. The first reason is that, that the MCS depends on the channel quality information and the signal to noise ratio (SINR), and the number of PRBs depends on resource allocation strategies and various network state variables (e.g., the physical channel state, traffic/data load and inter-cell interference level), TBS affected by all the above-mentioned variables (including some fine-scale structures and rapid phenomena) may vary a lot from time slot to time slot. The second reason is that the noisy and inaccurate measurements/reports increase the difficulty of learning a regression model for prediction.

As a background [Sessia et. Al (2009)] on LTE transmission and reception, the metrics of interest are calculated in the following manner:

1. The UE receives the downlink transmission from the eNodeB
2. The UE calculates the SINR of the received signal by way of embedded pilot tones in the received signal
3. The UE calculates the Channel Quality Indicator (CQI) based on capacity calculations (for example for AWGN channels) and feeds back the CQI to the eNodeB [Dhonti et. al (2011)]
4. The eNodeB receives the CQI and forms its understanding of the SINR of the UE.
5. The eNodeB obtains the number of Physical Resource Blocks (PRBs) to be allocated to the UE in the next Transmission Time Index (TTI) by using the cell load and its scheduler algorithm.
6. The SINR calculated in 4 is used to select the Modulation Coding Scheme for the UE in the next
TTI. Thus MCS is purely a channel quality driven metric.

7. The MCS and PRB thus calculated in 5 and 6 above is used to calculate the Transport Block Size (TBS) for transmission in the next TTI by way of the lookup table of 3GPP TS 36.213.

One example of the use of the prediction would be to control the data rate from an HTTP Adaptive Streaming (HAS) server to its client. In Famaey (2013) the algorithm for HAS is outlined that the HAS Client uses its buffer occupancy in requesting an increase or decrease of the video quality. If the prediction of expected rate is available at the next client request time - then either the client can delay the request or change the rate requested. TCP buffers often overflow from wireless link degradation Gurtov (2001). We believe MCS which is a direct indicator of the channel capacity of the UE is the ideal metric for rate control. This is the motivation for our work.

The modeling, function learning and prediction of higher level metric such as the MCS not only will try to capture the behavior of nature (as in SINR modeling) but also capture the evolution and dependencies that exist in possibly intractable eNodeB algorithms, and user behavior.

The rest of the paper is organized as follows. In Section 2 we describe the methodology used in our prediction. In Section 3 we describe the wireless network simulation and the parameters used. Section 4 explores our results. Section 5 demonstrates the effect f prediction of spectral capacity of wireless channels. Section 6 explores the effects of cloud latency on the prediction process and the cloud architecture. The paper concludes with our summary and next steps in Section 7.

2. METHODOLOGY

In this section we describe two methodologies for prediction of eNodeB internal metrics. We first look at the ARIMA method for SINR, MCS, PRB and TBS; and then look at a simplified Least-Mean-Squared (LMS) Regression technique based on polynomial fitting for SINR prediction.

2.1 ARIMA Method

We utilize the Auto-regressive Integrated Moving Average (ARIMA) models as described in Box (1994). The model is shown in Figure 1 Here, \( \theta_1, \theta_2, \ldots, \theta_q \) are the \( q \) feedback taps of the moving average part of the model; \( \phi_1, \phi_2, \ldots, \phi_p \) are the \( p \) feedback taps of the autoregressive part of the model and the \( d \) integrators complete the ARIMA\((p,d,q)\) model. Note that the integrators are necessary to model the non-stationarity of the observed data \( z_t \) with \( a_t \) being a Gaussian iid process with mean 0 and variance \( \sigma_a^2 \). Here \( z_t \) is not necessarily a zero mean process and hence the model requires the solution to find \( p + q + 2 \) parameters in addition to the order of \( d \) which is usually 0, 1, or 2 [Box (1994)]. In our work, \( z_t \) is sinr, mcs, prb or tbs. We use the statistical process to fit the model and to predict the process at a future time.

![Figure 1: ARIMA model used in Modeling and Prediction](image)

Figure 2 shows the block diagram of the prediction process of our metrics of interest (sinr is not shown - but is predicted as well). The metrics are generated by the eNodeB Base Band Unit (BBU) algorithms as described in Section 1. We apply a smoothing function (in our case an averaging every \( \tau \) samples) to smooth out fast variations and to see the effects of channel-memory and noise smoothing on the prediction performance. Note that the metrics are produced every TTI (that is 1 m-sec). Some TTI metrics may be zero - in which case we use an interpolator to fill in missing data and then smooth. We introduce (optionally) a non-linearity, which is \( \log_{10}(\cdot) \) in our case in line with producing homoscedasticity, if needed, as described in Moisseev (2009). We test the prediction performance both with and without the non-linearity. The ARIMA model fitting and the subsequent prediction is done in the R statistical tool. The training period is first studied to find the required sample size to train under the selected wireless conditions as described in Section 3. Once the suitable training duration is found we use that duration and then look at the behavior of the prediction performance under the wireless conditions, namely velocity of surrounding scatterers and smoothing duration.

Since one of our goals is to predict the throughput (which is nicely captured by the TBS per TTI) we are faced with two different methods. If we look at the TBS Table (Table Table 7.1.7.2.1-1) [3GPP (2009)] we see that knowing a predicted value of MCS and PRB the TBS value is uniquely found. Hence one method could be to predict only the MCS and the PRB and use those two values to find the corresponding predicted TBS. But as we shall later see, such look-up-table like prediction leads to high errors in our case. The reason may be in the convexity of the surface and
the application of Jensen's inequality. This convexity is shown in Figure 3.

![ARIMA prediction block diagram](image)

**Figure 2: ARIMA prediction block diagram**

To circumvent this performance degradation we use a functional regression analysis as shown in Figure 4. Here the same smoothing is applied as in the regular ARIMA based prediction as before. However now we learn a functional relationship between the input pair (MCS, PRB) and the output TBS. We use the technique fregre.np (nonparametric functional regression) from the R package in de la Fuente (2012). The input variable is \( X = \{MCS, PRB\} \) and the output variable is \( y = TBS \). We first find the functional relationship [de la Fuente (2012)]. The input and output is related by the relationship:

\[
y_i = r(x_i) + \epsilon_i
\]

where, \( \epsilon_i \) is the error term and \( r() \) is estimated by [de la Fuente (2012)]:

\[
f(X) = \hat{r}(X) = \sum_{i} K(h^{-1}d(X,X_i))y_i / \sum_{i} K(h^{-1}d(X,X_i))
\]

Where, we select the kernel \( K \) to be the Triweight kernel:

\[
K(u) = \frac{35}{32}(1 - u^2)^21_{|u| \leq 1}.
\]

We have evaluated with a variety of kernels and Triweight yields the best performance. \( h \) is the smoothing parameter (optimally evaluated within the function fregre.np) and \( d \) is a metric.

Once the function is estimated we use the predicted MCS and PRB pair to evaluate the predicted TBS value by way of the function \( \hat{r}( ) \). The R utility used from [de la Fuente (2012)] is predict.fregre.fd.

2.2 LMS Method

In this sub-section we look at a much simpler way of SINR prediction where we use Matlab’s polyfit() function to perform an extrapolation of the polynomial into the future time. The procedure is shown in Figure 5.

The raw SINR metrics are smoothed as before. The output of the smoothing has a time-stamp and a metric value in pairs. A polynomial fitting of varying order is performed to create a least-mean-square fit of the data over a certain training interval window, which slides as new input comes in from the smoothing block. The polynomial at each smoothing epoch is extrapolated into the training windows time index, plus one smoothing epoch to the right. The
value obtained at one time unit into the prediction horizon is the SINR predicted value.

Figure 4: TBS prediction using Functional Regression

Figure 5: SINR prediction using polyfit()

Once a new data point is received from the smoothing block, the new data value is used in the training interval and the process repeats into the future.

Later on in the paper we shall look at the relative performance difference of the ARIMA and LMS methods for SINR performance.

3. SIMULATION SETUP

In our simulation we generate the eNodeB metrics by using an Alcatel-Lucent sample level simulator. The salient settings of the simulator for this paper are as below:

- Carrier Frequency (DL): 700 MHz
- Carrier: FDD, 5 MHz (25 PRBs)
- Number of cells: Hexagonal 7
- UE of interest: center cell
- UE velocity: static
- UE local clutter velocity: {3, 10, 30} km/hr
- UE Receive mode: 2 Rx Antennas, Antenna Selection Off
- CQI reporting: periodic every 20 m-sec
- Rayleigh Channel Model: ITU.EPA
- Other Cell Load: 100%
- Simulation Duration: 800 Seconds
- Rayleigh Channel Model for 3 km/hr (ITU Pedestrian A Model):

<table>
<thead>
<tr>
<th>Relative Delay (ns)</th>
<th>Relative Power (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.0</td>
</tr>
<tr>
<td>110</td>
<td>-9.7</td>
</tr>
<tr>
<td>190</td>
<td>-19.2</td>
</tr>
<tr>
<td>410</td>
<td>-22.8</td>
</tr>
</tbody>
</table>

- Rayleigh Channel Model for 10 km/hr (ITU Pedestrian B Model):

<table>
<thead>
<tr>
<th>Relative Delay (ns)</th>
<th>Relative Power (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.0</td>
</tr>
<tr>
<td>200</td>
<td>-0.9</td>
</tr>
<tr>
<td>800</td>
<td>-4.9</td>
</tr>
<tr>
<td>1200</td>
<td>-8.0</td>
</tr>
<tr>
<td>2300</td>
<td>-7.8</td>
</tr>
<tr>
<td>3700</td>
<td>-23.9</td>
</tr>
</tbody>
</table>
Rayleigh Channel Model for 30 km/hr (ITU Vehicular A Model):

<table>
<thead>
<tr>
<th>Relative Delay (ns)</th>
<th>Relative Power (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.0</td>
</tr>
<tr>
<td>310</td>
<td>-1.0</td>
</tr>
<tr>
<td>710</td>
<td>-9.0</td>
</tr>
<tr>
<td>1090</td>
<td>-10.0</td>
</tr>
<tr>
<td>1730</td>
<td>-15.0</td>
</tr>
<tr>
<td>2510</td>
<td>-20.0</td>
</tr>
</tbody>
</table>

The output metrics SINR, TBS, MCS and PRB are saved once every TTI at the eNodeB. For SINR - since CQI reporting is periodic with 20 m-sec period the same value is output to the log file between CQI updates. The other metrics change according to the error control loop, scheduler and other algorithms in place at the eNodeB every TTI. If the UE is not to be scheduled in a TTI then the corresponding PRB and TBS will be zero. In the smoothing block of Figure 2 we take such zero possibilities into account and interpolate the values so that the averaging is meaningful given the averaging window size $\tau$. It is important to note that the fading velocities of 3, 10 and 30 km/hr at the carrier frequency of 700 MHz yields Doppler frequencies $\Delta f$ of 2.1, 6.5 and 20.8 Hz, yielding a coherence time of the channel ($1/\Delta f$) of 480, 154 and 48 m-sec respectively.

The static UE case with fading at the given velocities is of interest where the user may be sitting in a plaza with motion around him or her due to slow moving cars at differing velocities or traffic conditions. Note that fading is a result of relative motion of the UE, eNodeB or the clutter in between. The faster velocity may imply a downtown area with traffic flowing smoothly whereas the slowest velocity may correspond to an urban plaza during rush hour with low mobility of traffic.

Once the simulated metrics are available after the simulation, we feed the metrics into R to do the prediction/regression analysis.

4. Simulation and Analysis Results

4.1 ARIMA Method

In Figure 6 and Figure 7 we plot some sample traces of SINR, MCS, PRB and TBS from the simulation. We see that the 3 km/hr SINR curve shows less variation than the corresponding 30 km/hr curve. All metrics show some sort of memory in the processes which means that they can be analyzed by means of regression analysis.

We have analyzed the prediction performance using the metric called $\text{MAPE}$ (Mean Absolute Prediction Error) for SINR, TBS, MCS and PRBs as defined below.

$$\text{MAPE} = \frac{\sum_{i=1}^{N} \text{abs} (a_i - p_i)}{N} \times 100$$

For SINR, canonically, we also use the RMSE of the dB SINR to analyze the prediction performance. The reason for this is that when the eNodeB uses a table look up to map the SINR in dBs to select the MCS based on Shannon’s Capacity Equation, $\text{spectral efficiency (bits per sec per Hz)} = \log_2(1 + \text{SINR}_{\text{dB}})$.
SINR received from the UE. However, even though the Shannon equation uses the linear SINR; the table used in partitioning the SINR to MCS look-up-table is in dB scale. Hence, for our performance evaluation we shall use the RMSE of the dB difference of the actual SINR value to the predicted SINR value.

In Figure 8 we look at the effect of training duration on the performance of the ARIMA modeling. We look at 25 and 50 m-sec smoothing for the three velocities 3, 10 and 30 km/hr. We see that around 200 m-secs all metrics yield sufficiently low and steady prediction error. We choose our training duration to be 200 smoothed samples in the subsequent analysis. We use $\tau = 10, 25, 50, 75, 100, 150, 200$ m-secs as the smoothing window and for all window sizes we use 200 smoothed samples as the ARIMA training duration. However, note that with the ARIMA model training is only done once for the entire data set as the training is a computationally intensive process. This may suffice for a stationary UE in a Rayleigh fading environment where; however, with mobility that would create a performance loss or a complexity gain. With LMS this is alleviated as described in Sayeed et. al. (2015).

We look at an exemplary trace of the predicted and actual value of SINR (dB), with dB non-linearity for training, in Figure 9.

![Figure 8: Effect of Training Interval on Performance. Red, Green, Blue: 3, 10, 30 km/hr; solid, dashed = 25, 50 m-sec smoothing](image)

We see that, the actual SINR value trace loses the channel memory as the averaging is increased - and resembles a Gaussian noise process (AWGN is present in the simulation) and resembles less and less a Rayleigh Channel. When the averaging is low the traces look like a Rayleigh channel SINR. The ARIMA model tends to track the local mean of the metric as the averaging is increased rather than the process itself. The MAPE of the SINR (in dB) is presented in Figure 10.

![Figure 9: Prediction of SINR (Blue: Actual, Red: Predicted)](image)

![Figure 10: MAPE Performance of SINR Prediction (Red, Green, and Blue: 3, 10, 30 km/hr)](image)

The SINR MAPE performance shows an interesting phenomenon. As the averaging duration approaches the coherence time of the channel for the given velocity, the error rises. and then starts to taper off. The rise in error is due to the loss of memory in the data that trains the ARIMA model - and the decline in error is related to the noise reduction of the Gaussian process that remains after the channel memory is completely removed from the input metric time series. As such, too much averaging is not the optimum for use of ARIMA type processing for prediction.
If we look at the RMSE in dB of the SINR prediction, the nice form of the error trends are lost, however, the RMSE in dBs can be seen to be within 0.2 dBs for the optimum averaging window for both linear and log training of the ARIMA models. This is reflected in Figure 11. We have also analyzed the effect of adaptive training for our Static UE case - but adaptation yields no benefit as the log-normal and path loss is static. For mobility cases adaptation is expected to yield benefit - but that is for further study at present.

Exemplary prediction traces of the MCS, PRB and TBS is provided in Figure 12. Since the MCS is directly related to the SINR and eNodeB algorithms that depend on the SINR as input, we see that the MCS trace shape resembles the Rayleigh SINR shape. However, the trace of the TBS is related to the cell-load, the cell-load memory process and the scheduler algorithm. As such, the TBS does not follow the Rayleigh trace pattern. However, if we add more cell load or UEs to the eNodeB of interest in our simulation then the shape and memory of the process may change to some other form. This is to be further studied in our subsequent explorations of this topic.

The ultimate metric of interest, the TBS’s prediction performance is shown in Figure 13. As described in Section 2, the LUT method of TBS prediction yields very poor performance (The diamond \( \bigcirc \) points) because of the convexity of the table as a function of predicted MCS and PRB. The functional regression begins to perform as well as the direct linearly trained predictor at higher averaging window sizes. However, for the averaging window sizes of interest here - the direct linear training performance is the best (the circle ○ points. For higher averaging durations functional regression techniques may yield the best performance. That is for further study at present.

4.2 LMS Method

In the above subsection we have seen how the ARIMA method has been able to predict the SINR. We have seen that in Figure 11 is bounded below 0.35 dB error. In this section we take a look at the performance of a much more computationally efficient adaptive LMS based polynomial fitting algorithm as described in subsection 2.2.

The performance of polynomial fitting of orders one and two are shown in Figure 14. We see that the errors are bounded below 0.4 dB error. We see also that there is very little difference between 1\textsuperscript{st} order and second order fitting for prediction performance. Since the polynomial of order 1 is much less computationally intensive, we propose a straight line fitting for prediction. However, slower fading, a higher order polynomial may be better suited. This is for further study. We also see that as the velocity increases the RMSE of the 2\textsuperscript{nd} order and the 1\textsuperscript{st} order prediction performance differences narrow.

A 40 sample training implies a training duration of 2 seconds for the 50 m-sec prediction horizon point. This is quite significant, as this means that only after two seconds of detecting a flow in the wireless air-interface; we can start predicting the throughput (via Shannon’s Equation) and adapt the TCP or application layer behaviors.
For completeness, we also show the performance of the polynomial based prediction for a 100 sample training. This does show a slightly weaker performance than for the shorter training. In reality, the training period should be a function of the coherence time of the SINR as depicted in Figure 21.

The traces of predicted and actual SINRs are shown in Figure 16 for 3 km/hr Rayleigh fading. The RMSE for this velocity, for a 50 m-sec smoothing window and 40 sample training is 0.28 dBs only. Note that in the LTE SINRdB to MCS mapping table the average distance between MCS’s is about 2 dB.
5. EFFECT OF PREDICTION ON SPECTRAL EFFICIENCY OF WIRELESS CHANNELS

We would like to look at the effects of spectral efficiency for prediction. The spectral efficiency is defined as the number of bits/sec/Hz that can be transmitted over a given channel with SINR of $s$, in the linear scale, as defined by $S.E. = \log_2(1 + s)$.

![Figure 15: RMSE of LMS prediction of polynomial order 1 and 2 with 100 sample training](image)

![Figure 16: Prediction of SINR using polynomial fitting of order 1 at 3 km/hr (Blue: Actual, Red: Predicted)](image)

The sample trace of similar prediction is shown in Figure 17. The RMSE for this trace is 0.18 dB.

![Figure 17: Prediction of SINR using polynomial fitting of order 1 at 30 km/hr (Blue: Actual, Red: Predicted)](image)

The number $s_i$ is transmitted back to the eNodeB at a delay of $\tau$ time units, which is the reporting delay. The eNodeB uses $s_i$ to decide the modulation coding scheme (MCS) to use for the $(i+1)th$ packet [Sessia et al. (2002)].

The capacity of the $i$th packet, theoretically, can be $\log_2(1 + s_i)$. If $s_i > s_{i+1}$ then the packet is lost. If $s_i < s_{i+1}$ then the packet is error free.

Therefore, the b/s/Hz received over $N - 1$ channel uses is:

$$A = \frac{1}{N - 1} \sum_{i=2}^{N} \{\log_2(1 + s_i)\} \mathbf{1}_{i \leq s_{i-1} + \alpha_i}$$
Where, \(1_{\{s_i < s_{i-1} + \alpha_i\}}\) is the indicator function, the value of which is 1 when the condition of its argument is true and 0 otherwise. \(\alpha_i\)'s added to the argument, because a finite length code, as opposed to Shannon’s infinite-length code, by the Asymptotic-Equipartition-Principle (AEP) [Cover et. al. (1991)], can only come close to within \(\alpha\) in SINR. For our purposes, we pick \(\alpha_i = 0.1\) in linear non-dimensional units.

Now since, \(\{s_i\}_{i=1}^N\) has memory, \(\hat{s}_i\) can be predicted, by linear regression techniques, as we have seen in the last sub-section.

Now, let us say, \(\{\hat{s}_i\}_{i=2}^N\) are predicted at times \(i = 1, 2, … N\) from \(\{s_i\}_{i=1}^N\). At time \(i + 1\) the eNodeB uses \(\{s_j\}_{j=1}^{i}\) to find the SINR to use to select the MCS for the \((i + 1)\)th transmission.

Now the actual b/s/Hz over the \(N - 1\) channel uses is:
\[
B = \frac{1}{N-1} \sum_{i=2}^{N} \left( \log_2(1 + \hat{s}_i) \right) 1_{\{\hat{s}_i < s_i + \alpha_i\}}.
\]

We use the simulation of Section 3 to generate \(\{s_i\}_{i=1}^N\) a random sequence with memory. We calculate \(\{\hat{s}_i\}_{i=2}^N\) using a simple LMS technique of the previous sub-section from \(\{s_j\}_{j<i}\). We plot \(B - A\), from the above two equations, for \(\tau = 10, 25, 50, 75, 100\) m-secs for the three Rayleigh fading velocities, where path-loss was held constant. This is shown in Figure 18 which shows the b/s/Hz greater for predicted SINR based MCS selection capacity than the capacity without prediction for the \(\tau\)'s considered.

In fact, for a 5 MHz LTE carrier, for the case of 50 m-sec reporting delay, prediction yields a carrier capacity of 1 Mbps improvement over that of a system with no prediction. This is sufficient to allow one or two more HTTP-Adaptive-Streaming (HAS) user in the carrier, which is a significant capacity gain.

Prediction yields higher capacity because the eNodeB acts upon stale SINR to select the MCS’s. Prediction extrapolates past SINR to the future by taking advantage of the memory inherent in the SINR time series. Prediction thus becomes an important aspect of future wireless communications systems such as 5G.
6. Architecture

In Figure 20 we show our architecture for LTE metric prediction in the cloud. Here, ePC is the enhanced Packet Core of the LTE network. The eNodeB A’s internal metrics are smoothed locally and then transmitted to a cloud processor, incurring some delay, for predictive analysis and delivered to application server (A) and/or UE application client (communication to TCP Layer not shown).

Avidor (2006) has a closed form expression for SINR correlation - which indicates half the coherence time for $S/I$.
than Rayleigh faded signal ($S$). The correlations of 3 and 30 km/hr $S/I$ are shown in Figure 21. The correlation of the SIR (Signal to Interference Ratio) is derived in a generalized manner in the reference (Equation 16 in Avidor (2006)), and is found to be $1 + f_0^2(\omega_m \tau)$, where $\omega_m$ is the Doppler frequency in radians/sec and $\tau$ is the time-lag of auto-correlation calculation.

In Figure 22 we plot the normalized cross-correlation of the actual metric and the delayed predicted value of that metric vs. m-sec lag. All predictions remain more than 94% correlated (normalized) to the actual value even. This is a significant result as it states that cloud prediction of short term LTE metrics is possible and that the predictions are fairly accurate.

7. CONCLUSIONS

We have analyzed the prediction performance of some key metrics of the eNodeB that allow for the wireless network to be able to control the application behaviors that are running through the system. The control of such applications is problematic without the network metric knowledge. Such knowledge is extracted in abstraction by HAS applications [Cicco, et.al. (2014)]. However, with the prediction of future metrics (that may be indicative of the network delay, throughput for the users in the cellular system) a completely new way of controlling these application may be created - as we see in the predictive control theoretic papers [Li et.al 2007], [Liu et.al. (2007)], [Liu et. al. (2011)]. It is conceivable that the control of HAS processes and other processes that rely on "network sounding" feedback can benefit from predictive control. We wish to look at this feasibility and performance in the future.

We also wish to study the effect of variable cell load and mobility on the performance of prediction. We need to characterize the performance bounds that will make the prediction meaningful for different applications. Different applications will have different tolerances to prediction errors for stability. This needs to be characterized on a per application basis.

Our present study has shown that it is feasible to predict network conditions by regression analysis. As we noted before, the regression can model the collective memory effect of the channel, eNodeB algorithms, user and application behaviors. Thus it allows us to build a model for the concatenation of all these systems - which is difficult if not impossible to do analytically. In the future we plan on looking deeper into the model and prediction behavior under mobility and user migration - and ultimately to study application behaviors with and without predictive control.

8. REFERENCES


Long Range Prediction and Reduced Feedback for Mobile Radio Adaptive OFDM Systems, Alexandra Duel-


Improved networked predictive control with different transmission delays in both forward and feedback channels, X. Liu and Y. Xia, International Conference on Intelligent Control and Information Processing (ICICIP), 2011, pp 604-609

Authors

Dr. Zulfiquar Sayeed was born in Dhaka, Bangladesh. He received the B.S. in electrical engineering from the California Institute of Technology in 1990, the M.S. and Ph.D. degrees in electrical engineering from the University of Pennsylvania in 1993 and 1996, respectively. He also holds a B.A. in Liberal Arts from Ohio Wesleyan University. He has been with Bell Laboratories since 1997. He is currently involved in Predictive Content Services for Wireless Communications. He was previously involved with network modeling, algorithm development, simulation, and performance analysis of wireless systems. He was a key contributor to the system architecture and algorithm development of a state-of-the-art CD quality satellite digital audio radio system that is now fully operational. Dr. Sayeed holds 28 U.S. patents and has 18 pending applications. He is the 2016 recipient of the Thomas Alva Edison Award for telecommunications from the R&D Council of the State of New Jersey for his work in SDH Quality Timing Synchronization over Packet Networks.