This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE Globecom 2010 proceedings.

On Cognitive Network Channel Selection and the Impact on Transport Layer Performance

Yao Liu, Bheemarjuna Reddy Tamma, B. S. Manoj, and Ramesh Rao
University of California San Diego, San Diego, CA 92093, USA
E-mail: {yal019, btamma, bsmanoj, rrao}@ucsd.edu

***Abstract*—In this paper, we investigate the machine learning based strategies for dynamic channel selection in Cognitive Access Points (CogAPs) of WLANs. We employ Multi-layer Feedforward Neural Network (MFNN) models that utilize historical traffic information from network environment for learning the influence of spatio-temporal-spectral factors on the network and then predicting future traffic loads on each of the channels. Based on the future traffic loads, CogAP chooses the best channel for serving wireless clients. An important factor is the time scale of traffic prediction. We construct three kinds of traffic predictors that predict traffic at different time scales: MLP (Minute Level Prediction), MILP (Minute Interval Level Prediction), and HLP (Hourly Level Prediction) schemes and study their prediction accuracy. Experiment results show that MFNN predictors perform better than traditional autoregressive models in terms of prediction accuracy. In addition to accurate prediction, another factor that influences the design of cognitive network channel selection is the impact of channel selection strategy on the transport layer performance. We, therefore, conduct performance studies on the TCP throughput achieved on the above mentioned cognitive channel selection strategies. The MFNN predictors will also help CogAP to find and switch to the optimal channel, leading to a higher and more sustained throughput.**

I. Introduction

The presence of heterogeneous wireless networks and increasing wireless spectrum congestion add new dimensions and complexities to network configuration and performance optimization. The area of cognitive wireless networking involves developing communication network systems that have awareness about their operations and relationships among network parameters, network protocols, and the network environment, all towards a goal of effectively using this network awareness for stack-wide and network-wide performance optimization. In a cognitive network paradigm, all network elements track the spatial, temporal, and spectral dynamics of their own behavior and the behavior associated with the environment, and report that information to a cognitive controller. The information so gathered is used by cognitive controller to learn, plan, and act in a way that meets network or application requirements [1]. While Cognitive radios apply cognition only to the physical layer, mainly to dynamically detect and use white spaces in spectrum allocated to TV broadcasting, the objective of cognitive networks is to apply cognition to all layers of the network protocol stack and network components for achieving network-wide performance goals.

One of the applications of cognitive networking in wireless networks is the problem of dynamic channel selection in IEEE 802.11 Wireless Local Area Networks (WLANs). In most campus and home WLANs, the optimal operating channel varies as a function of space and time. In order to find the optimal operating channel in a cognitive WLAN, the cognitive controller needs to gather historical network traffic information across all channels and predict future traffic loads on each of the channels. However, traffic monitoring is very challenging in multi-channel wireless networks. The 802.11b/g-based wireless networks operate on ISM band and have their transmissions overlap multiple channels. Even though orthogonal channels are typically used for configuring APs, in some cases (*e.g.,* non-802.11 sources such as Bluetooth, microwave ovens, and other noise sources render an orthogonal channel useless) other channels are also being used in the configuration of WLAN access points (APs). In some scenarios, APs belong to different WLANs co-exist on the same channel and compete for radio resources in the same geographic region. To gather traffic information in such network environments, the cognitive wireless networking system should have capability to monitor all wireless channels in a spatio-temporal fashion.

We rely on artificial neural network (ANN) framework for traffic prediction across all 11 channels because ANNs can model the complex relationship between multiple inputs and the output in a way similar to biological neural networks. An important aspect of ANN-based traffic prediction framework is the time scales of prediction. That is, the duration of input (past traffic history) to the projected duration. In this work, we design traffic prediction frameworks that involve predicting future traffic at three different time scales (*i.e.,* Hourly, Five minute wise, and Minute wise). Note here that in a cognitive radio system, the time scales vary in microseconds or milliseconds whereas in cognitive networks, time scales of operation can be longer in duration. We can see that dynamic channel selection schemes based on traffic prediction frameworks at different time scales result in varying number of channel switches per time interval. Each channel switch temporarily disconnects wireless clients from the network and, therefore, can disrupt on-going end-to-end transport sessions. In this work, we will show the impact of the channel selection strategies on the TCP throughput.

The rest of this paper is organized as follows: Section II briefs the related work in this area, Section III presents the architecture of Cognitive AP, Section IV discusses various traffic prediction schemes, and Section V contains prediction accuracy results. Section VI has prototype implementation details and the effect of CogAP's channel selection on transport layer performance. Finally, Section VII contains concluding remarks.

II. Related Work

In [2], the authors compared the performance of linear regression models with that of neural network models for the purpose of building models of the performance in mobile ad hoc wireless networks as a function of external factors such as traffic load and configurable parameters such as the routing protocol being used; the authors concluded that neural network models are the best modeling choice for their scenario. In [3], we designed three different ANN-based hourly traffic prediction schemes by slightly varying inputs of traffic predictors and studied their performance in a campus WLAN. In [4], we conducted early studies on a set of ANN-based traffic prediction frameworks for predicting network traffic at different time scales and studied their performance in terms of Mean Squared Error (MSE), Relative prediction Error (RE), Channel Selection Accuracy (CSA), and Regression coefficient (R-value). For that study, we considered traffic from two locations in a campus WLAN environment, which exhibit entirely different traffic patterns.

In comparison to the existing literature, in this paper, we conduct a detailed study on the performance of three different traffic prediction frameworks where each framework associated with a different time scale of operation. We also compare the performance of proposed schemes with autoregressive schemes. Further, we also study the impact of the dynamic channel selection schemes based on these traffic prediction mechanisms on the transport layer performance.

III. CogAP Architecture

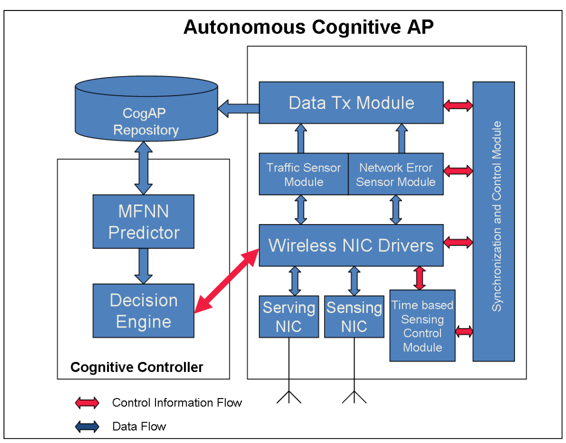


Fig. 1: Architecture of the autonomous Cognitive AP.

Figure 1 shows the schematic diagram of autonomous Cognitive AP (CogAP), which contains two main modules: a sensing & serving module and a cognitive controller module.

A. Sensing and Serving Module

The sensing & serving module helps CogAP to obtain sensorial information about the surrounding communication environment. This module uses two wireless NICs, one for monitoring the traffic related information on each of the 802.11 channels in the 2.4 GHz spectrum, and the other for communication and serving wireless clients associated with the AP. In [5], we suggested packet sampling with single wireless interface card which rotates a radio's operating channel in a round-robin fashion for traffic monitoring in cost effective manner. We found that Systematic Timer-driven Time-based (STT) packet sampling strategy with a sampling period of 11 seconds and a sampling duration of one second allows accurate traffic collection across all 11 channels in 802.11 b/g based WLANs. We, therefore, employ the STT sampling strategy in the sensing interface for collecting sampled traffic traces in a real campus WLAN environment in this work.

B. Cognitive Controller Module

This module is designed to identify the state of the network and the impact of different configuration settings on the performance. It has two sub-modules: prediction and decision engine.

Prediction: CogAP is expected to be able to predict traffic for all the channels based on the historical traffic samples collected from the environment. In this work, we implement five prediction schemes for this prediction block. Three of them are ANN-based and the other two are based on traditional autoregressive models (ARIMA and FARIMA). These prediction schemes are explained in Section IV.

Decision Making: Once the CogAP has estimated the dependency between the network state and the performance with respect to different network settings, it is provided with suitable means of predicting the evolution of the environment across all channels. Based on these predictions, CogAP needs to take its decisions, *i.e.,* to select the most desirable network configuration.

IV. Traffic Prediction Schemes

In this section, we present ANN based prediction models and autoregressive models for IEEE 802.11 b/g based WLANs.

A. ANN Based Models

We employ Multi-layer Feedforward Neural Networks (MFNNs) for designing traffic prediction models as multiple layers of neurons with non-linear transfer functions allow us to learn the linear and non-linear relationships between inputs and outputs. Specifically, we used two-layer feedforward back-propagation networks with one hidden layer and one output layer. We then constructed three kinds of WLAN traffic prediction schemes, namely MLP (Minute Level Prediction), MILP (Minute Interval Level Prediction), and HLP (Hourly Level Prediction) schemes. In MLP (HLP) scheme, we use inputs to predict mean value of future traffic load[1] over next one minute (hour) interval. However, in the case of MILP, we aggregate input traffic in every five minutes to be a 5-minute interval and predict traffic load over next 5-minute interval. We used the following parameters as the inputs/outputs of ANN models.

- *Channel*: ranging from 1 to 11 for 802.11b/g WLANs.
- *DayOfWeek*: ranging from 1 (Monday) to 7 (Sunday).
- *HourOfDay*: ranging from 1 to 24.
- *Traffic(t-i)*: average traffic observed during interval $(t-i-1, t-i]$, $i \geq 0$, measured in *Kbps*.
- *Traffic(t+j)*: average traffic over next interval $(t, t+j]$, $j \geq 1$, measured in *Kbps*.

[1]Traffic load is defined as the ratio of the sum of sizes (in Kilo bits) of packets exchanged in the network over a time interval to the number of time units (in seconds) in that time interval.

We design the following ANN based traffic prediction models: MLP(3,1), MILP(3,1), and HLP(3,1). All models use parameters *Channel, DayOfWeek, and HourOfDay* as their inputs in addition to traffic related parameters. For example, in the case of MLP(3,1) scheme, we use traffic from previous 3 minutes as the 3 one-minute traffic inputs to the ANN model for predicting the next one minute's traffic as the ANN output. For MILP(3,1), we use previous 15 minutes traffic as the 3 five-minute traffic inputs to the ANN model. We made use of MATLAB neural network toolbox to implement traffic prediction schemes and study their accuracy on traffic traces collected in a real campus WLAN environment. In our experiments, the training is done by using Levenberg-Marquardt algorithm [6] and the maximum number of epochs is set at 100. We consider the prediction of traffic only for the orthogonal channels, 1, 6, and 11 because only those channels contained significant traffic in our campus WLAN environment as can be seen from Figure 2.

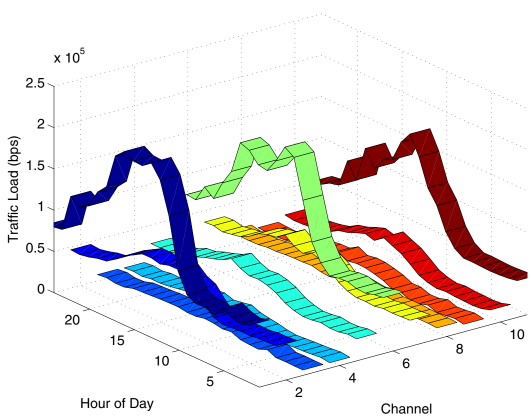


Fig. 2: Working days: Traffic Load vs Channel vs Time.

B. Auto Regressive Models

Recent research has shown that high speed network traffic has self-similarity property. Due to the self-similarity, a specified correlation structure can be found over a wide range of time scales. Self-similarity, also known as long-range dependence, is characterized by a hyperbolically decaying autocorrelation function as the lag increases. This long-range dependent property is quite different from traditional short-range dependent models, of which the autocorrelation function decays exponentially. Typically, we use Auto Regressive Integrated Moving Average (ARIMA) model to represent short-range dependent processes and Fractional ARIMA (FARIMA) model to represent long-range dependent processes. In this work, we design ARIMA and FARIMA based models to predict network traffic and compare them with ANN based models.

We employ Box-Jenkins methodology [7] in order to find an appropriate ARIMA model for a given network traffic trace. For model identification, the first step is to make the traffic trace stationary, which implies the window mean and standard deviation are same as the overall traffic dataset mean and standard deviation. In order to achieve stationarity, we may need to calculate the difference of the traffic trace several times and typically we use d to denote the number of differencing required. At the second step, with a stationary dataset, the autocorrelation function (ACF) and partial autocorrelation function (PCF) are helpful to identify the order of the model. Typically, for an MA process of order q the ACF decays to zero after the lag q, and for an AR process of order p the PCF goes to zero after the lag p. We can model the traffic dataset as an ARIMA(p, d, q) model by observing the ACF and PCF functions. Finally, we calculate the estimate of the traffic value $X(t)$ using the following equation:

$$A(Z) * (1 - Z)^d * X(t) = B(Z) * W(t) \tag{1}$$

$$A(Z) = 1 - a(1) * Z - a(2) * Z^2 - a(3) * Z^3 -, \ldots, -a(p) * Z^p \tag{2}$$

$$B(Z) = 1 + b(1) * Z + b(2) * Z^2 + b(3) * Z^3 +, \ldots, +b(q) * Z^q \tag{3}$$

Note that $A(Z)$ and $B(Z)$ are polynomials of order p and q, respectively. The coefficients $a(1), a(2), \cdots, a(p)$ and $b(1), b(2), \cdots, b(q)$ can be calculated using Yule-Walker equation. The Z is the backward-shift operator, *i.e.,* $X(t) * Z^p = X(t-p)$. The $(1 - Z)^d$ means taking the difference of the dataset d times and $W(t)$ is a white noise process. We predict network traffic value X(t) as a linear combination of previous traffic values and white noise.

FARIMA prediction is very similar to ARIMA prediction. It also obeys Equation 1, but the order of differencing, d, is fractional now. Typically, this value d is between (-0.5, 0.5). In this case, we get:

$$(1 - Z)^d = 1 + C(1) * (-Z) + C(2) * (-Z)^2 + C(3) * (-Z)^3 + \cdots \tag{4}$$

where the coefficients $C(k)$ can be calculated in a recursive manner: $C(0) = 1$ and $C(k+1) = (k+d) * C(k)/(k+1)$.

V. Performance Results

The following metrics are used to evaluate prediction accuracy of Traffic predictors.

Regression Coefficient, also called as R-value, is a measure of how well the variation in the predicted values is explained by the actual traffic values. If this value is equal to 1, then there is perfect correlation between predicted and actual values.

Relative Error (RE) is a measure of the proportion that the predicted traffic drifts from the actual traffic.

Channel Selection Accuracy (CSA) The cognitive controller of CogAP chooses the channel with the lowest predicted traffic value among all channels and compares that with the channel with the lowest actual traffic *a posteriori*. CSA is the percentage of right channel selection decision that our predictor makes.

After defining the input and output of various prediction schemes from one week traffic traces present in cognitive repository, we perform cross-validation. The cross-validation involves splitting traffic trace randomly into two subsets, 70% for generating ARIMA models and training ANN models and 30% for testing accuracy of these predictor models. We repeat cross-validation process 10 times and report average values of metrics of interest.

Note that for ARIMA and FARIMA model generating, the ACF and PCF information is used. For example, Figure 3 shows the normalized ACF and PCF for a traffic trace. This plot is obtained after differencing the traffic trace once. As we can see, the PCF decays to a very small value at lag 3 (decaying under the solid horizontal line means theoretical PCF is zero) and ACF decays exponentially. Hence we fit this traffic trace

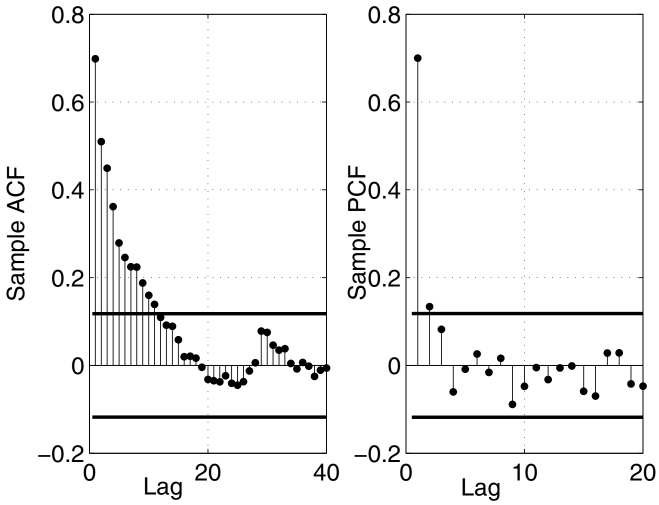


Fig. 3: ARIMA(2,1,0) Model.

into a ARIMA(2,1,0) model. For FARIMA model generation of the same trace, we first obtain an estimate of *d*, which is 0.3 for this specific trace. After differencing, ACF and PCF are shown in Figure 4. In this case, we can model the trace as a FARIMA (2,0.3,0). Note that since this real traffic trace is not a perfect ARIMA or FARIMA process, the ACF is not a perfect exponentially decaying curve, but the difference is allowable in this case.

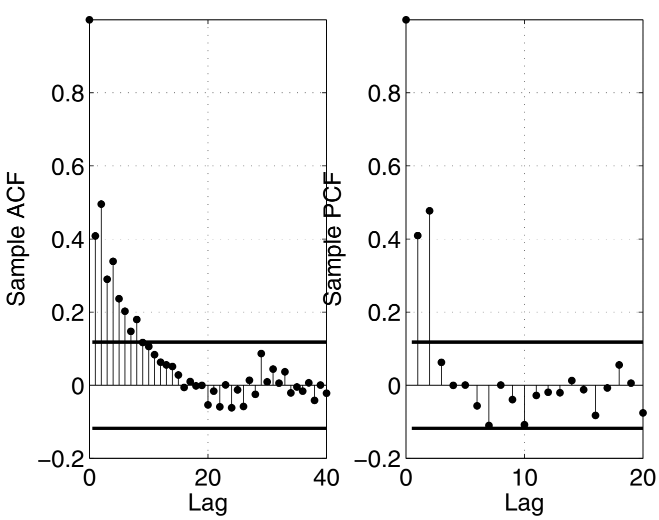


Fig. 4: FARIMA(2,0.3,0) Model.

A. Analysis of Traffic Prediction Schemes

Figure 5a shows traffic prediction performance of MLP and corresponding ARIMA and FARIMA schemes in terms of RE, CSA, and R-values. All the predictors are fed with the same traffic dataset for fair comparison. We can see from figure that ANN based MLP scheme outperforms the other two schemes. We also compared MILP and HLP schemes with corresponding ARIMA and FARIMA schemes in Figures 5b and 5c, respectively. Again, ANN models outperform ARIMA based models. Hence, we can conclude that ANN based prediction schemes have better prediction accuracy than traditional ARIMA and FARIMA prediction schemes. The reason for that is the ability of ANNs to model complex non-linear relationship between inputs and outputs. We can also see that FARIMA model performs better than ARIMA model, this indicates that FARIMA model is more suitable than ARIMA model to predict wireless network traffic which has self-similarity. The performance of MILP(3,1) is about the same as that of MLP(3,1), while HLP(3,1) has poor performance than MLP(3,1) and MILP(3,1) schemes. This is because for any given trace dataset the total number of training samples is much smaller compared to other schemes. Further, traffic related input and output parameters are separated by one hour intervals for HLP(3,1) and, therefore, these samples do not have strong correlation.

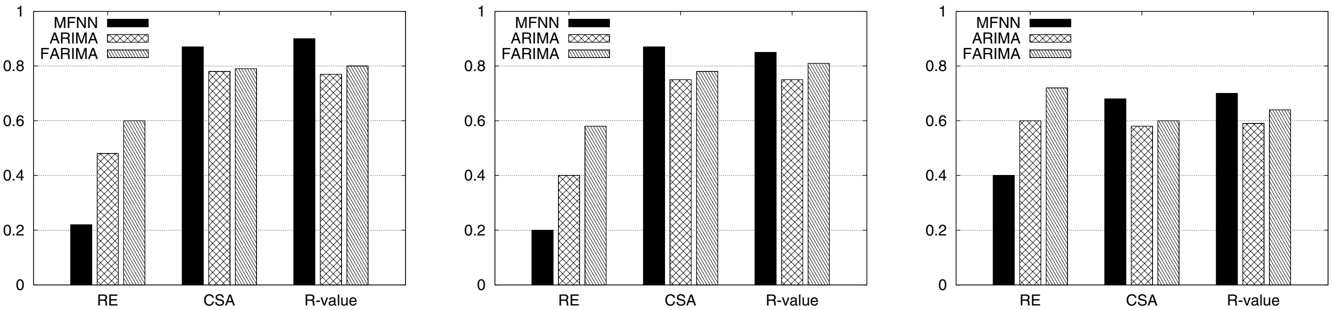
VI. Cognitive AP Prototype

We now present a prototype implementation of CogAP system described in the previous sections. Hardware components used in CogAP include an ALIX2C2 embedded system board, two Atheros chipset based 802.11 b/g miniPCI cards (one for sensing and another for serving users), 8 GB Compact Flash (CF) card (for storing OS and database repository), and two omni-directional pigtail antennas. ALIX 2C2 board, from PC Engines Inc., has 500 MHz AMD Geode processor, two miniPCI slots, and one Ethernet port which make it a perfect choice for CogAP prototype development. The following software base is used for developing cognitive function modules of CogAP prototype: Linux Voyage 0.5.2, Fast ANN (FANN) library [8], MySQL, MadWiFi driver, *tcpdump* and PHP. Cognitive controller is implemented in C programming using FANN library. C-MySQL API interface is used to query MySQL server from C environment. We installed FANN library in the embedded CogAP device in order to implement MFNN predictions as part of CogAP.

A. TCP performance with Cognitive APs

The experimental setup consists of one Linux-based laptop acting as a wireless client and the CogAP device. We use *iperf* software to monitor available bandwidth on the predicted best channel. We exclude self traffic of CogAP network (all the traffic generated by the CogAP and its client during *iperf* runs) for making a fair comparison.

Figure 6 shows that there is a short period of service disruption when CogAP switches its serving channel. That is, throughput drops to zero for a period of 1-3 seconds when client is re-establishing wireless connection to the CogAP on the newly chosen channel. This short service break penalty is mainly due to the processing delays contributed by the driver software associated with radio interface and the delay due to channel scan process. In this work, we did not attempt to improve or reduce the service break delay, however, we strongly believe that the connection disruption delay can be minimized by making the CogAP convey the best channel information to the clients in advance of its actual switching. It can also be noticed in Figure 6, that after the switching



(a) Minute Level Prediction (b) Minute Interval Level Prediction (c) Hour Level Prediction

Fig. 5: Comparison of Prediction Schemes: **(a)** MLP(3,1) vs ARIMA(2,1,0) vs FARIMA(2,0.3,0) **(b)** MILP(3,1) vs ARIMA(1,1,1) vs FARIMA(2,0.4,0) **(c)** HLP(3,1) vs ARIMA(1,1,1) vs FARIMA(2,0.3,0).

process is completed, the client achieves a higher and more stable throughput. Further, we noticed that Wi-Fi cards running on different driver softwares and operating systems result in different kinds of disruptions where in some cases it resulted in broken TCP connections and required to manually reconfigure client's radio interface to the new channel.

TCP throughputs of ANN based prediction schemes are compared in Figure 7. We can observe that MILP and MLP result in higher throughput than HLP. This is because these schemes have better prediction accuracy as seen in Figure 5. It also shows that regardless of which prediction scheme is implemented, the use of any prediction scheme is better than no prediction scheme where AP always serves on a fixed channel.

VII. Conclusions

In this paper, we showed that neural network based traffic predictors exhibit higher prediction accuracy compared with traditional ARIMA and FARIMA based predictors. We also showed that neural network based predictors can indeed help the CogAP to find the best channel which gives higher and more sustained bandwidth. Although we have seen a drastic drop of the TCP throughput during the switching operation of CogAP, which takes about 1 to 3 seconds, the throughput achieved after switching was higher compared to a system that does not use cognitive channel switching. Further, the overall throughput increases for cognitive wireless networks using the dynamic channel switching mechanisms based on traffic prediction schemes.

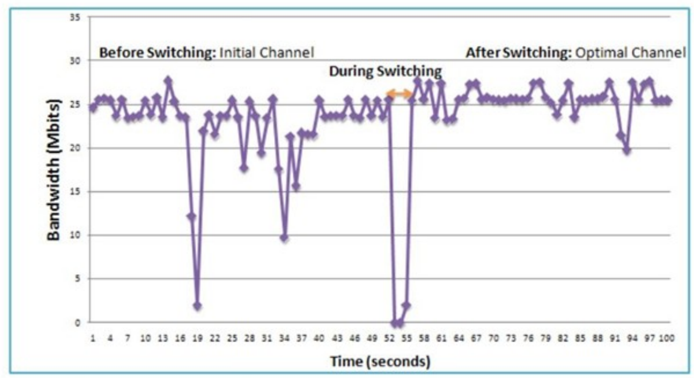


Fig. 6: TCP throughput with dynamic channel switching.

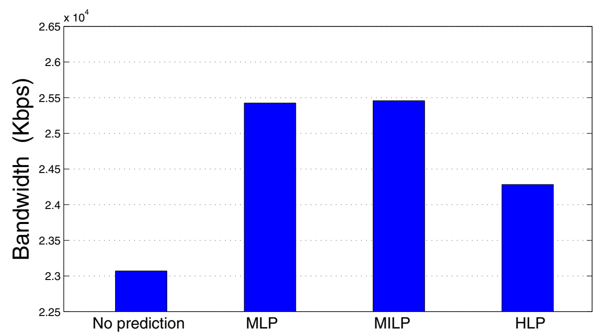


Fig. 7: Throughput for different prediction schemes.

Acknowledgment

This work was partially supported by U.S. Army Research Office (grant no. W911NF-09-1-0456), UCSD-CWC (Center for Wireless Communications), and UC-Discovery grants.

References

[1] R. W. Thomas, D. H. Friend, L. A. DaSilva, and A. B. MacKenzie, "Cognitive networks: Adaptation and learning to achieve end-to-end performance objectives," *IEEE Communications Magazine*, vol. 44, no. 12, pp. 51–57, December 2006.

[2] A. Moursy, I. Ajbar, D. Perkins, and M. Bayoumi, "Building empirical models of mobile ad hoc networks," in *Proc. of the International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS)*, July 2007.

[3] B. R. Tamma, B. S. Manoj, and R. Rao, "An autonomous cognitive access point for wi-fi hotspots," in *Proc. of IEEE GLOBECOM*, December 2009.

[4] Y. Liu, B. R. Tamma, B. S. Manoj, and R. R. Rao, "Traffic prediction for cognitive networking in multi-channel wireless networks," in *Proc. of IEEE Infocom Workshop on Cognitive Wireless Communications and Networking*, March 2010.

[5] B. R. Tamma, N. Baldo, B. S. Manoj, and R. Rao, "Multi-channel wireless traffic sensing and characterization for cognitive networking," in *Proc. of IEEE ICC*, June 2009.

[6] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *Journal of the Society for Industrial and Applied Mathematics*, vol. 11, no. 2, pp. 431–441, June 1963.

[7] G. Box, G. Jenkins, and G. Reinsel, *Time Series Analysis: Forecasting and Control*, 3rd ed. Prentice Hall, 1994.

[8] http://leenissen.dk/fann/.