

# A Predictive Network Resource Allocation Technique for Cognitive Wireless Networks

Alia Asheralieva, Jamil Y. Khan, Kaushik Mahata and Eng Hwee Ong

School of Electrical Engineering and Computer Science

The University of Newcastle

Callaghan, NSW, 2308, AUSTRALIA

{Jamil.Khan, [Kaushik.Mahata](mailto:Kaushik.Mahata@newcastle.edu.au)}@newcastle.edu.au, {enghwee.ong, alia.asheralieva}@uon.edu.au

**Abstract**—Future wireless networks are evolving towards a heterogeneous cooperative and cognitive architecture to support broadband communication needs of different types of traffic. Future wireless networks will allocate network resources based on cooperative techniques. Resource controllers will apply the cognitive principle to find out status of various networks. We propose a predictive resource allocation strategy where we employ adaptive algorithms to predict the network loading. These algorithms can detect changes in the traffic characteristics, and adapt automatically. To validate our claim, we present the results of applying these adaptive algorithms on real-world network traffic traces.

**Keywords**—Cognitive Networks, Traffic Prediction, Resource Allocation.

## I. INTRODUCTION

With widespread use of wireless networks and the emergence of multiple deployed wireless standards the wireless network design paradigm is changing rapidly [1]. In future wireless services will be offered through heterogeneous networks rather than using single standard wireless networks [2, 3]. Also, the emergence of software defined radio (SDR) will allow customers of any subscribed network to connect to any other network when either the QoS (Quality of Service) or the capacity of its subscribed network falls below a certain threshold value [2]. In future radio access networks (RAN) will be IP (Internet Protocol) based network which are expected to host many wireless access points and base stations. Figure 1 shows a typical heterogeneous wireless network where number of access networks are connected to a single RAN. Various access points and base stations will connect to the spectrum manager using an IP network. In this network architecture using SDR terminals mobile users can connect to any of the access points or base stations within their coverage area. The concept of the dynamic network selection algorithm using a network architecture known as TONA (Terminal Oriented Network Assisted) has been proposed and elaborated in reference [4]. The TONA architecture allows terminals to operate in the cooperative domain to improve the overall capacity of networks and QoS of user connections. In this work we propose an IP based heterogeneous network architecture where network resources are shared between a primary (licensed) and a secondary (unlicensed) network using the cognitive

networking concept and the TONA architecture. The TONA architecture allows an IP based controller to exchange resource allocation information among its member networks. In a traditional cognitive network a channel sensing mechanism is employed by a secondary network to look for *unused spectrum* which can be borrowed by the secondary network [5]. In this paper we are proposing a layer MAC (Medium Access Control) based predictive resource allocation technique for a cognitive network where resources can be shared using an IP based heterogeneous network. Since various applications generate different types of traffic it is necessary for traffic controllers to deal with variety of traffic characteristics. The prediction algorithms must adapt to range of traffic patterns predict various traffic flows with good accuracy. To cope with this issue we use adaptive tracking algorithms to track different traffic prediction models. In particular, we consider autoregressive (AR), moving average (MA) and autoregressive-moving average (ARMA) models in this paper. The algorithms have been tested on real-world network traffic traces to validate our claims.

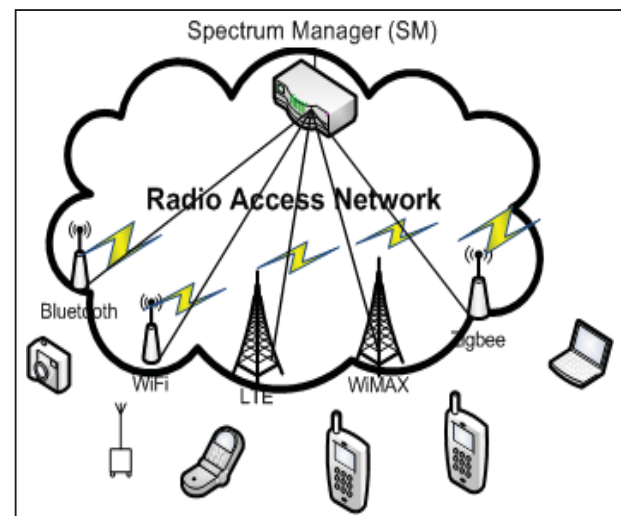


Figure 1. A cognitive and cooperative network with a single RAN, and multiple access points and base stations

The organization of the paper is as follows. Section 2 describes a cognitive and cooperative network architecture.

Section 3 presents the predictive resource allocation technique for cognitive networks. Section 4 presents short and medium term prediction techniques that can be used for the resource allocation in cognitive networks. Conclusions are drawn in section 6.

## II. COOPERATIVE AND COGNITIVE NETWORK ARCHITECTURE

In a heterogeneous network as shown in figure 1 different networking entities will collaborate with each other via the IP controller shown as the spectrum manager (SM) to offer best connectivity to users as well as to maximize utilization of their own network capacity. The cognitive networking concept has been developed with a view that each secondary (unlicensed) network on its own will find available unused capacity from other primary (licensed) network and utilize them as required. We propose a different scenario for cognitive networks where all the networks will cooperate by exchanging their network status information and share available spare capacity in an orderly manner. The main justifications for using the cooperative and cognitive networking (CCN) architecture are listed below.

- A cooperative network will allow networking entities to share aggregate network resources by the user population using SDR terminals in different geo-clusters. Users whose QoS requirements are not met by the existing network can handover to a different network [2].
- A cooperative approach will allow networks to temporarily borrow network resources from primary networks in a more efficient manner. In a physical layer spectrum sensing approach a secondary spectrum could sense free spectrum from other networks and could use the spectrum without knowing the future requirements of the primary network. This approach could lead to QoS degradation in both networks.
- An IP based resource allocation technique will eliminate the need for continuous physical layer channel sensing which could lead to increased power consumption of mobile terminals.

In contrast to the conventional cognitive network architecture we propose a CCN architecture where each primary network broadcast resources information to allocate additional resources in secondary networks. The spectrum manager shown in figure 1 will collect information from all primary networks and distribute that information to secondary networks that are looking for additional resources. In this model each access points and base stations will send their channel occupancy level and current traffic volume to the spectrum manager. The spectrum manager will use that information and predict channel availability probability for its each member network. The spectrum manager will also have a repository of requests from different secondary networks. Secondary networks will predict their future traffic requirements and generate their

resource requirement requests. Network resources can be allocated in different ways. Network resources can be allocated by the SM using following resource allocation techniques; scheduled, contention and auction based mechanisms.

Using a scheduled allocation technique the SM will match the resource requests with the available spectrum and send a resource offer to a secondary network based on the best match, request arrival time and the priority of the request. The offered resource will be allocated for a fixed duration based on the prediction result. Each network resource will have a finite lifetime which is calculated using a prediction technique. If the secondary network accepts the request then the SM will remove the resource from the available resource list and inform the primary network about the allocation. In case of a contention based allocation, the SM advertises available resources and its parameters to all secondary networks. A number of secondary networks sends requests to the SM using a contention mode. The SM will allocate the resource to the first successfully received request. In an auction based approach each channel is advertised with a minimum price and then kept resource open for certain duration so that secondary networks can bid for the resource. The highest bidder from the list is allocated the resource. The CCN architecture avoids the need for continuous channel sensing process however; the architecture requires accurate prediction process to generate channel availability information.

## III. PREDICTIVE RESOURCE ALLOCATION TECHNIQUES

Accurate traffic prediction is crucial for a CCN which would be an important part of radio resource management system for future networks. When radio resource occupancy is predicted accurately for each wireless networks in a CCN, users can select the optimal channels, which will help to reduce probability of connection loss for users both in primary and secondary networks. This approach could increase the overall network resource utilization in a heterogeneous network.

Because of its heterogeneous structure, the CCN will consist of different wireless networking standards with different channel conditions, and user distributions, whose traffic demand and behavior could vary quite widely. Traffic modeling and prediction in such a network is very complicated, and requires a long observation of history. Besides, the stochastic parameters could vary very slowly, and short-term forecasting in this case might not perform well. For more accurate prediction, the parameters should be estimated in “*real time*” to track time-varying traffic characteristics. These algorithms should have low dependency on priory data, and modest requirements on the memory. These features are common for recursive identification methods which can be applied together with non-realtime identification methods to make more accurate parameter estimation. Moreover, taking into account the traffic pattern and network conditions of nodes in a CCN which could be very different, it may be necessary to use several traffic models to make accurate prediction.

In the following sections we describe recursive estimation techniques and models that could be deployed for forecasting network resources in a CCN. We also present some results of 1- and 10-step-ahead prediction for different traffic patterns, and discuss their performance for different traffic conditions.

#### A. Traffic Model

To generate predictions, information about past events, time-series data, is collected. According to definition, time series is a sequence of observations  $y(t), y(t-1), \dots, y(1)$  of a random process  $Y$  at discrete time intervals, where an observation at time  $t$  is given by  $y(t)$ . Many time series models, such as AR( $n_a$ ) (autoregressive), MA( $n_c$ ) (moving average), ARMA( $n_a, n_c$ ) (autoregressive moving average) have been proposed for time-series analysis [6]. These methods use the following model [7]:

$$y(t) = \varphi^T(t)\theta + e(t), \quad (1)$$

where  $\theta$  is the system parameter vector, and the regressor  $\varphi(t)$  depends on the past data and the model structure. For the AR model:

$$\varphi(t) = (-y(t-1) \dots -y(t-n_a))^T, \theta = (a_1 \dots a_{n_a})^T. \quad (1a)$$

For the MA model:

$$\varphi(t) = (e(t-1) \dots e(t-n_c))^T, \theta = (c_1 \dots c_{n_c})^T. \quad (1b)$$

For the ARMA model:

$$\begin{aligned} \varphi(t) &= (-y(t-1) \dots -y(t-n_a) \quad e(t-1) \dots e(t-n_c))^T, \\ \theta &= (a_1 \dots a_{n_a} \quad c_1 \dots c_{n_c})^T. \end{aligned} \quad (1c)$$

#### B. Parameter Estimation

In the adaptive (real-time) identification methods, the parameter estimate  $\hat{\theta}(t)$  based on data up to time  $t$  is computed in a recursive way by modification of an estimate  $\hat{\theta}(t-1)$ . The advantages of recursive methods can be summarized as follows:

- they are central part of adaptive systems where the filtering action is based on the most recent model;
- they have relatively small (compare to off-line identification methods) requirements on primary memory;
- they can be modified into real-time algorithms to track time-varying parameters;
- they can be deployed for fault detection when the observed system has changed significantly.

There are four recursive methods (recursive least squares (RLS), recursive instrumental variable (RIV), pseudo-linear

regression (PLR), and recursive prediction error (RPEM)) used for parameter estimation of different models. Two of them are PLR and RPEM, which can be applied to track parameters of AR, MA, and ARMA models. The following general algorithm is used:

$$P(t) = \frac{1}{\lambda} \left[ P(t-1) - \frac{P(t-1)\psi(t)\psi^T(t)P(t-1)}{\lambda + \psi^T(t)P(t-1)\psi(t)} \right], \quad (2a)$$

$$K(t) = P(t)\psi(t), \quad (2b)$$

$$\varepsilon(t) = y(t) - \varphi^T(t)\hat{\theta}(t-1), \quad (2c)$$

$$\hat{\theta}(t) = \hat{\theta}(t-1) + K(t)\varepsilon(t). \quad (2d)$$

Here  $\lambda$  is a forgetting factor to discount the measurements obtained previously; the smaller is the value of  $\lambda$ , the faster information is forgotten (usually  $\lambda = 0.95-1$ );

$$P(t) = \bar{R}^{-1}(t), \quad \bar{R}(t) = tR(t), \quad (2e)$$

$R(t)$  - Hessian approximation in Gauss-Newton algorithm;  
 $K(t)$  - the gain vector showing how much the value of  $\varepsilon(t)$  will modify the different elements of  $\theta$ ;

$\varepsilon(t) = y(t) - \hat{y}(t)$  - Prediction error of estimation;

$\psi(t) = (-y^F(t-1) \dots -y^F(t-n_a) \quad \varepsilon^F(t-1) \dots \varepsilon^F(t-n_c))^T$  - negative gradient of  $\varepsilon(t)$  with respect to  $\theta$ ;

$y^F(t), \varepsilon^F(t)$  - filtered data.

For the RPEM:

$$\begin{aligned} y^F(t) &= y(t) - \hat{c}_1(t)y^F(t-1) - \dots - \hat{c}_{n_a}(t)y^F(t-n_a), \\ \varepsilon^F(t) &= \varepsilon(t) - \hat{c}_1(t)\varepsilon^F(t-1) - \dots - \hat{c}_{n_c}(t)\varepsilon^F(t-n_c). \end{aligned} \quad (2f)$$

For the PLR:

$$y^F(t) = y(t), \quad \varepsilon^F(t) = \varepsilon(t), \quad (2g)$$

i.e. filtering of RPEM is neglected [7].

The effect of initial values on performance of recursion was widely discussed in literature [7, 8]. Without any priori information it is common practice to set:

$$\hat{\theta}(0) = 0, \quad P(0) = \rho I, \quad (2h)$$

where  $\rho$  is a “big” number. Usually  $P$  is set so that ( $t_0 = 10-25$ ):

$$P^{-1}(0) \ll \sum_{s=1}^{t_0} \varphi(s)\varphi^T(s). \quad (2i)$$

#### IV. PERFORMANCE ANALYSIS OF PREDICTION TECHNIQUES

To observe the performance of time-series models in different environments, we use one theoretical (Poisson packet arrival process with constant and varying mean) and two experimental traffic traces (data packages LBL-Conn-7 and DEC-Pkt1).

The first trace, LBL-Conn-7, contains the TCP (Transmission Control Protocol) traffic data between the Lawrence Berkeley Laboratory and the rest of the world in the format where timestamps have microsecond precision. After processing the trace for uplink connections only we get another trace where overall uplink data rate is calculated for each microsecond. Recursion starts immediately with the observation data. Figures 2 and 3 show 50-ms-long for 1-step-ahead and 500-ms-long for 10-step-ahead PLR prediction respectively using AR(1), MA(1), and ARMA(1,1) models. We use normalized mean squared error NMSE, as shown in equation (3) to evaluate performance of the prediction after a certain period of recursions. The prediction error ratio PER shown in equation (4) is used to measure the accuracy of real time prediction during recursion. The NMSE values for 1- and 10-step-ahead prediction obtained after 3000 recursion are given in Table 1.

$$NMSE = \frac{\sum_{t=1}^N (y(t) - \hat{y}(t))^2}{\sum_{t=1}^N (y(t) - \bar{y}(t))^2}, \quad (3)$$

$$PER = \left| \frac{y(t) - \hat{y}(t)}{y(t)} \right| \cdot 100\%. \quad (4)$$

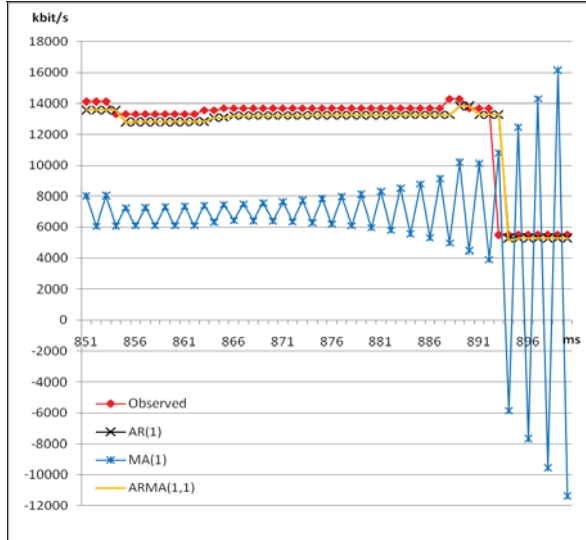


Figure 2. 50-ms-long observation and recursive 1-step-ahead PLR prediction using AR(1), MA(1), and ARMA(1,1) models (LBL-Conn-7)

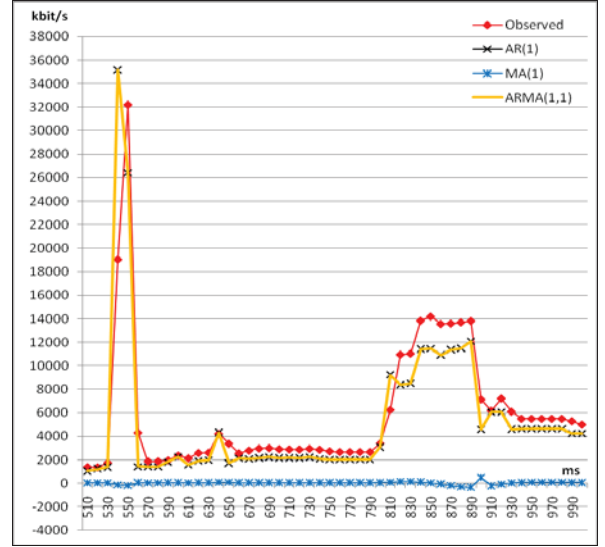


Figure 3. 500-ms-long observation and recursive 10-step-ahead PLR prediction using AR(1), MA(1), and ARMA(1,1) models (LBL-Conn-7)

TABLE I. VALUES OF NMSE FOR DIFFERENT TRACES WITH PLR

Trace	Model	NMSE	
		1-step-ahead prediction	10-step-ahead prediction
LBL-Conn-7	AR(1)	0.071397	0.151657
	MA(1)	0.637899	1.225446
	ARMA(1,1)	0.072005	0.146376
	ARIMA(0,1,1)	0.000741	-
	ARIMA(1,1,0)	0.000742	-
	ARIMA(1,1,1)	0.002977	-
DEC-Pkt1	AR(1)	$4.81302 \cdot 10^{-7}$	$4.60807 \cdot 10^{-6}$
	MA(1)	1.742629	3.44529
	ARMA(1,1)	$4.89 \cdot 10^{-7}$	$1.75 \cdot 10^{-6}$
	ARIMA(0,1,1)	$4.81302 \cdot 10^{-7}$	-
	ARIMA(1,1,0)	$6.3 \cdot 10^{-8}$	-
	ARIMA(1,1,1)	$8.16 \cdot 10^{-8}$	-
Poisson	AR(1)	1.578521	-
	MA(1)	3.576409	-
	ARMA(1,1)	1.035006	-
	ARIMA(0,1,1)	0.68656	-
	ARIMA(1,1,0)	0.471443	-
	ARIMA(1,1,1)	0.610663	-

Both short term and medium term results show that best prediction performance is obtained using the AR(1) model. Results also show that the MA model is unstable and fail to predict traffic value accurately. Thus, we select from the family of  $AR(n_a)$  models using the appropriate order of  $p$  by minimizing the Akaike Information Criterion (AIC) shown in equations (5) [9], and get the optimal value for  $n_a = 1$ . Figure 4 presents the values of AIC(N,  $n_a$  after  $N = 3000$  recursions).

$$AIC = N \log V_N(\theta_N) + 2p, \quad (5a)$$

$$V_N(\theta_N) = \frac{1}{N} \sum_{t=1}^N \varepsilon^2(t, \theta). \quad (5b)$$

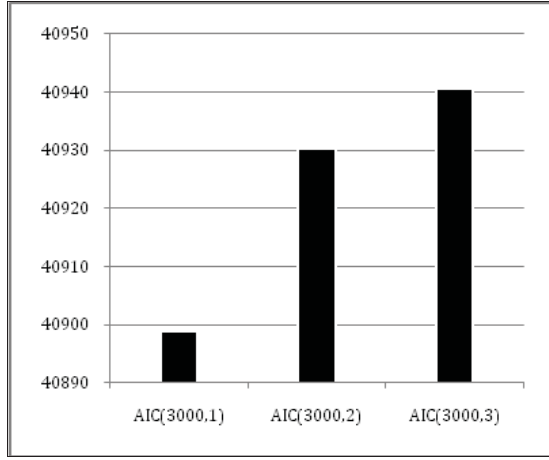


Figure 4. The values of AIC(N, na) after N = 3000 recursions (LBL-Conn-7)

To compare performance of different estimation methods we tracked the same trace using the RPEM for the AR(1) model. Previous research reported that both PLR and RPEM techniques offers consistent performance, but behavior of the PLR in the transient phase might be better than that of the RPEM [7]. Our results confirm this observation – at the beginning of recursion prediction error ratio using the RPEM technique generates larger values than the PLR technique as shown in figures 5, 6. However, after a certain period the RPEM shows better performance than the PLR technique as shown in figures 7, 8. Thus, RPEM takes more time than the PLR to estimate the parameters of the model. In our case the trace convergence delay of PLR is 10-15 recursions and for the RPEM; 50-60 recursions.

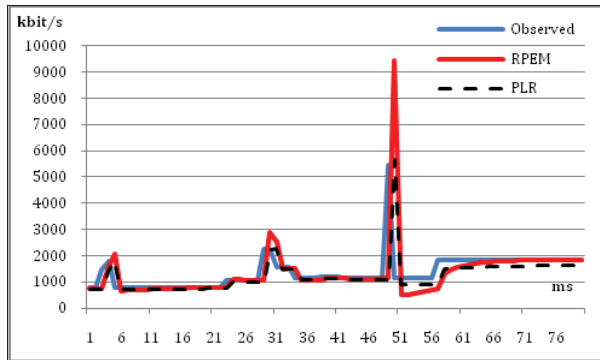


Figure 5. Observed and predicted values for AR(1) model using PLR and RPEM methods at the beginning of recursion (LBL-Conn-7)

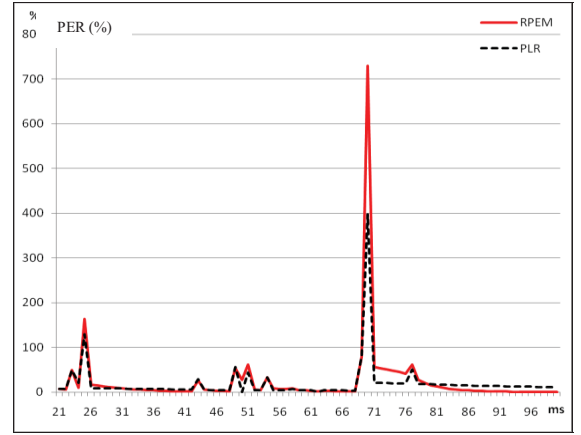


Figure 6. Prediction error ratio for AR(1) model using PLR and RPEM methods at the beginning of recursion (LBL-Conn-7)

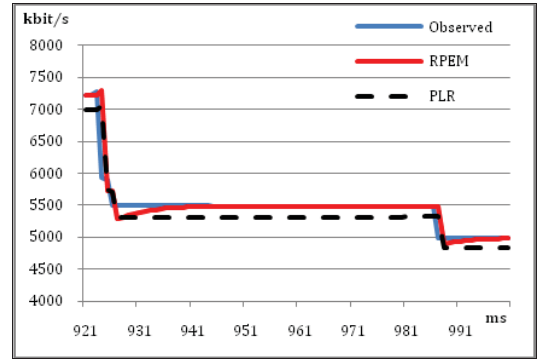


Figure 7. Observed and predicted values for AR(1) model using PLR and RPEM methods after 920 recursions from the beginning (LBL-Conn-7)

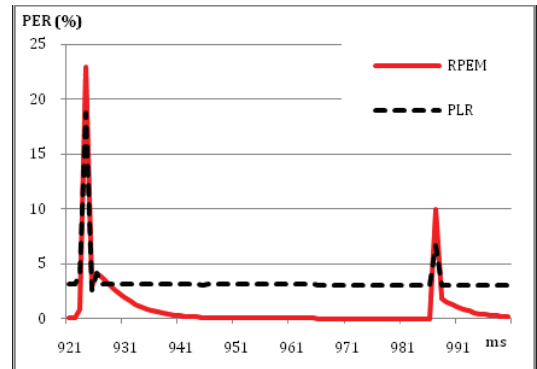


Figure 8. Observed and predicted values for AR(1) model using PLR and RPEM methods after 920 recursions from the beginning (LBL-Conn-7)

The second trace used in this work is the DEC-Pkt1 contains the all wide-area traffic between Digital Equipment Corporation (DEC) and the rest of the world with cumulative traffic volume given for each microsecond.



Similar to the first trace, we made 30000 recursions for AR, MA, and ARMA models. Figures 9 and 10 show the performance for 1- and 10-step ahead PLR predictions; corresponding NMSE values are listed in Table 1.

We can select the model order by applying the famous Akaike Information Criterion [7,9], which yields an optimal value 1 for AR models. We also compare performance of prediction errors using the PLR and the RPEM techniques for the AR(1) model as shown in Figures 11 and 12. Results show that the PLR gives smaller PER values not only at the beginning as shown in figure 11, but also after 20000 recursions as shown in figure 12.

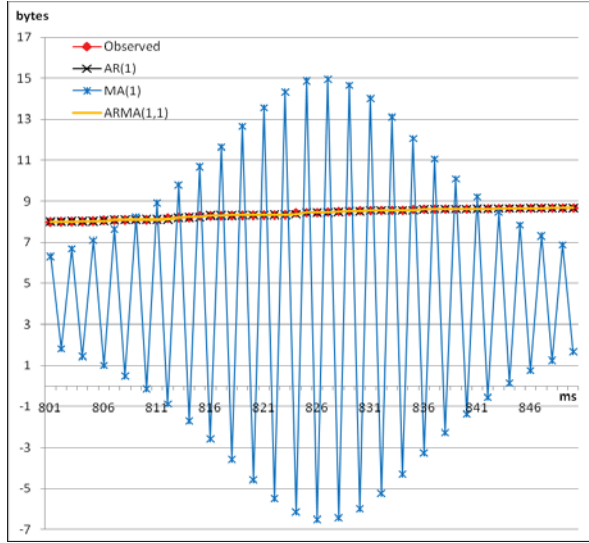


Figure 9. 50-ms-long observation and recursive 1-step-ahead PLR prediction for AR(1), MA(1), and ARMA(1,1) models (DEC-Pkt1)

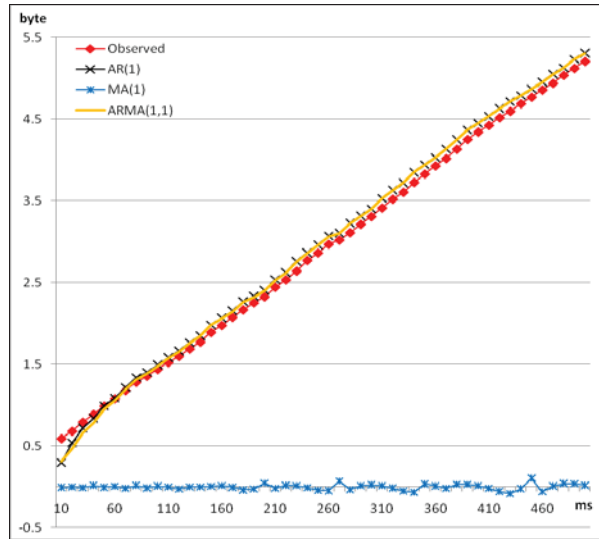


Figure 10. 500-ms-long observation and recursive 10-step-ahead PLR prediction using AR(1), MA(1), and ARMA(1,1) models (DEC-Pkt1)

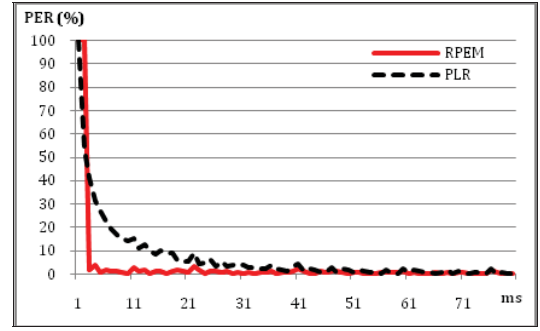


Figure 11. Prediction error ratio for AR(1) model using PLR and RPEM methods at the beginning of recursion (DEC-Pkt1)

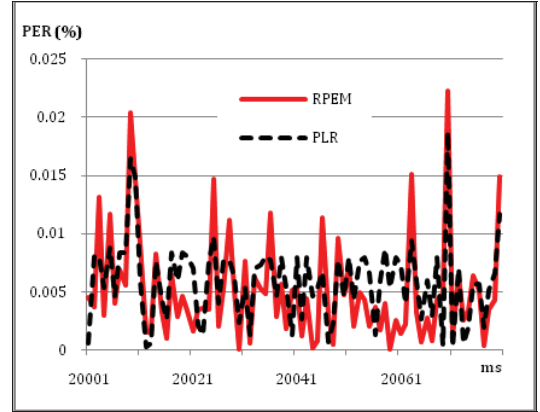


Figure 12. Prediction error ratio for AR(1) model using PLR and RPEM methods after 20000 recursions from the beginning (DEC-Pkt1)

Next trace used in the performance evaluation is a theoretical Poisson generated sequence. Even though the Poisson model has been reported to be unsuitable for Internet traffic modeling [10], Poisson generated traffic is still widely used in communication networks, and can be used as a good example of a process with non-zero mean and highly random (unpredictable) pattern (a Hurst parameter of a Poisson process  $H = 0.5$ ). We generated Poisson sequence, and run 50000 recursions for AR, MA, and ARMA models. It can be seen that Poisson generated sequence has local spikes that cannot be predicted using any of these models as shown in figure 13. Corresponding NMSE values for each model are listed in Table 1.

It has been shown in previous research works that drifting disturbances and non-zero means can be treated by the family of ARIMA (autoregressive integrated moving average) models [11]. An ARIMA model is an ARMA model constrained to have the factor of  $y(t) - y(t-1)$ . Table 1 show that the lowest value of NMSE for Poisson traffic prediction was achieved by the ARMA(1,1) model. However, the AR(1) model is more accurate in predicting peaks in Poisson distributed traffic traces as shown in figure 13. Figure shows the curve of ARMA prediction is

smoother than that of the AR model. Therefore we use the ARIMA(1,1,0) which corresponds to the AR(1), and the ARIMA(0,1,1) model corresponds to ARMA(1,1) models to generate the next prediction. Results confirm that the ARIMA model is more efficient in case of a heavily peaked traffic. Both ARIMA(0,1,1) and ARIMA(1,1,0) models are able to predict the spikes of Poisson process as shown in figures 14 and 15, and Table 1.

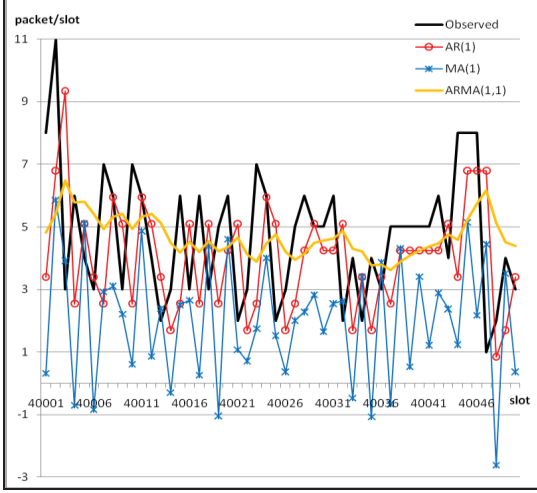


Figure 13. 50-ms-long observation and recursive 1-step-ahead PLR prediction using AR(1), MA(1), and ARMA(1,1) models (Poisson process)

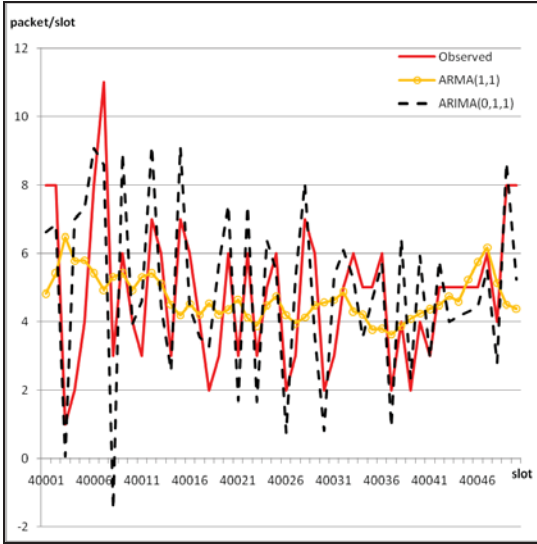


Figure 14. 50-ms-long observation and recursive 1-step-ahead PLR prediction using ARMA(1,1) and corresponding ARIMA(0,1,1) models (Poisson process)

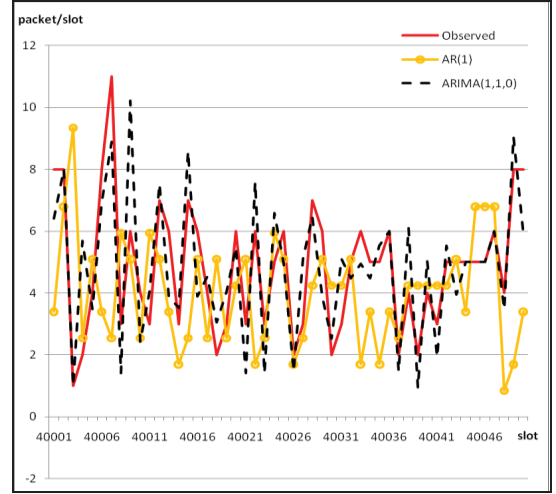


Figure 15. 50-ms-long observation and recursive 1-step-ahead PLR prediction using AR(1) and corresponding ARIMA(1,1,0) models (Poisson process)

In order to observe performance of the ARIMA prediction technique for real traces LBL-Conn-7 and DEC-Pkt-1), we applied it for the traces that we tracked before using AR, MA, and ARMA models LBL-Conn-7 and DEC-Pkt-1). Results show that the ARIMA can predict future traffic load in a network with a higher accuracy as shown in figures 16 and 17, and in Table 1. Based on these observations the ARIMA model can be recommended for a predictive resource allocation system in a CCN.

Following observations can be made using the recursive prediction for different time-series models:

- Both PLR and RPEM techniques can be used for traffic prediction with time-series models. Even though PLR technique does not use filtering applied in the RPEM. The PLR technique converges more quickly than RPEM technique (10-15 recursions for traces that we tracked). Besides that the RPEM is more memory-demanding and complex than the PLR technique. Therefore, PLR seems to be more attractive for CCN applications where prediction should be produced in a very short time.
- Time-series models like AR and ARMA offers quite correct prediction for traffic traces, but fail to predict for random processes like Poisson. In such cases ARIMA model can be applied. On the other hand, ARIMA can be used for real-time traces as well. For our traces, in particular, it increased the accuracy of prediction with AR and ARMA model.

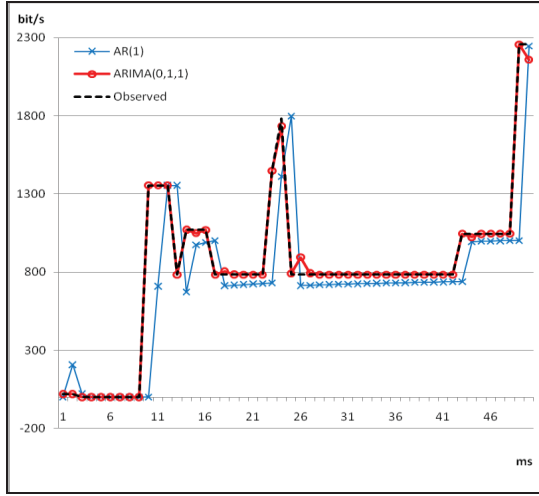


Figure 16. 50-ms-long observation and recursive 1-step-ahead PLR prediction using AR and ARIMA models (LBL-Conn-7)

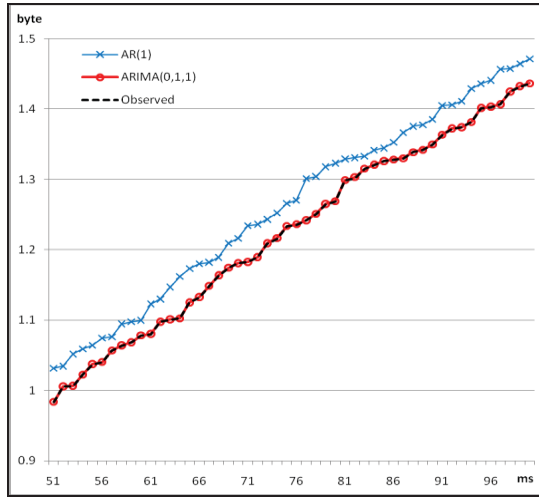


Figure 17. 50-ms-long observation and recursive 1-step-ahead PLR prediction using AR and ARIMA models (DEC-Pkt-1)

In a CCN architecture as mentioned in section 2 that both primary and secondary networks need to employ traffic prediction techniques. It is obvious, that some of the models those have been widely used (for example, ARMA) earlier in networking applications may not be able to predict traffic parameters accurately. Based on the performance analysis we propose to use the ARIMA model for predictive resource allocation technique. Generated prediction can be presented in two ways: (i) as a table with models and their prediction errors as shown in Table 2; (ii) a weighted sum of predicted values, where weighting coefficients should be related to prediction errors as shown in equations (6a, b).

TABLE II. EXAMPLE OF TABLE FOR TRAFFIC PREDICTION

Model	Predicted value	PER
AR(1)	Value <sub>1</sub>	PER <sub>1</sub>
ARMA(1,1)	Value <sub>2</sub>	PER <sub>2</sub>
ARIMA(0,1,1)	Value <sub>3</sub>	PER <sub>3</sub>
ARIMA(1,1,0)	Value <sub>4</sub>	PER <sub>4</sub>

$$\text{Predicted value} = \sum_{i=1}^M w_i \cdot \text{Value}_i, \quad (6a)$$

$$w_i = \frac{1/\text{PER}_i}{\sum_{j=1}^M 1/\text{PER}_j}. \quad (6b)$$

## CONCLUSIONS

In this paper we introduced a new layer MAC/RLC based predictive network resource allocation algorithm. The proposed algorithm will improve the network resource allocation policies in a heterogeneous cooperative and cognitive network. Spectrum sensing based secondary network resource allocation techniques could introduce instability in the network particularly when secondary networks are not aware of their and primary networks future needs. A prediction based algorithm will remove that uncertainty hence improving the network stability. One of the main aims of this paper is to develop suitable traffic prediction techniques for the resource allocation. Through extensive analytical and simulation results we have shown that it is possible to accurately predict network traffic demand in communication networks. Next stage of the work we will further develop the predictive resource allocation algorithm and analyze the performance of the predictive resource allocation algorithm using a discrete event simulation and realistic traffic generators.

## REFERENCES

- [1] R. W. Thomas, D. H. Friend and L. A. DaSilva and A. B. MacKenzie, "Cognitive Networks Adaption and Learning to Achieve End-to-End Performance Objectives", IEEE Communications Magazine, December 2006, pp. 51-57.
- [2] E. H. Ong and J. Y. Khan, "Cooperative Radio Resource Management Framework for Future IP based Multiple Radio Access Technologies Environment", Computer Networks, vol:54, no:7, May 2010, pp. 1083-1107.
- [3] K-C Chen and R. Prasad, Cognitive Radio Networks, John Wiley & Sons, 2009.
- [4] E. H. Ong and J. Y. Khan, "Dynamic Access Network Selection with QoS Parameters Estimation: A Step Closer to ABC", Proc. of the IEEE Vehicular Technology Conference 2008, 11-14 May, Singapore, pp. 2671-2676.
- [5] B. Letaief and W. Zhang, "Cooperative Communications for Cognitive Networks", Proc. of the IEEE, vol:97, issue:5, 2009, pp. 878-893.
- [6] G. Janacek and L. Swift. Time series: forecasting, simulation, applications. Ellis Horwood Limited, 1993.
- [7] T. Soderstrom and P. Stoica. System identification. Prentice Hall International (UK) Ltd, 1989.



- [8] Y. C. Ho. On the stochastic approximation method and optimal filtering theory. *Journal of Mathematical Analysis and Applications*, Vol. 6, pp. 152-154, 1963.
- [9] H. Akaike. Fitting autoregressive models for prediction. *Ann. Inst. Statist. Math.*, Vol. 21, pp. 243-247, 1969.
- [10] V. Paxson and S. Floyd. [Wide-Area Traffic: The Failure of Poisson Modeling](#). *IEEE/ACM Transactions on Networking*, 3(3), pp. 226-244, June 1995.
- [11] G. Box and G. Jenkins. *Time series analysis: forecasting and control*. Holden-Day, San Francisco, 1976.