An Autonomous Cognitive Access Point for Wi-Fi Hotspots

Bheemarjuna Reddy Tamma, B. S. Manoj, and Ramesh Rao
California Institute for Telecommunications and Information Technology – UC San Diego, USA
E-mail: {btamma, bsmanoj, rrao}@ucsd.edu

***Abstract*—In this paper, we present an application of the Cognitive Networking paradigm to the problem of development of autonomous Cognitive Access Point (CogAP) for small scale wireless network environments such as Wi-Fi hotspots and home networks. In these environments we typically use only one AP per service provider/residence for providing wireless services to the users. However, note that larger number of APs from multiple service providers/residences vie for bandwidth in any geographic region. Here we can reduce the cost of autonomic network control by equipping the same AP with a cognitive functionality.**

We first present architecture of our autonomous CogAP. Then we introduce our algorithmic solution, in which a Neural Network-based traffic predictor makes use of historical traffic traces to learn network traffic conditions and predicts traffic loads on each of 802.11 b/g channels. The cognitive decision engine makes use of traffic forecasts to dynamically decide which channel is best for CogAP to operate on for serving its clients. One of the challenges in autonomous cognitive decision making is the computation resource constraints in today's embedded APs. We have built a prototype CogAP device using cognitive software modules and off-the-self hardware components. We carried out performance evaluation of the proposed CogAP system by conducting experimental measurements on our testbed platform; the obtained results show that the proposed CogAP is effective in achieving performance enhancements with respect to state-of-the-art channel selection strategies.

I. Introduction

Wireless LAN (WLAN) deployment is a hard and laborious task even with a moderate number of APs; today, this is commonly carried out by a manual operation with several reconfiguration cycles. The deployment involves selecting locations for placing APs and (for each AP) selecting operating radio channel (802.11*b/g* and *a* based APs have 3 and 13 orthogonal channels, respectively, to choose from), transmit power (which decides the coverage area of AP), data rate, and medium access control (MAC) layer related parameters. However, due to the dynamic and shared nature of wireless medium (shared with APs in the same WLAN, with APs in other WLANs, and with devices that are not APs at all), parameters controlling access to the wireless medium on each AP must be monitored frequently and modified in a coordinated fashion to maximize WLAN performance. Manually monitoring the network traffic and determining an optimum configuration for the parameters related to the wireless medium is a task that takes significant time and effort; for this reason, autonomic network control has attracted a lot of attention from the research community.

This work was partially supported the NSF (award no. 0331690), UCSD-CWC (Center for Wireless Communications), and the Army Research Office.

Autonomic network control can be seen as the automation of the network reconfiguration process carried out by an intelligent network controller without manual interactions. In order to achieve performance optimization through automated control, the recently proposed cognitive networking paradigm can be utilized. A cognitive networking system observes the historical behavior of the network in the context of space, time, spectrum, and the social characteristics of the users and thereby learns the interrelationship between various network parameters in order to take decisions for the current and future operation of the network in such a way as to achieve local, system-wide, or end-to-end performance objectives. Examples of early cognitive networking approaches can be found in [1], [2]. In this work, we demonstrate the effectiveness of cognitive network control for solving the problem of dynamic channel selection in Wi-Fi hotspots.

Rest of the paper is organized as follows: Section II presents related work in this area. In Section III, we outline architecture of CogAP system. Design of CogAP's traffic sensing and cognitive controller modules is given Sections III-A and III-B, respectively. CogAP prototype development is in Section IV. Section V presents performance evaluation of CogAP's channel selection strategy. Finally, Section VI contains concluding remarks.

II. Related Work

Cognitive networking is a relatively new networking paradigm. Though cognitive radios and cognitive networking have a lot of differences [3], they are similar in one aspect: cognition from their respective operating environments. Work on an early centralized cognitive networking system for enterprise WLANs is suggested in [4]. In that work we proposed a central autonomic control system that gathers network state information from each of the APs and uses neural network approaches to determine the best channel for APs in the network. Some works on channel allocation strategies used a static approach where the channel decision is carried out separately from network deployment by using techniques such as graph coloring [5]. However, such centralized or static network control system is very expensive for small scale WLANs such as Wi-Fi hotspots and home networks which typically contain only one AP per service provider/residence. In these network environments, we can reduce the cost of network control by equipping the same AP with cognitive functionality. Hence in this work, we propose an architec-

978-1-4244-4148-8/09/$25.00 ©2009

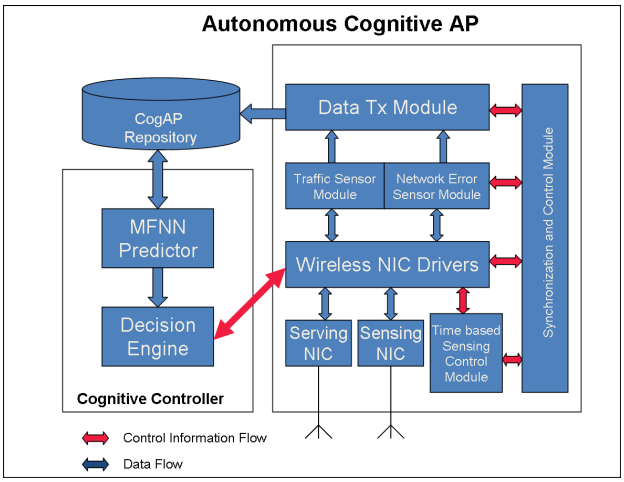

Fig. 1. Architecture of CogAP device

ture for autonomous Cognitive AP (CogAP) and develop a prototype system using embedded single board computers. Note that the entire cognitive system, including the cognitive controller, cognitive repository, and traffic sensing and serving mechanism, is co-located in the embedded AP.

An interesting approach to the introduction of effective learning strategies in a cognitive network system is to use past experience, i.e., data from past environmental and performance measurements, to model the correlation between environmental conditions, possible network settings, and expected network performance. In this way, the cognitive network will be able to learn which parameter settings are most effective in which conditions. The authors of [6], dealing with the problem of network traffic prediction, evaluate the effectiveness of traditional prediction techniques such as ARIMA (Auto Regressive Integrated Moving Average) and Fractional ARIMA, and compare them with MFNNs (Multilayer Feedforward Neural Networks), concluding that MFNNs are more practical due to lower complexity and the ability to model non-linear relationships. In [7] the authors compare the performance of linear regression models with that of MFNN for the purpose of building models of the performance in mobile ad hoc networks as a function of traffic load and routing protocol being used; again, the authors conclude that MFNNs are the best modeling choice for the considered scenario.

III. CogAP Architecture

Figure 1 shows the schematic diagram of autonomous cognitive AP. It has two main modules: a sensing & serving module and a cognitive controller module.

The sensing module helps CogAP to obtain sensorial information about the surrounding communication environment. The sensing & serving module consists of two wireless network interface cards (NICs): one for sensing (monitoring) the wireless traffic related information on each of the 802.11 channels in the 2.4 GHz spectrum, and another for serving clients associated with the AP. The time-based sensing control submodule controls the time for which the traffic samples from each of the channels are to be collected. This sub-module helps CogAP to sample traffic in all the channels in a predefined static or dynamic fashion. The synchronization and control sub-module performs time synchronization of the CogAP to a public NTP server. The network error sensor sub-module is used to gather additional measurements which can be used to better characterize the conditions of the wireless medium. In particular, we are interested in collecting additional statistics like CRC errors and PHY errors (defined later in this draft). The data transfer sub-module dumps all these traffic related measurements in a local database (CogAP repository).

The Cognitive controller module is designed to identify the status of the network and the impact of different configuration settings on the network performance. It is of course possible to provide the controller directly with this knowledge, e.g., by hard coding the actions to be taken in response to different network conditions. However, a cognitive controller is expected to be able to learn these dependencies, thus relieving the engineering effort in providing the needed knowledge to the controller. The cognitive controller module has two sub-modules: prediction and decision engine. The prediction module predicts the evolution of the communication environment for all the channels by employing artificial intelligence techniques. Further this module also estimates the relationship between the network status and the network settings. Once the estimation of the dependency between the network status and the performance with respect to different network settings, CogAP's decision making module selects the most desirable network configuration.

A. Traffic Sensing Module Design

The traffic in WLANs is spread across a number of channels, however, a single wireless NIC that we have in APs for traffic sensing can typically monitor only one channel at a time. For example, the 802.11b/g-based wireless networks operate on ISM band and have 11 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 noise render an orthogonal channel useless) other channels are also being used in the configuration of APs. In small scale WLAN systems such as home WLANs, APs belong to different WLANs co-exist on the same channel and compete for radio resources in the same geographic region. To characterize traffic in such network environments, the wireless monitoring system should have the capability of monitoring all wireless channels in a spatio-temporal fashion. In order to characterize wireless traffic on all channels, the traffic sensing module has either to have a dedicated wireless NIC per channel in order to measure all the traffic in that channel, or to switch a single NIC across all the channels in a round-robin fashion, thus measuring a fraction of traffic on each channel. In the first case, the traffic sensing module would be very complex due to the presence of large number of channels in the wireless environment, and moreover storage and analysis of the captured packets from a large number of simultaneous channels could be problematic for single

978-1-4244-4148-8/09/$25.00 ©2009
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2009 proceedings.

board computer based CogAP device. Therefore, such a multi-interface complete capture solution would be very expensive and would scale poorly to multi-channel wireless network environments. The multi-channel traffic sampling scheme is therefore essential.

In our prior work [8], we studied the accuracy of various time-based sampling schemes using Normalized Kullback-Leibler Divergence (NKLD) metric [9] for designing a scalable traffic sensing module with a single NIC. We found that Systematic Timer-driven Time-based (STT) sampling strategy with a sampling period of 11 seconds and a sampling duration of one second allows accurate traffic sensing across all 11 channels in IEEE 802.11 b/g based WLANs. Hence we implemented STT scheme in time-based sensing control module of CogAP which rotates sensing NIC's channel every one second in a round-robin fashion.

B. Cognitive Controller Module Design

Traffic sensing module stores wireless traffic related information in CogAP local database repository which helps the Traffic prediction sub-module of Cognitive controller module to predict future network traffic across all channels. In this paper, we consider the problem of predicting mean network traffic (measured in Kbps) and mean CRC/PHY error rates for next one hour interval on each of the 11 channels. With this predicted information, the decision engine could then see which channel would be less crowded and reconfigure the serving NIC's channel accordingly. We designed MFNN based traffic prediction schemes to predict traffic from a large historical data set consisting of traffic information spanning several months.

Neural network based traffic prediction schemes were proposed in [10] for wireless networks. We designed three different MFNN architectures by varying inputs and studied their prediction accuracy in terms of Mean Square Error (MSE) and regression coefficient (R-value). The following parameters are used as inputs/outputs of neural network architectures: **(I) Channel**: range [1 to 11], **(II) DayOfWeek**: range [1 (Monday) to 7 (Sunday)], **(III) HourOfDay**: range [1 to 24], **(IV) BinaryDayOfWeek**: 0 for Working days and 1 for Holidays, **(V) Traffic(t)**: mean network traffic observed in time interval, hours $(t-1, t]$, measured in **Kbps**, **(VI) Traffic(t-1)**: mean traffic during hours $(t-2, t-1]$, **(VII) Traffic(t-2)**: mean traffic during hours $(t-3, t-2]$, **(VIII) HistoricalTraffic(t)**: one week back mean traffic during hours $(t-1, t]$, **(IX) CRCErrorRate(t)**: mean number of incoming 802.11 transmissions for which the link-layer CRC check failed during time interval, hours $(t-1, t]$, and **(X) PHYErrorRate(t)**: mean number of times acquisition was triggered at the PHY layer but reception was aborted because of a failure in the PLCP header check during time interval, hours $(t-1, t]$.

We studied the following three different MFNN architectures for network traffic prediction: (a) NNTP-DayOfWeek, (b) NNTP-BinaryDayOfWeek, and (c) NNTP-DayOfWeek-HistoricalHour. All of these MFNN architectures have same parameters as outputs (*Traffic(t)*, *CRCErrorRate(t)*, and *PHY-ErrorRate(t)*), however, they differ in their number of inputs. For example, NNTP-DayOfWeek predictor has *Channel, DayOfWeek, HourOfDay, Traffic(t-1), Traffic(t-2)* as its inputs. Similarly, NNTP-BinaryDayOfWeek predictor is designed by replacing the input *DayOfWeek* from NNTP-DayOfWeek with *BinaryDayOfWeek* parameter. Finally, the NNTP-DayOfWeek-HistoricalHour predictor has an additional input (*Historical-Traffic(t)*) to NNTP-DayOfWeek predictor.

Here we would like to mention that MFNN architectures that we consider for CogAP design are entirely different from the one used in [4] for centralized cognitive control of enterprise WLANs. In [4], we chose *Application Layer Throughput* as the output parameter of MFNN system. But that requires active measurements (i.e., injecting artificial test traffic into the network) to find achievable thoughputs. Such active measurement setup causes service disruptions to users when serving NIC's channel is changed frequently to find throughputs on the other channels. Hence such active measurement setup is very expensive and not feasible in Wi-Fi hotspots and therefore in this work we depend only parameters derived through passive measurements (i.e., traffic sampling) for designing our NNTP predictors.

We made use of MATLAB neural network toolbox to implement NNTP predictors and study their prediction accuracy. We constructed NNTP predictors as two-layer feedforward backpropagation networks with one hidden layer and one output layer. Hidden layer has 20 neurons with tan-sigmoid transfer function and output layer has three neurons with linear transfer function. Multiple layers of neurons with nonlinear transfer functions allow the network to learn nonlinear and linear relationships between inputs and outputs. Picking the learning rate for a nonlinear network is a challenge. As with linear networks, a learning rate that is too large leads to unstable learning. Conversely, a learning rate that is too small results in incredibly long training times. In our experiments, the backpropagation training is done by using Levenberg-Marquardt algorithm [11] with a learning rate of 0.1 and a number of epochs of 1000. Since NNTP predictors require large amount of historical data for training the neural network, we configured traffic sensor module of CogAP device to collect traffic samples for three months (January-March 2009) in the CALIT2 building of University of California San Diego campus. As we would like to predict traffic generated by other WLANs operating in the wireless environment, we process the collected packet traces to exclude CogAP's self-traffic (i.e., management, control, and data traffic generated by CogAP and its associated wireless clients). Such processed traces are grouped together based on Channel, DayOfWeek/BinaryDayOfWeek, and HourOfDay parameters. This process generated 20,738 input/output tuples which are randomly divided into three sets. 70% of the tuples for the training phase, 15% for the validation phase, and the remaining 15% for the testing phase.

MSE measures error as the average of the squared difference between the output and target values, with ideal performance yielding zero MSE. The R-value is a measure of how well the

978-1-4244-4148-8/09/$25.00 ©2009
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2009 proceedings.

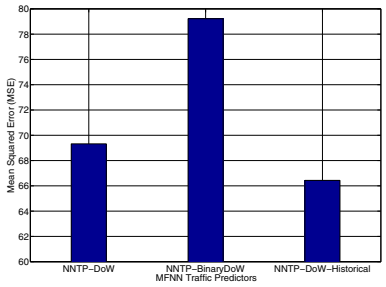

Fig. 2. MSEs of MFNN Traffic Predictors

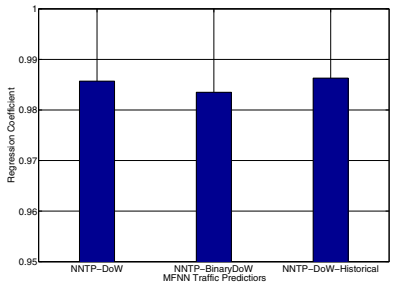

Fig. 3. R-values of MFNN Traffic Predictors

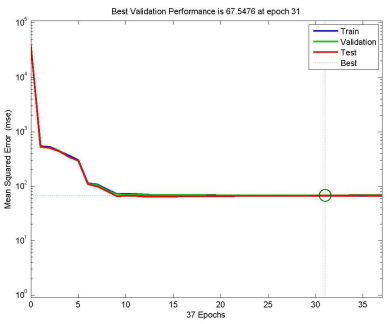

Fig. 4. MSE vs Epochs for MFNN Predictor

variation in the output is explained by the targets. If this value is equal to one, then there is perfect correlation between targets and outputs. Figures 2 and 3 show MSE R-value between predicted outputs and actual outputs (*targets*), respectively for NNTP predictors after testing phase. As shown in figures, NNTP-DayOfWeek performs better compared to NNTP-BinaryDayOfWeek which classifies DayOfWeek into only two categories. However, NNTP-DayOfWeek-HistoricalHour with its additional previous week traffic input outperforms all other schemes in terms of MSE and R-value. Thus it proves that the historical traffic information is indeed useful for optimally configuring a cognitive network. Further, in our experiments, the R-value is very close to 1, which indicates a very good fit between targets and outputs. We also conducted experiments to study the effect of various input parameters used in NNTP architectures on their prediction accuracy in terms of MSE metric. We observed that, though parameters like Channel, Hour, and DayOfWeek help in bringing down MSE value, significant reduction in MSE value is achieved only after addition of historical traffic related parameters (i.e., Traffic(t-1), Traffic(t-2), and HistoricalTraffic(t)) to the input of NNTP architectures. Thus it proves that the historical traffic information is indeed useful for accurately predicting network traffic. For NNTP-DayOfWeek-HistoricalHour predictor, the best validation performance (minimum MSE of 67.54) is obtained at epoch 31 (refer Figure 4), which demonstrates the predictor's quick learning capability.

Figures 5, 6, and 7 show regression plots of the predicted outputs with respect to training, validation, and testing target data sets, respectively. The network outputs are plotted versus the targets as open circles. The best linear fit is indicated by a solid line. The perfect fit (output equal to targets) is indicated by the dashed line. In this case, it is difficult to distinguish the best linear fit line from the perfect fit line because the fit is so good. The Y-axis defines output as a function of target ($Output = m * Target + b$), where m and b correspond to the slope and the Y-intercept of the best linear regression relating targets to network outputs. If there were a perfect fit (outputs exactly equal to targets), the slope would be one, and the Y-intercept would be zero. In this case, we can see that the numbers are very close. These plots are also useful in viewing outliers which are the samples that have extreme deviations from the line of regression. It can be observed that training phase contains more outliers compared to other two phases, this is because during training phase network does not know the underlying relationship between inputs and outputs and hence results in less accurate prediction of outputs.

1) Decision Making: Decision making engine part of Cognitive controller module makes use of traffic forecasts made by NNTP predictor to decide which channel is best for serving NIC to operate on. Pseudo code of Cognitive controller module is given in Algorithm 1. This module chooses the channel with least amount of predicted traffic load as the best channel. If it finds more than one channel with almost same amount of traffic load (less than 5% difference in traffic loads), it makes use of predicted CRC/PHY error rates to break the tie.

Algorithm 1 Cognitive Controller

```
Fetch historical input/output tuples from CogAP database repository
Train NNTP-DayOfWeek-HistoricalHour predictor with historical tuples
loop
    Perform Incremental training using instantaneous tuples
    Predict Traffic load and CRC/PHY error rates for next hour interval for all channels
    Determine best channel using the predicted data
    If current channel of Serving NIC is different from predicted best channel, reconfigure it
    sleep one hour
end loop
```

IV. CogAP Prototype Development

We now present the development of a CogAP prototype system. Hardware components used in the development include an ALIX 2C2 embedded system board, two Atheros chipset based 802.11 b/g miniPCI cards (one for sensing and another one for serving users), 8 GB Compact Flash card (for storing OS and database repository), and two omni-directional pigtail antennas. ALIX 2C2 board has 500 MHz AMD Geode processor, two miniPCI slots, and ethernet ports which make it a perfect choice for CogAP prototype development. The following software base is used for developing cognitive function modules: Linux Voyage 0.5.2, MATLAB, MySQL, MadWiFi driver, *tcpdump*, and PHP.

978-1-4244-4148-8/09/$25.00 ©2009
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2009 proceedings.

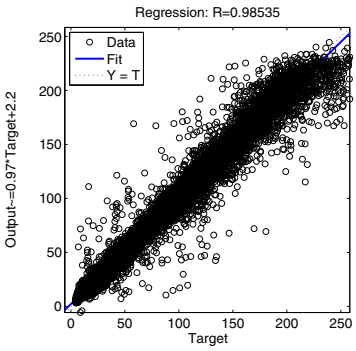

Fig. 5. Regression in Training Phase

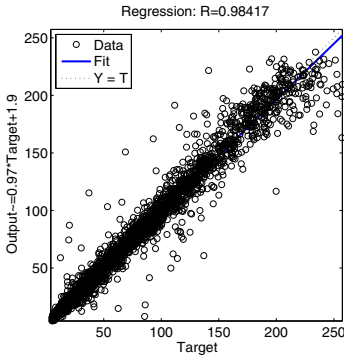

Fig. 6. Regression in Validation Phase

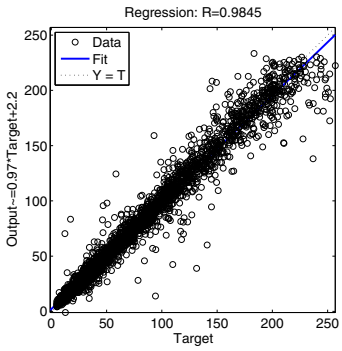

Fig. 7. Regression in Testing Phase

The traffic sensing module employing STT sampling scheme (explained in Section III-A) uses *tcpdump* packet sniffer to collect sampled packet traces in files and remits them to the data transfer module. To further reduce the storage and processing cost, *tcpdump* is configured to capture only the first 250 bytes of each sampled packet. This is a reasonable solution, since all protocol headers and Prism monitoring header are located at or near the start of the packet. The data transfer module extracts all header fields and stores them in local CogAP database repository as instantaneous traffic records, implemented using MySQL server, from which they can be queried to extract the training and testing data sets, as well as for generic analysis purposes. Sampled packet traces contain minimum of 2 million packets and maximum value goes up to 20 million packets per day. Such large size packet traces result in huge size tables (Upton 4 GB per day) in CogAP database repository. Cognitive controller based on NNTP-DayOfWeek-HistoricalHour predictor requires traffic traces for a duration of two weeks for its incremental training. Since CogAP's storage space is very limited, it creates historical traffic records from instantaneous traffic records, by averaging data referring to the same channel over one-hour intervals for the entire day. Historical records help in achieving scalability by reducing storage cost and speeding up subsequent reprocessing of information by not needing to process the complete traffic records every time (which is a very time consuming operation).

Implementation of the time-based sensing module is done using a combination of shell scripts running *wireless-tools* and MadWiFi tools. These are used to periodically switch the sensing NIC's channel setting; channel switching is done in parallel to the traffic sensing activity. In addition to the traffic samples, we gather CRC and PHY error rates with the help of the *athstats* tool, associated with MadWIFI driver.

Cognitive controller outlined in Algorithm 1 is implemented in MATLAB using neural network toolbox. MATLAB-MySQL interface is used to query MySQL server from MATLAB environment. We faced a number of issues in the implementation of MATLAB based Cognitive controller because it consumes a lot of computing power of CogAP device during the initial training phase. As given in Algorithm 1, historical tuples are used for initial training of NNTP predictor. However, the initial training needs to performed only once and it took approximately 15 minutes for CogAP device to complete this job using historical records that span a duration of three months. Incremental training and other controller tasks are performed hourly basis and CogAP took only a few seconds for completing these tasks. If historical records are not available[1], CogAP skips initial training phase and does only incremental training. For the first one week for incremental training, it could substitute HistoricalTraffic(t) with Traffic(t-3). In such scenarios, due to the lack of enough traffic samples for training the NNTP predictor, channel selection based on such predictions may not be optimal. However, over a period of time the prediction accuracy improves and Cognitive controller will make optimal reconfigurations.

V. Performance Results

In this section, we study the performance of CogAP in terms of uplink and downlink throughput obtained by wireless clients. For this purpose, we used a testbed containing one CogAP device and one wireless LAN client, which are separated by a physical distance of 10 meters (non-line-of-sight). This setup was placed in a academic laboratory building where many access points contend for the channel. Wireless test client was built using the same hardware as that of CogAP, but having a single wireless NIC, in managed mode, used to connect to the CogAP. The test WLAN client runs a modified version of the `iperf` software to carry out active measurements by performing TCP data transfers both in the uplink and in the downlink direction. We ran these active measurements for one week. In this active measurement setup CogAP and test WLAN client pair measures performance of all channels by switching serving NIC's channel to one of the 11 channels after every measurement in a round robin fashion. In this way application-layer throughput can be measured on each of the channels which is useful to determine best performing channel during *a posteriori analysis*. We compared the throughput performance of CogAP with three other schemes which are outlined below.

1. Random Scheme: In typical hotspot environments, AP's

[1]This happens when CogAP is deployed in a new geographical area for the first time.

978-1-4244-4148-8/09/$25.00 ©2009

serving NIC's channel is fixed and it is configured on one of the available orthogonal 802.11 channels. Hence to simulate such scenario, in Random Scheme we periodically (on hourly basis) switch the NIC to one of channels randomly (1, 6, and 11) which is then considered as the best channel.

2. Auto Regressive Integrated Moving Average (ARIMA): It uses lags and shifts in the historical traffic load to uncover patterns (e.g., moving averages, seasonality) and predict the future network traffic load. In our study, we use ARIMA to predict future traffic on all channels as given below.

$$\begin{aligned}\text{Traffic(t)} &= 0.3 \times \text{Traffic(t}-2) + 0.3 \times \text{Traffic(t}-1) \\ &\quad +0.4 \times \text{HistoricalTraffic(t)}\end{aligned}$$

Channel that is associated with the least predicted value for traffic load is considered as the best channel.

3. Best: Since we measure throughput on all channels, channel that is having highest *a posteriori* throughput is considered as the best channel.

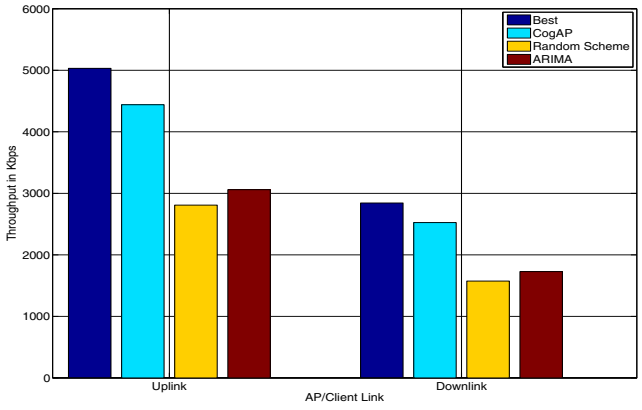

Fig. 8. Throughput performance of different Channel Selection Schemes

The uplink and downlink throughput performance for the channel selection schemes just described is reported in Figure 8. Throughput measurements are averaged over the whole measurement period of one week. Random and ARIMA schemes perform very poorly compared to CogAP channel selection scheme. Random scheme does not have any knowledge on current network conditions and randomly changes its channel, which may cause it operating on non-optimal channel. In case of ARIMA scheme, as it depends only on the moving average of past traffic to estimate future traffic it could not able to take into account the effect of environment parameters like DayOfWeek and Hour on future traffic. It is also be noted that CogAP's performance is slightly less compared to the best *a posteriori* performance. This can be attributed to prediction error of NNTP predictor. The "best" performance can be better than the one achievable by any possible predictor, especially when there is a high short-term variance in the network conditions, in the performance measurements, or in both. It is also interesting to note that downlink performance is much lesser compared to uplink performance in our experiments. This is due to spatial variation in network traffic conditions at AP and test WLAN client locations.

These results, as well as other similar measurements that have been omitted here due to space constraints, show that CogAP with its NNTP-DayOfWeek-HistoricalHour predictor based channel selection scheme outperforms legacy channel assignment strategies used in Wi-Fi hotspots and home networks, thus proving that the MFNN based approach to autonomic cognitive network control is practical and effective.

VI. Conclusions

Cognitive networking is an emerging networking paradigm for wireless networks. According to this paradigm, the wireless network should have capability to observe the behavior and events in the past in order to plan, decide, and act for optimized decisions of the present. Unlike early conceptual approaches for cognitive networking, we proposed a realization of a cognitive network element, a Cognitive AP for residential/hotspots.

We presented an architecture of CogAP, design for the cognitive controller for optimized channel selection decisions, and results from performance evaluation carried out on a prototype system. Our cognitive controller is based on MFNNs and we compared three different architectures before arriving at the best possible architecture. We observed that all the input parameters of NNTP predictor had played a key role in determining the optimal channel. Results from the performance evaluation show that the proposed autonomous CogAP outperforms other competing schemes. Further, our solution achieves performance close to the best possible performance.

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] B. S. Manoj, R. Rao, and M. Zorzi, "Architectures and Protocols for Next Generation Cognitive Networking," in *Cognitive Wireless Networks: Concepts, Methodologies and Visions*, M. Katz and F. Fitzek, Eds. Springer, 2007.

[3] C. Fortuna and M. Mohorcic, "Trends in the development of communication networks: Cognitive networks," *Computer Networks*, vol. 53, no. 9, pp. 1354–1376, June 2009.

[4] N. Baldo, B. R. Tamma, B. S. Manoj, R. Rao, and M. Zorzi, "A Neural Network based Cognitive Controller for Dynamic Channel Selection," in *Proceedings of IEEE ICC 2009*, June.

[5] J. Riihijarvi, M. Petrova, and P. Mahonen, "Frequency allocation for WLANs using graph colouring techniques," in *Wireless On-demand Network Systems and Services (WONS 2005)*, 2005, pp. 216–222.

[6] H. Feng and Y. Shu, "Study on network traffic prediction techniques," in *Proc. of the WiCOM*, vol. 2, September 2005, pp. 1041–1044.

[7] 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.

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

[9] S. Kullback, in *Information Theory and Statistics*. John Wiley and Sons Inc., 2007.

[10] S. Gowrishankar and P. S. Satyanarayana, "Neural network based traffic prediction for wireless data networks," *Computational Intelligence Systems*, vol. 1, no. 4, pp. 379–389, December 2008.

[11] 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.

978-1-4244-4148-8/09/$25.00 ©2009
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2009 proceedings.