Empirical Model-based Adaptive Control of MANETs

Abdelhamid Moursy, Ikhlas Ajbar, Dmitri Perkins, and Magdy Bayoumi
The Center for Advanced Computer Studies
University of Louisiana at Lafayette
Lafayette, LA 70504, USA
Emails: {agm0015, ixa2388, perkins, mab}@cacs.louisiana.edu

Abstract—**The performance of mobile ad hoc networks (MANETs) depends upon a number of dynamic factors that ultimately influence protocol and overall system performance. Adaptive protocols have been proposed that adjust their operation based on the values of factors, such as traffic load, node mobility, and link quality. In this work, however, we are investigating the feasibility of an adaptive model-based self-controller that can manage the values of controllable factors in MANETs. In general, the proposed self-controller should determine a set of factor values that will maximize system performance or satisfy specific performance requirements. The model-based controller adapts or reconfigures system-wide parameters or protocol operation as a function of the dynamically changing network state. In this paper, we describe the proposed self-controller, its design issues, and provide a preliminary case study to demonstrate the effectiveness and tradeoffs of two potential empirical-modeling techniques: regression and artificial neural networks.**

Keywords- empirical modeling; autonomic network management; neural networks; adaptive control.

I. INTRODUCTION

Mobile Ad Hoc Networks (MANETs) are comprised of mobile nodes that self-organize to form a communications network among themselves, without the assistance of any predefined or centralized infrastructure (e.g., access points in WLANs). MANETs can be characterized as a network of mobile nodes that communicate using fully distributed protocols over a multi-hop wireless topology that changes frequently due to node mobility and wireless channel effects. These unique characteristics of MANETs facilitate a quickly deployable, low cost, and flexible network solution that is applicable in several application areas, including emergency search and rescue, environmental monitoring, military and law enforcement, ad-hoc gaming, and others [1,2]. However, these unique characteristics result in a plethora of factors that influence system performance and management capabilities, especially as the network scales to hundreds or thousands of nodes. Factors that can influence system performance include both uncontrollable factors (e.g., wireless channel effects), and controllable factors (e.g., protocol parameters, transmission power). Some factors (e.g., traffic load) may be controllable or uncontrollable depending upon the deployment strategy and the network protocols (e.g., admission control) used.

In general, the goal is to select a combination of values for the controllable factors that yield the optimal performance or, alternatively, to meet specified performance requirements. However, the network state may vary dynamically (due changes in uncontrollable factors), requiring a different set of values for the controllable factors. So, dynamically determining the "best" factor values based on network performance and state changes is more appropriate, although non-trivial. For example, the system performance may depend upon the main and interactive effects of multiple factors, such that the appropriate value for factor A depends upon the value of factor B. The complexity of ad hoc networks and the interactions among parameters and protocols at different layers brings up an important question regarding self-management: *Can a MANET system automatically determine the appropriate combination of controllable factor values that will optimize system performance or satisfy a set of performance demands as the network state changes (i.e., as the uncontrollable factor values change)*?

In this work, we focus on the design of a system-level (as opposed to protocol-level) adaptive control framework that can coordinate protocol actions and parameter values across layers, with the aim of maximizing overall system performance. As shown in Fig. 1, our proposed approach includes the design of a feedback-based controller that uses empirical models to autonomously control and manage system performance [3, 4]. The idea is to develop models of the system that accurately characterize functional relationship among (1) the performance metrics, (2) the controllable factors, and (3) the uncontrollable factors. While, in principle, the models can be either analytical or empirical, we believe the large number of factors and the dynamic nature of MANETs leads to too many simplifying assumptions when using analytical models.

The remainder of this paper is organized as follows. Section II provides the proposed feedback Adaptive controller, which includes experimental design and empirical models, ranking table and optimization approach. Followed by case study in section III and conclusions and future work in section IV.

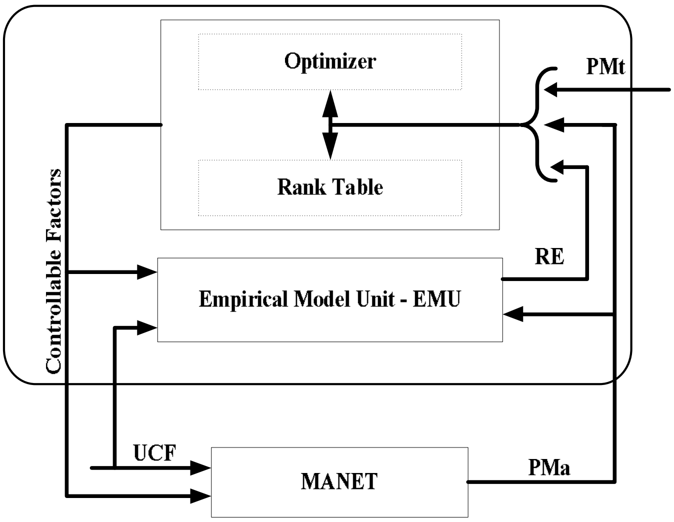


Figure 1. Model-based Control of MANETs

PMt: Targeted Performance Metrics, **PMa**: Actual Performance Metrics

UCF: Uncontrollable Factors, **RE**: Regression Equations

II. PROPOSED FEEDBACK ADAPTIVE CONTROLLER

The proposed feedback adaptive controller is composed of three parts that work together to optimize the set of controllable factor values to optimize the network performance or to meet specific performance requirements. These parts are the empirical model unit, the rank table, and the optimizer. The empirical model unit is responsible for continuously monitoring the MANET environment, i.e. network factors and performance metrics. It monitors the controllable and the uncontrollable network factors and the correspondent network performance metrics to build the regression model and to extract regression equations. These regression equations together with the actual and the targeted performance metrics values are then sent to the factors controller (Optimizer and Rank table). The controller passes these inputs to the proposed ranking approach to fill a performance metrics ranking table that will be used by the optimizer to optimize the controllable factors to meet required network performance. Finally, the MANET controllable factor values are adjusted to the new optimized values. Because there are still uncontrollable factors that will deviate the actual performance metrics values from the targeted ones, the whole optimization process will be repeated again. In the next subsections we will introduce each part of the whole control process.

A. Factors and Performance Metrics

In MANETs, a large number of factors (e.g., node mobility, network size, traffic load, energy consumption, transmission power, node density, bandwidth, …) can affect every measure of performance, including user-level quality of service-QoS, throughput, packet delivery ratio-PDR, and jitter. Not only is performance determined by the main effects of individual factors, but also their interactions effect should be considered (e.g. increasing the transmission range increases the node density, which reduces available bandwidth). These controllable factors and their operation ranges can be fed to statistical Analysis of Variance-ANOVA tool to study the main and interaction effects of these factors on the system performance metrics in order to decrease the number of factors to only those that are statistically significant [5]. A comprehensive factors and performance investigation is important and should be done in order to build an accurate model.

B. Empirical Models

To date, we have examined the prediction capabilities of two empirical modeling techniques—linear regression and artificial neural networks. Linear regression models can be expressed as a mathematical equation that characterizes a response metric as a function of the independent factors and a set of parameters. The predicted performance *y* for the second order regression model is of the form:

$$\boldsymbol{y = \beta_0 + \sum_{i=1}^{k} \beta_i x_i + \sum_{i=1}^{k} \sum_{j=i+1}^{k} \beta_{ij} x_i x_j + \sum_{i=1}^{k} \beta_{ii} x_i^2 + \varepsilon}$$

where $\boldsymbol{k}$ is the number of factors, $\boldsymbol{\beta_0}$ is the mean of y, the $\boldsymbol{\beta}_i$ and $\boldsymbol{x}_i$ are the regression coefficient and factor value for network factor $\boldsymbol{X_i}$, respectively. The advantage of the linear regression approach is that it results in a physical equation that can be manipulated, while also capturing the main and interactive effects among factors. It is important to note that linear regression implies that the model is linear with respect to the regression coefficients, not the factor values. So, curvature due to changes in factor values is captured by linear regression models [5].

The artificial neural network model, used as a nonlinear modeling technique, is constructed as a feed-forwarded back propagation network that is composed of three layers: input, hidden, and output layers, as in Fig. 2. The number of neurons at the input layer corresponds to the number of factors (controllable and uncontrollable). The hidden layer neurons use a hyperbolic tangent sigmoid function to calculate the layer's output. This transfer function is mathematically equivalent to the *tanh* function [6]. The output layer has one neuron, corresponding to the performance metric modeled, with a linear transfer function.

The predicted performance $\boldsymbol{y}$ given by this neural network model is as follows:

$$y = Purelin\left(\sum_{i=1}^{5} w_{2i1} * Tansig\left(\sum_{j=1}^{n} w_{1ji} x_j + \theta_{1i}\right) + \theta_{21}\right)$$

where $\boldsymbol{x_j}$ is the input to node **j** in the input layer. $\boldsymbol{W_{1ji}}$ is the weight between node **j** in the input layer to node **i** in the hidden layer. $\boldsymbol{\Theta_{1i}}$ is the bias of node **i** (plays the role of an intercept in the linear regression) in the hidden layer. $\boldsymbol{w_{2i1}}$ is the weight between node **i** in the hidden layer to the node in the output layer. $\boldsymbol{\Theta_{21}}$ is the bias of the node in the output layer.

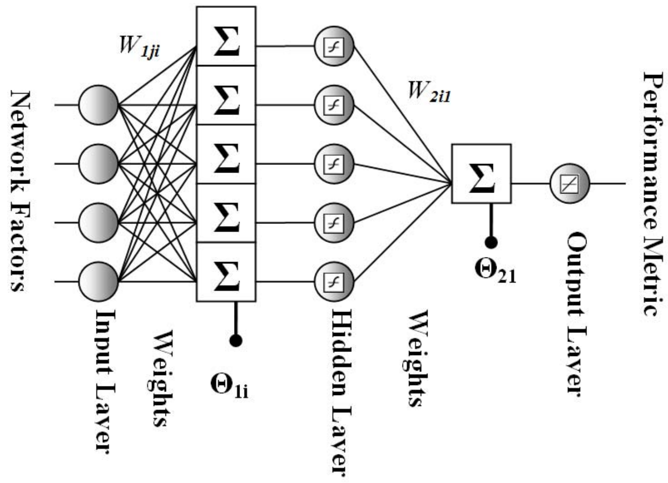


Figure 2. The proposed Neural Network Model

The numbers 5 and **n** are the number of nodes in the hidden layer and in the input layer, respectively. To construct the neural network models, first output from the experimental scenarios are divided into a training set, a testing set, and a validating set. The training set is used to train the network for function approximation (nonlinear regression). The validating set is used to stop training early if the network performance on the validation set fails to improve. Test set is used to provide an independent assessment of the model predictive capability.

C. Rank table and optimizer

After an empirical model is developed, its regression equations are passed to the optimizer unit together with the actual and targeted performance metric values. Finding the controllable factor values that meet the targeted performance metrics is done in the optimizing unit by using constrained nonlinear Sequential Quadratic Programming- SQP technique to optimize these empirical regression equations [6]. The objective is to get the factor values that minimize the performance metrics regression equations, given a lower and upper bound for each network factor. For performance metrics that need to be maximized (e.g. PDR) its regression equation is multiplied by (-1) before the optimization process starts. However, using such an algorithm to search the entire factor space will be slow and may lead to finding a local minimum. Therefore, we propose an approach that starts by dividing the factor space (i.e. main cube in Fig. 4 a and 4 b) into smaller sub-regions. In our case study, two levels (high and low) per each of the three factors are used, i.e. total of 8 sub-regions. Factor ranges can be divided into more than two levels for more accuracy but with added time complexity. Each of these sub-regions has its own factors ranges (upper and lower bounds) different from other sub-regions. In each of these sub-regions, we propose to rank each performance metric using a scale that goes from zero (bad performance, e.g. Delay is high) to two (good performance, e.g. Delay is low). To know the performance metric rank in a sub-region, the performance metric values in the corner points of that specific sub-region are averaged and rated as either good or bad on the scale, two or zero, respectively. Following the same approach, a rank table is then filled with the ranking for all performance metrics in all sub-regions (see table I). This table is constructed only once and will not be updated unless the network environment is changed. Each row in the table represents a specific sub-region (with its upper and lower factor bounds). Each column in the table represents a performance metric ranking. Adding all the rankings in a row gives an overall network performance rank for this network sub-region. The best overall network performance should be in the sub-region with largest total performance rank.

The optimizer will use the rank table to identify the best sub-regions to search for optimum controllable network factors. In case a specific network performance is required, e.g. Delay and Jitter must be low, the rank table will suggest the sub range, i.e. the row, which will satisfy the network performance requirements. Otherwise, the sub range with the largest total performance rank will be reported to the optimizer to search in its boundaries for the optimum controllable network factors. Searching for optimum factors values in small ranges will lead to the global minimum and will also save time of searching the whole factors ranges. In the case where two rows have the same total rank, the optimizer will search in both sub-regions and then rank both results. The factor values that will give higher overall performance rank will be selected. Finally, MANET controllable factor values will be adjusted according to the optimizer output.

III. CASE STUDY

To demonstrate the feasibility of the proposed adaptive control framework on MANET, the following sections present an illustrative case study conducted on basic MANET.

A. Factors and Performance Metrics

In this case study, three quantitative factors are considered: node speed (m/s), traffic load (number of source nodes), and number of nodes. One qualitative factor, routing protocol, is also considered. The three quantitative factors have been shown to have a significant impact on performance in previous studies [7]. The selected values for node speed represent a relatively slow-moving node at 5 m/s = 11 miles/hour (walking/running scenario), an average speed of 15 m/s = 35 miles/hour (city driving), and a faster moving node at 25 m/s or 55 miles/hour (freeway driving scenario). The offered load is adjusted by varying the percentage of nodes within the network that act as source nodes (10%, 15%, and 20%). To explore the impact of network size on MANET performance control (scalability of the proposed model), we consider a small size network of 100 nodes, a moderate size of 300 nodes, and a relatively large network of 500 nodes. The network density (i.e., the average number of nodes in a given area) is held constant at 6 nodes, for all simulations runs, by varying the terrain size to avoid the congested areas and congestion related problems. Previous work has shown that the optimal throughput and connectivity are achieved when the network density ranges from 6-8 nodes [7]. The reactive routing protocols AODV and DSR are chosen to show how the proposed technique can quantify and compare the impact of distance vector routing versus source routing in MANETs.

Each combination of factor values is called a design point or experimental scenario. Therefore, in this study we covered scenarios that range from small size network with slow speed nodes and lightweight traffic to large-scale networks with fast speed moving nodes and heavy traffic. For each of these design points, we computed the system performance with respect to three response metrics: (1) *End-to-end delay* – computed at the application layer and measures the average time taken for a packet to be transmitted from a source to a destination node in seconds; (2) *Packet delivery ratio* – measures the average number of packets received to the number of packets sent; and (3) *Jitter* –measures the average variation in time between arriving packets in seconds.

B. Simulation Setup

The Qualnet network simulator [8] is used to simulate all design points. The simulation time for each scenario is chosen to be 1500 seconds. Each scenario is replicated five times and the resulting metrics are averaged to reduce the number of outliers in the statistical analysis of the network behavior. The traffic sources transmit constant bit rate-CBR traffic of 512-byte packets at a rate of 2 packets per second during each simulation run. The traffic load is varied by varying the ratio of source nodes to total network size. The random waypoint mobility model is used to model the node mobility patterns using a node pause time of 30 seconds and node speed of 5, 15, and 25 meters per second. The IEEE 802.11b standard is used at both the physical and medium access control layers with a channel data rate set to 2 Mbps, with the optional RTS/CTS handshake is always on. Free space propagation model is used with transmission range of 300 m.

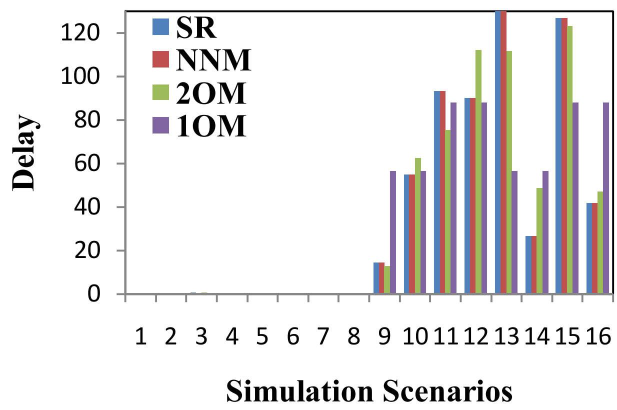
C. Empirical Models

Using statistical design of experiments approach, including full and fractional factorial designs, and standard regression analysis techniques, we run 30 simulation experiments and build (1) first and (2) second-order regression models and (3) neural network models for each of the three performance metrics described in Section A. For more details on the models construction, validation process, and design matrix see [7].

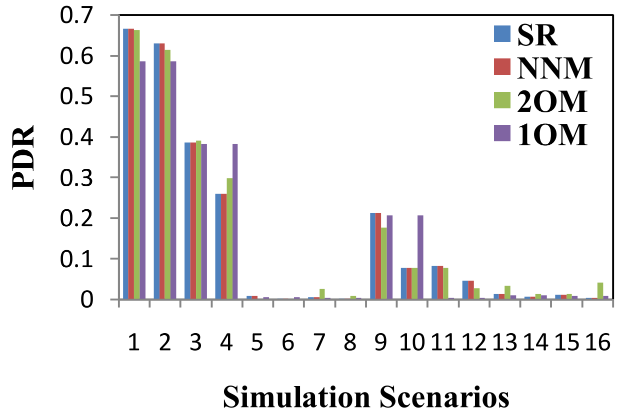
D. Discussion of Simulation Results

Fig. 3 shows the simulation results for the 16 common simulation scenarios between all models (linear and neural network models). The first eight scenarios and the second eight are for AODV and DSR, respectively. Fig. 3 shows the prediction ability for the first order regression model (1OM), second order regression model (2OM), and neural network model (NNM). The most accurate model was the neural network model as it captures the nonlinearity of the MANET system and has the ability to learn this nonlinearity that exists in the factor readings. The disadvantage, however, of the neural network model, is that it is used as a black box. Although, the second order model is less accurate than neural network model, it is able to yield relatively accurate results while also generating physical regression equations.

Fig. 3 also presents the first order regression model ability to model the MANET with least accuracy as compared to the previous two models. We present the comparisons with the first order regression to show that in some cases it gave comparable results to the real simulation readings, therefore it can be used for its simplicity when high accuracy is not that much needed. Cases like: 10, 11, 12 in Fig. 3a and 2, 3, and 9 in Fig. 3b, support our statement. Fig. 4a and 4b visualize the 3-D response surface models for End-to-End Delay and Packet Delivery Ratio, respectively. In Fig. 4a, we can see that Delay is minimum in two regions, lower number of nodes/traffic loads and larger number of nodes/traffic loads. This is because, as we mentioned earlier, network density is kept the same throughout simulations. So, for cases when a network has low traffic and low number of nodes, delay is also expected to have lower values. Increasing offered load and number of nodes but with keeping the node density the same, this is expected not to have that effect on delay. Fig. 5 summaries a qualitative comparison between the three models according to the number of scenarios needed to build the model, complexity of the model, and the model prediction error.

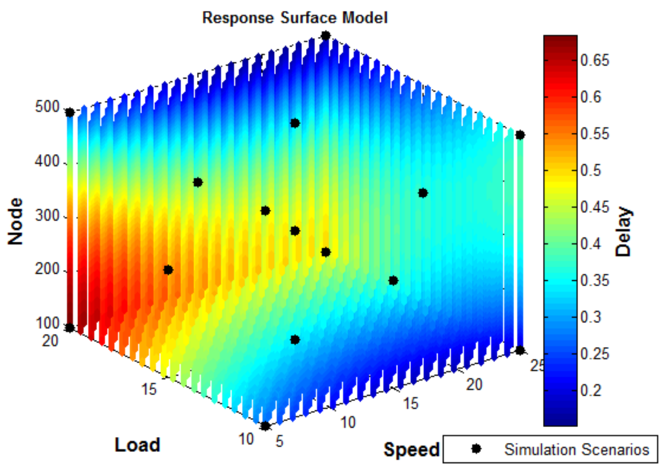


(a) End-to-End Delay

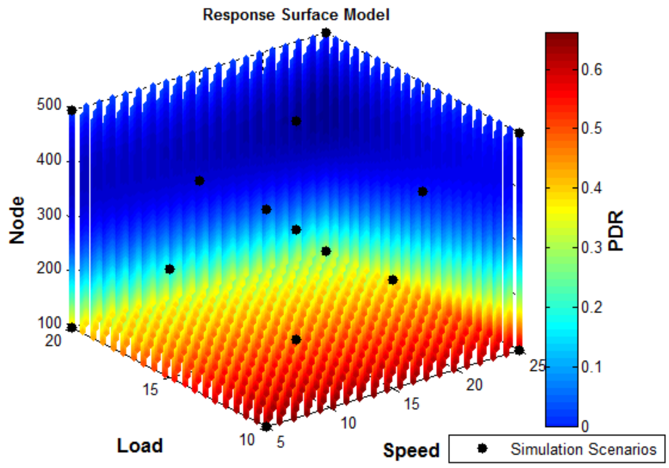


(b) Packet Delivery Ratio

Figure 3. MANET Simulation Results–SR, Neural Network Model-NMM, Second Order Model- 2OM, and First Order Model- 1OM



(a) End-to-End Delay



(b) Packet Delivery Ratio- PDR

Figure 4. AODV Response Surface Models

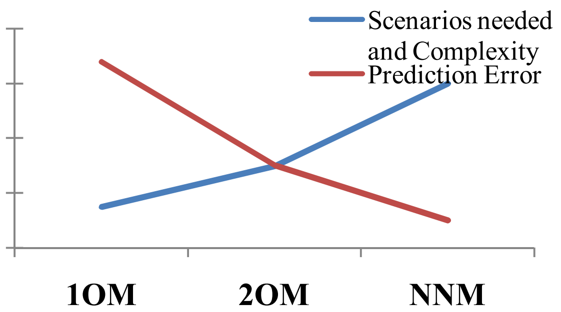


Figure 5. Relation between empirical models 1OM, 2OM, and NNM

It is worthy to mention that delay and jitter are found to have a high correlation (strong relationship) in all simulation scenarios, especially when AODV is used. The response surface models for Delay almost match the response surface model of the Jitter.

E. *Rank table*

Table I shows the performance rank table for MANET running AODV routing protocol. Each row in the table represents a bounded sub-region. For example, sub-region one has lower bound of (S=5, L=10, N=100) and upper bound of (S=15, L=15, N=300). Sub-region two has lower bound of (S=5, L=15, N=100) and upper bound of (S=15, L=20, N=300), and so on. Sub-regions one and three suggest best overall network performance. While in sub-region 7, the network performance is the worst.

TABLE I. PERFORMANCE METRICS RANK TABLE FOR AODV

Network Sub-regions	Delay	PDR	Jitter	Total Rank
(N=L, S=L, L=L) 1	2	2	2	**6**
(N=L, S=L, L=H) 2	0	2	0	2
(N=L, S=H, L=L) 3	2	2	2	**6**
(N=L, S=H, L=H) 4	0	0	0	2
(N=H, S=L, L=L) 5	2	0	2	4
(N=H, S=L, L=H) 6	2	0	2	4
(N=H, S=H, L=L) 7	0	0	0	0
(N=H, S=H, L=H) 8	2	0	2	4

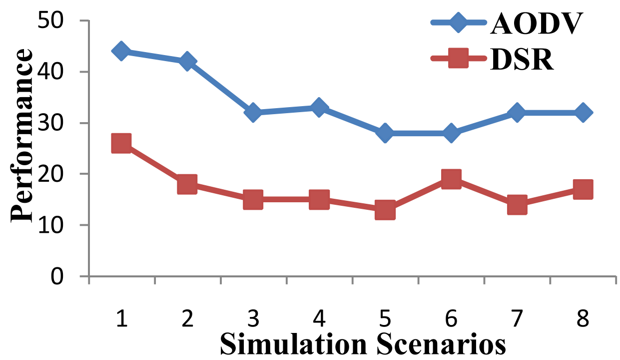


Figure 6. Simulation Scenarios overall performance

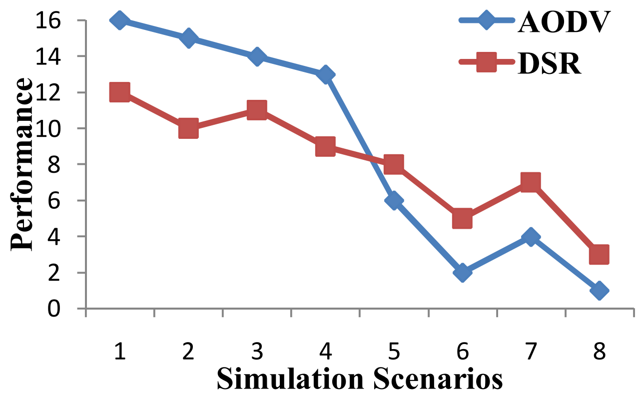


Figure 7. Simulation Scenarios PDR performance.

To show how the proposed technique of performance metric ranking helps in better understanding of MANET, Fig. 6 and 7 compare the network performance for both AODV and DSR routing protocols. Ranking each performance metric in each simulation scenario and summing them gives an overall ranking of how well the network is in that scenario. Figure 6 shows that AODV outperform DSR in all the simulation scenarios when considering overall performance. However, considering only PDR performance metric, figure 7, AODV outperform DSR in the first 4 scenarios (100 network nodes) and DSR outperform AODV in the remaining scenarios (500 nodes) which suggest using DSR in large number of nodes networks if packet delivery ratio is of concern.

As mentioned before, constructing the ranking table is done only once. Updating its data will be needed when the MANET environment changes. The complexity to construct the table is $\boldsymbol{O(n)}$, since all factors will be traversed only once to calculate the factors rank. Sorting of total performance metrics rankings of all the table rows to select the search region is of $\boldsymbol{O(nlog_2^n)}$, to the base two because we divide each factor range into two levels.

Finally, in this proposed approach, the number of performance metrics and the number of network factors are not limited and can grow, as the model needs. The time complexity to construct the performance metrics rank table, which is done only once, will be compensated by faster searching in small factors regions instead of the entire design space.

F. Optimizer unit (numerical example)

Once the empirical regression equations and rank table are ready, the optimizer will search for optimum network factors values that satisfies the targeted network performance metrics. For example, suppose that the network's nodes speed is 5 m/s, network Load is %10, and number of network nodes is 100. From simulation, the network performance metrics for this network state are 0.769, 0.385, and 2.777 for the Delay, PDR, and Jitter, respectively. To optimize these performance metrics assuming equal priority for all of them, the optimizer will check the rank table, table I, to identify the sub-regions that will be searched. In this case, the rank table will suggest that the search will be in both cubes 1 and 3 only (total rank equal 6). The optimizer then searches both sub-regions looking for the optimized factors values. The optimized factors in region one are 15, 10, and 100 for the Speed, Load, and Number of nodes, respectively. For these factors values, the Delay, PDR, Jitter values are 0.183, 0.647, and 0.0205, respectively. For region three, Speed, Load, Number of nodes values are 19.06, 10, 100, respectively. For these factors values, the Delay, PDR, Jitter values are 0.175, 0.636, and 0.039, respectively. Both set of optimized controllable network factors satisfy overall network performance by setting the number of nodes to be 100, traffic load to be %10, and speed to either 15 or 19.06 m/s. If, for example, Delay is required to be 0.2 Sec, i.e. ranked high (=2), then the controller will not search in sub-regions 2, 4, or 7 (high delay regions) and will only search in sub-regions 1 and 3 since they are low delay regions with higher total rank equal to 6.

IV. Conclusions and Future Work

In this paper, we presented an adaptive control technique to be used jointly with empirical models to analyze and optimize the performance of MANETs. Linear regression and neural network models are constructed to model the effects of network factors on MANET performance metrics. These models are described and investigated according to their prediction accuracy, complexity, and amount of data required to build the models. The proposed performance metrics ranking technique for searching for the optimum factors values that enhances the overall network performance is then described to be used jointly with the constructed regression models.

The next step in this study is to conduct an exhaustive factor analysis to identify the critical factors and to quantify their respective performance effects on various performance metrics. We must also investigate prediction performance/tradeoffs of non-linear regression techniques, including auto-regressive techniques that allow the empirical model to be updated in real-time. After building the models, we intend to implement/test the self-controller in a simulated network, while addressing practical questions regarding efficient monitoring, data collection, and deployment strategies. The final step will be the deployment of the self-controller in our outdoor campus testbed, comprised of static mesh backhaul and mobile ad hoc nodes (laptops and PDAs) [9]. The proposed technique is expected to give better performance optimization results when used with wireless mesh networks since they are more static than MANETs.

Acknowledgment

This work was supported in part by the National Science Foundation (NSF) under NSF Career Grant No. 0448055, the U.S. Department of Energy (DOE) under Award Number DE-FG02-04ER46136, and by the State of Louisiana, Louisiana Board of Regents under Contract Numbers DOE/LEQSF (2004-07)-ULL and LEQSF(2003-06)-RD-A-35.

References

[1] Jagannathan Sarangapani, Wireless Ad Hoc and Sensor Networks: Protocols, Performance, and Control. CRC Press, 2007.

[2] Shih-Lin Wu, Yu-Chee Tseng, Wireless Ad Hoc Networking. Auerbach Publications, 2007.

[3] Gelenbe E, "Towards autonomic networks," in the proceedings of International Symposium of applications and the Internet, 23-27 Jan 2006.

[4] Richard Mortier and Emre Kiciman, "Autonomic network management: some pragmatic consideration," in the proceedings of SIGCOMM workshop on Internet network management, Pisa, Italy, 2006, pp. 89-93.

[5] D.Perkins, H. Hughes, and C. Owen, "Fators afficting the performance of ad hoc networks," in the proceedings of IEEE International Conference on Communications 2002, New York, April 2002.

[6] Math Works MatLab 2007b, www.mathworks.com/products/

[7] Abdelhamid Moursy, Ikhlas Ajbar, Dmitri Perkins, and Magdy Bayoumi, "Building Emperical models of Mobile Ad Hoc Networks," in the Proceedings of the SCS/IEEE International Symposium on Performance Evaluation of Computer and Telecommunications systems, July 16-18, 2007.

[8] Q. Support, Qualnet user manual, Scalable Network Technologies, Culver City, CA, Tech. Rep., 2003.

[9] **U**biquitous **Co**mputing and **M**onitoring **S**ystem-UCoMS, http://www.ucoms.org