# LOSS MODEL APPROXIMATIONS AND SENSITIVITY COMPUTATIONS FOR WIRELESS NETWORK DESIGN

John S. Baras, Vahid Tabatabaee, George Papageorgiou, Nicolas Rentz and Yadong Shang
Institute for Systems Research, University of Maryland, College Park, MD 20742

ABSTRACT

*We develop and evaluate a new method for estimating various performance metrics of multi-hop wireless networks, including MANETs. The estimates depend implicitly on network design parameters, and therefore the method can form the basis for designing such networks so as to achieve predictable performance bounds. Explicit simulation models for large wireless networks are too complex and can not be used for analysis and design of wireless networks in realistic settings. We introduce a simple approximate (throughput) loss model that couples the physical, MAC and routing layers. The model provides quantitative statistical relations between the loss parameters used to characterize multiuser interference and physical path conditions on the one hand and traffic rates between origin-destination pairs on the other. We use a fixed point approach together with recently proposed models for the MAC layer to find a consistent solution satisfying all equations in the set. The result is an implicit model of performance metrics, such as throughput, packet loss, delay, parameterized by design variables, such as path routing probabilities, power levels, noise levels. We next apply Automatic Differentiation (AD) to these implicit performance models, and develop a methodology for design and parameter selection/tuning of wireless protocols.*

## I. INTRODUCTION

While there has been impressive progress in wireless technologies, widespread deployment of multi-hop wireless networks is still far away. One of the main reasons for this state of affairs is the lack of systematic methodologies and toolkits, for the design and dimensioning of such networks so as to have predictable performance bounds, as measured by a few key performance metrics. This is principally due to the inherent uncertainty and variability of the wireless medium and the interdependence between the performance of wireless links. In wired networks, link capacities are fixed and unless there is a failure there is no variation in network topology and link capacities. This is not true in wireless networks. Link capacity in wireless networks is not fixed and depends on different factors, such as transmission power, interference caused by transmissions over other links in the network, mobility and environmental factors. Due to the performance variability and interdependence, design, analysis, optimization, management and maintenance of such systems are daunting tasks. Modelling and model-based performance evaluation tools are badly needed to assist wireless network engineers and researchers in these tasks.

It is possible to develop packet level simulation tools based on physical (PHY) and medium access control (MAC) layer models using various packages. However the packet level simulation of multi-hop wireless networks with the appropriate PHY and MAC layer modelling turns out to be too complex and time consuming for the design and analysis of wireless networks in realistic settings. Our objective is to develop low complexity combined analytical and computational (numerical) models, which can efficiently *approximate* wireless networks performance. Such models have several applications in the design and analysis of wireless networks:

*Protocol analysis:* Performance of a multi-hop wireless network under practical settings depends on many factors including the lower layer protocols, the physical environment conditions, the number of users and their mobility patterns. It is almost impossible to analytically evaluate and predict the impact and interaction of all these factors. This task cannot be accomplished by extensive simulations either. Therefore, we need systematic model-based methods to evaluate performance, reliability, robustness, sensitivity and scalability of proposed protocols.

*Component based design:* In a hierarchical and modular approach to the design of complex systems, system functionality is divided among several components at multiple levels. Alternative designs and solutions are proposed for each component. For a particular application, we have to rely on fast and effective evaluation models to figure out the appropriate combination of alternative components for optimal performance.

*Parameter Tuning:* Performance of alternative compo-

[1]Prepared through collaborative participation in the Communications and Networks Consortium sponsored by the U. S. Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement DAAD19-01-2-0011. The U. S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon.

1-4244-1513-06/07/$25.00 ©2007 IEEE

nents and layers in a multi-hop wireless network depends on many design parameters such as power, modulation at the PHY layer, and back-off window size at the MAC layer, number of paths and routing policy at the network (routing layer). Whether we use local search algorithms or more sophisticated methodologies such as automatic differentiation (AD) [1], for sensitivity analysis and performance optimization we need to have an efficient model in our design loop.

We propose an alternative approach based on the fixed point method and loss network models for performance evaluation and optimization. Loss network models [2] were originally used to compute blocking probabilities in circuit switched networks [3] and later were extended to model and design ATM networks [4]–[7]. In [7] reduced load approximations were used effectively to evaluate quite complex ATM networks, with complex and adaptive routing protocols, and multi-service multi-rate traffic (different service requirements). The main challenge in developing loss network models for wireless networks is the coupling between wireless links. This coupling is due to the transmission interference between different nodes in proximity with each other. By using probabilistic physical and MAC layer models, we propose to approximate interference and contention as inter-link traffic dependent loss factors. Such an approximate model provides a system of equations describing the relations between reduced link rates. We then develop iterative fixed point methods [4], [5], [7] to approximately compute consistent traffic rates, packet loss and throughput of the network from these equations.

*Solution Overview:* Our goal is to develop a systematic methodology for performance evaluation, and optimization of wireless networks. In this paper, we focus on system throughput and packet loss as the main performance metrics[1]. The performance of every link $l$ in the network is parameterized by two separate loss factors that incorporate the impacts of the PHY and MAC layer protocols respectively. The loss factors of a link are functions of the corresponding link and its neighbors' incoming traffic rates. Note that due to these loss factors the incoming and outgoing rates of each link are different.

Let us assume that we know the incoming traffic rate for every source-destination pair in the network, the set of paths and the fraction of traffic that is forwarded over each path in the network. We want to compute the network throughput and packet loss over each link of the network. We use two different sets of equations. In the first set, we use simple network flow relations to find the incoming and outgoing traffic rate of each link as a function of the link loss parameters. In the second set of equations we use PHY and MAC layer models to determine the link loss parameters as a function of the incoming traffic rates of the network links. Then, we use the two sets of coupled equations iteratively, in a fix point setting, until they converge to a consistent solution that satisfies both sets. The solution provides an approximation to the packet loss over each link and the throughput (ratio of the outgoing to the incoming traffic) of the network.

The quality of the approximation depends on the quality of the PHY and MAC layer loss models. For the MAC layer, we use the seminal work of Bianchi [8] that models the MAC layer IEEE 802.11 protocol as a Markov chain and evaluates the IEEE 802.11 channel capacity as a function of the number of users. For the PHY layer, we have simply assumed that the loss rate is fixed. Clearly we can incorporate more complex and variable models in our overall approach.

In addition to the performance model, for design and parameter optimization, we need to perform sensitivity and gradient analysis for parameter optimization and robustness evaluation. We use Automatic Differentiation (AD) [1] for sensitivity analysis. AD is a powerful method to numerically compute the derivatives of a software-defined function (i.e. a computer program implementation of the function, which we also consider a 'model' following modern systems engineering and hybrid systems formalisms [9]–[11]. The analysis model that we generated based on the fixed point iterations and loss models is the input function to the AD and the output of the AD is the partial derivative of the performance metric (e.g. throughput) with respect to defined input parameters (i.e. design variables or parameters). It is important to note that the method allows for very complex design parameters to be implicitly embedded in the input function to the AD module (see for example the work of Liu and Baras in [7]). As an example, we show how we can use this methodology to find the optimal load distribution among multiple paths in the network to maximize throughput (this can be easily extended to probabilistic multi-path routing as in [12], [13]). In this example the gradient projection algorithm is used to find the optimal load distribution, and AD is used to compute the gradient of the network throughput with respect to the load distribution parameters.

The rest of the paper is organized as follows: In Section II, we briefly review the Bianchi model results for the IEEE 802.11 MAC layer protocol. In Section III, we elaborate on the fixed point methodology and develop the set of equations used. In Section IV, the automatic differentiation method and its application to routing optimization using gradient-projection algorithm is described. In Section V,

[1] A similar methodology can be developed for end-to-end delay which is the other important performance metric.

we provide examples of applying the methodology, and use them to describe performance and characteristics of the developed algorithms and methodology.

## II. MAC LAYER MODELLING

The IEEE 802.11 protocol is the dominant local area wireless technology. The primary MAC layer algorithm of the IEEE 802.11 is carrier sense multiple access with collision avoidance (CSMA/CA) with binary slotted exponential backoff. Bianchi [8] provides a simple, but extremely accurate analytical model to compute the IEEE 802.11 throughput, under the assumptions of finite number of terminals and ideal channel conditions. The analysis applies to both of the packet transmission schemes, namely, the basic access and the RTS/CTS access mechanisms. We use Bianchi's model to approximate the MAC layer induced loss factor that we use in our analysis model.

Consider a fixed number $n$ of contending connections. In Bianchi's model every connection is in saturation condition, i.e. it always has a packet available for transmission. This is appropriate for our problem and method, since we are interested in design and performance bounds. Since all packets are consecutive and there is no empty queue, each packet needs to wait for a random backoff time before the transmission attempt. Let $W$ be the minimum contention window size and $m$ be the maximum backoff stage, then the maximum contention window size is $2^m W$.

Let $b(t)$ be the stochastic process representing the backoff time counter for a given connection and $q(t)$ the stochastic process representing the backoff stage of the connection at time $t$. Let $\tau$ be the stationary probability that the connection transmits a packet in a generic time slot. It is also assumed that at each transmission attempt, regardless of the previous retransmissions numbers, each packet collides with constant and independent probability $p$. The random process $(q(t), b(t))$ is modelled as a two-dimensional discrete-time Markov process. Hence, we can compute the state transition probabilities and the stationary distribution of the corresponding Markov chain. We can also obtain the following relations [8] between the key parameters $p$ and $\tau$:

$$p = 1 - (1-\tau)^{n-1}$$
$$\tau = \frac{2(1-2p)}{(1-2p)(W+1) + pW(1-(2p)^m)} \tag{1}$$

Note that the above two equations represent a nonlinear system in the two unknowns, which can be solved using numerical techniques. The system has a unique solution. To reproduce Bianchi's result, we use a fixed point method to solve those two equations. For each value of $n$ we solve these equations once and use the results in our fixed point model. Let $S$ be the normalized system throughput, defined as the fraction of *time* the channel is used to successfully transmit payload bits. In the paper [8], it is shown how we can directly compute the system throughput from parameters $p$ and $\tau$ for 802.11 networks working based on the basic or the RTS/CTS access mechanisms. We use the same methodology to derive $S$ for different values of $n$.

## III. FIXED POINT ALGORITHM

The network topology graph consisting of $N$ nodes is given. Each node has the ability to transmit packets at the rate of $\Lambda$ bits/second to the nodes that are connected to it, i.e., for simplicity, we assume that the physical layer capacity of all links is fixed and equal to $\Lambda$. The extension to different rates for every pair of adjacent links is straightforward; it only complicates notation and thus the main thrust of the argument may be hidden. All nodes use omni-directional antennas, and all neighbor nodes of the transmitting node receive the signal. The connectivity and interfering property between each node pair is decided by the Signal-to-Noise (SNR) ratio (transmission power, distance, modulation, etc). Note that two nodes are neighbors and connected if they can directly communicate, and two nodes are interfering if one of them can not receive data, while the other one is transmitting data to a third node. Let $c = 1, \cdots, C$ be the set of *commodities* in the network. Each commodity is specified by its source-destination pair $(I(c), O(c))$, and traffic demand rate $r_c$ between them. Network links (connections between neighbor nodes) are specified either by their index $l = 1, \cdots, L$ or their source-destination pair $(i, j)$. Note that $r_c$ is the average traffic rate generated at the source node (the incoming rate), which may not be equal to the traffic received at the destination node (the outgoing rate) if there is packet loss in the network. We assume that $\eta_l$, the PHY layer link loss probability is known and fixed.

The routing is known and fixed and it is defined by the set of end-to-end paths (i.e. the paths between the origin and destination of each commodity) and the fraction (probability) of the incoming traffic that is transmitted on each of these paths. Let $\Pi_c$ be the set of the paths that are used for commodity $c$. Consider a path $\pi_{c,k} \in \Pi_c$, then $\alpha_{\pi_{c,k}}$ is the fraction (probability) of commodity $c$ traffic transmitted over path $\pi_{c,k}$ at $I(c)$, the source node of commodity $c$. We have

$$\sum_{\pi_{c,k} \in \Pi_c} \alpha_{\pi_{c,k}} = 1, \quad for \quad each \quad c = 1, \cdots, C \tag{2}$$

Our goal is to find a consistent set of link loss parameters and traffic rate parameters that satisfy two sets of equations.

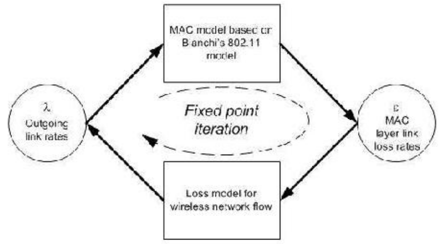


Fig. 1. Block Diagram of the Fixed Point Model Components.

The first set is derived from the network loss model and computes the link outgoing traffic rates $\lambda_l$, $l = 1, \cdots, L$ from the MAC layer effective loss parameters $\epsilon_l$, $l = 1, \cdots, L$. The second set is based on Bianchi's model and computes the loss parameters $\epsilon_l$ from the rates $\lambda_l$. As depicted in Fig. 1, the fixed point method applies these two mappings iteratively, until convergence to a consistent solution $(\epsilon^*, \boldsymbol{\lambda}^*)$ that satisfies both mappings is achieved. The existence of a consistent solution follows from the facts that both mappings are continuous and bounded and map a compact subset of $R^{2L}$ into itself, via an application of a fixed point theorem [14]. In the rest of this section we derive the two mappings.

*First Mapping: From $\epsilon$ to $\lambda$*

We assume that the packet loss probabilities due to MAC layer contention are given. We compute the $\lambda_l$'s, which are the *effective* (outgoing) data rates of the network links. Let $(l^\pi(1), \cdots, l^\pi(k_\pi))$ be the set of the links in path $\pi$, which are ordered from the first to the last hop in the path. Let $s_\pi$ be the commodity that path $\pi$ is serving. Then, $\lambda^\pi_{l^\pi(i)}$, the outgoing data rate of the $i$-th link of path $\pi$ is,

$$\lambda^\pi_{l^\pi(i)} = r_{s_\pi} \alpha_\pi \prod_{j=1}^{i} (1 - \eta_{l^\pi(j)})(1 - \epsilon_{l^\pi(j)}). \tag{3}$$

Note that $r_c$ is the corresponding source incoming traffic rate for commodity $c$, $\alpha_\pi$ is the fraction of that traffic routed on path $\pi$. The two terms in parentheses specify the percentage of the traffic that is successfully transmitted over the first $i$ links of the path $\pi$. Let $E(l)$ be the set of paths that share link $l$. Then the total traffic rate on link $l$ is,

$$\lambda_l = \sum_{\pi:\pi \in E(l)} \lambda^\pi_{l^\pi(j_\pi)} \tag{4}$$

The notation convention in the last equation is that there exist an index $j_\pi$ for path $\pi$ such that the link $l$ whose total traffic rate we calculate is identical with the link $l^\pi(j_\pi)$ of path $\pi$. Equations (3), (4), taken together, provide the desired mapping from the vector of $\epsilon$'s to the vector of $\lambda$'s. These computations are executed throughout the network synchronously.

*A. Second Mapping: From $\lambda$ to $\epsilon$*

This mapping is based on the Bianchi's [8] results, which we reviewed in Section II. Let $H_i$ be the set of interfering nodes with node $i$. The set of nodes that interfere in the transmission from node $i$ to $j$ is the union of node $i$'s and $j$'s sets of interfering nodes. The total traffic exiting nodes in the set $\{H_i \bigcup H_j \bigcup \{i\} \bigcup \{j\}\}$ contends with each other and share the same multi-access channel. The total traffic demand for this channel is,

$$X_{ij} = \sum_{n=1}^{N} \sum_{m \in \{H_i \bigcup H_j \bigcup \{i\} \bigcup \{j\}\}} \lambda_{mn}. \tag{5}$$

For now assume that we have calculated $S$ for this channel. Recall that $S$ is the fraction of time that the channel is successfully used to transmit data packets. We assume that the contending links have equal capacity $\Lambda$. Therefore, the channel capacity is $S\Lambda$. We assume that the channel capacity is divided proportionally between all contending connections. Therefore, the link $(i, j)$ effective data rate is,

$$u_{ij} = \begin{cases} \lambda_{ij} & \text{if } X_{ij} \leq S\Lambda \\ \frac{S\Lambda}{X_{ij}} \lambda_{ij} & \text{if } X_{ij} > S\Lambda \end{cases} \tag{6}$$

The intermediate MAC layer loss factor for link $(i, j)$ is,

$$\epsilon'(i, j) = \frac{\lambda_{ij} - u_{ij}}{\lambda_{ij}}. \tag{7}$$

Assume now that we are at iteration $k+1$, the new value for MAC layer loss factor of link $(i, j)$ is the weighted average of the previous iteration value and the intermediate value computed in (7):

$$\epsilon^{k+1}_{(i,j)} = \beta \epsilon^k_{(i,j)} + (1 - \beta)\epsilon'(i, j) \tag{8}$$

The weighted average is introduced to avoid rapid changes and oscillations in the computation of $\epsilon$ and $\lambda$ in the fixed point iterations.

In summary, for each connection (link) in the network we have defined and computed the channel capacity. The channel capacity is based on the Bianchi's saturation model, and hence a function of the number of interfering nodes with the corresponding link. After convergence, the total effective (outgoing) data rate of each channel is less than its throughput. Equations (5), (6), (7), (8) taken together, provide the desired mapping from the vector of $\lambda$'s to the vector of $\epsilon$'s. These computations are executed throughout the network synchronously.

In addition to the throughput, the fixed point method provides the loss rate on every link of the network. Note that in (6), we assume that the channel capacity is proportionally fairly distributed among the contending links.

However, the 802.11 protocol is not a fair protocol, and it is even possible that some links face rate starvation in an 802.11 network [15]. To avoid this problem, we make the additional assumption that the incoming traffic of each link is controlled by a rate-controller, so that it does not exceed its allocated rate, $\lambda_{ij}$. Efficient modelling of the multi-hop 802.11 MAC protocols without rate control is the subject of future research, as it becomes much more complicated even in simpler settings [15]–[17].

## IV. DESIGN METHODOLOGY

The fixed point algorithm enables us to do performance analysis for a given network configuration. However, analysis alone is not enough, and we need to develop a methodology for network configuration and optimization too. We use optimal routing design as an example to illustrate our proposed design methodology. Assume that we want to find optimal values for the routing parameters $(\alpha_1, \cdots, \alpha_K)$ to maximize the network throughput. The fixed point method provides a computational scheme that, after convergence (i.e. the fixed point), describes the performance metric as an implicit function of the design parameters. Thus, we do not have (or obtain) analytic expressions of the performance metric evaluations, but instead, we have a program that computes the values of the performance metric, while implicitly providing the dependence of the values on the design parameters.

For parameter optimization, we need to compute the sensitivities of the performance metric(s) (here the throughput) with respect to the design parameters (here the routing parameters). Many powerful methods for design and sensitivity analysis are based on gradient-based schemes. Since we do not have an explicit functional description of the network performance metric, we need to rely on computational methods that numerically approximate the gradients.

Automatic Differentiation (AD) [18], [19] is a method to numerically evaluate the derivative of a function specified by a computer program. In order to compute the gradient values, we use the ADIC package [20] that is a source translator augmenting ANSI-C programs with statements for the computation of derivatives using the AD method. In general, given a source code $C$, with $n$ inputs and $m$ outputs, we want to generate another source code $C'$ that computes the original outputs and their derivatives with respect to the input variables. Ideally, $C'$ should be fast, accurate and require little development time. One approach is to use divided differences, which does not directly produce a derivative code but rather approximates the derivatives by evaluating the function $f$ at multiple points. The main drawback of this method is difficulty to assess the accuracy of the approximation. Further, computation of $n$ partial derivatives requires $n+1$ function computations.

The alternative approach is AD. AD works by systematically applying the chain rule of the differential calculus at the elementary operator level and thus does not incur the errors inherent in divided difference approximations. The execution of any program is basically a series of elementary operations. Thus, a particular set of input values to a program induces an execution path that transforms input values into the output values. ADIC uses the source-to-source transformation method [20] to implement the AD algorithm.

In our case, the input to the ADIC is the fixed point algorithm that we developed in ANSI C code. ADIC generates a new version of the program that computes both the original result, which is throughput, and its derivatives with respect to the input parameters, which are the path routing probabilities. We then use the computed gradients to find the optimal routing parameters which maximize the throughput. To that end, we use the gradient projection method which is explained in the following.

### A. Optimization and Path Selection Algorithms

We use the gradient projection method to maximize the throughput. In this approach, we iteratively compute or estimate the gradients of the throughput with respect to the routing parameters and based on that will update the path probabilities. In addition, we have to project the computed values back into the constraint set, i.e. we have to make sure that all probabilities are positive and the path probabilities of every source-destination pair sum up to one. The explicit computation of the gradients is clearly impossible, hence we use the code generated by ADIC from our fixed point program which numerically computes the throughput and its gradients.

We have implemented the Dreyfus K-shortest path algorithm [21] for path selection. For a given set of link weights and integer value $k$ and source-destination pair, this algorithm finds $k$ loop free paths with minimum total weight. We set all link weights to one, but it is possible to use other weights based on the distance, bandwidth, interference or other performance related criteria.

## V. NUMERICAL EXPERIMENTS

We consider the network shown in Fig. 2. The network has 11 nodes and edges specify nodes that can directly communicate with each other. A communication between node $i$ and $j$ is successful if and only if no other neighbor node of $i$ and $j$ transmits data at the same time. For the MAC layer, we use the IEEE 802.11 FHSS with 1 Mbps capacity for each link. The contention window size, $W$ is

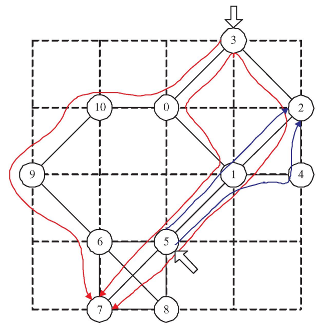


Fig. 2. Network Topology 1.

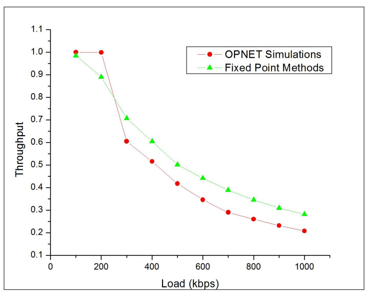


Fig. 3. Comparison between simulation results and fluid approximation.

32, and the backoff stage, $m$ is 3. We use the fixed point method to derive the MAC layer loss parameters.

For validation purposes, we compare the loss model results with simulation results from OPNET 12.0 platform. First, we consider one connection, using one path from node 3 to node 7, using the path (3-0-1-5-7). In the OPNET model, we set up the data rate supported by each station to 1Mbps. The transmission range of these stations is set to 300m radius and the buffer size to practically infinite. Moreover, we set OPNET's Long Retry Limit parameter to 1. Setting the Long Retry Limit parameter to 1 results to packet losses at the MAC layer. These losses are associated to the $\epsilon$ parameter in our fluid approximation. The packet size as well as the packet sending rate is constant, and according to the data rate we wish to achieve. We collect statistics concerning the load and the throughput achieved in the network, as well as the Bit Error Rate (BER) for each link in our connection. We set the packet error rate of our loss model according to the computed BER of the OPNET model. The derived throughput curves are shown in Fig. 3, where the derived throughput from the OPNET validates and follows the loss model throughput curve.

Another metric of interest is the computation time. Using a 1.66GHz computer, for networks containing between 1 and 10 connections, it takes at most 7 seconds to compute the throughput achieved for 9 different data rates between 100kbps and 500kbps, while OPNET requires between 23

TABLE I

RUNNING TIME COMPARISON BETWEEN OPNET AND LOSS MODELS (IN SECONDS).

| Number of conn. | 1 | 3 | 5 | 7 | 9 |
|---|---|---|---|---|---|
| Loss Model | 0.21 | 1.09 | 2.36 | 4.23 | 6.16 |
| OPNET | 1421 | 2334 | 1817 | 2250 | 2458 |

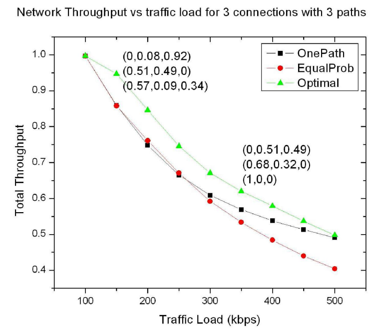


Fig. 4. Throughput for 3 connections each with 3 paths.

and 43 minutes to obtain these results, as shown in Table I.

Next, we consider multiple connections, which are: from node 3 to 7 with 3 paths: (3-0-1-5-7), (3-2-1-5-7), (3-0-10-9-6-7), from node 4 to 9 with 3 paths: (4-1-5-6-9), (4-1-0-10-9), (4-2-3-0-10-9), and from node 8 to 6 with 3 paths: (8-6), (8-5-6), (8-7-6). The network throughput for 3 different routing policies is given in Fig. 4. The routing policies are, single path minimum hop routing, 3 path routing with equal probabilities $(1/3)$ and optimal routing derived from the fixed point algorithm and ADIC as described in Section IV. The optimal routing algorithm outperforms all the other policies. It is also interesting that using multiple paths with equal probabilities is not necessarily better than a single path. In fact, for greater load values single path outperforms multi-path with equal probability. This indicates that in the wireless networks we have to design and analyze protocols and policies carefully and should not rely on intuition.

The second topology that we consider is shown in Fig. 5. The MAC and physical layer models are similar to the first scenario. In this network, we have connection 1 from node 3 to 5, connection 2 from node 16 to 21, and connection 3 from node 17 to 22. Fig. 6 shows the throughput vs. number of paths. If we have more paths available, our optimization algorithm will use them properly and find the best traffic allocation and achieve the highest total throughput.

## VI. CONCLUSIONS AND FUTURE WORK

We have developed a new method for performance analysis of wireless networks, by combining loss network models for the MAC and PHY layers, with routing, through

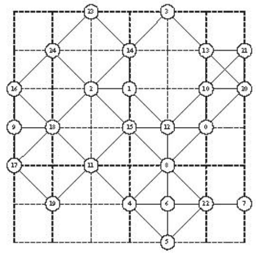

Fig. 5. Network Topology 2.

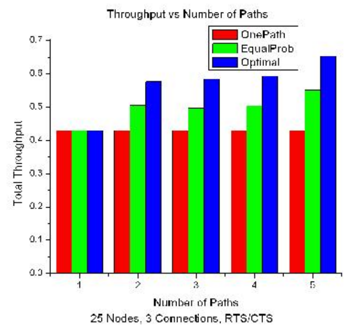


Fig. 6. Throughput for 3 routing policies for various path counts.

fixed point iterations. The method computes sensitivities of performance metrics with respect to design parameters using automatic differentiation. We illustrated applications in a few examples. Future work includes additional modules for MAC and routing, considering hidden node problem in MAC layer, derivation of other performance metrics such as delay and buffer over-flow, extending the sensitivity analysis via duality, scalability analysis for large networks, convergence proofs, and accuracy bounds on various metrics. Furthermore, we will perform extensive cross-validation of the method's performance predictions and designs against discrete event simulations.

## ACKNOWLEDGEMENTS

This work is prepared through collaborative participation in the Communications and Networks Consortium sponsored by the U.S. Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement DAAD19-01-2-0011. Research was also supported by NASA Marshall Space Flight Center under award no. NCC8-235. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U. S. Government.

## REFERENCES

[1] C. H. Bischof, L. Roh, and A. J. Mauer-Oats, "ADIC: an extensible automatic differentiation tool for ANSI-C," *Software Practice and Experience*, vol. 27, no. 12, pp. 1427–1456, Dec. 1997.

[2] F. P. Kelly, "Loss networks," *The Annals of Applied Probability*, vol. 1, no. 3, pp. 319–378, Aug. 1991.

[3] ——, "Blocking probabilities in large circuit switched networks," *Advances in Applied Prob.*, vol. 18, no. 2, pp. 473–505, June 1986.

[4] D. Mitra, J. A. Morrison, and K. G. Ramakrishnan, "ATM network design and optimization: A multirate loss network framework," *IEEE/ACM Trans. Net.*, vol. 4, no. 4, Aug. 1996.

[5] S. Chung, A. Kashper, and K. W. Ross, "Computing approximate blocking probabilities for large loss networks with state-dependent routing," *IEEE/ACM Trans. Net.*, vol. 1, no. 1, Feb. 1993.

[6] A. G. Greenberg and R. Srikant, "Computational techniques for accurate performance evaluation of multirate, multihop communication networks," *IEEE J. Select. Areas Commun.*, vol. 5, no. 2, pp. 266–277, Feb. 1997.

[7] M. Liu and J. S. Baras, "Fixed point approximation for multirate multihop loss networks with adaptive routing," *IEEE/ACM Trans. Networking*, vol. 12, no. 2, pp. 361–374, Apr. 2004.

[8] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE J. Select. Areas Commun.*, vol. 18, no. 3, pp. 535–547, Mar. 2000.

[9] M. A. Austin, J. S. Baras, and N. I. Kositsyna, "Combined research and curriculum development in information-centric systems engineering," in *Proc. of INCOSE'02*, Las Vegas, NV, July 2002.

[10] M. A. Austin and J. S. Baras, "An introduction to information-centric systems engineering," in *Proc. of INCOSE'04*, Toulouse, France, June 2004.

[11] A. Bemporad, A. Bicchi, and G. Buttazzo, Eds., *Proceedings of the $10^{th}$ International Conference on Hybrid Systems: Computation and Control*. Pisa, Italy: Springer, Apr. 2007.

[12] R. G. Gallager, "A minimum delay routing algorithm using distributed computation," *IEEE Trans. Commun.*, vol. 25, no. 1, pp. 73–85, Jan. 1977.

[13] D. Bertsekas, E. M. Gafni, and R. G. Gallager, "Second derivative algorithms for minimum delay distributed routing in networks," *IEEE Trans. Commun.*, vol. 32, no. 8, pp. 911–919, Aug. 1984.

[14] D. R. Smart, *Fixed Point Theorems*. London, U.K.: Cambridge University Press, 1974.

[15] M. Garetto, T. Salonidas, E.W. Knightly, "Modeling Per-flow Throughput and Capturing Starvation in CSMA Multi-hop Wireless Networks," *Proc. of the IEEE Conference on Computer Communications, (INFOCOM 2006)*, Barcelona, Spain, Apr. 2006.

[16] K. Medepalli and F. A. Tobagi, "Towards performance modeling of IEEE 802.11 based wireless networks: A unified framework and its applications," in *Proc. of the IEEE Conference on Computer Communications, (INFOCOM 2006)*, Barcelona, Spain, Apr. 2006.

[17] M. M. Hira, F. A. Tobagi, and K. Medepalli, "Throughput analysis of a path in an IEEE 802.11 multihop wireless network," in *Proc. of the IEEE Wireless Communications and Networking Conference, (WCNC 2007)*, Hong Kong, China, Mar. 2007.

[18] A. Griewank, *Evaluating Derivatives; Principles and Techniques of Algorithmic Differentation*, ser. Frontiers in Applied Mathematics. Philadelphia, PA: SIAM, 2000.

[19] M. Bucker, G. Corliss, P. Hovland, and U. N. amd Boyana Norris., Eds., *Automatic Differentiation: Applications, Theory and Implementations*, ser. Lecture Notes in Computational Science and Engineering. Berlin, Germany: Springer, 2006, vol. 50.

[20] ADIC Resource Center, "Argonne National Laboratory, University of Chicago," http://www-new.mcs.anl.gov/adic.

[21] S. E. Dreyfus, "An appraisal of some shortest path algorithms," *Operations Research*, vol. 17, no. 3, pp. 395–412, May 1969.