Modelling and Optimization for Multi-hop Wireless Networks Using Fixed Point and Automatic Differentiation

John S. Baras, Vahid Tabatabaee, George Papageorgiou, and Nicolas Rentz
Institute for Systems Research,
University of Maryland,
College Park, MD 20742
Email: {baras, vahidt, gpapag, nrentz}@umd.edu

***Abstract*—We develop and evaluate a new method for estimating and optimizing various performance metrics of multi-hop wireless networks, including MANETs. We introduce a simple approximate (throughput) loss model that couples the physical, MAC and routing layers effects. The model provides quantitative statistical relations between the loss parameters that are used to characterize multiuser interference and physical path conditions on the one hand and the traffic rates between origin-destination pairs on the other. The model considers effects of the hidden nodes, node scheduling algorithms, MAC and PHY layer failures and unsuccessful packet transmission attempts at the MAC layer in arbitrary network topologies where multiple paths share nodes. We apply Automatic Differentiation (AD) to these implicit performance models, and develop a methodology for sensitivity analysis and parameter optimization for wireless protocols. Finally, we provide simulation experiments to evaluate the effectiveness and performance estimation accuracy of the proposed models and methodologies.**

I. Introduction

Our objective is to develop hybrid (analytical and numerical) models which can efficiently *approximate* the performance of a wireless network. These models can assist us in a component based design and parameter tuning procedures for wireless networks. The main mathematical tools we use are fixed point methods and Automatic Differentiation (AD) for sensitivity analysis. We assume we know the exogenous traffic rate for each source-destination pair, the set of paths and the fraction of traffic traversing each path in the network. We apply our methodology on IEEE 802.11 wireless networks and we want to compute the network throughput as well as the packet loss per link. Two sets of equations are defined. The equations of the first set are derived from the specific MAC and PHY models we consider and allow us to express the loss parameters and the outgoing rate of every link in terms of the incoming rate of the link and the other links in its neighborhood. The second set of equations is defined through the use of simple network flow relations to find the incoming traffic rate of a link from the outgoing traffic rate of the upstream links in the paths that this link belongs to and the scheduling parameters at the network nodes. These two sets of equations are coupled iteratively in a fixed point setting, until they converge to a consistent solution that satisfies both sets. The solution provides an approximation to the packet loss per link and the throughput (outgoing to the incoming traffic ratio) of the network. Furthermore, we perform sensitivity analysis to evaluate the resilience and robustness of the solution. For this, we use AD which is a powerful method to numerically compute the derivatives of a software-defined function [6]–[8]. The generated analysis model, based on the fixed point iterations, is the input to the AD. The AD provides the partial derivative of the performance metric (e.g. throughput) with respect to defined input parameters (i.e. design variables or parameters). This 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 [9]). In this paper, we use this methodology to compute the optimal load distribution among multiple paths to maximize the network throughput.

Our MAC model goes beyond Bianchi's seminal work [1] that considers saturated users with ideal (no channel losses) and homogeneous (equal physical data rate) channel conditions with no hidden terminals. The MAC model we present in the current paper modifies and generalizes the IEEE 802.11 models presented in [4] and [5] by Hira et al. Their model takes into account

blocking and interference, and computes the throughput of the individual nodes in an IEEE 802.11 network with hidden nodes. In [4] only one hop connections are considered and every node can only transmit to a single node. In [5] the model is extended to consider one single path in the network, and it is explained that the same methodology can be used to model multiple paths as long as there is no common node between the paths. Here, we modify, correct and generalize the models to consider multiple paths with common nodes. We also provide a general framework to consider effects of non-saturated flows, the scheduling algorithm, losses and MAC layer transmission failures.

II. The PHY and MAC layer Modelling

A. Preliminaries

In this section we provide the set of equations that we use to approximate the wireless link loss parameter. We consider the 802.11 MAC layer with RTS/CTS mechanism. The network consists of N nodes and a path set P that is used to forward traffic between the source destination (S-D) pairs in the network. Let P_i denotes the set of the paths that goes through a node i. The unit of time is a time slot, which is equal to the back-off slots of the 802.11 protocol. The following notation is used to represent different nodes and node subsets in the network: C_i is the set of the nodes within carrier sense range of node i, C_i^+ is C_i plus node i, C_i^- is the set of the nodes not in C_i^+ and $h_{i,p}$ is the next hop of node i in path p.

In the 802.11 RTS/CTS protocol there are two stages for packet transmission: in the first stage the RTS and CTS are sent between two nodes and in the second stage the data packet and the ACK are sent. While PHY layer failure can happen in both stages, we assume that the MAC layer failure (collision) only occurs during the first stage. Different transmission failures from node i to node j or from node i over path p are represented as follows: $\beta_{i,p}$ is the probability of PHY or MAC layer transmission failure during stage 1 or 2, $\epsilon_{i,p}$ is the probability of PHY layer transmission failure during stage 2, and $l_{i,j}$ is the probability of PHY layer transmission failure at stage 1 or 2 from node i to node j.

Let $\lambda_{i,p}$ be the arrival rate and $T_{i,p}$ be the service time of the path p packets at node i. *The service time, $T_{i,p}$ is the time that node i scheduler spends serving a path p packet, and starts from the time that scheduler selects a path p packet to be served and not from the time that the packet becomes head of the queue.*

The scheduler behavior is specified by the scheduler coefficient $k_{i,p}$, which is the average serving rate of the path p packets at node i. For simplicity, we assume that all packets have the same length. The total average throughput $\bar{\rho}_i$, of node i, is $\bar{\rho}_i = \sum_{p \in P_i} k_{i,p} E(T_{i,p})$. We assume that the scheduler coefficients are:

$$k_{i,p} = \begin{cases} \frac{\lambda_{i,p}}{(1-\beta_{i,p}^m)} & \sum_{p' \in P_i} \frac{\lambda_{i,p'} E(T_{i,p'})}{(1-\beta_{i,p'}^m)} \leq 1 \\ \frac{\frac{\lambda_{i,p}}{(1-\beta_{i,p}^m)}}{\sum_{p' \in P_i} \frac{\lambda_{i,p'}}{(1-\beta_{i,p'}^m)} E(T_{i,p'})} & \text{otherwise} \end{cases} \tag{1}$$

where m is the maximum number of packet transmission retries in the IEEE 802.11. If utilization of node i is less than one, we can serve all incoming packets as described in the first line of (1). In the 802.11, if m packet transmission attempts fail the packet will be discarded. However, we assume that the scheduler keeps scheduling the same packet until it is successfully transmitted by the MAC layer. Therefore, to compensate for the transmission failures at the MAC layer, the scheduling rate should be higher than the node arrival rate by the $1/(1-\beta_{i,p}^m)$ factor. On the other hand, if utilization is equal to one, all packets can not be served, but the service rate for each path is still proportional to its compensated arrival rate as given in the second line of (1). In this way, we model a FCFS scheduling policy. For now we assume that all nodes have infinite buffer capacity, and hence there is no packet drop in a node (this assumption is not critical and can be removed later).

The fraction of time $\rho_{i,p}$ that node i is serving path p packets is specified by $\rho_{i,p} = k_{i,p} E(T_{i,p})$.

B. Failure and Hidden Nodes Modelling

Suppose that node i is scheduled to serve a packet on path p. Assuming that node accesses the channel with a fixed probability $\alpha''_{i,p}$, and there are L back-off stages and the minimum window size is W, we can use the following relation from [1]:

$$\alpha''_{i,p} = \frac{2(1-2\beta_{i,p})}{W(1-2\beta_{i,p}) + \beta_{i,p}(W+1)(1-(2\beta_{i,p})^L)}$$

We denote the average transmission time of node i during $T_{i,p}$ with $v_{i,p}$. There are two different components in $v_{i,p}$: (i) the average time spent in the successful transmission and (ii) average time spent in failed transmissions, which we denote by $f_{i,p}$. We have,

$$f_{i,p} = \frac{\epsilon_{i,p}}{\beta_{i,p}} \tau_P + (1 - \frac{\epsilon_{i,p}}{\beta_{i,p}}) \tau_H \tag{2}$$

The first term is the average transmission time when there is a packet transmission failure and the second

term is the average transmission time when there is an RTS/CTS failure. The transmission times are,

$$\tau_H = T_{\text{RTS}}(i,p) + \text{SIFS}$$

and

$$\begin{aligned}\tau_P = T_{\text{RTS}}(i,p) + \text{SIFS} + T_{\text{CTS}}(h_{i,p},p) \\ + \text{SIFS} + T_{\text{P}}(i,p) + \text{SIFS}\end{aligned}$$

where $T_{\text{RTS}}(i,p)$, T_{CTS}, $T_{\text{P}}(i,p)$ are the transmission times for the RTS, CTS and data packet on the corresponding connection respectively. Now we can compute the average transmission time $v_{i,p}$,

$$\begin{aligned}v_{i,p} &= (1-\beta_{i,p}^m)d_{i,p} + (\beta_{i,p} + \beta_{i,p}^2 + \cdots + \beta_{i,p}^m)f_{i,p} \\ &= (1-\beta_{i,p}^m)d_{i,p} + \frac{1-\beta_{i,p}^m}{1-\beta_{i,p}}\beta_{i,p}f_{i,p} \qquad (3)\end{aligned}$$

where $d_{i,p}$ is the successful transmission time,

$$\begin{aligned}d_{i,p} = T_{\text{RTS}}(i,p) + \text{SIFS} + T_{\text{CTS}}(h_{i,p},p) + \text{SIFS} \\ + T_{\text{P}}(i,p) + \text{SIFS} + T_{\text{ACK}}(h_{i,p},p)\end{aligned}$$

and $T_{\text{ACK}}(h_{i,p},p)$ is the time to send the ACK packet.

Consider a node j in the neighborhood of node i. Node j expects to receive a path p packet from node i, if i is scheduled to serve path p, and there is no transmission from node i neighbors that are hidden from j and i accesses the channel. Therefore, the probability that j receives a path p packet from i in a time slot is,

$$\begin{aligned}\alpha_{i,p,j} &= \rho_{i,p}(1-\theta_{i,j})\alpha''_{i,p} \text{ for all } j \in C_i \\ \alpha_{i,p,i} &= \rho_{i,p}\alpha''_{i,p}\end{aligned}$$

where $\theta_{i,j}$ is the probability of transmissions from node i neighbors that are hidden from node j.

$$\theta_{i,j} = 1 - \prod_{n \in C_i \bigcap C_j^-} (1 - \sum_{p' \in P_n} \rho_{n,p'} \frac{v_{n,p'}}{E(T_{n,p'})})$$

The probability that a path p transmission from node i is successful is:

$$\begin{aligned}1-\beta_{i,p} = \left(1-l_{i,h_{i,p}}\right)\left(1-\theta_{h_{i,p},i}\right) \\ \times \prod_{j \in C^+_{h_{i,p}} \bigcap C_i} \left(1 - \sum_{p\prime \in P_j} \alpha_{j,p\prime,h_{i,p}}\right) \\ \times \prod_{j \in C^+_{h_{i,p}} \bigcap C_i^-} \left(1 - \sum_{p\prime \in P_j} \alpha_{j,p\prime,h_{i,p}}\right)^{V_{i,p}}\end{aligned}$$

In the above equation $\theta_{h_{i,p},i}$ is the probability of transmission from one of the $h_{i,p}$ neighbors that are hidden from i. This represents probability of ongoing transmission from those neighbors of $h_{i,p}$ that are hidden from i, and due to their transmission, $h_{i,p}$ does not receive RTS and/or does not transmit CTS. $l_{i,j}$ is the PHY layer transmission error probability (including RTS and CTS transmission) from i to j. The first product term is the probability of no new transmission from those neighbors of $h_{i,p}$ that are not hidden from i, and hence they can detect node i transmission after the first time slot. In the second product term, $V_{i,p}$ is the vulnerable period during which those neighbors of $h_{i,p}$ that are hidden from i are not aware of the ongoing transmission and may cause collision. Therefore, the second product term is the probability of no new transmission from those neighbors of $h_{i,p}$ that are hidden from i during the vulnerable period. For the RTS/CTS mode the vulnerable period is, $V_{i,p} = T_{\text{RTS}}(i,p) + \text{SIFS}$. Note that after $V_{i,p}$ time slots the receiving node $h_{i,p}$ will send the CTS message and unless there is an error (which is taken into account in the first term) neighbors will remain silent during the packet transmission.

C. Computing the Service Time Components

$T_{i,p}$ is the time to finish a *successful or unsuccessful* transmission of a path p packet at node i, *after* it is scheduled for transmission at node i. The average service time $E(T_{i,p})$ has four components: $d_{i,p}$ is the time spent for successful transmission of path p packets at node i, $u_{i,p}$ is the average time consumed for successful transmission of node i neighbors, $b_{i,p}$ is the average back-off time of node i for path p packets, $c_{i,p}$ is the average time spent in failed transmissions.

$$E(T_{i,p}) = (1-\beta_{i,p}^m)d_{i,p} + u_{i,p} + b_{i,p} + c_{i,p} \qquad (4)$$

For the RTS/CTS mode of operation,

$$\begin{aligned}d_{i,p} = T_{\text{RTS}}(i,p) + \text{SIFS} + T_{\text{CTS}}(h_{i,p},p) + \text{SIFS} \\ + T_{\text{P}}(i,p) + \text{SIFS} + T_{\text{ACK}}(h_{i,p},p)\end{aligned}$$

The average back-off time is

$$b_{i,p} = \sum_{n=0}^{m} W_n \beta_{i,p}^n$$

where $W_n = CW_n/2$ is the average back-off time at the n^{th} stage.

The probability of successful transmission of node i, when it is scheduled to transmit path p packets is $q_{i,p} = \alpha''_{i,p}(1-\beta_{i,p})$ and probability of successful transmission in the neighborhood of i, when it is scheduled to transmit path p packets is,

$$\begin{aligned}r_{i,p} = 1 - (1-q_{i,p}) \\ \times \prod_{j \in C_i} (1 - (\sum_{p' \in P_j} q_{j,p'}\rho_{j,p'})(1-\theta_{ji})) \qquad (5)\end{aligned}$$

Here, we assume events of unsuccessful simultaneous transmission of node i neighbors are mutually independent.

The probability that the next successful transmission is by node i, given that there is a successful transmission in the i neighborhood is, $\gamma_{i,p} = \frac{q_{i,p}}{r_{i,p}}$. Let $Q_{i,p}$ be the number of successful transmissions by neighbors of i and $t_{k,i,p}$ be the time taken by the kth successful transmission of the node i neighbors. If we assume that $t_{k,i,p}$ and $Q_{i,p}$ are independent, we have $u_{i,p} = E(Q_{i,p})E(t_{k,i,p})$. The average number of successful transmissions is, $E(Q_{i,p}) = \frac{1-\gamma_{i,p}}{\gamma_{i,p}}$. Probability that a successful transmission in the neighborhood of i belongs to a neighbor j, given that it does not belong to i is,

$$g_{j,i,p} = \frac{\sum\limits_{p\prime \in P_j} q_{j,p\prime}\rho_{j,p\prime}(1-\theta_{j,i})}{r_{i,p} - q_{i,p}}$$

and $E(t_{k,i}) = \sum\limits_{j\in C_i} g_{j,i,p}d_j$, where

$$d_j = \frac{\sum\limits_{p\prime \in P_j} k_{j,p\prime}d_{j,p\prime}(1-\beta^m_{j,p'})}{\sum\limits_{p\prime \in P_j} k_{j,p\prime}(1-\beta^m_{j,p'})}$$

This completes the computation of the $u_{i,p}$.

For $c_{i,p}$ we need to compute: (i) $x_{i,p}$ probability of successful transmission of node i given that at least one transmission has occurred in the i neighborhood and (ii) $y_{i,p}$ the probability that a failure occurs in the neighborhood of i given that at least one transmission has occurred in its neighborhood. Let $z_{i,p}$ be the probability of at least one transmission in the neighborhood of node i, when it is scheduled to transmit a packet from path p:

$$z_{i,p} = 1-(1-\alpha''_{i,p}) \prod_{j\in C_i}\Big(1-\big(1-\theta_{j,i}\big)\big(\sum_{p'\in P_j}\rho_{j,p'}\alpha''_{j,p'}\big)\Big),$$

then $x_{i,p}$ and $y_{i,p}$ can be computed using conditional probability rule:

$$x_{i,p} = \frac{q_{i,p}}{z_{i,p}} \quad \text{and} \quad y_{i,p} = 1 - \frac{r_{i,p}}{z_{i,p}}$$

The average number of collisions during $T_{i,p}$ is $y_{i,p}/x_{i,p}$ and the average collision time is $c_{i,p} = \frac{y_{i,p}}{x_{i,p}}w_{i,p}$, where $w_{i,p}$ is the average time consumed for failure transmissions in the neighborhood of i:

$$w_{i,p} = \frac{\sum\limits_{j\in C^+_i}\big(\sum\limits_{p'\in P_j}\alpha''_{j,p'}\beta_{j,p'}\rho_{j,p'}\big)(1-\theta_{j,i})f_{j,p'}}{\sum\limits_{j\in C^+_i}\big(\sum\limits_{p'\in P_j}\alpha''_{j,p'}\beta_{j,p'}\rho_{j,p'}\big)(1-\theta_{j,i})}$$

III. The Fixed Point Network Model

The fixed point algorithm attempts to find a consistent solution for two set of the equations. The PHY and MAC layer model equations presented in the previous section constitute one set of the equations. The second set simply computes the incoming traffic rates of the nodes from the scheduling and loss rates of their upstream links as follows:

$$\lambda_{h_{i,p},p} = k_{i,p}(1-\beta^m_{i,p}) \text{ for all } i,p. \tag{6}$$

The fixed point algorithm starts from the source node of each path at each iteration where the arrival rate $\lambda_{i,p}$ are fixed and given. Given the input arrival rates of a node i and its neighbors it uses the PHY and MAC layer equations provided in the previous section to compute the scheduling rates $k_{i,p}$. Then, we use (6) to compute the next hop incoming traffic rate. Then we repeat the same procedure for the next hop. We continue iterating and updating over all paths in the network until a fixed point is reached.

IV. Design Methodology

Although the fixed point algorithm provides for a performance analysis of a given network configuration, we need a methodology for network configuration and optimization. We use optimal routing design as an example to illustrate our proposed design methodology. We implement the Dreyfus K-shortest path algorithm [12] 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 the 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. We use the gradient projection method to find the optimal values for the routing parameters (routing probabilities) to maximize the network throughput.

The gradient projection method requires iterative computation of the throughput gradient. The fixed point method provides a computational scheme that, after convergence (i.e. the fixed point), describes the performance metric (i.e. throughput) as an implicit function of the design parameters (i.e. routing 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. We use Automatic Differentiation (AD) to compute the gradients.

AD is a numerical method to compute the derivatives of a program [10]. Using the fact that a computer

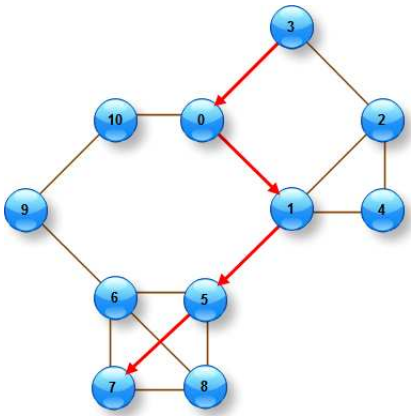
program is in fact a sequence of primary operations, automatic differentiation records the relationships between them and using the chain rule, it is able to provide the derivative of a function in a short amount of time. We implemented Automatic Differentiation by Operator Overloading in C++ using ADOL-C (Automatic Differentiation by OverLoading in C++) [11]. Operator Overloading consists of changing the type of the variables involved in the computation to a proprietary type given by the Automatic Differentiation tool to allow it to compute derivatives based on its linked libraries.

V. Results

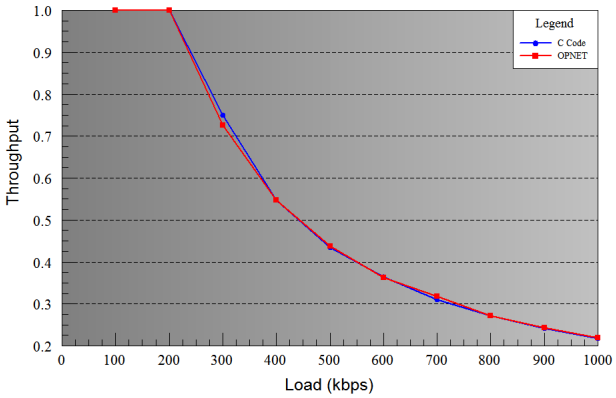
The first experiment compares the variation of the throughput computed by our fixed point method according to the desired load in the network with the same metric estimated by OPNET. We set up a simple network, presented in Fig 1(a). The blue nodes represent the wireless stations, the black links the possible wireless connections between the nodes, and the pointed red links the path used in the connection between node 3 and node 7. Our routing algorithm finds the shortest paths between nodes 3 and 7, which is 3-0-1-5-7. Using this path we apply our set of fixed point equations to compute the throughput of this connection according to the desired load. As can be seen in Fig 1(b) our code's outputs fit OPNET's predictions.

We also consider the network topology shown in Fig 2(a). This network contains three active connections: from node 3 to node 5 going through the network vertically, from node 16 to node 21 crossing the network horizontally, and from node 17 to node 22. To simplify the figure, only one path for each connection is shown in Fig 2(a). Fig 2(b) shows the variation of the network throughput according to the number of available paths for each of the connections. The performance of the optimization algorithm improves as the number of available paths increases.

The main advantage of our tool over discrete events simulation platforms such as OPNET is clearly the computation time. While our fixed point converges on the order of seconds, OPNET often requires several minutes to compute the throughput. This makes our model more suitable to compute approximations of throughput in emergency situations. Table I compares the time needed by our model and by OPNET as a function of the number of active connections in the network.



(a) Three connections network.



(b) Comparison between OPNET and fixed point method.

Fig. 1. Fixed point method.

TABLE I
Comparison of the computation time (in seconds) between OPNET and our fixed point algorithm.

Number of conn.	1	3	5	7	9
C code	0.51	2.86	4.37	5.90	10.38
OPNET	190	309	352	466	476

References

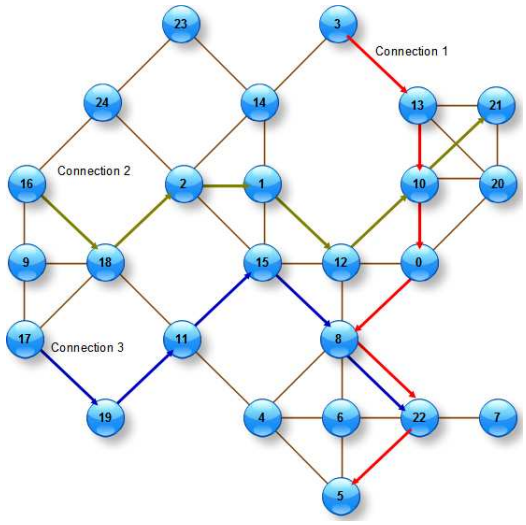
[1] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 3, pp. 535–547, Mar. 2000.

[2] A. Kumar, E. Altman, D. Miorandi, M. Goyal, "New Insights from a Fixed Point Analysis of Single Cell IEEE 802.11 WLANs," *Proceedings of the IEEE Conference on Computer Communications, (INFOCOM 2005)*

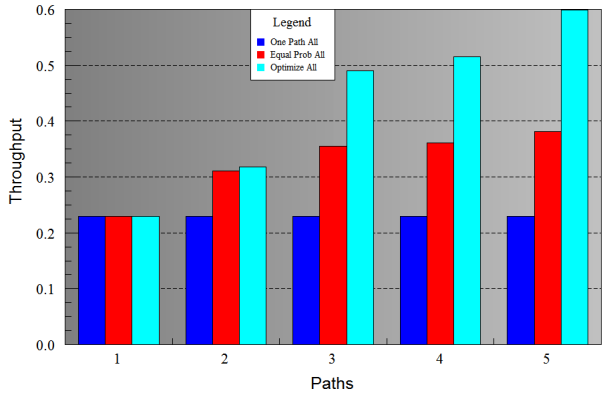
[3] M. Garetto, T. Salonidas, E.W. Knightly, "Modeling Per-flow Throughput and Capturing Starvation in CSMA Multi-hop Wireless Networks," *Proceedings of the IEEE Conference on Computer Communications, (INFOCOM 2006)*, Barcelona, Spain, Apr. 2006.

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

[5] M. M. Hira, F. A. Tobagi, and K. Medepalli, "Throughput analysis of a path in an IEEE 802.11 multihop wireless network," in *Proceedings of the IEEE Wireless Communications and Net-*



(a) Three connections.



(b) Total network throughput according to the number of available paths for an input load of 500 kbps

Fig. 2. Sensitivity analysis.

working Conference, (WCNC 2007), Hong Kong, China, Mar. 2007.

[6] 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.

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

[8] 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.

[9] 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.

[10] 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.

[11] ADOL-C, "Automatic Differentiation by OverLoading in C++, URL: http://www.math.tu-dresden.de/ adol-c/".

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