THE INSTITUTE FOR SYSTEMS RESEARCH

ISR TECHNICAL REPORT 2008-08

Performance Metric Sensitivity Computation for Optimizationand Trade-off Analysis in Wireless Networks

John S. Baras, Vahid Tabatabaee, George Papageorgiou, and Nicolas Rentz

Performance Metric Sensitivity Computation for Optimization and Trade-off Analysis in Wireless Networks

John S. Baras
Institute for Systems Research
University of Maryland
College Park, MD 20742
baras@isr.umd.edu

Vahid Tabatabaee
Institute for Systems Research
University of Maryland
College Park, MD 20742
vahidt@umd.edu

George Papageorgiou
Institute for Systems Research
University of Maryland
College Park, MD 20742
gpapag@umd.edu

Nicolas Rentz
Institute for Systems Research
University of Maryland
College Park, MD 20742
nrentz@umd.edu

ABSTRACT
We develop and evaluate a new method for estimating and optimizing various performance metrics for multi-hop wireless networks, including MANETs. We introduce an 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 takes into account effects of the hidden nodes, scheduling algorithms, IEEE 802.11 MAC and PHY layer transmission failures and finite packet transmission retries 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, parameter optimization and trade-off analysis for key wireless protocols. Finally, we provide simulation experiments to evaluate the effectiveness and performance estimation accuracy of the proposed models and methodologies.

1. INTRODUCTION
The interest in multi-hop wireless networks and their deployment, in particular mobile ones, is rapidly increasing due to the multitude of applications that are becoming available for portable wireless devices, such as mobile phones and PDAs. Nevertheless, multi-hop wireless networks still lack widespread commercial deployment due to the lack of systematic methodologies and tools that would allow for the efficient design and dimensioning of such networks with the provision of accurate performance bounds. The main reason for this is the different nature of wired and wireless networks rendering the use of wired network techniques inappropriate for the case of wireless networks. Key quantities, such as the link capacity, that remain constant in a wired network, vary in wireless communication environments with the transmission power, the interference, the node mobility and the channel condition. 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* the performance of wireless networks. 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 even in simpler settings. 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 components 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) [10], for sensitivity analysis and performance optimization we need to have an efficient model in our design loop.

Network Management and Provisioning: Models help us to understand and predict performance of the network under expected traffic and mobility patterns. Further, we can use them to detect and resolve bottlenecks and congestion points in the network.

Besides packet level simulation, the common approach for design and analysis of wireless networks is based on deterministic mathematical programming. Deterministic mathematical programming for wireless networks is based on fluid model approximations. In [13], Kelly used a deterministic fluid model to remove session and packet level details and stochastic evolution of queueing states in the network. In this approach [14, 13, 15, 16, 17], network traffic is modelled as a deterministic fluid with infinite backlog. This approach is appropriate for evaluation and optimization of performance metrics such as power, or user defined utility functions when there is no loss in the network. However, many important performance metrics, such as stochastic stability, blocking probability, packet loss, average number of active sessions are not measurable by this approach.

We propose an alternative approach based on the fixed point method and loss network models for performance evaluation and optimization. Loss network models [18] were originally used to compute blocking probabilities in circuit switched networks [19] and later were extended to model and design ATM networks [20, 21, 22, 9]. In [9] 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.

Figure 1 depicts the main blocks of the fixed point method and their interdependencies. For the fixed point model, three set of equations are derived: The first set of equations are derived from the specific MAC and PHY models and allow us to implicitly express the transmission loss parameters (transmission failure probability) and the average packet service time for each path on each node as a function of the node throughputs. The throughput of a path p at a node i is the fraction of time that node i spends in serving path p packets. The routing model derives (computes) the arrival rate for each path at each node as a function of the loss parameters and the serving rates. Finally, the scheduling model computes the serving rate and throughput of path p packets at node i as a function of packet arrival rates, loss parameters and packet arrival rates. These three sets of equations are coupled iteratively in a fixed point setting, until they converge to a consistent solution that satisfies all sets. The solution provides an approximation to the packet loss per link and the throughput (outgoing to the incoming traffic ratio) of the network.

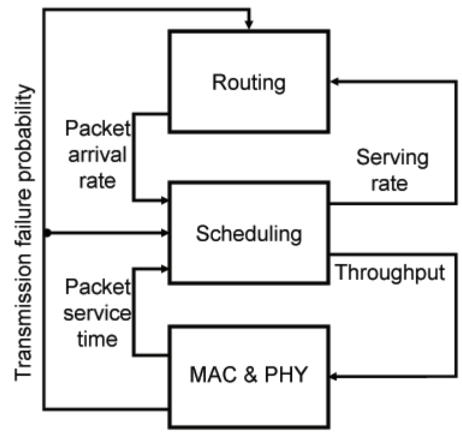


Figure 1: Cross-layer interdependence. All parameters are computed for each node of a path.

Furthermore, we perform sensitivity analysis to evaluate the resilience and robustness of the solution. For this, we use Automatic Differentiation (AD), which is a powerful method to numerically compute the derivatives of a software-defined function. The generated *implicit* 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. We use this methodology to compute the optimal load distribution among multiple paths to maximize the network throughput.

For the 802.11 MAC layer modelling, in his seminal work Bianchi [1] considers saturated users with ideal (no channel losses) and homogenous (equal physical data rate) channel conditions. The analysis generally works for the channels with no hidden terminals. This results in synchronous channel conditions. Kumar et al. [2] showed that the derivation of the access probability can be simplified by viewing the exponential back-off as a renewal process. In [4] and [3] different models are presented for the derivation of the individual node throughput in arbitrary IEEE 802.11 network topologies. Our MAC model goes beyond Bianchi's seminal work [1], in that it 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 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.

The rest of the paper is organized as follows: Section 2 describes the scheduler model, Section 3 provides the set of equations that describe our MAC and PHY layer models, Section 4 describes the routing model and the fixed point approach to the problem, while Section 5 discusses how we use Automatic Differentiation in the current framework for performance metric sensitivity computations. Finally, Section 6 provides simulation results, compares performance of the fixed point model with OPNET, and demonstrates the effectiveness of the design methodology.

2. THE SCHEDULER MODELLING

We consider a network that 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 scheduler behavior is specified by the scheduler coefficient $k_{i,p}$, which is the average serving rate of path p packets at node i. For simplicity, we assume that all packets have the same length. Let $\lambda_{i,p}$ be the arrival rate and $T_{i,p}$ be the service time of 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 the scheduler selects a path p packet to be served and not from the time that the packet becomes head of the queue.*

The scheduling rate is a function of MAC and PHY layer packet failure probabilities. 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:

- $\beta_{i,p}$: The probability of PHY or MAC layer transmission failure during stage 1 or 2.
- $\epsilon_{i,p}$: The probability of PHY layer transmission during stage 2 (data packet and ACK transmission).
- $l_{i,j}$: The probability of PHY layer transmission failure at stage 1 or 2 from node i to node j.

The total average throughput $\bar{\rho}_i$, of node i, is,

$$\bar{\rho}_i = \sum_{p \in P_i} k_{i,p} E(T_{i,p}). \tag{1}$$

In order to model a FCFS queueing policy, we assume that the scheduler coefficients are:

$$k_{i,p} = \begin{cases} \frac{\lambda_{i,p}}{(1-\beta_{i,p}^m)} & \text{if } \sum\limits_{p' \in P_i} \frac{\lambda_{i,p'}}{(1-\beta_{i,p'}^m)} E(T_{i,p'}) \leq 1 \\ \frac{\frac{\lambda_{i,p}}{(1-\beta_{i,p}^m)}}{\sum\limits_{p' \in P_i} \frac{\lambda_{i,p'}}{(1-\beta_{i,p'}^m)} E(T_{i,p'})} & \text{otherwise} \end{cases} \tag{2}$$

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 (2). 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 (2). In this way, we can 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

$$\rho_{i,p} = k_{i,p} E(T_{i,p}). \tag{3}$$

3. THE PHY AND MAC LAYER MODELLING

3.1 Preliminaries

In this section we provide the set of equations that we use to approximate the wireless link loss parameters and packet service time. This set of equations will be used as an *implicit* function to derive loss parameters and packet average service times from the node throughputs. We consider the 802.11 MAC layer with RTS/CTS mechanism. The unit of time is a time slot, which is equal to the back-off slot of the 802.11 protocol. The following notation is used to represent different nodes and node subsets in the network:

- C_i: Set of nodes within carrier sense range of node i.
- C_i^+: Nodes in the set C_i plus node i.
- C_i^-: Set of nodes not in C_i^+.
- $h_{i,p}$: Next hop of node i in path p.

3.2 Failure and Hidden Nodes Modelling

Suppose that node i is scheduled to serve a packet on path p. Assuming that the 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]:

$$\alpha''_{i,p} = \frac{2(1-2\beta_{i,p})}{W(1-2\beta_{i,p}) + \beta_{i,p}(W+1)(1-(2\beta_{i,p})^L)}, \tag{4}$$

If $b_{i,p}$ is the average time spent in back-off, while node i is serving a path p packet, then the fraction of time that node

i spends in back-off during the service time $T_{i,p}$ is:

$$\psi_{i,p} = \frac{b_{i,p}}{E(T_{i,p})} \tag{5}$$

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) the average time spent in failed transmissions, which we denote by $f_{i,p}$. We have,

$$f_{i,p} = \frac{\epsilon_{i,p}}{\beta_{i,p}}\tau_P + (1 - \frac{\epsilon_{i,p}}{\beta_{i,p}})\tau_H \tag{6}$$

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. Recall that the RTS/CTS error can be due to both the PHY layer error or the MAC layer collision, while the data packet transmission failure is only due to the PHY layer error. The transmission times are,

$$\tau_H = T_{\mathrm{RTS}}(i,p) + \mathrm{SIFS} \tag{7}$$

$$\begin{aligned}\tau_P = {} & T_{\mathrm{RTS}}(i,p) + \mathrm{SIFS} + T_{\mathrm{CTS}}(h_{i,p},p) + \mathrm{SIFS} \\ & + T_{\mathrm{P}}(i,p) + \mathrm{SIFS}\end{aligned} \tag{8}$$

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

$$\begin{aligned}v_{i,p} &= (1-\beta_{i,p}^m)d_{i,p} + (\beta_{i,p} + \beta_{i,p}^2 + \cdots + \beta_{i,p}^m)f_{i,p} \\ &= (1-\beta_{i,p}^m)d_{i,p} + \frac{1-\beta_{i,p}^m}{1-\beta_{i,p}}\beta_{i,p}f_{i,p}\end{aligned} \tag{9}$$

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

$$\begin{aligned}d_{i,p} = {} & T_{\mathrm{RTS}}(i,p) + \mathrm{SIFS} + T_{\mathrm{CTS}}(h_{i,p},p) + \mathrm{SIFS} \\ & + T_{\mathrm{P}}(i,p) + \mathrm{SIFS} + T_{\mathrm{ACK}}(h_{i,p},p),\end{aligned} \tag{10}$$

and $T_{\mathrm{ACK}}(h_{i,p},p)$ is the time to send the ACK packet. The first term in (9) is the average time for successful transmission and the second term is the average time for failed transmissions (due to both PHY and MAC layer failures).

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,

$$\alpha_{i,p,j} = \rho_{i,p}(1-\theta_{i,j})\alpha''_{i,p} \quad \text{for all} j \in C_i \tag{11}$$

$$\alpha_{i,p,i} = \rho_{i,p}\alpha''_{i,p} \tag{12}$$

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

$$\theta_{i,j} = 1 - \prod_{n \in C_i \bigcap C_j^-} (1 - \sum_{p' \in P_n} \rho_{n,p'} \frac{v_{n,p'}}{E(T_{n,p'})}) \tag{13}$$

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

$$\begin{aligned}1-\beta_{i,p} = {} & \big(1 - l_{i,h_{i,p}}\big)\big(1-\theta_{h_{i,p},i}\big) \\ & \times \prod_{j \in C^+_{h_{i,p}} \bigcap C_i} (1 - \sum_{p' \in P_j} \alpha_{j,p',h_{i,p}}) \\ & \times \prod_{j \in C^+_{h_{i,p}} \bigcap C_i^-} (1 - \sum_{p' \in P_j} \alpha_{j,p',h_{i,p}})^{V_{i,p}}\end{aligned} \tag{14}$$

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. $l_{i,j}$ 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 transmissions 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, $V_{i,p} = T_{\mathrm{RTS}}(i,p) + \mathrm{SIFS}$.

$$V_{i,p} = T_{\mathrm{RTS}}(i,p) + \mathrm{SIFS}. \tag{15}$$

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.

3.3 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.

$$E(T_{i,p}) = (1-\beta_{i,p}^m)d_{i,p} + u_{i,p} + b_{i,p} + c_{i,p} \tag{16}$$

For the RTS/CTS mode of operation,

$$\begin{aligned}d_{i,p} = {} & T_{\mathrm{RTS}}(i,p) + \mathrm{SIFS} + T_{\mathrm{CTS}}(h_{i,p},p) + \mathrm{SIFS} \\ & + T_{\mathrm{P}}(i,p) + \mathrm{SIFS} + T_{\mathrm{ACK}}(h_{i,p},p)\end{aligned} \tag{17}$$

The average back-off time is

$$b_{i,p} = \sum_{n=0}^{m} W_n \beta_{i,p}^n, \tag{18}$$

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

The probability of successful transmission of node i, when it is scheduled to transmit path p packets is

$$q_{i,p} = \alpha''_{i,p}(1-\beta_{i,p}) \tag{19}$$

and the probability of successful transmission in the neighborhood of i, when it is scheduled to transmit path p packets is,

$$r_{i,p} = 1 - (1-q_{i,p}) \prod_{j \in C_i} \big(1 - (\sum_{p' \in P_j} q_{j,p'}\rho_{j,p'})(1-\theta_{ji})\big) \tag{20}$$

We assume that 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,

$$\gamma_{i,p} = \frac{q_{i,p}}{r_{i,p}} \tag{21}$$

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 node i neighbors. If we assume that $t_{k,i,p}$ and $Q_{i,p}$ are independent, we have

$$u_{i,p} = E(Q_{i,p})E(t_{k,i,p}) \tag{22}$$

The average number of successful transmissions is,

$$E(Q_{i,p}) = \frac{1-\gamma_{i,p}}{\gamma_{i,p}} \tag{23}$$

The probability that a successful transmission in the neighborhood of i belongs to a neighbor j, given that it does not belong to i, is

$$g_{j,i,p} = \frac{\sum_{p'\in P_j} q_{j,p'}\rho_{j,p'}(1-\theta_{j,i})}{r_{i,p}-q_{i,p}} \tag{24}$$

and

$$E(t_{k,i}) = \sum_{j\in C_i} g_{j,i,p}d_j \tag{25}$$

where

$$d_j = \frac{\sum_{p'\in P_j} k_{j,p'}d_{j,p'}(1-\beta^m_{j,p'})}{\sum_{p'\in P_j} k_{j,p'}(1-\beta^m_{j,p'})} \tag{26}$$

For $c_{i,p}$ we need to compute: (i) $x_{i,p}$, the probability of successful transmission of node i given that at least one transmission has occurred in the neighborhood of i, 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. If we define $z_{i,p}$ as

$$z_{i,p} = 1-(1-\alpha''_{i,p})\prod_{j\in C_i}\Big(1-(1-\theta_{j,i})\big(\sum_{p'\in P_j}\rho_{j,p'}\alpha''_{j,p'}\big)\Big) \tag{27}$$

then

$$x_{i,p} = \frac{q_{i,p}}{z_{i,p}} \quad \text{and} \quad y_{i,p} = 1-\frac{r_{i,p}}{z_{i,p}} \tag{28}$$

Then, the average number of collisions during $T_{i,p}$ is $y_{i,p}/x_{i,p}$ and the average collision time is

$$c_{i,p} = \frac{y_{i,p}}{x_{i,p}}w_{i,p} \tag{29}$$

where $w_{i,p}$ is the average time consumed for failure transmissions in the neighborhood of i:

$$w_{i,p} = \frac{\sum_{j\in C_i^+}\big(\sum_{p'\in P_j}\alpha''_{j,p'}\beta_{j,p'}\rho_{j,p'}\big)(1-\theta_{j,i})f_{j,p'}}{\sum_{j\in C_i^+}\big(\sum_{p'\in P_j}\alpha''_{j,p'}\beta_{j,p'}\rho_{j,p'}\big)(1-\theta_{j,i})} \tag{30}$$

4. THE ROUTING MODEL AND THE FIXED POINT IMPLEMENTATION

The routing model specifies a fixed set of paths and the fraction of incoming traffic that is sent over each path at the source node. Note that due to PHY and MAC layer loss parameters the incoming traffic rate at successive nodes of a path is a decreasing function. The incoming traffic rates of the nodes are derived from the scheduling and loss rates of their upstream links as follows:

$$\lambda_{h_{i,p},p} = k_{i,p}(1-\beta^m_{i,p}) \text{ for all } i,p. \tag{31}$$

The fixed point algorithm attempts to find a consistent solution for the three sets of equations given by the PHY and MAC layer, the routing and the scheduling models. 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, MAC, and the scheduling model equations provided in the previous sections to compute the scheduling rates $k_{i,p}$. Then, we use (31) 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, as shown in Fig. 2.

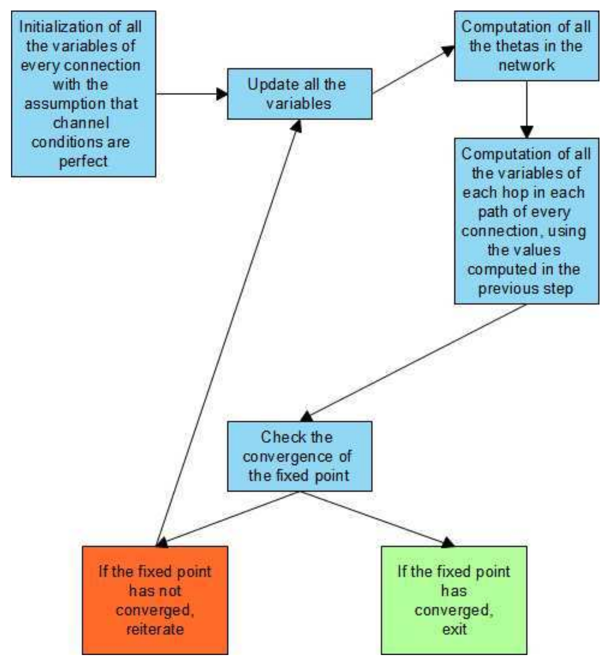


Figure 2: Architecture of the fixed point algorithm

In order to make the convergence of our fixed point algorithm faster, several choices of design have been made:

- **Initialization**: We initialize the values of our parameters assuming communication is perfect for every connection, i.e. the input rate of the paths are propagated all the way through the nodes with no loss, i.e., there is no transmission failure and collision in the path. We assume that the time $T_{i,p}$ consists only of the time taken by successful transmission, plus the back-off time needed for the first trial (we assume no retransmission is needed). Moreover, as we assume perfect channel conditions, every probability of failure is initialized to zero.

- **Use of memory in the fixed point**: To ensure convergence of the fixed point, we need to introduce the concept of memory when we update our results in our algorithm. This guarantees that the fixed point algorithm converges and there is no oscillation between multiple points. We introduce memory using the following method: To compute the new value of the parameter, we use two results: the value of the parameter that was saved in the previous iteration, which will be called *old*, and the value outputted by the result of the equation computing our parameter, called *new_equation*. We then define the new value as follows:

$$new\ value = \eta \cdot old + (1-\eta) new\_equation, 0 \leqslant \eta \leqslant 1 \tag{32}$$

- **Convergence condition**: The fixed point is exited when the expectation of the service time, $E[T_{i,p}]$, converges for all nodes in all path considered, during the same iteration. Note that the same node can have different values of $E[T_{i,p}]$ as each node can be involved in communications taking place in different paths, thus convergence happens if and only if all the expectation times of all nodes involved in an active connection in the network have converged. We choose the expectation of the service time as the variable on which to control the convergence of our fixed point iterations, as this variable depends on all the other parameters in our set of equations.

Upon convergence of the fixed point iterations, denoting $\lambda_{first,p}$ and $\lambda_{last,p}$ to be respectively the arrival rate of packets of the source or destination of path p, we define the throughput of a source-destination pair c to be:

$$T_c = \frac{\sum_{p \in P_c} \lambda_{last,p}}{\sum_{p \in P_c} \lambda_{first,p}} \tag{33}$$

5. AUTOMATIC DIFFERENTIATION FOR DESIGN

Although the fixed point algorithm can provide the basis for 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, 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.

In this section, we present the methodology employed to optimize the overall throughput in the network by changing the path probability distribution of each connection on the network. We denote by P_c the set of paths used in connection c and by C the set of all active connections in the network. The total network throughput T is:

$$T = \frac{\sum_{c \in C} (\sum_{p \in P_c} \lambda_{last,p})}{\sum_{c \in C} (\sum_{p \in P_c} \lambda_{first,p})} \tag{34}$$

Then, assuming there are $m = |C|$ active connections in the network, n_c paths used in the connection c and denoting by $\pi_{i,c}$ the probability associated with using path i in connection c, we know that the total throughput is a function of these input probabilities, namely:

$$T = T(\pi_{1,c_1}, \cdots, \pi_{n_{c_1},c_1}, \cdots, \pi_{n_{c_m},c_m}) \tag{35}$$

Thus, we can write our optimization problem in the following way:

$$\begin{gathered} max\ T = T(\pi_{1,c_1}, \cdots, \pi_{n_{c_1},c_1}, \cdots, \pi_{n_{c_m},c_m}) \\ \text{subject to} \quad \sum_{i \in P_c} \pi_{i,c} = 1, \forall c \in C \\ \pi_{i,c} \geqslant 0, \forall (i,c) \in P_c \times C \end{gathered} \tag{36}$$

To solve this optimization problem, we perform gradient projection, a solution particularly adapted to this problem as we need to compute the gradient of the throughput according to the input routing probabilities so as to be able to maximize it, and have to project the gradients obtained on the constraint space to get results respecting our conditions. Naming $\overline{\nabla}_c$ the average gradient obtained for connection c, we need to subtract that value from each of the gradients obtained for the paths in P_c to insure we still meet the constraint $\sum_{i \in P_c} \pi_{i,c} = 1$. In other words, at each iteration we update the set of routing probabilities using the following formula:

$$\pi_{i,c_k} = max\ (0, \pi_{i,c_k} + \beta(\frac{\delta T}{\delta \pi_{i,c_k}} - \overline{\nabla}_{c_k})), \forall k \in \{0, \cdots, m\} \quad (37)$$

Where in this formula, β is a parameter used to control the size of the steps taken during the update process of each iteration. This iteration is continued until for each connection, every path having a non-nil probability of being used has the same gradient than other non-nil paths of that same connection. Namely, the iteration stops when:

$$\forall \pi_{i,c} \neq 0 \in P_c, \frac{\delta T}{\delta \pi_{i,c}} = \overline{\nabla}_c, \forall c \in C. \quad (38)$$

Once we have reached convergence of that algorithm, we can now output a new set of results for our network configuration containing: a) The optimized throughput of each active connection in the network and b) The set of routing probabilities for each connection needed in order to achieve such throughput.

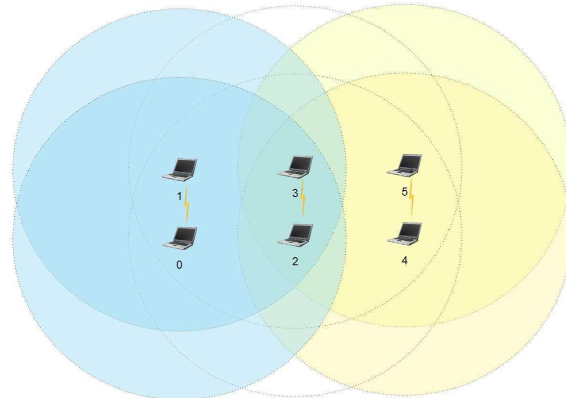
6. RESULTS

6.1 Starvation Models

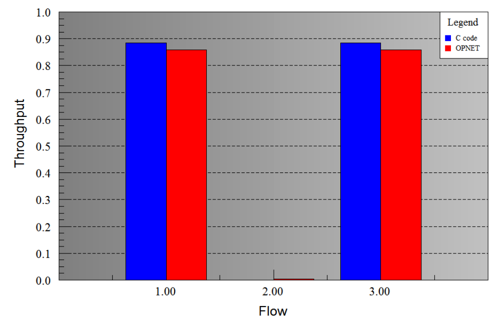
Multi-hop ad-hoc wireless networks are known to display unfairness in the throughput achieved by the different source-destination pairs in the network. In some cases, the discrepancies are such that they will induce the starvation of one flow while other flows in the network achieve an almost perfect throughput. These cases are known as *starvation models* [3]. In this subsection, we address two network topologies known to display unfairness in throughput distribution and test the validity of our model by comparing the throughputs computed by our fixed point to results obtained using a discrete event simulator, OPNET 12.0.

The first experiment considered is the Flow-in-the-Middle (FIM) scenario. The network is composed of 6 nodes and 3 links, as depicted in Fig 3(a). In this scenario, node 0 can hear that node 2 is transmitting, but it is not within hearing range of node 4. Symmetrically, node 4 can sense whether or not node 2 is sending data, but is not able to detect transmissions by node 0. On the other hand, node 2 can sense both node 0 and node 4 when they are transmitting on the channel. If all connections are backlogged, the middle flow will have a negligible throughput, while the two other flows will have an almost perfect throughput. Since node 0 and node 4 can not hear each other, they are not synchronized. Thus, their transmission will overlap randomly, and as node 2 senses both of them, it will sense the channel busy for its own transmissions most of the time. For node 2 to be able to transmit, nodes 0 and 4 have to be in their back-off stage simultaneously. To validate our model, we create such a topology in OPNET 12.0 and in our model to compare the results of both methods. As can be seen in Fig 3(b), our fixed point algorithm models this scenario very accurately.

The second starvation case is the Information Asymmetry (IA) scenario, represented in Fig 4(a). In this case, the sources of the two flows are not within hearing range of each other. The main problem in this scenario is that while S2 is aware of the presence of another flow in its neighborhood (it can sense the activity of D1), S1 has no knowledge of the fact that a communication affecting its transmission is



(a) Flow-in-the-Middle (FIM) scenario



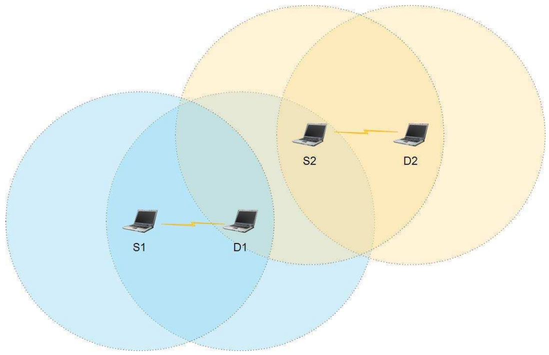
(b) Throughput for the different flows in the FIM model

Figure 3: Flow-in-the-Middle

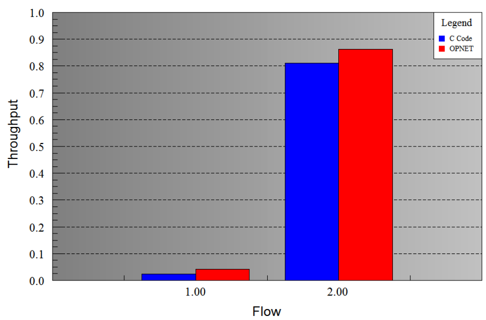
happening simultaneously in the vicinity. This means that flow 1 will not be able to fairly compete with flow 2: S2 will hear the CTS or ACK packets sent by D1, and thus will adapt by setting an accurate NAV, and will know when to contend for the channel for its own transmission. As S1 can not sense any activity of flow 2, it will have to request access to the channel in a random manner. This leads to S1 experiencing many unsuccessful attempts as D1 will not be able to correctly receive packets from S1 because of transmission from S2, forcing S1 to timeout and double its contention window. Thus, the packet loss probability for flow 1 will be very large, sometimes close to 100%. Ultimately, this will result in flow 1 having a much lower throughput than flow 2. Fig 4(b) shows the accurate modelling of this unfairness by our fixed point analysis as results match simulation results obtained with OPNET 12.0.

6.2 Multihop Connections: Throughput Approximation and Optimization

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 5(a). The blue nodes represent the wireless stations, the brown links the possible wireless connections between the nodes, and the pointed colored links the paths used in the three connections: (i) from node 3 to node 5, (ii) from node 17 to node 22 and (iii) from node 16 to node 1. Our routing

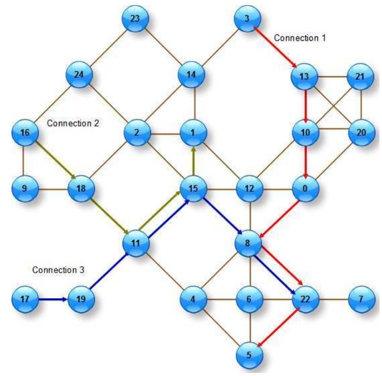


(a) Information Asymmetry scenario

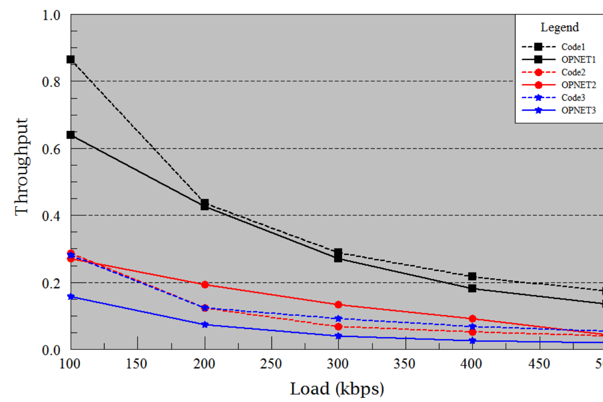


(b) Throughput comparison for Flow 1 and Flow 2 in the IA scenario

Figure 4: Information Asymmetry



(a) Three connections network.



(b) Comparison between OPNET and fixed point method.

Figure 5: Fixed point method.

Table 1: 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

algorithm finds the shortest paths between the source and destination nodes having nodes involved in different connections. Using these paths we then employ our set of fixed point equations to compute the throughput of these connections according to the desired load. As can be seen in Fig 5(b) the fixed point model results are close to the OPNET results.

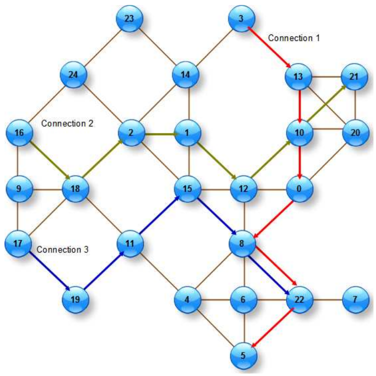
The main advantage of our fixed point model over discrete event simulation platforms, such as OPNET, is clearly the computation time. While our fixed point converges on the order of seconds, OPNET often requires several minutes to compute the throughput. This makes our model more suitable to compute approximations of throughput for network management and design which require fast and/or multiple simulations. Table 1 compares the time needed by our model and by OPNET as a function of the number of active connections in the network.

Next we use the fixed point model with AD to enhance the routing performance. Here we assume that a fixed set of paths are given and we want to tune the probabilities (portions) of sending traffic over the paths to maximize the throughput. We consider the network topology shown in Fig 6(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 6(a). We consider three alternative routings: (1) using shortest path only, (2) using all available paths with equal probability, and (3) using AD and gradient projection method to find the optimal probabilities. Fig 6(b) shows the network throughput v.s. the number of available paths for each one of the connections. The performance of the optimization algorithm improves as the number of available paths increases and it clearly outperforms other policies.

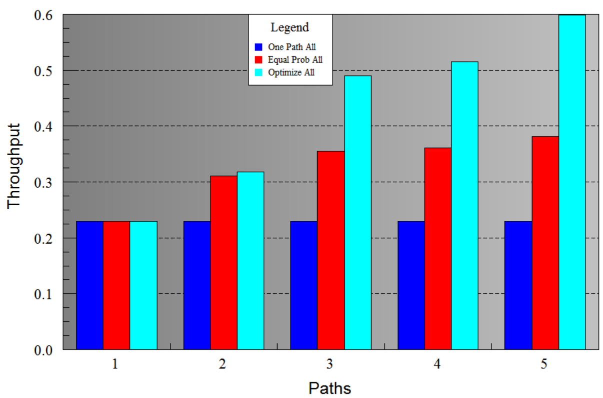
7. SUMMARY

In this paper, we introduce numerical methods and models for design and analysis of multi-hop wireless networks. In the analysis model, routing, scheduling, PHY and MAC layer are presented with a set of equations that are interdependent. For the MAC layer, we introduce a new set of equations that model the performance of an arbitrary multihop wireless network based on the 802.11 model.

A fixed point iteration is proposed to find a consistent solution for all equations and to form a fixed point implicit



(a) Three connections.



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

Figure 6: Sensitivity analysis.

model for the network performance. For design and optimization, we use Automatic Differentiation on top of the fixed point model to numerically compute the gradient of the performance metric (throughput) with respect to the design parameters (routing probabilities). Then, gradient projection method is used to compute routing probabilities that maximize the throughput.

The concept and methodology provided here can be generalized and extended to model alternative protocols and design criteria.

8. 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. Salonidis, 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 Networking 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.

[13] F. P. Kelly, A. K. Maulloo, and D. K. Tan, "Rate control for communication networks: Shadow prices, proportional fairness and stability," *Journal of Operations Research Society*, vol. 49, no. 3, pp. 237–252, Mar. 1998.

[14] F. P. Kelly, "Fairness and stability of end-to-end congestion control," *European Journal of Control*, vol. 9, no. 2-3, pp. 159–176, 2003.

[15] F. P. Kelly and T. Voice, "Stability of end-to-end algorithms for joint routing and rate control," *ACM SIGCOMM Computer Communication Review*, vol. 35, no. 2, pp. 5–12, Apr. 2005.

[16] F. P. Kelly and R. J. Williams, "Fluid model for a network operating under a fair bandwidth-sharing policy," *The Annals of Applied Probability*, vol. 14, no. 3, pp. 1055–1083, 2004.

[17] F. P. Kelly, "Scalable TCP: Improving performance in highspeed wide area networks," *ACM SIGCOMM Computer Communication Review*, vol. 33, no. 2, pp. 83–91, Apr. 2003.

[18] ——, "Loss networks," *The Annals of Applied Probability*, vol. 1, no. 3, pp. 319–378, Aug. 1991.

[19] ——, "Blocking probabilities in large circuit switched networks," *Advances in Applied Probability*, vol. 18,

no. 2, pp. 473–505, Jun. 1986.

[20] D. Mitra, J. A. Morrison, and K. G. Ramakrishnan, "ATM network design and optimization: A multirate loss network framework," *IEEE/ACM Trans. Netw.*, vol. 4, no. 4, pp. 531–543, Aug. 1996.

[21] S. Chung, A. Kashper, and K. W. Ross, "Computing approximate blocking probabilities for large loss networks with state-dependent routing," *IEEE/ACM Trans. Netw.*, vol. 1, no. 1, pp. 105–115, Feb. 1993.

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