A Delay Based Multipath Optimal Route Analysis for Multi-hop CSMA/CA Wireless Mesh Networks

Jiazhen Zhou and Kenneth Mitchell

Dept. of Computer Science Electrical Engineering
University of Missouri - Kansas City, Kansas City, MO, 64110
{jzmpc, mitchellke}@umkc.edu

Abstract. In this paper we present a method for determining optimal routes along selected paths in a wireless mesh network based on an interference aware delay analysis. We develop an analytic model that enables us to obtain closed form expressions for delay in terms of multipath routing variables. A flow deviation algorithm is used to derive the optimal flow over a given set of routes. The model takes into account the effects of neighbor interference and hidden terminals, and tools are provided to make it feasible for the performance analysis and optimization of large-scale networks. Numerical results are presented for different network topologies and compared with simulation studies.

1 Introduction

Wireless mesh networks are multi-hop access networks used to extend the coverage range of current wireless networks [1]. They are composed of mesh routers and mesh clients, and generally require a gateway to access backhaul links. Access to the medium is either centrally controlled by the base station or distributed, typically using some form of CSMA/CA protocol.

As pointed out by Tobagi [2], the exact throughput-delay analysis for multihop networks requires a large state space. For large topologies, an exact analysis is almost impossible, leading us to consider an approximate analysis. Similar work by Leiner [3] and Silvester and Lee [4], assume that frames are independently generated at each node for transmission to a neighbor, whereby the amount of traffic generated is a function of the topology, routing, and end-to-end traffic requirements. They also develop a model of the neighborhood around a node and characterize it with a number of parameters representing average behavior. Parameters for all nodes are then found through an iterative process. However, in Leiner's work, single-hop models are used for the neighborhood around each node, which means that all of the interfering nodes of a certain node interfere with each other. This makes the model relatively simple, but generally not applicable for most multi-hop networks.

In Boorstyn et al. [5], large networks are decomposed into smaller groups and Markov chains are constructed for those nodes that can transmit simultaneously.

The throughput of the network is then studied, based on the assumption of Poisson arrivals. The algorithm is iterative, and due to the need to compute all independent sets in the network, the complexity of the algorithm is prohibitive. Wang and Kar [6] present work based on the same architecture as Boorstyn, while paying special attention to the optimal min/max fairness and throughput with the RTS/CTS mechanism.

Other than the node-group based method proposed by Boorstyn, single node or flow based methods have also been used in recent research and are generally scalable to large networks. Carvalho and Garcia-Luna-Aceves [7] present a model that takes into consideration the effects of physical layer parameters, MAC protocol, and connectivity. They mainly focus on the throughput of nodes for the saturated case. Garetto, Salonidis, and Knightly [8] address fairness and starvation issues by using a single node view of the network that identifies dominating and starving flows and accurately predicts per-flow throughput in a large-scale network.

We can see that most recent studies mainly deal with throughput and fairness issues. In our work, the model we build is not only suitable for throughput analysis, but also for delay analysis. Based on our closed form solution for delay, multi-path route optimization becomes possible. The analytical model we introduce is based on a single node analysis. Interfering nodes and hidden terminals are taken into account when computing the probability that a node successfully transmits frames.

The rest of this paper is organized as follows: In section 2 we describe the basic model and exploit the neighbor relationships to derive solutions using iterative algorithms. In section 3, the closed form representation of delay at each node is derived and a corresponding optimization model is introduced. Examples using our method for the analysis and optimization of wireless mesh networks are shown in section 4, and section 5 concludes this paper.

2 Basic Model

Similar to work presented in [6] and [8], our model is based on a generic carrier sense multiple access protocol with collision avoidance (CSMA/CA). We generalize on the work of Kleinrock and Tobagi [9, 10] and Boorstyn et al. [5] to include a finite number of nodes, multiple hops, and interference caused by routing. Nodes having frames to transmit can access the network if the medium is idle. If the medium is detected as being busy, a node will reattempt to access the medium after a specified time interval. We assume that there is some mechanism (such as RTS/CTS in the 802.11 standard) that allows the node to determine if the medium is available or if it must wait and reattempt access to the channel. We use a nodal decomposition method that relies on an iterative process to determine the probability that a transmission attempt is successful.

We assume that messages at each node i are generated according to a Poisson distribution with mean rate λ_i. All message transmission times are exponentially distributed with mean $1/\mu$. Likewise, the channel capacity is taken to be μ. We

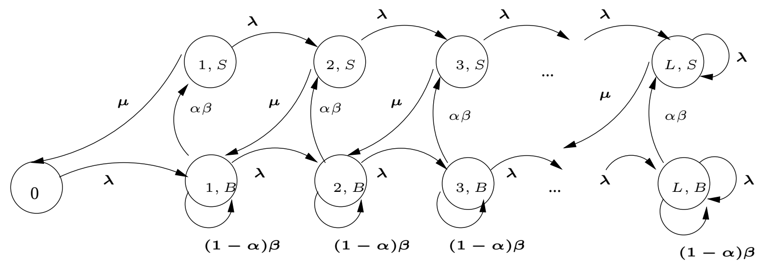

Fig. 1. Markov chain diagram of a single node.

assume an ideal collision avoidance mechanism that can always detect if the medium is busy or free at the end of a transmission attempt waiting period. All waiting periods between transmission attempts (backoff periods) are exponentially distributed with mean $1/\beta$, resulting in a geometrically distributed number of backoff attempts (see Cali et al. [11]). An infinite number of backoff periods are possible. Each node backs off after a successful transmission to ensure fairness. The probability that node i finds the medium free and is able to successfully transmit a message is denoted as α_i. If node A interferes with node B, then node B also interferes with node A (symmetrical transmission range.) All successfully transmitted frames are received error free.

In multi-hop networks, some nodes directly interfere with each other and some indirectly interfere (hidden terminal problem [10].) Those nodes that directly interfere or are hidden terminals to each other cannot send messages at the same time. We refer to these nodes as "neighbors" in this paper and introduce a "neighbor matrix", $\boldsymbol{N}$, in section 2.2, to derive these relationships.

Fig. 1 depicts the queueing model for a single node. For each state (l, S) or (l, B), S means the node is sending (transmitting), B means it's backing off, l represents the number of messages waiting in the queue, and L is the queue length. State 0 means there is no frame at this node, so the node is in idle state. This is an M/G/1/L model from which the steady state, busy probability, blocking probability, etc. can be easily derived [12]. Strictly speaking, for internal nodes in the network that relay messages, the arrivals from different sources can be correlated with each other, so the total arrival stream will not be Poisson. However, the assumptions we make allow us use the M/G/1/L model, which produces results that are extremely close to simulation.

2.1 Calculating successful transmission probabilities

We have defined α_i as the probability that node i successfully accesses the medium during a transmission attempt, so α_i is a *statistical view* of the medium being idle when node i has a frame to send. Now consider the state of the medium in the region around node i. There are three possible states for node i: 1) being "idle", with probability $P_I[i]$, 2) being in "sending" state, with probability $P_S[i]$, and 3) being in "backoff" state, with probability $P_B[i]$. When a node is transmitting frames, we denote this node as being in its "sending" state. Let ρ_i be the queuing system utilization of node i, which means this node is either in its "backoff" or "sending" state, so $\rho_i = P_S[i] + P_B[i]$. Only when it is in "backoff", will node i sense the medium (attempt to transmit). The corresponding probability is $\rho_i - P_S[i]$. To make sure node i's attempt is successful, no neighbor of node i can be sending at that moment, so the probability of a successful attempt is $\rho_i - P_S[i] - \rho_i \cup_{k\in\omega_i} P_S[k]$, where ω_i represents all nodes that are neighbors of node i, and $\cup_{k\in\omega_i} P_S[k]$ represents the "sending" probability of neighbors as viewed by node i.

The parameter α_i can be interpreted as the probability that node i transmits successfully given that it *attempts* to do so.

$$\alpha_i = \frac{\rho_i - P_S[i] - \rho_i \cup_{k\in\omega_i} P_S[k]}{\rho_i - P_S[i]} = \frac{1 - P_S[i]/\rho_i - \cup_{k\in\omega_i} P_S[k]}{1 - P_S[i]/\rho_i}. \quad (1)$$

The value of α_i is determined by the "sending" probability of node i itself and its neighbors $k \in \omega_i$. Likewise, each neighbor k will have node i as its neighbor, and its successful transmission probabilities will depend on node i. Therefore, we need use an iterative method to find the value of α_i.

In order to compute $\cup_{k\in\omega_i} P_S[k]$ (the medium busy probabilities as seen by node i), we need to solve several problems first: Which nodes will prevent node i from sending? Will all the sending times of neighboring nodes k ($k \in \omega_i$) be mutually exclusive? If not, how should we decide the possible nodes that can transmit simultaneously? (We call them *simultaneously transmitting nodes.*) How do we calculate the corresponding simultaneous transmitting probability? In the following sections we will introduce ways to solve the above problems.

2.2 Neighbor matrix

As mentioned by Jain et al. [13], an interference matrix, $\boldsymbol{F}$, can be easily configured based on the interference relationship between nodes. However, deriving hidden terminal relationships is not provided in their paper. Here we provide a way to identify the hidden terminal relationship based on the known routing information and the interference relationship. The hidden terminal relationship and the direct interference relationship will be combined into a "neighbor matrix", $\boldsymbol{N}$.

We define a binary routing matrix $\boldsymbol{R}$ to represent the routing relationship. Denote $\boldsymbol{R}_{ij} = 1$ if node i sends messages to j, otherwise $\boldsymbol{R}_{ij} = 0$. In the interference matrix $\boldsymbol{F}$, if node i and j are interfering with each other, we denote $\boldsymbol{F}_{ij} = 1$ and $\boldsymbol{F}_{ji} = 1$, otherwise $\boldsymbol{F}_{ij} = 0$ and $\boldsymbol{F}_{ji}$=0. The algorithm to derive the neighbor matrix is shown below, (note that all Multiply and Add operations are Boolean algebra operations.)

Algorithm 1: Neighbor matrix

Step 1: *Generate the hidden terminal relationship:* Multiply $\boldsymbol{R}$ by $\boldsymbol{F}$ to get a new matrix $\boldsymbol{H} = \boldsymbol{R}\boldsymbol{F}$. The hidden terminal information is already embedded in

$\boldsymbol{H}$, since if node i and j are hidden terminals to each other, there must exist one or more nodes k such that $\boldsymbol{R_{ik}} = 1$ (node i wishes to talk to node k) and $\boldsymbol{F_{kj}} = 1$ (node k and node j interferes with each other), so $\boldsymbol{H_{ij}} = \sum_k \boldsymbol{R_{ik}} \boldsymbol{F_{kj}} = 1$;

Step 2: *Combine the hidden terminal relationship with the direct interference relationship:* let $\boldsymbol{Y} = \boldsymbol{X} + \boldsymbol{F}$;

Step 3: *Remove the self-neighbor relationship:* Change all of the diagonal elements of $\boldsymbol{Y}$ to 0 (a node is not considered a neighbor to itself). The resulting matrix is the neighbor matrix $\boldsymbol{N}$. This matrix incorporates both the interference relationship and the hidden terminal relationship. Note that $\boldsymbol{N_{ij}} = 1$ means that node i and j are "neighbors" to each other; $\boldsymbol{N_{ij}} = 0$ means that they are not "neighbors", allowing them to transmit simultaneously. Since the interference relationships and the hidden terminal relationships are both symmetrical, $\boldsymbol{N}$ is a symmetrical matrix. We can now define ω_i in equation (1) as the set of nodes represented by 1's in the ith row of the neighbor matrix $\boldsymbol{N}$.

2.3 Simultaneously transmitting nodes

There may be nodes that are neighbors to node i that are neither hidden terminals nor directly interfering nodes with each other. Thus, the probability that two or more nodes can send messages simultaneously (they do not interfere with each other, but do interfere with node i) is very important information for calculating the medium busy probability around node i, which we defined as $\cup_{k \in \omega_i} P_S[k]$.

When there are 2 nodes that can transmit simultaneously, we call the set of those nodes *simultaneously transmitting pairs* denoted as STS_2; when there are 3 or more nodes that can transmit simultaneously, we call the set of those nodes *simultaneously transmitting groups* and denote them as $STS_3, STS_4 ....$ We define "group degree" as the number of nodes that can transmit simultaneously.

Algorithm 2: Simultaneously transmitting pairs/groups

Step 1: Take the complementary set of the neighbor matrix $\boldsymbol{N}$ to identify the simultaneously transmitting pairs. Since this matrix describes the relationship between any *two* nodes, we denote it as $\boldsymbol{S}_2$, so $\boldsymbol{S}_2 = \overline{\boldsymbol{N}}$.

Step 2: For all node pairs (i, j) such that $\boldsymbol{S_{2,ij}} = 1$ (to avoid the duplication, we just consider the upper diagonal part of $\boldsymbol{S_2}$), list all possible nodes k (different from i, j) such that $\boldsymbol{S_{2,ik}} = 1$ and $\boldsymbol{S_{2,kj}} = 1$, and put all valid node groups (i, j, k) into set STS_3.

Step 3: For each node group (i, j, k) in STS_3, find all possible nodes m such that $\boldsymbol{S_{2,im}} = 1$, $\boldsymbol{S_{2,jm}} = 1$ and $\boldsymbol{S_{2,km}} = 1$, and put all valid node groups (i, j, k, m) into STS_4. The above process continues until we reach n such that no n nodes can transmit simultaneously.

2.4 Simultaneous transmitting probabilities

The busy probability of the medium around each node i in the multi-hop environment can be calculated as

$$\cup_{k \in \omega_i} P_S[k] = \sum_{k \in \omega_i} P_S[k] - \sum_{(k_1,k_2) \in STS_2} P_S[k_1 k_2] + \sum_{(k_1,k_2,k_3) \in STS_3} P_S[k_1 k_2 k_3] - \ldots, \quad (2)$$

where $k_1, k_2, k_3... \in \omega_i$. Now we need to calculate the overlapped sending probabilities $P_S[k_1 k_2], P_S[k_1 k_2 k_3] \ldots$.

For two nodes that are not neighbors to each other, if they also don't have shared neighbors, we assume that they can independently transmit; if they have shared neighbors, they are independent only during the period when no messages are being transmitted to or from the shared neighbors. In the latter case, these nodes can be viewed as "conditionally independent".

The neighbors of node k_1 will be $\omega_{k_1} = \{q : \boldsymbol{N}_{k_1 q} = 1\}$, and the neighbors of node k_2 is $\omega_{k_2} = \{q : \boldsymbol{N}_{k_2 q} = 1\}$. Denote $\omega_{k_1 k_2} = \omega_{k_1} \cup \omega_{k_2}$. When both node k_1 and k_2 are sending, none of the nodes in $\omega_{k_1 k_2}$ can be sending.

$$P_S[k_1, k_2] = P_S[k_1, k_2, \overline{\omega_{k_1 k_2}}] = P_S[k_1, k_2 | \overline{\omega_{k_1 k_2}}] P_S[\overline{\omega_{k_1 k_2}}]. \tag{3}$$

Since nodes k_1, k_2 are independent conditioned on the probability that none of the nodes in $\omega_{k_1 k_2}$ are sending, we have

$$P_S[k_1, k_2 | \overline{\omega_{k_1 k_2}}] = P_S[k_1 | \overline{\omega_{k_1 k_2}}] P_S[k_2 | \overline{\omega_{k_1 k_2}}] = \frac{P_S[k_1, \overline{\omega_{k_1 k_2}}]}{P_S[\overline{\omega_{k_1 k_2}}]} \frac{P_S[k_2, \overline{\omega_{k_1 k_2}}]}{P_S[\overline{\omega_{k_1 k_2}}]}. \tag{4}$$

$P_S[\overline{\omega_{k_1 k_2}}]$ represents the probability that no neighbor of node k_1, k_2 is sending, which can be written as $1 - P_S[\omega_{k_1 k_2}]$ instead. $P_S[k_1, \overline{\omega_{k_1 k_2}}]$ represents the probability that node k_1 is sending while all the neighbors of node k_1, k_2 are not sending. As we know, neighbors of node k_1 must not be sending when node k_1 is sending, if we denote $\omega_{k_2 \overline{k_1}}$ as the nodes that are neighbors of node k_2 but not of node k_1, we have $P_S[k_1, \overline{\omega_{k_1 k_2}}] = P_S[k_1, \overline{\omega_{k_2 \overline{k_1}}}] = P_S[k_1] - P_S[k_1, \omega_{k_2 \overline{k_1}}]$. Similarly we can get $P_S[k_2, \overline{\omega_{k_1 k_2}}] = P_S[k_2] - P_S[k_2, \omega_{k_1 \overline{k_2}}]$.

After combining equation (3), (4), and the computation for $P_S[k_1, \overline{\omega_{k_1 k_2}}]$, $P_S[k_1, \overline{\omega_{k_1 k_2}}]$, and $P_S[\overline{\omega_{k_1 k_2}}]$, the resulting expression is

$$P_S[k_1, k_2] = \frac{(P_S[k_1] - P_S[k_1, \omega_{k_2 \overline{k_1}}])(P_S[k_2] - P_S[k_2, \omega_{k_1 \overline{k_2}}])}{1 - P_S[\omega_{k_1 k_2}]}. \tag{5}$$

The calculation of $P_S[\omega_{k_1 k_2}], P_S[k_1, \omega_{k_2 \overline{k_1}}], P_S[k_2, \omega_{k_1 \overline{k_2}}]$ can be done similarly by using equation (2). We can get the exact solution by solving the system of equations, or by using iterative methods. After we get $P_S[k_1, k_2]$, $P_S[k_1, k_2, k_3]$ etc. can be computed similarly.

3 Path Delays and Optimization

If we make the assumption that the queue length is infinite and thus no loss occurs, we can get closed form solutions for α_i, which makes system optimization possible.

The service time distribution at each node consists of both the transmission time and the queueing delay (waiting time when frames ahead are transmitting or the node is in "backoff" state). It has a matrix exponential distribution representation

$$F(t) = 1 - \boldsymbol{p} \exp(-\boldsymbol{B} t) \boldsymbol{e}', \tag{6}$$

where $\boldsymbol{p}$ is the starting vector for the process, $\boldsymbol{B}$ is the progress rate operator for the process, and $\boldsymbol{e}'$ is a summing operator consisting of all 1's [12]. The moments of the matrix exponential distribution are

$$E[X^n] = n!\boldsymbol{p}\boldsymbol{B}^{-n}\boldsymbol{e}'. \tag{7}$$

Based on the Markov chain of Fig. 1, the matrix exponential representation of the service distribution at each node i is

$$\boldsymbol{p} = \begin{bmatrix} 1 \; 0 \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} \beta\,\alpha_i & -\beta\,\alpha_i \\ 0 & \mu \end{bmatrix}. \tag{8}$$

Using equation (7), the mean and the second moment of the of the service distribution at node i are

$$E[S_i] = \frac{1}{\mu} + \frac{1}{\alpha_i\,\beta} = \frac{\mu + \alpha_i\,\beta}{\alpha_i\,\beta\,\mu}, \qquad E[S_i^2] = 2\,\frac{\mu^2 + \alpha_i\,\beta\,\mu + {\alpha_i}^2\beta^2}{{\alpha_i}^2\beta^2\mu^2}. \tag{9}$$

When the queue is infinite, $\rho_i = \lambda_i \frac{\mu+\alpha_i\,\beta}{\alpha_i\,\beta\,\mu}$, where λ_i is the mean arrival rate to node i. Also, since there is no loss, the "sending" probability will be λ_i/μ (percentage of total channel capacity that node i is using). Substituting ρ_i and $P_S[i]$ into equation (1) and solving for α_i, we get

$$\alpha_i = \frac{\mu(1 - \cup_{k\in\omega_i} P_S[k])}{\mu + \beta \cup_{k\in\omega_i} P_S[k]} = \frac{1 - \cup_{k\in\omega_i} P_S[k]}{1 + \beta/\mu \cup_{k\in\omega_i} P_S[k]}. \tag{10}$$

Since $P_S[k] = \lambda_k/\mu$, α_i can now be represented in terms of λ_k – the arrival rate of each node.

Using the P-K formula for M/G/1 queues, the mean waiting time in the queue at each node is $E[W] = \frac{\lambda E[S^2]}{2(1-\lambda E[S])}$. By substituting the expressions for the mean and second moment of the service times and noting that the mean total time spent at node i is $E[T_i] = E[W_i] + E[S_i]$, we get

$$E[T_i] = \frac{\mu + \alpha_i\,\beta - \lambda_i}{\alpha_i\,\beta\,\mu - \lambda_i\,\mu - \lambda_i\,\alpha_i\,\beta}. \tag{11}$$

To express the delay as an optimization problem, we use the following notation:

$\mathcal{K}$	Set of all origin-destination nodes that have traffic.
$\mathcal{I}$	Set of communicating nodes in the network.
Λ_k	Average arrival rate for origin destination pair k.
Λ	Total arrival rate to the network, $\Lambda = \sum_{k\in\mathcal{K}} \Lambda_k$.
$\mathcal{P}_k$	Set of possible paths for o-d pair k.
λ_{kj}	Amount of flow on path j for pair k.
α_i	Transmission success probability at node i, which is expressed as a function of the path flow variables λ_{kj} using equation (10).

δ^i_{kj}	Node path indicator: 1 if path j for pair k passes through node i.
F_i	Total flow through node i, $F_i = \sum_{k\in\mathcal{K}} \sum_{j\in\mathcal{P}_k} \delta^i_{kj}\lambda_{kj}$.

The optimization problem for minimizing the mean delay a frame experiences in the network is

$$\min_{\lambda_{kj},F_i} \frac{1}{\Lambda} \sum_{i\in\mathcal{I}} F_i \frac{F_i - \mu - \alpha_i\beta}{-\alpha_i\beta\mu + F_i\mu + F_i\,\alpha_i\beta}, \tag{12}$$

such that

$$\sum_{j\in\mathcal{P}_k} \lambda_{kj} = \Lambda_k, \quad k \in \mathcal{K}, \tag{13}$$

$$\sum_{k\in\mathcal{K}} \sum_{j\in\mathcal{P}_k} \delta^i_{kj}\lambda_{kj} - F_i = 0, \quad i \in \mathcal{I}, \tag{14}$$

$$\lambda_{kj} \geq 0,\ F_i \geq 0. \tag{15}$$

The objective function is rational, with polynomials in both the numerator and denominator. The constraints are linear, so we use a flow deviation algorithm using the projection method [14] to solve this problem. Convergence is very fast for the examples we present under the assumption that the network is stable and the starting point is feasible. Routing strategies based on optimal path delays and QoS requirements for different traffic classes can also be constructed, but here we only present the case for mean delay.

4 Numerical Results

4.1 Simulation model

We use CSIM simulation tools to construct the simulation model. If a node has a frame to transmit, it will first wait one backoff period which is exponentially distributed with mean $1/\beta$. Upon completion of the backoff period, the node initiates an RTS to see if the medium is available. We use the same assumptions in the simulation as in the analytic model, namely, that RTS/CTS communication is instantaneous and that there are no errors. If the channel is not available, the node will go into backoff, otherwise the frame is transmitted with a mean time of $1/\mu$. Frames are forwarded based on the route indicated in the frame header.

4.2 Evaluation of wireless mesh networks

In this subsection we show the effectiveness of our model by comparing analytical results to simulation. In the scenarios we show, we assume the maximum transmission rate is 10 Mbps and the average frame size is 1250 bytes ($10,000$ bits), resulting in a mean transmission rate of $\mu = 1000$ frames per second (fps).

For the multi-hop mesh topology depicted in Fig. 2, we show a simple example where both node 1 and 2 have frames to send through the gateway to the

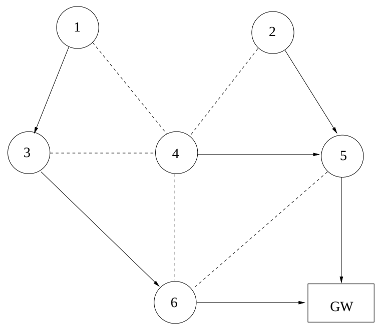

Fig. 2. Six node mesh network.

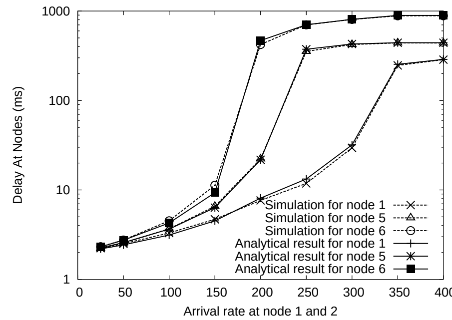

Fig. 3. Comparison of the delay.

wired network and all other nodes have no frames to send. We use solid lines with arrows to specify the routing relationship between nodes and dashed lines between nodes that directly interfere with each other. No lines between nodes means that they will not interfere directly with each other. For convenience, we assume node 1 and node 2 have the same arrival rate. We denote the gateway, GW, as node 7. Also, since the gateway does not send messages upstream on the same channel, we can omit the direct relationships in the neighbor matrix. Using the algorithm described in section 2.2, we get

$$\boldsymbol{R} = \begin{bmatrix} 0\,0\,1\,0\,0\,0\,0 \\ 0\,0\,0\,0\,1\,0\,0 \\ 0\,0\,0\,0\,0\,1\,0 \\ 0\,0\,0\,0\,1\,0\,0 \\ 0\,0\,0\,0\,0\,0\,1 \\ 0\,0\,0\,0\,0\,0\,1 \\ 0\,0\,0\,0\,0\,0\,0 \end{bmatrix}, \quad \boldsymbol{F} = \begin{bmatrix} 0\,0\,1\,1\,0\,0\,0 \\ 0\,0\,0\,1\,1\,0\,0 \\ 1\,0\,0\,1\,0\,1\,0 \\ 1\,1\,1\,0\,1\,1\,0 \\ 0\,1\,0\,1\,0\,1\,1 \\ 0\,0\,1\,1\,1\,0\,1 \\ 0\,0\,0\,0\,1\,1\,0 \end{bmatrix}, \text{ and } \boldsymbol{N} = \begin{bmatrix} 0\,0\,1\,1\,0\,1\,0 \\ 0\,0\,0\,1\,1\,1\,1 \\ 1\,0\,0\,1\,1\,1\,1 \\ 1\,1\,1\,0\,1\,1\,1 \\ 0\,1\,0\,1\,0\,1\,1 \\ 1\,1\,1\,1\,1\,0\,1 \\ 0\,0\,0\,0\,1\,1\,0 \end{bmatrix}.$$

The comparison of analytical results and simulation results are shown in Figs. 3 - 5. Since nodes 1 and 2 have the same offered load and interference, they exhibit the same behavior. Therefore, we just show the performance for node 1. For the same reason, we only show performance measures for node 5 and not node 3.

In Figs. 3, 4, and 5, we show the delay, mean backoff times, and blocking probabilities at various load for nodes having a finite capacity of size $L = 100$. We can see that for low load and heavy load, the analytical results match perfectly with the simulation results, while for moderate load, there is a slight difference. In total, the analytical results are very good at catching the abrupt increase in delay as the load increases.

A more complicated scenario is shown in Fig. 6. There is one gateway, nodes 1-5 are actively sending messages, and the other nodes are acting as mesh routers. The buffer size at each node is 100.

The comparison of simulation and analytical results for the delay at nodes 1, 6, 8, and 9 are shown in Fig. 7 and blocking probabilities are shown in Fig. 8.

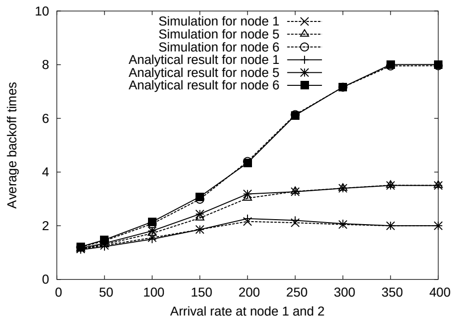

Fig. 4. Average backoff times at each node.

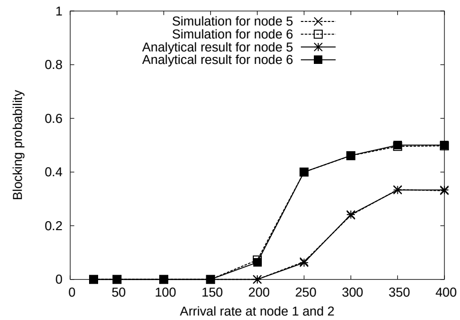

Fig. 5. Blocking at each node.

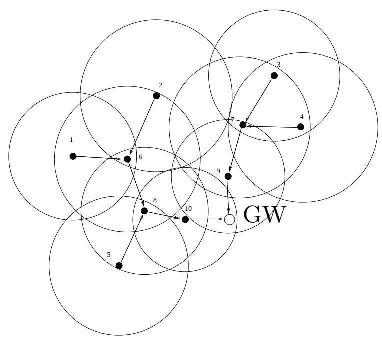

Fig. 6. Ten node multi-hop network.

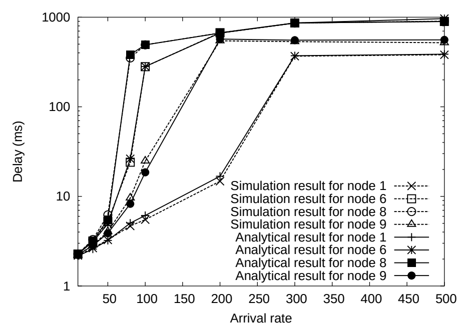

Fig. 7. Delay at nodes 1, 6, 8, and 9.

We can see that node 6 and node 8 are the bottlenecks in this network. Note that the results are accurate over a wide variety of offered loads.

4.3 Optimization results

In a wireless mesh network, nodes can also communicate in ad-hoc mode, which means that they can transmit frames to peer nodes through other intermediate nodes. Referring to the topology in Fig. 2, we assume that there are two communicating pairs. Node 2 is sending to node 6 and node 3 is sending to node 5. The paths available to node 2 are path $2, 4, 6$ and $2, 5, 6$ and the paths available to node 3 are $3, 4, 5$ and $3, 6, 5$. The mean arrival rate at node 2 and 3 are denoted as λ_2 and λ_3 respectively. The path flow variables are denoted as λ_{246}, λ_{256}, λ_{345}, and λ_{365}. The constraints are $\lambda_{256} = \lambda_2 - \lambda_{246}$ and $\lambda_{365} = \lambda_3 - \lambda_{345}$. The load at node 4 is $F_4 = \lambda_{345} + \lambda_{246}$, the load at node 5 is $F_5 = \lambda_{256}$, and at node 6 the load is $F_6 = \lambda_{365}$. Solving the optimization problem (equations (12), (13), (14), and (15)), we get the optimal routing for different values of λ_2 and λ_3. We can see that the system delay function is convex, as shown in Fig. 9. We set $\lambda_2 = 200$ fps and $\lambda_3 = 230$ fps.

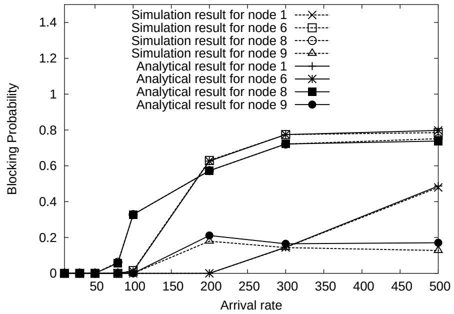

Fig. 8. Blocking at nodes 1, 6, 8, and 9.

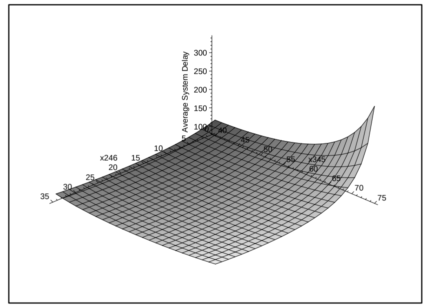

Fig. 9. Convex cost function.

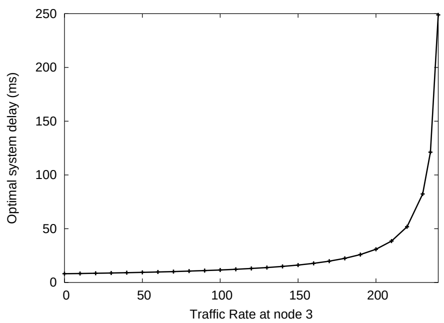

Fig. 10. Optimal system delay vs. load.

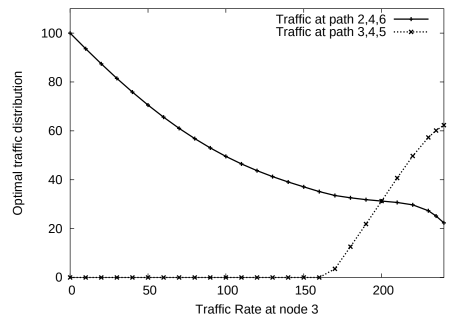

Fig. 11. Traffic distribution vs. load.

To show the effect of one node's traffic on the choice of routes, we set the traffic at node 2 fixed at 200 fps, and then watch the changes of routing and traffic distribution according to different traffic from node 3. In Fig. 10, we show the optimal system delay for different loads at node 3, and Fig. 11 shows the resulting traffic sent through path $2, 4, 6$ and $3, 4, 5$. We can see that when the traffic originating from node 3 is low, all segments will take path $3, 6, 5$ because node 4 is heavily interfered and will result in high delay. But traffic from node 2 will take advantage of route $2, 4, 6$ since the traffic from node 2 is already high. As the load at node 3 increases, part of traffic from node 3 will take path $3, 4, 5$ and the volume will keep increasing. At the same time, traffic at path $2, 4, 6$ will decrease due to the stronger interference at node 4.

5 Conclusion

In this paper, the neighbor concept is extended to incorporate both directly interfering nodes and hidden terminals of each node based on the topology and routing in the network. Based on the relationships of "neighbors", we use a node based analysis where an iterative process is used to find the probability of a successful transmission at each node. To facilitate neighbor identification, identifying algorithms are provided. For the infinite buffer case, we derive a means to identify the optimal multipath flow that minimizes the mean delay in the network. The comparison of simulation and analytical results show that our analytical method is accurate under both saturated and unsaturated cases.

The evaluation of wireless mesh networks shows that the system performance is sensitive to the number of interfering neighbors and route selection. For future work, we plan on adding cost functions to the multipath optimization to insure QoS fairness for different classes of traffic at each node.

References

1. I. Akyildiz, X. Wang, and W. Wang, "Wireless mesh networks: a survey," *Journal of Computer Networks*, vol. 47, pp. 455–487, 2005.
2. F. A. Tobagi, "Modeling and performance analysis of multihop packet radio networks," *Proceedings of the IEEE*, vol. 75, no. 1, pp. 135–155, Jan. 87.
3. B. M. Leiner, "A simple model for computation of packet radio network communication performance," *IEEE Transactions on Communications*, vol. 28, no. 12, pp. 2020–2023, Dec. 1980.
4. A. Silvester and I. Lee, "Performance modeling of buffered csma-an iterative approach," in *Proceedings of the IEEE Conference on Global Communications (GLOBECOM)*, Miami, FL, 1982.
5. R. Boorstyn, A. Kershenbaum, B. Maglaris, and V. Sahin, "Throughput analysis in multihop CSMA packet radio networks," *IEEE Transactions on Communications*, vol. COM-35, no. 3, pp. 267–274, Mar. 1987.
6. X. Wang and K. Kar, "Throughput modelling and fairness issues in CSMA/CA based ad-hoc networks," in *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, Miami, FL, 2005.
7. M. Carvalho and J. J. Garcia-Luna-Aceves, "A scalable model for channel access protocols in multihop ad hoc networks," in *MOBICOM*, Philadelphia, PA, 2004.
8. T. S. M. Garetto and E. Knightly, "Modeling per-flow throughput and capturing starvation in CSMA multi-hop wireless networks," in *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, Barcelona, Spain, 2006.
9. L. Kleinrock and F. Tobagi, "Packet switching in radio channels: Part I-carrier sense multiple-access modes and their throughput-delay characteristics," *IEEE Transactions on Communications*, vol. 23, no. 12, pp. 1400–1416, 1975.
10. F. Tobagi and L. Kleinrock, "Packet switching in radio channels: Part II- the hidden terminal problem in carrier sense multiple-access and the busy-tone solution," *IEEE Transactions on Communications*, vol. 23, no. 12, pp. 1417–1433, Dec. 1975.
11. F. Cali, M. Conti, and E. Gregori, "IEEE 802.11 wireless LAN: Capacity analysis and protocol enhancement," in *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, San Francisco, Mar. 1998.
12. L. Lipsky, *Queueing Theory - A Linear Algebraic Approach*. Macmillan Publishing Company, 1992.
13. K. Jain, J. Padhye, V.Padmanabhan, and L. Qiu, "Impact of interference on multihop wireless network performance," in *Mobicom'03*, 2003, pp. 66–80.
14. D. Bertsekas and R. Gallager, *Data Networks*. Prentice Hall, 1992.