A Scalable Delay Based Analytical Framework for 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

Abstract—We present an analytical framework for the performance analysis of CSMA/CA based wireless mesh networks. This framework can provide an accurate throughput-delay evaluation for both saturated and unsaturated cases. As another important application of this framework, 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, both for a single class and multiple classes of traffic. 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.

Keywords: wireless mesh networks, performance analysis, MAC, optimization, multipath routing

I. 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 gateways to access backhaul links. High performance, including high throughput and low delay, is required for mesh networks because they are mainly used to serve commercial and residential customers. In this paper we aim to provide a scalable analytical framework for throughput-delay analysis in wireless mesh networks, and explore possible ways to improve their performance based on the analytical framework.

Like most work in this field [5], [13], [8], [9], [10], [7], our analysis mainly considers the effects of the medium access control (MAC) layer, the layer that has the largest difference between wired and wireless packet networks. Factors from the upper protocol layers that affect the performance are beyond the scope of this study and are not considered here.

The medium access control of mesh routers can be either centrally controlled by the base station (e.g. TDMA/FDMA/CDMA), or distributively controlled by each mesh router, typically using some form of CSMA/CA protocol. Despite its inefficiencies, the CSMA/CA based IEEE 802.11 protocol dominates in mesh network applications because it is economical. For this reason, IEEE 802.11 and the more fundamental CSMA/CA protocol are the main MAC protocols studied in mesh networks, and our work presented in this paper is also based on CSMA/CA. In more detail, it is based on non-persistent CSMA brought out by Kleinrock and Tobagi [11], which is the basis for the commercial IEEE 802.11 protocol.

As pointed out by Tobagi [2], the exact throughput-delay analysis for multi-hop networks requires a large state space. For large topologies, an exact analysis is almost impossible, leading us to consider an approximate analysis.

The most common approximation methods are single node based models, where each node has a view of the neighborhood, characterized by a number of parameters representing the average behavior of the neighboring nodes. Parameters for all nodes are then found through an iterative process. Representative work of this type include Leiner [3], Silvester and Lee [4], Bianchi [10], Medepalli and Tobagi [9], and Garetto, Salonidis and Knightly [8]. In this paper, the analytical model we introduce is also 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. However, with our analytical framework we not only provide both throughput and delay based performance evaluation, but also provide ways to improve the performance of the network, for example, by choosing the best routing paths, including multipath routing.

Multipath routing, also known as alternate path routing (APR), is an efficient way for avoiding congestion and loss in mesh networks and achieving higher capacity. By distributing traffic using different paths alternatively, the load in the network can have better balance, and thus achieve better performance [18], [19], [20].

The main contributions of this paper include: 1) A scalable model for the throughput-delay analysis of wireless mesh networks at both saturated and unsaturated loads. The performance of each node can be analyzed in isolation based on the knowledge of interfering neighbors and hidden terminals, which has much lower complexity than methods that maintain state of the complete network. The algorithm for deciding the hidden terminals also guarantees that our method can be easily applied for networks with large topologies. 2) Under an infinite buffer assumption, the Pollaczek-Khinchin (P-K) formula is used to derive closed form expressions for the mean waiting time in terms of path flow variables, which makes it possible for optimizing the network based on multipath routing. 3) For networks supporting multiple classes of traffic, a two step optimization method is presented to protect high priority traffic while guaranteeing the performance of the whole network.

The rest of this paper is organized as follows: In Section II, we discuss related work; in Section III we describe the basic model and exploit the neighbor relationships to derive solutions using iterative algorithms. In section IV, the closed
form representation of delay at each node is derived and a corresponding optimization model is introduced, both for a single class and multiple classes of traffic. Examples using our method for the analysis and optimization of wireless mesh networks are shown in section V, and section VI concludes this paper.

II. RELATED WORK

In the work by Leiner [3], a model of the neighborhood around each node is developed and characterized by 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 is generally not applicable for most multi-hop networks.

In a more recent work, Carvalho and Garcia-Luna-Aceves [6] present a single node based model that takes into consideration the effects of the physical layer parameters, MAC protocol, and connectivity. However, they mainly focus on the throughput of nodes for the saturated case, and no delay based analysis is addressed. 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. Although they also address the unsaturated load case, a delay-based analysis is also not included.

The work that is closest to ours is that of Medepalli and Tobagi [9], which is based on the framework of Bianchi’s work [10] with IEEE 802.11 Distributed Coordination Function (DCF). They extend Bianchi’s work to include multi-hop networks, attacking the unsaturated load situation, and providing a delay based analysis using an $M/M/1$ assumption. Their computing complexity is also not high due to the use of a single node based analysis. However, no closed form about delay is formed in their work, thus making delay based optimization impossible.

Boorstyn et al.’s work with node-group based decomposition [5] is another representative approach for the performance analysis of CSMA/CA based multi-hop networks. Wang and Kar [7] basically follow the same framework, but extend it into more complex MAC by considering RTS/CTS exchange, and study the fairness issues especially. Their main idea is that large networks can be decomposed into smaller groups, called “independent sets”, consisting of nodes that can transmit simultaneously. Markov chains are then built for those “independent sets” and product form solutions for steady state are obtained. Due to the need to compute all possible independent sets in the network, the complexity of the algorithm is prohibitive. Furthermore, this method can only be used for throughput and fairness analysis when the system is saturated, and analysis about delay can’t be done.

It is worthy to mention that, although applying the idea of “independent set” to the analysis of the whole network is formidable, it is profitable to just use it for neighboring nodes around a certain node. This is what is done in Garetto, Salonidis, and Knightly [8].

With respect to multipath routing in multi-hop networks, main work include Hass et al.[18], Du, Wu, Liu and Fang [21], Valera, Seah and Rao [19], Mosko and Garcia-Luna-Aceves [20]. In Hass et al. [18], the effect of route coupling on the efficiency of multi-path routing is studied, both for the multiple channel and single channel case. The coupling between routes is gauged as the number of nodes that are unable to receive on one path while nodes on another path are transmitting. The analysis pays more attention to the routing protocol itself and is more heuristic, and no quantified analysis for delay is given. Du, Wu, Liu and Fang [21] utilize the benefit of heterogenous networks. The path along nodes with high power (thus having higher data rates, larger transmission range and less hops) is chosen to take the most traffic. Their idea is close to multipath routing, in that most traffic will take the best path. However, similar to the work of Hass et al. [18], their work is routing protocol based and heuristic. Valera, Seah and Rao’s work [19] present the benefit of caching and using alternative paths when some routes fail in wireless ad hoc networks. Again their work defines a protocol rather than provide a performance analysis, while in Mosko and Garcia-Luna-Aceves[20], their main concern is to exploit the mesh connectivity to save path discovery operations, thus having less additional cost while obtaining better performance.

In conclusion, most current work about multipath routing in wireless multi-hop networks focus on the development of protocols, while the quantitative analysis, like which paths should be taken, what is the best traffic distribution etc., is lacking. In our work, we will take a system view to study the effect of interference and load on the choice of routes that maximize the system performance. As the result of optimization, we might need multipath routing, or just find the single best path. This work can be a good tool for finding the best deployment in practice.

Preliminary analytical work for this paper was presented in [22], and in [23] we show a few examples that support multi-class traffic. The topologies considered in those papers are simple. In this paper, we provide analysis about the computing complexity, and use an arbitrarily generated topology for the performance analysis and optimization applications.

III. BASIC MODEL

Similar to work presented in [7] 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 [11], [12] and Boorstyn et al. [5] to include a finite number of nodes, multiple hops, and interference caused by routing (hidden terminals). 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 $\lambda_i$. All message transmission times are exponentially distributed with mean $1/\mu$. Likewise, the channel capacity is taken to be $\mu$. We 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 transmission attempts (see Cali et al. [13]). Besides, each node backs off after a successful transmission to ensure that other nodes can get chance to transmit, which is especially important in order for the protocol to achieve fairness. The probability that node $i$ finds the medium free and is able to successfully transmit a message is denoted as $\alpha_i$. If node $i$ interferes with node $j$, then node $j$ also interferes with node $i$ (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 [12].) Those nodes that directly interfere or are hidden terminals to each other cannot send messages at the same time. We refer to all these nodes as “neighbors” in this paper and introduce a “neighbor matrix”, $N$, in section III-B, to derive these relationships.

![Fig. 1. Markov chain diagram of a single node.](image)

Fig. 1 depicts the queueing model for a single node. For each state $(l, S)$ or $(l, B)$, $S$ represents that the node is sending (transmitting), $B$ means that it is backing off, and $l$ represents the number of frames waiting in the queue. The buffer size is $L$. When $l = 0$, it means that there is no frame at this node, so the node is in idle state. This is an $M/G/1/I$ model from which the steady state, busy probability, blocking probability, etc. can be easily derived [14]. Strictly speaking, for internal nodes in the network that relay messages, the arrivals from different sources may be correlated with each other, so the aggregate arrival stream will not be Poisson. However, the assumptions we make allow us to use the $M/G/1/I$ model, which produces results that are extremely close to simulation.

### A. Calculating successful transmission probabilities

We have defined $\alpha_i$ as the probability that node $i$ successfully accesses the medium during a transmission attempt, so $\alpha_i$ 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 $\rho_i$ be the queuing system utilization of node $i$, which means this node is either in its “backoff” or “sending” state, so $\rho_i = P_S[i] + P_B[i]$. Node $i$ will sense the medium (attempt to transmit) only when it is in “backoff” state (the corresponding probability is $\rho_i - P_S[i]$). However, the transmission attempt will succeed only when no neighbor of node $i$ is transmitting, which has the probability $\rho_i - P_S[i] - \rho_i \cup_{k \in \omega_i} P_S[k]$. Here $\omega_i$ represents all nodes that are neighbors of node $i$, and $\cup_{k \in \omega_i} P_S[k]$ is the total “sending” probability of all those neighbors.

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

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

The value of $\alpha_i$ 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 to use an iterative method to find the value of $\alpha_i$.

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.

### B. Neighbor matrix

As mentioned by Jain et al. [15], an interference matrix, $F$, 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”, $N$.

We define a binary routing matrix $R$ to represent the routing relationship. Denote $R_{ij} = 1$ if node $i$ sends messages to $j$, otherwise $R_{ij} = 0$. In the interference matrix $F$, if node $i$ and $j$ are interfering with each other, we denote $F_{ij} = 1$ and $F_{ji} = 1$, otherwise $F_{ij} = 0$ and $F_{ji} = 0$. 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 $R$ by $F$ to get a new matrix $H = RF$. The hidden terminal information is already embedded in $H$, since if node $i$ and
\( j \) are hidden terminals to each other, there must exist one or more nodes \( k \) such that \( R_{jk} = 1 \) (node \( i \) wishes to talk to node \( k \)) and \( F_{kj} = 1 \) (node \( k \) and node \( j \) interferes with each other), so \( H_{ij} = \sum_k R_{ik} F_{kj} = 1 \).

**Step 2:** Combine the hidden terminal relationship with the direct interference relationship: let \( Y = X + F \).

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

### C. 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 are not neighbors to each other, but all are neighbors of 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 \( m \) (two or more) nodes that can transmit simultaneously, we call the set of those nodes simultaneously transmitting groups denoted as \( STG_m \), here \( m \) is defined as “group degree”.

**Algorithm 2: Simultaneously transmitting groups**

**Step 1:** Take the complementary set of the neighbor matrix \( N \) to identify the simultaneously transmitting pairs. Since this matrix describes the relationship between any two nodes, we denote it as \( S_2 \), so \( S_2 = \overline{N} \).

**Step 2:** For all node pairs \((i, j)\) such that \( S_{2,ij} = 1 \) (to avoid the duplication, we just consider the upper diagonal part of \( S_2 \)), list all possible nodes \( k \) (different from \( i, j \)) such that \( S_{2,ik} = 1 \) and \( S_{2,kj} = 1 \), and put all valid node groups \((i, j, k)\) into \( STG_3 \).

**Step 3:** For each node group \((i, j, k)\) in \( STG_3 \), find all possible nodes \( m \) such that \( S_{2,im} = 1 \), \( S_{2,jm} = 1 \) and \( S_{2,km} = 1 \), and put all valid node groups \((i, j, k, m)\) into \( STG_4 \). The above process continues until we reach \( n \) such that no \( n \) nodes can transmit simultaneously.

### D. Simultaneous transmitting probabilities

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

\[
\begin{align*}
\cup_{k \in \omega_i} P_S[k] &= \sum_{k \in \omega_i} P_S[k] - \sum_{(k_1,k_2) \in STG_2} P_S[k_1,k_2] \\
&+ \sum_{(k_1,k_2,k_3) \in STG_3} P_S[k_1,k_2,k_3] - \ldots
\end{align*}
\]  

(2)

where \( k_1, k_2, k_3, \ldots \in \omega_i \). Now we need to calculate the simultaneous transmitting 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 \( \omega_{k_1} = \{ q : N_{k_1q} = 1 \} \), and the neighbors of node \( k_2 \) is \( \omega_{k_2} = \{ q : N_{k_2q} = 1 \} \). Denote \( \omega_{k_1k_2} = \omega_{k_1} \cup \omega_{k_2} \). When both node \( k_1 \) and \( k_2 \) are sending, none of the nodes in \( \omega_{k_1k_2} \) can be sending.

\[
P_S[k_1,k_2] = P_S[k_1,k_2,\omega_{k_1k_2}] = P_S[k_1,k_2,\omega_{k_1k_2}]P_S[\omega_{k_1k_2}] = \sum_{\omega_{k_1k_2}} P_S[k_1,k_2,\omega_{k_1k_2}]P_S[\omega_{k_1k_2}], \quad (3)
\]

Since nodes \( k_1, k_2 \) are independent conditional on the probability that none of the nodes in \( \omega_{k_1k_2} \) are sending, we have

\[
P_S[k_1,k_2,\omega_{k_1k_2}] = P_S[k_1]P_S[k_2]P_S[\omega_{k_1k_2}]
\]

(4)

\( P_S[\omega_{k_1k_2}] \) represents the probability that no neighbor of node \( k_1, k_2 \) is sending, which can be written as \( 1 - P_S[\omega_{k_1k_2}] \) instead.

\[
P_S[k_1,\omega_{k_1k_2}] = P_S[k_1] - P_S[k_1,\omega_{k_1k_2}]
\]

Similarly we can get

\[
P_S[k_2,\omega_{k_1k_2}] = P_S[k_2] - P_S[k_2,\omega_{k_1k_2}]
\]

(5)

The calculation of \( P_S[\omega_{k_1k_2}] \), \( P_S[k_1,\omega_{k_1k_2}] \), \( P_S[k_2,\omega_{k_1k_2}] \) 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.

### E. Analysis of computing complexity

To guarantee a reasonable level of performance in wireless mesh networks, neighbors of each node should be limited. In fact, in [24] they suggest that 6 direct neighbors will help to achieve the best throughout performance. So, the number of simultaneously transmitting pairs can be as large as \( \binom{\alpha}{2} \) at most. Obviously, it will not change as the total number of nodes in the system - \( n \) increases. As the computation for \( \alpha \) will be computed one by one, the total complexity for computation on all \( n \) nodes is \( \binom{\alpha}{2} n \), or say \( O(n) \). As a result, we can conclude that our method is scalable.
IV. PATH DELAYS AND OPTIMIZATION

With the methods provided in the above section, performance evaluation in terms of both throughput and delay can be executed using an iterative algorithm similar to what has been used in [3], [10], [9]. However, for the case that loss does not occur in a network, we can get closed form solutions for $\alpha_i$ by making an infinite buffer assumption. This will help us improve the speed of computation for delay drastically. Even more important, multi-path based network optimization, which is very important for avoiding congestion and improving the network capacity, becomes possible. To our knowledge, our analytical framework is the first work that can provide optimization analysis in CSMA/CA based wireless multi-hop networks.

A. Closed form expression of delay

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

$$F(t) = 1 - p \exp(-Bt)e^t,$$

where $p$ is the starting vector for the process, $B$ is the progress rate operator for the process, and $e^t$ is a summing operator usually consisting of all 1’s [14]. The moments of the matrix exponential distribution are

$$E[X^n] = n!pB^{-n}e^t.$$

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

$$p = [1 \ 0] , \quad B = \begin{bmatrix} \beta \alpha_i & -\beta \alpha_i \\ 0 & \mu \end{bmatrix}. $$

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

$$E[S_i] = \frac{\mu + \alpha_i \beta}{\alpha_i \beta \mu},$$

$$E[S_i^2] = 2 \frac{\mu^2 + \alpha_i \beta \mu + \alpha_i^2 \beta^2}{\alpha_i \beta \mu^2}. $$

Using the P-K formula for M/G/1 queues, the mean waiting time in the queue at each node is $E[W] = \frac{\lambda E[S]^2}{\pi(1-\lambda E[S])}$. By substituting the expressions for the first 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

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

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

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

Since $P_S[k] = \lambda_k / \mu$, the simultaneous transmitting probability in terms of $\lambda_k$ – the arrival rate of each node, can be obtained either by solving equation groups shown in equation (5), or getting an approximation by iteration. So now we can get a closed form expression for $\alpha_i$. This will allow us to optimize the network delay.

B. Convexity of multipath routing

For a wired network, the service rate of each link is fixed, and the capacity region and response time function are both convex, for either M/M/1 or M/G/1 assumptions. Consequently, the delay-based multi-path routing optimization problem is easy to solve since the objective function is convex. In wireless multi-hop networks, the rate region is convex if some type of scheduling coordination is utilized [25]. In practice, a CSMA/CA based access protocol is the most common, and the convexity of rate region can be highly dependent on the specific protocol.

For the type of access protocol considered by Hedge and Proutiere [25], a single node can utilize the full capacity of the channel (achievable channel utilization is 1). As more nodes join, every node needs to back off to avoid collision. Due to the lack of centralized coordination, this kind of distributed behavior will cause overlapped backoff periods during which no node is transmitting, thus resulting in a lower utilization of the channel. For this reason, the capacity region will become concave.

The types of protocols we study will force each node to back off after each successful transmission allowing other nodes the opportunity to seize the channel. Although there will be issues about wasting channel resource when there is only one node, this should not be a problem in mesh networks since there will be many active nodes in order to ensure that multi-hop transmission works. In conclusion, the backoff-after-transmission scheme we are studying is especially suitable for multi-hop networks.

With the type of protocol we consider, the addition of more nodes with balanced traffic will only lead to higher channel utilization, thus causing a convex capacity region. For the case where all $n$ nodes can interfere with each other in a single hop network, we can derive the throughput for saturated load at each node as $\frac{\lambda_i \alpha_i \beta}{\alpha_i \beta \mu + \lambda_i}$, and the total channel utilization is $\frac{n \rho \mu}{\mu + n \rho i + 1}$ [23]. This result is similar to that of Kleinrock and Tobagi’s work for non-persistent CSMA ([11], P.1404) - $\frac{G}{G+1}$, where $G$ is the offered channel traffic. Both expressions show that the channel utilization will increase as more load or more nodes are added, which indicates that the capacity region is convex.

For multi-path routing, the key idea is to distribute traffic on multiple possible paths to involve more nodes (not more hops) and achieve better load balance. This is similar to the idea that more competing nodes and more balanced traffic will help achieve best channel utilization. In most realistic topologies, nodes will only interfere with a subset of the number of nodes in the network, allowing the total capacity of the whole network to be even greater.
The multi-path routing problem is a flow-deviation problem. This means that the total traffic through all possible paths will be a constant value. As shown in Figure 2, the feasible path distribution will either totally under the rate region, or will cross the rate region only once. The greater the distance from the rate region bound, the lower the average system delay, making the multi-path routing optimization problem to be convex.

C. Multipath optimization for single class

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

- **\( K \)**: Set of all origin-destination nodes that have traffic.
- **\( \mathbb{I} \)**: Set of communicating nodes in the network.
- **\( \Lambda \)**: Total arrival rate to the network, \( \Lambda = \sum_{k \in K} \Lambda_k \).
- **\( \Lambda_k \)**: Average arrival rate for origin-destination pair \( k \).
- **\( \mathcal{P}_k \)**: Set of possible paths for o-d pair \( k \).
- **\( \lambda_{kj} \)**: Amount of flow on path \( j \) for pair \( k \).
- **\( \alpha_i \)**: Transmission success probability at node \( i \), which is expressed as a function of the path flow variables \( \lambda_{kj} \) using equation (12).
- **\( \delta_{kj} \)**: Node path indicator: 1 if path \( j \) for pair \( k \) passes through node \( i \).
- **\( F_i \)**: Total flow through node \( i \), \( F_i = \sum_{k \in K} \sum_{j \in \mathcal{P}_k} \delta_{kj} \lambda_{kj} \).

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

\[
\min_{\lambda_{kj}, F_i} \frac{1}{\Lambda} \sum_{i \in \mathbb{I}} F_i \left( \frac{F_i - \mu - \alpha_i \beta}{-\alpha_i \beta \mu + F_i \mu + F_i \alpha_i \beta} \right),
\]

such that

\[
\sum_{j \in \mathcal{P}_k} \lambda_{kj} = \Lambda_k, \quad k \in K,
\]

\[
\sum_{k \in K} \sum_{j \in \mathcal{P}_k} \delta_{kj} \lambda_{kj} - F_i = 0, \quad i \in \mathbb{I},
\]

\[
\lambda_{kj} \geq 0, \quad F_i \geq 0.
\]

The objective function is rational, with polynomials in both the numerator and denominator. The constraints are linear, so we use a flow deviation algorithm [16](pp.468) 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.

With the closed form representation for response time obtained, we can also formulate some other optimization problems. Possible problems include but are not limited to: (1) Obtain the maximum throughput at the gateway while satisfying certain QoS requirements; (2) Compute the maximum throughput at each node under the principle of rate control with proportional fairness; (3) Guarantee the load balance by obtaining the MaxMin value of delay at each node. In this paper, we only present the case for mean delay as an example for the application of our analytical framework.

D. Optimization of multiple classes of traffic

In wireless mesh networks, both real-time and non-real-time traffic is possible. To support traffic with different QoS requirements, one method is to apply priority at each node.

Assume that each node \( i \) has \( R \) classes of arrivals, where class 1 has the highest priority and class \( R \) has the lowest priority. We denote the arrival rate of each class \( r \) as \( \lambda_{i,r} \). According to Cobham’s formula [17], the waiting time of each class can be expressed as

\[
E[W_{i,r}] = \frac{E[Q_{i,r}]}{(1 - \sigma_{i,r})(1 - \sigma_{i,r-1})},
\]

where \( E[Q_{i,r}] = \sum_{k=1}^{R} \lambda_{i,k}E[S_{i,k}^2]/2 \), \( \sigma_{i,r} = \sum_{k=1}^{r} \rho_{i,k} \), and \( \rho_{i,k} = \lambda_{i,k}/E[S_{i,k}] \). The first and second moment of the service times for each class can be computed using equations (9) and (10).

The mean total time spent at node \( i \) for each class \( r \) is the sum of waiting time in the queue and the service time: \( E[T_{i,r}] = E[W_{i,r}] + E[S_{i,r}] \).

For wireless mesh networks that support multiple classes of traffic, our goal is to guarantee the best system performance in terms of average path delay, and make sure that the high priority traffic will have short response times.

In a wired network, this goal can be achieved by optimizing class by class, from high to low. By doing this we can guarantee the highest priority traffic with the shortest path delay while the network is still optimized for traffic as a whole. When low priority traffic is added and optimized class by class, it will not affect the delay of the higher priority traffic that has been optimized. However, similar methods will not work for wireless mesh networks because the newly added low priority traffic will cause interference on the neighboring nodes, and
lead to the increased delay of high priority traffic. This makes the prior optimization on high priority traffic meaningless.

If we optimize the average system delay based cost functions for all classes of traffic, the optimization will tend to minimize the system delay of the traffic with higher volume, which are generally the lower priority classes. Thus, this kind of optimization may result in greater path delays for the higher priority traffic, which must be avoided.

As a solution, we propose the following efficient algorithm:

1. Optimize the system delay for all O-D pairs as a single class.
2. Among paths chosen for each O-D pair, optimize for each class, from high to low priority.

The first step of this algorithm can guarantee optimal system delay and the traffic load for each path. In the second step, the optimal distribution of load among optimal chosen paths is explored for high priority classes.

V. NUMERICAL RESULTS

A. 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.

In the scenarios we show in this section, 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). The backoff rate is equal to the transmission rate.

B. Performance evaluation of wireless mesh networks

In this subsection we show the effectiveness of our model by comparing analytical results to simulations.

An example of mesh network with 10 mesh nodes and a gateway is shown in Fig. 3. The circles around each node indicate the interference range of each node when it is transmitting messages. Nodes 1-5 are sources of traffic, and the other nodes are acting as mesh routers. The buffer size at each node is 100.

To apply the neighbor matrix algorithm from section III-B, we denote the gateway, GW, as node 11. Since the gateway does not send messages upstream on the same channel, the information about node 11 can be removed after we have obtained the neighbor matrix.

According to Fig. 3, we get the routing matrix and the interference matrix as:

$$R = \begin{bmatrix}
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
\end{bmatrix}$$

$$F = \begin{bmatrix}
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
\end{bmatrix}$$

We can then obtain the neighbor matrix using the algorithm described in section III-B:

$$N = \begin{bmatrix}
0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\
1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
\end{bmatrix}$$

After removing the information about gateway, which will not affect any node from transmitting, we have:

$$N = \begin{bmatrix}
0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\
1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
\end{bmatrix}$$

For the ease of evaluation, we assume that nodes 1-5 have the same amount of traffic. The comparison of simulation
and analytical results for throughput and delay are shown in Figs. 4-7.

As we can see, the analytical results for throughput are almost the same as the simulation for all loads. For the delay, the analytical results match perfectly with the simulation results for low load and heavy load, while there is a slight difference for moderate load. In total, the analytical results are very good at capturing the abrupt increase in delay as the load increases.

An interesting phenomenon we can observe is that the throughput of nodes 6, 7, 8, and 9 decreases after the loads at the source nodes surpass a certain point. The reason is that the lightly interfered nodes 1-5 tend to have higher throughput as the traffic increases, causing more interference and lower throughput at nodes 6-9. The result is that nodes 6-9 become the bottlenecks of this network.

Ironically, although nodes 1-5 prevail in the competition with those bottleneck nodes, resulting in more frames sent to the downstream bottleneck nodes, a greater number of them are dropped by the bottleneck nodes due to stronger interference at these nodes. Consequently, as more frames are transmitted from the source nodes, fewer will arrive at the gateway, causing lower throughput of the whole network.

Moreover, we observe that the delay at congested nodes eventually converge. This is because the throughput of each node will become a fixed value as the load is over certain limit, and the average queue length will approach the buffer size. According to Little’s law, the average waiting time for those accepted sessions will converge to a fixed value.

As another example, we show a mesh network with arbitrary topology as shown in Fig. 8. There are 20 nodes arbitrarily distributed in a 400m × 400m area, the transmission range of each node is 100m. Nodes 3, 4, and 7 have traffic to be sent through the gateway, which is node 20. Nodes that can interfere each other are connected by the dotted lines, while the solid lines represent the routing relationship among them. The arrival rate at node 4 is assumed to be fixed at 50 fps, and the load at node 3 and 7 can vary. For the ease of evaluation, we assume that the traffic rates at node 3 and 7 are always the same.

The analytical and simulation results for delay at nodes are shown in Fig. 9 and 10. Again, they are very close over a wide variety of offered loads.

In Fig. 11, the throughput of node 10, 19 and gateway are compared. The differences between the analytical result and simulation are small.

Similar to the 10-node example shown previously, the throughput at the gateway decreases after the load at the source nodes achieve a certain point.

From Fig. 11 we can also see that when the load is low, no loss happens; as the loads keep increasing, loss occurs at node 19 and heavily interfered upstream nodes like node 6 and node 1. However, the bottleneck problem can be solved by choosing routes wisely for certain source-destination pairs. This will be shown in the next subsection.
Remarks 1:
(1) Without any control, source nodes with light interference can cause very low throughput in the whole mesh network.
(2) The throughput analysis based on a saturated load assumption is impractical in a wireless mesh network. Main reasons include: (a) Some intermediate nodes in a mesh network will never become saturated if they are less interfered than upstream nodes. (b) A certain congestion control mechanism must be applied to make sure the network achieves high throughput and provides QoS for real-time traffic. Thus source nodes can not transmit at full rate.
(3) Analysis based on unsaturated load case is necessary and also possible. With our closed form expression of the constraints (including queueing system utilization, response time etc.), further analysis on the practical throughput becomes feasible.

C. Path delay based multipath optimization

With the topology shown in Fig. 8, traffic from node 4 to gateway can take five possible paths: path 4-15-1-8-19-GW, path 4-2-18-13-19-GW, path 4-2-6-17-10-GW, path 4-5-6-17-10-GW, and path 4-9-6-17-10-GW. We denote those paths as path 1 to 5 correspondingly. For different values of $\lambda_3$, $\lambda_4$ and $\lambda_7$, we can get the optimal paths by solving the optimization problem (equations (13), (14), (15), and (16)).

As a first example, we let $\lambda_4$ and $\lambda_7$ be fixed at 50 fps and 120 fps respectively. Optimization results for the traffic distribution out of node 4 are then obtained for different traffic rates at node 3 ($\lambda_3$). As shown in Fig. 12, the paths taken are path 2 and 5. However, when the traffic from node 3 becomes

Fig. 8. A mesh network with 20 nodes.

Fig. 9. Delay at nodes 3, 11 and 15.

Fig. 10. Delay at nodes 10 and 19.

Fig. 11. Throughput at nodes 10, 19 and 20

Fig. 12. Optimized traffic distribution with traffic from node 3 changes
higher, more traffic will take path 5 because the interference (from path 3-GW) on nodes along path 2 will become heavier, and makes the cost of taking this path higher. After a certain point, the interference generated by path 3-GW is so high that all traffic from node 4 takes path 5.

As a symmetric case, we let $\lambda_3$ and $\lambda_4$ be fixed at 120 fps and 50 fps, and $\lambda_7$ varies. The corresponding optimization results are shown in Fig. 13. When the traffic from node 7 is low, the paths taken are still path 2 and 5. However, more and more traffic goes to path 2 since the traffic increase at node 7 makes the loads and interference for nodes along path 5 go up. When $\lambda_7$ keeps increasing, the interference on nodes along path 2 becomes heavier and heavier, leading more and more traffic into path 1.

Finally, we can watch the variation of the traffic distribution with the increase of traffic from node 4 by letting both $\lambda_3$ and $\lambda_7$ are equal to 120 fps. In Fig. 14 we can see that path 2 and 5 are still the most favorite choices for light traffic loads. However, when the traffic becomes high, path 1 is also taken. The reason is that, after a certain point, the added burden on path 2 or 5 causes heavier interference on each other, making it more costly than distributing the traffic on path 1. Also note that the traffic on path 2 becomes less after the source traffic becomes substantially high. This is because path 1 and 5 are much further apart and generate less interference than path 2. In return, more traffic on path 1 and 5 will lead to high interference on path 2, making it even more unfavorable.

As a benefit of optimization, in Fig. 15 we can see that the system can support as much as 150 fps traffic from node 4. In contrast, in Fig. 16 we show that, if we just use a single path, the most traffic that can be supported is about 100 fps ($\lambda_3$ and $\lambda_7$ are also fixed at 120 fps, and path 4 is omitted here since its path delay is same as path 5). Among those single paths, we can see that path 2 is the best single path, and the maximum supported traffic from node 4 is about 100 fps. In contrast, path 1 can only support 75 fps of traffic from node 4. This also explains why path 1 is not chosen until the load is very high (Fig. 14).

From the experiments shown above, we can conclude that multipath routing tends to find the paths that is least loaded and interfered, which helps to balance the load in the system and improve the effective throughput of the network.

D. Optimization for multiple classes of traffic

With our algorithm for optimization of multiple classes of traffic in section IV, the high priority traffic will tend to take the path that is cheapest (in term of delay). So, when the cost of paths are not different enough, it is also possible that the high priority traffic will distribute on several different paths. In this subsection we are to show some examples to verify the validity of our algorithm.

Still consider the topology of Fig. 8. Assume that there are two classes of traffic starting from node 3, 4 and 7, and the class 1 traffic at each source node is 20 fps. Let $\lambda_3 = 120$ fps and $\lambda_7 = 120$ fps, we observe the traffic change as the load
on node 4 varies. The results are shown in Fig. 17.

Recall what we have shown in Fig. 14, when the traffic from node 4 is low, path 2 and 5 is chosen, and path 2 is favored over path 5; when the load becomes heavy, path 1 will also be taken. However, we can see that path 2 is not favored by the high priority traffic. In fact, path 5 and path 1 are better choices. The reason behind this is that the load on path 1 and 5 is much higher than path 2 due to the traffic originating from node 3 and 7, which makes path 2 a better choice to distribute traffic from node 4. However, the interference on path 1 and 5 are even lighter than path 2, makes them favorable for high priority traffic.

Another interesting phenomenon is that although path 5 tends to take all of high priority traffic (when $\lambda_4 \leq 80$), it can only take partial traffic since the optimized total traffic allocated for path 5 is less than the high priority traffic out of node 4. This means that to guarantee the optimized performance of the whole system, high priority traffic will sacrifice a little bit of performance.

In Fig. 18, the corresponding system delay for class 1, class 2 and the all traffic is shown. Obviously, the class 1 traffic is little affected by the increase of the system load, while for class 2, it can increase very fast as the load becomes heavy. The privilege of class 1 traffic is especially obvious when the system load is heavy. This indicates that the algorithm we provide is efficient at guaranteeing QOS of high priority traffic in wireless mesh networks.

Remarks 2:
1. The high priority traffic will always take the paths that are least interfered.
2. The performance of high priority traffic might be sacrificed a little bit to ensure the optimized performance of the whole network.

VI. 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. The comparison of simulation and analytical results show that our analytical method is accurate under both saturated and unsaturated cases. In addition, the bottleneck nodes can be easily identified using our analytical method.

For the infinite buffer case, we derive a closed form representation for response time, which opens the door for much more sophisticated analysis. As a representative application, we develop a model to identify the optimal multipath flow that minimizes the mean delay in the network. The optimization helps to find the best paths and traffic distribution, which improves the performance and capacity of the whole network. Furthermore, With the application on multiple classes of traffic, providing QoS for high priority traffic in wireless mesh network becomes reliable.

With the closed form representations for queueing system utilization and response time, another important class of study is throughput analysis under unsaturated load, with different throughput and delay constraints. Possible open problems include but are not limited to: (1) Obtaining the maximum throughput at the gateway while satisfying certain QoS requirements; (2) Computing the maximum throughput at each node under the principle of rate control with proportional fairness; (3) Maximum throughput analysis with the constraints of relaying at certain nodes. All of these are possible future work.

REFERENCES


