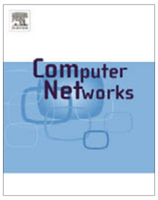
ARTICLE IN PRESS

Computer Networks xxx (2009) xxx–xxx

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet



A scalable delay based analytical framework for CSMA/CA wireless mesh networks

Jiazhen Zhou, Kenneth Mitchell *

Dept. of Computer Science Electrical Engineering, University of Missouri, Kansas City, MO 64110, United States

ARTICLE INFO

Article history:
Received 2 May 2008
Received in revised form 15 January 2009
Accepted 3 May 2009
Available online xxxx

Keywords:
Wireless mesh networks
Performance analysis
MAC
CSMA/CA
Optimization
Multipath routing

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. An efficient algorithm that determines the collision domain for each node based on both the interference range and routing in the network is presented. 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 for any number of classes. 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 gateways to access the 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.

Like most work in this field [5,7–10,14], we focus on 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.

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 studied by Kleinrock and Tobagi [12], which is the basis for current IEEE 802.11 type protocols.

Most implementations of CSMA/CA based wireless networks, including IEEE 802.11, are slotted systems with fixed size data frames. This would indicate the use of discrete models. However, an important observation made by Kleinrock and Tobagi [12] was that only the first moment of the retransmission delay distribution has an effect on the throughput-delay performance. This property allows for accurate continuous time models to be developed that avoid the difficulties of modeling cyclical behavior and synchronization found in discrete models.

---

* Corresponding author. Tel.: +1 816 235 1227; fax: +1 816 235 5159.
E-mail address: mitchellke@umkc.edu (K. Mitchell).

1389-1286/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2009.05.013

Please cite this article in press as: J. Zhou, K. Mitchell, A scalable delay based analytical framework for CSMA/CA wireless mesh networks, Comput. Netw. (2009), doi:10.1016/j.comnet.2009.05.013

Recently, Medepalli and Tobagi [9], go so far as to show that a simple M/M/1 model can give a reasonable approximation for the throughput-delay analysis of IEEE 802.11 networks.

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 et al. [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 [19–21].

The main contributions of this paper include: (1) A scalable model for the throughput-delay analysis of wireless mesh networks that is accurate 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. The biggest advantage of this optimization is that the network can achieve better load balance and accommodate more traffic. A two step method is presented to provide reliable QoS for high priority traffic while guaranteeing the performance of the whole network.

The rest of this paper is organized as follows: In Section 2, we discuss related work; in Section 3 we describe the basic model and exploit the neighbor relationships to derive solutions using iterative algorithms. In Section 4, 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 5. Section 6 concludes this paper.

2. 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 et al. [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 not included. Cali et al. [14] used a *p*-persistent CSMA mechanism instead of binary exponential backoff (BEB) to model the backoff behavior in IEEE 802.11 LANs. They discovered ways to maximize the throughput by finding the optimal contention window size for backoff.

The work of Medepalli and Tobagi [9] 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 low due to the use of a single node based analysis. However, no closed form about delay is presented in their work, thus making delay based optimization impossible.

The work of Boorstyn et al. on 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 use the same framework, but extend it to more complex MAC protocols by considering RTS/CTS exchange, and study fairness issues. Their main contribution 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. The need to compute all possible independent sets in the network makes the complexity of the algorithm prohibitive. Furthermore, this method can only be used for throughput and fairness analysis when the system is saturated, so a delay analysis is not provided.

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 use it for neighboring nodes around a certain node. This technique was used by Garetto et al. [8].

With respect to multipath routing in multi-hop networks, work appearing in the literature include Hass

Please cite this article in press as: J. Zhou, K. Mitchell, A scalable delay based analytical framework for CSMA/CA wireless mesh networks, Comput. Netw. (2009), doi:10.1016/j.comnet.2009.05.013

et al.[19], Du et al. [22], Valera et al. [20], and Mosko and Garcia-Luna-Aceves [21]. In Hass et al. [19], the effect of route coupling on the efficiency of multipath 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 et al. [22] utilize the benefit of heterogenous networks. The path along nodes with high power (those 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. [19], their work is protocol based and heuristic. Valera et al. [20] 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 provides a performance analysis, while in Mosko and Garcia-Luna-Aceves [21], 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 [23], and in [24] we show a few examples that support multiclass 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 performance analysis and optimization.

3. Basic model

Similar to work presented in [7,8], 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 [12,13] 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 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. Kleinrock and Tobagi [12,13] found that the throughput-delay performance of single-hop networks was dependent on only the first moment of the transmission delay and that there was no significant performance difference between continuous and slotted models. More recent work by Medepalli and Tobagi [9], and the simulation results presented in this paper confirm this observation for multi-hop networks. These observations allow us to use exponential distributions for the transmission times and backoff periods without any significant loss of accuracy. All frame transmission times have a mean duration of $1/\mu$. Likewise, the mean channel capacity is taken to be μ. We assume a mechanism similar to the two phase collision resolution used by Yang and Vaidya [25] where collisions are resolved during backoff stages. This mechanism allows the node to determine if the medium is available or if it must wait and reattempt access to the channel. We assume that the RTS/CTS collision resolution (including RTS/CTS collisions and retransmissions) occur during this backoff period which has mean duration of $1/\beta$, resulting in a geometrically distributed number of transmission attempts (see Cali et al. [14].) Additionally, 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 α_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 [13].) 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" and introduce a "neighbor matrix", $\boldsymbol{N}$, in Section 3.3, to derive these relationships.

When the system achieves steady state, we assume that the probability (α) that a node finds the medium idle when it attempts to transmit is a constant. That means the number of backoff periods (denoted as k) that a node needs to experience before it successfully accesses the medium is geometrically distributed. At the same time, the length of the total backoff period before a transmission is the convolution of a possibly infinite number of backoff time units. The Laplace transform for k successive backoff periods is $\left(\frac{\beta}{(\beta+s)}\right)^k$. The probability mass function for the number of backoffs k in this period is geometric $Pr[\text{number of backoffs} = k] = \alpha(1-\alpha)^{k-1}$. The Laplace transform for the total backoff period:

$$F^*(s) = \sum_{k=1}^{\infty} \alpha(1-\alpha)^{k-1}\left(\frac{\beta}{(\beta+s)}\right)^k = \frac{\alpha\beta}{\alpha\beta+s}. \tag{1}$$

That is to say, the backoff period is exponentially distributed with the average rate being $\alpha\beta$, thus the queueing model for a single node can be depicted as in Fig. 1.

For each state (l,S) or (l,B) in Fig. 1, S represents that the node is transmitting (sending), 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 $M/G/1/L$ model with a load dependent server from which

Please cite this article in press as: J. Zhou, K. Mitchell, A scalable delay based analytical framework for CSMA/CA wireless mesh networks, Comput. Netw. (2009), doi:10.1016/j.comnet.2009.05.013

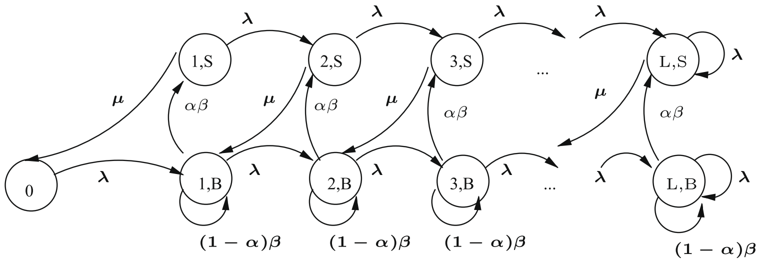


Fig. 1. Markov chain diagram of a single node.

the steady state, busy probability, blocking probability, etc. can be derived [15].

3.1. Calculation of state probabilities, blocking probabilities, throughput and delay

The service time distribution at each node consists of both the transmission time and the waiting time at the head of a queue (when the node is in "backoff" state). It has a matrix exponential distribution representation

$$F(t) = 1 - \boldsymbol{p}\exp(-\boldsymbol{B}t)\boldsymbol{e}', \tag{2}$$

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 usually consisting of all 1's [15]. The moments of the matrix exponential distribution are

$$E[X^n] = n!\boldsymbol{p}\boldsymbol{B}^{-n}\boldsymbol{e}'. \tag{3}$$

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

$$\boldsymbol{p} = [1 \quad 0], \quad \boldsymbol{B} = \begin{bmatrix} \beta\alpha_i & -\beta\alpha_i \\ 0 & \mu \end{bmatrix}. \tag{4}$$

The steady state probability vectors $\boldsymbol{\pi}_l$ for the M/G/1/L system has the following closed form solution.

$$\boldsymbol{\pi}_l = \pi_0 \boldsymbol{p}\boldsymbol{U}^l, \quad 1 \leqslant l \leqslant L \tag{5}$$

where the matrix geometric generator $\boldsymbol{U}$ is given by $\boldsymbol{U} = \lambda(\lambda\boldsymbol{I} + \boldsymbol{B} - \lambda\boldsymbol{e}'\boldsymbol{p})^{-1}$, $\boldsymbol{I}$ is the identity matrix of the same dimension as $\boldsymbol{B}$, and $\boldsymbol{p}$ and $\boldsymbol{B}$ are defined in Eq. (4). The value of π_0 can be found using the normalization equation $\pi_0 + \sum_{l=1}^{L} \boldsymbol{\pi}_l \boldsymbol{e}' = 1$. The probability of blocking, P_{Bk}, mean throughput, Tpt, average number of frames at a node, $\overline{N}$, and mean delay, $\overline{T}$, are given below.

$$P_{Bk} = \boldsymbol{\pi}_L \boldsymbol{e}', \tag{6}$$

$$Tpt = \lambda(1 - P_{Bk}), \tag{7}$$

$$\overline{N} = \sum_{l=1}^{L} l\boldsymbol{\pi}_l \boldsymbol{e}', \tag{8}$$

$$\overline{T} = \overline{N}/Tpt. \tag{9}$$

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/L* model, which produces results that are extremely close to simulation.

3.2. Calculating successful transmission probabilities

As defined above, α_i is 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. As shown in Fig. 2, 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, it is 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 $\rho_i = P_S[i] + P_B[i]$ and $P_B[i]$ can be expressed as $\rho_i - P_S[i]$.

Instead of calculating α_i directly, it is easier to compute its complementary part – the failure probability given a transmission attempted. The computation of this conditional probability is shown as follows:

$$\begin{aligned} 1 - \alpha_i &= P[\text{Failure}|\text{An attempt tried}] \\ &= \frac{P[\text{An attempt tried and failed}]}{P[\text{An attempt tried}]} \\ &= \frac{P[\text{Node } i \text{ in backoff and neighbors busy}]}{P[\text{Node } i \text{ in backoff}]}. \end{aligned} \tag{10}$$

Note that a node will attempt to transmit only when it is in backoff and the attempt will fail if at least one neighbor is busy. Now the problem becomes how to compute the probability that a node is in backoff while at least one of its neighbors is busy.

As can be seen in Fig. 2, when at least one of node i's neighbors is busy, node i is at either idle or in backoff, thus

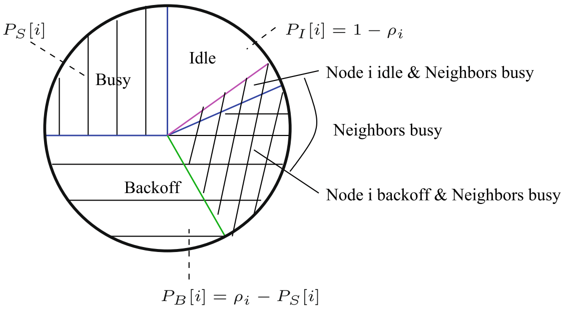


Fig. 2. Busy probability of neighbors viewed by node i.

Please cite this article in press as: J. Zhou, K. Mitchell, A scalable delay based analytical framework for CSMA/CA wireless mesh networks, Comput. Netw. (2009), doi:10.1016/j.comnet.2009.05.013

$$\begin{aligned} &P[\text{Node } i \text{ in backoff and neighbors busy}] \\ &\quad = P[\text{Neighbors busy}] \\ &\qquad - P[\text{Node } i \text{ idle and neighbors busy}]. \end{aligned} \tag{11}$$

As the busy state of neighbors will only cause node i to backoff, *the event that node i is idle is independent from the event that neighbors become busy*. This means

$$\begin{aligned} &P[\text{Node } i \text{ idle and neighbors busy}] \\ &\quad = P[\text{Node } i \text{ idle}]P[\text{neighbors busy}] \\ &\quad = (1-\rho_i)\cup_{k\in\omega_i} P_S[k]. \end{aligned} \tag{12}$$

Here ω_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.

After combining Eqs. (10)–(12), the failure probability for a transmission attempt can be computed as

$$1-\alpha_i = \frac{\rho_i \cup_{k\in\omega_i} P_S[k]}{\rho_i - P_S[i]}. \tag{13}$$

The success probability for a transmission attempt is thus

$$\begin{aligned} \alpha_i &= \frac{\rho_i - P_S[i] - \rho_i \cup_{k\in\omega_i} P_S[k]}{\rho_i - P_S[i]} \\ &= \frac{1 - P_S[i]/\rho_i - \cup_{k\in\omega_i} P_S[k]}{1 - P_S[i]/\rho_i}. \end{aligned} \tag{14}$$

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 to either solve a system of equations or use an iterative method to find the value of α_i as explained in subsection 3.6.

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 subsections we will introduce ways to solve the above problems.

3.3. Neighbor matrix

As mentioned by Jain et al. [16], 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 R by F to get a new matrix $H = R\,F$. 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 $\boldsymbol{R}_{ik} = 1$ (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 $Y = H + F$;

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 $\boldsymbol{N}_{ij} = 1$ 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. 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 ω_i in Eq. (14) as the set of nodes represented by 1's in the ith row of the neighbor matrix N.

3.4. 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 $\boldsymbol{S}_2$, 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 STG_3.

Please cite this article in press as: J. Zhou, K. Mitchell, A scalable delay based analytical framework for CSMA/CA wireless mesh networks, Comput. Netw. (2009), doi:10.1016/j.comnet.2009.05.013

Step 3: For each node group (i, j, k) in STG_3, find all possible nodes m such that $\mathbf{S_{2,im}} = 1, \mathbf{S_{2,jm}} = 1$ and $\mathbf{S_{2,km}} = 1$, 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.

3.5. Simultaneous transmitting probabilities

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

$$\cup_{k\in\omega_i} P_S[k] = \sum_{k\in\omega_i} P_S[k] - \sum_{(k_1,k_2)\in STG_2} P_S[k_1k_2] + \sum_{(k_1,k_2,k_3)\in STG_3} P_S[k_1k_2k_3] - \ldots, \quad (15)$$

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 do not 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 $\omega_{k_1} = \{q : \mathbf{N}_{k_1q} = 1\}$, and the neighbors of node k_2 is $\omega_{k_2} = \{q : \mathbf{N}_{k_2q} = 1\}$. 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.

$$P_S[k_1, k_2] = P_S[k_1, k_2, \overline{\omega_{k_1k_2}}] = P_S[k_1, k_2|\overline{\omega_{k_1k_2}}]P_S[\overline{\omega_{k_1k_2}}]. \quad (16)$$

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

$$P_S[k_1, k_2|\overline{\omega_{k_1k_2}}] = P_S[k_1|\overline{\omega_{k_1k_2}}]P_S[k_2|\overline{\omega_{k_1k_2}}] = \frac{P_S[k_1, \overline{\omega_{k_1k_2}}]}{P_S[\overline{\omega_{k_1k_2}}]} \frac{P_S[k_2, \overline{\omega_{k_1k_2}}]}{P_S[\overline{\omega_{k_1k_2}}]}. \quad (17)$$

$P_S[\overline{\omega_{k_1k_2}}]$ 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. $P_S[k_1, \overline{\omega_{k_1k_2}}]$ 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 $P_S[k_1, \overline{\omega_{k_1k_2}}] = P_S[k_1, \overline{\omega_{k_2\overline{k_1}}}] = P_S[k_1] - P_S[k_1, \omega_{k_2\overline{k_1}}]$. Similarly we can get $P_S[k_2, \overline{\omega_{k_1k_2}}] = P_S[k_2] - P_S[k_2, \omega_{k_1\overline{k_2}}]$.

After combining Eqs. (16) and (17), and the computation for $P_S[k_1, \overline{\omega_{k_1k_2}}]$, $P_S[k_1, \overline{\omega_{k_1k_2}}]$, and $P_S[\overline{\omega_{k_1k_2}}]$, the resulting expression is

$$P_S[k_1, k_2] = \frac{(P_S[k_1] - P_S[k_1, \omega_{k_2\overline{k_1}}])(P_S[k_2] - P_S[k_2, \omega_{k_1\overline{k_2}}])}{1 - P_S[\omega_{k_1k_2}]}. \quad (18)$$

The calculation of $P_S[\omega_{k_1k_2}], P_S[k_1, \omega_{k_2\overline{k_1}}], P_S[k_2, \omega_{k_1\overline{k_2}}]$ can be done similarly by using Eq. (15). 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.

3.6. Iterative process for computing throughput and transmission probabilities

The throughput and loss probability of node i can be obtained by solving the Markov chain shown in Fig. 1. However, two parameters involved – λ_i and α_i, are dependent on the state of the neighboring nodes. The aggregate arrival rate at node i, λ_i, consists the originating traffic at node i and the throughput of upstream nodes that flow into node i. As for α_i, it represents the average view of neighbors, which in turn will be expressed in terms of the throughput at neighboring nodes. The iterative process used here is similar to those used in [3,4,8–11,26]. The method we use to determine the collision probabilities does not affect the convergence properties of the iterative methods used above.

Algorithm 3. Iterative method

Step 1: For each node i, initialize λ_i (total arrival rate at each node) and α_i.

Step 2: At each iteration, the throughput of each node is computed using Eq. (7) with the current values of λ_i and α_i.

Step 3: α_i and λ_i will be recomputed in each new iteration
- (**a**) $\lambda_i = \lambda_i^o + \sum_{k:R_{kj}=1} Tpt_k$, in which λ_i^o is the originating arrival rate of node i, and Tpt_k is the throughput of node k – an immediate upstream node of node i.
- (**b**) α_i is computed using Eqs. (14) and (15), in which $P_S[k] = \frac{Tpt_k}{\mu}$.

The iteration ends when the difference of the throughput of all nodes i between two consecutive iterations $< \epsilon$.

3.7. 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 [27] it is suggested that six direct neighbors will help to achieve the best throughout performance. So, the number of simultaneously transmitting pairs can be as large as $\binom{6}{2}$ at most. Obviously,it will not change as the total number of nodes in the system – n increases. As the computation for α_i will be computed one by one, the total complexity for computation on all n nodes is $\binom{6}{2}$, or say $O(n)$. As a result, we can conclude that our method is scalable.

Another observation is that in the current IEEE 802.11 based metropolitan wireless mesh networks implemented in Philadelphia, Houston, Taipei, and Hongkong [28], the number of mesh nodes that can access a gateway is limited due to the inefficiency of CSMA based protocol in multi-hop networks. In other words, a huge metropolitan

Please cite this article in press as: J. Zhou, K. Mitchell, A scalable delay based analytical framework for CSMA/CA wireless mesh networks, Comput. Netw. (2009), doi:10.1016/j.comnet.2009.05.013

network must be divided into small clusters. The analysis can be done cluster by cluster. Thus the size of the problem is controllable.

4. Delay based 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 α_i by making an infinite buffer assumption. This will help us improve the speed of computation for delay drastically. Even more important, multipath 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.

4.1. Closed form expression for delay

Using Eq. (3) from subsection 3.1, the mean and the second moment of the service time at node i are

$$E[S_i] = \frac{\mu + \alpha_i \beta}{\alpha_i \beta \mu}, \qquad E[S_i^2] = 2\,\frac{\mu^2 + \alpha_i \beta \mu + \alpha_i^2 \beta^2}{\alpha_i^2 \beta^2 \mu^2}. \tag{19}$$

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 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

$$E[T_i] = \frac{\mu + \alpha_i \beta - \lambda_i}{\alpha_i \beta \mu - \lambda_i \mu - \lambda_i \alpha_i \beta}. \tag{20}$$

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 $P_S[i]$ will be λ_i/μ (portion of time that the medium is used by node i for transmitting frames.) Substituting ρ_i and $P_S[i]$ into Eq. (14) and solving for α_i, we get

$$\alpha_i = \frac{\mu(1 - \cup_{k\in\omega_i} P_S[k])}{\mu + \beta \cup_{k\in\omega_i} P_S[k]} = \frac{1 - \cup_{k\in\omega_i} P_S[k]}{1 + \beta/\mu \cup_{k\in\omega_i} P_S[k]}. \tag{21}$$

Since $P_S[k] = \lambda_k/\mu$, the simultaneous transmitting probability in terms of λ_k – the arrival rate of each node, can be obtained either by solving equation groups shown in Eq. (18), or getting an approximation by iteration. So now we can get a closed form expression for α_i, and thus $E[T_i]$. This will allow us to optimize the network delay.

4.2. Multipath optimization for single class

The main idea of multipath routing is to distribute flows on several feasible paths. This can be formulated as a flow-deviation problem [17] since the total traffic for each source–destination pair is known. As for the objective function, there can be different choices such as minimizing the average system delay, or minimizing the maximum delay of links or source–destination pairs. As a preliminary example we use the average system delay.

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

$\mathscr{K}$	set of all origin-destination nodes that have traffic
$\mathscr{I}$	set of communicating nodes in the network
Λ_k	average arrival rate for origin-destination pair k
Λ	total arrival rate to the network, $\Lambda = \sum_{k\in\mathscr{K}} \Lambda_k$
$\mathscr{P}_k$	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 Eq. (21)
δ^i_{kj}	node path indicator: 1 if path j for pair k passes through node i
F_i	total flow through node i, $F_i = \sum_{k\in\mathscr{K}} \sum_{j\in\mathscr{P}_k} \delta^i_{kj} \lambda_{kj}$

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

$$\min_{\lambda_{kj},F_i} \frac{1}{\Lambda} \sum_{i\in\mathscr{I}} F_i \frac{\mu + \alpha_i\beta - F_i}{\alpha_i\beta\mu - F_i\mu - F_i\alpha_i\beta}, \tag{22}$$

such that

$$\sum_{j\in\mathscr{P}_k} \lambda_{kj} = \Lambda_k, \quad k \in \mathscr{K}, \tag{23}$$

$$\sum_{k\in\mathscr{K}} \sum_{j\in\mathscr{P}_k} \delta^i_{kj} \lambda_{kj} - F_i = 0, \quad i \in \mathscr{I}, \tag{24}$$

$$\lambda_{kj} \geqslant 0, \quad F_i \geqslant 0. \tag{25}$$

Eq. (23) specifies that the total traffic along all possible paths should be equal to that of the source–destination pair. As shown in Eq. (24), the total traffic out of each node is equal to all flows going through it. Another constraint is that the load on a node will not be greater than the capacity. This usually can be guaranteed in the algorithm, since the delay (penalty) at a node becomes infinite as the flow approaches the capacity.

The systems studied by Hegde and Proutiere [31] and many wired networks do not have a convex rate region as they allow a single transmitting node to utilize 100% of the channel. The CSMA/CA systems we are modeling have a backoff before transmit mechanism. As a result, the capacity region as a function of the number of nodes in a collision domain as determined by Eq. (14) is convex

$$C(N) = N/(N\beta - \mu) \tag{26}$$

where β is the backoff rate and μ is the transmission rate. This same expression was also presented by Kleinrock and Tobagi [12].

Consider a single communicating pair with transmission path length k, thus requiring k transmissions for each frame to travel from the source to the destination. By routing some of the flow along an alternate path also of length k, we have doubled the number of nodes transmitting, but the number of transmissions needed to transmit a frame from the source to the destination still remains at k. Eq. (26) shows us that the network becomes more efficient when there are more transmitting nodes. For any single server queueing model, the term $1/(1-\rho)$ in the

Please cite this article in press as: J. Zhou, K. Mitchell, A scalable delay based analytical framework for CSMA/CA wireless mesh networks, Comput. Netw. (2009), doi:10.1016/j.comnet.2009.05.013

expression for delay implies that any linear increase in utilization results in an exponential increase in delay. Thus the delay function is optimal when traffic is distributed equally among both paths. For unequal length paths as much traffic as possible will be routed along the shorter path. For multipath routing, the key idea is to distribute traffic on multiple possible paths (especially through different collision domains) 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 better channel utilization. Based on exhaustive analytic and simulation studies, the convex property appears to hold for multi-commodity flows and proofs for both the single and multi-commodity cases is a topic of ongoing research.

The constraints are linear, so we use a flow-deviation algorithm [17, (p. 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.

4.3. 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 [18], the waiting time of each class can be expressed as

$$E[W_{i,r}] = \frac{E[Q_{i,R}]}{(1-\sigma_{i,r})(1-\sigma_{i,r-1})}, \tag{27}$$

where $E[Q_{i,R}] = \sum_{k=1}^{R} \lambda_{i,k} E\left[S_{i,k}^2\right]/2, \sigma_{i,r} = \sum_{k=1}^{r} \rho_{i,k}$, 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 Eq. (19).

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 is 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.

5. Numerical results

5.1. 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, collision resolution has been finished and this node has the accurate knowledge about the availability of the channel. If the channel is not available, the node will return to 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.

5.2. 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 a 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 $L = 100$.

To apply the neighbor matrix algorithm from Section 3.3, we denote the gateway, GW, as node 11. Since the

Please cite this article in press as: J. Zhou, K. Mitchell, A scalable delay based analytical framework for CSMA/CA wireless mesh networks, Comput. Netw. (2009), doi:10.1016/j.comnet.2009.05.013

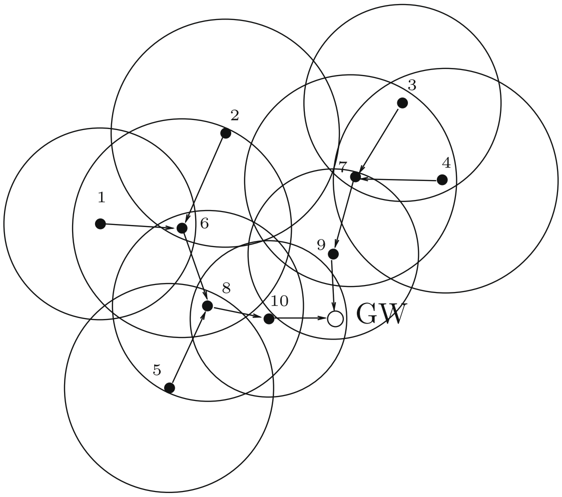


Fig. 3. Ten node multi-hop network.

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:

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

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

We can then obtain the neighbor matrix using the algorithm described in Section 3.3. After removing the information about the gateway, which will not affect any node from transmitting, we have:

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

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 and 5.

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 catching the abrupt increase in delay as the load increases.

An interesting phenomenon we can observe is that the throughput of nodes 6–9 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 overall throughput.

We also observe that the delay at congested nodes eventually converges. This is because the throughput at each node will become fixed as the load is over a 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.

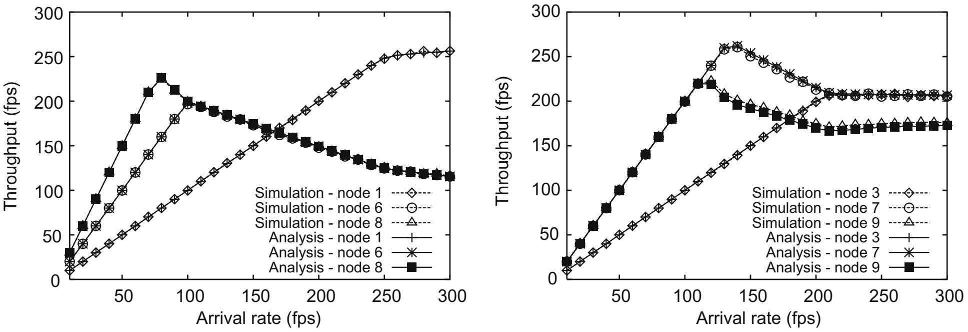
As another example, we show a mesh network with arbitrary topology as shown in Fig. 6.

There are 20 nodes arbitrarily distributed in a 400 m × 400 m area, the transmission range of each node is 100 m. 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. 7. Again, they are very close over a wide variety of offered loads.

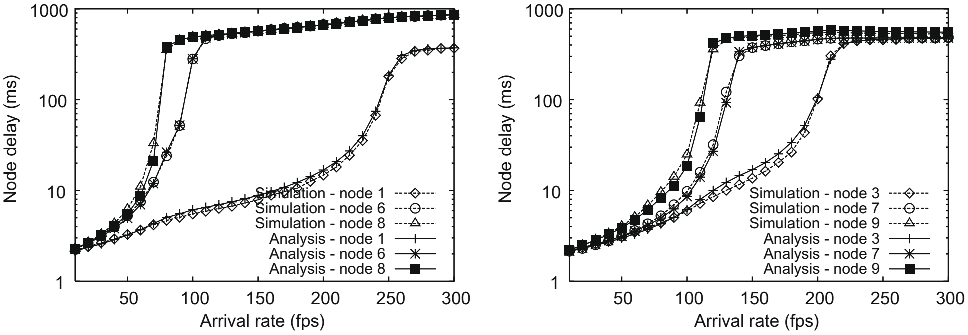
In Fig. 8, the throughput of node 10, 19, and the gateway are compared. The differences between the analytical

Please cite this article in press as: J. Zhou, K. Mitchell, A scalable delay based analytical framework for CSMA/CA wireless mesh networks, Comput. Netw. (2009), doi:10.1016/j.comnet.2009.05.013



(a) Throughput at nodes 1, 6, 8 (L = 100). (b) Throughput at nodes 3, 7, 9 (L = 100).

Fig. 4. Throughput analysis for the 10 node network.



(a) Delay at nodes 1, 6, 8 (L = 100). (b) Delay at nodes 3, 7, 9 (L = 100).

Fig. 5. Delay analysis for the 10 node network.

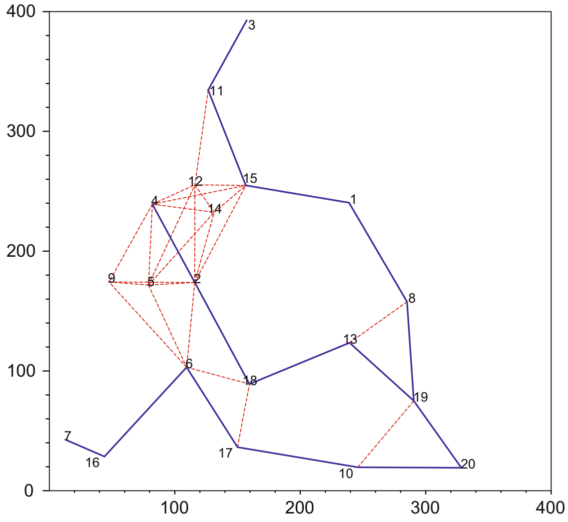


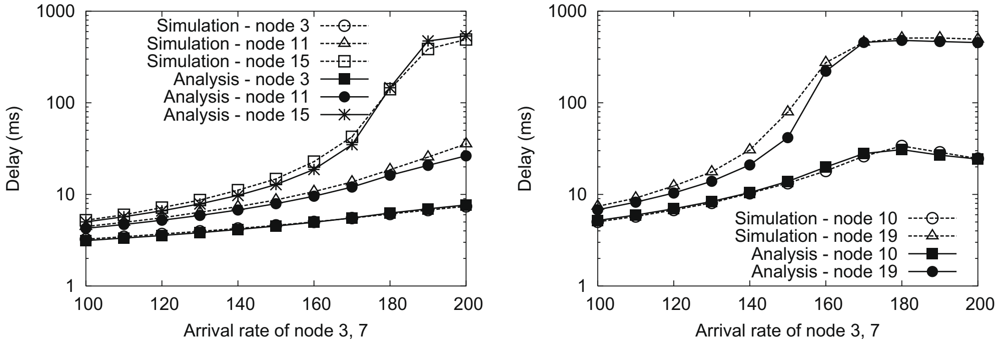
Fig. 6. A mesh network with 20 nodes.

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 reaches a certain point.

From Fig. 8 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.

5.2.1. 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 with 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) An analysis based on unsaturated loads 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.



(a) Delay at nodes 3, 11, and 15. (b) Delay at nodes 10 and 19.

Fig. 7. Delay analysis for the 20 node random topology.

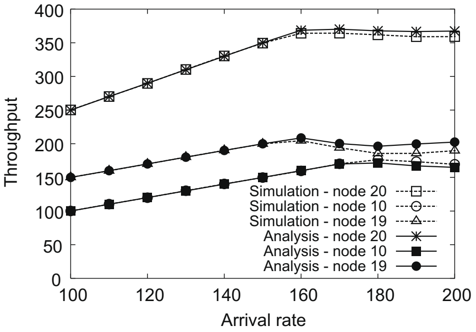


Fig. 8. Throughput at nodes 10, 19 and 20.

5.3. Comparison with IEEE 802.11

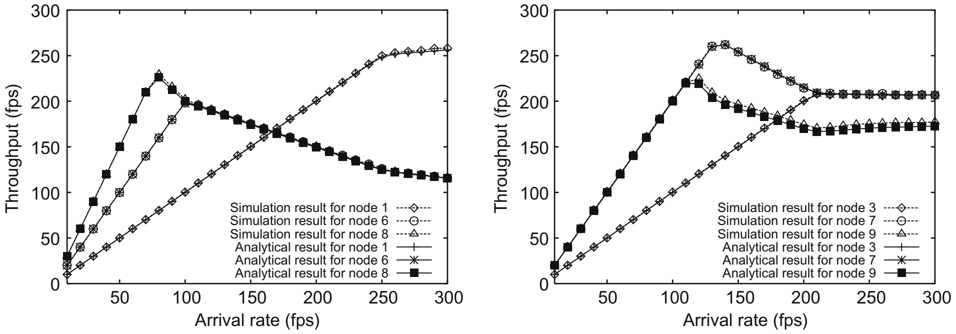
From the above subsection it can be seen that the analytical results obtained from the iterative methods are very close to the simulation results that employ the same protocol under the same assumptions. This subsection is to show how the analysis performs compared to IEEE 802.11.

The main differences between our protocol and IEEE 802.11 include: (1) IEEE 802.11 uses time slots for backoff time, and the length of backoff time is a *uniform random integer* between 1 and the contention window size; in contrast, the backoff time in our model is assumed to be continuous and exponentially distributed; (2) the service time for each frame is generally *determined* in 802.11, while it is assumed to be exponentially distributed in the protocol employed in this paper. The scenario evaluated is same as Fig. 3. From Figs. 9 and 10 it can be seen that the above differences do not cause any obviously large discrepancies between our analytical results and IEEE 802.11. This is not that surprising, as we are performing an average value analysis, so the result is sensitive to the average value rather than the distribution which confirms Kleinrock and Tobagi's observations [12,13]. In conclusion, this means that our analytical model can also be used for 802.11.

It is also noteworthy to mention that binary exponential backoff used in 802.11 will cause serious unfairness in a multi-hop network [29,30]. This is why we assume that each node achieves the same mean duration of the backoff period in the simulation for 802.11, just as if a constant backoff period is used (like in our protocol).

5.4. Path delay based multipath optimization

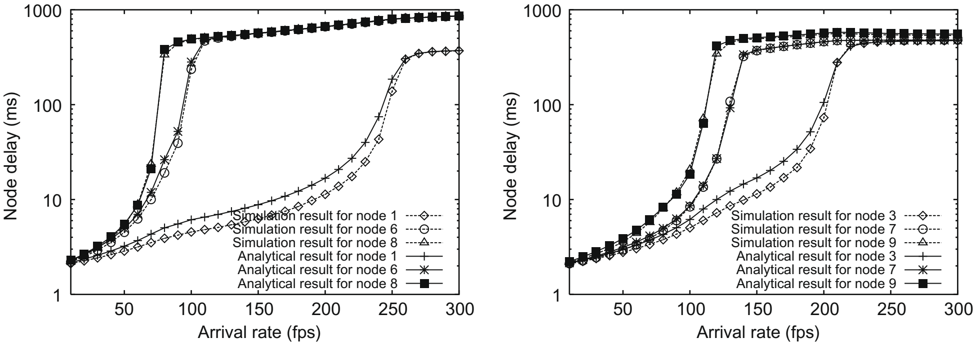
With the topology shown in Fig. 6, traffic from node 4 to gateway can take five possible paths: path 4-15-1-8-19-GW, path 4-2-18-13-19-GW, path 4-2-6-17-10-GW, path



(a) Comparison of nodes 1,6, and 8. (b) Comparison of nodes 3,7, and 9.

Fig. 9. Throughput comparison with 802.11.

Please cite this article in press as: J. Zhou, K. Mitchell, A scalable delay based analytical framework for CSMA/CA wireless mesh networks, Comput. Netw. (2009), doi:10.1016/j.comnet.2009.05.013



(a) Comparison of nodes 1,6, and 8. (b) Comparison of nodes 3,7, and 9.

Fig. 10. Delay comparison with 802.11.

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 λ_3, λ_4 and λ_7, we can get the optimal paths by solving the optimization problem (Eqs. (22)–(25)).

As a first example, we let λ_4 and λ_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 (λ_3). As shown in Fig. 11, the paths taken are 2 and 5. However, when the traffic from node 3 becomes 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.

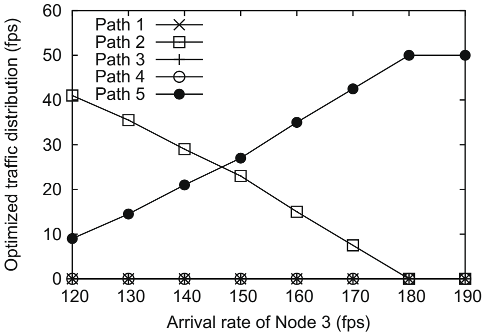
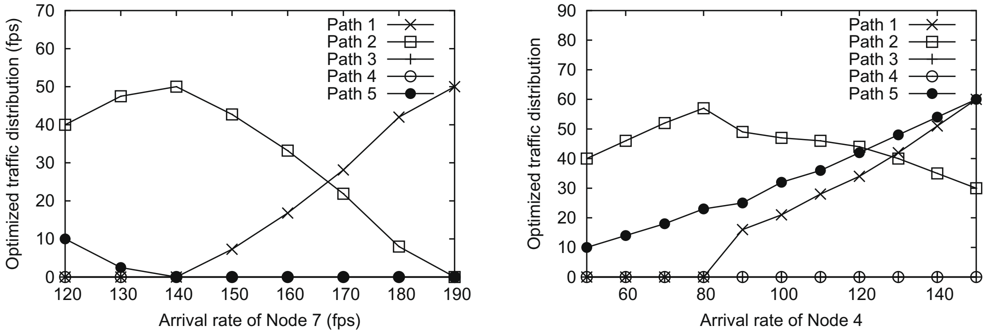


Fig. 11. Optimized traffic distribution for varying λ_3.

As a symmetric case, we let λ_3 and λ_4 be fixed at 120 fps and 50 fps, and let λ_7 vary. The corresponding optimization results are shown in Fig. 12a. 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 λ_7 keeps increasing, the interference on nodes along path 2 becomes heavier and heavier, leading more and more traffic to path 1.

Finally, we can watch the variation of the traffic distribution with the increase of traffic from node 4 by letting both λ_3 and λ_7 be equal to 120 fps. In Fig. 12b we can see that paths 2 and 5 are still the most favorable 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 paths 1 and 5 are much further apart and generate less interference than path 2. In return, more traffic on paths 1 and 5 will lead to high interference on path 2, making it even more unfavorable.



(a) Optimized traffic distribution when λ_7 changes. (b) Optimized traffic distribution when λ_4 changes.

Fig. 12. Optimized traffic distribution while varying λ_4 and λ_7.

Please cite this article in press as: J. Zhou, K. Mitchell, A scalable delay based analytical framework for CSMA/CA wireless mesh networks, Comput. Netw. (2009), doi:10.1016/j.comnet.2009.05.013

As a benefit of optimization, in Fig. 13a we can see that the system can support as much as 150 fps traffic from node 4. In contrast, in Fig. 13b we show that, if we just use a single path, the most traffic that can be supported is about 100 fps (λ_3 and λ_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. 12b).

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

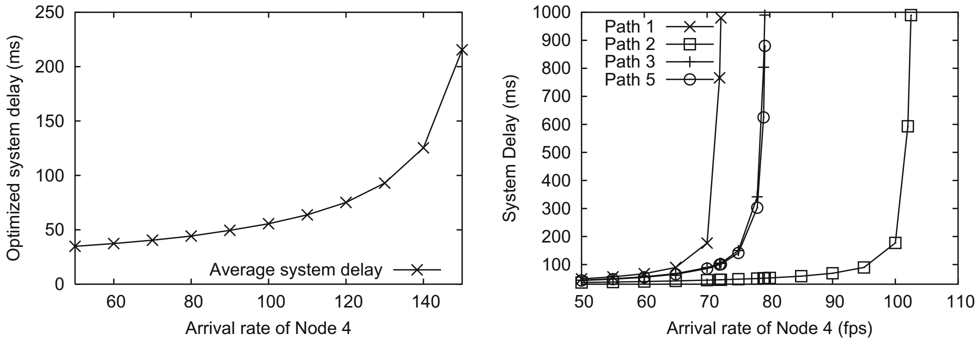
5.5. Optimization for multiple classes of traffic

With our algorithm for optimization of multiple classes of traffic in Section 4, 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 be distributed on several different paths. In this subsection we show some examples to verify the validity of our algorithm.

Still consider the topology of Fig. 6. 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. 14a.

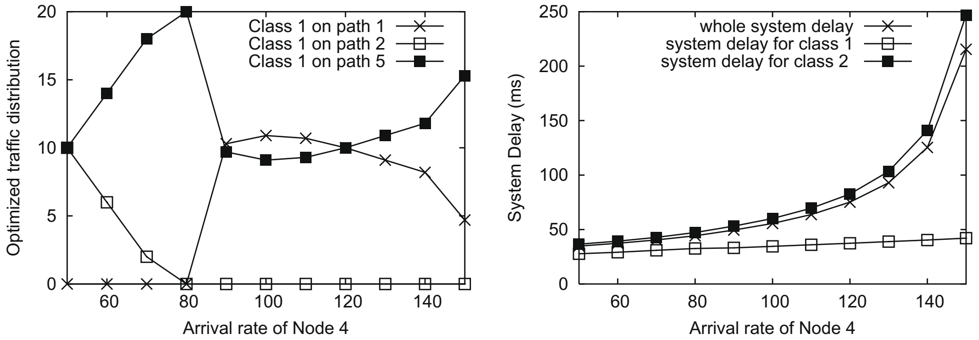
Recall what we have shown in Fig. 12b, when the traffic from node 4 is low, paths 2 and 5 are chosen, and path 2 is favored over path 5; when the load becomes heavy, path 1 will also be taken. However, we can see that path 2 is not favored by the high priority traffic. In fact, path 5 and path 1 are better choices. The reason behind this is that the load on paths 1 and 5 is much higher than path 2 due to the traffic originating from nodes 3 and 7, which makes path 2 a better choice to distribute traffic from node 4. However, the interference on paths 1 and 5 is even lighter than on path 2, making these paths favorable for high priority traffic.

Another interesting phenomenon is that although path 5 tends to take all of high priority traffic (when $\lambda_4 \leqslant 80$), 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



(a) Optimized system delay when λ_4 changes.
(b) Supported traffic with each single path.

Fig. 13. Optimized system delay and supported traffic.



(a) High priority traffic distribution vs. λ_4.
(b) Delay for multi class vs. λ_4.

Fig. 14. Optimized system behavior for changes in λ_4.

Please cite this article in press as: J. Zhou, K. Mitchell, A scalable delay based analytical framework for CSMA/CA wireless mesh networks, Comput. Netw. (2009), doi:10.1016/j.comnet.2009.05.013

performance of the whole system, high priority traffic will sacrifice a little bit of performance.

In Fig. 14b, the corresponding system delay for class 1 and 2 and all the 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.

5.5.1. Remarks 2

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

6. Conclusion

In this paper, the concept of neighboring nodes 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, an algorithm is 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 allows for a 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, for multiple classes of traffic, optimizing overall network delay while providing QoS for high priority traffic becomes possible.

With the closed form representations for queueing system utilization and response time, another important class of study is the 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

[1] I. Akyildiz, X. Wang, W. Wang, Wireless mesh networks: a survey, Journal of Computer Networks 47 (2005) 455–487.
[2] F.A. Tobagi, Modeling and performance analysis of multihop packet radio networks, Proceedings of the IEEE 75 (1) (1987) 135–155.
[3] B.M. Leiner, A simple model for computation of packet radio network communication performance, IEEE Transactions on Communications 28 (12) (1980) 2020–2023.
[4] A. Silvester, 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, V. Sahin, Throughput analysis in multihop CSMA packet radio networks, IEEE Transactions on Communications COM-35 (3) (1987) 267–274.
[6] M. Carvalho, J.J. Garcia-Luna-Aceves, A scalable model for channel access protocols in multihop ad hoc networks, in: Proceedings of the MOBICOM, Philadelphia, PA, 2004.
[7] X. Wang, 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.
[8] M. Garetto, T. Salonidis, E. Knightly, Modeling per-flow throughput and capturing starvation in CSMA multi-hop wireless networks, IEEE/ACM Transactions on Networking 16 (4) (2008) 864–877.
[9] K. Medepalli, F.A. Tobagi, Towards performance modeling of IEEE 802.11 based wireless networks: a unified framework and its applications, in: Proceedings of the Conference on Computer Communications (IEEE Infocom), Barcelona, Spain, 2006.
[10] G. Bianchi, Performance analysis of the IEEE 802.11 distributed coordination function, IEEE Journal on Selected Areas in Communications 18 (3) (2000) 535–547.
[11] A. Kumar, E. Altman, D. Miorandi, M. Goyal, New insights from a fixed-point analysis of single cell IEEE 802.11 WLANs, in IEEE/ACM Transactions on Networking 15 (3) (2007) 588–601.
[12] L. Kleinrock, F. Tobagi, Packet switching in radio channels: part I – carrier sense multiple-access modes and their throughput-delay characteristics, IEEE Transactions on Communications 23 (12) (1975) 1400–1416.
[13] F. Tobagi, 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 23 (12) (1975) 1417–1433.
[14] F. Cali, M. Conti, E. Gregori, Dynamic tuning of the IEEE 802.11 protocol to achieve a theoretical throughput limit, IEEE/ACM Transactions on Networking 8 (6) (2000) 785–799.
[15] L. Lipsky, Queueing Theory – A Linear Algebraic Approach, Macmillan Publishing Company, 1992.
[16] K. Jain, J. Padhye, V. Padmanabhan, L. Qiu, Impact of interference on multi-hop wireless network performance, Mobicom’03 (2003) 66–80.
[17] D. Bertsekas, R. Gallager, Data Networks, Prentice Hall, 1992.
[18] A. Cobham, Priority assignments in waiting line problems, Operations Research 2 (1954) 70–76.
[19] M.R. Pearlman, Z.J. Haas, P. Sholander, S.S. Tabrizi, On the impact of alternate path routing for load balancing in mobile ad hoc networks, in: Proceedings of the First ACM International Symposium on Mobile Ad hoc Networking and Computing (Mobihoc), Boston, Massachusetts, 2000.
[20] A. Valera, W. Seah, S. Rao, Cooperative packet caching and shortest multipath routing in mobile ad hoc networks, in: Proceedings of the Conference on Computer Communications (IEEE Infocom), San Francisco, CA, 2003.
[21] M. Mosko, J. Garcia-Luna-Aceves, Multipath routing in wireless mesh networks, in: First IEEE Workshop on Wireless Mesh Networks, Santa Clara, CA, 2005.
[22] X. Du, D. Wu, W. Liu, Y. Fang, Multiclass routing and medium access control for hetrogeneous mobile ad hoc networks, IEEE Transactions on Vehicular Technology 55 (1) (2006) 270–277.
[23] J. Zhou, K. Mitchell, A delay based multipath optimal route analysis for multi-hop CSMA/CA wireless mesh networks, in: International Teletraffic Congress (ITC) 20, Ottawa, June 17–21, 2007.
[24] J. Zhou, K. Mitchell, A single node decomposition based analytic model for multiclass route optimization in wireless mesh networks, in: Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC) HongKong, March 11–15, 2007.
[25] X. Yang, N. Vaidya, A wireless MAC protocol using implicit pipelining, IEEE Transactions on Mobile Computing 5 (3) (2006) 258–273.
[26] T. Saadawi, A. Ephremides, Analysis, stability, and optimization of slotted Aloha with a finite number of buffered users, IEEE Transactions on Automatic Control 26 (3) (1981) 680–689.

Please cite this article in press as: J. Zhou, K. Mitchell, A scalable delay based analytical framework for CSMA/CA wireless mesh networks, Comput. Netw. (2009), doi:10.1016/j.comnet.2009.05.013

[27] H. Takagi, L. Kleinrock, Optimal transmission ranges for randomly distributed packet radio terminals, IEEE Transactions on Communications COM-32 (3) (1984) 246–257.

[28] <http://www.phila.gov/wireless><http://www.houstontx.gov/it/wirelessrfp.html><http://english.taipei.gov.tw/rdec/index.jsp? categid=2938><http://www.hketosf.gov.hk/sf/ehk/ehongkong47/wi_fi.htm>.

[29] S. Xu, T. Saadawi, Does IEEE 802.11 MAC protocol work well in multi-hop wireless ad hoc networks?, IEEE Communications Magazine 39 (6) (2001) 130–137.

[30] R. Gupta, J. Walrand, Impatient backoff algorithm: fairness in a distributed ad-hoc MAC, in: Proceedings of ICC, 2007, pp. 3562–3569.

[31] N. Hegde, A. Proutiere, Packet and flow level performance of wireless multihop networks, in: Proceedings of IEEE GLOBECOM, 2006.

Jiazhen Zhou is a Postdoctoral researcher in Department of Systems and Computer Sciences at Howard University. He received his B.S. degree in Mathematics from ShanDong University, China in 1995, and M.S. degree in Automatic Control from Shenyang Institute of Automation, Chinese Academy of Sciences in 1998, and Ph.D. degree in Telecommunication Networking at University of Missouri-Kansas City, United States in 2008.

His research interests include modeling and performance analysis of wireless networks, queueing theory, wireless mesh networks, emergency communications, and delay tolerant networks.

Ken Mitchell is currently an Associate Professor in the department of Computer Science Electrical Engineering at the University of Missouri – Kansas City, where he has been on the faculty since 1999. He received a Bachelor of Music Degree in performance from Indiana University in 1979 and a Master of Music in performance in 1980. He received a M.Sc. and a Ph.D. in Computer Science from the University of Missouri – Kansas City in 1994 and 1997 respectively. His research interests include analytic modeling of telecommunications and computer networks and he specializes in Linear Algebraic Queueing Theory.

Please cite this article in press as: J. Zhou, K. Mitchell, A scalable delay based analytical framework for CSMA/CA wireless mesh networks, Comput. Netw. (2009), doi:10.1016/j.comnet.2009.05.013