*Manuscript

Fair packet scheduling in Wireless Mesh Networks

Faisal Nawab[a], Kamran Jamshaid[b], Basem Shihada[b,1], Pin-Han Ho[c]

[a] *University of California Santa Barbara, Santa Barbara CA USA*
[b] *King Abdullah University of Science and Technology, Thuwal, Saudi Arabia*
[c] *University of Waterloo, Waterloo ON Canada*

Abstract

In this paper we study the interactions of TCP and IEEE 802.11 MAC in Wireless Mesh Networks (WMNs). We use a Markov chain to capture the behavior of TCP sessions, particularly the impact on network throughput due to the effect of queue utilization and packet relaying. A closed form solution is derived to numerically determine the throughput. Based on the developed model, we propose a distributed MAC protocol called Timestamp-ordered MAC (TMAC), aiming to alleviate the unfairness problem in WMNs. TMAC extends CSMA/CA by scheduling data packets based on their age. Prior to transmitting a data packet, a transmitter broadcasts a request control message appended with a timestamp to a selected list of neighbors. It can proceed with the transmission only if it receives a sufficient number of grant control messages from these neighbors. A grant message indicates that the associated data packet has the lowest timestamp of all the packets pending transmission at the local transmit queue. We demonstrate that a loose ordering of timestamps among neighboring nodes is sufficient for enforcing local fairness, subsequently leading to flow rate fairness

Email address: basem.shihada@kaust.edu.sa (Basem Shihada)
[1] Phone: +966-2-808-0332, Fax: +966-2-808-0332

in a multi-hop WMN. We show that TMAC can be implemented using the control frames in IEEE 802.11, and thus can be easily integrated in existing 802.11-based WMNs. Our simulation results show that TMAC achieves excellent resource allocation fairness while maintaining over 90% of maximum link capacity across a large number of topologies.

Keywords: Wireless Mesh Networks, 802.11, TCP, fairness

1. Introduction

Wireless Mesh Networks (WMNs) have been proposed as a low-cost alternative for last mile access [1]. These dynamic, multi-hop networks can be built with commodity hardware (including off-the-shelf IEEE 802.11 radios) and open source software. A typical WMN is composed of distributed Mesh Points (MPs) that form a multi-hop backhaul. MPs may connect with multiple other MPs within their radio range. Some MPs have a wired back-channel to the public Internet; these act as gateway nodes and bridge traffic between the WMN and the Internet. End-users often communicate with their closest MP to access the Internet.

Multi-hop wireless networks, including WMNs, exhibit flow rate unfairness among competing nodes [2, 3, 4]. With backlogged traffic, the impact of flow unfairness can be significant and can lead to starvation for flows two or more hops away from the gateway. This problem is observed even with TCP, which is designed for fair allocation of network resources. A better understanding of the interaction between TCP congestion control algorithm and IEEE 802.11 MAC in a WMN is important to address the fairness problem. An analytical model that successfully predicts TCP flow characteristics can isolate the causes of such performance degradation. However, this is a challenging task since multi-hop wireless networks are subject to losses from collisions as well as random channel noise, which may eventually degenerate to the point of starvation.

In this paper we propose an analytical model that captures the behavior of competing TCP flows in a 802.11-based WMN. Our model uses the cumulative number of TCP data packets in the network for a given TCP flow. These are the packets generated by that flow but not yet delivered to the destination. At any given time, these packets are distributed over various queues along the path between the source and destination. For simplicity, we model the network as a closed system where the state of a flow is represented by the cumulative number of data packets existing in the network for a particular flow (called the cumulative network queue). Furthermore, our model uses the number of transmissions required by a particular flow from the perspective of the gateway (called the transmission step). Since transmissions beyond the carrier sense range of the gateway can be made concurrently while the gateway is transmitting, the transmission step for the majority of the nodes in a network varies between 1 and 3.

In this paper we improve flow rate fairness by proposing a new MAC scheduling protocol, called Timestamp-ordered MAC (TMAC). TMAC addresses the fairness and throughput degradation in WMNs using the age of a packet as a metric for prioritizing its scheduling. TMAC is based on the mutual exclusion algorithm of Lamport [5]. Lamport algorithm uses request timestamps to ensure that the node with the earliest request is served next. The algorithm relies on an explicit exchange of control messages to make all nodes aware of the network state. These communication requirements are more suited for fully-connected wired networks, but may scale poorly in large WMNs. TMAC addresses these challenges by limiting the exchange of these control messages to a set of neighboring nodes that contend for channel access. It improves fairness by prioritizing the transmission of packets that are generated before others (*i.e.,* have a larger age). We show that for backlogged TCP flows, scheduling packets according to their age when coupled with a specialized queuing discipline results in absolute[2] flow rate fairness.

The remainder of this paper is organized as follows: We provide an overview of the related work in Section 2. Our proposed model is described in Section 3, including a discussion of the causes of unfairness. In Section 4, we introduce TMAC and describe its various functional blocks. In Section 5, we describe the design challenges in implementing TMAC over 802.11 radios and propose optimizations for improving its behavior. We validate our model in Section 6. Furthermore, we present a simulation study of performance analysis of TMAC. We conclude with a discussion and a summary in Section 7.

2. Related Work

There has been a significant amount of research on modeling wireless links characteristics. This includes models for describing the detailed behavior of

[2] Absolute fairness is the equal distribution of resources among competing nodes.

random access protocols in wireless networks [6, 7]. These studies, however, assume that all nodes are fully aware of the network state, which is only feasible in the presence of additional signaling mechanisms on top of a distributed 802.11 WMN. Multi-hop wireless network models have also been proposed in [8, 9] and [2]. These models capture the MAC protocol interactions by assuming a connection-less backlogged traffic. Other models account for TCP traffic by considering the impact of an extra flow caused by the acknowledgment (ACK) packets. However, rather than capturing the interaction of TCP and MAC, these studies model the aftermath of these interactions. Some previously proposed models capture the interaction of MAC and TCP in wireless networks [10, 11]. We are mainly interested in the objective of [11], where the effect of multi-hop relaying and TCP data/ACK packets exchange are explicitly modeled. However, the work in [11] only considers a two-hop chain topology with a single flow with a conservative choice of TCP congestion window. The intractability of this limits its analysis to more reasonable multi-hop scenarios. However, in this paper, we focus on larger WMNs topologies with a larger number of flows. Thus, we maintain the objective of the work in [11, 12, 13] with a more tractable model that is applicable to complex scenarios.

A number of proposals modify the conventional backoff scheme to incorporate fairness or other objectives [14, 15, 16]. For example, some work modify the backoff scheme to achieve service differentiation and prioritization [17, 18]. In general, a transmission with a higher priority is assigned a lower MAC contention window, and vice versa. DFS [16] is an example of a protocol using backoff prioritization with a fairness objective. It is a fully distributed protocol that tries to emulate the centralized SCFQ [19]. The priority of a transmission is dependent on a timestamp associated with the corresponding packet. The authors postulate that giving higher priorities to lower finish timestamps will lead to SCFQ fairness. To translate that objective to an appropriate backoff assignment mechanism, they proposed several schemes to map finish timestamps to backoff intervals. The simplest one is a linear scheme that is inversely proportional to the flow weight and transmission priority. Linear mapping can lead to large backoff intervals, thus leading to lower utilization of the channel. To overcome this limitation they also proposed exponential and adaptive mappings. Another example of achieving a fairness objective through backoff manipulation is in [14]. The authors propose a distributed algorithm that first estimates the fair share of medium access without global knowledge, and then assigns backoff intervals according to the estimated fair share. Manipulating parameters other than backoff interval can also be used as means to achieve fairness objectives. Such parameters are the inter-frame spacing periods (IFS), slot size, etc. These parameters are used to achieve prioritization and service differentiation [20], and can further be manipulated to lead to fairness.

There are many distributed protocols designed to achieve fairness in wireless networks [16, 21, 22, 23, 14], including WMNs [2, 24]. Most of the proposed schemes achieve fairness by limiting flow rates to the fair share of the network capacity. This requires actively maintaining network global state and flow synchronization among distributed nodes. The proposed TMAC protocol, on the other hand, does not estimate the fair rate for each flow. Instead, it establishes priority scheduling for transmitting data packets at each node according to the local condition. The use of priority scheduling has also been investigated in the literature [18, 17, 16, 21], aiming to enforce service differentiation and ensure QoS. This is made possible by taking advantage of backoff increment functions or IFS [17, 16]. However, solely relying on backoff timers may worsen the impact due to the hidden terminal and masked node problems, which leads to starvation [25]. TMAC uses the packet age as the priority metric to resolve the above deficiency. Similar to [18, 26], TMAC uses request/grant control messages to achieve higher throughput and fairness.

Finally, centralized protocols [3, 27] that only require enforcement at traffic aggregation points (*e.g.*, gateway mesh routers) have been proposed to address fairness in WMNs. However, these protocols only work with adaptive transport protocols like TCP. In contrast, extensions proposed in TMAC allow us to enforce fairness for both TCP and UDP based streams.

3. Modeling TCP in WMNs

In this section we model TCP flows over 802.11-based WMNs. The treatment here is applicable to any MAC protocol that ensures fair access to the channel. We investigate the parameters required for capturing a TCP flow's characteristics and describe a methodology for constructing a Markov chain to model these parameters. We then analyze the causes of TCP unfairness.

3.1. Overview

We model TCP flows in WMNs while focusing on the fairness characteristics. Without loss of generality, our model considers a single mesh gateway. We assume that all nodes have backlogged TCP traffic destined to the gateway and the TCP streams are in a state of equilibrium (*i.e.,* all TCP flows are in the congestion avoidance phase). Similar to [11], we start by fixing the upper bound on TCP's congestion window size. This assumption is motivated by observing that TCP receivers provide an upper bound on the number of packets in transit at a given time. Later in this section, we investigate the effect of varying this limit on the rate of a flow. This will help us reason about the behavior of TCP flows with dynamic congestion windows.

Our model groups flows into a bundle that share a common queue at one-hop nodes from the gateway. We represent each such bundle as a *branch*. The relationship between resources allocated for different branches is dictated by the characteristics of the MAC protocol. With 802.11 CSMA/CA MAC, nodes have equal probability of accessing the channel. Assuming a uniform random node deployment around the gateway, each branch receives an equal share of the resources. For example, if two nodes, A and B, are in the transmission range of the gateway, the resources allocated for nodes in the branch sharing queue A is equal to the resources allocated to nodes sharing branch B, despite the difference in the respective number of nodes.

Given the assumptions above, the parameters necessary for modeling TCP throughput are the utilization of the network queues at various nodes and the order of packets in the queues (relative to their source and destination) for both data and ACK packets. However, deriving a closed form solution would be hard for topologies with a large number of nodes and active flows. Thus, the following two assumptions are introduced to simplify the problem. First, we model the queue utilization without considering the order of the packets; in other words, only the number of packets for each flow is taken into account, while the order is then considered by calculating possible permutations and assigning the transition probabilities accordingly. For example, consider a queue with three packets, two packets belonging to flow A and one packet belonging to flow B. Representing all the permutations will require three states, namely a state with B's packet in the head, in the middle, or in the tail of the queue. However, we simplify this notion by representing all those three states by a single state $(2, 1)$. Then, in our calculation of transition probability we consider that it is equally likely to be in any of these states. Thus, a transmission of a packet from a one-hop node to the gateway belonging to flow A is twice as likely compared to a transmission of a packet belonging to flow B. Second, by observing the behavior of single sink networks, we found that the queue belonging to the closest node to the gateway exhibits significant utilization. Thus, we model the cumulative network queue, which is a unified conceptual queue that contains all the packets in the network. The multi-hop effect is incorporated in this simplification by assigning the transition probabilities according to the number of necessary transmission steps in the network. Number of transmission steps is correlated with the cumulative wait times experienced by packets. Thus, flows originating from distant nodes have larger transmission steps value.

3.2. The Model

A Markov chain is used to model the TCP behavior. The system state is represented by the cumulative network queue utilization. Each state represents the number of data packets for each flow in the queue. We consider a 2-hop parking lot topology with two flows, namely flow 1 originating from the 1-hop away node and flow 2 originating from the 2-hops away node. We describe the network as the process $\{P_1, P_2\}$, where P_n indicates the number of data packets belonging to the n^{th} flow that exists in the network (*i.e.*, P_1 and P_2 represent the number of packets queued for the 1 and 2-hop flows). The model is a Markov chain with n-dimensions, where n is the number of flows. We use W_n to denote the TCP congestion window of the n^{th} flow. Thus an equivalent state description of the network is the number of ACK packets in the network, *i.e.,* the process $\{W_1 - P_1, W_2 - P_2\}$. State transitions are governed by three aspects: (1) the number of nodes competing for channel access; (2) the relative number of data and ACK packets in the network, which is further coupled with the probability of accessing the channel; and (3) the multi-hop effect, which is further modeled as additional self loops with an equal share of the transition probability of the original link. We assume that all nodes have an equal chance to access the channel. This assumption holds given that we only model the cumulative network queue. Note that the number of self loops corresponds to the number of transmission steps which in turn affects the cumulative network queues. For example, in a 2-hop parking lot topology a data transmission of a packet belonging to flow 2 is represented as a transition from state $\{P_1, P_2\}$ to state $\{P_1, P_2 - 1\}$ with probability

$$\frac{P_2}{k_2.l_{\{P_1,P_2\}}.(P_1 + P_2)} \tag{1}$$

where l_η is the number of stations competing for the channel for a given state, *e.g.* $\eta = \{P_1, P_2\}$. k_j is the number of transmission steps needed for flow j.

We summarize the possible transmissions for a packet belonging to flow i as follows,

- *Data Packet:* Transition from state $\{P_1, \ldots, P_i, \ldots\}$ to $\{P_1, \ldots, P_i{-}1, \ldots\}$ with probability $\frac{P_i}{k_i.l_\eta.(\sum_{j=1}^{n} P_j)}$, given that $P_i > 0$.
- *ACK Packet:* Transition from state $\{P_1, \ldots, P_i, \ldots\}$ to $\{P_1, \ldots, P_i{+}1, \ldots\}$ with probability $\frac{W_i - P_i}{k_i.l_\eta.(\sum_{j=1}^{n} (W_j - P_j))}$, given that $(W_i - P_i) > 0$.

The assignment of the number of competing nodes is as follows,

- $l_\eta = 1$, given that $\sum_{j=1}^{n} P_j = 0$ or $\sum_{j=1}^{n} P_j = \sum_{j=1}^{n} W_j$.
- $l_\eta = 2$, otherwise.

From the above we observe that the number of competing nodes is highly affected by the existence of data/ACK packets in the cumulative network queue. The number of transmissions necessary for a packet (*i.e.,* k_j) determines the number of transmission steps affecting the modeled queues. In other words, transmissions that do not contribute to the relative utilization of the cumulative network queue are discarded and not considered. This is as a result of our

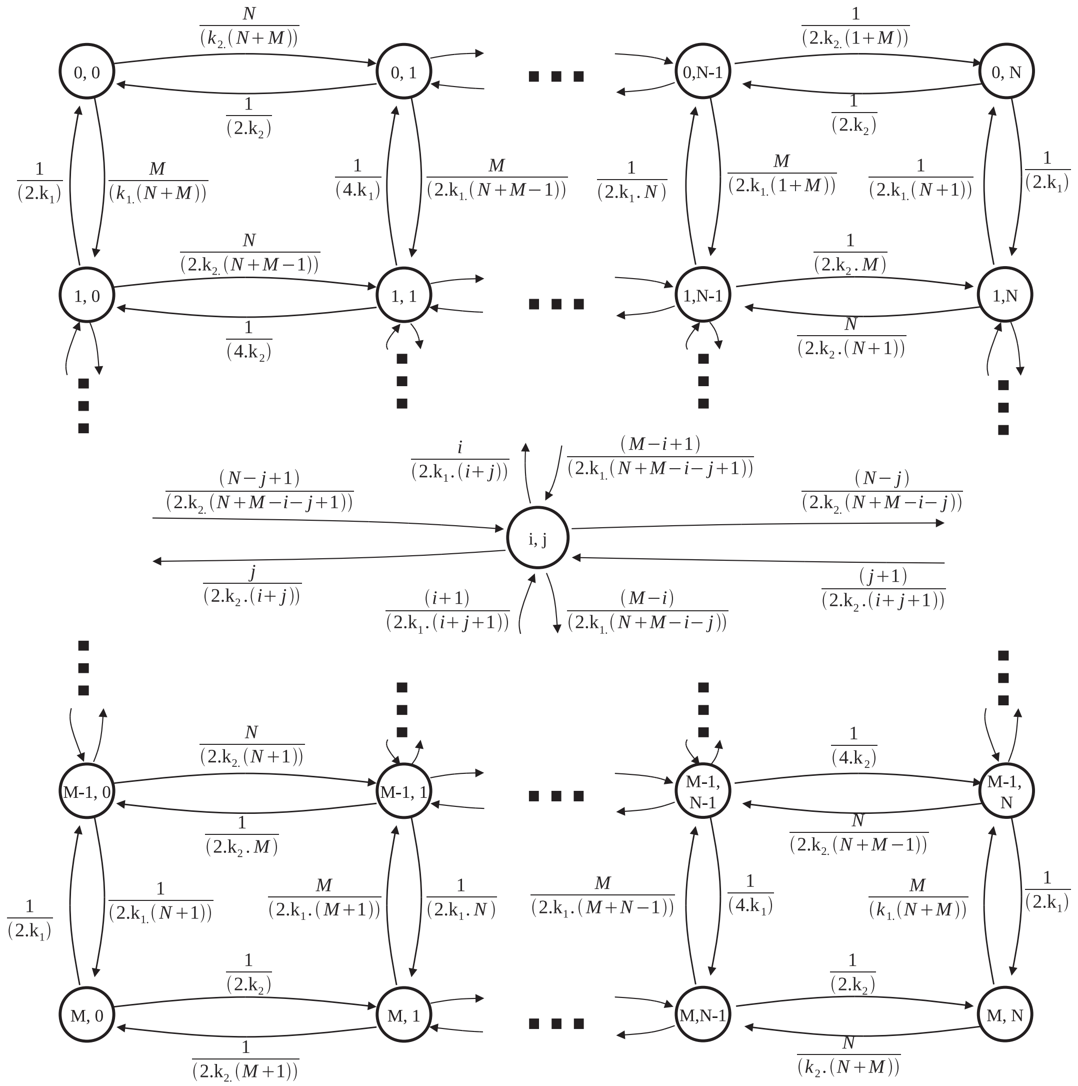

$\frac{N}{(k_{2.}(N+M))}$
$\frac{1}{(2.k_2)}$
$\frac{1}{(2.k_{2.}(1+M))}$
$\frac{1}{(2.k_2)}$
0, 0
0, 1
0,N-1
0, N
$\frac{1}{(2.k_1)}$
$\frac{M}{(k_{1.}(N+M))}$
$\frac{1}{(4.k_1)}$
$\frac{M}{(2.k_{1.}(N+M-1))}$
$\frac{1}{(2.k_1.N)}$
$\frac{M}{(2.k_{1.}(1+M))}$
$\frac{1}{(2.k_{1.}(N+1))}$
$\frac{1}{(2.k_1)}$
$\frac{N}{(2.k_{2.}(N+M-1))}$
$\frac{1}{(4.k_2)}$
$\frac{1}{(2.k_2.M)}$
$\frac{N}{(2.k_2.(N+1))}$
1, 0
1, 1
1,N-1
1,N
$\frac{i}{(2.k_1.(i+j))}$
$\frac{(M-i+1)}{(2.k_{1.}(N+M-i-j+1))}$
$\frac{(N-j+1)}{(2.k_{2.}(N+M-i-j+1))}$
$\frac{(N-j)}{(2.k_{2.}(N+M-i-j))}$
i, j
$\frac{j}{(2.k_2.(i+j))}$
$\frac{(j+1)}{(2.k_2.(i+j+1))}$
$\frac{(i+1)}{(2.k_1.(i+j+1))}$
$\frac{(M-i)}{(2.k_{1.}(N+M-i-j))}$
$\frac{N}{(2.k_{2.}(N+1))}$
$\frac{1}{(2.k_2.M)}$
$\frac{1}{(4.k_2)}$
$\frac{N}{(2.k_{2.}(N+M-1))}$
M-1, 0
M-1, 1
M-1, N-1
M-1, N
$\frac{1}{(2.k_1)}$
$\frac{1}{(2.k_{1.}(N+1))}$
$\frac{M}{(2.k_1.(M+1))}$
$\frac{1}{(2.k_1.N)}$
$\frac{M}{(2.k_1.(M+N-1))}$
$\frac{1}{(4.k_1)}$
$\frac{M}{(k_{1.}(N+M))}$
$\frac{1}{(2.k_1)}$
$\frac{1}{(2.k_2)}$
$\frac{1}{(2.k_{2.}(M+1))}$
$\frac{1}{(2.k_2)}$
$\frac{N}{(k_2.(N+M))}$
M, 0
M, 1
M,N-1
M, N

Figure 1: Transition diagram for two-hop parking lot topology. Congestion windows for the farthest and closest flow are M and N, respectively. The state (P_1, P_2) denotes that the network has P_1 data packets for flow 1 and P_2 data packets for flow 2, where a upward transition corresponds to the transmission of a data packet of flow 1, a downward transition is for an ACK transmission of flow 1, a leftward transition is for the transmission of a data packet of flow 2, and a rightward transition corresponds to an ACK transmission of flow 2

earlier observation that the queues closer to the gateway have significantly higher utilization than that of the other queues. Unless mentioned otherwise, in the rest of our discussion we assume that k_j equals to 1 for flows originating from any one-hop away node, and 2 otherwise. Figure 1 shows a generalization of the transition diagram of a 2-hop parking lot topology.

3.3. Model Analysis

Relative throughput of participating flows is an important performance metric for our study. We examine a network with n flows and derive a closed-form equation of the throughput of each flow using our model. First, we solve our model to come up with a formula of state probabilities. The Markov chain used as a model of our system is *reversible*. A proof of the model's reversibility is given in the Appendix. A reversible Markov chain satisfies the following relation between any two states, I and J

$$\pi_I Pr_{I \to J} = \pi_J Pr_{J \to I} \tag{2}$$

where $Pr_{I \to J}$ denotes the transition probability from state I to state J and π_I is the probability of state I. In other words, the probability of state I can be calculated from any other state, J, by only knowing the transition probabilities, *i.e.*, $\pi_I = \frac{Pr_{J \to I}}{Pr_{I \to J}} \pi_J$, given that those transition probabilities are greater than zero. We use the notation $\rho_{I \to J}$ to be equal to $\frac{Pr_{J \to I}}{Pr_{I \to J}}$ and we call it the intensity of transitioning from state I to J. The reversibility of the model allows us to calculate the probability of a state by transitions from state $\emptyset$, where the state

$\emptyset$ represents a state where all flows have 0 data packets, to the desired state through each dimension. For example, for our model's Markov chain of three flows, the probability of state $(1,0,2)$ is given by

$$\pi_{(1,0,2)} = \rho_{(0,0,0)\to(1,0,0)}\rho_{(1,0,0)\to(1,0,1)}\rho_{(1,0,1)\to(1,0,2)}\pi_{\emptyset=(0,0,0)} \tag{3}$$

where the first transition intensity, $\rho_{(0,0,0)\to(1,0,0)}$, accounts for calculating the transition in the first dimension (*i.e.*, first flow) and the next two transition intensities account for calculating the transitions in the third dimension. Notice that we did not include any transition intensities for the second dimension since the destination state, $(1,0,3)$, has the number of packets for flow two equal to the source state's number of packets for flow two, *i.e.*, 0.

Consider now the general case, where we want to calculate the probability of a state I, π_I, where $I = (i_1, \ldots, i_n)$, given $\pi_\emptyset$. We showed that this can be derived by finding a sequence of states from state $\emptyset$ to I. Call the set of ordered states in the sequence to be $\alpha_{\emptyset\to I}$. The probability of state I is given by the product of intensities of the ordered sequence $\alpha_{\emptyset\to I}$ multiplied by $\pi_\emptyset$, hence

$$\pi_I = \rho_{0\to I}\pi_\emptyset \tag{4}$$

Every state is reachable from $\pi_\emptyset$. A straight forward methodology to find such a sequence is to take each flow in turn and traverse through its direction. Having found the probability of any state giving $\pi_\emptyset$, observe that the total probability must equal to one, hence $\sum_{s\in states} \alpha_{\emptyset\to s}\pi_\emptyset = 1$. Reordering the equation for

$\pi_\emptyset$ gives

$$\pi_\emptyset = \frac{1}{\sum_{s \in states} \alpha_{\emptyset \to s}} \tag{5}$$

Substituting Equation (5) into (4) we show a closed-form solution of π_I as follows,

$$\pi_I = \frac{\alpha_{\emptyset \to I}}{\sum_{s \in states} \alpha_{\emptyset \to s}} \tag{6}$$

From Equation (6) the throughput (T_j) of flow j is the sum of probabilities of transmitting a data packet belonging to flow j for all states, hence

$$T_j = \sum_{s \in states} Pr_{s \to s-1_j} \pi_s \tag{7}$$

where state $s-1_j$ is the state equal to state s for all flows except flow j where the number of packets is short by one. Thus, $s-1_j$ represents the destination state after a data packet transmission of flow j at state s. Obtaining the fairness measure is straightforward by applying Equation (7) to the desired fairness model.

Identical congestion windows. We now examine the case where flows have identical congestion window values. This is of special interest due to the limit imposed by TCP's receiver window, leading congestion window values to converge to the same value in the equilibrium state. This introduces an additional symmetry in our model. Consider being at a state S. Now consider states I and J. State I is identical to state S except that for a flow i the number of packets in the state are larger by c packets. The same is true for state J but this time for flow j. Since both flows have the same congestion window value and the source state is the same, the product of the intensities at the sequences from S to I and J, $\alpha_{S \to I}$ and $\alpha_{S \to J}$, are equal. And by reversibility, we have $\pi_I = \pi_J$.

Now consider the general case where the source state is $\emptyset$ and the destination states are I and J. We want to establish a condition to equate their probabilities, π_I and π_J. We established that because of reversibility, the probability of the destination state depends on the intensities of a sequence of states from another state. If the source state is the same, hence $\emptyset$, two states, I and J, have equal probability if there exist sequences, $\alpha_{\emptyset \to I}$ and $\alpha_{\emptyset \to J}$, where the product of intensities through them are equal. We showed in the last paragraph that starting from the same state, an equal number of transitions in the same dimension lead to a state with equal probability to any other state reached with the same number of transitions on another dimension. The reason for this is that when transitioning in one direction, the intensity value only depends on the number of packets and window size of the flow representing the dimension, and the sum of the number of packets of all flows in the source state. By this observation, consider the sequences leading to I and J from $\emptyset$. Assume that our construction of $\alpha_{\emptyset \to I}$ is done by taking flows one by one and transitioning through its dimension. For the first flow assume that there are i_1 packets for the destination state I. Thus we transition to state $(i_1, 0, \ldots, 0)$, call it state $\emptyset + (i_1)_1$. Remember that the notation X_y refer to a difference of X packets in the y'th entry of the state, *i.e.*, corresponding to flow y. However, by our observation, the probability of that state is equal to states $\emptyset + (i_1)_s$, where s is any other flow. Thus, if state J contains any flow with a number of packets equal to i_1 we choose it as the first flow to transition through, *e.g.*, if flow j_k consists of i_1 packets we transition through a sequence to state $\emptyset + (i_1)_k$ that will have an equal probability to state $\emptyset + (i_1)_1$. Now we pick the second flow for state I and transition through it starting from state $\emptyset + (i_1)_1$. Thus, we transition to state $\emptyset + (i_1)_1 + (i_2)_2$. Now, if we find another flow in J that has i_2 packets we can choose this flow to transition through. Call this flow l. Thus, for state J's calculation the second traversal is to state $\emptyset + (i_1)_k + (i_2)_l$. Since the starting state for both I and J have the same number for the sum of packets for the destination state of the first traversal, the destination states of the second traversal are also equal. By taking this to all dimensions, we arrive to the conclusion that states I and J have equal probability if there is a one-to-one mapping from each flow value in I to a *distinct* flow value in J that have the same number of packets but not necessarily the same flow. In other words, if the frequency (*i.e.*, number of occurrences) of each value across states are the same, then the states are equivalent, hence

$$\pi_{(\{i_1,\ldots,i_n\})} = \pi_{(\{j_1,\ldots,j_n\})}, \quad if \;\; \forall_{i_x} \exists_{unique j_y} i_x = j_y \tag{8}$$

Using the identity in Equation (8) we know that each state has other *mirroring* states, where a mirroring state satisfies the identity and thus has an equal probability. We are interested in one special kind of mirroring between states. Consider two flows I and J. For any state we define an $i - j - mirroring$ to be achieved by switching the number of packets at flow i with flow j. For example, the $1 - 2 - mirroring$ of state $(1, 4, 2)$ is state $(4, 1, 2)$. Given Equation 8, these two states have equal probability. Furthermore, consider the transition probability corresponding to a data packet of flow I at a state X and the transition probability corresponding to a data packet of flow J at the $i - j - mirroring$ state of X, called X'. Remember that the transition probability is given by $\frac{P_j}{k_j l \sum_{i \in flows} P_i)}$ for flow j, where P_j is the number of packets of flow j, k_j is the transmission step of j and l depends on the number of competing nodes as defined earlier in the model. Now consider the following state transition probabilities, $Pr_{X \to X-1_i}$ and $Pr_{X' \to X'-1_j}$. The ratio of the first transition probability to the next is $\frac{k_j}{k_i}$. This is because the number of packets that correspond to the flow transmitting a packet is the same, hence the upper term, the sum of the number of packets is also equal, and the number of competing nodes is also the same since the number of flows with 0 packets is the same for both states. Combining this identity with our previous observation that $i - j - mirroring$ states have equal probabilities, consider calculating the ratio of two throughput equations of two different flows, I and J, in one model. Since for any state there is an $i - j - mirroring$ state, and since we are considering flows I and J transmissions, the throughput term of every state cancel with the throughput term of the $i - j - mirroring$ state except for the transmission step value. Thus, we have

$$\frac{T_i}{T_y} = \frac{k_y}{k_i} \tag{9}$$

Equation 9 shows that if two flows with the same congestion window compete for channel access, their relative throughput depends only on the number of transmission steps.

Summary. We showed in this section that we are able to obtain a closed-form solution for the state probabilities of our model, given in Equation 6. Throughput of a certain flow can then be calculated via our model using Equation 7. Finally, we showed that for an instance of the model where all flows have identical congestion window values, the ratio between throughput values of different flows depends only on the transmission step values of those flows (Equation 9).

4. Timestamp-ordered MAC

We conclude from our discussion in Section 3.3 that fairness is affected by two factors, namely the difference in congestion window, and the number of transmission steps for each flow. Thus, one way to achieve fairness is by eliminating these two factors. We propose a MAC-layer solution to minimize the difference in the congestion window between various flows and to force a transmitted packet to reach its destination atomically (*i.e.,* make the number of transmission steps equal to one). In this section we provide an overview of TMAC, including a description of the proposed TMAC priority scheduling mechanism and its impact on flow rate fairness. We also describe the role of queueing discipline in improving fairness for TCP flows.

4.1. Overview

The fundamental idea behind TMAC is to schedule packets based on their age as identified in their timestamps. In wired networks, control messages can be used to achieve consensus between nodes. However, any message exchange requiring global co-ordination incurs a significant overhead in multi-hop WMNs. Our proposed TMAC protocol addresses this by limiting the exchange of control messages to a subset of direct neighboring nodes only, (*i.e.,* one-hop away). We argue that the single-sink property of WMNs allows us to limit ordering enforcement on nodes with a parent-child relationship[3]. This local ordering can be achieved by an explicit exchange of control messages between nodes. Each node maintains a table of its child nodes. Whenever a node has a packet to send, it advertises the priority (*i.e.,* age) of the packet in the head of the transmission queue by multicasting a *request* message to its child nodes. When a child receives this message it responds with a *grant* message only if the requesting node has a higher priority than any packet pending transmission at the child node. When grant messages are received from all children, the packet is transmitted.

4.2. Impact on starvation and global fairness

TMAC uses timestamps to measure the packet age and influence its scheduling priority. These timestamps enforce a local ordering between neighboring

[3] A parent node is the next node on the route towards the gateway.

nodes. For example, a node cannot transmit a packet until the packet has a higher priority (*i.e.,* a lowest timestamp) than the packets of its child nodes. The mechanics of TMAC require a transmitter to poll its children and seek confirmation that they do not have older packets awaiting transmission. This explicit polling ensures that a node can not starve its children at the cost of its own transmission.

The local ordering enforced by TMAC creates a backpressure that translates into global ordering in WMNs with a single gateway. Since all flows traverse this gateway, the local ordering enforced on one-hop neighbors of the gateway propagates to all flows traversing them. For example, suppose nodes N_1 and N_2 are one-hop away from the gateway and nodes N_3 and N_4 are two or more hops away. Suppose that there exist flows f_3 and f_4 originated from nodes N_3 and N_4 respectively. The local ordering between N_1 and N_2 creates a backpressure such that packets of f_3 and f_4 are relayed according to their priorities. The time for backpressure to propagate within the network is a function of the node depth. This determines the latency incurred in converging distant nodes to their fair rate. We evaluate these flow rate convergence characteristics of TMAC in Section 6.

4.3. TCP fairness and queuing discipline

TCP flow rate is clocked with its Round Trip Time (RTT). With a faster feedback loop, nodes closer to the gateway can quickly build up larger TCP congestion windows compared to distant flows. Thus, the buffers at one-hop and two-hop nodes are largely populated with packets originating locally. If we use a simple DropTail queue with FIFO discipline, distant flows will experience packet drops from queue overflows when they reach these two-hop and one-hop nodes. Thus, the queueing discipline is integral in improving the fairness of TCP streams in WMNs.

TMAC uses a variant of Fair Queueing (FQ) by separating DATA packets from ACKs. Since TCP ACKs are cumulative, its congestion control mechanisms may not be triggered even when some ACKs are lost as long as an ACK with a higher sequence number gets delivered. Both DATA and ACK queues are sorted by timestamps such that the packets at the head of the respective queues are the oldest packets that are next scheduled for transmission. A locally generated packet is assigned a timestamp when it reaches the head of queue. This, however, leaves the locally generated traffic vulnerable to indefinite preemption by relay packets that have already been assigned a timestamp by their source nodes. We prevent this by partitioning our queue space into *rounds* such that relay packets from a round k cannot preempt packets generated locally in round $k + 1$.

5. TMAC design for CSMA/CA radios

In this section we describe our TMAC implementation for CSMA/CA radios using the control frames in IEEE 802.11. We also propose an optimization technique to reduce the overhead of these control messages.

5.1. TMAC implementation over 802.11

TMAC can be implemented through minor modifications to the IEEE 802.11 protocol. The modifications include the design of request/grant messages, as well as associating a timestamp with a data frame through its propagation in the network.

Request/grant messages: A TMAC node requires request/grant messages to poll its neighbors about the state of packets pending for transmission. Our TMAC implementation uses modified RTS/CTS control frames to build this request/grant messaging framework. We introduce two modifications in the way RTS/CTS control frames are exchanged. First, the RTS frame is delivered to selected child nodes rather than a single designated receiver. This can be achieved either by transmitting RTS as a broadcast frame or by making the neighbors promiscuously capture the frame. Second, all neighbors receiving an RTS message should respond with a CTS message as long as the received RTS has a lower timestamp than any pending local transmission. The initial sender triggers data transmission only after receiving CTS frames from a sufficient number of children.

The proposed scheme may result in collision among CTS frames when multiple child nodes respond to an RTS. Therefore, these child nodes need to schedule their transmissions. We have implemented the scheme proposed in [28] using broadcasting and multicasting wireless transmissions and adapted it to control messages. The main idea is to append the neighbor addresses in the RTS in the order which they are expected to transmit CTS. Thus, a node responding

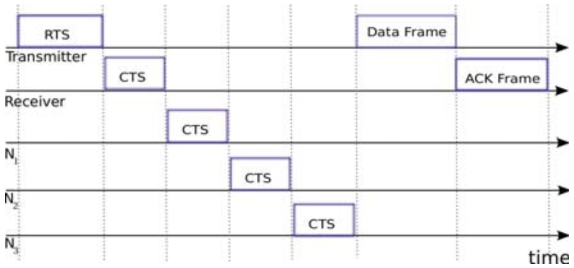


Figure 2: Multicasting control frames in IEEE 802.11 radios

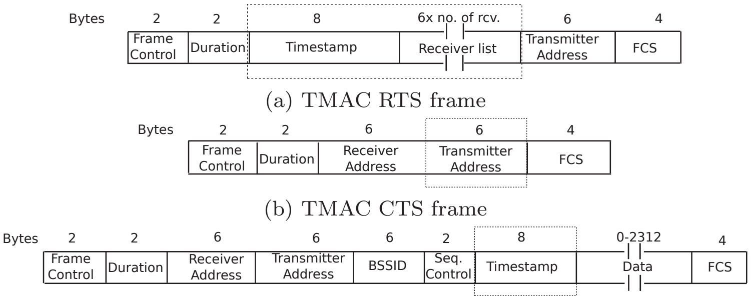
to an RTS waits for a predefined amount of time T before launching the CTS message as follows:

$$T = (order - 1) \times (CTS\ transmission\ time)$$

These modifications to the RTS frame structure not only support the scheduling of CTS transmissions, but also enables the polling of specific neighbors for their CTS messages. Figure 2 shows the exchange of these control frames between a transmitter and a receiver with three neighboring nodes N_1, N_2, and N_3.

Timestamp generation: 802.11 radios achieve time synchronization by periodically exchanging timestamp-carrying beacons between neighboring nodes. We have implemented timestamps based on the synchronized clock among nodes. Our results in Section 6 show that such synchronization is sufficient to ensure the ordering of packet transmissions required for the proposed TMAC protocol.

Revised RTS/CTS and Data frames format: We modified the 802.11 RTS/CTS and DATA frames to support TMAC protocol as shown in Figure 3.



(a) TMAC RTS frame

(b) TMAC CTS frame

(c) TMAC DATA frame

Figure 3: 802.11 modified frame structure as used in our TMAC implementation

RTS frames have been modified as follows: a Timestamp field is appended (8 Bytes); this corresponds to the Timestamp field included in the Beacon frames per the 802.11 standard specifications. The Receiver address list (6 Bytes × no. of receivers) specifies the list of child nodes that are required to respond with a CTS. The Duration field is updated such that it reflects the time required for completing the transactions, including the additional CTS transmissions from selected children.

CTS frames are appended with a Transmitter Address field (6 Bytes). This allows the RTS transmitter to differentiate between CTS frames from various children.

DATA frames are appended with a Timestamp field (8 Bytes). This allows the receiver to sort its transmit queue based on the age of the DATA packet.

5.2. Interface queue

An interface queue design satisfying the fairness requirements discussed in Section 4.3 is implemented as follows: packets arriving to a certain queue are either *fresh* (*i.e.,* locally generated) or timestamped packets (DATA or ACK).

A *fresh* packet is placed at the tail of the queue. A timestamped packet is inserted in the queue sorted according to its timestamp. Note that *fresh* packets in the queue have not been assigned a timestamp yet. At this stage, *fresh* packets should first be placed between rounds of transmissions to prevent preemption by other flows packets. Consequently, if the tail of the queue has a *fresh* packet and a timestamped packet arrived with a timestamp larger than all the other timestamped packet, then it is placed in the tail of the queue. It is then scheduled in the next round of transmissions.

5.3. Mitigating TMAC control message overhead

The control message exchange required for TMAC may cause significant performance penalty. Recall, each RTS frame triggers CTS frames from child nodes. By default, these control frames are transmitted at the base rate, further increasing the impact of this overhead. We now propose an optimization technique to alleviate this overhead.

We propose using data bursts to amortize the overhead associated with the control message exchange. This allows a node to forfeit requesting grant messages from its neighbors for a fraction of the transmissions, allowing the grants to be effective for more than one transmission. For example, a burst length of five indicates that each received grant is effective for five transmissions. Selecting the proper burst size is an important configuration parameter. Larger bursts can significantly reduce the control frame overhead, yet it may introduce short-term unfairness between flows. Figure 4 shows the effect of bursts on network utilization in a 5-hop chain topology. We define the *network utilization*

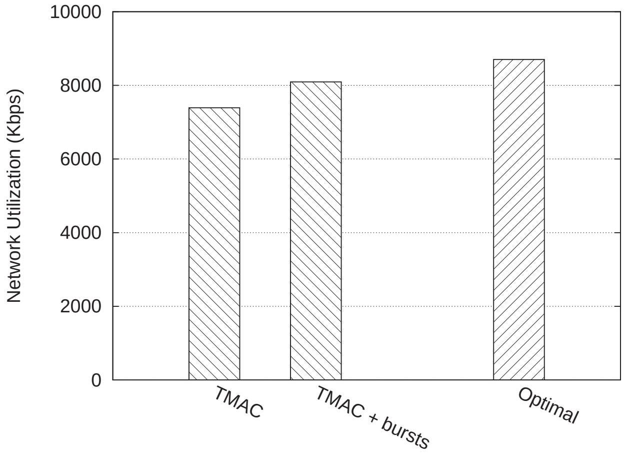


Figure 4: Impact of TMAC optimizations on the utilization of a 5-hop chain with 12 Mb/s wireless link rates

metric as $\sum_{i=1}^{N} x_i \times l_i$, where N is the number of active flows, x_i is the goodput of the i^{th} node, and l_i is the number of hops along the routing path between the i^{th} node and the gateway. Figure 4 shows that bursts can increase the network utilization by 9.5% in a 5-hop parking lot topology. Our experiments in Section 6 use TMAC with data bursts.

6. Analysis and Evaluation

We now present a simulation study to validate our proposed model and evaluate the performance characteristics of TMAC. We use the ns-3 network simulator with the simulation parameters shown in Table 1. We discard the first 20 sec. of simulation trace as the initial transients for establishing routes and populating ARP tables.

Parameter	Value
Link rate	12 Mb/s
Duration	120 sec.
MAC protocol	IEEE 802.11a
Packet size	1500 B
Interface queue size	500 packets
Routing protocol	OLSR
Transmission range	250 m
Carrier sense range	550 m

Table 1: Simulation parameters

6.1. Model Validation

We performed a set of experiments on several parking lot and grid topologies to validate our model. We used our model to numerically calculate the expected rate of each flow. This rate is then scaled by the maximum achievable throughput for a single TCP flow over a one-hop network, shown in Table 2. The measured goodput is lower than the link rate due to PHY and MAC layer overhead incurred per frame. For TCP segments, TCP ACK overhead incurs an additional penalty., reducing the achievable goodput to values shown in Table 2.

Link rate	CSMA (Mb/s)
12 Mb/s	8.5

Table 2: Measured optimal goodput for a TCP flow in a 1-hop network

Our first set of experiments is performed on a two-hop parking lot topology. The maximum congestion window of each flow is varied in both the simulation and model. Our results are shown in Figure 5. The model predicts the experimental results closely. We performed an additional set of experiments without limiting the congestion window. Since congestion windows converge to TCP's receiver window, these results were approximately identical to those obtained by limiting the maximum congestion window size for the two flows to the same value.

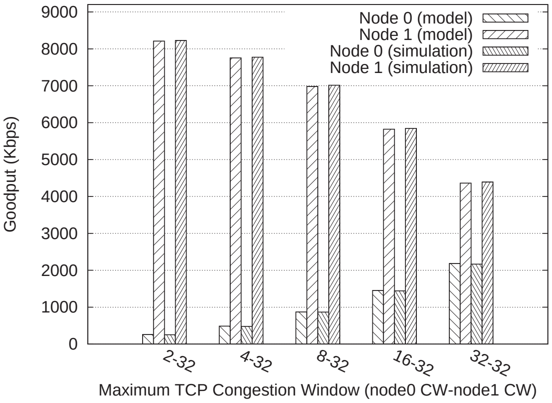


Figure 5: Numerical results obtained from the model vs. simulation results for a 2-hops parking lot topology

We next experiment with larger topologies, including a four-hop parking lot topology and a 2x2 grid topology. Our results are shown in Figures 6 and 7, averaged over the equilibrium period. In Figure 6 the closest node (*i.e.*, node 3) has a transmission step of one while other nodes have a transmission step of two. Simulation results show that node 3 achieves twice the throughput as other nodes. In our 2x2 grid topology (Figure 7), nodes 1 and 2 are one-hop away from the gateway while node 0 is two-hop away and relays its traffic via node 2. This network has two branches, one consisting of node 1 and another containing node 0 and 2. Our simulation results show that the cumulative throughput values of nodes in each branch are equal. Furthermore, node 0 achieves half the throughput of node 2 as predicted by our model.

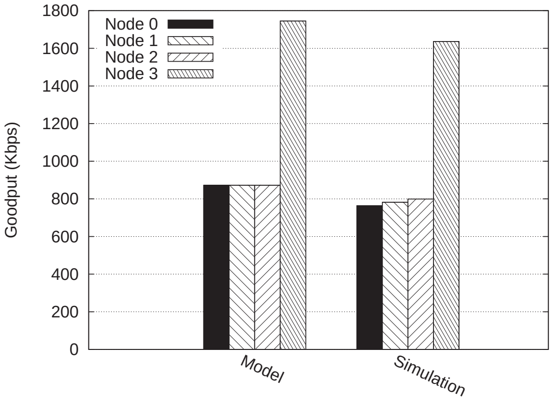


Figure 6: Numerical results obtained from the model vs. simulation results for a 4-hops parking lot topology

6.2. Performance evaluation of TMAC

We have implemented and evaluated the performance of our proposed TMAC protocol in ns-3. We use *Jain's Fairness Index* (JFI) [29] to quantify the fairness of our measured rate allocation, and present corresponding network utilization values for the measured allocation. We normalize the simulation results to the *optimally fair* flow rate distribution obtained with the collision domain network capacity model proposed by Jun and Sichitiu [30], using 'optimal' (*i.e.,* no collisions) achievable goodput results from Table 2.

6.2.1. Parking lot topologies

We evaluated TMAC with several variations of parking lot and grid topologies. The spacing between nodes is 200 m. Using the default ns-3 radio parameters, only adjacent nodes in the chain are within transmission range and nodes upto two-hops away are within interference range. Each node initiates an uplink TCP flow to the gateway.

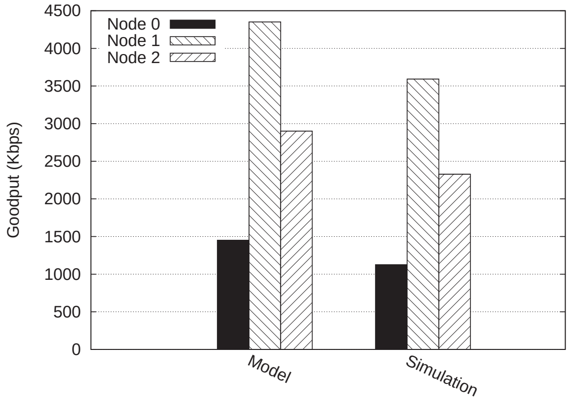


Figure 7: Numerical results obtained from the model vs. simulation results for a 2x2 grid topology.

TMAC fairness: First, we analyze flow rate fairness characteristics of TMAC. We first describe our results for a 5-hop chain. Nodes are numbered from 0 to 4, where node 0 is the farthest node from the gateway. Figure 8 shows the throughput obtained by TMAC compared to the reference optimal results discussed earlier in this section. TMAC registers a capacity drop of approximately 7% compared to these reference results.

We have extensively evaluated TMAC over a number of additional parking lot topologies, varying the size from 2-hops up to 6-hops. Our results are tabulated in Table 3. For network utilization, we list the values normalized to the reference optimal results. TMAC achieves a minimum JFI of 0.99 and an average network utilization of around 93%.

TMAC convergence rate: We use a 4-hop parking lot topology to characterize the convergence time for various TCP flows. This experiment aims to obtain the time required for a new flow to converge to its fair rate allocation. At the beginning of the simulation, all flows are inactive. At time 10 sec. the 1-hop

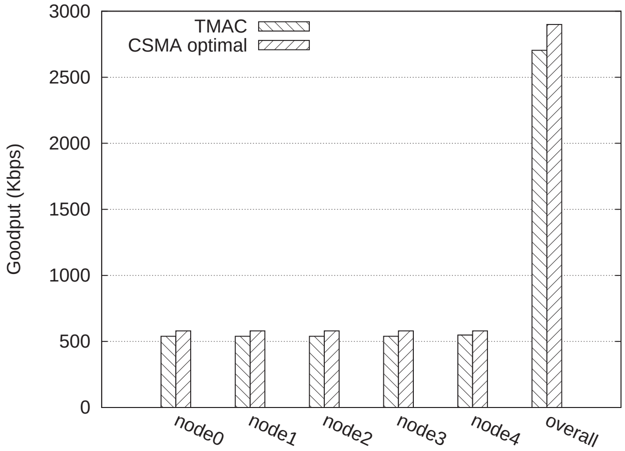


Figure 8: Per-node goodput for a 5-hop parking lot topology

Scenario	Norm. Network Utilization	JFI
2-hops	93.38%	0.999
3-hops	93.35%	0.999
4-hops	94.20%	0.999
5-hops	93.04%	0.999
6-hops	93.80%	0.998

Table 3: TMAC performance analysis in parking lot topologies with 12 Mb/s wireless links

node (node 3) initiates a flow destined to the gateway. Every 10 sec, each of the other nodes initiates a new flow destined to the gateway. Figure 9 shows the instantaneous throughput of each flow (averaged over a 1 sec. interval). We observe that all flows successfully converge to their fair share assignments within 1 sec. from initiation.

6.2.2. Grid topologies

We extend our experiments to grid topologies as shown in Figure 10. The vertical and horizontal spacing between nodes is 200 m; thus, nodes in a 4x4

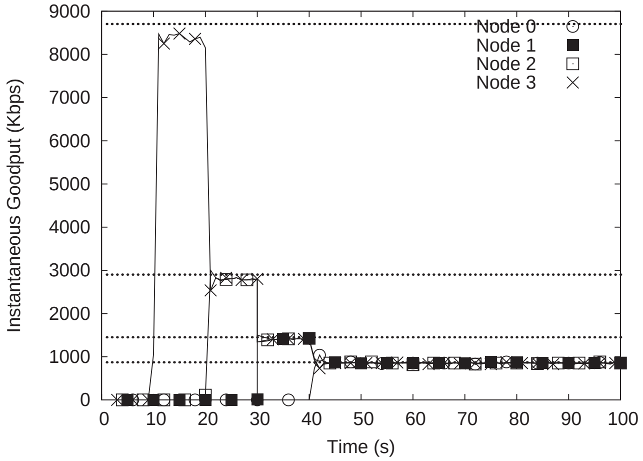


Figure 9: The instantaneous goodput of a 4-hop parking lot topology. New flows are initiated every 10 sec. starting with the 1-hop flow from (node 3) at time 10 sec. Dotted lines show the optimal throughput.

grid topology have up to four neighbors in the transmission range and up to 11 neighbors in the interference range. The number inside a node indicates its hop-count number along the shortest path to the gateway. All nodes in the network have an active TCP connection with the gateway.

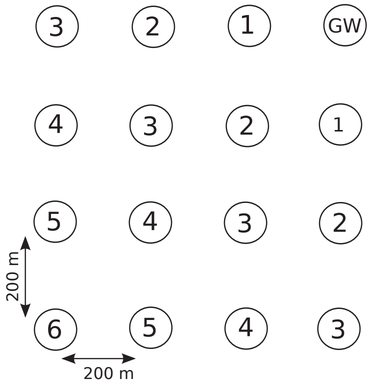


Figure 10: A 4x4 grid topology. Node numbers represent the number of hops along the shortest path to the gateway

We performed several simulations while varying the grid size from 2x2 to

Grid size	Norm. Network Utilization	JFI
2x2	92.42%	0.999
3x3	91.17%	0.992
4x4	90.18%	0.993

Table 4: TMAC results for grid topologies with 12 Mb/s links

4x4. Our results are shown in Table 4. TMAC achieves a network utilization higher than 90% while maintaining a JFI fairness of at least 0.99. Figure 11 shows detailed results for TMAC compared to CSMA and CSMA/CA for the 4x4 grid. Both CSMA and CSMA/CA have nodes experiencing unfairness and starvation; approximately 45% and 65% of the nodes starve with CSMA and CSMA/CA, respectively. TMAC (both with and without data bursts) improves fairness amongst nodes. Using data bursts further increases average flow rate by approximately 15–20%. Thus our conservative choice of burst size parameters maintains a balance between the throughput and fairness requirements of this network.

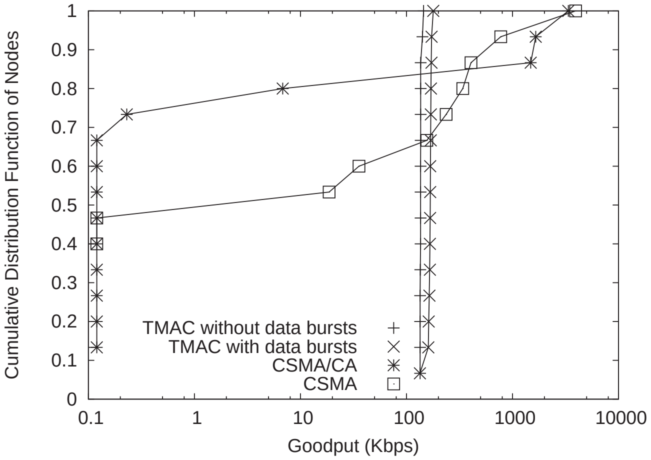


Figure 11: The CDF of goodput values in a 4x4 grid topology

The TMAC convergence time, which is the time where the instantaneous throughput (we use granularity of 1 sec.) reaches its fair share, for the presented grid topologies is shown in Figure 12. We observe that all nodes converge to their fair allocation within 7 sec. of the simulation run.

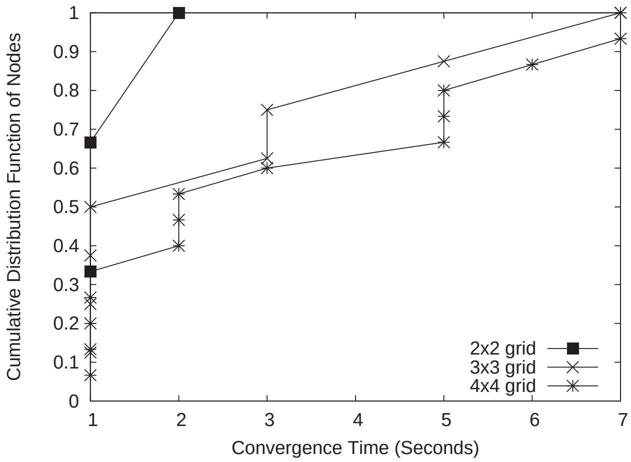


Figure 12: The CDF of convergence time in grid topologies

6.2.3. Random topologies

We have also verified the performance characteristics of TMAC across large random topologies. We used a 25-node random topology with all nodes transmitting TCP traffic towards a gateway. Our results are summarized in Figure 13, where we list relative flow rates, with the throughput of each node normalized to the maximum throughput obtained by any node. The JFI value for this allocation was measured to be 0.995.

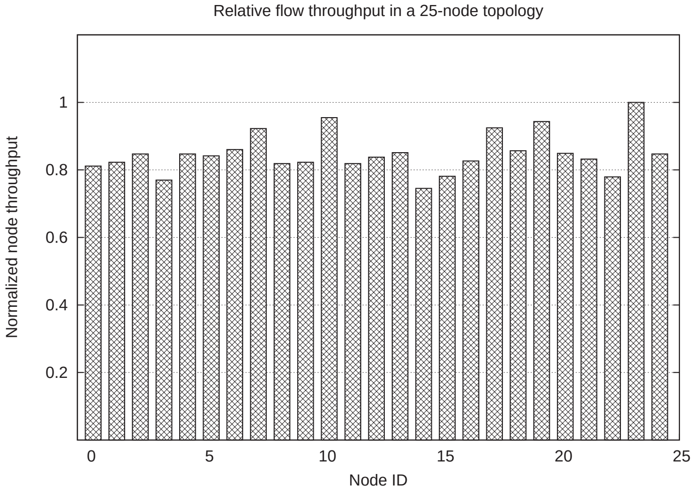


Figure 13: A 25-node random topology. Throughput values are normalized to the maximum throughput obtained by a given node.

7. Conclusions and Future Work

We present an analytical model to evaluate TCP throughput fairness over 802.11-based WMNs. Our model captures the interaction between multiple TCP streams and 802.11 MAC protocol by focusing on the relative flow utilization of the cumulative network queue. The multi-hop effect on TCP performance is modeled by embedding the number of transmission steps affecting the modeled queues. We then propose TMAC, a distributed MAC protocol to overcome the unfairness characteristics of 802.11 in multi-hop networks. TMAC effectively addresses the various causes of unfairness as observed in our model. TMAC enforces fairness through prioritizing the transmission of packets based on their age. We implemented TMAC message exchange using RTS/CTS control frames in the 802.11 radios. We proposed data bursts to overcome the RTS/CTS exchange overhead in WMNs. TMAC successfully achieves resource allocation fairness and maintains over 90% of maximum link capacity in parking lot and grid topologies. We performed a simulation evaluation to validate the model and found that it can accurately predict flow throughput. Further experiments were performed on TMAC to confirm its fairness. TMAC is found to achieve resource allocation fairness in parking lot and grid topologies while maintaining over 90% of maximum link capacity.

We are currently investigating extending the use of this model to propose WMN-aware protocols in other layers of the protocol stack. Transport-layer protocols are a potential candidate for such work, *e.g.,* TCP's congestion control mechanism may be adapted to support the fairness requirements. The proposed model and the derived throughput equations can be formulated as an optimization problem to achieve fairness by limiting the congestion windows. This will enable us to achieve fairness by applying modifications to the gateway only. However, it is necessary to obtain real-time information and perform the optimization on the fly.

We are further investigating optimizing the performance of TMAC by inverting the use of grant messages. For example, in response to a request message from a sender S, a neighbor N instead responds with a deny message if it has a packet with an older timestamp pending for transmission. This can also suppress further transmission of messages from neighbors which have pending transmissions with a lower timestamp than S but higher than N. However, the challenge is accounting for lost messages; for example, the loss of deny-messages can inherently be interpreted as a grant. This is problematic for wireless networks with high loss rates and needs to be further investigated.

References

[1] I. Akyildiz, X. Wang, W. Wang, Wireless mesh networks: a survey, Computer Networks and ISDN Systems 47 (4) (2005) 445–487.

[2] V. Gambiroza, B. Sadeghi, E. Knightly, End-to-end performance and fairness in multihop wireless backhaul networks, in: Proceedings of the ACM MobiCom '04, 2004, pp. 287–301.

[3] K. Jamshaid, P. A. Ward, Centralized feedback-driven rate allocation mechanism for CSMA/CA-based wireless mesh networks, in: Proceedings of the ACM MSWiM '08, 2008, pp. 387–394.

[4] L. Jiang, S. C. Liew, Proportional fairness in wireless LANs and ad hoc networks, in: Proceedings of the IEEE WCNC '05, 2005.

[5] A. Acharya, A. Misra, S. Bansal, Time, clocks, and the ordering of events in a distributed system, Communications of the ACM 21 (7) (1978) 558–568.

[6] N. Abramson, The ALOHA system: Another alternative for communications, in: Proceedings of the AFIPS FJCC '70, 1970, pp. 281–285.

[7] G. Bianchi, Performance analysis of the IEEE 802.11 Distributed Coordination Function, IEEE Journal on Selected Areas in Communication 18 (3) (2000) 535–547.

[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] R. Boorstyn, A. Kershenbaum, B. Maglaris, V. Sahin, Throughput analysis in multihop CSMA packet radio networks, IEEE Transactions on Communications 35 (3) (1987) 267–274.

[10] S. Pilosof, R. Ramjee, D. Raz, Y. Shavitt, P. Sinha, Understanding tcp fairness over wireless lan, in: Proceedings of the IEEE Infocom '03, 2003, pp. 863–872.

[11] A. Kherani, R. Shorey, Throughput analysis of TCP in multi-hop wireless networks with IEEE 802.11 MAC, in: Proceedings of the IEEE WCNC '04, 2004, pp. 237–242.

[12] F. Nawab, K. Jamshaid, B. Shihada, P.-H. Ho, TMAC: Timestamp-ordered MAC for CSMA/CA Wireless Mesh Networks, in: Proceedings of the IEEE ICCCN '11, 2011, pp. 1–6.

[13] F. Nawab, K. Jamshaid, B. Shihada, P.-H. Ho, MAC-Layer Protocol for TCP Fairness in Wireless Mesh Networks, in: Proceedings of the IEEE ICCC '12, 2012, pp. 507–512.

[14] X. Huang, B. Bensaou, On max-min fairness and scheduling in wireless adhoc networks: analytical framework and implementation, in: Proceedings of the ACM MobiHoc '01, 2001, pp. 221–231.

[15] T. Nandagopal, T. Kim, X. Gao, V. Bharghavan, Achieving MAC layer fairness in wireless packet networks, in: Proceedings of the ACM MobiCom '00, 2000, pp. 87–98.
URL citeseer.ist.psu.edu/nandagopal00achieving.html

[16] N. H. Vaidya, A. Dugar, S. Gupta, P. Bahl, Distributed fair scheduling in a wireless LAN, IEEE Transactions on Mobile Computing 4 (6) (2005) 616–629.

[17] I. Aad, C. Casteulluccia, Differentiation mechanisms for IEEE 802.11, in: Proceedings of the IEEE INFOCOM '01, 2001, pp. 209–218.

[18] C. Cicconetti, I. F. Akyildiz, L. Lenzini, FEBA: a bandwidth allocation algorithm for service differentiation in IEEE 802.16 mesh networks, IEEE/ACM Transactions on Networking 17 (3) (2009) 884–897.

[19] S. Golestani, A self-clocked fair queueing scheme for broadband application, in: Proceedings of the IEEE Infocom '94, 1994, pp. 636–646.

[20] J. L. Sobrinho, A. S. Krishnakumar, Real-time traffic over the IEEE 802.11 medium access control layer, Bell Labs Technical Journal 1 (1996) 172–187.

[21] H. Luo, S. Lu, V. Bharghavan, A new model for packet scheduling in multihop wireless networks, in: Proceedings of the ACM MobiCom '00, 2000, pp. 76–86.

[22] L. Tassiulas, S. Sarkar, Maxmin fair scheduling in wireless ad hoc networks, IEEE Journal on Selected Areas in Communications 23 (1) (2005) 163–173.

[23] Z. Fang, B. Bensaou, Fair bandwidth sharing algorithms based on game theory frameworks for wireless ad-hoc networks, in: Proceedings of the IEEE Infocom '04, 2004, pp. 1284–1295.

[24] J. Lee, W. Liao, M. C. Chen, An incentive-based fairness mechanism for multi-hop wireless backhaul networks with selfish nodes, IEEE Transactions on Wireless Communications 7 (2) (2008) 697–704.

[25] S. Ray, J. B. Carruthers, D. Starbinski, Evaluation of the masked node problem in ad hoc wireless LANs, IEEE Transactions on Mobile Computing 4 (5) (2005) 430–442.

[26] J. I. Choi, M. Jain, M. A. Kazandjieva, P. Levis, Granting silence to avoid wireless collisions, in: Proceedings of the IEEE ICNP '10, 2010.

[27] K. Jamshaid, P. A. Ward, Gateway-assisted max-min rate allocation for wireless mesh networks, in: Proceedings of the ACM MSWiM '09, 2009, pp. 38–45.

[28] S. Jain, S. Das, MAC layer multicast in wireless multi-hop networks, in: Proceedings of the Comsware '06, 2006.

[29] R. K. Jain, D.-M. Chiu, W. R. Hawe, A quantitative measure of fairness and discrimination for resource allocation in shared computer system, Tech. rep., DEC Research Report TR-301 (September 1984).

[30] J. Jun, M. L. Sichitiu, The nominal capacity of wireless mesh networks, IEEE Wireless Communications (2003) 8–14.

[31] F. P. Kelly, Reversibility and stochastic networks, Cambridge University Press, 2011.

Appendix A. Reversibility proof

Here we prove the reversibility of the Markov chain model presented in Section 3.2. We recall that a state consists of n items, where each item is the number of data packets, P_i, of a single flow, i, in the system. The maximum number of packets for a flow i is the congestion window, W_i, that can be different for different flows. There are two types of transitions in the model. The first corresponds to data transmissions, represented by a transition from a state S to state $S - 1_i$, where $S + (k)_i$ represents the state that is equivalent to S except that the number of packets of flow i have increased by the number k. The state transition probability from S to $S - 1_i$, $Pr_{S \to S-1_i}$, is

$$\frac{P_i}{k_i.l_\eta.(\sum_{j=1}^{n} P_j)} \tag{A.1}$$

where k_i is a constant that could be different across flows and $l_\eta = 1$ if either $\sum_{i=1}^{n} P_i = 0$ or $\sum_{i=1}^{n} P_i = \sum_{i=1}^{n} W_i$, or $l_\eta = 2$ otherwise. The transition probability of an ACK transmission, $Pr_{S \to S+1_i}$, is calculated by

$$\frac{W_i - P_i}{k_i.l_\eta.(\sum_{j=1}^{n} W_j - P_j)} \tag{A.2}$$

To show the model's reversibility, it suffices to demonstrate that the model satisfies the Kolmogorov's criterion [31] defined below:

Definition 1. *(**Kolmogorov's criterion**) A stationary Markov chain is reversible if and only if*

$$Pr_{j_1 \to j_2} Pr_{j_2 \to j_3} \ldots Pr_{j_n \to j_1} = Pr_{j_1 \to j_n} \ldots Pr_{j_3 \to j_2} Pr_{j_2 \to j_1}$$

for any finite sequence of states $j_1, j_2, \ldots, j_n \in \bar{S}$, where $Pr_{x \to y}$ is the transition probability from state x to state y and $\bar{S}$ is the set of possible states sequences.

Consider any sequence of states, α. Let β be the sequence of states that traverse the same sequence of states as in α but in the reverse order. Thus, to show that the Kolmogorov's criterion hold is to show that for any α, the product of transition probabilities through the sequence of states in α and β yield the same value. By definition, any transition $x \to y$ in α results in a transition $y \to x$ in β, and we call it the *feedback* of $x \to y$. Denote the sum of the number of packets of all flows, $\sum_{i=1}^{n} P_i$, of a state x as the *data-weight* of any $x \to y$. Also, let the sum of the number of ACKs of all flows, $\sum_{i=1}^{n} W_i - P_i$, of a state x be the *ack-weight* of any $x \to y$.

In what follows we will use an expansion of each transition probability, $Pr_{x \to y}$, to three terms. The first term, denoted $Pr_{x \to y}^{upper}$, is the numerator as shown in Equation A.1 or A.2. The second term, denoted $Pr_{x \to y}^{kl}$, is equivalent to the k_i and l_η terms in the equations. The final term is the sum of either the number of packets or ACKs and is denoted as $Pr_{x \to y}^{sum}$. Thus, $Pr_{x \to y} = Pr_{x \to y}^{upper} Pr_{x \to y}^{kl} Pr_{x \to y}^{sum}$

We now show that for every term in the expansion of a transition probability $Pr_{x \to y}$ in α, there exist expansion terms in the transition probabilities of β that is equal to them. Let us start with $Pr_{x \to y}^{kl}$

Lemma 1. *Every $Pr^{kl}_{x\rightarrow y}$ in α is equal to unique terms in β*

Proof. We know that for any $x\rightarrow y$ in α there exist $y\rightarrow x$ in β. $Pr^{kl}_{x\rightarrow y}$ have two variables, k_i and l_η. k_i is identical for transitions of the same flow. Both $x\rightarrow y$ and $y\rightarrow x$ are transitions of the same flow, thus the k_i terms are equal for both $Pr^{kl}_{x\rightarrow y}$ and $Pr^{kl}_{y\rightarrow x}$. The term l_η is equal to value 2 for all transition probabilities except for $2n$ transition probabilities. Those are the transition probabilities of transitions out of states $\emptyset$ and U, where $\emptyset$ is the state where all flows have 0 data packets and U is the state where all states satisfy $P_i = W_i$. There are three possibilities:

1. Both $x\rightarrow y$ and $y\rightarrow x$ are not transitions out of $\emptyset$ and U. In this case, the l_η terms in $Pr^{kl}_{x\rightarrow y}$ and $Pr^{kl}_{y\rightarrow x}$ are equal.
2. Both $x\rightarrow y$ and $y\rightarrow x$ are transitions out of $\emptyset$ and U. This case is possible if there is a flow with a window size equal to 1 and one transition is out of $\emptyset$ and the other is out of U. In this case, the l_η terms in $Pr^{kl}_{x\rightarrow y}$ and $Pr^{kl}_{y\rightarrow x}$ are equal.
3. $x\rightarrow y$ is out of $\emptyset$ or U while $y\rightarrow x$ is not. The probability transition $x\rightarrow y$ makes the value of l_η equal to 1. This transition is in α. This means that there exists a transition into $\emptyset$ or U, call it tr, in α. Since tr is in α, thus a transition tr' exists in β, where it is a transition out of $\emptyset$ or U. The l_η terms in $Pr^{kl}_{x\rightarrow y}$ and $Pr^{kl}_{tr'}$ are equal. Also, the l_η terms in $Pr^{kl}_{y\rightarrow x}$ and Pr^{kl}_{tr} are equal.

Thus, for all cases, a transition probability term $Pr^{kl}_{x\rightarrow y}$ in α is equal to a unique term in β. □

Next we show the lemma regarding $Pr^{upper}_{x\rightarrow y}$ of ACK transmissions.

Lemma 2. *Every* $Pr^{upper}_{x\rightarrow y}$ *in* α *is equal to unique terms in* β *if* $x\rightarrow y$ *corresponds to an ACK transmission.*

Proof. Let y be state $x+1_j$, thus $x\rightarrow y$ is a transition corresponding to an ACK transmission of flow j. There is necessarily a transition probability, $Pr_{w\rightarrow z}$ (call $w\rightarrow z$ the *return* of $x\rightarrow y$), where z equals $w-1_j$. This is because the sequence starts and ends in the same state. Thus, there is an equal number of data and ACK transitions for each flow. Either: (1) $x=z$ and $y=w$, that means going back through the same state in the sequence. This is trivial and both cancel each other. (2) $x\neq z$ and $y\neq w$. In this case, consider the feedback of both transitions in β. They are $y\rightarrow x$ and $z\rightarrow w$. Note that the term $Pr^{upper}_{x\rightarrow y}$ depends on the number of packets and window of the corresponding flow. The transition that is the feedback of the return of $x\rightarrow y$ is out of a state with the same number of packets and they both correspond to the same flow, and hence have the same window size. Since they both correspond to ACK transmissions, the upper term is the number of packets for both of them, hence $Pr^{upper}_{x\rightarrow y}=Pr^{upper}_{z\rightarrow w}$.

□

Likewise we show the lemma of data transmissions.

Lemma 3. *Every* $Pr^{upper}_{x\rightarrow y}$ *in* α *is equal to unique terms in* β *if* $x\rightarrow y$ *corresponds to a data transmission.*

Proof. Following the same discussion in Lemma 2, consider the state transitions $y\rightarrow x$ and $w\rightarrow z$. These correspond to data transmission. Also, they are the feedback of the return of each other. Since the upper term depends on the number of packets and the number of packets of both of them are equal, then $Pr^{upper}_{y\to x} = Pr^{upper}_{w\to z}$.

□

Now we turn to the case of $Pr^{sum}_{x\to y}$

Lemma 4. *Every $Pr^{sum}_{x\to y}$ in α is equal to unique terms in β.*

Proof. As shown in Equations A.1 and A.2, $Pr^{sum}_{x\to y}$ depends on either the $data-weight$ or $ack - weight$ for data or ACK transmissions, respectively. In each transition either $data-weight$ or $ack-weight$ increases and the other decreases both by the value 1. Note that the sequence α starts and ends in the same state. Thus, the number of increases and decreases are equal. Moreover, for every transition, tr, out of a state with a specific $data-weight$ and $ack-weight$ value, there is necessarily a transition, tr', into a state with the same $data - weight$ and $ack - weight$ value, where both states are in α. Note that the feedback of tr', $feedback(tr')$, is a transition of the same type as tr and have the same value of both $data-weight$ and $ack-weight$. Thus, Pr^{sum}_{tr} is equal to $Pr^{sum}_{feedback(tr')}$.

□

Now, our final result is the theorem:

Theorem 1. *The Markov chain model of Section 3.2 is reversible.*

Proof. By the previous Lemmas 1, 2, 3, and 4, each term in any sequence α has a unique term in β that is equivalent to it. Thus, the product of transition probabilities of a sequence α and the product of transition probabilities of its corresponding sequence in the other direction, β, are equivalent. By Definition 1, the Markov chain model of Section 3.2 is reversible. $\square$

*Author Biography

Author Bios. will be provided with final accepted manuscript.

***Author Photo**

Author photos will be provided with final accepted manuscript.