Modeling the Short-term Unfairness of IEEE 802.11 in Presence of Hidden Terminals

Zhifei Li, Sukumar Nandi, and Anil K. Gupta
School of Computer Engineering, Nanyang Technological University, Singapore-639798
Email: {pg03802331, assukumar, asgupta}@ntu.edu.sg

Abstract:

IEEE 802.11 exhibits both short-term and long-term unfairness [15]. The short-term fairness automatically gives rise to long-term fairness, but not vice versa [11]. When we thoroughly investigated a simple scenario with hidden terminals, we found it to be unfair on the short-term basis, though it provides fair access on a long-term basis. It implies that the protocol cannot be used to provide fair access for delay sensitive traffic even in a simple scenario. In this paper, we analyze the *short-term* behavior using the embedded-Markov chain method to answer the following two questions: *(i)* once a node gets control of the medium, what is the average number of packets this node can transmit *consecutively* without experiencing any collision, *(ii)* once a node loses its control of the medium, what is the average time the node has to wait before it gets control of the medium again. The first question reflects on how long a node can *capture* the medium, whereas the second question reflects on how long a node may be *starved*. The analytical model is validated by the simulation results. Our work is distinct from most of the work published in the literature in two aspects: we focus on the short-term behavior rather than the long-term, and the analytical method is adopted for the study.

Keywords: IEEE 802.11, Hidden-terminal, Short-term Fairness, Embedded Markov Chain,

1 Introduction

Recently, mobile computing and wireless networks have attracted considerable research interest. IEEE 802.11 [9] is the de facto industry standard for Wireless LANs, which defines two MAC protocols: Point Coordination Function (PCF) and Distributed Coordination Function (DCF). However, only the DCF is popular. DCF describes two techniques for packet transmission, a two-way handshake and a four-way handshake. The two-way handshake severely suffers from the well-known hidden-terminal problem. On the contrary, the four-way handshake greatly reduces the hidden-terminal problem. In this paper, we mainly focus on the four-way handshake case.

As DCF operates in a *distributed* manner, fairness in accessing the medium is one of the most challenging issues [15]. Based on the *length* of the time over which we observe the system, the fairness can be defined on a short-term basis and a long-term basis. The short-term fairness implies long-term fairness, but not vice versa [11]. In particular, under certain scenarios, though the bandwidth allocation is fair in a long-term, it is very unfair if we view the system from a short-term viewpoint. The short-term fairness is very important for adaptive traffic (e.g., TCP kind of traffic) and for delay- or jitter-sensitive traffic [11]. In this paper, we model and analyze the short-term behavior of IEEE 802.11 in the presence of hidden-terminal.

In the literature, there have been many analytical models (e.g., [2-4, 18]) developed to evaluate the *long-term* performance (e.g., capacity and average packet delay) in a *single-hop* IEEE 802.11 network. However, none of these models has thrown any light on the short-term unfairness in a hidden-terminal scenario. In general, there are two challenges in modeling the *short-term unfairness*. *Firstly*, to model the *short-term* behavior, the model should capture the protocol in a very detailed manner, as otherwise the required details are lost in the model. Therefore, those models developed to study the *long-term* behavior (e.g., capacity and average packet delay) may not be appropriate to study the short-term performance as they may neglect the required details for simplicity. *Secondly*, while analyzing the throughput or packet delay, the models (e.g., [2-4, 18]) assume that all the nodes in the system have the same average behavior, and thus they only need to analyze the performance at a *typical* or *tagged* node. In contrast, to predict the *unfairness* precisely, the model should take into account the *difference* in the behavior of the various contending nodes. In other words, the fairness reflects the *relative* behavior, and can be predicted only by comparing the behavior at different nodes. Therefore, in the representation of the system state, we need to include one or more *separate* state variables for each of the contending nodes. Due to this, the modeling of the short-term unfairness of IEEE 802.11 becomes extremely complex. Moreover, the hidden-terminal problem itself incurs additional complexity. For example, in a single-hop network, a collision occurs mainly in the situation when two or more contending nodes have a same value of the back-off timers. On the contrary, in a hidden-terminal scenario, a collision will occur as long as the difference in the back-off values is small as will be shown in this paper. In the design of our model, we bear in mind the above challenges.

Jain's index [5] can be used to reflect fairness over different time scale. Though the index is useful in comparing fairness of two *different* protocols, the absolute value of the index for a *given* protocol does not express the fairness of the protocol very clearly. Therefore, we measure the short-term fairness in an alternate way by evaluating the following two metrics: *(i)* once a node gets control of the medium, what is the average number of packets this node can transmit *consecutively* without experiencing any collision, *(ii)* once a node loses its control of the medium, what is the average time the node has to wait before it gets control to the medium again. The first metric reflects on how long a node can *capture* the medium, whereas the second metric reflects on how long a node may be *starved*.

In this paper, using an embedded Markov chain model, the two metrics are measured based on the concepts of "*expected state holding time*" and "*expected first passage time*". The analytical model is validated by the simulation results. The results show that IEEE 802.11 exhibits substantial short-term unfairness in the presence of hidden terminals.

The remainder of the paper is organized as follows. In Section 2, through a simple simulation, we show that IEEE 802.11 exhibits considerable short-term unfairness in the presence of hidden-terminals. In Section 3, the Markov model is described. Sections 4 and 5 deal with the computation of the transition probabilities and the expected state holding time, respectively. Section 6 presents the analytical and simulation results. The related work is discussed in Section 7 and the paper is concluded in Section 8.

2 Short-Term Unfairness in IEEE 802.11

In order to show the short-term unfairness in the presence of hidden-terminal, we simulate a simple scenario depicted in Figure 1, which is commonly used to illustrate the hidden-terminal problem. In the figure, there are three nodes, A, B and C, with two single-hop flows: flow from A to B, and flow from C to B. Since nodes A and C cannot hear from each other, they may simultaneously try to communicate with a common node, i.e., node B, resulting in a collision. In such a situation, nodes A and C are referred as the *hidden terminals* of each other. The reason, that nodes A and C cannot hear from each other, may be that A and C are out of the range of each other. It may also be that there are some physical obstacles such as a wall between A and C.

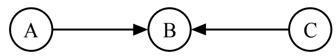

Figure 1: Hidden-terminal Scenario

For the study, the simulation environment is as follows. NS-2 with CMU wireless extensions [6] and static routing is used. For each single-hop flow, a Constant Bit Rate (CBR) traffic generates 200 packets per second. Each packet is 1000-bytes long, resulting in a traffic source rate of 1.6 Mbps. The raw bandwidth is set with 2 Mbps, leading to a *maximum* throughput of around 1.4 Mbps due to the overhead of IEEE 802.11. The source rate is made greater than the medium capacity since unfairness occurs only when the system is overloaded.

The simulation results show that each flow gets an average throughput of about 0.7 Mbps (more specifically, 0.68 Mbps), which means that the two flows share the medium fairly on a *long-term* basis. However, if we compute the values of the two *short-term* metrics defined in Section 1, the protocol exhibits substantial unfairness. For metric *(i)*, we find that, on an average, once a node gets control to the medium, it can transmit about 6.4 packets *consecutively* without experiencing any collision in between. For the metric *(ii)*, once a node (say node C) loses control of the medium, it has to wait for the other node (i.e., A) to transmit about 27 packets before it gets control of the medium again. However, this does not mean that node A can transmit 27 packets consecutively without experiencing any collision. We illustrate this using the following example. Consider that after node C loses control of the medium, node A gets control of the medium and it transmits one or more packets consecutively. Then, one or more collisions occur. After the collision(s), node A *again* gets control and transmits one or more packets consecutively. This kind of process may repeat several times before node C gets control of the medium, which explains why the values of the two metrics differ so much. In addition to the average values, we also observed the corresponding *maximum* values, which are equal to 35 and 160, respectively, for the two metrics. This shows how much short-term unfairness is ingrained in the current IEEE 802.11, which is unacceptable for jitter-sensitive traffic. In the next subsection, we describe some basic techniques adopted in IEEE 802.11, based on which, we intuitively explain in Section 2.2 how the hidden-terminal problem causes short-term unfairness.

2.1 Basic Techniques in IEEE 802.11

To combat with the hidden-terminal problem, IEEE 802.11 defines a four-way handshake. In the handshake, before the transmission of a Data frame, the sender first sends out a Request To Send (RTS) frame. In response to this request, the receiver sends back a Clear To Send (CTS) frame if the medium is determined to be idle. Then, the sender sends out the Data frame and the receiver responds with an ACK frame if it receives the Data frame without error. For the convenience, we call the exchange of RTS/CTS/Data/ACK frames as a *frame exchange sequence* (FES). Moreover, FES (A, B) represents a FES between nodes A and B, *initiated* by node A, implying that node A sends one *packet* successfully to node B. For simplicity, we call node A as the *transmitting* node and B as the *receiving* node, while node C is called the *waiting* node. Note that the word *'packet'* implies the protocol data unit (PDU) of a higher layer whereas *'frame'* is the MAC layer PDU.

In the four-way handshake, once the RTS/CTS handshake has been completed successfully, the hidden-terminal problem does not arise any more. For example, in Figure 1, once node B sends back a CTS to node A, node C also hears this CTS and thus defers its transmission, avoiding collision. The four-way handshake solves the hidden-terminal problem *largely* by introducing the RTS/CTS handshake before the real Data frame is transmitted. However, it cannot eliminate the problem completely as RTS/CTS cannot always be transmitted successfully. In a while, we will derive the condition under which the RTS/CTS can be successfully completed.

The IEEE 802.11 adopts the well-known Binary Exponential Back-off (BEB) algorithm as its Contention Resolution (CR) mechanism, which is described as follows. Whenever an attempt to transmit a packet fails, a retransmission is scheduled, unless a retry limit (say n) is reached. Every node maintains a Contention Window (CW) and a back-off timer. Before every transmission (including the retransmission), the node first defers by a back-off timer, which is generated according to Equation (1), unless the back-off timer already contains a non-zero value, in which case it is unnecessary to generate a new random back-off timer.

$$Backoff\ Time = Random() \times SlotTime \qquad (1)$$

In the Equation (1), the *SlotTime* is specified by the physical layer, and the *random* value is uniformly distributed over the range [0, CW]. For the first transmission attempt of a packet, the CW will be set to CW_{min}. Whenever a retransmission is initiated, the CW is *doubled*. The CW increases to its maximum value after m retries. After that, the CW remains at the maximum value until the retry limit (i.e., n) is reached. Once the retry limit is reached, the CW will be reset to CW_{min}. The CW is also reset to CW_{min} whenever a transmission attempt is successful. For the convenience, we call a retransmission attempt as a *stage*. If we denote the *stage number* by index i, which is in the range [0, n-1], the CW can be expressed as:

$$CW = \begin{cases} (CW_{\min}+1)\times 2^{i}-1; & 0 \le i \le m \\ (CW_{\min}+1)\times 2^{m}-1; & m < i < n \end{cases} \qquad (2)$$

When a FES is in progress, the *waiting* node *freezes* its back-off timer. After the FES is successfully completed, all the nodes first defer for a DCF Inter-Frame Space (DIFS) period. Then, the *transmitting* node generates a new random value from its CW and backs-off before it initiates another FES. On the other hand, the *waiting* node simply resumes counting down from its *frozen* back-off timer. Clearly, due to the *freezing mechanism*, the *transmitting* node may transmit several packets *consecutively* before the *waiting* node's back-off timer is reduced to zero, leading to short-term unfairness.

Contrary to a successful transmission, when a collision occurs, all the colliding nodes will generate a new random value from their corresponding CWs. Since IEEE 802.11 uses a timeout mechanism to detect the collision, and since the starting time of the two collided frames may not be the same, the time at which the two nodes detect the collision may be different.

Now let us derive the *condition* under which the RTS/CTS can be successful when two hidden nodes (A and C) contend for the medium (Figure 1). The condition is as follows: after a collision or a successful FES, the difference between the back-off timers at the two hidden nodes should be large enough for node B to send back a CTS to node A (C) before that node C (A) starts sending its RTS. The *minimum* time difference required is equal to the transmission time of RTS plus a Short Inter-Frame Space (SIFS). This can be expressed as:

$$|Z| > Len = TxTime(RTS) + SIFS \qquad (3)$$

where Z is the difference between the back-off timers. *Len* is equal to about 19 slots when the slot time is 20 μs for DSSS [9]. Note that the *TxTime*(*RTS*) corresponds to the time needed for the complete transmission of the MAC layer frame, i.e., it includes the physical layer overhead (192 μs) and the MAC layer contents (20×8 = 160 μs). It is easy to see that the condition is difficult to satisfy when the CWs at the contending nodes are small (e.g., *31*). Note that in the above analysis, we have omitted the propagation delay, which is negligible. We also assume that the two nodes begin to back-off at the same time, i.e., after a collision, the time difference between the two nodes detecting the collision is neglected.

2.2 Explanation for the Short-term Unfairness

Now let us explain how the hidden-terminal problem causes short-term unfairness. Consider the situation that the CWs at nodes A and C are very small (e.g., 31). As discussed above, under such situation the condition illustrated in equation (3) is difficult to satisfy, i.e., the transmission of RTSs of nodes A and C may overlap partially, and as a result collide. The collision may occur several times until the CWs are large enough to allow either node to get control of the medium. In particular, one of the two nodes (let us say, node A) may select a small back-off time from its CW, while the other node (i.e., C) selects a large value. The difference between these two values may be large enough to satisfy the condition illustrated in equation (3) and, the FES (A, B) can be successfully completed. Once the FES (A, B) is completed, node A resets its CW and backs-off before initiating another FES. However, the *remaining* back-off timer at node C may be large compared to the back-off timer at node A, which is drawn from the range [0, CW_{min}]. In that case, nodes A and B may exchange several more FESs before node C's back-off timer reduces to zero.

Whenever the back-off timer at node C reduces to zero, node C contends for the medium. However, as the CW at node A is equal to CW_{min}, the contention is most likely to result in a collision. After the collision, node A doubles its CW from CW_{min} whereas node C doubles its CW from a larger value (at least 63). Therefore, the CW at node C is greater than that at A, and node A is more likely to get control of the medium *again*. This is obviously unfair for node C since A has already transmitted several packets while C is starved during this period. Moreover, this process (i.e., several packet transmissions by node A, followed by collisions, and then packet transmissions by node A again) may repeat several times, leading to starvation at node C for a long period (compared to the time needed for a FES).

However, several mechanisms incorporated in the IEEE 802.11 prevent node C from starving *completely*, such as: *(i)* after every FES, node A will back-off before initiating another FES, which gives node C a chance to contend for the medium with node A; *(ii)* the CW at node C will be reset to CW_{min} after the retry limit n is reached. These mechanisms ensure *long-term* fairness between the two flows.

Though the above analysis intuitively explains how the hidden-terminal problem causes short-term unfairness, we still need to evaluate the unfairness *quantitatively*, which is carried out in the following sections.

3 Analytical Modeling

In this section, we model the hidden-terminal scenario indicated in Figure 1 using an *embedded* Markov chain. In the discussion, the reader must differentiate between the words *stage* and *state*, which are very often used.

3.1 Markov Chain Model

At any point of time, the *medium* is in one of the following five states: T_A, T_B, T_C, *Col* and *Idle*, where T_A means that node A is getting the control of the medium and transmitting its packet, i.e., FES(A,B) is in progress. Similarly, T_B and T_C correspond to nodes B and C, respectively. However, in our considered scenario, T_B does not arise, as node B does not have any data packets to send. State *Col* means that there is a collision on the medium, while state *Idle* means that there is no transmissions or collisions over the medium. As our objective is to analyze the short-term fairness rather than the capacity utilization of the medium, we do not need to consider the *Idle* state. As a result, only three states of the medium are considered: T_A, T_C, and *Col*. When the medium is either in T_A or in T_C, we simply say, for the convenience of the presentation, that the medium is in a *T* state.

As mentioned before, to reveal the unfairness in IEEE 802.11, we should model the system state in a way that there is at least one separate parameter to represent the state of every contending station (i.e., nodes A and C in the hidden-terminal case). Since the transition probabilities among these three states (i.e., T_A, T_C, and *Col*) depend on the values of CWs at nodes A and C, which, in turn, are determined by the corresponding *stages*, the *system* can be modeled using three random variables: state of the medium, *stage* at node A, and *stage* at node C. Therefore, the system states are (T_A, k, l), (T_C, k, l), and (Col, k, l), where k and l denote the stages at nodes A and C, respectively. Obviously, $k, l = 0, \ldots, n-1$. When the medium is in state T_A, it is easy to see that the stage at the *transmitting* node (i.e., node A) must be zero, that is, only $(T_A, 0, l)$ system states are possible. Similarly, when the medium is in state T_C, the possible system states are $(T_C, k, 0)$. Therefore, if the retry limit is n, there are n^2 number of *Col* states, and n number of states corresponding to *each* of T_A and T_C, and thus the number of all possible system states is:

$$N_{state} = n^2 + 2n \qquad (4)$$

From a *Col* state, whenever a transition occurs, the system can enter anyone of the three kinds of states as shown in Figure 2. If the next state is a *Col* state, both of the stages, k and l, are incremented by one, except that the stage is reset to zero whenever the retry limit n is reached. On the other hand, if the next state is a *T* state, the stage at the *transmitting* node is reset while the other stage remains unchanged.

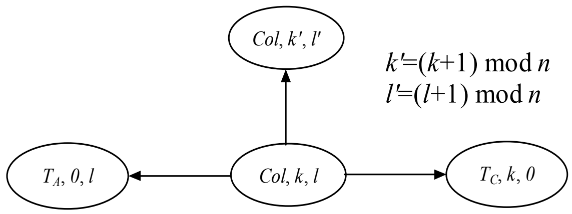

Figure 2: State Transitions from a Col State

In state T_A, whenever node A transmits another *packet*, it is natural to view this event as a self-transition. However, if we model the system in such a way, the transition probabilities depend upon the *remaining* back-off timer at the *waiting* node (i.e., node C), which in turn, depends upon how many times this timer has been frozen and decremented, i.e., how many self-transitions have occurred in the T_A state. This requires memorizing the history to obtain the current transition probabilities, which will violate the memoryless requirement of the Markov chain. Therefore, we do not treat this event as a state transition. Rather, *only* when the system enters a *Col* state or a T_C state, a state transition occurs in state T_A. Therefore, whenever the system enters a T_A state, the time that the system will *remain* in that state depends on the number of packets that node A can transmit consecutively before that node C controls the medium or that a collision occurs. Similar explanation will apply for the state T_C. The state transition at a T state is illustrated in Figure 3. It is easy to see that the chain obtained in such a way is an embedded Markov chain, which is defined at the instances where a transition occurs whenever the control of the medium is passed from node A to C or vice-versa, or a collision takes place after successful transmission of packet(s) from any of these two nodes.

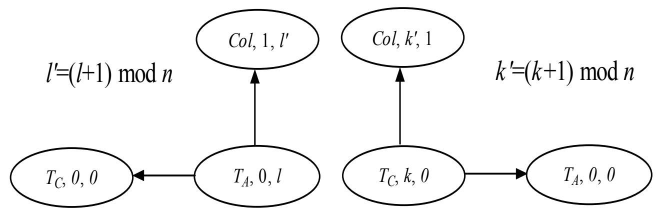

Figure 3: State Transitions from a T State

To illustrate the model clearly, we present the complete state transition diagram in Figure 4, where we assume n=3. The diagram is primarily based on the transitions diagrams shown in Figures 2 and 3. We realize that the diagram is quite complex even in such a simple case, and it is difficult to draw it on paper when n is large since the number of states is n^2+2n.

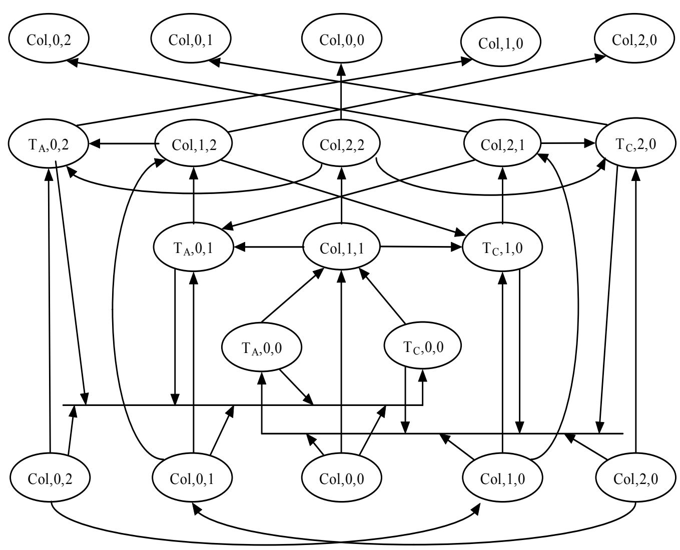

Figure 4: State Transition Diagram for n=3

3.2 Basic Analysis of the Model

For the convenience of presentation, rather than representing the system states using three variables as above, we assign a single variable to represent the states, by ordering them as indicated in Table 1.

Table 1: Re-designating the State Variables

States	Single Variable Range
(T_A, 0, 0) to (T_A, 0, n-1)	[0, n-1]

(T_C, *0*, *0*) to (T_C, *n-1*, *0*)	[*n*, *2n-1*]
(*Col*, *0*, *0*) to (*Col*, *n-1*, *n-1*)	[$2n$, n^2+2n-1]

For the embedded Markov chain, we need to know the transition probability matrix, *P*. Once we get the matrix *P*, we can find the steady state probability vector, π, by solving the following equation:

$$\begin{cases} \pi = \pi \cdot P & \pi = [\pi_0, \pi_1, \pi_2, \ldots], \quad P = [p_{ij}] \\ \sum_i \pi_i = 1 & i \in [0, n^2 + 2n - 1] \end{cases} \tag{5}$$

In a discrete Markov chain, if the interval between two *consecutive* transitions (including self-transition) is *identical*, the steady state probability π_i reflects the proportion of the *time* that the system is in state *i*. However, in our model, since the interval between a *T* state and its next state is a random variable, π_i can only tell us the probability that the system enters state *i* whenever a transition occurs [8]. To get the *time average* state probabilities of being in state *i*, we must first analyze the holding time of state *i*.

Let μ_{ji} denote the average time that the system will remain in state *i* once a transition from state *j* to *i* occurs. The μ_{ji} is given by:

$$\mu_{ji} = \begin{cases} 0 & p_{ji} = 0 \\ 1 & p_{ji} > 0 \; \& \; i \in [2n, n^2 + 2n - 1] \\ r \times num(j, i) & p_{ji} > 0 \; \& \; i \in [0, 2n - 1] \end{cases} \tag{6}$$

Whenever the transition probability p_{ji} is zero, the μ_{ji} must also be zero. The holding time in a *Col* state, since it does not depend upon the previous or the present state, is assigned a *unit* value as indicated in the second line of the above formula. For expressing the holding time of a *T* state, let us define *num(j, i)*, which denotes the average number of packets the transmitting node can transmit *consecutively* once the system reaches the *T* state (i.e., state *i*) from state *j*. Clearly, the holding time of a *T* state is proportional to *num(j, i)*. Moreover, we define the *'FES time'* as the time needed for a FES to be completed, and *'Col time'* as the time needed to detect the collision, while *r* is the ratio between the *'FES time'* and the *'Col time'*. Therefore, whenever the system transits to a *T* state, the average holding time can be represented by $r \times num(j, i)$, if the time required to detect a collision is unity. This explains the last line of equation (6). From μ_{ji}, which corresponds to a *transition*, we can get μ_i, the expected holding time of state *i*, as follows:

$$\mu_i = \sum_j \frac{\pi_j \times p_{ji}}{\pi_i} \mu_{ji} \tag{7}$$

Clearly, by the definition of a Markov chain, the state holding time of state *i* should *not* dependent on any previous state (i.e., state *j*), and thus the chain obtained here is indeed an approximation of the Markov model. Also, the average is calculated as in equation (7) since $\pi_i = \sum (\pi_j \times p_{ji})$. It is easy to see that the μ_i corresponding to a *Col* state is always equal to one. Corresponding to a *T* state, we define *num(i)* in equation (8), which denotes the average number of packets that the transmitting node can transmit consecutively once the system reaches state *i*.

$$num(i) = \sum_j \frac{\pi_j \times p_{ji}}{\pi_i} num(j, i) \quad i \in [0, 2n - 1] \tag{8}$$

As a result, formula (7) can be replaced by:

$$\mu_i = \begin{cases} num(i) \times r & i \in [0, 2n-1] \\ 1 & i \in [2n, n^2 + 2n - 1] \end{cases} \quad (9)$$

Now we can get ρ_i, which represents the *time average* state probability of state *i*, as follows [8]:

$$\rho_i = \frac{\pi_i \times \mu_i}{\sum_j \pi_j \times \mu_j} \quad i, j \in [0, n^2 + 2n - 1] \quad (10)$$

3.3 Derivation of the Metrics

Based on the above analysis, we can obtain the two metrics defined in Section 1. To recall, the *first* metric is, once a node gets control of the medium, what is the average number of packets this node can transmit *consecutively* without any collision. In our Markov chain model, let us say for node A, it is not possible for the system to travel from one T_A state to another T_A state without in-between visiting a T_C or *Col* state. Moreover, once the system enters a given T_A state (let us say, state *i*), the average number of units of *'FES time'* for which the system remains in that T_A state, is simply equal to *num(i),* which is defined by equation (8). Therefore, the metric-1 can be obtained by taking average of all *num(i)* corresponding to T_A states:

$$H_{metric-1} = \sum_{i \in [0, n-1]} \frac{\pi_i}{\pi(A)} \times num(i), \quad where\, \pi(A) = \sum_{i \in [0, n-1]} \pi_i \quad (11)$$

Since the behavior is identical at nodes A and C, the $H_{metric-1}$ is also applicable to node C.

Now, we recall that the *second* metric is, once a node loses its control of the medium, what is the average time the node has to wait before it gets control to the medium again. Since there are only two nodes (i.e., A and C) contending for the medium (Figure 1), the metric, let us say for node A, can be replaced by: once the medium is controlled by node A, what is the average number of packets that A can transmit before the medium is controlled by node C. This simply implies that *once the system enters a T_A state, what is the average number of units of 'FES time' that the system can remain in any T_A state (via visiting Col state in-between) before the system enters a T_C state.* Note that the second metric allows a visit to a *Col* state in-between two T_A states, which is not permitted in the first metric.

Here we use the concept of "*expected first passage time*" [8], which means that if the system starts in a given T_A state (let us say, state *i*), what is the expected time after which the system will enter any T_C state for the *first* time. The expected first passage time, V_i, can be expressed as follows [8]:

$$V_i = \begin{cases} 0 & i \in [n, 2n-1] \\ R_i + \sum_j p'_{ij} V_j & i \in [0, n-1] \cup [2n, n^2 + 2n - 1] \end{cases} \quad (12)$$

where R_i is the *immediate reward* once the system enters state *i*. For a T_A state, it is equal to the corresponding *num(i)*. Since all T_C states are the *trapping* states, R_i is zero for these states [8]. We should also assign R_i with zero for all *Col* states, as our objective is to obtain the number of FESs, rather than the number of collisions. Therefore, R_i:

$$R_i = \begin{cases} num(i) & i \in [0, n-1] \\ 0 & i \in [n, n^2 + 2n - 1] \end{cases} \quad (13)$$

In equation (12), p'_{ij} is a modified value of the original transition probability p_{ij}. Since the T_C states are considered as *trapping* states, the transition probabilities out of a T_C state is set to zero, whereas the self-transition probability for each T_C state is set to one. Other transition probabilities remain unchanged. Therefore, the p'_{ij}:

$$p'_{ij} = \begin{cases} 0 & i \in [n, 2n-1] \; \& \; j \neq i \\ 1 & i \in [n, 2n-1] \; \& \; j = i \\ p_{ij} & i \in [0, n-1] \bigcup [2n, n^2 + 2n - 1] \end{cases} \tag{14}$$

Obviously, V_i corresponding to the trapping states (i.e., all T_C states) should be zero. For all the other states, V_i is equal to the 'immediate reward' R_i plus the expected reward earned from whatever state is entered next. This explains the equation (12).

The metric-2 is obtained by taking average over all V_i values corresponding to the T_A states:

$$H_{metric-2} = \sum_{i \in [0, n-1]} \frac{\pi_i}{\pi(A)} V_i, \quad where \quad \pi(A) = \sum_{i \in [0, n-1]} \pi_i \tag{15}$$

The $H_{metric\text{-}2}$ computed for node A, is also applicable to node C.

To solve the above equations, we *only* need to know the transition probability matrix P and num(j, i), which are computed in sections 4 and 5, respectively.

4 Transition Probability Computation

The transition probabilities depend upon the value of the back-off timers at nodes A and C. In each of the subsections 4.1 and 4.2, we first compute the transition probabilities in general terms based on the possible ranges of the back-off timers, and then obtain the specific transition probabilities.

4.1 Transition Probabilities out of a *Col* State

Assuming two nodes are contending for the medium, after a collision, each node will generate a random number from its CW according to the uniform distribution. Let us denote the two random numbers by X and Y, and their corresponding CW by CW_x and CW_y. Moreover, we denote T_x as a T state under which the node, whose contention window is CW_x, gets control of the medium, while T_y is a T state corresponding to the node with the CW_y. It is clear that the difference between X and Y determines what is the next state and what is the corresponding transition probability. As in formula (3), we refer to this difference as Z, i.e., $Z=X-Y$. Since one of the CWs must be smaller than or equal to the other one, we can assume $CW_x \leq CW_y$ and thus Z is in the range $[-CW_y, CW_x]$. As a result, all the possible transition probabilities from *Col* states can be expressed as follows in terms of Z:

$$\begin{aligned} &\Pr(T_x \mid (Col, CW_x, CW_y)) = \Pr(-CW_y \leq Z < -Len) \\ &\Pr(Col \mid (Col, CW_x, CW_y)) = \Pr(-Len \leq Z \leq Len) \\ &\Pr(T_y \mid (Col, CW_x, CW_y)) = \Pr(Len < Z \leq CW_x) \end{aligned} \tag{16}$$

The distribution function of Z can be easily derived as follows:

$$\Pr(Z = z) = \begin{cases} (CW_y + z + 1) P_{xy} & z \in [-CW_y, CW_x - CW_y] \\ (CW_x + 1) P_{xy} & z \in [CW_x - CW_y, 0] \\ (CW_x - z + 1) P_{xy} & z \in [0, CW_x] \end{cases} \tag{17}$$

$$where \quad P_{xy} = 1/((1 + CW_x)(1 + CW_y)) .$$

We can compute the transition probabilities using equations (16) and (17). It is easy to see that the transition probabilities depend upon the value of *Len* with respect to those of CW_x, CW_y, and $(CW_x - CW_y)$. We here show in Figure 5 how to obtain the $Pr(T_x |(Col, CW_x, CW_y))$.

Similarly, we obtained all the other probabilities under various conditions. However, due to the space limitation, we directly present the results in Figure 6 under the assumption that $0<Len<CW_{min}$, which is true in the considered case where $Len = 19$ and $CW_{min} = 31$. Clearly, when $CW_x < CW_y$, the $CW_y - CW_x$ must be greater than CW_{min}. Therefore, in this case, *Len* must be in the range of $[0, CW_y - CW_x]$.

$$\text{When } Len \in [0, CW_y - CW_x)$$

$$\Pr(T_x \mid Col, CW_x, CW_y) = \sum_{-CW_y}^{-Len-1} P(z)$$

$$= P_{xy}\{ \sum_{-CW_y}^{CW_x - CW_y - 1} (CW_y + z + 1) + \sum_{CW_x - CW_y}^{-Len-1} (CW_x + 1)\}$$

$$= \frac{CW_y - Len - CW_x / 2}{CW_y + 1}$$

$$\text{When } Len \in [CW_y - CW_x, CW_y)$$

$$\Pr(T_x \mid Col, CW_x, CW_y) = \sum_{-CW_y}^{-Len-1} P(z)$$

$$= P_{xy} \sum_{-CW_y}^{-Len-1} (CW_y + z + 1) = \frac{(CW_y - Len)(1 + CW_y - Len)}{2(1 + CW_x)(1 + CW_y)}$$

$$\text{When } Len \in [CW_y, \infty)$$

$$\Pr(T_x \mid Col, CW_x, CW_y) = \sum_{-CW_y}^{-Len-1} P(z) = 0$$

Figure 5: Computation of $Pr(T_x|Col,CW_x,CW_y)$

If $CW_x = CW_y = cw$:

$$\Pr(Col|(Col,cw,cw)) = \frac{1+cw+2cw\times Len+Len-Len^2}{(1+cw)^2}$$

$$\Pr(T_y|(Col,cw,cw)) = \Pr(T_x|(Col,cw,cw)) = \frac{(cw-len)(1+cw-len)}{2(cw+1)^2}$$

If $CW_x < CW_y$:

$$\Pr(T_x|(Col,CW_x,CW_y)) = \frac{CW_y-Len-CW_x/2}{CW_y+1}$$

$$\Pr(T_y|(Col,CW_x,CW_y)) = \frac{(CW_x-Len)(1+CW_x-Len)}{2(1+CW_x)(1+CW_y)}$$

$$\Pr(Col|(Col,CW_x,CW_y)) = \frac{1+(3-Len)\times Len/2+CWx(1+2Len)}{(1+CW_y)(1+CW_x)}$$

Figure 6: Transition Probabilities out of a *Col* state

Now it is easy to compute the transition probabilities out of a given Col state. This is achieved by translating the stages at the two nodes to the corresponding CWs, and then, assign the smaller CW to CW_x and the greater one to CW_y.

4.2 Transition probabilities out of a *T* state

Once the system enters a *T* state, the transmitting node may transmit one or several packets before the waiting node's back-off timer reduces to zero. Figure 7 shows how the back-off timer at the *waiting* node decrements.

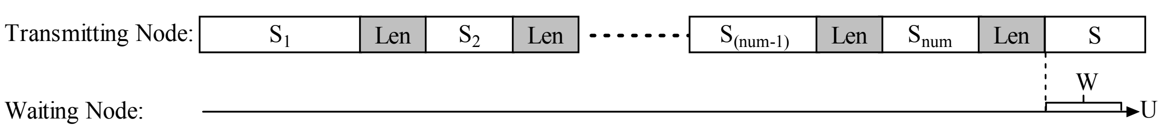

Figure 7: Counting Down of the Back-off Timer at the Waiting Node

In the diagram, *U* denotes the back-off timer at the waiting node when the system enters a *T* state, and *num* (≥ 1) denotes the number of packets transmitted by the transmitting node before the back-off timer at the waiting node reduces to zero. S_k represents the back-off timer at the transmitting node corresponding to the k^{th} consecutive FES. Now let us consider how much the *U* will be reduced during each *'FES time'*. During each S_k, the waiting node counts down its back-off timer. From the time, the transmitting node initiates a RTS, to the time, the receiving node begins to send back a CTS, the waiting node also counts down its back-off timer. This period is equal to *Len*. As soon as the waiting node overhears the CTS, it *freezes* its back-off timer. Therefore, during each *'FES time'*, the waiting node reduces its back-off timer by (S_k +*Len*). When the last FES (i.e., the num^{th} FES) is completed, the transmitting node generates a new random number (let us say, *S*), while the waiting node simply counts down its frozen back-off timer. *W* denotes the *remaining* value of the frozen back-off timer, which can be expressed as:

$$W = U - \sum_{k=1}^{num} (S_k + Len) \qquad (18)$$

To find the next state and the corresponding transition probability, we need to know the ranges and the distributions of S and W. It is easy to see that S is uniformly distributed over the interval [0, CW_{min}]. However, it is not so trivial to get the distribution of W.

It is intuitive that, when the CW at the waiting node (let us say, CW_u), from which U is drawn, is large, the *num* is also large. Though, to get the distribution of W, computationally it is possible to consider all values of CW_u, for simplicity, we consider only two cases here: $CW_u = CW_{min}$ and $CW_u > CW_{min}$.

When $CW_u = CW_{min}$, the *num* is unity, therefore, equation (18) can be simplified as:

$$W = U - (S_1 + Len) \qquad (19)$$

We note that the (U-S_1) must be greater than Len, otherwise the system will not enter the T state. Moreover, as U is drawn from the range [0, CW_{min}], the range of S_1 must be [0, CW_{min} - Len]. Therefore, from equation (19), W must be in the range of [0, CW_{min} - Len]. Since both S_1 and U are uniformly distributed and their ranges are known, the distribution of W can be derived as follows:

$$\Pr(W = w) = \frac{2(1 + CW_{\min} - Len - w)}{(1 + CW_{\min} - Len)(2 + CW_{\min} - Len)} \qquad (20)$$

When $CW_u > CW_{min}$, we note that W must be smaller than $CW_{min} + Len$, otherwise, the *transmitting* node will be able to transmit one more packet. Since in this case, it is difficult to obtain the exact distribution of W, we approximate it with a uniform distribution in the range [0, CW_{min}].

Once we know the ranges and the distributions of the back-off timers (i.e., S and W), we can obtain the transition probabilities as done in Section 4.1. However, we here present the results directly in Figure 8.

If $CW_u = CW_{min}$:

$$\Pr(T \mid CW_u = CW_{\min}) = \frac{2(CW_{\min} - Len)}{3(1 + CW_{\min})}$$

$$\Pr(Col \mid CW_u = CW_{\min}) = \frac{3 + CW_{\min} + 2Len}{3(1 + CW_{\min})}$$

If $CW_u > CW_{min}$:

$$\Pr(\mathrm{T|CW_u} > CW_{\min}) = \frac{(CW_{\min} - Len)(1 + CW_{\min} - Len)}{2 + (3 + CW_{\min})CW_{\min} + Len + 2Len \times CW_{\min} - Len^2}$$

$$\Pr(\mathrm{Col|CW_u} > CW_{\min}) = \frac{2(1 + CW_{\min} + Len + 2Len \times CW_{\min} - Len^2)}{2 + (3 + CW_{\min})CW_{\min} + Len + 2Len \times CW_{\min} - Len^2}$$

Figure 8: Transition Probability out of a T state

Now the transition probabilities can be found according to the following rule: if the stage at the *waiting* node is equal to zero, the probabilities corresponding to $CW_u = CW_{min}$ are used, otherwise, the other probabilities are used.

5 State Holding Time Computation

Now we show how to compute *num(j, i),* the average number of packets the *transmitting* node can transmit consecutively once the system reaches a *T* state (i.e., state *i*) from state *j*. If the previous state (i.e., state *j*) is also a *T* state, both the stage number in the present state must be *zero* (see Figure 3)*,* and therefore *num(j, i)* is unity in this case. However, if the previous state is a *Col* state, computation of *num(j, i),* is not so trivial.

In section 4.2, when the system enters a *T* state (Figure 7), the back-off timer at the waiting node is *U* while the back-off timer at the transmitting node is S_1. The corresponding CWs are CW_{s_1} and CW_u, respectively. It is easy to see that, when the system is in the state *j* (i.e., previous state), the CWs are also CW_{s_1} and CW_u. Moreover, we have concluded that, during each *'FES time'*, the waiting node reduces its back-off timer by $S_k + Len$. Since S_k (when k>1) is uniformly distributed over the range [*0*, CW_{min}], the average of S_k (when k>1) is (CW_{min} +1)/2. However, as the CW_{s_1} may not be the same as CW_{min}, we need to compute the average value of S_1 *separately*. When the last FES (i.e., the *num*th FES) is completed, the remaining back-off timer at the waiting node is *W*, which is a very small value as seen from Section 4.2, and so its average value can be neglected. Therefore, by renewal theory, the *num(j, i)* can be expressed as:

$$num(j,i) = 1 + \frac{Avg(U) - (Avg(S_1) + Len) - Avg(W)}{Avg(S_k\,;\,k>1) + Len} \approx 1 + \frac{Avg(U) - Avg(S_1) - Len}{(CW_{\min}+1)/2 + Len} \qquad (21)$$

In order to solve (21), we only need to know the average values of *U* and S_1. Since we are considering the average values of *U* and S_1 under the *condition* that the system enters a *T* state (i.e., *U*- S_1>*Len*), we must first derive the *conditional* distribution of *U and* S_1, from which we can obtain their average values. In the following we illustrate how to compute the average value of S_1. The conditional distribution of S_1 is:

$$\Pr(S_1 = s \mid (U - S_1 > Len)) = \frac{\Pr(S_1 = s, U - s > Len)}{\Pr(U - S_1 > Len)} = \frac{\Pr(S_1 = s)\times \Pr(U > s + Len)}{\Pr(U - S_1 > Len)} = \frac{P_{s_1} \times \{P_u \times (CW_u - Len - s)\}}{\Pr(U - S_1 > Len)}$$

where $P_{s_1} = 1/(CW_{s_1}+1)$ and $P_u = 1/(CW_u + 1)$.

In the above formula, Pr(*U*- S_1 > *Len*) is equal to $(CW_u - Len - CW_{s_1}/2)/(CW_u + 1)$ if $CW_u > CW_{s_1}$, which can be easily computed from the formula (17). Therefore,

$$\Pr(S_1 = s \mid (U - S_1 > Len)) = \frac{2(Len + s - CW_u)}{(1 + CW_{s_1})(2Len + CW_{s_1} - 2CW_u)}$$

The average of S_1 under the condition *U*- S_1>*Len* (i.e., $E(S_1|\ldots)$) is:

$$E(S_1 \mid (U - S_1 > Len)) = \sum_{s=0}^{CW_{s_1}} s \times \Pr(S_1 = s \mid (U - S_1 > Len)) = \frac{CW_{s_1}(3CW_u - 3Len - 2CW_{s_1} - 1)}{6(CW_u - Len - CW_{s_1}/2)}$$

Similarly, we can get the average value of U, and then using equation (21) we can evaluate *num(j, i)*. The results are directly presented in Figure 9.

When $\mathrm{CW_u} > \mathrm{CW_{s_1}}$:

$$E(S_1 \mid (U - S_1 > Len)) = \frac{CW_{s_1}(3CW_u - 3Len - 2CW_{s_1} - 1)}{6(CW_u - Len - CW_{s_1}/2)}$$

$$E(U \mid (U - S_1 > Len)) = \frac{1}{12}\left(6(1 + CW_u + Len) + CW_{s_1}\left(3 + \frac{2 + CW_{s_1}}{CW_{s_1} - 2CW_u + 2Len}\right)\right)$$

$$num(j,i) = \frac{-2CW_{s_1}^2 - 6(CW_u - Len)(2 + CW_{\min} + CW_u + Len) + CW_{s_1}(5 + 3CW_{\min} + 6CW_u)}{3(1 + CW_{\min} + 2Len)(CW_{s_1} - 2CW_u + 2Len)}$$

When $\mathrm{CW_u} \doteqdot \mathrm{CW_{s_1}}$:

$$E(S_1 \mid (U - S_1 > Len)) = \frac{1}{3}(-1 + CW_u - Len)$$

$$E(U \mid (U - S_1 > Len)) = \frac{1}{3}(1 + 2CW_u + Len)$$

$$num(j,i) = \frac{7 + 2CW_u + 3CW_{\min} + 4Len}{3(1 + CW_{\min} + 2Len)}$$

Figure 9: State Holding Time Computation

Now it is easy to compute a specific *num(j, i)* when state *j* is a *Col* state. To achieve this, the stages in state *j* are first translated to the corresponding CWs, and then the CW belonging to the *present transmitting* node is assigned to CW_{s_1} and the other one to CW_u.

Before we present the numerical results, we would like to restate the main approximations used in the analytical model: *(i)* In equation (3), we have neglected the propagation delay and the time difference between two nodes detecting a collision. *(ii)* In equation (7), we obtained the average state holding time using a Markov approximation. *(iii)* In the text right below equation (20), we have approximated the *W* with a uniform distribution. *(iv)* In equation (21), we have neglected the value of *W*. *(v)* In the derivation and the results presented in the next section, we assume that CW_{min} = 31 and *Len* =19. Clearly, all these approximations may lead to error in the performance prediction. Nevertheless, the results presented in the next section show that the model predicts the short-term fairness very accurately.

6 Numerical Results

In Section 6.1, values of the transition probabilities and *num(j, i)* calculated from the Makov chain are presented. Based on these results, in Section 6.2, we compute the metrics and compare them to those obtained from the simulation.

6.1 Transition Probabilities and State Holding Time

Transition Probabilities in Col States:

The transition probabilities from the *Col* states are presented in Table 2, which are computed from the formulas derived in Section 4.1. From the results, we can see that, when the CWs are small (e.g., 31), the probability of the system transiting to another *Col* state is very large (e.g., 0.848). To resolve collisions, we must reduce this probability, so that the system is more likely to enter a *T* state. From the table we see that the transition probability from a *Col* state to another *Col* state is small in the following two situations: *(i)* when both the CWs are large, *(ii)* when the difference between the two CWs is large. However, in such situations, if the system *does* enter a *T* state, the system will remain in that *T* state for a long time (this is clearly brought out by the later results), which results in short-term unfairness. *This shows that there is a fundamental conflict in IEEE 802.11 between resolving collisions and achieving short-term fairness in the hidden-terminal scenario.*

Table 2: Transition Probabilities out of a *Col* State

CW_x	CW_y	$Pr(T_x \mid Col)$	$Pr(T_y \mid Col)$	$Pr(Col \mid Col)$
31	31	0.076	0.076	0.848
	63	0.445	0.038	0.517
	127	0.723	0.019	0.258
	255	0.861	0.010	0.129
	511	0.931	0.005	0.064
	1023	0.965	0.002	0.033
63	63	0.241	0.241	0.518
	127	0.598	0.121	0.281
	255	0.799	0.060	0.141
	511	0.900	0.030	0.070
	1023	0.950	0.015	0.035
127	127	0.359	0.359	0.282
	255	0.674	0.180	0.146
	511	0.837	0.090	0.073
	1023	0.918	0.045	0.037
255	255	0.426	0.426	0.148
	511	0.712	0.213	0.075
	1023	0.856	0.107	0.037
511	511	0.462	0.462	0.076
	1023	0.731	0.231	0.038
1023	1023	0.481	0.481	0.038

Transition Probabilities in T States:

Similarly, we compute the transition probabilities out of a *T* state (see Table 3) using the results of Section 4.2. We see that the transition probability from a *T* state to another *T* state is small in comparison to the transition probability from a *T* state to a *Col* state. This shows, after a node loses control of the medium, the system is more likely to experience collisions before the other node gets control of the medium.

Table 3: Transition Probabilities out of a *T* State

CW_u	Pr(Col|T)	Pr(T|T)
31	0.75	0.25
>31	0.918	0.082

State Holding Time: num(j, i)

As discussed in Section 5, the *num(j, i)* is equal to unity when the previous state (i.e., state *j*) is also a *T* state. When the previous state is a *Col* state, the values of *num(j, i)* are presented in Tables 4. From the table, we can see that when the two CWs are very large (e.g., both are 1023), or the difference between the two CWs is very large (e.g., one is 31 and the other one is 1023), the *num(j, i)* is very large (e.g., 10.581 and 15.137, respectively). *This strengthens our earlier conclusion that resolving collisions and achieving short-term fairness are contradictory objectives when using IEEE 802.11 in a hidden terminal scenario.*

Table 4: num(j, i) when the state *j* is a *Col* state

CW_{s1}	CW_u	Avg(S_1)	Avg(U)	num(j,i)
≡31	31	3.667	27.333	1.133
31	63	12.509	47.754	1.464
	127	14.578	80.790	2.349
	255	15.113	145.057	4.170
	511	15.321	273.161	7.824
	1023	15.414	529.207	15.137
≡63	63	14.333	48.667	1.438
63	127	27.039	87.020	2.171
	255	29.831	152.416	3.960
	511	30.759	280.879	7.603
	1023	31.149	537.075	14.912
≡127	127	35.667	91.333	2.048
127	255	55.586	165.293	3.592
	511	60.314	295.657	7.181
	1023	62.048	552.524	14.471
≡255	255	78.333	176.667	3.267
255	511	112.517	321.759	6.435
	1023	121.269	582.135	13.625
≡511	511	163.667	347.333	5.705
511	1023	226.315	634.657	12.124
1023	1023	334.333	688.667	10.581

6.2 Comparison between Analytical and Simulation Results

In this section, we evaluate the equations derived in Section 3, and the analytical results are compared with those obtained from the simulation. The simulation environment is the same as described in Section 2. The results correspond to the case when $CW_{min}=31$, $Len=19$, $m=5$, and $n=7$, which are typically used in IEEE 802.11. Since our main objective is to analyze the behavior at the T states, and the behavior at nodes A and C are identical, we *only* present the results for the T_A states.

State Probabilities:

In Table 5, we present the values of π_i and ρ_i. To recall, π_i represents the proportion of transitions entering state i, while ρ_i reflects the proportion of time spent in state i. We can see that the analytical results are quite close to those of simulation. We also notice that, though the sum of π_i is quite small (about 0.24), the sum of ρ_i is quite large (0.496). Since the T_C states also have the same values, the total fraction amount of *time* spent in T states is about 0.992, which implies that only a very small amount of time is spent in the large number of *Col* states. The reason is that the ratio (i.e., r) between the *'FES time'* and the *'Col time'* is very large (i.e., 20 in our case). Note that in the above discussion, we have excluded the time spent in the *idle* states. However, as mentioned in Section 2, the aggregate throughput of the two flows in the hidden-terminal scenario is about 1.36 Mbps. In comparison, the aggregate throughput in a single-hop scenario with also two contending flows is about 1.43 Mbps. Therefore, though the hidden-terminal problem leads to considerable collisions, it does not result in substantial capacity wastage in our context. This shows the advantage of using the *short* RTS/CTS frames before the long Data frame is transmitted. However, we notice that the throughput degradation due to collisions becomes substantial when the ratio *r* becomes smaller, which is true in the emerging standards such as IEEE 802.11 A and G.

Table 5: State Probabilities Comparison

From		$(T_A,0,0)$	$(T_A,0,1)$	$(T_A,0,2)$	$(T_A,0,3)$	$(T_A,0,4)$	$(T_A,0,5)$	$(T_A,0,6)$	Total
π_i	Model	0.025	0.031	0.037	0.038	0.037	0.035	0.032	0.236
	Simulation	0.025	0.038	0.039	0.039	0.037	0.034	0.031	0.243
ρ_i	Model	0.008	0.014	0.025	0.046	0.088	0.165	0.150	0.496
	Simulation	0.008	0.015	0.023	0.045	0.088	0.167	0.150	0.496

Expected State Holding Time:

Table 6 presents the results of *num(i)*, which denotes the average number of packets node A can transmit consecutively once the system enters a given T_A state (i.e., state *i*). We see that the results match very closely. As the *stage* at the *waiting* node (i.e., node C) increases, the *num(i)* also increases, which indicates that it becomes more unfair for the node C.

Table 6: Expected Holding Time Comparison

num(i)	$(T_A,0,0)$	$(T_A,0,1)$	$(T_A,0,2)$	$(T_A,0,3)$	$(T_A,0,4)$	$(T_A,0,5)$	$(T_A,0,6)$
Model	1.006	1.450	2.143	3.830	7.496	14.858	14.893
Simulation	1.002	1.206	1.829	3.616	7.461	15.336	15.348

Expected First Passage Time:

Table 7 presents the results of V_i, which represents, if the system starts in a given T_A state (i.e., state *i*), what is the expected number of *FESs* after which the system will enter any T_C state for the *first* time. Again, the results obtained from the model are quite close to those from simulation. When the stage at the waiting node (i.e., node C) increases (from 0 to 4), the V_i also increases, which implies that it becomes more unfair for node C. However, when the stage at the node C further increases (i.e., from 4 to 5, and then 6), the V_i decreases. First, let us explain why the V_i corresponding to the (T_A, 0, 6) state is small. We recall that V_i is equal to the immediate reward (i.e., *num(i)*) plus the expected reward earned from whatever state is entered next. When the system departs from this (T_A, 0, 6) state, the system enters the state (*Col*, 1, 0) where the stage at node C has been *reset*. From this *Col* state, the system is more likely to enter a T_C state, in comparison to, from other *Col* states. For instance, when the system transits to a *Col* state from the (T_A, 0, 5) state, the stage at node C will not be reset, and the probability of transiting to a T_C state is small. This implies that after leaving (T_A, 0, 6) state, the expected reward earned from the future states is smaller in comparison to that after leaving the state (T_A, 0, 5). Therefore, the V_i corresponding to (T_A, 0, 6) is small compared to the (T_A, 0, 5) state. Now, we explain why the V_i corresponding to the (T_A, 0, 5) state is smaller than that for the (T_A, 0, 4) state. Since the state (T_A, 0, 6) is a *future* state of (T_A, 0, 5), a small V_i for (T_A, 0, 6) will also affect the V_i for the (T_A, 0, 5) state. However, the effect of the reset behavior decreases rapidly as the stage at the waiting node becomes smaller than 4. From the above discussion, it is clear that *the resetting mechanism* *adopted in the IEEE 802.11 improves the short-term fairness.*

Table 7: Expected First Passage Time Comparison

V_i	$(T_A,0,0)$	$(T_A,0,1)$	$(T_A,0,2)$	$(T_A,0,3)$	$(T_A,0,4)$	$(T_A,0,5)$	$(T_A,0,6)$
Model	11.796	25.416	31.081	34.059	34.404	30.273	17.661
Simulation	10.413	23.584	29.998	34.031	34.896	31.430	18.555

Metrics:

Table 8 presents the values of the two metrics defined earlier. We also present the corresponding maximum values. Again, the analytical results match the simulation results. Note that the values of the two metrics differ largely. The reason is that whenever the system departs from a T_A state, the probability that the system enters a *Col* state is very large (See Table 3). After the collision(s), the system is more likely to enter another T_A state (rather than a T_C state) as the CW at node A is smaller than that at node C. This may be repeated several times, resulting in such a large difference.

Table 8: Comparison of the Metrics

From	metric-1	Max metric-1	metric-2	Max metric-2
Model	6.683	NA	27.379	NA
Simulation	6.413	35	27.090	160

General Applications of the Model

The above results correspond to the case when $CW_{min}=31$, $Len=19$, $m=5$, and $n=7$. By varying parameters n, m, CW_{min}, and Len, the short-term behavior for other scenarios can be obtained. For example, by making *Len=TxTime(Data)+SIFS*, we can model the *two-way* handshake in the presence of hidden terminals. On the other hand, by making $Len=0$, we can model *anyone* of the two handshakes without hidden terminals. By varying n, m and CW_{min}, we can model different physical layers, such as FSSS, DSSS and IR [9]. Moreover, since the model can predict the short-term behavior *precisely*, it would also predict the long-term behavior accurately.

7 Discussion and related work

7.1 Future work

In this paper, we have presented a novel embedded-Markov model to study the short-term unfairness in a simple 3-nodes hidden-terminal case. One obvious future work is to extend the model into a scenario with more number of hidden terminals. This may not be trivial. However, it is necessary to mention that the modeling process described in Section 3 is quite general for the study of *short-term* behavior, especially the adoption of the *first passage time*.

Another focus is to propose a solution to cope with the short-term unfairness problem. From the results, we have already seen that the resetting CW mechanism improves the short-term fairness. Therefore, in addition to the standard resetting mechanism, we are of the opinion that the CW should also be reset whenever the short-term unfairness occurs while the short-term unfairness can be detected using dynamic measurements. Our preliminary results [12] show that this method improves the fairness. However, the aggregate throughput may degrade.

7.2 Related work

In the literature, many analytical models have focused on the calculation of the throughput (or capacity) of IEEE 802.11. For example, paper [3] derives the theoretical throughput of IEEE 802.11 by relating IEEE 802.11 to a p-persistent CSMA protocol. Paper [17] derives the throughput of IEEE 802.11 by using average analysis technique. However, the above two models have over-simplified the calculation of the transmission probability (normally represented by τ), which characterizes the main feature of the BEB algorithm. In contrast, paper [2] derives a more precise value of τ by modelling the stochastic process representing the back-off time counter as a discrete-time Markov chain. Furthermore, paper [20] extends the model by considering the frame retry limit used in IEEE 802.11. While most of the published models focus on the throughput of IEEE 802.11, two recent work [4, 18] concentrate on the analysis of the packet delay in a single-hop IEEE 802.11 network. As mentioned in the introduction, while the above models capture the long-term behavior (e.g., capacity or packet delay) of IEEE 802.11 very well in a single-hop scenario, they are not appropriate to study the short-term unfairness in a hidden-term scenario.

In another related work [11], the authors have studied the short-term fairness by first developing two short-term fairness metrics and then applying the metrics in analyzing two MAC protocols: CSMA/CA and ALOHA. Though IEEE 802.11 is mainly based on CSMA/CA, it has many other features, and therefore, [11] cannot be applied to IEEE 802.11. Moreover, they do not consider the hidden-terminal problem and they mainly focus on developing general fairness metrics, which are different from our work.

We should note that the short-term unfairness problem is also known as “capture” in Ethernet [7, 16]. However, the main reason of capture in Ethernet is due to the deficiency of the BEB algorithm when the number of contending stations is very large. Moreover, the capture phenomenon scarcely occurs in an Ethernet when there are only two contending stations. On the contrary, in an IEEE 802.11 wireless network, the capture phenomenon poses severe problem even in the scenario with just two contending stations (e.g., in the hidden-terminal scenario). Clearly, in addition to the deficiency of the BEB, the *freezing* mechanism of the back-off timer, and the hidden-terminal problem itself (which is rooted in wireless networks) causes unfairness.

At last, while there are very few analytical models to *quantify* the unfairness of IEEE 802.11, there is a large number of papers (e.g., [1, 10, 13-15, 19]) focusing on developing algorithms to solve the unfairness problem in wireless networks.

8 Conclusions

In this paper, we have presented an *analytical* model that is *extremely accurate* in predicting the *short-term* unfairness of the system, which is not available in the literature. Our main contributions include: *(i)* a Markov chain model to depict the IEEE 802.11 in detail, *(ii)* conclusion that the IEEE 802.11 exhibits substantial short-term unfairness in presence of hidden terminals, *(iii)* deduction that resolving collisions and achieving short-term fairness are contradictory objectives when using IEEE 802.11 in a hidden terminal scenario, and finally *(iv)* the resetting of the contention window improves the short-term fairness.

References

[1] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, “MACAW: A Media Access Protocol for Wireless LANs,” ACM SIGCOMM, 1994.

[2] G. Bianchi, “Performance Analysis of the IEEE 802.11 Distributed Coordination Function,” IEEE Journal on Selected Areas in Communications, March 2000, pp.535-547.

[3] 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, December 2000, pp.785-799.

[4] M. M. Carvalho, J. J. Garcia-Luna-Aceves, ``Delay Analysis of IEEE 802.11 in Single-hop Networks," in IEEE ICNP, 2003.

[5] D.M. Chiu, R. Jain, “Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks,” Computer Networks and ISDN Systems 17 (1989),1-14.

[6] CMU Monarch Group. CMU Monarch Extensions to NS, http://www.monarch.cs.cmu.edu/.

[7] S. Deng, “Capture Effect in Residential Ethernet LAN”. In Proc. IEEE GLOBECOM (Nov. 1995).

[8] R.G. Gallager, “Discrete Stochastic Processes,” Kluwer Academic Publishers, Boston,1996.

[9] IEEE, “Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications,” IEEE 802.11 standards, June 1999.

[10] V. Kanodia, C. Li, A. Sabharwal, B. Sadeghi, E, Knightly. “Ordered Packet Scheduling in Wireless Ad Hoc Networks: Mechanisms and Performance Analysis,” in ACM MOBIHOC, 2002.

[11] C.E. Koksal, H. Kassab, H. Balakrishnan, “An Analysis of Short-Term Fairness in Wireless Media Access Protocols,” ACM SIGMETRICS, 2000.

[12] Z.F. Li, S. Nandi, A.K. Gupta, ``Achieving MAC Fairness in Wireless Ad-hoc Networks using Adaptive Transmission Control," in IEEE ISCC, 2004.

[13] H. Lou, S. Lu, V. Bharghavan, “A New Model for Packet Scheduling in Multi-hop Wireless Networks,” ACM MOBICOM, 2000.

[14] S. Lu, T. Nandagopal and V. Bharghavan, "A Wireless Fair Service Algorithm for Packet Cellular Networks," ACM MOBICOM'98, Dallas, TX, October 1998.

[15] T. Nandagopal, T. Kim, X. Gao, V. Bharghavan, “Achieving MAC Layer Fairness in Wireless Packet Networks,” ACM MOBICOM, 2000.

[16] K. K. Ramakrishna and H. Yang, "The Ethernet Capture Effect: Analysis and Solution," in Proc. 19th Conference on Local Computer Networks (LCN '94), (Minneapolis, MN, USA), Oct. 1994.

[17] Y.C. Tay, K.C. Chua, “A Capacity Analysis for the IEEE 802.11 MAC protocol,” ACM Wireless Network, 7, 2001.

[18] O. Tickoo, B. Sikdar, ``Queueing Analysis and Delay Mitigation in IEEE 802.11 Random Access MAC based Wireless Networks," in IEEE INFOCOM, 2004.

[19] N.H. Vaidya, P. Bahl, S. Gupta, ``Distributed fair scheduling in a wireless LAN," in ACM MOBICOM, 2000.

[20] H.T. Wu, Y. Peng, K.P. Long, S.D. Cheng, J. Ma, ``Performance of Reliable Transport Protocol over IEEE 802.11 Wireless LAN: Analysis and Enhancement," in IEEE INFOCOM, 2002.