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

Zhifei Li[1], Sukumar Nandi[2], and Anil K. Gupta[1]

[1] School of Computer Engineering, Nanyang Technological University, Singapore, 639798
E-mail: {pg03802331, asgupta}@ntu.edu.sg
[2] Dept. of Computer Science & Engineering, Indian Institute of Technology, Guwahati, India, 781039
E-mail: sukumar@iitg.ernet.in

Abstract. In this paper, using a simple hidden-terminal scenario, we show that IEEE 802.11 exhibits substantial short-term unfairness, though it provides long-term fairness. We analyze the short-term behavior using 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.

1 Introduction

IEEE 802.11 [6] is the de facto standard for Wireless LANs, and it defines two MAC protocols: Point Coordination Function (PCF) and Distributed Coordination Function (DCF). However, only the DCF is popular. As DCF operates in a *distributed* manner, achieving fairness in accessing the medium is one of the most challenging issues. 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 automatically gives rise to long-term fairness, but not the vice versa [7]. 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 important for the adaptive traffic (e.g., TCP traffic) and for the delay- or jitter-sensitive traffic [7]. In this paper, we aim to analyze the short-term unfairness of IEEE 802.11.

The *duration*, over which the short-term fairness should be measured, is difficult to define as it depends upon the requirements of applications as well as upon the channel bandwidth. To get around this problem, Jain's index [7] can be used to reflect fairness over different time scales. 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.

In this paper, using an embedded Markov chain model, the above 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. Our results show that the IEEE 802.11 exhibits substantial short-term unfairness even in a very simple hidden-terminal scenario.

The remainder of the paper is organized as follows. In Section 2, using simulation method, we show that IEEE 802.11 exhibits considerable short-term unfairness in the hidden-terminal scenario. In Section 3, the Markov chain model is described. Section 4 presents the analytical and simulation results. Related work is reviewed in Section 5 and the paper is concluded in Section 6.

2 Short-Term Unfairness in IEEE 802.11

To show the short-term unfairness in IEEE 802.11, we simulate the well-known hidden-terminal scenario depicted in Figure 1. 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.

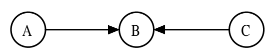

Fig. 1. Hidden-terminal Scenario

In the simulation, NS-2 with CMU wireless extensions [4] is used. For each single-hop flow, a Constant Bit Rate (CBR) traffic is adopted, where each packet is 1000-bytes long. The raw bandwidth is set with 2 Mbps, and the *maximum* throughput is about 1.4 Mbps due to the overhead of IEEE 802.11. The source rate of each single flow 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, indicating 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 of the medium, it can transmit about 6.4 packets *consecutively* without collision. 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 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. 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 IEEE 802.11, which is unacceptable for jitter-sensitive traffic. In this section, we explain how hidden-terminal problem causes short-term unfairness.

2.1 Basic Techniques in IEEE 802.11

IEEE 802.11 adopts the well-known binary exponential back-off (BEB) algorithm as its contention resolution mechanism, which is described as follows. Every node maintains a Contention Window (CW) and a back-off timer. Before every transmission, 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.

$$BackoffTime = Random() \times SlotTime \tag{1}$$

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 is set to CW_{min}. Whenever a retransmission is initiated, the CW is doubled until the CW_{max} is reached. After that, the CW remains at the maximum value until the retry limit (say, 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 each retransmission attempt as a *stage*, whose number is in the range $[0, n-1]$.

To combat with the hidden-terminal problem, IEEE 802.11 defines a four-way handshake, where a sequence of Request To Send (RTS), Clear To Send (CTS), Data, and ACK frames, is transmitted for the transmission of every single data packet. For the convenience, we call the exchange of RTS/CTS/Data/ACK frames as a *frame exchange sequence (FES)*. Moreover, *FES (X, Y)* represents a FES between nodes X and Y, *initiated* by node X, implying node X sends one *packet* successfully to node Y. Moreover, we call node X as the *transmitting* node and Y as the *receiving* node, while all the other nodes are called the *waiting* nodes. Note that the word '*packet*' implies the protocol data unit (PDU) of a higher layer whereas '*frame*' is the MAC layer PDU.

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 to count down from its *frozen* back-off timer. It is easy to see that the *transmitting* node may transmit several packets *consecutively* before the *waiting* node's back-off timer is reduced to zero. Contrary to a successful transmission, when a collision occurs, *all* the *colliding* nodes will generate a new random value from their corresponding CWs.

2.2 Explanation for the Short-term Unfairness

In the four-way handshake, once the RTS/CTS has been completed successfully, the hidden-terminal problem may not arise any more. For example, in Figure 1, once node B sends back a CTS to node A, node C overhears 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.

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 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:

$$\mid Z \mid > Len = TxTime(RTS) + SIFS \tag{2}$$

where Z is the difference between the back-off timer. Len is equal to about 19 slots when the slot time is 20 μs for DSSS [6]. It is easy to see that the condition is difficult to satisfy when the CWs at the contending nodes are small (e.g., 31).

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 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 (say, node A) to get control of the medium. Once the FES (A, B) is completed, node A resets its CW to CW_{min} 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, and thus nodes A and B may exchange several more FESs before node C's back-off timer reduces to a small value.

Whenever the back-off timer at node C reduces to a small value, 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 again. 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*. Moreover, this process (i.e., several packet transmissions by node A, followed by collisions, and then 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 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. Once node C controls the medium, it can transmit consecutively in a similar way, and thus the long-term fairness between the two flows is ensured.

3 Analytical Modeling

In this section, we model the hidden-terminal scenario in Figure 1 using an *embedded* Markov chain.

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 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 that the medium is in a T state. Since the transition probabilities among these three states 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, $k, l = 0, ..., (n-1)$. Note that we must use *stages* rather than the values of CW to represent the system state, since once the CW reaches the maximum value, it remains unchanged before resetting. 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, only $(T_C, k, 0)$ system states are possible. 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:

$$N_{state} = n^2 + 2n \tag{3}$$

From a *Col* state, whenever a *transition* occurs, the system can enter anyone of the three kinds of states as shown in the leftmost diagram of Figure 2. If the *next* state is also 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.

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*, 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 violates the *memoryless* requirement of a Markov chain. Therefore, we do *not* treat this event as a state transition. Rather, whenever in state T_A, a state transition occurs *only* when the system enters a *Col* state or a T_C state. 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 applies for the state T_C. The state transitions from a T state are illustrated in the remaining diagrams of Figure 2. It is easy to see that the chain obtained in such a way is an *embedded* Markov chain, modelling the underlying semi-Markov process.

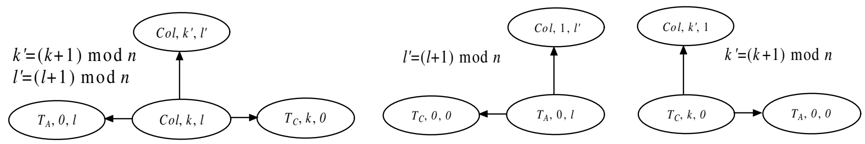

Fig. 2. State Transitions Diagrams

To illustrate the model clearly, we present the complete state transition diagram in Figure 3, where we assume n=3.

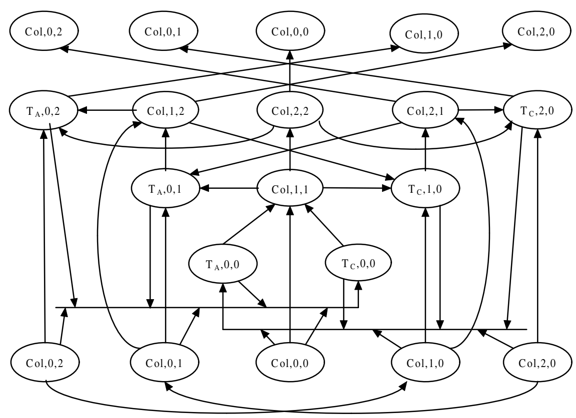

Fig. 3. State Transition Diagram for n=3

3.2 Basic Analysis of the Model

For the convenience, 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:

$$\begin{cases} \pi = \pi \cdot P \;\; \pi = [\pi_0, \pi_1, \pi_2, \ldots], \; P = [p_{ij}] \\ \sum_i \pi_i = 1 \; i \in [0, n^2 + 2n - 1] \end{cases} \tag{4}$$

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, this is not true in our model. For example, the interval between a T state and its next state is a *random* variable. Therefore, π_i can only tell us the probability that the system *enters* state i whenever a *transition* occurs [5]. To get the *time average* state probability 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:

$$\mu_{ji} = \begin{cases} 0 & p_{ji} = 0 \\ 1 & p_{ji} > 0 \& i \in [2n, n^2 + 2n - 1] \\ r \times num(j, i) & p_{ji} > 0 \& i \in [0, 2n - 1] \end{cases} \tag{5}$$

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 present state, is assigned a *unit* value as in the second line of the above formula. To express 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*'. The '*Col time*' is corresponding to the CTSTimeout interval defined in [6], which is independent from the length of the Data frame. Therefore, whenever the system enters 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 (5). From μ_{ji}, which corresponds to a *transition*, we can get μ_i, the expected holding time of state i, as follows:

$$\mu_i = \sum_j \tfrac{\pi_j \times p_{ji}}{\pi_i} \mu_{ji} \tag{6}$$

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 (7), which denotes the average number of packets the *transmitting* node can transmit consecutively once the system reaches state i.

$$num(i) = \sum_j \tfrac{\pi_j \times p_{ji}}{\pi_i} num(j, i) \;\; i \in [0, 2n - 1] \tag{7}$$

As a result, formula (6) can be replaced by:

$$\mu_i = \begin{cases} num(i) \times r & i \in [0, 2n - 1] \\ 1 & i \in [2n, n^2 + 2n - 1] \end{cases} \tag{8}$$

Now we can get ρ_i, which represents the *time average* state probability of state i [5]. Note that here we have ignored the time the meidum being idle.

$$\rho_i = \frac{\pi_i \times \mu_i}{\sum_j \pi_j \times \mu_j}; \; i, j \in [0, n^2 + 2n - 1] \tag{9}$$

3.3 Derivation of the Metrics

We now 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 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 (7). Therefore, the metric-1 can be obtained by taking average of all $num(i)$ corresponding to T_A states:

$$H_{metric-1} = \sum_{i=0}^{n-1} \frac{\pi_i}{\pi(A)} \times num(i); \quad \pi(A) = \sum_{i=0}^{n-1} \pi_i \tag{10}$$

Since the behavior is *identical* at nodes A and C, the metric obtained for node A are 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 C, 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*" [5], 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 [5]:

$$V_i = \begin{cases} 0 & i \in [n, 2n-1] \\ R_i + \sum_j p'_{ij} V_j & i \in [0, n-1] \cup [2n, n^2 + 2n - 1] \end{cases} \tag{11}$$

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 [5]. 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,

$$R_i = \begin{cases} num(i) & i \in [0, n-1] \\ 0 & i \in [n, n^2 + 2n - 1] \end{cases} \tag{12}$$

In equation (11), p'_{ij} is a *modified* value of the 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,

$$p_{ij}^{'} = \begin{cases} 0 & i \in [n, 2n-1] \& j \neq i \\ 1 & i \in [n, 2n-1] \& j = i \\ p_{ij} & i \in [0, n-1] \cup [2n, n^2+2n-1] \end{cases} \quad (13)$$

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 (11).

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

$$H_{metric-2} = \sum_{i=0}^{n-1} \frac{\pi_i}{\pi(A)} \times V_i; \quad \pi(A) = \sum_{i=0}^{n-1} \pi_i \quad (14)$$

To solve the above equations, we *only* need to know the transition probability matrix P and the state holding time $num(j, i)$. The calculation of the above values is not trivial. However, due to space limitation, we do not present it here. Please refer to [8] for a detailed analysis.

4 Numerical Results

Here we evaluate the equations derived in Section 3, and compare the analytical results with the simulation results. The simulation environment is the same as described in Section 2. The results correspond to the case when CW_{min}=31, *Len*=19, CW_{max}=1024, and n=7, which are typically used in IEEE 802.11. Since our main objective is to analyze the behavior at T states, and the behavior at nodes A and C is *identical*, we only present the results for T_A states.

State Probabilities: Table 2 presents 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. The analytical results are quite close to the simulation results. 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, implying 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). This shows the advantage of using the *short* RTS/CTS frames before the transmission of the *long* Data frame.

Table 2. 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 3 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 node C.

Table 3. 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 4 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. 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 the (T_A, 0, 6) state, the system is likely to enter 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 CW mechanism* adopted in IEEE 802.11 improves the short-term fairness.

Table 4. 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 5 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. 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 5. 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, CW_{max}=1024, and n=7. By varying these parameters, 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 and CW_{min}, we can model different physical layers, such as FSSS, DSSS and IR [6]. Moreover, since the model can predict the *short-term* behavior *precisely*, it would also predict the *long-term* behavior accurately.

5 Discussion and related work

5.1 Future work

We have presented a novel embedded-Markov model to study the short-term unfairness in a simple 3-nodes hidden-terminal case. We are extending the model to a more general scenario. However, it is necessary to mention that the modelling 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 that can be detected using dynamic measurements. Our preliminary results show that this method improves the fairness. However, the aggregate throughput may degrade.

5.2 Related work

The fairness problem in wireless networks have been extensively addressed [9–11]. However, most of these work do not consider the hidden-terminal problem *explicitly*. Also, they mainly consider *long-term* unfairness. More importantly, there is no analytical model predicting the unfairness in IEEE 802.11. In contrast, in this paper, we have developed an analytical model to explain and predict the short-term unfairness due to the hidden-terminal problem.

Several *analytical* models of IEEE 802.11 [1–3] have studied the *long-term* behavior (i.e., capacity) by *ignoring* many details of the protocol and adopting *simplified* assumptions. For example, the model in [1] overlooks the *resetting CW mechanism* and assumes a *constant* collision probability, which is clearly imprecise as shown in our results. While these models are able to predict the *long-term* behavior, they cannot be used to study the *short-term* behavior *accurately* because the required details are lost in their models. In contrast, we model the IEEE 802.11 in a *detailed* manner (e.g., by including the resetting mechanism and all the possible collision states) to predict the short-term fairness *precisely*.

The authors of [7] have studied the short-term fairness by first developing two 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 thus [7] cannot apply to IEEE 802.11. Moreover, they have not considered the hidden-terminal problem and they mainly focus on *developing* general fairness metrics, which are different from our work.

6 Conclusions

In this paper, we have presented an Embedded-Markov chain model for IEEE 802.11. Our model is novel in that it predicts the *short-term unfairness* of IEEE 802.11 very precisely, which is not available in the literature. The key concepts used in the model include the 'state holding time' and the 'first passage time'. Our results show that IEEE 802.11 exhibits substantial short-term unfairness in presence of hidden terminals. One important implication of our results is that the resetting of the contention window may improve the short-term fairness.

References

1. G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function," IEEE Journal on Selected Area in Communications, March 2000, pp.535-547.
2. 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.
3. H.S. Chhaya, S. Gupta, "Performance modeling of asynchronous data transfer methods of IEEE 802.11 MAC protocol," ACM Wireless Networks, 3, 1997, pp.217-234.
4. CMU Monarch Group. CMU Monarch Extensions to NS, http://www.monarch.cs.cmu.edu/.
5. R.G. Gallager, "Discrete Stochastic Processes," Kluwer Academic Publishers, Boston,1996.
6. IEEE, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications," IEEE 802.11 standards, June 1999.
7. C.E. Koksal, H. Kassab, H. Balakrishnan, "An Analysis of Short-Term Fairness in Wireless Media Access Protocols," in ACM SIGMETRICS, 2000.
8. Z.F. Li, S. Nandi, A.K. Gupta, "Modeling the Short-term Unfairness of IEEE 802.11 in Presence of Hidden Terminals," full version of the paper, avaliable at http://www.ntu.edu.sg/home5/pg03802331/.
9. H. Lou, S. Lu, V. Bharghavan, "A New Model for Packet Scheduling in Multi-hop Wireless Networks," in ACM MOBICOM, 2000.
10. T. Nandagopal, T. Kim, X. Gao, V. Bharghavan, "Achieving MAC Layer Fairness in Wireless Packet Networks," in ACM MOBICOM, 2000.
11. N.H. Vaidya, P. Bahl, S. Gupta, " Distributed fair scheduling in a wireless LAN," in ACM MOBICOM, 2000.