Simulating CSMA/CA Behavior for Performance Evaluation of Multi-hop Wireless Networks

Akihito Hiromori*, Hirozumi Yamaguchi* and Teruo Higashino*
* Graduate School of Information Science and Technology, Osaka University
1-5 Yamadaoka, Suita, Osaka, Japan 565-0871
{hiromori,h-yamagu,higashino}@ist.osaka-u.ac.jp

***Abstract*—In this paper, we design an efficient method of simulating wireless networks that use CSMA/CA-based protocols in the MAC layer. In the method, a stochastic model to estimate the CSMA/CA frame transmission delay is naturally incorporated into the conventional fully event-based model. The stochastic model can simplify the interactions between a frame transmitter and its surrounding nodes, which alleviates the event scheduling overhead in simulation. The important feature is that the stochastic model can be applied in "per-node" and "time" basis, *i.e.* we may simulate the behavior of some intended nodes precisely while the others are simplified by the stochastic mode to save computational resources. To the best of our knowledge, this is the first approach to coexistence of the stochastic and event-based models in wireless multi-hop network simulation. We have implemented this scheme in a commercial network simulator and conducted several experiments. From the results, it is confirmed that the proposed method could perform simulation of frame transmission much faster than the fully event-based simulation achieving the same accuracy as the conventional model.**

***Index Terms*—Wireless Network Simulation, CSMA/CA**

I. Introduction

Wireless networks including most-popular IEEE802.11-based networks are deployed everywhere nowadays. In order to assess the behavior of those networks and evaluate their performance, field experiments, mathematical analysis and network simulations are usually conducted depending on the network size and evaluation purposes. Network simulators are often required to be efficient in terms of memory space and processing power because they should usually deal with large-scale networks such as vehicular ad-hoc networks and wireless sensor networks, or they should be run a number of times for exhaustive tests of applications. In particular, it is known that event scheduling in the MAC-layer frame transmission consumes a lot of computational resources in wireless network simulations [1], [2]. Accordingly, several techniques have been proposed to deal with this issue [3], [4], [5], [6].

In this paper, we design an efficient method of simulating wireless networks that use CSMA/CA-based protocols in the MAC layer. In the method, a stochastic model to estimate the CSMA/CA frame transmission delay is naturally incorporated into the conventional fully event-based model. The idea is that the channel acquisition before a frame transmission sequence in CSMA/CA, which needs scheduling/rescheduling many events to many nodes, is modeled as a probabilistic duration that can be computed simply. Since this abstraction is done within the event-driven simulation scheme, the stochastic model can adaptively be applied in "per-node" and "time" basis. In other words, the abstract level of simulation can be adapted spatially and temporally, depending on upper-layer applications such as routing and video streaming. We have implemented this scheme in a commercial network simulator called Scenargie [7] and conducted several experiments. From the results, the proposed method could perform simulation of frame transmission faster than the "pure" event-based simulation.

978-1-4673-1298-1/12/$31.00 © 2012 IEEE

II. Fully Event-based Modeling of CSMA/CA

The significant feature of IEEE 802.11 DCF is the back-off mechanism to avoid collision. When a node wishes to send a MAC frame, the node has to listen to the channel during a random back-off time before sending the frame. The back-off is suspended during its neighboring nodes' transmission.

The fully event-based model simulates these operations by scheduling three events, "back-off timer", "send-begin" and "send-end". Let us assume that node A and node B try to send frames to node C in Fig. 1. These nodes schedule "back-off timer" events with random delay. In this figure, node B can send the frame earlier than node A because node A has a longer back-off time value and the "back-off timer" event of node B is executed first. Node B schedules "send-begin" events and "send-end" events to its neighbor nodes A and C immediately after its "back-off timer" event, in order to notify its transmission. Meanwhile, node A re-schedules its "back-off timer" event to wait for the channel to be idle as the node receives "send-begin" and "send-end" events.

The fully event-based model can also represent frame reception by those "send-begin" and "send-end" events. Fig. 2 shows an example that node C receives frames from node A and node B. Let us assume the following situation. Firstly, nodes A and B cannot listen to each other's transmission (*i.e.* they are hidden terminals of each other). Node A schedules "send-begin" and "send-end" events to its neighbor nodes. Node C knows that node A sends a frame to node C by these events, examines whether the radio signal strength is sufficient to receive the frame or not, and finally determines that the frame cannot be received due to weak signal strength (this information is obtained from the physical layer simulation). Then node B begins to send a frame and node C cannot receive it due to interference at node C by the signal from node A (hidden terminal interference). As such, if node C receives the "send-begin" event, it needs to check whether or not there are other "send-begin" events scheduled before the "send-end" event to know the possibility of data frame collision.

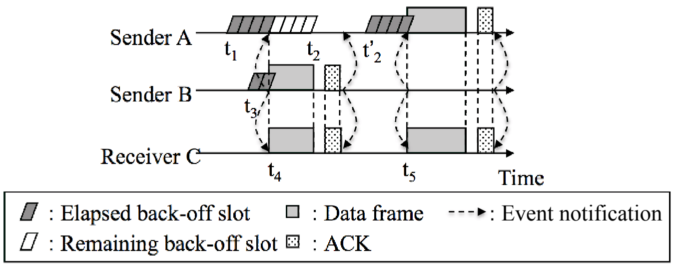

Fig. 1. Sending a frame

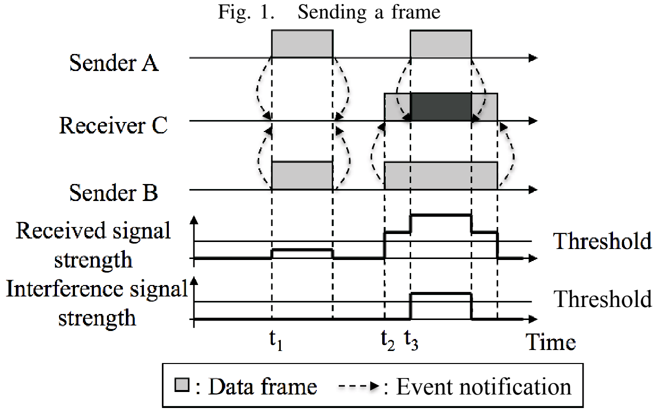

Fig. 2. Receiving a frame

In this way, the fully event-based model can exactly simulate the complete interactions of CSMA/CA, but it is necessary to schedule/reschedule a lot of events. Therefore, large-scale network simulation requires a lot of computing resources.

III. Stochastic event-driven model of CSMA/CA

In our proposed model, each node that wishes to transmit a frame never utilizes the original back-off process. Instead, it computes the estimated delay to acquire the channel, which is called *channel acquisition delay* hereafter. This computation is done online knowing the surroundings' recent traffic status, *i.e.*, the traffic amount transmitted by its neighbors as well as its own traffic amount to be transmitted. Based on the information, the probability function is designed for this estimation purpose. Following the probability, the channel acquisition process of CSMA/CA is simplified keeping very similar packet loss ratio with the original CSMA/CA process. In this way, the proposed model can simulate each frame transmission independently without confirming other frame transmissions. Therefore, when a node sends a frame, the proposed model schedules "send-begin" event and "send-end" event for only its receiver node to which the frame is delivered. It is not necessary to schedule any events for other neighbor nodes. This can contribute to the reduction of simulation costs. Although this basic idea is based on a technique to analyze IEEE 802.11 DCF [8], the other parts of the original mechanism such as the retransmission mechanism are kept intact for accuracy of simulation. As stated above, our goal is to reduce the number of events to be scheduled and/or rescheduled due to back-off interruption.

Basically, this model is based on the stochastic model proposed in Ref. [8] and modified to derive channel acquisition delay for the event-based simulation. In the method, we use two probabilities $S'_n(t')$ and L_n to model the operation of IEEE 801.11 DCF. We let N_n denote a set of node n and its neighbors. $S'_n(t')$ is a probability that at least one node of N_n sends frames in idle time slot t'. L_n is a probability that a frame received by node n collides. The symbols S and L respectively come from "Sending of frames" and "Loss of frames", and symbols with prime mean that those symbols are defined with respect to idle time slots. We design $S'_n(t')$ and L_n in the following.

A. Computation of Probabilities in Stochastic Event-driven Model

We denote as $h_n(t)$ the frame arrival ratio from the network layer in time slot t of node n. This is given as an input and we assume it follows $\sum_{t=0}^{\infty} h_n(t) = 1$. Let us denote an average transmission time as T_n, which is duration from transmission of RTS till reception of ACK or ACK timeout. We also denote the probability that at least one node of N_n sends a frame in time slot t as $S_n(t)$. By using these variables, we can derive the probability denoted by $g_n(t)$ that node n begins to process back-off at time slot t as follows.

$$g_n(t) = \left(\prod_{x=t-T_n+1}^{t} (1 - S_n(x)) \right) \cdot h_n(t) + S_n(t - T_n) \cdot \sum_{x=t-T_n}^{t-1} h_n(x) \tag{1}$$

The first part of this equation means the probability that node n can begin its back-off process immediately after the node receives a frame from the network layer.The node can enter the back-off status without waiting because no neighbor node sends a frame in time slot t and a wireless channel is idle. Meanwhile, if one of the neighbor nodes send a frame, the node n cannot begin its back-off process immediately, and has to delay the transmission.The second part of this equation means that frames from the network layer between $t - T_n$ and $t - 1$ time slots are scheduled to process at time slot t because one of neighbor nodes has sent a frame at time slot $t - T_n$. We note that $\sum h_n(t)$ indicates the probability of receiving at least one frame from the network layer before t.

A frame collision may occur when two or more nodes send frames at the same time slot. In order to estimate how often the frame collision occurs, we focus on idle time slots that are not occupied by frame transmission. Let us recall that a "prime symbol" represents probabilities that only focus on idle time slots (*i.e.* busy slots are ignored). Let us assume k frames have been sent by time slot t. This means that $k \cdot T_n$ time slots have been occupied and the last idle time slot t' is $t - k \cdot T_n$. Therefore, we can derive $g'_n(t')$ from $g_n(t)$ and $C_n(t, k)$ by defining $t' \triangleq t - k \cdot T_n$ where $g'_n(t')$ is the probability that exactly k frames have been sent by time slot t among N_n.

$$g'_n(t') = \sum_{k=0}^{\infty} C_n(t' + k \cdot T_n, k) \cdot g_n(t' + k \cdot T_n) \tag{2}$$

$$C_n(t, k) = C_n(t-1, k-1) \cdot S_n(t) + C_n(t-1, k) \cdot (1 - S_n(t)) \tag{3}$$

$$C_n(0, k) = \begin{cases} 1 - S_n(0) & (k = 0) \\ S_n(0) & (k = 1) \\ 0 & (k \geq 2) \end{cases} \tag{4}$$

$C_n(t, k)$ consists of two parts: The first part represents that $k - 1$ frames have been sent by time slot $t - 1$ and a frame is sent at time slot t. The second part represents that k frames have been sent by time slot $t - 1$ and no frame is sent at time slot t.

Next, we will show how to calculate the probability denoted by $f'_n(t')$ that node n sends a frame at idle time slot t'. If node n sends a frame at idle time slot t', the node had begun a back-off process at certain idle time slot x and had waited for $t'-x$ idle time slots. Thus, $f'_n(t')$ can be represented by two probabilities, $g'_n(t')$ and $B(t'-x, pc_n)$. The latter is the probability that node n sends a frame after $t'-x$ idle time slots, given the probability pc_n of frame loss of node n.

$$f'_n(t') = \sum_{x=0}^{t'} g'_n(x) \cdot B(t'-x, pc_n) \tag{5}$$

$$B(t'-x, pc_n) = \sum_{i=0}^{r} (pc_n)^i \cdot B_i(t'-x) \tag{6}$$

$B(x, pc_n)$ represents the back-off process itself and is affected by frame retransmissions. It can be the sum of probability $B_i(x)$ that a node ends a back-off process of the i-th retransmission after x idle time slots. $B_i(x)$ means that a node failed the $(i-1)$-th back-off process at certain time slot y and the node begins the i-th back-off process with duration $x-y$.

Using $f'_n(t')$, we can derive the probability $S'_n(t')$ that at least one node in N_n sends a frame at idle time slot t'. Although this probability is used to simulate frame transmissions in the proposed model, we need frame loss probability pc_n to calculate it and continue to process this formulation. We can also derive $S_n(t)$ using $S'_n(t')$ and $C'_n(t', k)$. $C'_n(t', k)$ is the probability that exactly k frames have been sent by idle time slot t'.

$$S'_n(t') = 1 - \prod_{n \in N_n} (1 - f'_n(t')) \tag{7}$$

$$S_n(t) = \sum_{k=0}^{\infty} C'_n(t - kT_n, k) \cdot S'_n(t - kT_n) \tag{8}$$

$C'_n(t', k)$ is derived from $S'_n(t')$ in the same manner as $C_n(t, k)$. If k frames have been sent by idle time slot t', t' is time slot $t + k \cdot T_n$. Thus, $S'_n(t')$ is equivalent to $S'_n(t' + k \cdot T_n)$ and we can derive (8) by defining $t \triangleq t' + k \cdot T_n$.

Finally, we can derive the collision ratio in frame transmission. This is used to determine whether or not a node can receive a frame. We denote this probability as L_n.

$$L'_n(t') = \frac{\sum_{i \in N_n} f'_i(t') \left\{ 1 - \prod_{j \neq i, j \in N_n} (1 - f'_j(t')) \right\}}{\sum_{i \in N_n} f'_i(t')} \tag{9}$$

$$L_n = \lim_{t' \to \infty} \frac{\sum_{x=0}^{t'} L'_n(x)}{t'} \tag{10}$$

$$pc_n = \frac{\sum_{i \neq n, i \in N_n} L_i}{|N_n| - 1} \tag{11}$$

$L'_n(t')$ is the average length of transmission started at time slot t'. $L'_n(t')$ can be represented by their $f'_i(t')$ since nodes in N_n might send frames at the same time. In addition, using L_n, we obtain the probability pc_n that a node fails to send a frame.

As shown in Eq.7, Eq.10 and Eq.11, in order to calculate $S'_n(t')$, L_n and pc_n for node n, it is necessary to refer to L_i and $f'_i(t')$ where note i is each neighbor node. Therefore, we must calculate the probabilities of all nodes at the same time so that we can simulate how the frame transmission of the nodes affects each other. The proposed method initially sets $S_n(t) = 0$, and calculates the expressions repeatedly until L_n converges. We can get $S'_n(t')$ and L_n of all nodes after that calculation.

B. Modeling Frame Transmission

1) Sending a Frame: If node n in the proposed model begins the back-off process at idle time t', the node first selects its back-off time randomly from its contention window size in the same way as the fully event-driven model. We denote this back-off time as b. Then using the expected number of frames sent, the total waiting time around node n, which is denoted as $B_n(t', b)$, is expressed as follows.

$$B_n(t', b) = b + T_n \cdot (M'_n(t' + b) - M'_n(t')) \tag{12}$$

$$M'_n(t') = \sum_{x=0}^{t'} S'_n(x) \tag{13}$$

$M'(t')$ is the expected number of frames sent before idle time slot t', and consequently $M'(t'+b) - M'(t')$ means the number of frames sent during the back-off process. After the total waiting time $B_n(t', b)$ elapses, the node sends "send-begin" event and "send-end" events to the receiver node.

2) Receiving a Frame: In the proposed model, a node determines whether a frame can be received or not based on the probability calculated by Eq.(10). Concretely, when a node executes a "send-begin" event in order to receive a frame, the proposed model deferimine whether the node succeed to receive the frame or not. If so, the node continues to process the event and can receive the frame. If not, the node cancels the event.

3) Proposed and Fully Event-driven Models: By the proposed model, we can reduce the computational cost for simulations, but the simulation accuracy may not be perfect. To pursue the best trade-off between the computational cost and accuracy, we may apply the fully event-driven model to the set of nodes that need accurate simulations, and the proposed model to the rest of the nodes to reduce the number of events for speedup of simulations. In such case, the two models need several interactions to work properly. Basically, the nodes simulated by the fully event-driven model not only conduct conventional event-driven simulations but also calculate the above two probabilities like the nodes simulated by the proposed model. We will omit the details due to lack of space.

IV. Experiments

We have implemented the proposed mechanism in a commercial network simulator called Scenargie [7], and have conducted several experiments. In these experiments, we have measured both of the frame transmission delay (*i.e.* channel acquisition delay) and frame loss ratios in the MAC layer to evaluate the accuracy of simulations. We have also measured the number of scheduled "send-begin" events to observe the saving effect by our method.

Firstly, we have prepared a scenario where nodes can listen to each other (all the nodes are in the single hop range) and there are exactly one transmitter and one receiver. The transmitter sends a MAC frame in IEEE 802.11b to the receiver every 0.5 second. The frame size is 512 bytes. This scenario is designed to observe how a single frame transmission is scheduled in both models. Varying the number of neighbor

TABLE I
THE NUMBER OF SIMULATION EVENTS (WITH ONE TRANSMITTER)

# of nodes	# of events (Fully event-driven)		# of events (Prpposed)	
	send-begin	all	send-begin	all
10	47230	141696	4720	29500
20	94400	271399	4815	30165
30	141600	401199	4872	30564
40	188800	530998	4964	31208
50	236150	661255	5002	31474

TABLE II
THE NUMBER OF SIMULATION EVENTS (WITH $n-1$ TRANSMITTER)

# of nodes	# of events (Fully event-driven)		# of events (Prpposed)	
	send-begin	all	send-begin	all
10	472170	1423622	47200	295001
20	1919380	5527932	96049	601545
30	4384830	12402331	146278	917749
40	7919720	22148162	198336	1246756
50	12563150	34848197	251693	1584861

nodes from 10 to 50, we have measured the number of "send-begin" events. Table I shows the number of "send-begin" and the number of all simulation events. The number of events in the traditional fully event-based model is proportional to the number of neighbor nodes, because the events were scheduled not only to the receiver node but also to the neighbor nodes in the fully event-based model. On the other hand, since the events were scheduled only to the receiver node in the proposed model, the number of the events in the proposed model was constant. We have also measured the frame loss ratios and frame transmission delay, and they were almost zero in both models. From these results, we can see that the proposed model can significantly reduce the number of the events, achieving the same simulation results.

Next, we have evaluated the affect of node density and behavior on the simulation performance and accuracy. We modified the previous scenario to let EVERY node except the receiver node transmit a MAC frame in IEEE 802.11b every 0.5 second and evaluated the same metrics. Table II shows the number of "send-begin" events and the number of all simulation events. From the result, we can see that the proposed model can reduce considerably the number of events especially when nodes are densely deployed. Fig. 3 shows the frame loss ratios in the MAC layer. In the proposed model, the frame loss ratio is directly determined by some probabilities, while it is adhere to the original CSMA/CA mechanism taking into consideration the physical layer effect in the fully event-based model. In Fig. 3, we can observe that the calculated loss ratios are very close to those in the fully event-based model. Although a small gap between two lines has been observed, we have analyzed that this was due to probabilistic and unexpected behavior of the physical layer, which is almost impossible to perfectly follow for any abstraction technique that does not simulate radio propagation. The important feature here is that the two lines follow the same trend with at most 1% deviation. The same tread has been observed in frame transmission delay shown in Fig. 4. The maximum deviation was about less than 1ms, which is negligible for many large-scale networks such as WSNs.

V. CONCLUSION

In this paper, we have proposed a new stochastic model for simulating the MAC layer behavior. The proposed model determines whether a node can transmit MAC frames or not using probabilities to reduce the number of simulation events. We have implemented the proposed model in a commercial network simulator, and carried out several experiments. From the experimental results, the proposed model could reduce the number of events maintaining certain accuracy.

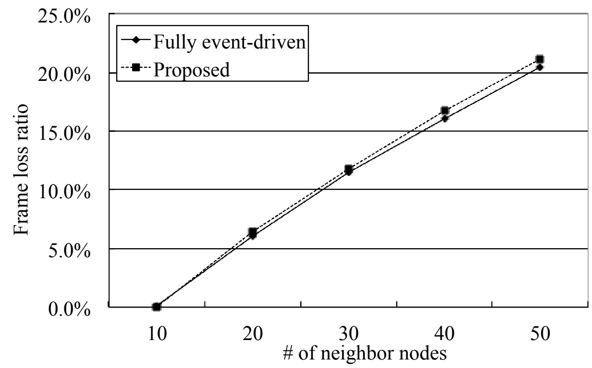

Fig. 3. Frame loss ratio ($n-1$ transmitters)

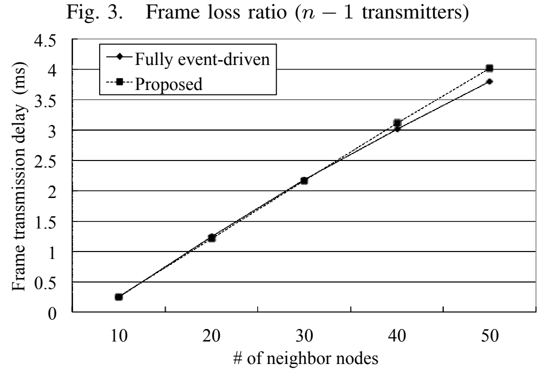

Fig. 4. Frame transmission delay ($n-1$ transmitters)

REFERENCES

[1] P. Gopalakrishnan, S. Demers, and L. Kant, "Speedup techniques for analyzing scalability of large scale mobile ad hoc network management systems," in *Proceedings of the 2006 international conference on Wireless communications and mobile computing (IWCMC 2006)*, 2006, pp. 299–304.

[2] E. Egea-Lopez, J. Vales-Alonso, A. Martinez-Sala, P. Pavon-Marino, and J. Garcia-Haro, "Simulation scalability issues in wireless sensor networks," *IEEE Communications Magazine*, vol. 44, no. 7, p. 64, 2006.

[3] F. Roijers, H. van den Berg, and M. Mandjes, "Fluid-flow modeling of a relay node in an IEEE 802.11 wireless ad-hoc network," *Lecture Notes in Computer Science*, vol. 4516, p. 321, 2007.

[4] J. Liu, D. M. Nicol, L. F. Perrone, and M. Liljenstam, "Towards high performance modeling of the 802.11 wireless protocol," in *Proceedings of Winter Simulation Conference 2001*, 2001, pp. 1315–1320.

[5] M. Killat, F. Schmidt-Eisenlohr, H. Hartenstein, C. Rössel, P. Vortisch, S. Assenmacher, and F. Busch, "Enabling efficient and accurate large-scale simulations of vanets for vehicular traffic management," in *Proceedings of the 4th ACM international workshop on Vehicular ad hoc networks (VANET 2007)*, 2007, pp. 29–38.

[6] H. Kim and J. C. Hou, "Mixed-mode simulation for ieee 802.11-operated wlans," *Computer Networks: The International Journal of Computer and Telecommunications Networking*, vol. 51, no. 6, pp. 1379–1402, 2007.

[7] Space-Time Engineering, LLC, "Scenargie," http://www.spacetime-eng.com/.

[8] G. Bianchi, "Performance analysis of the ieee 802.11 distributed coordination function," vol. 18, no. 3, mar 2000, pp. 535 –547.