

# Scalable Simulation of Large-Scale Wireless Networks with Bounded Inaccuracies <sup>\*</sup>

Zhengrong Ji, Junlan Zhou, Mineo Takai, and Rajive Bagrodia  
Computer Science Department  
University of California Los Angeles  
Los Angeles, CA 90095  
{jizr, zjl, mineo, rajive}@cs.ucla.edu

## ABSTRACT

Discrete event network simulators have emerged as popular tools for verification and performance evaluation for various wireless networks. Nevertheless, the desire to model such networks at high fidelity implies high computational costs, prohibiting most researchers from simulating wireless networks with thousands of nodes. There have been attempts on performance optimizations for large-scale wireless network simulation, but they have not appropriately modeled accumulation of weak interference, thereby suffering inaccuracies which may be magnified by upper layer protocols. This paper presents analysis of the effects of common optimization techniques for large-scale wireless network simulation on the overall network performance and also proposes modifications and novel techniques that introduce only limited inaccuracies or no additional inaccuracy at all. The study quantifies the effects of those optimizations on the simulation results for given thresholds and network parameters, and also identifies thresholds tolerable to most network studies. The experimental results show that these optimizations can improve the runtime performance of an already efficient wireless network simulator substantially, by a factor of up to 55 for wireless networks with 3200 nodes without compromising accuracy of the simulation results.

## Categories and Subject Descriptors

I.6.5 [Simulation and Modeling]: Model Development;  
I.6.4 [Simulation and Modeling]: Model Validation and Analysis

## General Terms

Performance, Algorithms

## Keywords

Discrete Event Simulation, Wireless Network, Scalability

<sup>\*</sup>This work is funded in part by NSF under NRT grant ANI-0335302, and DARPA NMS program under contract N66001-00-1-8937.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

MSWiM'04, October 4-6, 2004, Venezia, Italy.

Copyright 2004 ACM 1-58113-953-5/04/0010 ...\$5.00.

## 1. INTRODUCTION

As both size and number of wireless networks rapidly increase, there is an accompanying need for performance evaluation and analysis of large-scale wireless networks. Unlike wired networks where the physical and MAC (Media Access Control) layers are relatively reliable and can be abstracted out, wireless networks rely on unguided media shared by many nodes in the network, introducing packet loss due to collision and interference. In particular, interference is the biggest factor that distinguishes wireless media from wired, and is also a growing concern due to the rapid increase in the number of devices operating at similar frequencies in close proximity. This issue makes it hard to adopt analytical models for wired networks to characterize the performance of wireless networks.

The most commonly used technique to model and analyze wireless networks is discrete event simulation [9][4][12], as it provides great flexibility in modeling collision, interference and other environmental effects. However, it is computationally expensive, particularly for modeling large-scale wireless networks. One common practice to alleviate this problem is to limit simulation of signal propagation to a certain range such that the simulator does not need to schedule as many events per signal transmission [14][8]. However, even very weak signals can accumulate and become strong enough to cause packet errors or change sensing status of network nodes. In fact, a comparative study of two commonly used wireless network simulators revealed that not simulating the cumulative nature of interference for even small networks could significantly alter the projected performance of higher layer protocols [15]. Therefore, this paper attempts to improve scalability of a wireless network simulator without compromising its accuracy.

The following contributions are made in this paper: First, we analyze the effects of a common optimization technique, limiting propagation for large-scale wireless network simulation, upon the accuracy of the simulation results at upper layers. Using knowledge of channel and physical characteristics, we derive a better distance limit other than Carrier Sensing Threshold (CST) used by previous studies. We demonstrate that our approach preserves accuracy of the simulation (in terms of upper layer traffic statistics) while maintaining good speedup. Further, we propose a novel technique of events aggregation for simulation of physical radio device, which does not cause inaccuracy at all. The experimental results show that the runtime performance of an already efficient wireless network simulator has been improved by our optimization techniques significantly, in particular, by a factor of up to 55 for wireless networks with

**Table 1: Common experiment parameters**

<b>Propagation Model</b>	TWO-RAY	<b>Physical Data Rate</b>	2Mbps
<b>Channel Frequency</b>	2.4GHz	<b>Antenna Height</b>	1.5m
<b>Physical Model</b>	802.11b DSSS	<b>Transmission Power</b>	15dBm
<b>Modulation Scheme</b>	DPSK	<b>Receiving Threshold</b>	-81dBm
<b>MAC Model</b>	802.11 DCF	<b>Receiver Sensitivity</b>	-91dBm

3200 nodes without compromising accuracy of simulation results.

This paper is organized as follows. The next section gives a brief overview of past studies on simulation of large-scale wireless networks, and demonstrates the effects of limiting signal propagation on the predicted overall network performance. Section 3 presents mathematical analysis characterizing wireless propagation models, to quantify the effects of common optimizations for given thresholds and network parameters. Following the analysis, Section 4 and 5 present runtime performance optimization techniques applicable to signal propagation and wireless device models respectively, and show simulation runtime speed improvements by individual techniques. The last section concludes the paper.

## 2. BACKGROUND

In common discrete event network simulators, a propagation limit is often introduced to reduce the number of events required to simulate interference which results in reduced computational complexity [14][8]. However, there has been no attempt to quantify potential inaccuracies introduced by the propagation limit or to derive a propagation limit with bounded inaccuracies. Some past work including *ns-2*, a widely used simulator, utilizes CST (Carrier Sensing Threshold) as the propagation limit [9][8]. That is, signals with reception power below CST will not be delivered to a node. To observe its impact on accuracy of simulation results, we have conducted experiments for a network of 100 nodes with parameters shown in Table 1. All nodes are uniformly distributed in a  $2000 \times 2000 m^2$  terrain with no mobility. Thirty randomly chosen pairs of nodes are each assigned one CBR (Constant Bit Rate) session at a rate ranging from two to ten 512-byte packets/sec. AODV [11] is used as the routing protocol. To minimize the difference in simulation results caused by random number sequences, random numbers are only used to determine backoff timers in 802.11 devices. The experiments are conducted using QualNet [12] and results with/without the CST propagation limit are shown in Figure 1.

Figure 1(a) shows that using CST as the propagation limit changes the PDR (Packet Delivery Ratio) significantly except in one case with extremely low traffic. It is interesting to note that the CST propagation limit yields a PDR lower than the PDR with no propagation limit, for cases with low to moderate traffic loads. On the other hand, it yields higher PDRs when traffic load further increases. This phenomenon is due to the fact that the effects of the CST propagation limit are two-fold. First, nodes sense the channel to be idle more often than they should. Hence, they are more aggressive in transmission, resulting in more packet collisions. In addition, since more nodes cannot sense each other's transmissions when the propagation limit equals CST, the hidden terminal problem becomes more significant with more RTS packets colliding, which triggers more retransmissions (see Figure 1(b)). Second, many interference signals are ignored, resulting in better (but incorrect) SINR. The former deteriorates

PDR while the latter exaggerates it. These two effects always compete, with one or the other dominantly deviating PDR from the actual value regarding specific scenarios.

In addition to setting the propagation limit to CST, past studies based on *ns-2* do not model the accumulation of interference signals. It has been demonstrated that this could impair accuracy of simulation results significantly [15]. The high computational costs of wireless network simulation have also motivated efforts [1][10] to parallelize simulation of wireless networks. By distributing network models on multiple processors, parallel simulation improves scalability substantially compared to sequential simulation without changing simulation results. Nevertheless, the speedup factor is limited to the number of processors used.

To our best knowledge, this paper is the first attempt to utilize the characteristics of wireless propagation model to analyze the effects of common optimization techniques such as the propagation limit. In addition, novel optimization method is proposed and the impacts on simulation results are quantified. To facilitate this study, QualNet [12] was used to implement the proposed optimization techniques, as it is one of the efficient wireless network simulators with detailed physical layer models.

## 3. ANALYSIS

### 3.1 Wireless Propagation Model

Most commonly used wireless network devices use CSMA (Carrier Sensing Multiple Access) based physical and MAC sub-layers, such as those defined in the IEEE 802.11 standard [6]. The underlying radio channel condition places fundamental limitations on the performance of wireless network devices. Many wireless network simulators use propagation models such as Free Space [3], Two Ray [2] or Hata model [5] to simulate signal propagation. Assuming other parameters (such as antenna height) are constant, most of the analytical and empirical propagation models express path loss as a function of distance between transmitter and receiver as

$$PL(d) [dB] = \overline{PL}(d_0) [dB] + 10\alpha \log\left(\frac{d}{d_0}\right) + X_\sigma \quad (1)$$

where  $d$  is the distance between transmitter  $t$  and receiver  $r$ ;  $d_0$  is the close-in reference distance determined from measurements close to the transmitter;  $\alpha$  is the path loss exponent;  $X_\sigma$  is a zero-mean Gaussian random variable (in dB) modeling log-normal shadowing effects. For simplicity of the analysis, expected path loss  $\overline{PL}$  is used in this context, as it is often the major component of overall attenuation experienced by a signal [13].

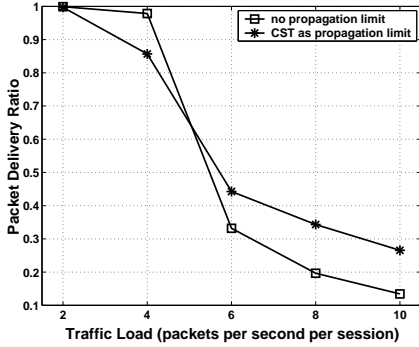
Suppose that each signal has zero-mean amplitude and is independent from others. The aggregated power of all received signals at node  $r$  is given by

$$P_r = \sum_{t: t \neq r} P_t \times \overline{PL}(d_{tr})^{-1} \quad (2)$$

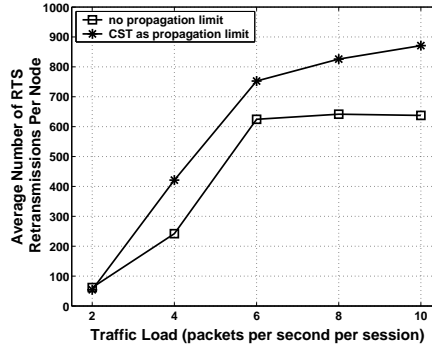
where  $P_t$  is the transmitted power of a signal originating from node  $t$ , and  $d_{tr}$  is the distance between  $t$  and  $r$ , assuming antenna gains are already included in path loss calculation. Thus, SINR  $\Gamma_r$  of a signal received by  $r$  is

$$\Gamma_r = \frac{P_t \times \overline{PL}(d_{tr})^{-1}}{\sum_{s: s \neq t \wedge s \neq r} P_s \times \overline{PL}(d_{sr})^{-1} + \sigma_n^2} \quad (3)$$

where  $P_t$  is the transmitted power of the desired signal from node  $t$ ;  $P_s$  stands for the transmitted power of an interference signal from node  $s$ ;  $\sigma_n^2$  is expected ambient noise power.



(a) Packet delivery ratio



(b) Number of RTS retransmissions

Figure 1: Impacts of using CST as propagation limit

### 3.2 Characterization of the Propagation Model

Consider a wireless network in which radio signals transmitted by nodes propagate in accordance with the above mathematical model. By investigating the constraints placed by the propagation model on network performance, we will derive some properties that shed light on mechanisms to improve scalability of wireless network simulation. A few definitions are presented first to facilitate the analysis.

#### 3.2.1 Definition

*Effective node density*  $\lambda$  is the maximum density of nodes at which concurrent transmission of all neighbors will not cause the node to sense that the channel is busy.

*Minimum sensing distance*  $S$  is the distance between a transmitter and a receiver at which the received power of the signal equals to CST, when the signal is transmitted at minimum transmission power.

*Maximum transmission power*  $P_{\max}$  is the maximum power a node can use for transmission.

#### 3.2.2 Model Characteristics

**PROPOSITION 1.** *If all nodes in a densely connected wireless network operate asynchronously in CSMA and propagation delay among any pair of nodes in sensing distance is negligible, then  $\lambda \leq 2/(\sqrt{3}S^2)$ .*<sup>1</sup>

**PROPOSITION 2.** *Suppose all nodes operate in CSMA mode and are distributed in a plain disk of radius  $\xi$ , regardless of the node density. If path loss is a power law function of distance with exponent  $\alpha > 1$ , there exists an upper bound to aggregate power ( $\widehat{P}$ ) of signals received at the center of the disk plane from all nodes outside distance  $D$ :  $d_0 \leq D \leq \xi$ .*

**PROOF.** According to Proposition 1, there exists an upper bound to the effective node density  $\lambda$ . Thus the aggregate power of signals received at the disk center from all nodes outside distance  $D$  can be expressed as

$$\widehat{P} = \sum_{s: d_s > D} \frac{P_s}{PL(d_s)} \leq \frac{P_{\max} d_0^\alpha}{PL(d_0)} \sum_{s: d_s > D} \frac{1}{d_s^\alpha} \quad (4)$$

Approximating (4) with a continuous function gives

$$\widehat{P} \leq \begin{cases} \frac{2\pi\lambda P_{\max} d_0^\alpha}{PL(d_0)} (\ln \xi - \ln D) & \alpha = 2 \\ \frac{2\pi\lambda P_{\max} d_0^\alpha}{(2-\alpha)PL(d_0)} (\xi^{2-\alpha} - D^{2-\alpha}) & \alpha \neq 2 \end{cases} \quad (5)$$

The upper bound for aggregate power received at the center of plain disk is given in (5).  $\square$

<sup>1</sup>Due to space limitation, proofs will only be provided upon request.

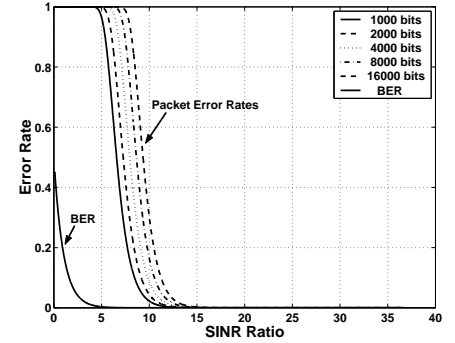


Figure 2: BER and PER vs. SINR plot for DPSK modulation

**COROLLARY 1.** *Suppose  $\xi \rightarrow +\infty$  regardless of the node density. If path loss is a power law function of distance with exponent  $\alpha > 2$ , there exists an upper bound to aggregate power of signals received at the center of the disk plane from all nodes outside distance  $D$ :  $D > d_0$ .*

We argued at Section 2 that if errors introduced at the physical layer are not bounded properly (such as using CST as the propagation limit), the inaccuracy to simulation results can be amplified drastically at upper layers. Proposition 2 and its corollary establish the relationship between the propagation(distance) limit and the upper bound on

$\widehat{P}$ , which makes it possible to measure the inaccuracy ( $\widehat{P}$ ) caused by a distance limit, or the reverse. If the association

between behavioral sensitivity of upper layer protocol and  $\widehat{P}$  is known, derivation of the appropriate distance limit would be possible<sup>2</sup>.

## 4. SIMULATION OF SIGNAL PROPAGATION

### 4.1 Derivation of Distance Limit

Existing discrete event network simulators [9][4][12] typically suffer from a common overhead: whenever a node transmits a packet, the impact of a transmitted signal over all other nodes must be simulated by generating at least two events for each of the nodes, a signal arrival event and a signal end event. If  $N$  is the total number of nodes in the wireless network, each transmission produces  $\Theta(N)$  events, which increases event-scheduling complexity. As indicated by Proposition 2 and its corollary, there exists a distance limit to a receiving node such that the aggregated power of all concurrently received signals originating from nodes outside the distance limit is below a threshold. If this threshold is low enough such that it is insignificant for a receiver to change sensing status and SINR, then the corresponding distance limit can be used to filter out signals originating from distant nodes. Note that the exact threshold value depends on many parameters, such as antenna height, radio sensing sensitivity and the relationship between SINR and BER (Bit Error Rate). In this study, the threshold value is set empirically. The following discussion illustrates how a distance limit is derived, assuming that the path loss exhibits a power law function with  $\alpha \neq 2$  and the aggregated signal

<sup>2</sup>To study the sensitivity of upper layer behavior to the inaccuracy of PHY/MAC layer modeling is important future work.

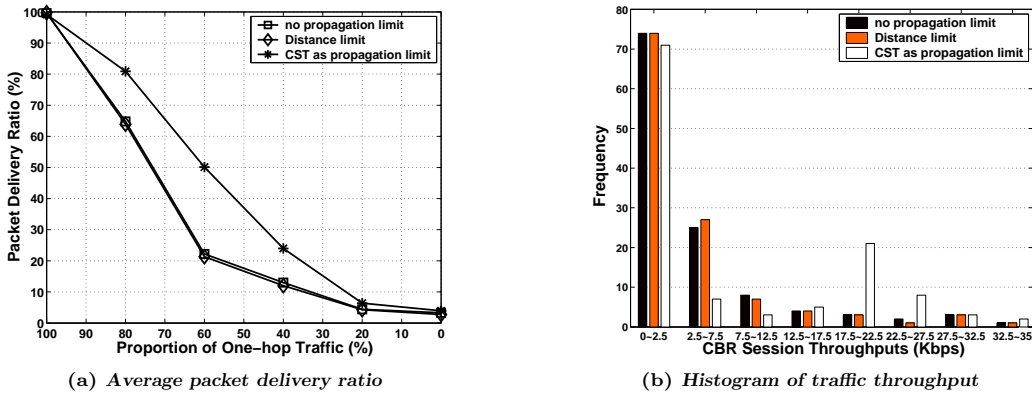


Figure 3: Validation of our distance limit with network of 400 nodes

power is below expected noise power (so as to limit inaccuracies introduced by carrier sensing and SINR calculation). Substitution into (5) gives

$$\frac{2\pi\lambda P_{\max} d_0^\alpha}{(2-\alpha)\overline{PL}(d_0)} (\xi^{2-\alpha} - D^{2-\alpha}) \leq \sigma_n^2 \quad (6)$$

Hence the corresponding distance limit is found to satisfy

$$D \geq \left( \xi^{2-\alpha} - \frac{(2-\alpha)\sigma_n^2\overline{PL}(d_0)}{2\pi\lambda P_{\max} d_0^\alpha} \right)^{\frac{1}{2-\alpha}} \quad (7)$$

The distance limit varies depending on the exact value of the right hand side of (6). Generally the value needs to be small relative to the receiver radio sensitivity, so that the aggregate strength of the discarded signals will not affect the correctness of the carrier sensing by a wireless network device. The value also needs to be sufficiently small so that the aggregate strength of discarded signals does not ultimately drive mean PER (Packet Error Rate) error beyond inaccuracy bounds. For example, consider a wireless device uses DPSK modulation: based on the PER curve shown in Figure 2, one can project that simulation results will be reasonably accurate if the threshold value is 20dB below the weakest receivable signal (whose power equals the radio receiving threshold in Table 1).

## 4.2 Validation

We designed a set of experiments to validate the distance limit approach against a baseline simulator in which no weak signal is discarded. All experiments throughout this paper are conducted on a LINUX workstation with a 550MHz Pentium III CPU and 2GB of RAM. In addition to the common parameters shown in Table 1, there are 400 nodes uniformly distributed in a square terrain of size  $4000 \times 4000m^2$ , each has an IEEE 802.11b network interface and runs AODV [11]. Based on these settings and (7), a generic distance limit is calculated to be approximately 2500 meters regardless of the terrain size (see Corollary 1). There are in total 120 CBR traffic sessions in the network, and each has a data rate of eight 512-byte packets per second. To allow a reasonable amount of application traffic to successfully reach their destinations, a portion of the CBR traffic is specified among randomly chosen pairs of immediate neighbors, while source and destination pairs of the remaining CBR sessions are randomly chosen.

Average packet delivery ratio and distribution of average session throughput for all CBR traffic are the metrics for validation. The same experiments are repeated for three

scenarios: no propagation limit, distance limit, and CST propagation limit. Note that the CST propagation limit (which is -91dBm according to Table 1) is equivalent to 679 meters in terms of distance. The experiment results are shown in Figure 3. As depicted in Figure 3(a), when all CBR traffic is among direct neighbors, AODV achieves almost 100% packet delivery ratio for all three scenarios. As the proportion of one-hop traffic decreases towards 0%, the average packet delivery ratio drops as well. Results produced by applying the distance limit derived here are very close to that of baseline simulator in which no propagation limit is set. In contrast, the CST propagation limit persistently overestimates the average packet delivery ratio when the one-hop traffic occupies 20~80% of the total traffic session. The worst case is observed at the portion of 60% at which the CST propagation limit results in 50% average packet delivery ratio while the baseline simulator predicts the result to be around 20%. Note that the effect of the CST propagation limit is different from that in Section 2 (see Figure 1(a)) because the network size is much larger here. Figure 3(b) shows a traffic throughput histogram plotted from the results obtained when one-hop traffic is 40% of total traffic. Again, there is a good match between the result histogram of baseline simulator and that of applying the distance limit. In contrast, the histogram generated for CST propagation limit shows significant deviations for number of sessions achieving 2.5~7.5 Kbps and 17.5~27.5 Kbps average throughputs. The results show that the distance limit derived in our approach yields relatively much more accurate application level statistics.

## 4.3 Performance Evaluation

We designed three sets of experiments that compare the execution speeds of the QualNet simulator with or without distance limit, to evaluate the performance gain with our optimization. The common parameters again are shown in Table 1. In all sets of experiments, the nodes are uniformly distributed in a flat square terrain with 802.11b network interfaces running AODV. With the same settings as mentioned earlier, the applicable distance limit is 2500 meters. The first set of experiments is designed to fix the node density to one node per  $200 \times 200m^2$ , while varying total number of nodes in the wireless network from 400 up to 3200. Thirty percent randomly chosen nodes each have their own CBR session of ten 512-byte packets per second to another randomly selected receiver within two hops of distance. Figure 4(a) depicts the speedup of using our distance limit with

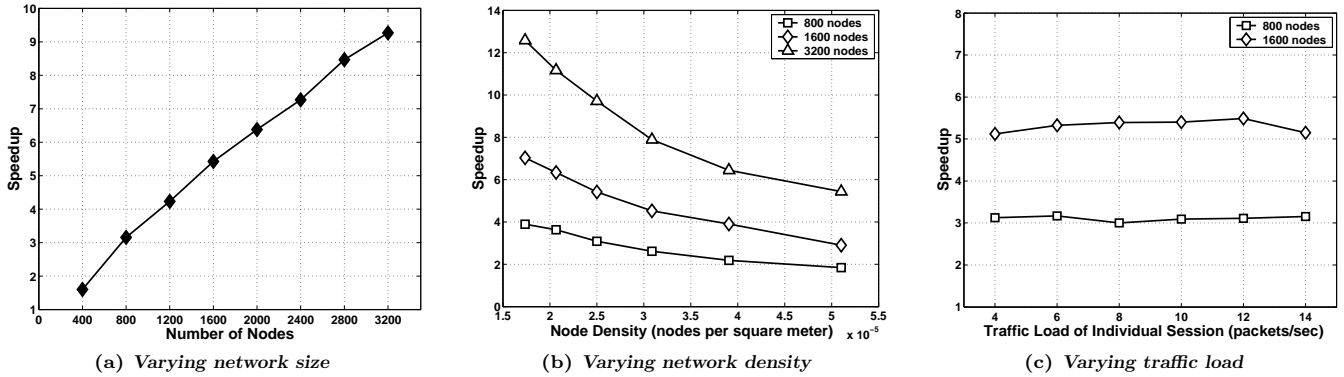


Figure 4: Speedups attributed to our distance limit

various network sizes. When there are 400 nodes in the wireless network, the relative speedup factor for applying our distance limit is about 1.6. However, it grows almost linearly with an increasing number of nodes. When there are 3200 nodes in the wireless network, speedup of greater than 9 is achieved by our distance limit. This shows that our distance limit still preserves good speedup for simulation of large scale wireless networks.

In the second set of experiments, the same configuration as above is employed, except that node density varies (by changing the terrain size) from one node per  $240 \times 240 m^2$  up to one node per  $140 \times 140 m^2$ . In accordance with previous set of experiments, results in Figure 4(b) also show that larger network size leads to greater speedup. However, for networks containing same number of nodes, increasing node density would reduce simulation speedup. This is because more neighbors are falling within the distance limit of a transmitting node when node density increases. As a result, the number of events generated per transmission increases with node density even when the distance limit is used.

A third set of experiments evaluates the speedup achievable by our distance limit with varying CBR traffic load, while other settings are the same as those of the first set of experiments. In this experiment, thirty percent of nodes each have a CBR session at rates ranging from four to fourteen packets per second. As depicted in Figure 4(c), the speedup of the distance limit remains constant for networks of both 800 nodes and 1600 nodes.

Compared with the results of using CST as propagation limit in [7], the speedup achieved by our distance limit is reduced by a factor of 1.5. When higher fidelity is required in the network simulation, this loss should be tolerable. Nevertheless, it motivated the author to use other means to further improve simulation scalability which will be described in detail in Section 5.

## 4.4 Impacts of Partitioning

Partitioning in wireless network simulation is not a novel concept, and has been studied in [14, 8]. In [8], partitioning is reported to improve the performance of *ns-2* by a factor of up to 30 with presence of CST propagation limit. We find that the improvement varies among different network simulators. For completeness of this study, we report our findings here. The experiment focuses on implementing grid-based partitioning for wireless networks of 1600 and 3200 nodes. Results in Figure 5(a) indicate that when the partition size is  $150m$  or  $300m$ , partitioning achieves the best speedup. Sim-

ulation runs 18% faster for network of 1600 nodes and 38% faster for 3200 nodes at density of one node per  $200 \times 200 m^2$ . The improvement is also dependent on the node density in the wireless network, as indicated by results in Figure 5(b).

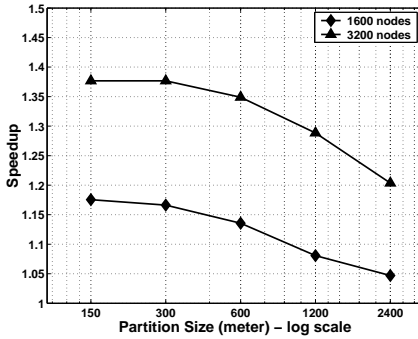
## 5. SIMULATION OF PHYSICAL DEVICE

The optimization technique in Section 4 improves wireless network simulation speed significantly, while demonstrating that bounding the aggregated power of ignored signals keeps good accuracy of simulation results at the application level. However, the propagation derived through our analysis (such as  $2500m$  in early experiments) is larger than CST propagation limit. As a consequence, a tremendous amount of events are still generated to all nodes within the distance limit for every packet transmission. Experiments show that after applying the distance limit, they still account for about 90% of total events, causing significant event scheduling overheads in wireless network simulation. In this section, a novel technique called Lazy Event Scheduling with Corrective Retrospection (LSCR) is presented to address this problem, using the IEEE 802.11 DCF (Distributed Coordination Function) operation as an example. With LSCR, events corresponding to the arrival or end of radio signals will not be scheduled, unless they may trigger state changes (e.g. begin to receive a packet) to a physical network device that might yield causality violations if processing of these state changes was delayed. The idea to be presented should not be restricted to 802.11, but is general for simulating CSMA-based networks. We assume readers are very familiar with details of the IEEE 802.11 DCF operation [6].

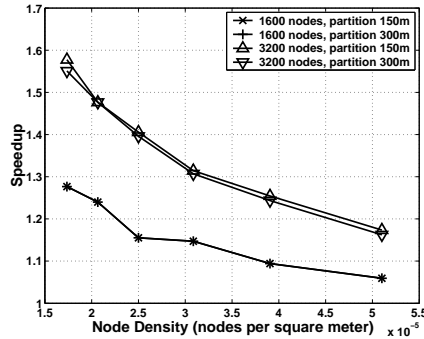
### 5.1 Events Reduction

#### 5.1.1 Lazy Event Scheduling

To further reduce event scheduling overheads for simulating signal transmissions, the effect of an incoming signal on receiver device operation is studied. Suppose a node begins transmission at time  $t_0$ , and its signal will arrive at a neighbor at  $t_1$ . A snapshot of what the simulation already knows about the neighbor at  $t_0$  is shown in Figure 6. As illustrated in the figure, presence of arriving signal can have no (case 1,2), partial (case 3) or complete (case 4) overlap with ongoing transmission (or reception) at the neighbor. When overlap exists, the neighbor will be unable to receive the arriving signal regardless of its strength. According to IEEE 802.11 standard, radio device uses signal energy and quality (PN code correlation strength) thresholds as the criteria



(a) Varying partition size



(b) Varying node density

Figure 5: Performance of partitioning

for channel assessment or signal reception. Most simulators use the same criteria (Receiving Threshold in Table 1) to decide if a signal is receivable. In other words, the incoming signal cannot be locked on for reception if its power is below the receiving threshold. Hence, when there is no overlap, simulation can at least decide that the arriving signal is non-receivable if its received power is below the receiving threshold. Intuitively, most signal arrival/end events correspond to *non-receivable signals* in the simulation.

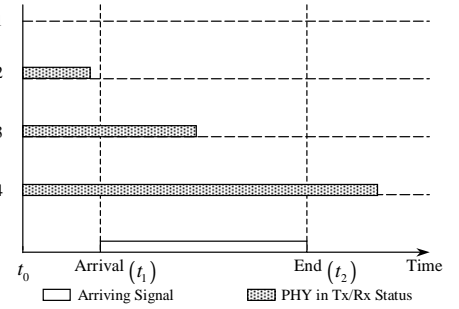
In 802.11 DCF operation, non-receivable signals can only change the result of physical carrier sensing, which in turn affects DIFS (DCF Inter-Frame Space), EIFS (Extended IFS) or BO (Back-Off) timer operation. When a node needs to start a transmission dialog, it has to wait for at least one DIFS/EIFS timer regardless of BO timer. Hence, arrival of a non-receivable signal does not cause a node to immediately receive or transmit a packet. That is, absence of signal arrival/end event will not yield causality errors regarding the fate of the signal. Here a simulator does not schedule events for a neighbor as long as the respective arriving signal is non-receivable, but rather add the signal directly to the neighbor's history of incoming signals with pertinent information such as signal arrival/end time etc. This technique is referred to as *Lazy Event Scheduling* (LS). Note that LS still violates other causality constraints because neighbors will not be able to simulate channel status (busy or idle) and SINR changes at exact times without corresponding events of non-receivable signals. Subsequently, network device operations may also be altered. To correct the causality errors, the following novel mechanisms utilizing IEEE 802.11 PHY and MAC device operation attributes, collectively referred to as *Corrective Retrospection* (CR), are introduced.

### 5.1.2 Corrective Retrospection

The physical channel sensing status changes will not be updated on time due to the dropping of events triggered by non-receivable signals with the LS approach. Ignoring the non-receivable signals will yield errors to changes of either channel sensing status or SINR of a node, which subsequently alter packet error evaluation and DIFS/EIFS or BO timer operations. The following will describe how these errors are corrected by CR mechanisms.

#### Packet error evaluation using CR

Network simulators use SINR as the input to lookup BER and then calculate PER depending on packet bits length during which SINR remains constant (See Figure 2 for example). With LS, all potentially receivable signals still trigger the scheduling of signal arrival events to a node. The

Figure 6: Snapshot of physical status of a neighbor node (at  $t_0$ ) and an arriving signal

node will lock onto the first arrived receivable signal when it processes the corresponding packet arrival event with earliest timestamp. The node therefore ignores subsequent packet arrival events throughout the duration of current packet reception, as those signals become merely interference. A packet end event will be scheduled to the same node when packet arrival event is processed. When it is time to process the corresponding packet end event, all interference signals must have already been present in the signal history of current node. Simulation will use the complete history of all arrived signals to evaluate the packet for error.

Figure 7 shows an example of packet error evaluation with CR. Every signal arrival or signal end transition during the reception of the current packet can be retrieved from the complete signal history at the time when the currently received signal ends ( $t_2$ ). As each interference signal adds a constant amount of energy during its presence, SINR is a piecewise step function that changes whenever a signal arrives or ends. Thus packet error evaluation can be conducted as usual along the reconstructed piecewise SINR step function as shown in the figure. Note that before optimization, the computational cost for evaluation of packet error is amortized among all signal arrival and signal end events. LSCR instead will postpone the packet error evaluations until the end of the currently received packet. As long as all signals are present in the history and their information (arrival/end time etc.) is correct, the evaluation of packet error will be correct. Note that as simulation progresses, the signal history at a node will grow steadily. A partial solution is that whenever a node finishes receiving a packet at the time of its signal end event, all signals ended before that time can be discarded from the history.

#### Timeout readjustment using CR

Besides packet error evaluation, LS also affects DIFS/EIFS or BO timer operations. With LS, simulation will not send signal arrival/end events to a node for non-receivable signals. Without these events, DIFS/EIFS and BO timers cannot be cancelled or restarted appropriately, as exact transitions of the channel status between busy and idle is unknown. Instead of actively cancel and resume timer when channel condition changes, CR technique lets false timeout occur and reverses the errors retrospectively. Consider DIFS and BO timer combinations, when current BO timeout event occurs, simulation can reconstruct past channel conditions via complete signal history since last BO timer started. As shown in Figure 8, current BO timeout event is false positive since channel is not idle throughout the entire backoff duration. With reconstructed history of channel conditions, the node

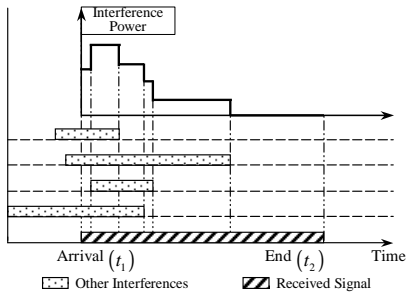


Figure 7: Packet error evaluation using CR

can investigate how much the BO counter should have been decremented by inspecting all the idle periods within backoff duration. In this example, there is a single idle period for the BO timer to be reduced by value  $BO'$ , and the channel is currently busy (when BO timeouts). It can be deduced that the next resumed backoff process will not complete until the channel becomes idle again for  $DIFS+BO-BO'$ , since the backoff will only be resumed when the channel becomes idle again for DIFS period. With above knowledge, the node can predict that the earliest time for BO timer to expire again is  $DIFS+BO-BO'$  after the next time when the channel becomes idle. Thus, a new timeout event will be scheduled for the projected BO timeout time and the process repeats when that timeout event is processed. Eventually the prediction will be correct and the node can transmit a packet when the last BO event is processed.

Besides the above example, two other issues need to be addressed. First, if the channel is idle (when BO timeout) at a node in the backoff process and was not idle for the past entire timeout duration, the prediction for the next BO timeout event needs to start from the most recent channel busy-to-idle transition in the history because the real backoff was resumed from then. Second, the backoff process can be terminated if a node receives a transmission dialog request from others such as an RTS or data packet as it needs to reply with CTS or ACK after an IFS time. When this occurs, the node will cancel the pending timeout event, calculate and memorize the remaining BO value. The backoff process will be reinitiated with the remaining BO value after the node finishes its obligated replying transmission. Last, signals that ended when a timeout event is processed will be removed from the node's signal history.

## 5.2 Greedy signal evaluation

With LSCR, assessment of channel condition and evaluation of PER is done retroactively by inquiring the signal history (implemented as a list). In a large network with high traffic volume, the signal history grows fast and scanning the complete list of signals can be costly. To reduce scanning cost, simulations have to find means to either reduce the number of signals kept in the signal history or skip some signals when performing CR. At the same time, the inaccuracies thus introduced should have minimal impacts on PER. Figure 2 plots BER and PER as a function of SINR given different packet length. It can be seen that there exists a critical region (For example, SINR is between 5 and 15 in Figure 2). A slight change in SINR will have significant impact on mean PER. To quantify the inaccuracies introduced by excluding weak signals in evaluating PER, we define an upper bound  $\varepsilon$  (such as 1%) to the error of estimated PER.

First, we will remove unnecessary signals from the history

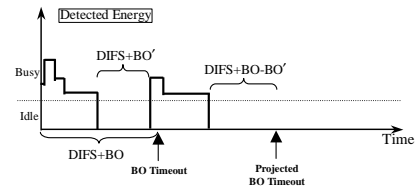


Figure 8: Timeout Readjustment using CR

list. Note that a unified distance limit derived in Section 4 is based on the power level of weakest receivable signal. The derived distance limit is very conservative since a received signal usually has stronger power. Thus, when a node locks on to a signal for reception, simulation can determine a dynamic distance limit. An interfering signal transmitted by another node beyond this distance limit will not be added to this node's signal history. Secondly, to skip unimportant signals, a signal history is subdivided into groups according to signal strength (A step of 10dB is used in the current implementation). The PER evaluation starts from the group containing signals with the highest power level. If the estimated SINR obtained by merely considering all the strong signals and noise is so low that the obtained segment PER is greater than  $1 - \varepsilon$ , PER evaluation terminates, concluding that the estimated segment PER is accurate enough for packet error evaluation. If no strong interferences are found for a segment of the packet and estimated aggregated signal strength in other groups is not significant enough to reduce actual SINR to critical region (applying Proposition 2), then simulation concludes that the current segment has no bit error without scanning the remaining groups, and continues evaluating SINR for the next segment.

## 5.3 Performance Evaluation

To evaluate additional performance gains produced by the optimization techniques described in this section, the following set of experiments was conducted. Exactly the same settings as those used in first set of experiments at subsection 4.3 plus a partition size of  $150m$  were employed here so that the simulation speedups obtained in this section can be multiplied to those in last section to derive the aggregated speedups achievable by optimization techniques throughout the paper. Note that baseline simulator here has incorporated the optimizations presented in Section 4.

The speedups of LSCR in the same experiments are depicted in Figure 9(a). When there are 400 nodes in the wireless network, the relative speedup factor for LSCR is 2.5. The speedup grows almost linearly with an increasing number of nodes. When there are 3200 nodes in the wireless network, more than 4 times speedup is achieved by LSCR. The reason why LSCR can achieve such a high speedup factor over baseline simulator already equipped with optimizations in Section 4 is revealed in Figure 9(b), which shows the average total number of events processed by the network simulator for each second of simulation before and after LSCR is exploited. The results show that on average, the number of events generated by the simulation has been reduced by more than 20 times when LSCR is used. For instance, in a network of 3200 nodes, the events generated for one second of simulation is reduced from 5,315,095 to 210,396, i.e. a 25 times reduction. Furthermore, Figure 9(a) also shows that when greedy signal evaluation is applied, the simulation speed is improved by about another 5% for all network sizes in the experiments. To evaluate the impact of different

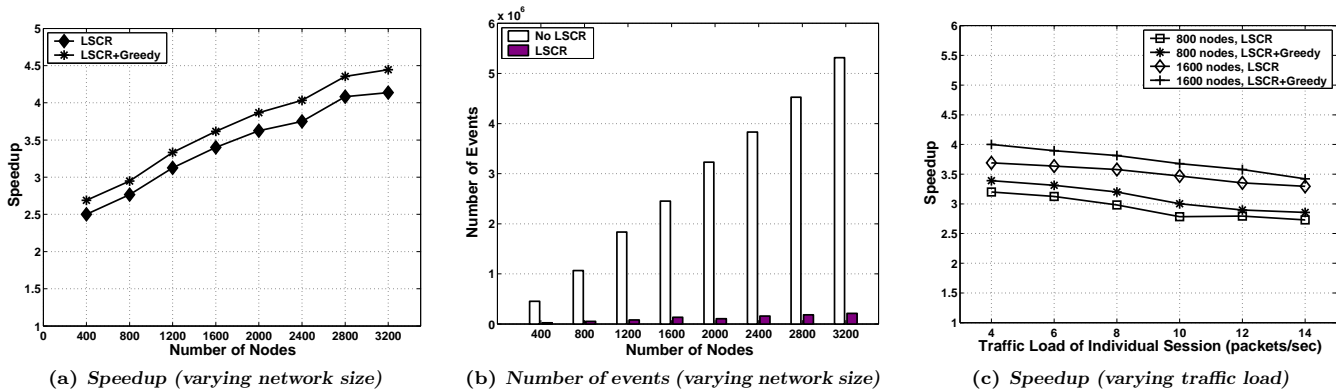


Figure 9: Performance of physical device optimization (varying traffic load)

traffic loads on the simulation speedup, the last experiment in Subsection 4.3 is repeated with the optimizations presented here. Figure 9(c) shows the simulation speedups for network of 800 or 1600 nodes. The speedups drop slightly when the network traffic increases, but it is projected to stabilize when individual CBR traffic rate is further increased. In all cases, about another 5% speedup can be achieved on average with greedy signal evaluation.

To give an idea of what is the total amount of simulation speedup achieved by various optimization techniques presented, consider 3200 nodes uniformly distributed in a square terrain with node density equal to one node per  $200 \times 200 m^2$ , with 960 CBR sessions randomly chosen among nodes within two hops of proximity. When the distance limit is applied, a speedup of 9.2 is obtained (See Figure 4(a)). When partitioning of  $150m$  partition size is included on top of it, additional speedup of 1.37 is achieved (See Figure 5(a)). Finally, another 4.4 speedup is obtained with LSCR and greedy signal evaluation (See Figure 9(a)). As the optimizations are included incrementally in the experiments, a total speedup of about 55 is obtained in this particular case.

## 6. CONCLUSIONS

This paper has presented a theoretical analysis to radio signal propagation. Based on the observed characteristics of signal propagation models, it presents corresponding distance limit and greedy signal evaluation optimizations that aims to quantify and bound the inaccuracies produced at the physical device simulation layer. Experiments have demonstrated that bounding inaccuracies at physical layer is critical to the fidelity of application level simulation results. Besides the above, impacts of a novel technique named Lazy event Scheduling with Corrective Retrospection are presented. This technique does not alter the correctness of the simulation at all while achieving significant performance gains. Using QualNet, one of the most efficient wireless network simulators, as an example, this paper has demonstrated that with various optimization techniques presented here, an aggregated speedup of up to 55 times can be achieved for simulation of wireless network of 3200 nodes while preserving simulation accuracy.

Although this paper has quantitatively studied the inaccuracies of various optimizations presented here, the bounds to inaccuracies at signal propagation modeling are still chosen empirically to achieve accurate results at application level. A future extension to this work would be to study the effects

of relationship between inaccuracies at physical or devices layers and those at upper layers. Another aspect of future work is to study the compatibility of these techniques with parallel network simulators, while a third dimension is to conduct similar performance optimizations to more complex network simulation with deterministic propagation models.

## 7. REFERENCES

- [1] R. Bagrodia, R. Meyer, M. Takai, Y. Chen, X. Zeng, J. Martin, B. Park, and H. Song. Parsec: A Parallel Simulation Environment for Complex Systems. *Computer*, 31(10):77–85, Oct 1998.
- [2] M. J. Feuerstein, K. L. Blackard, T. S. Rappaport, S. Y. Seidel, and H. H. Xia. Path Loss, Delay Spread, and Outage Models as Functions of Antenna Height for Microcellular System Design. *IEEE Transactions on Vehicular Technology*, 43(2):355–365, May 1991.
- [3] H. T. Friis. A note on a simple transmission formula. In *Proceedings of the IRE*, volume 34, pages 254–256, May 1946.
- [4] GloMoSim. <http://pcl.cs.ucla.edu/projects/gloimosim>.
- [5] M. Hata. Empirical Formula for Propagation Loss in Land Mobile Radio Services. *IEEE Transactions on Vehicular Technology*, 29(3):317–325, Aug 1980.
- [6] International Standard ISO/IEC 8802-11: 1999(E), ANSI/IEEE Standard 802.11, 1999 Edition.
- [7] Z. Ji, J. Zhou, M. Takai, and R. Bagrodia. Optimizing Parallel Execution of Detailed Wireless Network Simulation. In *Proceedings of PADS'04*, May 2004.
- [8] V. Naoumov and T. Gross. Simulation of Large Ad-hoc Networks. In *ACM MSWiM'03*, Sep 2003.
- [9] ns-2. <http://www.isi.edu/nsnam/ns2>.
- [10] PDNS. <http://www.cc.gatech.edu/computing/compass/pdns>.
- [11] C. E. Perkins and E. M. Royer. Ad hoc On-Demand Distance Vector Routing. In *IEEE WMCSA '99*, Feb 1999.
- [12] QualNet. <http://www.qualnet.com>.
- [13] T. S. Rappaport. *Wireless Communications: Principles & Practice*. Prentice Hall, 2 edition, 2002.
- [14] M. Takai, R. Bagrodia, A. Lee, and M. Gerla. Impact of Channel Models on Simulation of Large Scale Wireless Networks. In *ACM MSWiM'99*, Aug 1999.
- [15] M. Takai, J. Martin, and R. Bagrodia. Effects of Wireless Physical Layer Modeling in Mobile Ad Hoc Networks. In *ACM MobiHoc'01*, Oct 2001.