Analysis and Simulation Model of Physical Carrier Sensing in IEEE 802.11 Mesh Networks

Hui Ma, Eiman Alotaibi, Sumit Roy*
Dept. of Electrical Engineering
U. Washington Box 352500
Seattle, WA 98195
{mahui, eman76,roy}@ee.washington.edu

Abstract

In IEEE 802.11 Wireless Mesh networks, tuning of the Physical Carrier Sensing (PCS) threshold based has been shown to be an efficient way to balance the hidden vs. exposed terminal trade-off and hence improve aggregate network throughput. Clear Channel Assessment (CCA) indicates channel status by comparing the net RF energy with a suitable chosen PCS threshold to control which nodes may attempt channel access. Proper CCA implementation is critical for tuning of aggregate 802.11 network performance. However, we found that the CCA implementation in OPNET v.11 (henceforth termed as default OPNET model) has several shortcomings. These are identified with examples, and in the new model, the CCA and packet reception chain have been modified to conform better to the IEEE 802.11 specifications. The resulting new code will be made available at -
http:\\www.ee.washington.edu\research\funlab\pcs_model .htm

1 Introduction

Physical carrier sensing (PCS) is one of the two main interference mitigation (contention resolution) mechanisms defined in the PHY/MAC layers of 802.11 WLANs [1]. A node that intends to transmit first assesses the current channel state (this is generically termed as Clear Channel Assessment or CCA) by comparing the measured on-air received energy against a predefined PCS threshold to determine if it should contend for channel access as per the CSMA/CA protocol. Each node samples the *net* energy level on-air and initiates channel access only if the detected value is below the threshold, indicating that the channel is 'free' of significant ongoing transmissions. The PCS threshold effectively defines a carrier sensing *range* that denotes an area wherein a secondary transmitter is prevented from contending for access so as to not disrupt the reference transmission.

However (as is well known), sensing the channel at the transmitter's location is not always an accurate predictor of the channel state at the receiver, leading to the well known *hidden and exposed terminals*, both of which degrade aggregate throughput. By accepted definition, **hidden nodes** occur when *two transmitters* that are outside the mutual *carrier sensing range* attempt to transmit; if the secondary source lies *within* the interference range of a common (reference) receiver, it results in a packet loss (or 'collision'). Conversely, **exposed terminals** occur when a source lies *within* the sensing range of the reference transmitter but *outside* the interference range of the reference receiver. Hidden and exposed nodes lower aggregate network throughput in different ways: while, hidden nodes *disrupt* the reference transmission (lead to incorrect decoding or loss of the reference packet), exposed nodes represent *lost* throughput by needlessly suppressing simultaneous transmissions that could have occurred without

---

*This work was partially supported by a gift from Intel Corpn.

[1] The other approach is based on RTS/CTS or Virtual Carrier Sensing (VCS) that is not the subject of this work.

disruption to the reference transmission, i.e., exposed nodes represent a loss of spatial reuse. Further, there exists a typical trade-off between exposed and hidden nodes, as explained in [1]; i.e. increasing one reduces the other. Use of the PCS threshold is an effective method [5] for exploring this trade-off and achieving optimal aggregate network throughput. Several current .11 WLAN hardware/firmware support one or more parameters for PCS control; some of these are available at run-time for user definition via open source linux drivers such as IPW2200 [2] for Intel's Pro/Wireless chipsets.

We have shown analytically that under suitable conditions, the optimal PCS threshold is obtained when the carrier sensing range equals the interference range [1]. For corroboration, we sought to validate this result via OPNET. In developing the simulation, we discovered several bugs with OPNET CCA implementation that necessitated changes. The purpose of this paper is to describe these modifications and present results using the resulting modified OPNET simulator.

The rest of this paper is organized as follows. We first discuss the limitations concerning CCA/ED and the packet reception chain in OPNET v. 11 implementation. Next, we describe the new PCS Model that we implemented. Finally, we present a set of .11 network performance simulation results based on the new models.

2 Default OPNET PCS Model

The commentary in this section is aimed at educating users who are contemplating use of 802.11 PHY/MAC as implemented in OPNET v.11. This is timely and useful because of some important shortcomings of OPNET 802.11 code which was discovered during our investigations that are pertinent to a) physical carrier sensing and b) packet decoding. Since these are potential contributors to (significant) deviations between OPNET simulation results and those based on analytical models, we identify these 'bugs' and provide fixes in the new OPNET codes.

At the outset, it is useful to reiterate that OPNET represents the PHY as pipelined 'black-box' stages, whose input-output characteristics are based using analytical (equivalently, table lookup) functions. We next state a few key facts about OPNET link layer implementations.

1. In the absence of any ongoing transmission, the receiver at any node observes thermal noise; the thermal noise power level is constant and set at $P_n = -101$ dBm for the 20 MHz bandwidth of an 802.11a channel.

2. If a source node i is transmitting, OPNET computes the *average* power at the reference receiver due to this transmission as a constant $P_r(i)$ based on a) the transmit power of the source node i (which is configurable) and b) the distance between the source i and the reference node. The relation used for this computation in default OPNET is the standard Friis path loss model for RF propagation with a path loss exponent of $\gamma = 2$, i.e. free space path loss (which is clearly inappropriate for typical indoor scenarios). *The default OPNET model also does NOT include any short-term (multipath) fading.*

3. Although CCA and packet decoding at a node are two distinct functions (the first relates to a transmit node seeking channel access, the later relates to PHY layer reception at a receive node) that should logically be triggered by *two* distinct thresholds (i.e. the PCS threshold γ_{cs} and the receiver sensitivity P_{rx}), OPNET defines *only a single threshold* (called 'packet reception power threshold') that is used for both purposes. As we explain below, this (and other OPNET implementation artifacts) leads to significant performance deviations and necessitated modifications to the relevant OPNET PHY modules for the work reported here. The changes are consistent with features in several 802.11 hardware such as Wavelan-II [3] and Intersil for client-side adapter cards and the Alcatel OmniAccess APs that use *two* thresholds for transmit and receive.

2.1 CCA in Default OPNET

In OPNET, whenever a node transmits a packet it generates an interrupt at every other node in the network. This behavior allows every node to *know exactly when a new transmission* *begins and the associated source ID (along with the received power which is computed as described above), which is not possible in real systems.* Thus, any OPNET node seeking channel access uses the configurable parameter 'Packet Reception Power Threshold' P_R to set the "Clear Channel Assessment" CCA flag busy whenever the received power from any other *single* transmit node i exceeds P_R, i.e.,

$$P_r(i) > P_R \tag{1}$$

2.2 Packet Reception in Default OPNET

The following are *necessary* (but not sufficient) conditions for packet detection at a receive node from a source s in default OPNET v.11 implementation -

1. The CCA flag must be IDLE initially (i.e., the node cannot be contending for channel access);

2. The flag remains idle till a packet is received from a node s for which the received power $P_r(s) > P_R$; in that case, the CCA flag is set busy and packet detection is triggered, the receive node is then 'locked' into decoding this first arrival (after the busy timer of a receiver expires, the CCA flag will revert back to idle);

3. If a subsequent packet (from another node $i \neq s$) arrives within the duration of the reference packet from s, it is treated as interference; a) should $P_r(i) > P_R$, this leads to an automatic collision and the reference packet from s is lost; b) if $P_r(i) < P_R$, the node conducts packet decoding as described next;

4. The decoding of the reference packet is conducted by several pipelined stages in OPNET. One of the pipeline stages divides the reference packet duration into several sub-segments with constant SINR; for each such segment, the *effective SINR* is computed and used in a table lookup for the given uncoded modulation scheme to determine the number of bit errors in that segment. This process is continued over all the pipeline stages to find the cumulative number of bit errors in the entire packet; if this exceeds a threshold value, the packet is deemed to be lost[2].

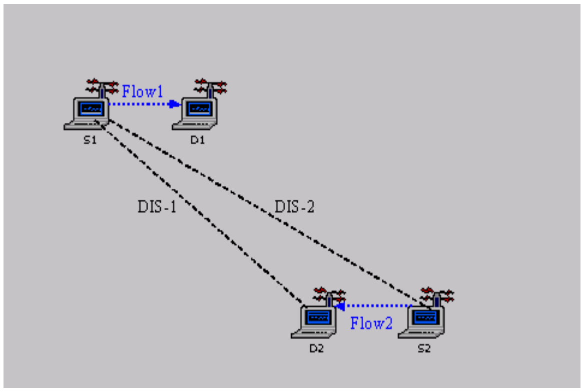

Figure 1: Example of Erroneous Collision Detection when CS range = 28m

In the following subsections we provide some examples to illustrate the bugs that exist in the default OPNET model.

2.3 Erroneous Collision Detection

Erroneous Collision Detection (ECD) problem occurs when a receive node has begun to receive a valid packet (a "valid' packet is one whose received power exceeds $> P_R$; by OPNET convention, a receiver 'locks' on to the *first* valid packet as reference and attempts to decode it) when a second 'valid' packet (from a second transmitter) appears at the input. In such cases, OPNET *always* declares this as a collision and drops the reference packet, irrespective of the SINR at the decoder input [3]. However, a more realistic physical layer model would allow the first packet to be decoded in the presence of a secondary interfering packet provided the net SINR is above a suitable threshold.

This bug results from inappropriate WLAN MAC implementation. Whenever the condition 3(a) in Section 2.2 holds, the collided_packet flag will always be set to true irrespective of the SINR at the decoder input and the reference packet will be dropped in wlan_physical_layer_data_arrival (). To verify this, the scenario with two flows shown in Figure 1 was stud-

---

[2] The default error correction code threshold value in OPNET v.11 for the number of code errors is 0, implying no error correction coding is accounted for in the link layer model.

[3] Specifically, the error resides in the function of wlan_mac_rcv_channel_status_update (int channel_id).

Table 1: Simulation set-up to test erroneous collision

Packet size (bytes)	1500
Traffic source rate (packets/second)	1000
Link data rate (Mbps)	12
Transmission power (mW)	1
Path loss exponent	2
Retry Limit	1
RTS/CTS mechanism	Disabled

ied in default OPNET 11.0 with parameters listed in Table 1. The Carrier sense threshold (Packet Reception Power Threshold) = -76 dBm is equivalent to carrier sense range R_{cs} of 28m. We test two cases using the setup shown for a) DIS-1 = 42 m, DIS-2= 46 m and b) DIS-1 = 26 m, DIS-2= 30 m respectively. Note that in both cases, the inter-source separation DIS-2 exceeds the carrier sensing range of 28 m. In the first instance, the two saturated flows can be sent simultaneously; although the two senders cannot sense each other, the interference produced at the other receiver does not disrupt the respective transmissions and per flow throughput of 10 Mbps is achieved. When DIS-1 is reduced to 26 m (second case)[4], the throughput at D2 drops to 0. In this case the senders (S1 and S2) still cannot sense each other but now receiver D2 sees a valid packet from S1 because DIS-1 is less than the carrier sense range (which equals receive range in default OPNET v.11). Simple calculation yields the SINR= 14.3 dB at D2 which is higher than 9.94 dB, the minimum SINR required for 12 Mbps as shown in Table 3. This implies that D2 could decode the packet from S1 in theory while in OPNET v.11 implementation (due to the "Erroneous Collision Detection"), the two transmissions overlap at D2 resulting in a collision and yielding 0 throughput.

2.4 No Power Aggregation in CCA

The CCA implementation in default OPNET model does not aggregate the received power from multiple ongoing transmissions. Here we verify it with a simple scenario with three node pairs as shown in Fig. 2.

The parameters used for the simulations are the same as those listed in Table 1, except the packet size = 1024 bytes, traffic source rate = 2000 packets/second and the Packet Reception Power Threshold = -95dbm with Retry Limit=7. We let the inter-source distance d to be 262 m and thus the individual received power (-95.09 dbm) at each source node from both the other two sources is just below the CS threshold (Packet Reception Power Threshold) of -95 dBm, but the combined power is about twice of that, i.e. 3dB above the threshold. In this simulation, we find that the "Data Traffic Sent" at MAC layer from S1, S2 and S3 are all 10 Mbps, which is just slightly less than the channel link rate 12Mbps. This means that they cannot sense each other and are always simultaneously sending, i.e. the default OPNET CCA model does not sense the channel as being busy despite the combined received power being above the CS threshold.

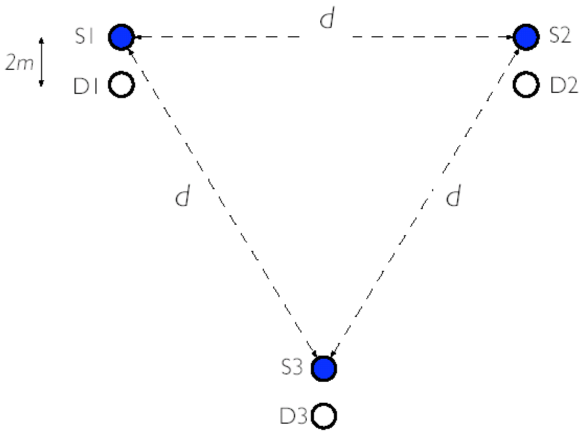

Figure 2: Example of No Power Aggregation in CCA

3 New OPNET Model Implementation Overview

In the previous section we provided examples of bugs in CCA and packet reception in the default OPNET model. Here, we describe the fixes to the above contained in our new model.

3.0.1 New PCS Model

I. Changes to CCA

1. The default CCA/ED module Eq (1) does *not* aggregate the power from (multiple) ongoing transmissions for ED [5].

---

[4] The distance between each pair of sender-receiver is 5 m

[5] To the best of our knowledge, Intel 2915 cards implement CCA per Eq.(2) and thus OPNET based simulation results can be expected to differ from those in a test-bed using the above hardware.

The corrected rule is now

$$\sum_i P_r(i) + P_n > \gamma_{cs} \tag{2}$$

When the threshold γ_{cs} is exceeded, the CCA flag is set busy. The l.h.s is computed at each instant when a (new) packet transmission begins and ends, and the flag remains busy as long as Eq. 2 is satisfied; it reverts back to idle when the sum power drops below γ_{cs}.

2. Along with the above, *two* distinct thresholds γ_{cs} and P_R are now introduced in the modified PCS module respectively - the second threshold P_R is used to set the PHY RX flag described next.

3. The fact that packet detection is triggered using the *same* threshold as CCA in default OPNET has implications for network performance evaluation. Consider the following: if the CCA is already busy (i.e. the receiver is already attempting to decode a reference packet), any subsequent new packet arrival whose received power exceeds P_R is deemed to automatically result in a collision, i.e. the reference packet will be dropped. Clearly, this is unduly conservative; the condition for decoding a packet should be based on a received SINR threshold, and not on a threshold on the interference power. Re-stated, it is possible in principle that the receiver with CCA busy can still detect an incoming packet, but this was not possible in default OPNET v.11.

II. Changes to Packet Reception

The following are revised *necessary* (but not sufficient) conditions for the packet to be received successfully at a receive node from a source s:

1. The PHY RX flag at the receive node is IDLE initially (i.e., the received power of no single packet exceeds P_R);
2. The PHY RX flag remains idle till a packet is received from a node s for which the received power $P_r(s) > P_R$; in that case, the PHY RX flag is set to busy and packet detection is triggered; the receive node is then 'locked' into decoding this first arrival;
3. If a subsequent packet (from a node i) arrives within the duration of the reference packet from s, it is treated as interference; but irrespective of the received power of the interference packet, the packet decoding pipelined chain is always triggered.

To summarize, we emphasize two important changes between the modified and default OPNET implementation for packet detection. First, packet detection is now triggered by *PHY RX flag set to busy* which is *different* from the CCA busy condition in the modified module. Second, when the receiver is attempting to decode a reference packet, the arrival of a subsequent packet whose received power $P_r(i) > P_R$ does *not* automatically imply a collision.

III. Changes to Pipelined Packet Decoding

In the pipelined packet decoder in default OPNET v.11, the decision to drop a packet is based on a table-lookup BER. Unfortunately, this is incorrectly done [6]- it uses the symbol SNR whereas $\frac{E_b}{N_0}$ should be used in computing BER (in other words, the symbol SNR should be converted to bit SNR taking into account the modulation scheme). The impact of this error in the default OPNET model is shown in Figure 4 and Table 2. The transmission range for the data rates (18, 12, 9 and 6 Mbps), (36 and 24 MBPS) and (54 and 48 MBPS) are identical as a result. This bug has been corrected in our modified modules [7].

We again remark that even the modified OPNET implementation of packet reception is based on an artifact that is incommensurate with reality: the presumption that multiple overlapping packets arriving at a receiver can be disambiguated with CCA/ED automatically due to the interrupts inherent in the discrete event nature of OPNET simulator. Note that OPNET has no facility to do preamble detect (PD) (based on .11 PLCP preamble, for example) that potentially would allow this in real hardware. A future implementation should consider a 2-stage process where a preliminary CCA/ED flag using a threshold P_R on the *total* received power is followed by a PD stage to separate multiple sources. This would allow computation of the SINR for each refer-

---

[6] Specifically, the error resides in the BER pipeline model Wlab_ber.

[7] As as has already been remarked, the link layer is presumed to be uncoded; future link models should account for the coding gains due to the error control coding schemes implemented in 802.11 standard.

ence packet (where the remainder are treated as interference) which can be used to trigger the pipelined packet decoding stages.

3.1 New PCS Model Implementation Details

In this section, we present how we modified the MAC layer process model “wlan_mac” and pipeline stages models to conform to the features above.

Other than two distinct thresholds γ_{cs} and P_R in the modified PCS module for the triggers of CCA Busy and PHY layer reception respectively, in the wlan_mac process model we also use two distinct flags to represent the state of CCA and PHY layer reception. The ”wlan_flags $- >$ receiver_busy” flag will be used only for CCA, which is the legacy flag used in the default model for both CCA and PHY RX; while a new flag ”rcv_channel_status” is introduced for PHY RX.

In the default OPNET, the change of CCA status and packet reception are *both* triggered by the value of the received power of the most recent individual packet at any node; this information is sent from the receiver module to the wlan mac module in the current model. In view of the proposed changes, the above is no longer suitable for triggering change of CCA status, as the latter should be based on the sum power (of all overlapping transmissions seen at the receiver) instead of the power of a single packet. So, in our new CCA/ED model, we preserve this part of code only for the trigger of PHY RX; but in addition, at each instant when a packet transmission begins and ends, we recalculate the sum power at each station in the power pipeline stage (wlan_power) and the last pipeline stage (wlan_ecc). When the sum power rises over or drops below γ_{cs}, the CCA status in the wlan_mac process will be updated with a newly introduced remote trigger, WLAN_PHY_REMOTE_INTRP.

3.2 OPNET PHY Models: Basic Range Definitions

Fundamental to any credible performance evaluation of 802.11 networks are appropriate definitions of the three basic ranges - namely, transmission, interference and carrier sensing ranges. All experiments were conducted in OPNET v.11 using 802.11a models by defining IP flows between any Source (S)-Destination (D) pair. Each flow consists of 1500 byte IP packets (including IP headers) that are input to the Layer 2 buffer at the source node; the default transmission power of 1 mW is used for all senders. Of these, the amendments to OPNET v.11 that impact CS range by aggregating all the received power as in Eq. (2) has already been described. We next describe the changes for the others.

3.2.1 Transmission Range

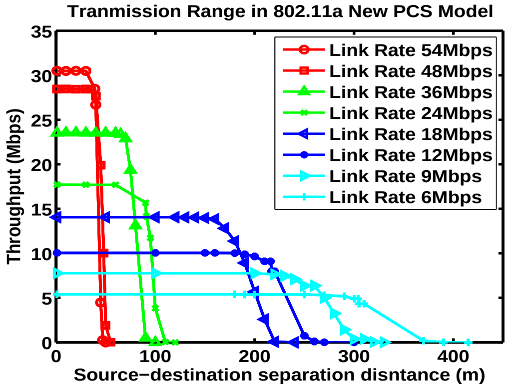

Figure 3: Transmission Range in OPNET using new PCS model (11a)

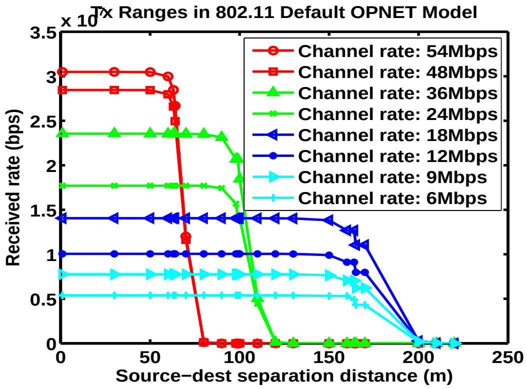

Figure 4: Transmission Range in OPNET using Default model (11a)

We setup the transmission range simulations in OPNET by using a single reference (S1-D1) pair with variable separation D. The range-rate curves in Figure 3 and Figure 4 were obtained as follows: the maximum throughput over the link is

Table 2: SNIR Threshold for .11a in OPNET (PER=10%)

Link Rate (Mbps)	Default OPNET (dB)	New PCS Model (dB)
6	9.94	4.58
9	9.94	6.64
12	9.94	7.55
18	9.94	9.63
24	14.42	15.16
36	14.42	16.86
48	18.25	21.57
54	18.25	22.42

Table 3: Rate-TransmissionRange for .11a in OPNET (PER=10%)

Link Rate (Mbps)	Default OPNET (m)	New PCS Model (m)
6	164	304
9	164	240
12	164	216
18	164	170
24	98	90
36	98	74
48	63	43
54	63	39

first obtained for a sufficiently small D by measuring the *receive rate* at the receiver (this should correspond to the chosen link capacity discounted by the MAC header overhead). Then D is gradually increased till 10% packet error rate (PER) relative to the maximum throughput is observed; the distance at which this occurs is defined as the transmission range for that link capacity. The resulting transmission ranges are listed in Table 3 and effectively represents the differences in transmission ranges between the two models. Usually the transmission range measured using the new model is longer than the one measured by the default model. Since our subsequent simulations are conducted for link rate of $12\,Mbps$, we note the corresponding $R_{tr} = 216\,m$ for new model and $R_{tr} = 164\,m$ for the default model.

3.2.2 Interference Range

Interference range (R_I) is estimated in OPNET via the experimental set-up depicted in Fig.5. A reference flow F1 (S1-D1) is established for a given separation D and the throughput at D1 measured. A second flow F2 (S2-D2) is then introduced as shown where the distance S1-S2 exceeds the carrier sensing range. The minimum value of DIS at which the throughput of F1 shows a 10% PER relative to the measured throughput in the absence of F2 is determined to be the interference range.

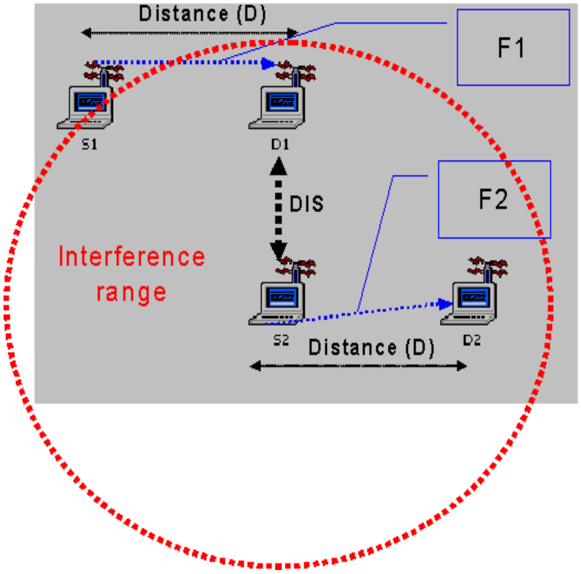

Figure 5: Interference Range setup in OPNET

Interference range using the new model gives more reliable results. This result becomes meaningful when we test both models and compare it to our theoretical model. In the following discussion we will give a brief description of this theoretical equation.

For nominal Tx (S1)-Rx(D1) separation D (where $D < R_{tr}$) and ONE interfering concurrent source S2 at a distance DIS from the receiver D1 as shown in Fig.5, the signal-to-interference cum noise ratio SINR at D1 is given by

$$SINR(D1) = \frac{P_{ref}/D^{\gamma}}{P_n + P_{ref}/\mathrm{DIS}^{\gamma}} \tag{3}$$

we will also use the definition of S_0,

$$S_0 = \frac{P_{ref}/R_{tr}^{\gamma}}{P_n} \tag{4}$$

where P_{ref} is the transmit power at a reference distance (from Tx.), P_n is the additive background noise power and γ is the path loss exponent.

The interference range R_I is the value of DIS whereby the $SINR(D1)$ from Eq.4 equals S_0 in Eq.3. Equating Eq.4,3 yields after some simple manipulations

$$R_I = \left(\frac{P_{ref}/P_n}{(R_{tr}/D)^\gamma - 1}\right)^{\frac{1}{\gamma}} = (S_0)^{\frac{1}{\gamma}} \frac{R_{tr}}{((R_{tr}/D)^\gamma - 1)^{\frac{1}{\gamma}}} \quad (5)$$

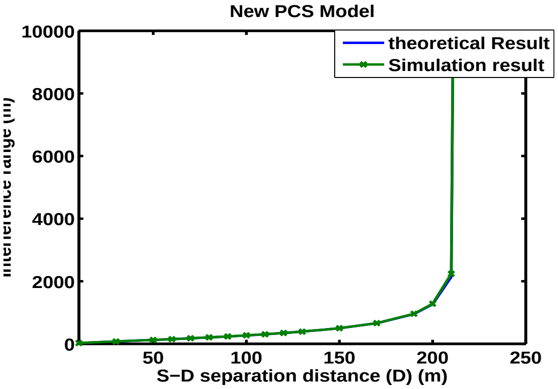

Figure 6: Estimating Interference Range: $R_{tr} = 216\,m$, Data rate = 12MBPS

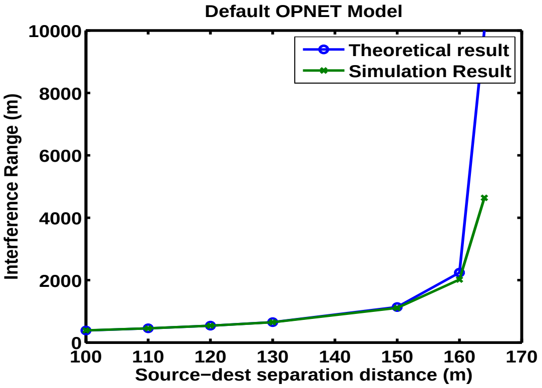

Figure 7: Estimating Interference Range: $R_{tr} = 164\,m$, Data rate = 12MBPS

Eq.5 provides a *theoretical* expression for R_I based on the amount of link margin available when $D < R_{tr}$; i.e., only a secondary transmitter within a radius R_I of the intended *receiver* will disrupt the reference transmission (i.e. lead to loss of reference packet). It can be shown by taking derivative of Eq.5 that R_I is monotonic increasing with D, implying that increasing $D \rightarrow R_{tr}$ results in the link becoming more vulnerable to interference as can be expected. Specifically, as $D \rightarrow R_{tr}$, $R_I \rightarrow \infty$ implying the loss of all link margin in the limit and hence ANY concurrent transmitter (at whatever distance from the reference receiver) causes the reference packet to be dropped as the receiver SINR drops below the threshold needed for packet decoding.

The result in Fig.6 displays excellent match between the new PCS model and the theoretical model, whereas Fig.7 shows that the default OPNET model doesn't conform to the functional dependance of R_I on D predicted by Eq.5. The reason behind this mismatch is the error while calculating the transmission range of the default model in OPNET due to incorrect S_0.

4 Performance of PCS Tuning on Throughput Using New OPNET Model

In this section, we use the new PCS model simulations to directly study the effect of modifying the carrier sense range on the network throughput.

OPNET simulations were run on a 10x10 square grid of nodes with a grid spacing of 10 m. The reception sensitivity was set such that the reception range was 10 m; thus a node can only receive packets from its one-hop neighbors that are upto 10 m away. The physical layer used was 802.11a at 12 Mbps and the carrier sense range Rcs was varied from 20m to 32m.

A one-hop traffic flow was set up on each edge of the grid in both directions, for a total of 360 saturated flows. Each flow consisted of a Poisson stream of packets generated directly at the IP layer, i.e. no transport protocol was used. $Tmax$ is found from simulation as *the maximum traffic that can be carried simultaneously on each link while maintaining* 10% *packet loss rate*. Figure 8 shows the variation of the aggregate throughput metric $Tmax$ as a function of the carrier sense range Rcs.

The maximum value that $Tmax$ attains in Figure 8 is 104 kbps when Rcs is varied from 26 m to 29 m. The optimal carrier sensing range is slightly higher than the interference range, which is 24 m from the results in Figure 6. Note that the loss in throughput relative to the maximum by setting $Rcs = R_I$ is about 4%, which supports the conclusion that using $Rcs = R_I$ provides a robust initialization point for adapting to actual network conditions at run-time.

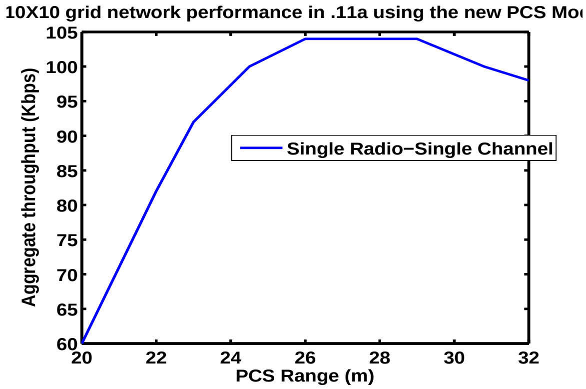

Figure 8: $Tmax$ as a function of Rcs from OPNET simulations on a 10x10 grid

5 Conclusion

Tuning of 802.11 networks using PCS requires several changes to the OPNET simulator. This paper described in detail the development of new OPNET model, and compared results with the default OPNET v.11 version. The simulation results show the performance improvement when using the enhanced model. The new code is available at: http:\\www.ee.washington.edu\research\funlab\pcs_model.htm

References

[1] S. Roy, H. Ma, R. Vijaykumar and J. Zhu, "Optimizing 802.11 Wireless Mesh Networks Performance Using Physical Carrier Sensing", (submitted), 2006. [Technical report version available at https:\\www.ee.washington.edu\techsite\papers\ documents\UWEETR-2006-0005.pdf]

[2] "Intel Pro/Wireless 2200BG Driver for Linux," http:\\ipw2200.sourceforge.net

[3] A. Kamerman and L. Montaban, "Wavelan-II: A High-Performance Wireless LAN for the Unlicensed Band," *Bell Systems Tech. J.*, Summer 1997.

[4] A. Vasan, R. Ramjee and T. Woo, "Enhanced Capacity 802.11 Hotspots," *Proc. IEEE Infocom*, Mar. 2005, Miami, FL.

[5] J. Zhu, X. Guo, L. L. Yang, W. S. Conner, and S. Roy and M. M. Hazra, "Adapting Physical Carrier Sensing to Maximize Spatial Reuse in 802.11 Mesh Networks", *Wiley J. Wireless Commun. Mob. Comput.* 2004, vol. 4, pp. 933-946.