On the Impact of Far-Away Interference on Evaluations of Wireless Multihop Networks

Douglas M. Blough*
Georgia Institute of Technology
Atlanta, US
doug.blough@gatech.edu

Claudia Canali
University of Modena and Reggio Emilia
Modena, ITALY
claudia.canali@unimore.it

Giovanni Resta
IIT-CNR
Pisa, ITALY
g.resta@iit.cnr.it

Paolo Santi
IIT-CNR
Pisa, ITALY
paolo.santi@iit.cnr.it

ABSTRACT

It is common practice in wireless multihop network evaluations to ignore interfering signals below a certain signal strength threshold. This paper investigates the thesis that this produces highly inaccurate evaluations in many cases. We start by defining a bounded version of the physical interference model, in which interference generated by transmitters located beyond a certain distance s from a receiver is ignored. We then derive a lower bound on neglected interference and show that it is approximately two orders of magnitude greater than the noise floor for typical parameter values and a surprisingly small number of nodes. We next evaluate the effect of neglected interference through extensive simulations done with a widely-used packet-level simulator (GTNetS), considering 802.11 MAC with both CBR and TCP traffic in networks of varying size and topology. The results of these simulations show very large evaluation errors when neglecting far-away interference: errors in evaluating aggregate throughput when using the default interference model reached up to 210% with 100 nodes, and errors in individual flow throughputs were far greater.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Wireless Networks

General Terms

Performance, Theory

1. INTRODUCTION

In wireless network evaluations, interference has been traditionally modeled as a binary, pair-wise phenomenon. By binary, we mean that interference either totally eliminates the ability to communicate or is non-existent, and by pairwise, we mean that interference is considered only between *pairs* of nodes or links. In reality, whether a communication is successful depends on whether signal power exceeds the sum of the interference powers plus noise by a threshold that is a property of the physical layer design. This SINR (signal to interference plus noise ratio)-based model is known as the *physical interference model* [4]. The complexity here is that interference is neither binary nor pairwise; aggregated interference from all communicating nodes must be considered to decide whether a communication is successful.

While physical interference models that account for all possible transmissions throughout the network are the most accurate, such models are very complex. As a result, several approximations are made in packet-level network simulators [12, 14, 17] to prevent simulation times from being slowed down by large factors. All major simulators (ns2, GTNetS, and Glomosim/QualNet) ignore individual interference contributions below a certain threshold (this can be implemented in different ways but can be thought of as an "interference range" beyond which interference is ignored). Inside the considered interference range, the simulators either accumulate all interference (Glomosim/QualNet) or consider only single interferers for a given transmission (ns2 and GTNetS). One of the main goals of this paper is to study the effects of these approximate interference models and to evaluate how much slowdown in simulation execution can be expected if more accurate models are used.

First, we concentrate on analytically evaluating the impact of the limited interference range assumption. We prove that, if the interference range is set to be a constant (independent of the number of nodes n), the neglected interference is large enough to cause significant errors in the accuracy of the model. This corresponds to the standard assumption in the literature and in all network simulators we are aware of. For a constant transmitter density, we also prove that, if the interference range is an arbitrary unbounded increasing function of n, the neglected interference vanishes as $n \to \infty$, meaning that the approximate interference model approaches the accuracy of the true interference model asymptotically.

Next, we present simulation results from a packet-level simulator (GTNetS), which we modified to accumulate interference inside the interference range, while keeping the

---

*D.Blough was supported in part by NSF Grant CNS-0721596.

MSWiM'09, October 26–29, 2009, Tenerife, Canary Islands, Spain.
Copyright 2009 ACM 978-1-60558-616-9/09/10 ...$10.00.

exact range as a tunable parameter. This allowed us both to validate our analytical results and to investigate trade-offs between simulation accuracy and simulation time by varying the interference range. The results demonstrate very large errors in evaluation of both packet delivery time and throughput for the commonly used interference models under several scenarios with 802.11 MAC and both CBR and TCP flows. Errors are considerable for CBR flows and even larger for TCP flows, reaching 200% in some cases. In terms of simulation time, using the most accurate model *does* have a significant negative impact. There is a roughly two orders of magnitude increase in simulation time for CBR flows and a roughly one order of magnitude increase for TCP flows.

2. RELATED WORK AND CONTRIBUTION

Several wireless interference models have been studied, ranging from pairwise models such as graph-based models [15] and the protocol model [4], to more complex and accurate models such as the physical [4] and generalized physical [9] interference models. A problem in using models such as physical and generalized physical is that interference is not confined within a bounded region around the transmitter, but it extends throughout the network, possibly corrupting message reception at far-away receivers.

Several studies have investigated the effects of different physical layer implementations on wireless network simulation accuracy [2, 6, 16]. In particular, in [16] Takai, et al. carried out a detailed analysis and comparison of the physical layer implementations of three commonly used wireless network simulators (ns2, GloMoSim, and OpNet). Among other things, they analyzed the relative impact of different physical layer sub-models (interference modeling, signal reception model, fading, path loss, etc.) on simulation results, and concluded that interference modeling is the sub-model with the strongest impact on simulation results.

Other research has focused on improving accuracy of ns2 [10]. In particular, considerable efforts have been made in the vehicular networking community to extend the ns2 design to incorporate an accurate physical interference model, which is however tailored to vehicular scenarios [3]. This work considers an "unbounded" interference model, however it does not specifically evaluate the effects of choosing this model. Effects of interference modeling on wireless network simulation have been investigated also in [5, 7], which, however, are focused on ad hoc simulators developed by the authors. Furthermore, [7] considers only "bounded" interference models, while in [5] simulations are limited to the 802.15.4 protocol and do not model the MAC layer, hence they do no consider important MAC-level issues that may have a significant impact on wireless network performance.

To the best of our knowledge, none of the previous work has investigated the issue of using "bounded" vs. "unbounded" interference models in widely used and existing simulators for wireless networks, and the effects on simulation accuracy and running time, which are the main focus of this paper. We emphasize that we are concerned with interference modeling only, and not with other physical layer aspects such as packet reception model (SINR- vs. BER-based), packet preamble capturing model, and so on. Our choice is motivated by the fact that previous work [3, 16] consistently found that the interference model is the PHY layer submodel which has the greatest impact on simulation results.

3. A BOUNDED PHYSICAL INTERFERENCE MODEL

In the physical interference model [4], successful reception of a packet sent by node u and destined to node v depends on the SINR at v. Denoting by $P_v(x)$ the received power at v of the signal transmitted by node x, a packet along link (u, v) is correctly received if and only if:

$$\frac{P_v(u)}{N + \sum_{w \in V' - \{u\}} P_v(w)} \geq \beta \ , \tag{1}$$

where N is the background noise, V' is the subset of nodes in V that are transmitting simultaneously, and β is a constant threshold (called the *SINR threshold* or *packet capture threshold*) that depends on the desired data rate, the modulation scheme, etc.

In the bounded physical interference (BPI) model considered herein, we consider only concurrent transmissions within a given region enclosing the receiver of a particular transmission when computing the SINR. For simplicity, in the following we assume that this region is a disk with a given radius, which we call the *interference range* (extension to arbitrary shapes of the enclosing region is straightforward). We refer to the total interference from outside the interference region as *far-away interference*.

More formally, the BPI model is defined as follows:

Definition 1. *A packet sent along link (u, v) is correctly received in the BPI model with interference range r if and only if*

$$\frac{P_v(u)}{N + \sum_{w \in V'_r(v) - \{u\}} P_v(w)} \geq \beta \ ,$$

where $V'_r(v)$ is the set of concurrent transmitters within distance r from v.

Note that both the BPI and the true physical interference models assume that interference from different transmitters on a given receiver is additive. Recent work has shown that, in some environments, interference is indeed additive [11].

4. ANALYSIS

In this section, we formally prove that, under some conditions on transmitter deployment, the aggregate far-away interference converges to 0 as $n \to \infty$. Due to lack of space, the results are presented without proofs, which are reported in the full version of the paper [1]. The results are valid under the two following assumptions: $a1$. Radio signal propagation obeys the log-distance path loss model, with path loss exponent $\alpha > 2$; and, $a2$. All nodes use the same transmit power P, which is such that the resulting transmission range r (in absence of interference) is at least 1.

Theorem 1. *Assume a constant density of transmitter nodes, and let u be an arbitrary node in the network which is at the receiver end of a communication link; the interference generated by nodes located at distance $d > s$ from u, where $s \geq 2r$ is the interference range, is upper bounded by*

$$C(\alpha) = \frac{\pi \rho P}{s^{\alpha-2}} \cdot \frac{1}{4 - 5\left(\frac{4}{5}\right)^{\frac{\alpha}{2}}} \ ,$$

where ρ is an upper bound to the transmitter density per unit area.

Corollary 1. *If the interference range s is chosen in such a way that $s = f(n)$, where $f(n)$ is an arbitrary unbounded increasing function of n, then the total interference at an arbitrary receiver node u due to nodes located at distance greater than s from u converges to 0 as $n \to \infty$.*

In [1], it is shown that condition $s = f(n)$ is not only sufficient, but also necessary for asymptotic accuracy of the BPI model. In particular, it is shown that if s is set to an arbitrary constant, aggregate far-away interference is order of magnitude larger than the noise floor (e.g., at least $-55dBm$ for typical 802.11a/b/g parameter settings). Furthermore, it is shown that similar results hold when nodes are distributed uniformly at random in the unit square.

5. SIMULATIONS

5.1 Simulation setup

To evaluate different interference models, we use the packet-level simulator GTNetS [14]. We start describing the basic GTNetS interference model, and then the modifications that we carried out to implement the true physical interference model.

5.1.1 Basic GTNetS interference model

The GTNetS simulator uses an approximate, bounded interference model where interference is considered only within a limited interference range (IR). Furthermore, GTNetS, similarly to *ns*2, takes into account only one interferer at a time among the nodes that are inside the IR, instead of accumulating all the interferences of concurrent transmissions. This pairwise interference calculation is equivalent to considering only the *maximum* interferer among the nodes transmitting inside the IR.

In GTNetS, a wireless link, which can be in one of the following states, is associated to each node:
IDLE: the node is not involved in any transmission or reception;
TX: the node is transmitting data;
RX: the node is receiving valid data;
CX: the node is receiving data that are not valid due to a collision occurred during the reception.

The 802.11 physical carrier sensing is implemented in GTNetS by testing whether the wireless link is in IDLE state (channel free) or not (channel busy).

Whenever a node x has to transmit a packet, it checks the state of its wireless link; if IDLE, the node starts the transmission phase. All nodes outside the IR of x ignore the interference due to the transmission. On the other hand, for each node u within the IR, the state of the wireless link might change depending on the distance d between u and x.

If u is outside the carrier sensing range of node x, i.e. the range up to which a busy channel can be detected, no action is taken. On the other hand, for each node u inside the carrier sensing range, the simulator computes $P_u(x)$, and the instant $t_u(x)$ when the reception of the signal transmitted by x will end ($t_u(x)$ is calculated based on distance d, radio signal propagation time, and packet length). A reception event is scheduled at time $t_u(x)$ for the node u. If u is outside the *transmission* range of node x, an error flag will be set to indicate that the packet can not be correctly received.

The subsequent actions are determined by the current link state of node u. If the link is IDLE, the link state is changed to RX, and at the time $t_u(x)$ the link will be set to IDLE again. If the link is in RX or CX state, the node u is currently receiving a packet from another transmitting node y. In this case, the ratio of the received powers at u ($P_u(y)/P_u(x)$) is compared with the *SINR threshold* β to determine whether the transmission from x may invalidate the ongoing transmission from y. If $P_u(y)/P_u(x) < \beta$, the link state of node u is changed to CX to indicate that a collision occurred and the packet reception is not valid. The link state of node u will be set to IDLE again at the end of the transmission that lasts longer between the ongoing transmission from y and the transmission from x.

When a node handles a reception event, it considers the received packet as valid if the packet error flag is false and the wireless link state is not CX.

5.1.2 Modifications to GTNetS

In contrast with the basic GTNetS design, our modified GTNetS version do not rely on the notion of wireless link state to detect channel conditions and correctness of data transmission.

For each node u of the network, we implement a data structure $Interf(u)$ that keeps track of the cumulative interference generated by *all* simultaneously transmitting nodes within the IR of u. When node x starts a transmission, three values are computed for each node u inside the IR of x: $P_u(x)$; $t1_u(x)$ and $t2_u(x)$, that are the times indicating the beginning and the end of the reception at u of the signal transmitted by x, respectively. These values are recorded in $Interf(u)$. Then, a reception event is scheduled for node u at time $t2_u(x)$; in the reception event, the initial time of the reception $t1_u(x)$ is also recorded. Finally, the error flag of the transmitted packet is set to true if u is outside the transmission range of x.

When node u handles a reception event, it determines the validity of the received packet based on both the value of the error flag and the information about the cumulative interference recorded in $Interf(u)$. The interferences caused by simultaneous transmissions that occur during the reception of the packet are added up using $t1$ and $t2$ times to calculate the cumulative interference $cumInterf(u)$ at the receiver u, that is considered for SINR computation. A packet transmitted by node x is correctly received at node u if the SINR condition is satisfied during the *entire* packet reception time.

Cumulative interference is used also to implement physical carrier sensing at the transmitter. When a node x has data to transmit, it compares the carrier sensing threshold with the cumulative interference $cumInterf(x)$ due to all the ongoing transmissions from nodes inside the IR of x. A value of $cumInterf(x)$ lower than the threshold means that the channel is idle, and x starts the transmission phase.

It is worth noting that in our extended version of GTNetS each node u inside the IR of a node x will be affected by the interference caused by the transmission of x even if the distance d between u and x exceeds the carrier sensing range. On the other hand, in the basic version of GTNetS all the nodes outside the carrier sensing range of a transmitting node will ignore the interference, even if they are inside the IR.

Summarizing, the main features of the extended GTNetS version, which will be made available for download, are: *i*) accumulated interference calculation within the interference range, and *ii*) tunable interference range, which allows us to mimic the true physical interference model.

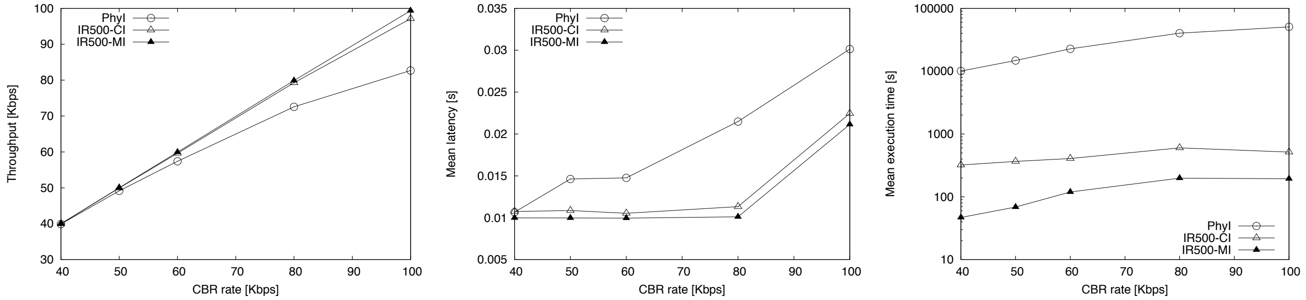

Figure 1: Throughput (left), delivery time (center), and simulation running time (right) for CBR traffic with $n = 100$ and varying packet generation rate in the grid topology.

5.1.3 Simulation methodology

We test the basic GTNetS interference model, denoted by *IR500-MI*, which considers only the maximum interferer within the interference range. We compared the basic GTNetS version with our implementation of the physical interference model, which takes into account the cumulative interference from all transmitters within the IR. We consider a bounded model with IR set to $500m$, denoted by *IR500-CI* (which resembles GloMoSim interference model), and the true physical interference model, denoted by *PhyI*, with unlimited IR. In general, we use notation IRXX-MI (IRXX-CI) to denote the interference model with IR equal to XXm, where only the maximum interferer (cumulative interference) is accounted for, and PhyI for the true physical interference model.

For our experiments, we consider two network topologies:
– *grid*: n nodes are located in a square, regularly-spaced grid; the distance between two adjacent nodes is $200m$;
– *random uniform*: n nodes are distributed uniformly at random in a square area of side $s = \sqrt{n} \times 100m$.

For space reasons, we report only results for the grid scenario. Results for the random scenario, which showed similar trends, are reported in the full version of the paper [1].

We evaluate the interference models in presence of two types of data traffic: CBR and TCP. In both cases, we use $\sqrt{n}$ flows to generate traffic, with pairs of sources and destinations randomly chosen among the n nodes in the network. We ensure that a node can not be a source or a destination for more than one flow.

Nodes are equipped with 802.11b radios, and radio signal propagation obeys the log-distance path loss model. For routing packets between far-away nodes we use the DSR routing algorithm [8]. The parameters used in our simulations are shown in Table 1. The simulated time interval is 600 seconds, which is high enough to ensure that steady-state conditions are reached in the considered network scenarios. For each simulation run, we began data collection only after 30 seconds to avoid transient effects due to initially empty routing tables. Results refer to data averaged over 15 runs. Simulations were run on Dual Intel Xeon 2Ghz machines with 2Gb of RAM.

We evaluate the interference models in terms of:
– *Throughput*: the total amount of data successfully delivered to destinations;
– *Packet delivery time*: the elapsed time between packet generation at the source and packet arrival at the destination (only for CBR traffic);
– *Simulation running time*: the elapsed real time between the start and the termination of the simulation run.

We define *error* to be the percentage difference between one of the performance measures under an approximate interference model and the same measure under the PhyI model.

Table 1: Simulation parameters

Link data rate	$11Mbps$
Transmission power	$100mW$
Transmission range	$250m$
Carrier sensing range	$500m$
Carrier sensing threshold	$-48dBm$
Path loss exponent	2.5
Background noise	$-90dBm$
SINR threshold	$10dB$

5.2 Results for CBR traffic

In this set of simulations, we have randomly created CBR flows and evaluated performance in terms of throughput and packet delivery time. The results for IR500-MI, IR500-CI, and PhyI for the grid topology with 100 nodes and varying rates of the CBR flows are reported in Figure 1. The figure reports also the average running time of a single simulation run (note the log-scale used in the y axis).

Results reported in Figure 1 clearly show that, while not accumulating interference within the shorter interference range has only a marginal effect on throughput and delivery time estimation, ignoring far-away interference has a dramatic effect on performance estimation: while the network appears to operate below saturation under the IR500-MI/CI models even for the highest data rate of $100Kbps$, it has already reached saturation at $80Kbps$ under the PhyI model. At a data rate of $100Kbps$, the IR500-MI model estimates a throughput above $99Kbps$, as compared to a throughput slightly above $82Kbps$ with the PhyI model, for an error of about 20%. The situation is even worse for the average delivery time, which is consistently underestimated by both the IR500-MI and the IR500-CI models (relative error as high as 60% at $80Kbps$). Note that the relatively high error in estimating delivery time is due to the fact that more accurate interference modeling tends to increase not only the number of dropped packets due to low SINR values (which affects both throughput and delivery time), but also the average channel access time. In fact, a higher interference level in the network tends to result in a higher frequency of busy channel detection at the transmitter nodes.

Unfortunately, the price to pay for accurately modeling wireless interference is a dramatic increase in the simulation running time, which increases by more than two orders of magnitude when using the PhyI model instead of the IR500-MI model. This increase in running time is due to the fact that, as the interference range increases, a larger number of events (one for every receiver in the transmitter's interference range) must be scheduled for a single transmission.

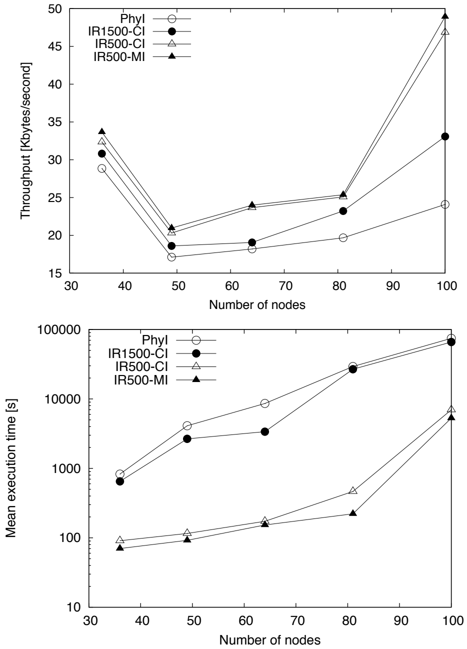

Figure 2: Average throughput (top), and simulation running time (bottom) for TCP traffic with varying network size.

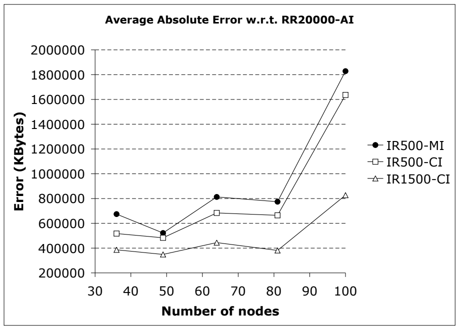

Figure 3: Average absolute error in throughput estimation with varying network size.

In another set of experiments we have fixed the CBR rate to $100Kbs$, and varied the number of nodes from 36 to 144. The results, not reported for lack of space (see [1]), have shown a decreasing accuracy of the IR500-MI/CI models for increasing number of nodes, thus validating our theoretical findings. In particular, accuracy of IR500-MI/CI models start to significantly degrade for values of $n \geq 64$. For what concerns simulation running time, also in this case we observe an up to two orders of magnitude increase when using the PhyI model instead of the IR500-MI model.

5.3 Results for TCP traffic

In the second set of experiments, we consider $\sqrt{n}$ TCP flows randomly created in a the grid topology with n nodes. Here, we focused on evaluating throughput and average simulation running time.

Figure 2 reports the simulation results for networks of varying size. As with CBR traffic, the IR500-MI model consistently tends to overestimate average throughput. Errors are even higher than for CBR (up to 210% for $n = 100$). Note that the average throughput of a TCP flow has a non-monotonic behavior with increasing network size. This behavior, which is consistently observed with all the interference models, can be explained as follows. With $n = 36$, we have relatively shorter TCP flows, and there was at least one one-hop flow in every simulated instance. Since one-hop TCP flows are not subject to intra-flow interference, they typically experience a much higher throughput than multihop flows, which explains the relatively high average throughput observed for $n = 36$. For larger values of n, no one-hop TCP flows occurred in the simulated instances, which explains the drop in throughput when passing from $n = 36$ to $n = 49$. After that value of n, the average throughput has an increasing trend, due to the fact that the number of TCP flows grows sub-linearly with n (we have $\sqrt{n}$ flows with n nodes), implying a lower network congestion level observed by each TCP flow as n increases.

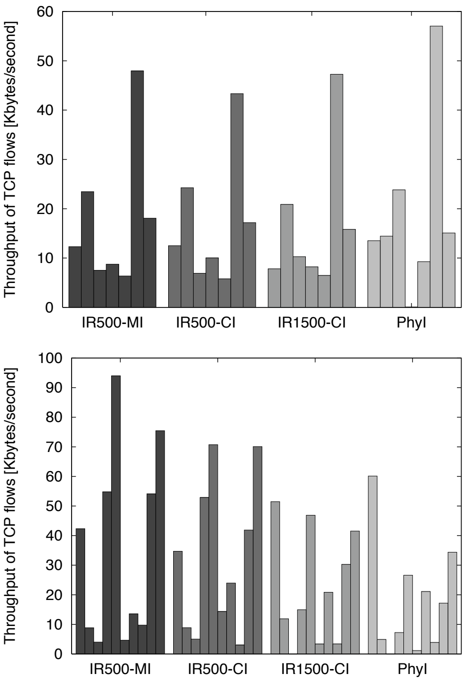

Figure 4: Disaggregated TCP throughput for a scenario with minimal (top) and maximal (bottom) average absolute error.

Compared to the CBR case, the increase in simulation running time using the PhyI model instead of the IR500-MI model is less dramatic: for the largest network size, the relative increase is about *one* order of magnitude. This relatively less dramatic increase is due to the congestion control TCP mechanism, which tends to reduce the number of transmitted packets (and, consequently, running time) as more congested network conditions are encountered.[1] Combining the fact that relatively less TCP packets are sent with the longer processing time of each packet, we have an overall increase of about one order of magnitude in simulation running time.

The statistics about the individual TCP flows are even more interesting than the aggregate data of $\sqrt{n}$ flows reported in Figure 2. Figure 3 reports the average *absolute* error (in KB) between the amount of data sent by each TCP flow as estimated by the approximate interference models and as estimated by the PhyI model. The average absolute

---

[1] This is in sharp contrast with the case of CBR traffic, where the number of generated packets *does not* depend on the network congestion level.

error (a.a.e.) is computed as follows:

$$\frac{\sum_{i=1,\ldots,\sqrt{n}} |Thr_{x-Y}(i) - Thr(i)|}{\sqrt{n}} ,$$

where $Thr_{x-Y}(i)$ is the total amount of data sent by the i-th TCP flow as estimated by model RRx-Y, and $Thr(i)$ is the same statistic for the PhyI model.

From Figure 3, we see that the a.a.e. tends to increase for growing values of n, although a sharp increase can be observed only for n larger than 80. As for the relative behavior of the considered interference models, we observe that IR500-CI reduces a.a.e. up to 24% w.r.t. IR500-MI, while IR1500-CI can reduce a.a.e. by as much as 65% w.r.t. IR500-MI. However, we recall (see Figure 2) that simulations with the IR1500-CI model have a running time comparable to that under the accurate PhyI interference model.

The choice of computing the absolute instead of percentage error is due to the fact that, for some simulated TCP flows, the percentage error can be extremely large. This occurs when a certain TCP flow is estimated to have a certain, non-negligible throughput under model IRx-Y, while under the PhyI model it is starved. As shown by the disaggregated flow data in Figure 4, this situation occurs frequently. More specifically, Figure 4 reports the disaggregated statistics of the amount of data sent by individual TCP flows for two specific simulation instances, referring respectively to the simulation runs with minimal a.a.e. (top) and maximal a.a.e. (bottom) for the IR500-MI model. It is interesting to observe that, even in the relatively "more accurate" simulation run, there exists a flow (the 4th bar from the left in each interference model) that is consistently estimated as carrying a non-negligible amount of data by models IR500-MI/CI and IR1500-CI, while it was starved under the PhyI model. Thus, the percentage error for this specific flow is enormous for all the approximate interference models. Another interesting observation is that *error in estimating throughput of individual TCP flows with approximate interference models is not uni-directional*. In fact, the estimated throughput of a TCP flow with the PhyI model may be considerably *higher* than the ones estimated with approximate interference models. This is the case, for instance, for flows 1, 5, and 6 in the top part of Figure 4, and for flow 1 in the bottom part. Summarizing, accuracy of individual TCP flow's statistics with approximate interference models is very low due to: (i) incorrect prediction of starved flows, leading to virtually unlimited percentage error for such flows, and (ii) the bi-directional nature of the errors. Note that (i) and (ii) occur even in relatively "accurate" simulation runs, i.e., those corresponding to the lowest a.a.e. among the simulated scenarios.

6. DISCUSSION AND CONCLUSIONS

The main contribution of this paper is an in-depth investigation of the accuracy of approximate, bounded interference models in estimating relevant performance metrics of wireless multihop networks, and of the tradeoff between simulation accuracy and running time. Summarizing, our results suggest that ignoring far-away interference leads to highly inaccurate simulation results, but the price to pay for this accuracy is a one-two orders of magnitude increase in simulation running time.

Overall, the issue of how to address the fundamental tradeoff between simulation accuracy and running time is the main problem left open by this paper. We believe optimally addressing this tradeoff is a very difficult task, since it depends on the specific network setting at hand, the type of transport-layer traffic, and so on. A promising research direction is deriving analytical/stastical methods to accurately estimate aggregate interference generated by far-away interferers, so that relatively accurate estimates of the SINR values at single receivers can be obtained without actually computing each far-away interference signal level. An interesting first step in this direction is [13], where the authors introduce a Markov-chain based interference model which is fed by testbed measurement. While the idea of measurement driven interference model is interesting, the approach of [13] considers only 1-hop flows, and is based on Markov chains with a number of states exponential in the number of nodes (and is thus not scalable). Hence, further work is needed to derive scalable approaches that account also for multi-hop flows.

7. REFERENCES

[1] D. Blough, C. Canali, G. Resta, P. Santi, "On the Impact of Far-Away Interference on Evaluations of Wireless Multihop Networks", *Tech. Rep. IIT-TR-03-2008*, IIT-CNR, Pisa - Italy,2008.

[2] D. Cavin, Y. Sasson, and A. Schiper, "On the accuracy of MANET simulators", *Proc. ACM POMC*, pp. 38–43, 2002.

[3] Q. Chen et al., "Overhaul of IEEE 802.11 Modeling and Simulation in NS-2", *Proc. ACM MSWiM*, 2007.

[4] P. Gupta and P.R. Kumar, "The Capacity of Wireless Networks," *IEEE Transactions on Information Theory,* Vol. 46, No. 2, pp. 388–404, 2000.

[5] E. Hamida, G. Chelius, and J.-M. Gorce, "Scalable versus Accurate Physical Layer Modeling in Wireless Network Simulations", *Proc. PADS*, pp. 127–134, 2008.

[6] J. Heidemann et al., "Effects of Detail in Wireless Network Simulation", *Proc. SCS Multiconference on Distributed Simulation*, pp. 3–11, 2001

[7] A. Iyer, C. Rosenberg, A. Karnik, "What is the Right Model for Wireless Channel Interference?", *Proc. ACM QShine*, 2006.

[8] D.B. Johnson, D.A. Maltz, "Dynamic Source Routing in Ad Hod Wireless Networks", *Mobile Computing*, n. 353, pp. 153–181, 1996.

[9] A. Keshavarz-Haddad, R. Riedi, "On the Broadcast Capacity of Multihop Wireless Networks: Interplay of Power, Density and Interference", *Proc. IEEE SECON*, pp. 314–323, 2007.

[10] D. Kotz et al., "Experimental Evaluation of Wireless Simulation Assumptions", *Proc. ACM MSWiM*, 2004.

[11] R. Maheshwari, S. Jain, S. Das, "A Measurement Study of Interference Modeling and Scheduling in Low-Power Wireless Networks," *Proc. ACM SenSys*, pp. 141–154, 2008.

[12] `http://www.isi.edu/nsnam/ns2/`

[13] L. Qiu, Y. Zhang, F. Wang, M. Han, and R. Mahajan, "A general model of wireless interference", *Proc. ACM MobiCom*, pp. 171–182, 2007.

[14] G. Riley, "The Georgia Tech Network Simulator," *ACM SIGCOMM MoMeTools Workshop,* 2003.

[15] G. Sharma, R.R. Mazumdar, and N.B.Shroff, "On the Complexity of Scheduling in Wireless Networks", *Proc. ACM Mobicom*, 2006.

[16] M. Takai, J. Martin, R. Bagrodia, "Effects of Wireless Physical Layer Modeling in Mobile Ad Hoc Networks", *Proc. ACM MobiHoc*, pp. 87–94, 2001.

[17] X. Zeng, R. Bagrodia, and M. Gerla, "GloMoSim: a Library for Parallel Simulation of Large-scale Wireless Networks", *Proc. Workshop on Parallel and Distributed Simulations,* (PADS), 1998.