TCP performance in mobile wireless environment: channel modelling and network simulation

Pavel V. Nikitin
Department of Electrical and Computer Engineering
Carnegie Mellon University, Pittsburgh, PA 15213, USA
email: pnikitin@andrew.cmu.edu

Onur Celebioglu
Dell Computer Corporation, One Dell Way
Round Rock, TX 78682, USA
email: onur_celebioglu@dell.com

and

Volkan Kukrer
Spinnaker Networks, 301 Alpha Drive
Pittsburgh, PA 15237, USA
email: kukrer@spinnakernet.com

ABSTRACT

In this paper, we explore TCP performance over a two-node mobile wireless link in five realistic scenarios with different propagation environments. We use a simple method to combine a physical layer modelling and a network simulation using the network simulator *ns-2*. We present the results of the propagation channel simulation and the results of the corresponding network simulation using TCP Tahoe protocol. Our work demonstrates that in order to improve TCP performance over wireless links, one needs not only to tune the parameters of TCP but also to modify the TCP semantics.

Keywords

TCP, performance analysis, network simulation, physical layer, mobile networks, wireless networks.

I. INTRODUCTION AND BACKGROUND

TCP/IP is the most widely used protocol stack in today's world of computing, with many applications readily available for it. TCP was first introduced for wired links. Its parameters and semantics were carefully tuned to maximize its performance on wired networks, where packet delays and losses are caused mainly by congestion, whereas on wireless networks they are also caused by mobility handoffs and transmission errors. With the recent developments in mobile wireless networking, TCP performance in mobile wireless environment is becoming a topic of interest.

TCP is a complicated collection of different algorithms, each having a few variable parameters. The protocol performance in any network depends on physical channel characteristics, traffic pattern, network size, etc. Improving TCP performance in wireless networks was addressed by several authors. A good overview of proposed techniques can be found in [1], [2]. Both radical changes in TCP semantics as well as slight modifications of parameters were proposed [3], [4], [5], [6], [7], [8]. For example, [8] concentrates on dealing with effect of mobility handoffs on TCP performance in the absence of transmission errors.

Accurate simulation of a physical layer in wireless networks is important for evaluating and optimizing the performance of routing protocols [9], [10]. A popular network simulator is *ns-2* [11]. It was originally developed by Lawrence Berkeley National Laboratory [12]. There are a few extensions for *ns-2* that account for physical layer effects [9], [13], [14], [15]. There are also other network simulators with a similar capability [16], but integrating physical layer effects into a network simulation is usually a complicated procedure. On the other hand, detailed wireless channel simulations [17], [18], [19], [20] have traditionally been done independently from the transport protocol performance research.

In this paper, we use a simple way to combine an accurate independent propagation channel modelling and a network simulation with *ns-2*. We focus on the performance impairment caused by wireless transmission errors due to the low signal strength at the receiver. Mobility handoffs are not present in our two-node peer-to-peer connection.

The remainder of the paper is organized as follows. Section II describes our simulation technique and scenarios. Simulation results are presented in Section III. Section IV contains the discussion. Conclusions are given in Section V.

II. SIMULATION

There are a few ways to incorporate wireless channel effects into the network simulator *ns-2*. One way is to look at each received packet individually, estimate its signal strength using physical channel model, and decide whether the packet should be kept or discarded [9]. This method allows to simulate capture effects and collisions, which are important in multiple node networks. It requires significant modifications to the existing base version of the network simulator *ns-2*, and physical layer modelling has to be done concurrently with the network simulation. The other method, used here, is to artificially interrupt the link between two nodes if the signal strength is below the receive threshold [14]. The advantage of this approach is simplicity. It does not require any modifications to *ns-2* and the physical channel modelling can be done beforehand, independently from the network simulation.

Our simulation process is illustrated in Figure 1. It involved the scenario generator, the channel modeler, and the network simulator. One-way TCP was used, where one node was transmitting and the other node was receiving. The scenario generator created a specific movement pattern of mobile nodes and other objects in a specific propagation environment. The channel modeler computed the signal strength at the receiving node as a function of time. Whenever the computed signal strength was below the receive threshold, the link was considered to be down. This temporal information was fed into the network simulator *ns-2*, which simulated a data exchange between the nodes and generated a datafile with an information about the TCP behavior ($ns-2$ tracefile). Packets still on the air, when the link came down, were considered to be lost.

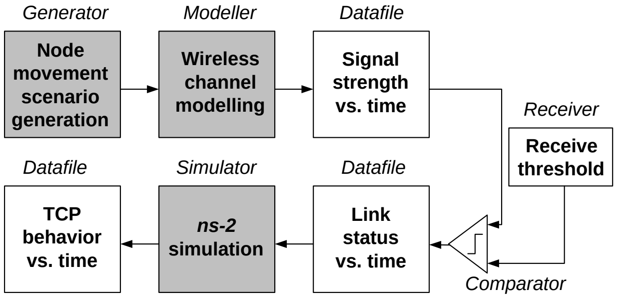

Fig. 1. Block diagram of simulation process: physical layer modelling combined with *ns-2* network simulation.

A. Scenario generation

Scenario generator created a propagation environment, which the communication takes place in. Each scenario was characterized by the specific movement pattern of mobile nodes and other objects (cars). Four outdoor scenarios and one indoor scenario were considered. In all cases, we assumed a flat terrain, a clear line of sight between the transmitter and the receiver, and modelled buildings and cars as rectangular-shaped objects.

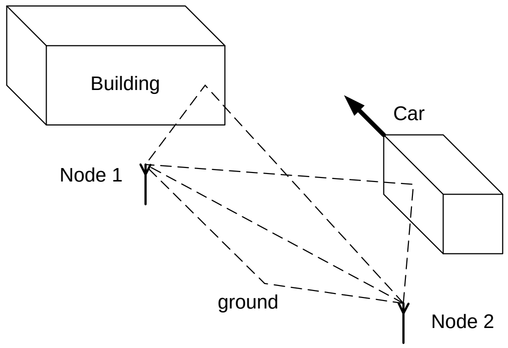

Fig. 2. Illustration of the propagation model: dashed lines show a direct ray, a ground-reflected ray, and rays reflected from the building and from the car.

B. Channel modelling

The frequency of 915 MHz (a center frequency of 902-928 MHz ISM band) was used. Both node antennas were considered to be omnidirectional, with a unity gain, positioned 1.5 meters above the ground. For simplicity, the transmitter power was assumed to be 0 dBm. Ray-tracing approach was used to predict the signal strength as a function of time. The deterministic propagation model took into account a direct ray, a ground-reflected ray, and rays reflected off other objects (buildings or cars) as shown in Figure 2.

The total power received is described by the following expression:

$$P_r = P_t \left(\frac{\lambda}{4\pi}\right)^2 \left| \frac{e^{-j\frac{2\pi}{\lambda}d}}{d} + \sum_{n=1}^{N} \Gamma_n \, \frac{e^{-j\frac{2\pi}{\lambda}d_n}}{d_n} \right|^2 , \qquad (1)$$

where P_t is the transmitted power, λ is the wavelength, d is the length of the direct ray path, Γ_n is the reflection coefficient of the n-th reflecting object (including ground), d_n is the length of the n-th reflected ray path, and N is the total number of reflected rays. Only single reflections are considered, and the reflection coefficient of all reflecting objects (ground, car, building, walls) was assumed to be $\Gamma_n = -1$.

Interference of reflected rays results in a multipath fading and causes enough signal variation to make our scenarios interesting. The output of the propagation model is the file containing the received signal strength as a function of time. After comparing it with the receive threshold, the datafile that contained the link status temporal behavior is produced and fed into the network simulator. In this approach, all wireless channel effects are contained in the link status, which has two possible states - up or down.

C. Network simulation

TCP Tahoe agent was selected for the network simulation using *ns-2*. Other TCP agents (Reno, Vegas, etc.) could be used as well. No network layer and no wireless MAC protocol was considered in our two-node system. The maximum capacity of the link, connecting two communicating nodes, was 2 Mbps (250 Kbps). TCP packet size was 1500 bytes, which corresponded to the packet transmission time of 6 ms. In our scenarios, the propagation delay always remained small compared to the packet transmission time and did not affect the TCP behavior.

Each network simulation run produced a tracefile containing the information about all sent and received packets. The following TCP parameters were recorded as functions of time: congestion window size, packet sequence number, packet transmission time. The throughput was calculated via moving average over the period of 100 ms (the period of a reasonable granularity, during which approximately 16 packets could potentially be transmitted).

III. RESULTS

Simulations were conducted for five different scenarios over the period of 20 seconds, which was sufficient to collect the necessary data and observe the behavior trends. A description of scenarios, along with the results, is given below.

Figures 3, 4, 5, 6, and 7, accompanying each of the five scenarios, show received signal strength, congestion window size, throughput, transmitted packet sequence number, and transmission events (a vertical line is drawn whenever a data packet is sent, which allows to illustrate a TCP exponential backoff strategy). In all cases, the slow start threshold was set to 16 packets, the initial/reset congestion window size was set to 1 packet, and the maximum bound on congestion window size was set to 64 packets (the maximum value allowed in TCP Tahoe). The congestion window never reached its maximum value because of the signal fading.

A. Scenario 1

One node is stationary (e.g., a base station), and the other node (e.g., a user with a mobile phone) is moving away from it with constant speed of 6 m/s (13.5 mph), initial node separation is 10 m. No other objects are present nearby, the only reflection comes from the ground. Power drops as d^{-2} at short and as d^{-4} at long distances.

The receive threshold was chosen to be -75 dBm. The simulation results are shown in Figure 3. In this scenario, using a constant backoff time could improve the performance. For example, the link is up at $t = 5$ s, but transmission does not start until about $t = 6$ s because the last packet has not timed out yet. Adjusting the slow start threshold value could also improve the throughput.

B. Scenario 2

One stationary node and one mobile node are initially located 10 m from the building wall and 30 m from each other. The mobile node starts moving away from the building with a constant speed of 5 m/s (11.25 mph). This scenario simulates a situation similar to two nodes communicating in front of a building, while one of the nodes is moving away from the building. In addition to a reflection from the ground, which was calculated in scenario 1, there is also a reflection component from the nearby wall. The receive threshold was chosen to be -75 dBm.

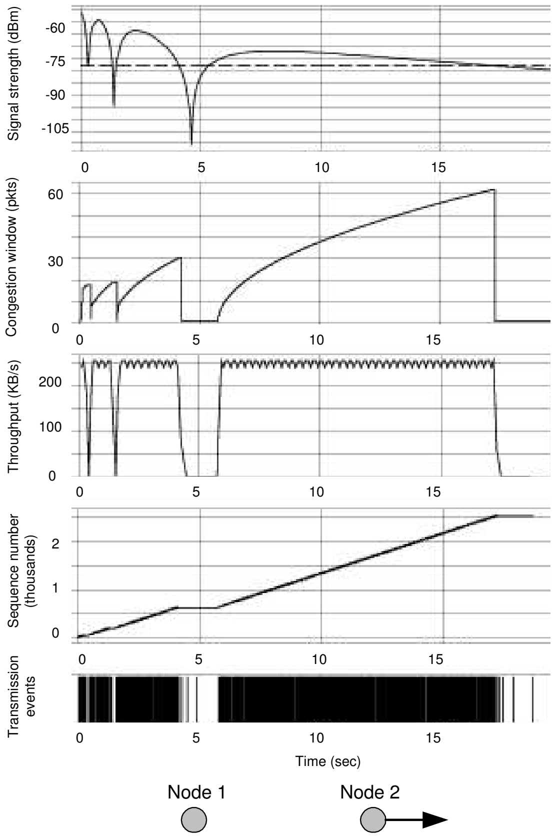

Fig. 3. Scenario 1.

The simulation results are shown in Figure 4. After about 10 s, the signal strength becomes low, which causes frequent threshold crossings and corresponding packet losses. The congestion window, which has grown constantly up to that point, drops down and can not re-grow further. However, some packet transmissions still take place during short intervals when the link is up. In such situation, keeping retransmission timer close to its minimum value could help to utilize those short time intervals when the link is up and the data can safely travel through the air. Another way to improve the performance would be to make packets shorter than the average period of signal strength oscillations. This could be done either by reducing the packet size or by increasing the transmission rate.

C. Scenario 3

Two stationary nodes sit 10 m away from the busy one-lane road, which is parallel to the line of sight between the

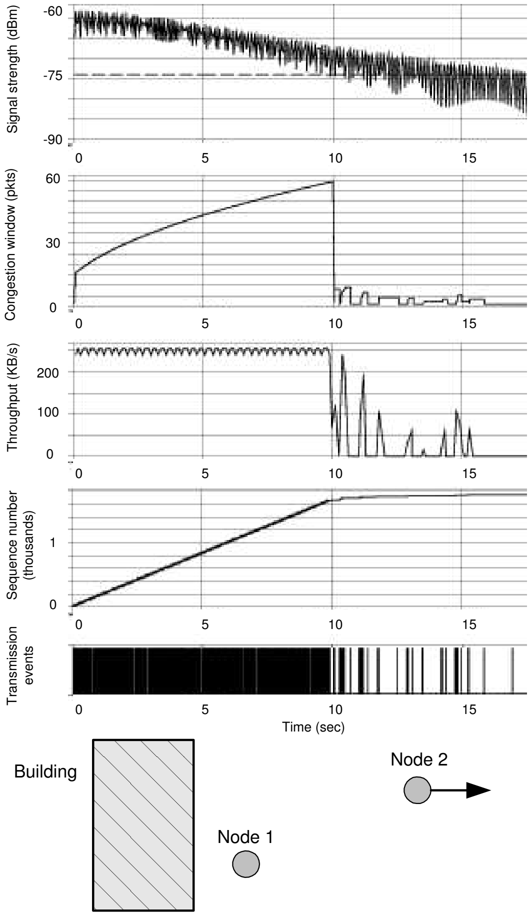

Fig. 4. Scenario 2.

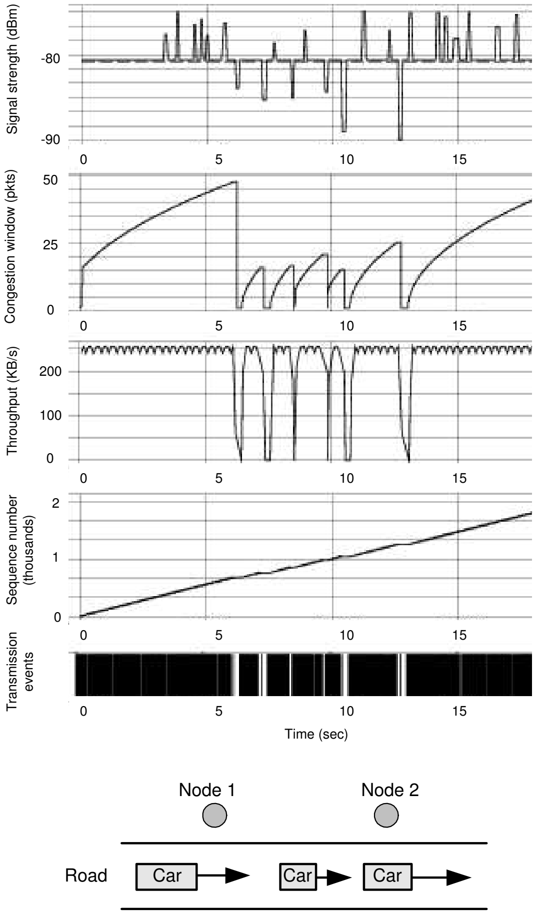

Fig. 5. Scenario 3.

nodes. The spacing between the cars, passing by on the road, randomly varies between 1 m and 30 m, the length of the cars randomly varies between 2 m and 5.6 m, and the car speed is assumed to be constant 30.2 m/s (67.5 mph). The received signal strength is barely above the chosen receive threshold of -80 dBm in the absence of any vehicles. Waves, reflecting from the cars, can interfere either constructively or destructively, favoring or impairing the transmission.

The simulation results are shown in Figure 5. The linear plot of sequence numbers shows that TCP performs quite good in this case, even though there are several delays and retransmissions. Changing the retransmission timer does not change the performance because fades are short, and TCP does not have time to backoff a lot. Some increase in the throughput can be achieved by resetting the window size (whenever a packet is lost) not to its minimum value (1 packet) but, e.g., to the half of the current window.

D. Scenario 4

Two cars follow each other along a centerline of a 20 m wide urban street with 50 m long buildings symmetrically located on both sides and spaced 20 m apart. Initial car separation is 100 m, the first car moves at a constant speed of 17.9 m/s (40 mph), the speed of the second car randomly varies between 15.6 m/s (35 mph) and 20.1 m/s (45 mph). Reflections come from the ground and surrounding buildings. The received signal is initially below the -78 dBm threshold, but whenever a building is passed, the reflected rays interfere constructively, allowing communication.

The simulation results are shown in Figure 6. Making the retransmission timeout smaller would help to increase the throughput in this case. For example, the link is up at $t = 9.5$ s, but transmission does not start until approximately $t = 10.5$ s.

E. Scenario 5

Two nodes (e.g., users with wireless devices), move inside an empty square room, 14 m x 14 m in size. For simplicity, nodes follow each other on a circular trajectory with a speed of 1.6 m/s (3.6 mph) and separation of 8.7 m. Reflections come from the walls and from the floor. We chose a threshold of -50 dBm to make this scenario interesting.

The results of simulation are shown in Figure 7. The distance from the nodes to the walls of the room varies, which creates an irregular temporal signal strength pattern. Due to the multipath fading, there are not many time periods when

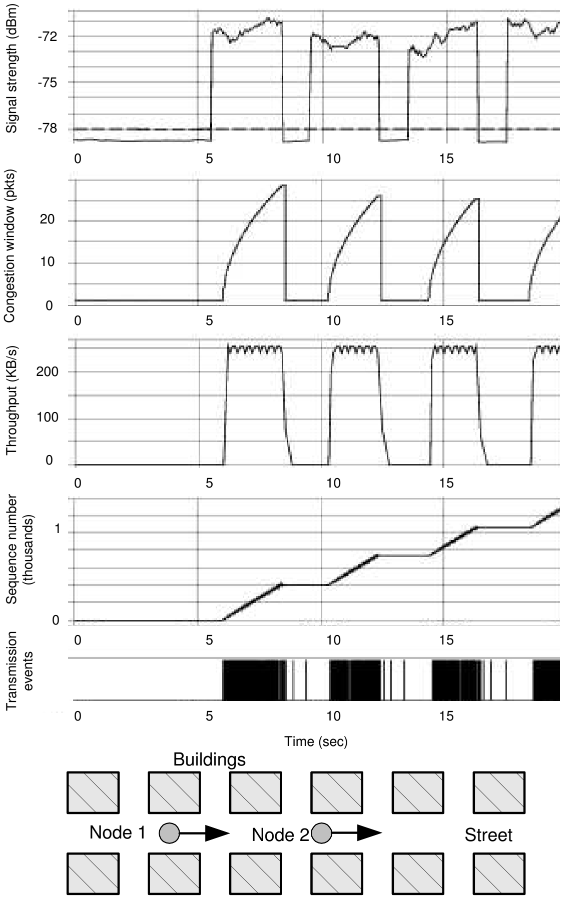

Fig. 6. Scenario 4.

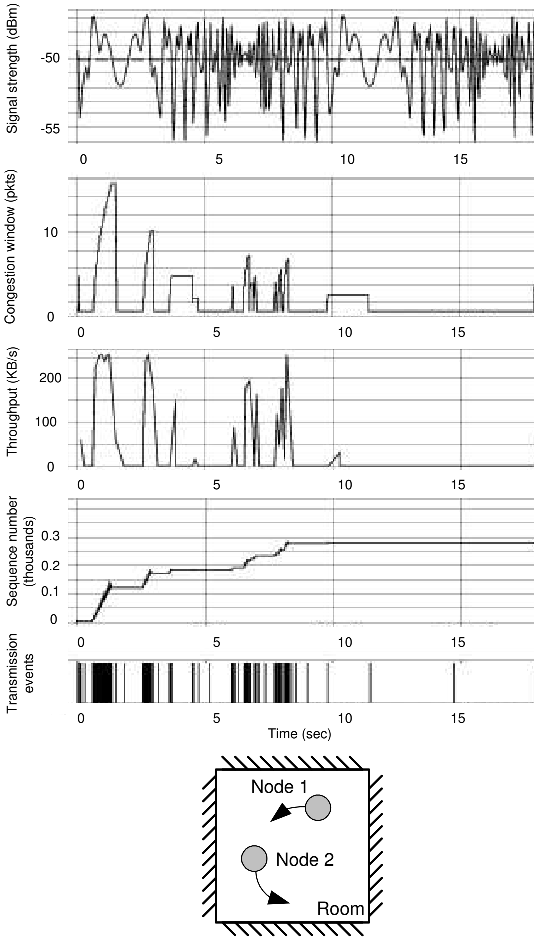

Fig. 7. Scenario 5.

the nodes are able to communicate. To improve the performance, one might consider either decreasing the retransmission timer or transmitting multiple copies of a single packet. Waiting for ACKs to arrive makes us miss those short time intervals when we could get a few packets through. Also, a short fade at the end of the long packet transmission causes the whole packet to be lost (no error correction is considered in our model). Making a packet size smaller than the average fade duration in this case would increase the throughput.

IV. DISCUSSION

Our channel behavior can be well described by a few parameters. The first parameter is the link downtime T_d: the ratio of time when the link is down to the total simulation time. The second parameter is the average number of fades N_f during the simulation time. The third parameter is the average fade duration T_f . In terms of the protocol performance, one of the parameters characterizing it is the average data throughput R. Table I gives the link downtime, the average number of fades, the average fade duration, and the average data throughput for all five scenarios.

Note that all our scenarios are quasi-static: the time scale on which the node movements happen is much larger than both the packet transmission time and the propagation delay. Also, the time needed to send out a congestion window size is much larger than the packet roundtrip time. Our observed simulation results, including the congestion window size behavior, are close to the results reported in [8], but we addressed wireless transmission errors rather than mobility handoffs.

Our physical layer modelling approach assumed that when the signal power was below the threshold, packets could not be successfully demodulated, the link was down, and the transmitted data were lost. This approach did not take into account impulse response characteristics of the wireless channel. In our scenarios, this effect was not significant since the multipath delay spread was very small (a few tens of ns) compared to the packet transmission time (a few ms).

V. CONCLUSIONS

In our work, we simulated and analyzed the TCP performance over a wireless communication link between two mobile nodes in five different scenarios. We used the packet simulator *ns-2*, which is a powerful tool for studying the network behavior. We described a simple way to model

TABLE I

LINK DOWNTIME T_d (%), AVERAGE NUMBER OF FADES N_f, AVERAGE FADE DURATION T_f (S), AND AVERAGE THROUGHPUT R (KBPS) FOR DIFFERENT SLOW START THRESHOLD VALUES SST (PACKETS) IN SCENARIOS 1, 2, 3, 4, AND 5.

Scenario	1	2	3	4	5
T_d	21	27	5	45	46
N_f	3	50	6	4	54
T_f	1.4	0.1	0.2	2.2	0.2
R (SST=8)	192	152	228	116	30
R (SST=16)	190	131	228	183	30
R (SST=32)	184	125	231	107	30

physical layer effects in a network simulation by creating an external command file, which breaks the communication link at certain time moments according to the scenario of node movement and the surrounding propagation environment. This approach allows to perform wireless channel simulation independently of network simulation and does not require any modifications to *ns-2*. The link condition is determined based on comparison of the received signal strength to the receive threshold.

Packet losses in mobile wireless environment depend on many factors that include both large-scale and small-scale fading, interference from other sources, etc. When packets are dropped because of the channel fading, TCP considers it to be a result of the congestion. The protocol backs off and shrinks the congestion window size. Frequent retransmissions are also not desirable in the presence of long fading periods and other nodes that need to use the bandwidth. The better strategy is to grow the retransmission timeout only until a certain value and keep it constant afterwards. This way we periodically try to transmit a packet, “probing” the channel to see if the link is up again.

If the behavior of a wireless communication link can be approximately predicted for a certain time period, it should be possible to maximize the throughput over this time period by optimizing TCP parameters without changing TCP semantics. However, developing an algorithm that would adaptively change TCP parameters based on channel condition information is a challenging task, and such an information may not always be available to the protocol. The best way to improve the protocol performance in a mobile wireless environment is to use specialized transport protocols, such as I-TCP [5] or Snoop TCP [21], which modify TCP semantics.

ACKNOWLEDGEMENTS

The authors are grateful to Professor David B. Johnson for valuable feedback, to Professor Daniel D. Stancil for helpful discussions, and to Ratish Punnoose, Satish Shetty, and Josh Broch for help with the network simulator *ns-2*.

REFERENCES

[1] G. Xylomenos, G. C. Polyzos, P. Mahonen, and M. Saaranen, “TCP performance issues over wireless links,” IEEE Communications Magazine, vol. 39, no. 4, pp. 52–58, April 2001.

[2] G. Huston, “TCP in a wireless world,” IEEE Internet Computing, vol. 5, no. 2, pp. 82–84, March-April 2001.

[3] H. Balakrishnan, S. Seshan, and R. H. Katz, “Efficient TCP over networks with wireless links,” Proceedings of Fifth Workshop on Hot Topics in Operating Systems, pp. 35–40, 1995.

[4] H. Balakrishnan, V. N. Padmanabhan, S. Seshan, and R. H. Katz, “A comparison of mechanisms for improving TCP performance over wireless links,” IEEE/ACM Transactions on Networking, vol. 5, no. 6, pp. 756 –769, December 1997.

[5] A. Bakre and H. R. Badrinath, “I-TCP: indirect TCP for mobile hosts,” Proceedings of the 15th International Conference on Distributed Computing Systems, pp. 136–143, 1995.

[6] K. Chandran, S. Raghunathan, S. Venkatesan, and R. Prakash, “A feedback-based scheme for improving TCP performance in ad hoc wireless networks,” IEEE Personal Communications, vol. 8, no. 1, pp. 34–39, February 2001.

[7] Fei Peng, Shiduan Cheng, and Jian Ma, “An effective way to improve TCP performance in wireless/mobile networks,” Information Systems for Enhanced Public Safety and Security, pp. 250–255, 2000.

[8] R. Caceres and L. Iftode, “Improving the performance of reliable transport protocols in mobile computing environments,” IEEE Journal on Selected Areas in Communications, vol. 13, no. 5, pp. 850–857, June 1995.

[9] Josh Broch, David A. Maltz, David B. Johnson, Yih-Chun Hu, and Jorjeta Jetcheva, “Performance comparison of multi-hop wireless ad hoc network routing protocols,” Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking, October 1998.

[10] Ratish J. Punnoose, Pavel V. Nikitin, Josh Broch, and Daniel D. Stancil, “Optimizing wireless network protocols using real-time predictive propagation modeling,” Proceedings of IEEE Radio and Wireless Conference (RAWCON), pp. 39–44, August 1999.

[11] http://www.isi.edu/nsnam/ns/.

[12] http://www nrg.ee.lbl.gov/ns/.

[13] http://www.monarch.cs.rice.edu/cmu-ns.html.

[14] A. Bonsall, D. Wang, and D. D. Stancil, “Propagation model embedded in a wireless network simulator,” Proceedings of IMPAS 3rd Annual Wireless Communications Conference, November 1998.

[15] R. J. Punnoose, P. V. Nikitin, and D. D. Stancil, “Efficient simulation of Ricean fading within a packet simulator,” Proceedings of the 52nd Vehicular Technology Conference, vol. 2, pp. 764–767, 2000.

[16] P. Mishra, M. Srivastava, P. Agrawal, and G. Nguyen, “Ethersim: a simulator for wireless and mobile networks,” Proceedings of the Global Telecommunications Conference, vol. 3, pp. 2018–2027, 1996.

[17] H. R. Anderson, “A ray-tracing propagation model for digital broadcast systems in urban areas,” IEEE Transactions on Broadcasting, vol. 39, no. 3, pp. 309–317, September 1993.

[18] V. Fung, T. S. Rappaport, and B. Thoma, “Bit error simulation for $\pi/4$ DQPSK mobile radio communications using two-ray and measurement-based impulse response models,” IEEE Journal on Selected Areas in Communications, vol. 11, no. 3, pp. 393–405, 1993.

[19] J. Espinosa, M. Montero, J. M. Hernando, F. Perez, J. A. Milanes, I. Hidalgo, and M. Gallardo, “Coverage simulator for wireless networks,” Proceedings of IEEE International Workshop on Factory Communication Systems, pp. 149–156, 1998.

[20] M. Patzold, R. Garcia, and F. Laue, “Design of high-speed simulation models for mobile fading channels by using table look-up techniques,” IEEE Transactions on Vehicular Technology, vol. 49, no. 4, pp. 1178 –1190, July 2000.

[21] J. Rendon, F. Casadevall, D. Serarols, and J. L. Faner, “Analysis of Snoop TCP protocol in GPRS system,” Electronics Letters, vol. 37, no. 10, pp. 651–652, May 2001.