Proceedings of the 2001 Winter Simulation Conference
B. A. Peters, J. S. Smith, D. J. Medeiros, and M. W. Rohrer, eds.

TOWARDS HIGH PERFORMANCE MODELING OF THE 802.11 WIRELESS PROTOCOL

Jason Liu
David M. Nicol
L. Felipe Perrone
Michael Liljenstam

Department of Computer Science &
Institute for Security Technology Studies
Dartmouth College
Hanover, NH 03755 U.S.A.

ABSTRACT

The IEEE 802.11 standard is a widely used protocol for wireless communications. It is a moderately complex algorithm involving collision detection, dynamic backoffs, channel reservations, and acknowledgments. Detailed simulation of 802.11 requires some care, and considerable execution time. We are interested in developing a rapidly executable model of 802.11's effect on network behavior. Our interest in this derives from investigations into routing algorithms for large scale ad-hoc networks, executing on parallel architectures. As our interest is in routing and not the MAC layer, a rapidly executed model of 802.11 will accelerate simulations focused on routing issues while giving us "good enough" estimates of packet latency, throughput, and loss.

1 INTRODUCTION

Wireless communication is poised to affect virtually everything we do with computers; what we are able to do using wireless depends significantly on what performance these networks deliver.

The work reported in this paper is a product of our interest in designing and evaluating large-scale ad-hoc networks that would be deployed in the context of an emergency response. One of the key issues there is the development of scalable routing protocols. Network characteristics such as latency and packet loss certainly affect routing decisions, but maintenance of forwarding tables happens at a time-scale that is much slower than packet transmission. Our intuition is that a highly detailed model of the Medium Access Control (MAC) layer which governs the transmission of packets is not necessary for routing studies, and might be simplified in order to accelerate the simulation. We explored this intuition in the development of the Simulator for Wireless Ad-Hoc Networks (SWAN) (Perrone, Nicol, Liu, Elliot, and Pearson 2001), which is based upon the DaSSF high performance simulation kernel (Cowie, Ogielski, and Nicol 1999). In SWAN we deployed a MAC layer simulation model that continuously measured "busy-ness"—an instantaneous measure of of messages available for transmission, and transformed that measure into message latency and packet loss probability. However, at the time of this work we had not yet implemented a more detailed model of the MAC, to support validation.

We have since ported the 802.11 model delivered with GlomoSim (Bajaj, Takai, Ahuja, Tang, Bagrodia, and Gerla 1999) to DaSSF and have incorporated it into SWAN. With a detailed 802.11 model available we found that under certain conditions our earlier simple model grossly under-estimated packet latency. This paper reports on our instrumentation of the detailed 802.11 model to reveal protocol behavior, and the ramifications that observed behavior must have on the simpler model.

2 MOTIVATION

Our goal is to develop a model that determines whether a new packet offered to the MAC layer ought to be accepted or not, and when that packet's turn for transmission arrives, predicts how long it will take for the channel to be safely acquired and the packet delivered. We aim for a simpler model to incorporate delays due to contention and retransmission, without actually emulating 802.11's behavior for contention control and message acknowledgment. This will reduce considerably the number of events needed to model packet transmission delays under heavy traffic conditions—the logic for listening to a channel, dealing with backoffs, detecting collisions and scheduling retransmission delays—none of that would be explicitly done. In the simple model, when a message is ready for transmission, we would compute a delay until its eventual receipt, then deliver the message to the recipient with the time at which the last bit of the message is received. The computed delay would be a function of the observed state and demand on the wireless channel.

A second benefit motivates us to develop a simpler model along the lines above. Our SWAN code runs on parallel computers. When it does so, the individual processors must coordinate in simulation time. DaSSF (upon which SWAN is built) exploits “lookahead” to coordinate these processors. In this context it means the smallest amount of time between when a message is transmitted and when the fact of that transmission can affect the state of the recipient. In the detailed 802.11 model, this lookahead is latency–the time between when a bit is transmitted and when it is received. This is just a few microseconds. The reason for this sensitivity is that node A may be receiving a message from node B, but node C (which cannot “hear” this transmission) begins sending to A anyway.

Parallelism is limited to simulation activity that happens within that window of a few microseconds. Synchronization overhead is incurred in every window of simulation time. The simpler model reduces the amount of overhead can be reduced by increasing lookahead.

3 IEEE 802.11 MAC PROTOCOL

IEEE 802.11 is a protocol standard for wireless local area networks (WLAN), that consists of both the physical (PHY) layer and the medium access control (MAC) layer specifications (IEEE Computer Society 1997). It provides asynchronous and time-bounded delivery service for wireless connectivity of fixed, portable, and mobile stations moving at pedestrian and vehicular speeds within a local area.

The 802.11 MAC layer protocol provides shared access to a wireless channel. The distributed coordination function (DCF) is the primary access method, which provides contention-based shared access to the medium. DCF is based on CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance). Another access method, known as point coordination function (PCF), is useful for infrastructure network configurations, using a point coordinators within access points to coordinate contention-free access to the medium. In this paper, we consider only the DCF protocol.

The core mechanism used by DCF is called *Basic Access Method*, which is summarized in Figure 1. Before a station initiates transmission of a data frame (called MAC protocol data unit, or MPDU), it needs to sense the channel in order to determine whether another station is currently transmitting. The station can proceed with its transmission if the medium is determined to be idle for a time interval of DIFS (DCF Inter-Frame Space). After a data frame is successfully received at the destination, the receiver must send an acknowledgment frame (ACK), because the transmitter cannot determine whether a frame has been faithfully delivered to its destination by simply listening to the channel—the sender may not observe frame collisions the receiver detects with other senders not observable by the first (this is the so-called “hidden terminal problem”). To transmit the ACK, the receiver waits for the channel to be idle for another time interval of SIFS (Short Inter-Frame Space). If the transmitter does not receive an acknowledgment within a certain time-out period, it presumes that the data frame is lost and schedules a re-transmission.

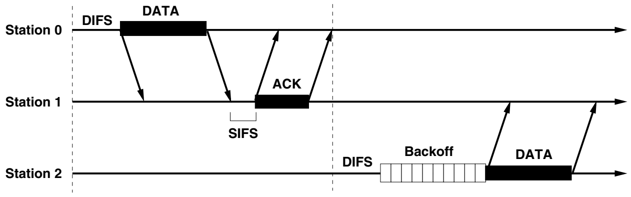

Figure 1: Basic Access Method of IEEE 802.11 MAC Protocol

If the medium is busy upon transmitting a data frame or an ACK, the transmission must be deferred until the end of the ongoing transmission. In this case, a random backoff interval is selected, as follows. A backoff timer is set with a random backoff integer (BV) drawn from a uniform distribution over the interval $[0, CW - 1]$, where CW (Collision Window) is an integer within the range of CW_{min} and CW_{max}. BV is the number of idle “slots” the station must wait until it is allowed to transmit—there is a specified and understood slot duration. The value is decremented by one for each idle slot detected. The backoff timer suspends when the medium becomes busy before BV reaches zero. The timer resumes only after the medium has been idle longer than the designated inter-frame space interval. The station starts transmitting the frame when the backoff timer reaches zero. For each successive retransmissions, the value of CW increases exponentially (i.e. $CW_{new} = CW_{old} * 2 - 1$), until it reaches and then stays at CW_{max}. CW will be reset to CW_{min} after a successful transmission. The backoff method is used to minimize collisions and maximize throughput at both low and high network utilizations.

DCF also provides an alternative way of transmitting data frames that involves transmission of RTS (request to send) and CTS (clear to send) prior to the actual data transmission. RTS and CTS are used to reserve the channel between the transmitter and receiver. An RTS frame is transmitted by a station which needs to transmit a data packet. The receiving station responds with an CTS frame. The rules for transmission of RTS and CTS frames are the same as those of the data frame and the acknowledgment frame. RTS and CTS frames contain a duration field that tells the period of time the channel is to be reserved for transmitting the data frame. This information is picked up by other stations in the area that are sensing the channel. It helps them to construct NAV (network allocation vector) –

the period of time a station is required to be kept silent. The technique is referred to as virtual carrier sense mechanism and is used to reduce contentions due to hidden terminals.

4 BEHAVIOR OF THE 802.11 MAC LAYER

In order for us to understand the behavior of the 802.11 MAC layer, we instrumented the detailed 802.11 simulation model and ran it under different offered loads. We are particularly interested in packet loss due to queueing overflow, retransmission behavior, and the composition of the aggregate packet latency—from the point it is presented to the MAC layer until it is received by the intended application. We study three synthetic networks in which every station can directly communicate with every other node (thereby eliminating the possibility of hidden terminals); we consider networks with 5, 10, and 20 stations. We assume a perfect channel. Each packet is comprised of 1K bytes plus MAC and PHY layer headers. Each station generates packets for the MAC layer in accordance with a Poisson process, every station uses the same rate. We therefore control the offered load through the common packet generation rate. We assume 802.11 uses direct sequence spread spectrum (DSSS) at the physical layer and use parameters according to the IEEE specification. Also, we ignore radio capture capability of tranceivers. That is, two packet frames are considered as colliding and therefore corrupted if their transmission is overlapped at the receiver side, despite their difference in the signal power.

Figure 2 shows the throughput versus the offered load for networks that consist of 5, 10, and 20 stations. Both throughput and offered load are normalized with respect to the channel capacity, that is, the units of both axes are fraction of channel capacity. As the offered load increases, the number of received packets increases as well, until saturation. Before saturation (at approximately 75% channel capacity), packet loss is very rare. At offered loads past the saturation point throughput remains constant. After saturation, the buffers between the MAC layer and the traffic generators are kept full, with the result that increasingly many offered packets are dropped due to buffer overflow. The saturation throughput value is a decreasing function of the number of stations. This occurs because adding stations increases the chance of collision, and increases the number of retransmissions.

Figure 3 shows that the fraction of packets retransmitted increases with the offered load, but levels off after the saturation point. After saturation the load contending for the channel does not appreciably change with increasing offered load, because the excess is shed by MAC layer buffer overflows.

We will be particularly interested in what we call the packet *service time*, which we define to be from the instant when the packet is at the head of its MAC queue ready to go onto the channel, to the instant when the post-transmission acknowledgment for the packet is received. It is in essence the length of time the MAC layer is holding that packet and preventing any other from being transmitted. Figure 4 shows the service times (in seconds) as we vary the (normalized) offered load. Prior to saturation the service time reflects essentially no contention for the channel. Contention and retransmissions increase as saturation is approached and past. The differences in post-saturation service times for different numbers of stations correlate with the differences in retransmission percentages, due to increased probability of collision.

From Figure 3 and 4, we see that the channel behavior can be divided into three states: relatively idle, transition, or saturated. In the relatively idle state (from 0 to approximately 60% of channel capacity), the throughput of the

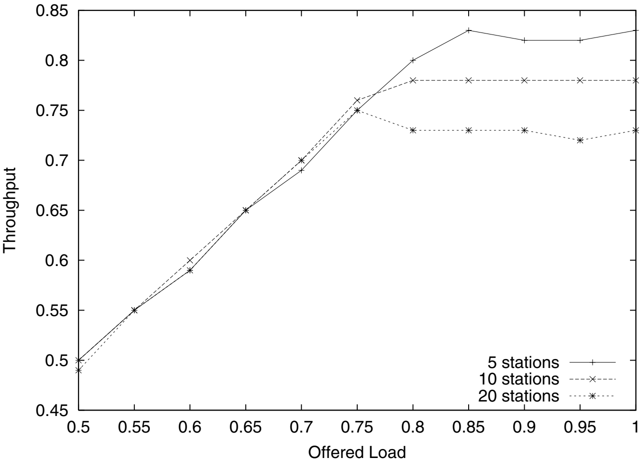

Figure 2: Throughput versus Offered Load for IEEE 802.11 MAC Protocol

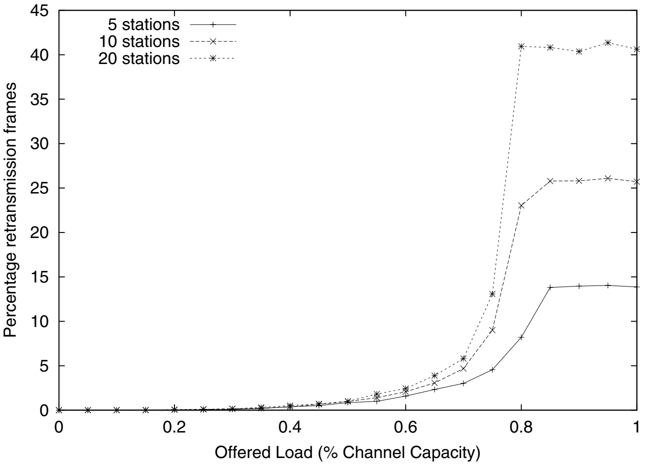

Figure 3: Number of Retransmissions versus Offered Load for IEEE 802.11 MAC Protocol

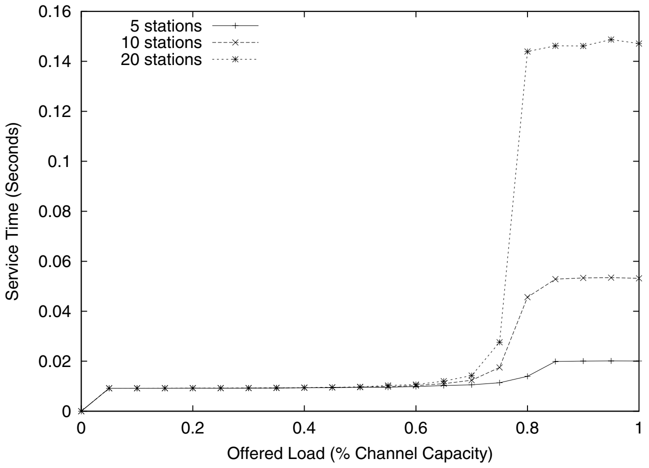

Figure 4: Service Time versus Offered Load for IEEE 802.11 MAC Protocol

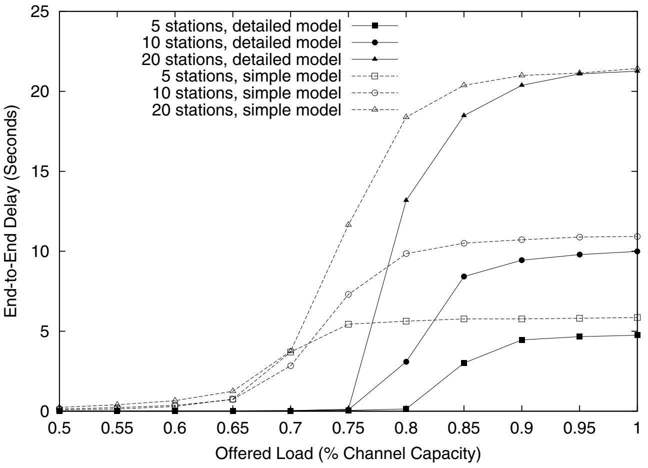

Figure 5: Packet Delay versus Offered Load for both detailed IEEE 802.11 MAC Model and the Simple Model

network keeps up with the offered load. The packets are delivered faithfully with nearly no losses and with only a few retransmissions. The “service time” of the packets stays low and remains roughly constant irrespective of offered load. In a transition state (between 60% and 85%), the packet loss, the fraction of retransmissions, and the packet service time all increase dramatically with offered load. However, in the saturated state all of these figures stablize, albeit at different levels for varying numbers of stations.

5 A SIMPLE MODEL

As we explained earlier, we are motivated to develop a simpler model of channel behavior under 802.11 in order to reduce the number of events, and to increase the inherent lookahead. Like the detailed model, our simpler model will maintain a queue of messages awaiting transmission from the MAC layer, and will drop messages that arrive to a full queue. However, unlike the detailed model, when a packet reaches the front of the queue we model (i) the time ultimately required for it to be received, and (ii) the time at which the sending queue is free to send another message. These delays will be a function of the channel state.

In our first effort then, we estimate by measurement whether the channel is saturated or not. Our measurements are detailed in that every station is made aware of every transmission made by every other station it can hear. Our earlier experiments suggest a saturation threshold of 75% of the channel capacity. We do not attempt to model transitional behavior—we assume that either the channel is idle enough for packets to be directly transmitted, or is saturated. As a function of the earlier experiments we ascribe constant service times to the packets, depending on channel state and number of stations. These constants are taken from the detailed model experiments; the significant issue of estimating these as a function of topology and load (and *not* by measurement) is not one we address here. Our goal in this paper is simply to assess the feasibility of using a simpler model of channel utilization.

To assess the simple model, we compare its results with that of the detailed 802.11 model. We compare the average end-to-end packet delay (from presentation at the sender’s MAC layer to delivery out of the receiver’s MAC layer), and the throughput, both as functions of offered load and numbers of stations. Figure 5 plots the delay data, while Figure 6 plots the throughput data. We should not be especially surprised that the delay curves match well outside of the transition region—after all, an end-to-end delay is essentially a sum of service times, and we matched the simple model’s service times to the detailed model’s at those extremes. The throughput comparison is more interesting, in that agreement increases with the number of stations. Since throughput must be the sum of the inverses of service times at each station, the fact that the simple model predicts throughput of the 20 station model better than the throughput of the 5 station model is directly due to the fact that the relative error of the service time on the 20 station model is much smaller than it is for the 5 station model.

No doubt finer tuning of the simple model’s constants can improve the comparisons, and certainly attention must be paid to modeling behavior when the channel is in the transition state. The point we wished to establish is that a simple model goes a long way towards reflecting the essential behavior of the MAC layer.

Performance improvement is the motivation for our work. We report here preliminary results of experiments designed to reveal the performance potential of the simplified model. We ran the experiments on a Linux workstation with a Pentium III 750MHz processor and 512K memory, using

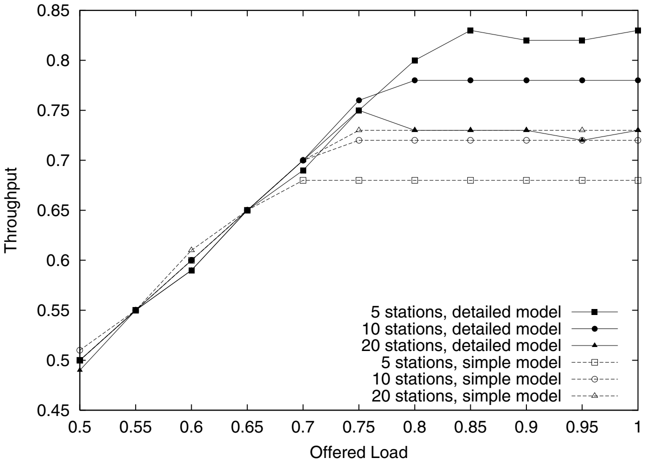

Figure 6: Throughput versus Offered Load for both detailed IEEE 802.11 MAC Model and the Simple Model

Table 1: Execution Timings (in seconds), Single Timeline

Stations	Load	Detailed	Simple	Speedup
5	10%	9.08	5.1	1.78
5	50%	43.7	21.8	2.00
5	90%	97.6	35.1	2.78
10	10%	14.6	7.3	2.0
10	50%	71.9	32.9	2.18
10	90%	170.6	52.2	3.26
20	10%	25.9	11.7	2.21
20	50%	129.4	54.8	2.36
20	90%	324.1	86.7	3.73

DaSSF (Cowie, Ogielski, and Nicol 1999) compiled with GNU C/C++ at optimization level three.

We ran one experiment to assess the performance gain solely due to reduced event and message complexity. The lookahead component relates to models with multiple independent timelines (i.e. threads) and can be eliminated by running the entire model on one timeline. Each model was advanced for 10000 simulated seconds. Table 1 gives the runtime in wallclock seconds, as a function of number of stations, offered load (in percentage of channel capacity) and model type. At low offered load, the basic difference between models is that the detailed one models acks and the simple one does not, reducing message traffic (and thus events) by half, and yielding a speedup of about 2. At higher load the detailed model engages in retransmissions that the simple model does not, and so the simple model delivers better speedups still. Evidently there is significant performance gain to be had by reducing the event complexity.

These results not withstanding, the supreme performance advantage is due to better lookahead. The lookahead used by by the simple model encompasses the entire frame transmission time, whereas for the detailed model it is essentially a propagation delay. The former is 1500 times larger than the latter. When we ran experiments where every station is on its own timeline, the simpler model ran 250 times faster on the 5 station model. The ramification of this are extremely important as we consider moving to parallel processing, for there we *must* use multiple timelines. We believe that with some effort we can improve the identification of lookahead in the detailed model; the results presented here certainly give us ample motivation for doing so.

6 RELATED WORK

Our simplistic model is by no means the only attempt to model 802.11. Worthwhile analytic efforts are reported in (Bianchi, Fratta, and Oliveri 1996; Cali, Conti, and Gregori 1998; Chhaya and Gupta 1995; Tay and Chua 2001). These models seek qualitative explanations for 802.11; our motivation is to deploy quantitative computational models. We are looking at this work though to provide ideas for our own needs.

7 CONCLUSION

There are significant performance advantages to simulating a wireless network's MAC layer with a model that is simpler than true 802.11. For serial simulation the key performance benefit is due to reduction of events needed. For parallel simulation the overwhelming performance benefit is from better lookahead.

The key contributions of this paper are to report on the implementation of 802.11 in the DaSSF framework, the identification of 802.11 model simplification as a worthy goal, and preliminary results that confirm our intuition that such a simplification can yield important performance gains.

ACKNOWLEDGMENTS

We thank Chip Elliot for fruitful discussions about modeling 802.11. This research is supported in part by DARPA Contract N66001-96-C-8530, NSF Grant ANI-98 08964, NSF Grant EIA-98-02068, NIJ contract 2000-DT-CX-K001, and Sandia National Laboratories under Department of Energy contract DE-AC04-94AL85000.

REFERENCES

Bajaj, L., M. Takai, R. Ahuja, K. Tang, R. Bagrodia, and M. Gerla. 1999. Glomosim: A scalable network simulation environment. Technical Report 990027, UCLA, Computer Science Department.

Bianchi, G., L. Fratta, and M. Oliveri. 1996. Performance evaluation and enhancement of the CSMA/CA MAC protocol for 802.11 wireless lans. *Proceedings of PIMRC '96 - 7th International Symposium on Personal, Indoor, and Mobile Communications*:392–6 vol.2.

Cali, F., M. Conti, and E. Gregori. 1998. IEEE 802.11 wireless LAN: capacity analysis and protocol enhancement. *Proceedings IEEE INFOCOM'98 Conference on Computer Communications Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies Gateway to the 21st Century*:142–9 vol.1.

Chhaya, H., and S. Gupta. 1995. Throughput and fairness properties of asynchronous data transfer methods in the IEEE 802.11 MAC protocol. *Proceedings of 6th International Symposium on Personal, Indoor and Mobile Radio Communications*:613–17 vol.2.

Cowie, J., A. Ogielski, and D. Nicol. 1999, January/February. Modeling the global internet. *Computing in Science and Engineering* 1 (1): 42–50.

IEEE Computer Society, L. M. S. 1997. Wireless LAN medium access control (MAC) and physical layer (PHY) specification.

Perrone, F., D. Nicol, J. Liu, C. Elliot, and D. Pearson. 2001, June. Simulation modeling of large-scale ad-hoc sensor networks. In *Proceedings of the 2001 Simulation Interoperability Workshop*. London, England.

Tay, Y., and K. Chua. 2001. A capacity analysis for the ieee 802.11 mac protocol. *Wireless Networks* 7 (2): 159–71.

AUTHOR BIOGRAPHIES

JASON LIU is a Ph.D student in the Computer Science Department at Dartmouth College. His research focuses on parallel discrete-event simulation, performance modeling and simulation of computer systems and communication networks. He is also interested in large-scale simulation of wireless communication systems. He received B.A. in Computer Science from Beijing Polytechnic University in China in 1993. He graduated from College of William and Mary in 2000 with his M.S. in Computer Science.

DAVID M. NICOL is Professor and Chair of the Department of Computer Science at Dartmouth College. He received the B.A. in Mathematics from Carleton College in 1979, and the Ph.D. in Computer Science from the University of Virginia, in 1985. He has since then contributed to the field of simulation, particularly in the area of parallel discrete-event simulation. He is Editor-in-Chief of the ACM Transactions on Modeling and Computer Simulation, and is co-author (with Banks, Carson, and Nelson) of the text *Discrete Event System Simulation, 3^{rd} Edition.* His current research interests are focused on simulation of large scale networks, and network security.

LUIZ FELIPE PERRONE is a Research Associate with the Institute for Security Technology Studies and Adjunct Professor at the Department of Computer Science at Dartmouth College. His research interests are modeling and simulation of large-scale wireless networks, and parallel simulation in general. He received the degree of Electrical Engineer from UFRJ in Brazil, in 1989, the M.Sc. in Systems Engineering and Computer Science from COPPE/UFRJ, in 1992 and the Ph.D. in Computer Science from The College of William & Mary, in 2000.

MICHAEL LILJENSTAM is a Postdoctoral Research Associate at the Computer Science Department at Dartmouth College. His current research focus is on large scale network modeling, internet traffic analysis, and modeling and simulation of wireless networks. He received his M.Sc. and Ph.D. degrees in Computer Science from the Royal Institute of Technology (KTH), Stockholm, Sweden in 1993 and 2000, respectively. His graduate work centered on parallel discrete event simulation techniques, and in particular their application to cellular wireless network models. He has participated in simulation related projects at KTH, 1994, and in industry (CAP Programator Stockholm), 1994, and during an internship at Telia Research AB in 1997.