

Speedup-Precision Tradeoffs in Time-Parallel Simulation of Wireless Ad hoc Networks

Damla Turgut, Guoqiang Wang, Ladislau Bölöni, and Dan C. Marinescu
School of Electrical Engineering and Computer Science
University of Central Florida
Orlando, FL 32816-2450
{turgut,gwang,lboloni,dcm}@cs.ucf.edu

Abstract

In this paper, we report on a series of experiments involving the speedups obtainable with time-parallel simulation of wireless ad hoc networks.

A mobile ad hoc network scenario involving the AODV and DSDV routing protocols was simulated. The results and the performance of the serial NS-2 simulator was compared to the time-parallel simulation method for wireless ad hoc networks, previously introduced by the authors. The results of the time-parallel simulation are approximations, and we find that there is a tradeoff between the precision of the simulation and the achievable speedup. However, it is possible to find compromises where a precision of the range of 95-98%, sufficient for most applications, can be obtained up to 10 times faster than the time needed by a serial simulation.

1 Introduction

Simulation of wireless networks is important for protocol design and wireless system research. As the research progresses from relatively simple systems composed of several nodes connected to a wireless access point to large systems which might be composed of thousands of nodes (such as wireless sensor networks), the size of the simulation problem becomes so large that it clearly exceeds the capabilities of a single machine. Therefore, it is important to look into parallel simulation approaches. Parallel discrete event simulation (PDES) is proposed in recent years to reduce the overall execution time by parallel execution of the simulation on multiple processors. The parallel simulation approaches merge to two main categories: space-parallel simulation (distributed simulation), and time-parallel simulation. In the space-parallel simulation approach [7, 9], the simulation model is decomposed into a number of components on a spatial basis. Each component is modeled by a *logical processor*. Logical processors establish a communication mechanism among each other to avoid or fix possible

causality errors. The Parallel/Distributed NS (PDNS) [16] project uses a space-parallel simulation approach based on the NS-2 network simulator [12]. However, the applicability of PDNS is limited to *wired* networks, and the traffic simulated at different spatial partitions cannot affect each other.

A parallel simulator, SWiMNet [3–5], is used for personal communication services (PCS) networks. It is based on a combination of optimistic and conservative paradigms and makes use of the event precomputation by the model independence within the PCS model. Independencies between processes allow to achieve parallelism. SWiMNet is used in simulation of PCS networks with fixed channel assignment by specifying fine grained mobility, variable call process, and arbitrary coverage area.

In the time-parallel simulation approach [1, 8, 14, 15], the long period of simulation time is partitioned into smaller adjacent simulation intervals, and each simulation interval is assigned to a processor with a *guessed* initial state. The simulation terminates when the final state of each interval matches the initial state of its successive interval. Thus, state matching is one of the key problems of time-parallel simulation.

The major difficulty in parallel simulation approaches is solving the dependencies among the partitions. A careful study of the temporal dependencies for wireless networks with various traffic patterns shows that there are few strong dependencies which span a large temporal interval. By ignoring the weak dependencies, we can achieve fast simulation results which are a good approximation of the exact results. Furthermore, we can assemble a system where coarse approximations are obtained very quickly, while the continuation of the simulation process yields results with increasingly higher precision. In the knowledge of the approximate results, the researcher might decide whether the expensive continuation of the simulations is justified or not.

In this paper, we investigate the relationship between the

speedup and the precision in a time-parallel simulation approach. In [2, 13], the authors have proposed a method of time-parallel simulation of wireless ad hoc networks, which uses an iterative approach to gradually increase the precision of the simulation. This inevitably represents a trade-off between speedup and precision. The goal of the experiments presented in this paper is to investigate whether there is a “sweet spot” which offers sufficient precision for practical purposes, but also provides a significant speedup which justifies the considerable expense involved in the use of a cluster computer with a large number of nodes.

2 Simulation Study

2.1 Simulation scenario

To run our simulations in a realistic setting, we had chosen a setting which is representative to the scenarios used by researchers. The scenario involves a set of mobile nodes moving in a rectangular simulation area, with a set of pre-determined communication patterns. The transmission range of the nodes is significantly smaller than the simulation area. The node-to-node communication is facilitated by a wireless ad hoc network routing protocol. One of the challenges of the scenario is that the mobility of the nodes changes the available routes. Frequently, the route needs to be recalculated while a transmission is in progress. Various routing protocols offer different solutions to this challenge. Note that in our simulations, we are not interested in the performance of the selected protocols, but in the speed and accuracy with which we can measure this performance.

We had chosen to use two well known protocols which are representative of the major classes of wireless ad hoc routing protocols. Destination Sequenced Distance Vector Routing (DSDV) [10] is a pre-emptive routing protocol in which the nodes maintain routing tables and perform actions to keep them up-to-date. Ad-hoc On-Demand Distance Vector Routing (AODV) [11] is a reactive, on-demand routing protocol where the routes are established only as a result of explicit demand. We shall see that these two classes of protocols exhibit different behavior in relation to the time-parallel simulation.

We use the “random waypoint” model [6] to simulate the node movement. Traffic patterns are generated by *constant bit rate* (CBR) sources sending 512-byte UDP packets at a rate of 1 packet per second. The simulation area is 500×500 and the default number of nodes is 80. All the nodes have a transmission range of 100 meters. The simulation time of 600 seconds is segmented into 20 time intervals of 30 seconds. We run several simulation experiments by varying the segment duration, number of CBR sources, and the speed. Table 1 shows the default settings and the range of the parameters for our experiments.

For each experiment, we randomize the source-destination pairs of CBR sources, and execute 10 times to

Table 1. The default values and the range of the parameters for the simulation scenario.

| Field | Value | Range |
|------------------------------|-----------------------|--------------|
| <i>simulation area</i> | $500 \times 500(m^2)$ | |
| <i>number of nodes</i> | 80 | |
| <i>transmission range</i> | 100 (m) | |
| <i>speed</i> | 1 (m/s) | 1 - 21 (m/s) |
| <i>pause time</i> | 15 (s) | |
| <i>simulation time</i> | 600 (s) | |
| <i>segment duration</i> | 30 (s) | 10 - 60 |
| <i>number of CBR sources</i> | 20 | 4 - 40 |
| <i>CBR packet size</i> | 512 (bytes) | |
| <i>CBR sending rate</i> | 4 (kbps) | |

obtain the average, as well as, 95% confidence interval for each quantity.

The simulation was run on the I²Cluster cluster computer of the UCF Interdisciplinary Research Laboratory. This computer is composed of 64 SunFire v20z computers, each with two 64-bit Opteron processors. The cluster is interconnected by four Cisco Catalyst Gigabit Ethernet switches. 3 TB of storage is attached to the front-end machines.

To keep the comparison fair, the sequential simulation was run on one of the cluster nodes. For the time-parallel simulation, the number of nodes used depended on the size of segment duration, and varied between 10 and 60.

2.2 Relative error in function of iteration number

To establish the accuracy of our time-parallel simulation relative to an exact sequential simulation, we evaluate the *relative error* for several performance indicators. Let M_0 be the result produced by the exact sequential simulation and M the one produced by our time-parallel algorithm; the *relative error* for this metric is $\varepsilon = \frac{M - M_0}{M_0} \times 100\%$. We investigate the relative error for the packet loss ratio and the throughput of the given algorithm.

The experiment was repeated with different values for the duration of the individual segment. This parameter determines the number of parallel processes in the simulation; for instance with a simulation time of 600 seconds and a segment duration of 20 seconds, we can run the simulation on 30 parallel threads.

This set of experiments (Figure 1) allows us to investigate the effect of segment duration and determine the number of iterations required to obtain results with a given level of accuracy; we choose as a threshold for the relative error $\alpha = 5\%$. We experiment with segment durations of 10, 20, 30, 40, 50, and 60 seconds. As the simulation time is fixed at 600 seconds, the number of segments are 60, 30, 20, 15, 12, and 10, respectively. We compare the simulation results after each iteration with the simulation results obtained by

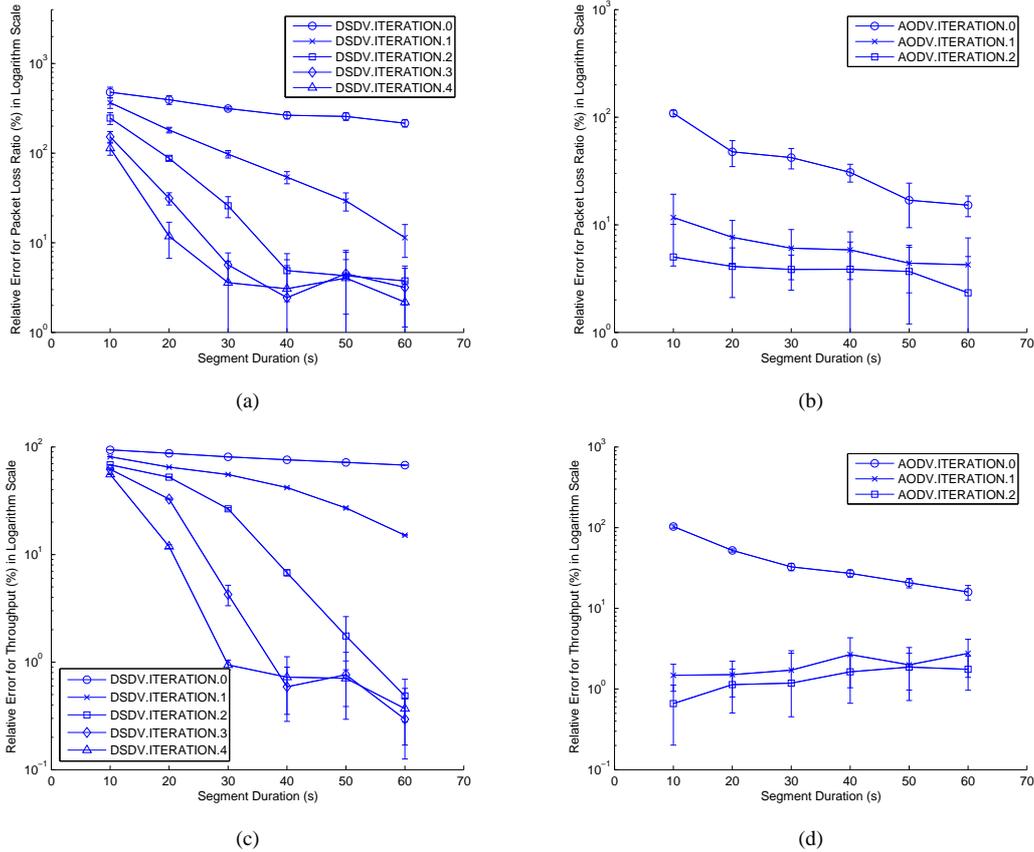


Figure 1. Relative error for packet loss ratio and throughput as function of the segment duration in a logarithmic scale; DSDV (left) and AODV (right). (a) and (b) show the relative error for the packet loss ratio; (b) and (d) show the relative error for the throughput.

exact sequential simulation.

The curves labeled *PROTOCOL.ITERATION.i* in Figure 1 show the relative error after iteration i for the two protocols. We note that the speedup for a single iteration is equal with the number of time segments (provided that there are a sufficient number of computational nodes); however, smaller segments require a larger number of iterations to achieve equivalent precision. Overall, however, the speedup tends to increase with the decrease of segment size. Thus, a proper segment duration need to be chosen according to the simulation time and the number of available computational nodes.

2.3 Speedup and maximum network size

In our second set of experiments, we studied the time needed to simulate a specific scenario for various protocols using the serial NS-2 simulator and our approach of time-parallel simulation. For this experiment, we used the scenario presented in Table 1 with 1400 nodes. The time-parallel simulation was run until the precision reached 95%. The results are shown in Figure 2 (left). We note that

the time-parallel simulation performs approximately 5 to 6 times faster than NS-2 in these scenarios. The simulation times for DSDV were somewhat higher than the ones for AODV, reflecting the additional overhead of that protocol required to maintain the routing tables.

An alternative way to study the benefits of time-parallel simulation is by investigating how many nodes can be simulated in a given amount of time. We fixed the maximum simulation time to 10 minutes. The results are presented in Figure 2 right. We observe that the number of nodes which can be simulated in this timeframe by serial NS-2 is in the range of 180-200 nodes, this increases to 1000-1200 nodes if time-parallel simulation is used.

3 Conclusions

In this paper, we report on a series of experiments concerning the precision-speedup tradeoffs of time-parallel simulation of wireless ad hoc networks. We find that the achievable parallelism is limited by the minimum segment size; when the segment duration is too small the simulation

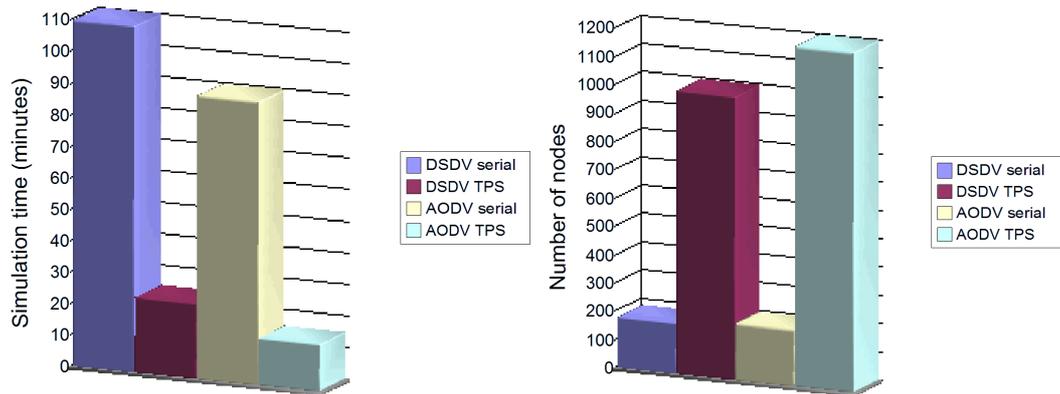


Figure 2. Simulation time for an identical scenario of 1400 nodes (left) and maximum number of nodes which can be simulated in 10 minutes timeframe for (a) DSDV using serial NS-2, (b) DSDV using time-parallel simulation (c) AODV using serial NS-2 and (d) AODV using time-parallel simulation.

converges very slowly, the number of iterations effectively negating the benefits of parallelism. We find that for typical scenarios, a 20-30 process parallelism leads to the optimal speedup. Comparing to serial simulation, we found that the simulation time can be reduced 5-6 times, or, viewed differently, the number of nodes which can be studied with a moderate length, 10 minute simulation can be increased from 180-200 to more than 1000 nodes.

The research reported in this paper was partially supported by National Science Foundation grants ACI0296035 and EIA0296179.

References

- [1] S. Andradóttir and T. J. Ott. Time segmentation parallel simulation of networks of queues with loss or communication blocking. *ACM Trans. on Modeling and Computer Simulations*, 5(4):269–305, 1995.
- [2] L. Bölöni, D. Turgut, G. Wang, and D. Marinescu. Challenges and benefits of time-parallel simulation of wireless ad hoc networks. In *to be presented at the First International Conference on Performance Evaluation Methodologies and Tools (Valuetools-2006)*, October 2006.
- [3] A. Boukerche, S. K. Das, and A. Fabbri. SWiMNet: A scalable parallel simulation testbed for wireless and mobile networks. *ACM/Kluwer Wireless Networks*, 7:467–486, 2001.
- [4] A. Boukerche, S. K. Das, A. Fabbri, and O. Yildiz. Exploiting model independence for pcs parallel simulation. In *Proceedings of 13th ACM/IEEE workshop on Parallel and Distributed Simulation*, pages 166–173, 1999.
- [5] A. Boukerche and A. Fabbri. Partitioning parallel simulation of wireless networks. In *Proceedings of 2000 Winter Simulation Conference*, pages 1449–1457, 2000.
- [6] J. Broch, D. A. Maltz, D. B. Johnson, Y. Hu, and J. Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *Proc. Mobile Comp. & Networking*, pages 85–97, 1998.
- [7] R. M. Fujimoto. Parallel discrete event simulation. *CACM*, 33(10):30–53, 1990.
- [8] Y.-B. Lin and E. D. Lazowska. A time-division algorithm for parallel simulation. *ACM Trans. on Model. & Comp. Sim.*, 1(1):73–83, 1991.
- [9] V. K. Madiseti, J. C. Walrand, and D. G. Messerschmitt. Asynchronous algorithms for the parallel simulation of event-driven dynamical systems. *ACM Trans. on Model. & Comp. Sim.*, 1(3):244–274, 1991.
- [10] C. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In *SIGCOMM '94*, pages 234–244, 1994.
- [11] C. Perkins and E. Royer. Ad hoc On-demand Distance Vector Routing. In *Proc. 2nd IEEE Workshop on Mobile Comp. Sys. & Apps*, pages 99–100, 1999.
- [12] VINT project. the ucb/lbnl/vint network simulator-ns (version 2). URL <http://www.isi.edu/nsnam/ns>.
- [13] G. Wang, D. Turgut, L. Bölöni, and D. Marinescu. Accuracy-speedup tradeoffs for a time-parallel simulation of wireless ad hoc networks. *to be presented at the Second IEEE International Workshop on Performance and Management of Wireless and Mobile Networks (P2MNet)*, November 2006.
- [14] J. J. Wang and M. Abrams. Approximate time-parallel simulation of queueing systems with losses. In *Proc. 24th Winter Sim. Conf.*, pages 700–708, New York, NY, USA, 1992. ACM Press.
- [15] J. J. Wang and M. Abrams. Determining initial states for time-parallel simulations. In *PADS '93: Proc. 7th Workshop on Parallel & Distrib. Sim.*, pages 19–26, New York, NY, USA, 1993. ACM Press.
- [16] PDNS – Parallel/Distributed NS. URL <http://www.cc.gatech.edu/computing/compass/pdns/>, 2004.