Time-parallel simulation with compressed history

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

Abstract

In this paper, we introduce a compressed history *method which promises a significant speedup of time-parallel simulation of wireless ad hoc networks simulation. We replace the warmup interval discussed in [2, 11, 13] with a shorter and/or simpler simulation interval. We present an implementation of time-parallel simulation, with the compressed history as the warmup method. A series of experiments indicate a good concordance between the results of the simulations and the predicted findings.*

1 Introduction

Time-parallel simulation of complex systems by partitioning the time domain, rather than the space domain and its advantages, as well as, the theoretical and practical limitations have been extensively researched for many applications. As a general rule, time-parallel simulation requires the stochastic model of the system to be regenerative, a situation rarely encountered in practice. Yet, oftentimes we are satisfied with approximate results which could give us some qualitative rather than quantitative results.

Wireless ad hoc networks are rarely amenable to analytical modelling due to the complexity of the models involved. Thus, to obtain a quantitative evaluation of such models we often resort to simulation. Yet, current simulation techniques limit our ability to study systems consisting of thousands of nodes for extended periods of time. A wireless network with 1200 nodes pushes the limits of serial simulators such as NS-2. Even when feasible, such simulations require a significant computing power and can take a very long time. An alternative is a parallel implementation of the simulation. Space-parallel simulation of mobile systems is impractical thus our interest in time-parallel systems of wireless ad hoc networks.

To analyze and predict the performance of time-parallel simulation of wireless ad hoc networks, we introduced the notion of perturbation of measurements and present a layer-by-layer analysis of the impact of perturbations on the functioning of the wireless network [2, 11, 13]. Our experiments led us to the conclusion that time-parallel simulations can be very useful in the study of the behaviors of wireless ad hoc networks, especially for the physical and the `MAC` layer. As the perturbation on the initial state of a simulation batch essentially influences the accuracy of the simulation results, we proposed several methods that could possibly enhance the correctness of the initial state. Finally, we proposed an iterative time-parallel simulation approach that gradually increases the size of the warmup window for obtaining more and more accurate simulations.

The contribution of this paper is a *compressed history* time-parallel simulation, a method which promises a significant speedup. Our study shows that time-parallel simulation of wireless networks with compressed history enjoys a larger speedup while maintaining the accuracy of the results is in an acceptable range.

2 Previous work

Parallel discrete event simulation (`PDES`) reduces the overall execution time by concurrent execution on multiple processors [3, 7]. There are two flavors of parallel simulation: space-parallel (distributed simulation), and time-parallel. In the space-parallel simulation approach, the simulation model is decomposed into a number of components on a spatial basis. Each component is modelled by a *logical processor*; logical processors establish a communication mechanism to avoid/fix possible causality errors. A space-parallel simulator based on the `NS-2` simulator [12] is Parallel/Distributed NS (`PDNS`) [15]. However, the applicability of `PDNS` is limited to *wired* networks as the traffic simulated at different spatial partitions cannot affect each other. In time-parallel simulation, a long simulated time is partitioned into smaller adjacent intervals. Each interval is assigned to a processor together with a *guessed* initial state [1, 4, 6, 14] and the simulation terminates when

Proceedings of the Third International
Conference on Wireless and Mobile Communications (ICWMC'07)
0-7695-2796-5/07 $20.00 © 2007 IEEE

the final state of each interval matches the initial state of its successor. State matching is one of the key problems of time-parallel simulation. A simulation is defined as *partial regenerative* if there exists a subset of the system state variables and the sub-system represented by the subset can repeat its state infinitely. The system is then partitioned at the *regeneration points* which trigger *regenerative substates*. In some cases, the regeneration points of a regenerative simulation can be found without performing a detailed simulation [8]; the state matching problem can be solved by performing a pre-computation. A *pre-simulation* to identify regenerative points by using Markovian modelling is discussed in [14]. A time-parallel simulation algorithm based on state matching is presented in [6].

The widespread use of time-parallel simulation is restricted by the state-match problem, due to the difficulty in identifying regeneration points; this problem is especially hard for systems such as wireless networks. The search for approximate, rather than exact results, may simplify the temporal decomposition of simulation models [5]. A hard problem for any method proposing an approximate solution is to provide a reliable error estimation and an error control strategy. Although time-parallel simulation rarely allows us to obtain accurate results; however, meaningful approximate results can be produced efficiently [4, 5].

Simulation of wireless networks is an important tool for protocol design and wireless networks research in general. The size of the simulation problems encountered in practice often exceeds the capabilities of a single machine and time-parallel simulation offers an attractive alternative. The speedup and the accuracy of a time-parallel simulation of ad hoc networks is discussed in [2, 11, 13]; the papers analyze the packet loss ratio and throughput for two ad hoc routing algorithms: `DSDV` [9] and `AODV` [10]. Experiments show that to achieve simulation results with an acceptable accuracy, a significant amount of computing is required for the warmup interval which does not contribute to the measurement. This motivates us to investigate a *compressed history* simulation when the warmup interval includes only the subset of events which produce perturbations.

3 Initial state approximation

3.1 Techniques for improving the accuracy of time-parallel simulations.

The speedup of time-parallel simulation is archived by splitting the simulated time in a set of *batches* which correspond to time intervals, often of identical duration. For instance, in Figure 1(a), we see a partitioning of the simulation interval into three equal time intervals: $[0, t_1 = t/3]$, $[t_1 = t/3, t_2 = 2t/3]$ and $[t_2 = 2t/3, t]$. The figure also illustrates a potential source of inaccuracy of the simulation: the perturbations due to event e_1 migrate from one interval to the next, thus simulation results will be inaccurate.

Time interval shift. One way to increase the accuracy of the simulation is to select the borders of the simulation intervals such that the perturbations are completely contained in the batches, as shown in Figure 1(b). These points are the equivalent of the regenerative states of the stochastic models. If we cannot find states where all the perturbations are extinguished, we can search for points where there are a comparatively smaller number of active perturbations (partially regenerative states). There are several difficulties with this approach. First, there might not be any regenerative states in the simulation, or their distribution might be such that it does not lead to batches of size suitable for parallelization. Second, even if regenerative, or partially regenerative states exist, it may be difficult or impossible to identify them. There are circumstances when the identification of such points is possible; if a routing protocol periodically flushes the complete set of routing information, that instance is a natural partial regenerative point for some simulations.

Warmup interval. As time interval shift is possible only in special circumstances, we examine a different approach. Now for each batch we consider two phases: warmup and measurement. During the warmup, we perform the simulation but do not carry out any measurements; during the second phase we perform measurements, which should have a higher degree of accuracy. Indeed, in this case we take into account perturbing events occurring during the warmup phase events which would have been otherwise ignored.

Let us consider the question of the warmup window size. This window should contain all perturbing events whose echoes extend beyond the start of the measured interval (see Figure 1(c)). The size of the required warmup period can vary between various batches, the first batch does not require a warmup. We cannot accurately compute the warmup window size without first running the simulation, thus in practice, we only consider warmup windows of equal size and we gradually increase the size of the window to obtaining increasingly more accurate simulation results.

Warmup with compressed history. In a practical run of time-parallel simulation with warmup, the size of the warmup period can be significantly longer than the measured interval; thus most of the computation time is spent into the simulation of the warmup interval, which does not contribute to the measurements. Traditionally, the warmup interval is the exact match of the simulated events for a period before the measured interval starts. We can replace the warmup interval with a shorter and/or simpler simulation interval which, however, would yield the same results. For instance, we can remove all the events from the warmup period which do not produce perturbations at the measured point. For the events which produce perturba-

Proceedings of the Third International
Conference on Wireless and Mobile Communications (ICWMC'07)
0-7695-2796-5/07 $20.00 © 2007 IEEE

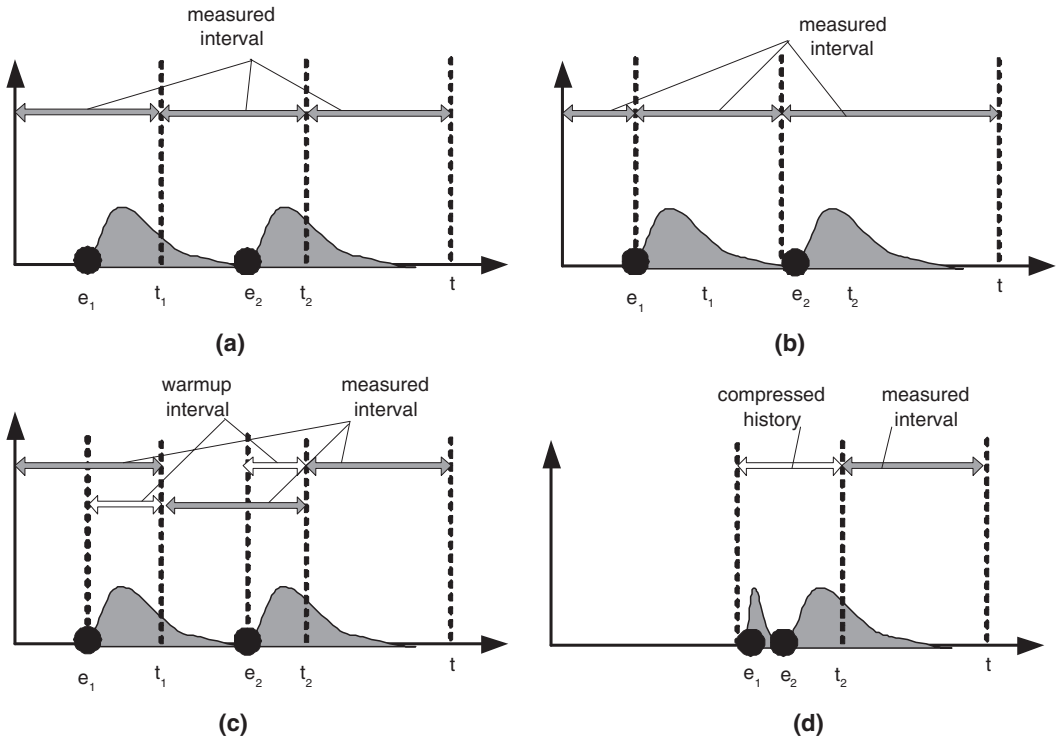

Figure 1. Techniques for improving the accuracy of time-parallel simulations. (a) Unimproved equal length batches. (b) Time interval shift. (c) Warmup intervals. (d) Warmup with compressed history.

tions, we might be able to replace them with events easier and faster to simulate. We call this modified warmup interval a *compressed history*.

3.2 Initial state approximation with compressed history.

The compressed history of a simulation defines the critical events that perturb the measurement the most.

A simulation S of an ad hoc network calculates the simulation trace $\mathcal{T}$ of event set E which occurs in geographic area A, within time interval τ, when the initial state is $\mathcal{I}$, and the final state is $\mathcal{F}$. Formally, we denote a simulation as a six-tuple set $S = (A, \tau, E, \mathcal{I}, \mathcal{F}, \mathcal{T})$. We partition the temporal dimension of the simulation S into m time intervals $\{\tau_i | \tau_i = [(i-1)\frac{D(\tau)}{m}, i\frac{D(\tau)}{m}], i \in \{1, \ldots, m\}\}$, where $D(\tau)$ calculates the duration of time interval τ. Then, we obtain a time-parallel simulation set $U = \{S_i | S_i = (A, \tau_i, E(S_i), \phi, \mathcal{F}(S_i), \mathcal{T}(S_i)), i \in \{1, \ldots, m\}\}$. Thus the time-parallel simulation set will operate on the full spatial component A, but a subset of the temporal span τ_i. In case of an exact simulation, the final state of S_i needs to match the initial state of S_{i+1}, $i \in \{1, \ldots, m-1\}$. However, since there is no prior knowledge for the initial states of $S_2, \ldots, S_m$ when they are parallelized, the combined simulation trace $\mathcal{T}(I) = \mathcal{T}(S_1) \cup \mathcal{T}(S_2) \cup \ldots \cup \mathcal{T}(S_m)$ is far from being approximate.

Define $\hat{E}$ as the set of events that produce significant perturbations on the measurement, $\hat{E} \subseteq E$ (in general case $\hat{E} \subset E$ and $|\hat{E}| << |E|$). The compressed history of the simulation S can be represented as $\hat{S} = (A, \tau, \hat{E}, \mathcal{I}, \mathcal{F}, \mathcal{T})$. Let $\underline{S_i}$ be the complete sequential simulation prior to S_i, $\underline{S_i} = (A, \bigcup_{j=1}^{i-1}(\tau_j), E(\underline{S_i}), \phi, \mathcal{F}(\underline{S_i}), \mathcal{T}(\underline{S_i}))$, for $2 \leq i \leq m$. Now, the initial state of a simulation batch S_i can be approximated by the final state of the compressed history of its prior simulation $\underline{S_i}$, that is, $\hat{\underline{S_i}}$, for $2 \leq i \leq m$. Thus, we obtain a new time-parallel simulation set $U' = \{S_i' | S_i' = (A, \tau_i, E(S_i'), \mathcal{I}(S_i'), \mathcal{F}(S_i'), \mathcal{T}(S_i')), i \in \{1, \ldots, m\}\}$, where

$$\mathcal{I}(S_i') = \begin{cases} \phi, & \text{for } i = 1; \\ \mathcal{F}(\hat{\underline{S_i}}), & \text{for } 2 \leq i \leq m. \end{cases}$$

3.3 Implementing compressed history

In [2], we analyzed the perturbations caused by events at different layers of wireless network protocols and concluded that events at the network layer have a much larger

Proceedings of the Third International
Conference on Wireless and Mobile Communications (ICWMC'07)
0-7695-2796-5/07 $20.00 © 2007 IEEE

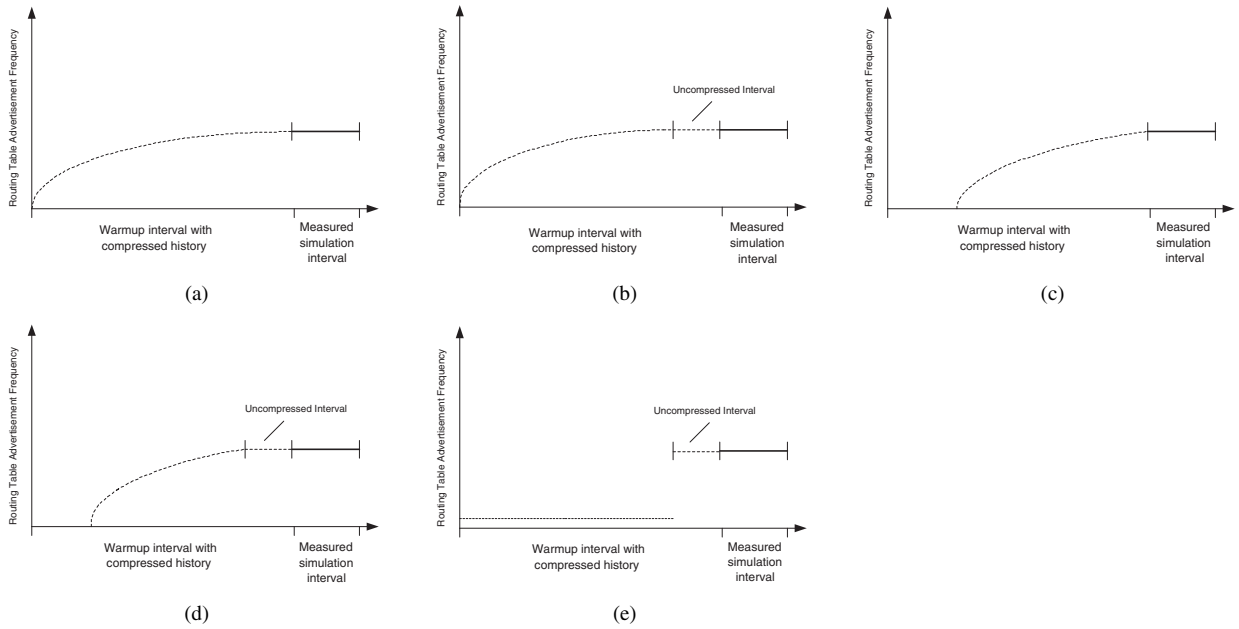

Figure 2. Strategies to adjust the routing table update frequency

influence on the measurements than the ones at the data link, MAC, and physical layer. Moreover, the events at the MAC and physical layer are generally correlated during the simulation and cannot be removed independently. Thus, our strategy to generate the compressed history of a simulation aims to identify insignificant network layer events.

Table-driven routing protocols require frequent routing table updates and an exact simulation will require a large amount of computations for the update packets. However, only the packets sent immediately prior to the measurement point could cause perturbations. Based on this observation, we can gradually increase the frequency of routing table update events in the compressed history, so that the events that produce significant perturbations are preserved, while those that do not produce perturbations are excluded.

Figure 2 illustrates the strategies to adjust the routing table update frequency. With strategy 2(a), the frequency increases gradually in the compressed history interval, and the regular frequency is met right before the measured point. With strategy 2(b), we included an interval right before the measured point, in which the routing table updates are sent regularly. The *uncompressed* simulation interval reduces the influence of perturbations at the measured point. With strategy 2(c), the routing updates sent long time ago are all ignored in the compressed history. The strategy illustrated in Figure 2(d) is the combination of strategies 2(b) and 2(c). Strategy 2(e) assumes that the perturbation of routing table update events is a small constant after a certain interval. A discussion and comparison of these strategies is necessary to generate the compressed history of a simulation; however, it is out of the scope of this paper. In our simulation, strategy 2(e) is implemented and investigated.

Another strategy suitable for both table-driven and on-demand routing protocols is to reduce the number of data packets. For table-driven protocols, the perturbation of the data packets is relatively small, since they normally do not affect the routing tables.[1] For on-demand protocols, the perturbation of a series of packets between the same pair of source and destination to the later events can be quite different. For instance, the attempt to send the first packet to an unknown destination will initiate a route discovery for finding a route, while the delivery of the remaining packets simply traverses through that established route. The perturbation of sending the first packet is more significant since the established route may be reused for the delivery of later packets. Thus, we can reduce the number of data packets as long as the route discovery process is progressed and the route is kept active for possible later use in the measured simulation interval.

The reduction of the routing table update packets or the data packets is promising to greatly reduce the execution time. However, a simulation warmed up with an over-simplified compressed history will lead to inaccurate simulation results. Thus, there is a tradeoff between the level of compression and the accuracy of the simulation.

[1] In case of highly loaded networks, the routing tables might be affected since the high volume of data packets may delay the delivery of the routing table update packets.

4 Simulation Study

We evaluate a set of metrics to establish the accuracy of the results of time-parallel simulation with the initial state approximated by the compressed history. The approximate results are compared to an exact sequential simulation.

We use the "random waypoint" model 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 simulated time is 600 seconds. Strategy 2(e) is used when we generate the compressed history. The routing table update frequency in the compressed history interval is set to be only $1/15$ of the regular frequency, and the duration of the uncompressed interval is set to 45 seconds. We run several simulation experiments by varying the segment duration. Table 1 shows the default settings of the parameters for our experiments.

Table 1. The default values of the parameters

Field	Value
simulation area	$500 \times 500(m^2)$
number of nodes	80
transmission range	100 (m)
speed	1 (m/s)
pause time	15 (s)
simulation time	600 (s)
segment duration	30 (s)
number of CBR sources	20
CBR packet size	512 (bytes)
CBR sending rate	4 (kbps)
uncompressed time interval	45 (s)

To establish the accuracy of our time-parallel simulation, we evaluate the *relative error* for several measurements. 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. Notice that both of these are average measurements.

For each experiment, we randomize the source-destination pairs of CBR sources, and execute 10 times to obtain the average, as well as, 95% confidence interval for each quantity. We use `DSDV` [9] routing protocol to investigate the performance of the time-parallel simulation with the application of the compressed history strategy. The simulation was run on a cluster computer composed of 128 64-bit `Opteron` processors.

Figures 3(a) and 3(b) show the packet loss ratio and throughput for the sequential simulation as well as the time-parallel simulation with *compressed history*, as a function of the segment duration. Figures 3(c) and 3(d) show the relative error for packet loss ratio and the throughput as a function of the segment duration. 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.

The relative error for packet loss ratio ranges from 1% to 4% (Figure 3(c)), and the relative error for throughput is below 1% (Figure 3(d)). The relative errors for both metrics are below the error threshold mentioned in [2]. The simulation results obtained with the aid of *compressed history* is more accurate than our previous results from [13]. The average speedup is larger than 10, almost three times the speedup reported in [13].

5 Summary and Future Work

In this paper we introduce *compressed history* time-parallel simulation with the initial state approximated by the *compressed history* of its prior simulation. We propose several strategies for generating the compressed history of a simulation. Finally, we present a simulation study of our algorithm.

We study the accuracy of simulated results for packet loss ratio and throughput for a table-driven routing protocol – `DSDV`. The results of the simulations are better than our previous results without the application of *compressed history*, in both aspects of speedup and accuracy. We conclude that with the aid of time-parallel simulation with *compressed history*, we can study relatively large wireless ad hoc networks consisting of a few thousand nodes for extended periods of time.

We plan to compare the strategies illustrated in Figure 2 and determine the best strategy to generate the compressed history of a simulation, and then to implement *compressed history* for simulation of on-demand routing protocols, such as `AODV` [10].

Acknowledgment

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. Model. and Comput. Simul.*, 5(4):269–305, 1995.

[2] L. Bölöni, D. Turgut, G. Wang, and D. Marinescu. Challenges and benefits of time-parallel simulation of wireless

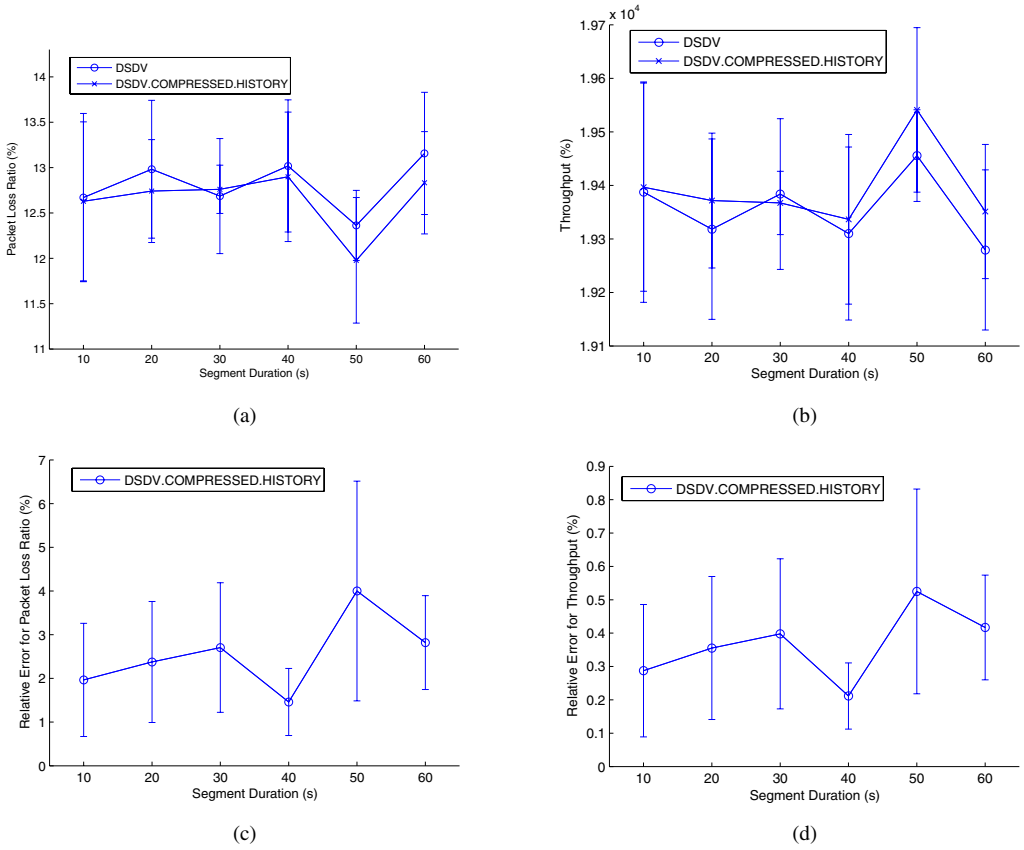

Figure 3. Time-parallel simulation of `DSDV` with *compressed history*. (a) Packet loss ratio function of the segment duration. (b) Throughput function of the segment duration. (c) Relative error for packet loss ratio function of the segment duration. (d) Relative error for throughput function of the segment duration.

ad hoc networks. In *Proc. 1st Int. Conf. on Perform. Evaluation Methodologies and Tools (Valuetools-2006)*, 2006.

[3] R. M. Fujimoto. Parallel discrete event simulation. *Commun. of ACM*, 33(10):30–53, 1990.

[4] M. Hoseyni-Nasab and S. Andradóttir. Time segmentation parallel simulation of tandem queues with manufacturing blocking. In *Proc. 1998 Winter Simul. Conf.*, 1998.

[5] T. Kiesling. Using approximation with time-parallel simulation. *Simulation*, 81(4):255–266, 2005.

[6] Y. Lin and E. D. Lazowska. A time-division algorithm for parallel simulation. *ACM Trans. Model. and Comput. Simul.*, 1(1):73–83, 1991.

[7] H. T. Mouftah and R. P. Sturgeon. Distributed discrete event simulation for communications networks. *IEEE J. on Selected Areas in Commun.*, 8(9):1723–1734, 1990.

[8] I. Nikolaidis, R. Fujimoto, and C. A. Cooper. Parallel simulation of high-speed network multiplexers. *IEEE Conf. on Decision and Control*, 3(1):2224–2229, 1993.

[9] C. E. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In *ACM Spec. Interest Group on Data Commun. (ACM SIGCOMM)*, pages 234–244, 1994.

[10] C. E. Perkins and E. M. Royer. Ad hoc On-demand Distance Vector Routing. In *Proc. 2nd IEEE Workshop on Mobile Comput. Syst. and Appl.*, pages 99–100, 1999.

[11] D. Turgut, G. Wang, L. Bölöni, and D. Marinescu. Speedup-precision tradeoffs in time-parallel simulation of wireless ad hoc networks. In *Proc. 10th IEEE Int. Symp. on Distrib. Simul. and Real Time Appl. (DS-RT)*, pages 265–268, 2006.

[12] VINT. 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. In *Accepted for publication in Proc. 2nd IEEE Int. Workshop on Perform. and Management of Wireless and Mobile Networks*, pages 730–737, 2006.

[14] J. J. Wang and M. Abrams. Determining initial states for time-parallel simulations. In *Proc. 7th workshop on Parallel and distrib. simul. (PADS '93)*, pages 19–26, 1993.

[15] PDNS – Parallel/Distributed NS. URL `http://www.cc.gatech.edu/computing/compass/pdns/`, 2004.