USC/Information Sciences Institute Research Report 00-522, April 2000,
submitted for publication to IEEE Computer

Expanding Confidence in Network Simulation

John Heidemann, Kevin Mills, Sri Kumar

Abstract

Networking research increasingly depends on simulation to investigate new protocol behavior, performance, and interactions. In spite of wide use of simulation, today there is no common understanding of what level of simulation validation is required for these tasks, and limited background of what validation techniques are being used and their effectiveness. This paper reports on discussions of these issues that arose from the Network Simulation Validation Workshop sponsored by DARPA and NIST in May 1999. We describe best-current-practices of general validation and validation of TCP, how scale and validation interact, and workshop consensus.

Network deployment will grow increasing complex as industry lashes together a mix of wired and wireless technologies into large-scale heterogeneous network architectures and as user applications and traffic continue to evolve. For example, this projected increase in complexity already affects Department of Defense combat networks, the Internet, and industrial wireless networks. Faced with this growing complexity, network designers and researchers almost universally use simulation in order to predict the expected performance of complex networks and to understand the behavior exhibited by networking protocols not originally designed to operate in such environments. Simulation is also increasingly used to predict the correctness and performance of developing protocols. In addition, the use of simulations now appears as a strict requirement in processes leading to international standards, such as the IMT-2000 standard for third-generation, wireless, cellular telephony.

This growing reliance on simulation raises the stakes with regard to establishing the correctness and predictive merits of specific simulation models. Yet no widely accepted practices and techniques exist to help validate network simulations and to evaluate the trustworthiness of their results. Early work in networking research and engineering involved both experimentation and mathematical modeling to prove feasibility and to establish bounds on expected performance. In the past 10 years, as networks have grown too large to allow easy experimentation and too complicated to admit easy tractable mathematical analysis, network simulation[1] has filled an increasingly important need, helping researchers and designers to understand the behavior and performance of protocols and networks. Today simulation is often used:

- to predict the performance of current networks and protocols in order to aid technology assessment and capacity planning and to demonstrate fulfillment of customer goals,
- to predict the expected behavior of new network protocols and designs through qualitative or quantitative estimates of performance or correctness, and
- to quickly explore a range of potential protocol designs through rapid evaluation and iteration.

For any of these purposes, the results produced from simulation, analytical, or hybrid models must be understood. *Validation* is the process of assuring that a model provides meaningful answers to the questions being investigated. (See Sidebar for a discussion of verification, validation, and accreditation.) Models often involve approximations or abstractions from reality; validation provides confidence that these approximations do not substantially alter the answers to the questions being posed. This implies that each set of ques-

---

[1] Of course, modern simulation models often also include analytical sub-models. Such hybrid models can be more effective than either simulation or analysis alone.

tions can require a distinct validation because a simulation might be valid for answering one question, while invalid for another. Modeling is not unique in requiring validation. Even laboratory experiments can prove invalid when they encompass unexpected effects, such as measurement artifacts, or when experiment results are extrapolated into inappropriate regions, such as predicting performance of a million-node network based on a hundred-node experiment.

Further, different situations can require different levels of validation; the level of validation required for a network simulation is influenced by the questions being asked and by the systems being used. Answers to qualitative questions (are lost packets recovered?) often require less complete validation than quantitative questions (how quickly are lost packets recovered?). Some domains seem more amenable to abstraction, as well. For example, simple delay-bandwidth-error models can often replace detailed physical and link layer simulations for high-speed wired networks with low bit error rates. Alternatively, a wireless network, which suffers the effects of fading, interference, and mobility, can show significant transmission losses and medium access delays; and, therefore, requires a more complex model to reflect interactions between protocols for the transport and physical/radio layers. Increasing use of simulation in the networking research community, along with the need to understand protocols in more complicated environments (for example, mixed wireless and wired networks), has raised the stakes with regard to validating network simulations.

In May 1999, the National Institute of Standards and Technology (NIST) and the Defense Advanced Research Projects Agency (DARPA) co-sponsored a workshop to discuss approaches to validate network simulations. The workshop brought together leading simulation practitioners from companies, such as AT&T, Lucent, ITT, Raytheon, Telcordia, and SAIC, as well as researchers from universities, including Carnegie-Mellon, Dartmouth, George Washington, Rutgers, UC Berkeley, UCLA, and USC/ISI. Workshop attendees submitted position papers addressing key issues with regard to simulation validation (see the acknowledgments section for a URL pointing to the papers). Discussions at the workshop revealed many approaches to validation currently pursued by practitioners and researchers. This paper summarizes for the community some of the conclusions of that workshop, offering insight into how validation applies in a networking context, suggesting some guidelines and examples for validation, and raising challenges for the community. Although this paper represents the opinions of the authors, we would like to thank all attendees of the workshop for their input.

A. VALIDATION IN NETWORK SIMULATIONS

When considering how to validate network simulations, one must first clarify what represents "ground truth". One obvious approach is to compare the simulation results to results from a particular real-world implementation of a network. This allows direct comparison of simulation results against live experiments. Direct comparison can work for small networks, especially given well-specified protocols. When network topologies are large or when protocols are under-specified, validation through direct comparison can prove difficult.

Traditionally, protocols have been specified only to the level necessary to ensure successful communication between nodes, and to obtain reasonable performance. This implies that many engineering decisions and optimizations may be left to protocol implementers. In most cases, different decisions lead to differences in performance, but without compromising the basic behavior encoded in the specification. For example, the details of acknowledgment timing are left as implementation decisions in the specification for TCP (see Request For Comments 1122). Such implementation decisions must be empirically determined or assumed when constructing a model for a specific protocol. As a result, protocol models typically embody behavior associated with specific implementations.

Comparison to particular protocol implementations might not be ideal in all cases, since a very accurate simulation can become outdated as protocols vary and evolve or as traffic mixes change. In these cases simulations may need to be validated against future, rather than current, implementations and traffic. Simulation users need to understand both what is provided in a simulator and what is appropriate for their experiments.

TCP provides an example where the specification admits a range of implementations with very different performance. Details of the acknowledgement algorithm and parameters such as window size and scaling can alter initial or steady-state throughput by a factor of 2-10. In this case, simulations may be validated against a specific implementation or against the performance envelope of the specification.

Protocol designs also evolve, and deployed implementations necessarily lag current research versions; simulations may track either. For example, the Reno implementation of TCP has known performance problems when multiple packets are lost in a single round-trip. Until standardization of the selective acknowledgement option in TCP, evaluations based on the then current Reno TCP could easily misrepresent obtainable performance.

Finally, the Internet has experienced dramatic changes in traffic mixes (for example, the growth of the web and possible growth of streaming real-time data). Validations against yesterday's traffic mix may miss the current situation, and validation against today’s traffic mix may misrepresent future patterns.

Given a choice of ground truth, either a specification or a particular implementation, validation methods must define metrics to compare simulation model results against that truth. A first step is to compare expected phenomena in the protocol. For example, TCP consists of several algorithms (such as windowed data transmission, slow-start, and fast retransmit). Testing these algorithms in simulation is akin to behavioral testing of a real-world implementation, and many of the same approaches can apply. In addition, time/event plots, packet animations, and trace comparisons are often useful tools in this process; however, finding general approaches to quantify differences among similar but not identical time/event plots remains an open research question. Successful behavior testing raises confidence that a simulated protocol will operate to specification.

Increasingly, model developers rely on visual comparisons among model outputs. While helpful, visual comparisons are limited in effectiveness because timing and behavioral differences are difficult to quantify visually, thus making it difficult to evaluate similarity.

Aggregate statistical measures, such as packets sent, throughput, and time-to-completion can provide an alternate useful picture. Aggregate measures should be chosen with care and used in conjunction with other approaches, though, since an improperly chosen metric can mischaracterize a comparison. For example, comparing average data sent over a period of time fails to capture differences in protocol burstiness.

Once a simulation has been validated under one set of conditions, sensitivity analysis helps understand how varying configurations change the accuracy of the simulation. For example, variations in how retransmission is handled may not be apparent if a TCP simulation is evaluated only under conditions of low loss. When considered on a large scale, network simulation presents an additional challenge, not addressed by sensitivity analysis, to verify that a simulation model exhibits specified behaviors regardless of variations in network topology, size, and traffic patterns. Such behaviors are sometimes called model invariants. Tools to assist the process of sensitivity analysis are an area of future work.

Finally, the extent, and therefore cost, of validation must be considered against the likely benefits. In some cases, detailed, expensive validation may be appropriate. Yet, in specific situations, it might prove impossible to achieve the desired level of validation no matter how much is spent. In other cases, extensive validation, while achievable, might well prove unnecessary. We have already described cases where comparison against an implementation is impossible or inappropriate. In general, more stable protocols, for which designs do not vary frequently or significantly, permit more specific validation. Ultimately, one must consider validation in the context of the research, and operational questions being considered. Validation of a simulation used to prove to a customer that a shipping product meets its specification might be much more exacting and costly than validation of a research simulation exploring dozens of possible protocol variants.

B. GUIDELINES FOR VALIDATION

A very useful result of the workshop was a better understanding of current practices the community is using to validate network simulations. One of the industry practitioners provided a concise summary of recommended practices as input to the workshop [Lubachevsky99], which we expand upon here.

- Validation is much easier when the model is focused on comparative, rather than absolute, behaviors. This is natural in many cases, where a new proposal is being compared against an existing scheme, already deployed.
- Design in as many means as possible for examining the state of the simulation, and use visual representations to their fullest. While careful statistical analysis is certainly valuable, more often than not, invalid behaviors will be recognized more quickly from viewing animations. Finding effective approaches to examine and visualize very large models (10,000 or more nodes), especially for small but significant differences, remains a research challenge. Such models demand integrated instrumentation with multi-stage filtering and classification of data.
- Various forms of models and implementations can emphasize different aspects of a networking system. For this reason, modelers should compare simulation results with as many alternate representations as possible. This might include laboratory experiments and field exercises, analytical models, and other, independently developed, simulations. Increasing the number of alternative representations against which a model is compared increases the likelihood that errors, inconsistencies, and invalid assumptions will be uncovered.
- Where the model involves interactions over time among various independent entities, be sure to introduce asynchrony where needed to mimic the operation of real systems. For example, each wireless basestation maintains an independent clock. These clocks drift over time. Modeling this behavior is often worth the extra effort.
- Simulation results must be reproducible. Many factors are important to promote reproducibility, including deterministic algorithms to generate pseudo-random number sequences, and mitigation of rounding errors from floating-point representations. Rounding errors can affect event concurrency, especially where optimistic synchronization is used when simulations are executed on parallel computer systems. In general, care must be taken to ensure that both time and causality are modeled accurately when parallel processing systems are used to execute simulations. Validation will prove impossible without the existence of appropriate reproducibility within a simulation.
- Where the size of the simulation must be reduced to execute within memory and CPU cycle limitations, care must be exercised to avoid introducing artificial boundaries into the model. For example, transient startup effects or an artificial physical topology can introduce inaccuracies.

Beyond these validation guidelines for network simulation practitioners, the workshop attendees discussed steps that could be taken to improve validation with respect to published research results. As an important step toward improving the quality of validation in the research community, simulation results should be reproducible. A paper employing simulation studies should be accompanied by a link to a publicly available and well-instrumented model (in either source or binary form) in order to allow independent confirmation of the results. Public availability of simulation source code and model protocol libraries is also important to allow examination for correct operation, and to permit modification for use in additional situations.

C. SCALE AND VALIDATION

Validation of small simulations remains challenging. Validation of large-scale simulations is even more difficult. Given the scope of today's Internet, understanding protocol behavior with large numbers of nodes, varied traffic levels, and with more or less detail, remain important questions. Another dimension of scale is the number of independently developed components within a model.

Two approaches to large-scale simulation -- parallel execution and abstraction -- are complementary. Several simulators support parallelism [Bagrodia98a, Cowie99a]. The use of machines with multiple CPUs or clusters of workstations brings more horsepower and memory to bear on a given problem, allowing 10-100x larger simulations. A complementary approach is the use of abstraction to factor out details unimportant to the simulation at hand [Huang98a]. Abstraction has been used to provide 100-1000-fold increases in possible simulation size for particular research questions. That said, abstraction must be applied with care because, in the absence of an explicit mathematic derivation, an abstracted model must still be validated against a more detailed model running at slower speed, or against field experiments of sufficiently large scale. Further, new collective phenomena might appear as networks increase in size.

Large-scale simulations can also build upon small-scale validated sub-models. One approach is recursive composition: begin with well-validated components, and a well-validated composition framework; then generate large models using hierarchical composition [Cowie99a]. Another approach is to compare detailed and abstract simulations at small scales, then generate large abstract scenarios [Huang98a]. Both construction and abstraction assume that potential inaccuracies in small-scale scenarios are not magnified at larger scales. This assumption must still be validated on a case-by-case basis. Preliminary research results suggest that detailed simulations can accurately reproduce Internet-like traffic, as described below in the section on “aggregate statistics”.

D. CASE STUDY: TCP MODELS

The TCP models in simulators, such as *ns* [Bajaj99a], represent a case study for validation of network simulation within the networking research community. Unlike many simulation models, the one-way TCP models included in *ns* do not attempt to model a particular TCP implementation or specification, but instead model a simplified protocol supporting one-way data transfer without message fragmentation. These models do however represent the details of the algorithms that make up TCP, including slow-start and fast retransmit. This design was chosen to support easy experimentation with TCP variants. These models have been validated in several different ways.

Phenomenon validation: The model and algorithms implemented in *ns* one-way TCP are described in a paper by Fall and Floyd [Fall96a]. To insure *ns* correctly implements this model, *ns* developers regularly validate the current implementation against this model. Initially a human expert compared current output (in the form of time/event graphs) to the model. Today this output is compared automatically (byte-for-byte) against saved output. The first approach is robust to minor simulator changes but requires expert analysis, the second is automatic but brittle.

These tests have also been applied to the independently written implementations of Tahoe and Reno TCP in the Scalable Simulation Facility (SSF) [Cowie99a]. Validation of SSF TCP has been patterned after the testing scenarios developed for use with *ns*. Although completely different in design and implementation, SSF TCPs produce identical results as *ns*. Because SSF shares no code with *ns*, these results provide increased confidence that the TCP implementations in both simulators can be regarded as trusted building blocks for inclusion in larger models. The success of this approach illustrates the importance of widely accepted test scenarios that include expected reference results.

Kernel validation: A subset of the *ns* TCP models has been ported to run over the Parsec simulation engine in addition to the native *ns* simulator. When all external services are held constant (including, for example, the random number generator), the two simulation kernels generated nearly identical outputs running the same scenario. This example of n-version programming argues against bugs in the exercised portions of the two simulator kernels.

End-to-end statistical validation: Two examples illustrate the use of end-to-end statistics to validate TCP simulation modules. Ya Xu has made small-scale comparisons of TCP throughput and traces in *ns* and on CAIRN, a high-speed network testbed. One result of these experiments is a better understanding of the care that must be taken when conducting real-world experiments. The expected throughput, as predicted by the simulation, was achieved only after iterations addressing a range of bugs and details in the end-node operating system, link configurations, and test applications deployed on CAIRN. In effect, in this case the experimental network had to be corrected to conform to the expected results from simulation and analysis. This example illustrates the need to validate experimental systems as well as simulation models.

In another validation experiment, within the challenging domain of wireless communications, the Monarch project compared simulated and emulated versions of *ns* TCP traffic operating over wireless and ad hoc routing simulation modules developed at CMU. In the comparison, identical end-to-end throughput was achieved; however, the temporal behavior of individual packets was not identical [Johnson99a]. So, in this case, the simulation proved valid for addressing questions of throughput, but invalid for addressing detailed questions of packet delay.

Aggregate statistics: Finally, researchers at AT&T have reproduced ISP-like traffic in *ns* and compared it to real-world traces using wavelet analysis [Feldmann99a]. The technique of wavelet analysis shows similarity between simulated and real traffic across a wide range of timescales and for reasonably large scenarios (400 nodes and 10,000 or more web requests). More importantly, their simulations are accurate enough to investigate what aspects of TCP influence aggregate network behaviors, an experiment impossible to undertake in the real world.

E. SUMMARY OF WORKSHOP CONSENSUS

During the workshop, a consensus developed around the following points.

(1) Researchers presenting papers based on simulation studies need to consistently present the approach used to validate their models. Ideally, simulations should be made publicly available concurrent with related papers. Tremendous positive community benefits can accrue through sharing knowledge at this level, both in terms of simulation development as well as developing widely accepted practices for validation. Such sharing benefits both the private and the public sectors, as new models for traffic, network protocols, and network control emerge in the future. Working with the community, DARPA and NIST plan to create a web-based resource for network simulation modeling knowledge. This resource will be open to everyone working in the network simulation community.

(2) Simulation users would benefit from standard approaches to document the model underlying a given simulation software module, including a description of how that software has been validated.

(3) The community needs a better understanding of the levels of validation required in different circumstances. For example, validation against a specific implementation can be mandatory or inappropriate, depending on the question at hand.

(4) The community should continue working towards platform-independent data formats (such as, tcpdump) and platform-independent validation tools.

(5) Finally, the set of available validation tools should be improved. Smarter tools to compare traces would be valuable, as would more sophisticated (multi-resolution) statistical techniques. A wider set of multi-simulator test scenarios could also prove helpful.

F. CONCLUSIONS

Increasingly, commercial and public organizations deploy large-scale networks incorporating heterogeneous technologies, such as multi-wavelength optical fibers and wireless communications links. In most cases, the Internet protocol suite is used over these diverse networks to provide an infrastructure for distributed applications and network services. To accommodate the growing challenges inherent in connecting diverse network technologies together, while also providing customers with attractive services, industrial and academic researchers continue to explore new network protocols. Whether deploying complex networks or experimenting with new protocol designs, networking engineers and researchers must increasingly turn to simulation modeling. Given their complexity, the networks being designed today may not be amenable to full analysis by mathematical models alone. A more productive approach may be to suitably incorporate mathematical models as subsystems in discrete-event simulations.

The growing role for simulation raises the stakes for validation of the models being developed and used. The workshop discussed in this paper provided a first step toward a larger effort required among the network engineering and research community. The workshop captured the current state of practice, and identified some of the difficult issues that must be resolved before network simulation modeling can reach a mature state. Future funded research that involves simulation modeling of networks should move the community toward the points of consensus identified at this workshop. Also as a concrete step forward, standards-setting organizations, such as the IETF, should encourage the creation of models and suites of test scenarios, together with expected behaviors, to be included as part of the specification of all protocols. The test scenarios should be described in a form that can be applied to simulation models, as well as full implementations.

ACKNOWLEDGEMENTS

Although this paper represents the opinions of the authors, we would like to thank all attendees of the workshop for their input. Workshop presenters were Rajive Bagrodia, Ronald Boisvert, Sally Floyd, Janet Jackel, David B. Johnson, Geoffrey Kelsch, Stefan Leigh, Boris D. Lubachevsky, Donald Porter, Gary Warren, and Pat Wirth. In addition Rajive Bagrodia, Jim Cowie, Sally Floyd, and Andrew Ogielski suggested improvements to this paper. A complete list of attendees, as well as all related papers, can be found on the workshop web site.(http://www.dyncorp-is.com/darpa/meetings/nist99may/index.html). .

REFERENCES

[Bagrodia98a] Rajive Bagrodia, Richard Meyer, Mineo Takai, Yu-an Chen, Xian Zeng, Jay Martin, and Ha Yoon Song. PARSEC: A Parallel Simulation Environment for Complex Systems. IEEE Computer, 31(10), 77-85, October, 1998.
http://dlib.computer.org/co/books/co1998/pdf/rx077.pdf

[Bajaj99a] Sandeep Bajaj, Lee Breslau, Deborah Estrin, Kevin Fall, Sally Floyd, Padma Haldar, Mark Handley, Ahmed Helmy, John Heidemann, Polly Huang, Satish Kumar, Steven McCanne, Reza Rejaie, Puneet Sharma, Kannan Varadhan, Ya Xu, Haobo Yu, and Daniel Zappala. Improving Simulation for Network Research. Technical Report 99-702, University of Southern California, March, 1999.

http://www.isi.edu/~johnh/PAPERS/Bajaj99a.html

[Balci94] Osman Balci. Validation, verification, and testing techniques throughout the life cycle of a simulation study. Annals of Operations Research 53, pp. 121-173.

[Cowie99a] James H. Cowie, David M. Nicol, and Andy T. Ogielski. Modeling the Global Internet. *Computing in Science & Engineering*, 30-36, January, 1999.
http://dimacs.rutgers.edu/Projects/Simulations/darpa/Papers/internet-cise99.pdf

[Feldmann99a] Anja Feldmann, Anna C. Gilbert, Polly Huang, and Walter Willinger. Dynamics of IP traffic: A study of the role of variability and the impact of control. In Proceedings of the ACM SIGCOMM, Cambridge, MA, USA, ACM. August, 1999.

[Fall96a] Kevin Fall and Sally Floyd, "Simulation-based Comparisons of Tahoe, Reno, and SACK TCP", Computer Communication Review, V. 26, N. 3, July 1996, pp. 5-12.
http://www.aciri.org/floyd/papers/sacks.pdf

[Huang98a] Polly Huang, Deborah Estrin, and John Heidemann. Enabling Large-scale simulations: selective abstraction approach to the study of multicast protocols. In Proceedings of the International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, Montreal, Canada, IEEE. July, 1998.
http://www.isi.edu/~johnh/PAPERS/Huang98a.html

[Johnson99a] David B. Johnson. Validation of Wireless and Mobile Network Models and Simulation. In Proceedings of the DARPA/NIST Network Simulation Validation Workshop, Fairfax, Virginia, USA, May, 1999.

[Lubachevsky99] Boris D. Lubachevsky. Recipes for Validation. In Proceedings of the NIST/DARPA Workshop on Validation of Network Simulations, Fairfax, Virginia May, 1999.
http://www.dyncorp-is.com/darpa/meetings/nist99may/papers/Lubachevsky.pdf

[Page97] Ernest H. Page, Bradford S. Canova, and John A. Tufarolo. A Case Study of Verification, Validation, and Accreditation for Advanced Distributed Simulation. In ACM Transactions on Modeling and Computer Simulation. Vol. 7, No. 3, July 1997, pp. 393-424.
http://www.mitre.org/pubs/journal/best_papers_97/page_case_study.html

G. SIDEBAR: VERIFICATION, VALIDATION, AND ACCREDITATION

The Department of Defense has a long-standing interest in methods and techniques for the verification, validation, and accreditation (V, V, & A) of simulation models [Balci94, Page97]. Although there is debate about the exact definition of these terms, general agreement exists surrounding the intent of the methods and techniques associated with each term.

Validation is a process to evaluate how accurately a model reflects the real-world phenomenon that it purports to represent. As we discuss, the degree of accuracy required by the validation depends on its specific intended use. For example, if a model is used to compare numerous design choices for new protocols, then the model need only be accurate enough to distinguish effectively between the performance and behavior of the various designs being compared. On the other hand, if a model is used to evaluate engineering alternatives against specific performance objectives and traffic loads, then, for the characteristics of interest, the model might need to exhibit accuracy within a statistically bounded range.

Verification is a process to evaluate how faithfully the implementation of a model matches the developer's intent, as expressed by conceptual descriptions and specifications, provided either in natural language or a formal notation. In effect, verification is akin to software function testing. While verification does not establish the accuracy of the predictive power of a simulation model, verification can uncover errors in coding and errors in implementation of protocol mechanisms. These errors may or may not invalidate the model. For example, an error might occur in the statistical representation of traffic. If the intent is to compare the performance of different protocols against identical offered load, then this error may have little effect on model validity. On the other hand, if the intent is to establish the absolute performance of a network design given a representative usage scenario, then the same error could well make the model invalid. Still, verification aims to catch programming and coding errors, rather than errors in the accuracy of model results.

Accreditation, a term often used by government agencies such as the Department of Defense or the Federal Aviation Administration, denotes a process leading to an official declaration that a given software program is fit for its intended use. In the absence of technical solutions that can guarantee that a software model is free from errors and will provide valid predictions, accreditation usually focuses on an external, or third-party, review of the processes used to verify and validate a model. The successful outcome of most accreditation processes is a written certificate signed by a recognized authority that attests that a prescribed set of processes was correctly applied during the development and testing of a simulation model.