

A Study of Research Tools and Techniques in Network Congestion Control

Vandana Kushwaha

Department of Computer Science, Banaras Hindu University, India

Abstract

The major requirement of Computer Network Research and development process is to design a new protocol or algorithm and in further steps to validate and verify their correctness of action and their deployment feasibility. We have several tool and techniques which are available for the research in the area of computer network congestion control. The purpose of this study is to explore various possible research tools and techniques available for congestion control in computer network.

1. Introduction

Design, development and evaluation of network congestion control protocols are a complex task involving various phases. Among them the experiment and evaluation phases, which eventually provides a global view, are vital steps in research and development process of distributed applications and communication protocols. In this context, three standard methods of research and development commonly used are *simulation, emulation* and *live experimentation* [1,2].

There are large number of research groups who employed Simulation using NS-2 for the design and evaluation of congestion control approaches. Wang et al [13] (2007) have proposed a framework for benchmarking TCP variants based on the NS-2 network simulator. Rhee et al. [14] have used a testbed based on Dummynet for experimenting and validating high speed TCP variants like BIC-TCP and CUBIC etc. Leith et al. have also worked on an experimental testbed based on Dummynet network emulator with their experiments on TCP [15]. In [9,10], Grid'5000 has been used for experimenting different TCP stacks and several types of workload corresponding to realistic grid computing and data-center applications. Wan-in-Lab [6] is another experimental networking testbed aimed at developing, testing and evaluating new communications protocols and technologies like FAST or TCP MaxNet [16]. In addition to that some more tools like NSBM [17], CAVT [18] are available for educational purpose which help the students to easily grasp the concepts of congestion control.

The purpose of this study is to review the major research tools and techniques for network Congestion control. An attempt has been made to categorize major research tools and techniques for network congestion

control by considering their features, relative merits and demerits. In this way this study will trace a better picture of major issues, challenges and possible solutions related to network congestion control research tools and techniques.

The paper is organized as follows: In Section 2, a brief review of research techniques used for network congestion control, has been mentioned. Section 3 presents the main observations found during this study. Finally Section 4 concludes the paper.

2. Techniques for Computer Network Research

In this section a concise summary of Network Research and Development tools are presented. The purpose is to elaborate various tools used for computer network research and development.

Simulation is a very efficient and economical way to experiment with protocols. Network simulation typically employs ad-hoc model and logical event-driven techniques. Standard tools such as NS-2 [3] or OPNET [4] provide a core simulation engine, as well as a large set of protocol models. These simulation tools allow experiments to be done economically with low cost. Simulators uses specific modelling techniques which simplify the studied problem by concentrating on the most critical issues related to design and development of communication protocols. For instance, network simulation using ns-2 has been used by most of the researchers to design and development of router based congestion control algorithms. However, simulation tools do not work in real-time environment because they are based on virtual clocking scheme. One more critical issue while using simulator is to verify that the services and performances offered by the simulation model are either consistent with the real experimental implementation of the protocol or not. Ultimately, to realize a real-time evaluation of a communication protocol, only two solutions are left: *live testing* and *network emulation*.

Live testing is a conventional method to test and debug network protocol during the implementation stage by using real hardware and/or software components. The protocol can be tested either on ad-hoc test bed using real equipments or on real target network. However, this approach is expensive if our network domain is a wide area network using

satellite communication. In addition, there are some situations when it is not possible to use this approach easily because the new technology support is not yet validated or available, e.g. when developing an application over a new satellite transmission technology that is not yet operating. Using real technology on target operational network has been widely deployed. An example of this scheme is well illustrated by PlanetLab [5]. Wan-In-Lab [6] is a testbed of the California Institute of Technology. It is built around a 2400 km optic fiber cable and arrays of optical switches to construct networks with variable length and RTT. Users can upload experimental kernels instrumented with the Web100 [7] tools, and run a set of predefined tests. Grid'5000 [8] is an experimental grid platform currently gathering 3500 processors over nine geographically distributed sites in France. The network infrastructure is an interconnection of LANs (*i.e.* grid sites) and a 10Gbps optical virtual private network (VPN). The particularity of this testbed is that it allows researchers to dynamically deploy any OS image or TCP stack on any end host that is part of the testbed [9, 10]. Nevertheless, this approach is considered to be very expensive and inflexible to evaluate all aspects of the protocol being tested.

Emulation is considered to be a combination of both simulation and live testing. Since several years, progresses in high speed processing and networking have allowed the rapid development of network emulators, such as Dummynet [11], NIST Net [12]. This approach involves executing and measuring real protocols and application implementations over a

certain network where part of the communication architecture is simulated in real-time. The aim of emulation is to allow a distributed software to run either in realistic conditions (e.g. over a satellite network) or specific conditions (e.g. when specific packets are dropped such as SYN packets in the TCP case). It is used to achieve experiments using both real protocol implementations and network models. Basically, this allows the creation of a controlled communication environment. This communication environment can produce specific target behaviours in terms of quality of service. Additionally, emulation aims at providing "artificial impairments" on the network to test particularities of the experimented protocol. These impairments include losing specific packets, reducing the network bandwidth with a specific timing or introducing delay over the network. Emulation is particularly useful in the debugging and testing phase of a system.

3. Some Observations on Tools and Techniques for Network Congestion Control

This section summarizes the observations, during this study, in the form of tables. A summary of tools used for congestion control research and development is mentioned in Table.1. In the table, techniques are categorized in four parts that is network simulator, network emulator, real test beds and other approaches. Network Simulators are further categorized in two sub parts Open source and Commercial.

Table 1. Tools used for congestion control research and development

	Tools		Reference
Network Simulator	Open source	Ns-2, Ns-3	[3]
		J-Sim	[19]
		GTNetS	[20]
		Omnet++	[21]
		REAL	[22]
	Commercial	QualNet	[23]
		NetSim	[24]
OPNET		[4]	
Network Emulator	NetEm	[25]	
	Dummynet	[11]	
	NIST Net	[12, 26]	
Real test-beds	Wan-In-Lab	[6]	
	Grid'5000	[8]	
	PlanetLab	[5]	
Others	Web-100	[7]	
	NSBM (Network Simulation by Mouse)	[17]	
	CAVT (Congestion Avoidance Visualization tool)	[18]	

3.1. Network Simulation Tools

Several tools are available for network simulation under open source as well as commercial categories.

The main features of these tools are listed in Table.2. Selection of appropriate network simulator depends upon the nature and requirement of network under consideration.

Table 2. Features of Network Simulation Tools

Network Simulator		Features
Open Source	Ns-2	<ol style="list-style-type: none"> 1. object-oriented, discrete event driven network simulator 2. uses C++ and OTcl Programming 3. separates control path implementations from the data path implementation 4. event scheduler
	Ns-3	<ol style="list-style-type: none"> 1. not backward-compatible with NS2 2. uses C++ and Python programming 3. alignment with real systems 4. support for virtualization
	J-Sim	<ol style="list-style-type: none"> 1. a Java-based simulation system 2. based on the component-based software architecture 3. uses Java and Tcl Programming
	Omnet++	<ol style="list-style-type: none"> 1. component-based architecture 2. compiler for the NED topology description language 3. graphical network editor 4. GUI /Command line interface for simulation execution 5. graphical output vector plotting tool
	REAL	<ol style="list-style-type: none"> 1. study the dynamic behavior of flow and congestion control schemes 2. uses C programming language
Commercial	QualNet	<ol style="list-style-type: none"> 1. commercial network simulator from Scalable Network Technologies 2. ultra high- fidelity network simulation software 3. predicts wireless, wired and mixed-platform network and networking device performance 4. simulate large, heterogeneous networks and the distributed applications
	NetSim	<ol style="list-style-type: none"> 1. object-oriented system modeling and simulation environment 2. simulates Cisco Systems networking hardware and software 3. support simulation and analysis of voice and data communication scenarios for High Frequency Global Communication Systems
	OPNET	<ol style="list-style-type: none"> 1. fast discrete event, hybrid and analytical simulation engine 2. object-oriented and hierarchical modeling environment 3. scalable wireless simulations support 4. 32-bit and 64-bit parallel simulation kernel 5. grid computing support

3.2. Network Emulation Tools

Typical network emulation tools include NS2 which is a popular network simulator that can also be used as a limited-functionality emulator. In contrast, typical network emulators specifically designed for network emulation are NetEm, Dummynet, and NISTNet etc. The main features of network

emulation tools are listed in Table.3. Selection of appropriate network emulator depends upon the network architecture, required functionality and timer granularity.

Table 3. Features of Network Emulation Tools

Network Emulation Tools	Features
NetEm	<ol style="list-style-type: none"> 1. test protocols by emulating the properties of wide area networks 2. enhancement of the Linux traffic control facilities 3. emulates variable delay, loss, duplication and re-ordering 4. timer granularity is the major limitation 5. system clock (up to 1 KHz) or high resolution timers 6. Included in Linux 2.6
Dumynet	<ol style="list-style-type: none"> 1. live network emulation tool 2. powerful bandwidth management functionality 3. doesn't allow to emulate degraded network conditions 4. uses system clock (up to 10 KHz) 5. Included in FreeBSD
NIST Net	<ol style="list-style-type: none"> 1. general-purpose tool for emulating performance dynamics in IP networks 3. can emulate the critical end-to-end performance characteristics 4. uses Real time clock 5. available for Linux 2.4 and 2.6

3.3. Real test-bed Tools

There are many real test-beds available for implementation of computer network algorithms. The main features of these real test-beds are listed in

Table 4. Selection of appropriate real test-bed depends upon the architectural behavior of the network.

Table 4. Features of Real test-bed Tools

Real test-beds Tools	Features
Wan-In-Lab	<ol style="list-style-type: none"> 1. hardware test-bed for the design, development, testing and evaluation of high speed network protocols 2. provide a realistic yet controlled environment 3. enables detailed monitoring of all aspects of protocol operation 4. uses real carrier-class networking hardware to avoid the artifacts introduced by network simulation and emulation
Grid'5000	<ol style="list-style-type: none"> 1. experimentation testbed to study large scale parallel or distributed systems 2. an experimental Grid platform gathering 8 sites geographically distributed in France. 3. allow experiments in all the software layers between the network protocols up to the applications.
PlanetLab	<ol style="list-style-type: none"> 1. Testbed for geographically distributed overlay network 2. global research network that supports the development of new network services 3. used for developing new technologies for distributed storage, network mapping, peer-to-peer systems, distributed hash tables, and query processing.

4. Conclusion

This work explores the available tools and techniques used for computer network research and development. Out of the three techniques: simulation, emulation and live testing, network simulation is cheap and provides quick results. However, if a simulation is not properly designed, there is a large difference between simulation result and actual

result. On the other hand live testing method is very authentic but lack of reproducibility and cost is the major factors. Network Emulation lies between the above two techniques, because it enables us to design a controlled experiment with high degree of reproducibility and provide a mechanism to work on real system effectively.

References

1. E. Lochin, T. Perennou, and L. Dairaine :When should I use network emulation? In:Annals of Telecommunications (Online), pp. 1-9, July 2011.
2. R. Guillier and P. Vicat-Blanc Primet :Methodologies and tools for exploring transport protocols in the context of high-speed networks In: IEEE TCSC Doctoral Symposium, May 2008.
3. Ns-2 Network Simulator: [Online], Available: <http://www.isi.edu/nsnam/ns/>
4. Opnettechnologies: 2001. [Online], Available: <http://www.opnet.com>,
5. L. Peterson, D. Culler, T. Anderson, and T. Roscoe:A blueprint for introducing disruptive technology into the internetIn:1st Workshop on Hot Topics in Networks (HotNets-I), Princeton, New Jersey, USA, October 2002.
6. G. S. Lee, L. L. H. Andrew, A. Tang and S. H. Low:Wan-in-lab: Motivation, deployment and experiments In:PFLDnet'07, February 2007.
7. M. Mathis, J. Heffner, R. Reddy:Web100: extended TCP instrumentation for research, education and diagnosisIn:ACM Comput. Commun. Rev. 33, pp. 69–79, 2003.
8. R. Bolze et al.: Grid'5000: a large scale and highly reconfigurable experimental grid testbed In:International Journal of High Performance Computing.
9. R. Guillier et al.:A study of large flow interactions in high-speed shared networks with grid5000 and gtrenet-1 In: PFLDnet'07, February 2007.
10. R. Guillier, S. Soudan, and P. Vicat-Blanc Primet:TCP variants and transfer time predictability in very high speed networks In:Infocom 2007 High Speed Networks Workshop, May 2007.
11. Luigi Rizzo :Dummynet: a simple approach to the evaluation of network protocols In :ACM Computer Communication Review, 27(1):31–41, 1997.
12. M. Carson and D. Santay:NIST Net: A Linux-Based Network Emulation ToolIn :ACM Computer Communication Review, 2003.
13. G. Wang, Y. Xia, and D. Harrison:An NS2 TCP Evaluation Tool Suite In: IETF Internet Draft, [Online], Available: <http://www.icir.org/tmrg/draft-irtf-tmrg-ns2-tcp-tool-00.txt>, April 2007.
14. S. Ha, L. Le, I. Rhee, and L. Xu:A Step toward Realistic Performance Evaluation of High-Speed TCP Variants In:Elsevier Computer Networks (COMNET) Journal, Special issue on "Hot topics in transport protocols for very fast and very long distance networks", 2006.
15. Y. T. Li, D. Leith, and R. N. Shorten. : Experimental Evaluation of TCP Protocols for High-Speed Networks In: Transactions on Networking, June 2006.
16. R. Guillier and P. Vicat-Blanc Primet :Methodologies and tools for exploring transport protocols in the context of high-speed networks In: IEEE TCSC Doctoral Symposium, May 2008.
17. M. Welzl, M. Ali, and S. Hessler:Network Simulation By Mouse (NSBM): A GUI Approach for Teaching Computer Networks with the ns-2 Simulator In:Proceedings of the International Conference on Interactive Computer Aided Learning (ICL 2006), September 2006.
18. M. Welzl and M. Muhlhauser:CAVT: a congestion avoidance visualization tool In:SIGCOMM Comput. Commun. Rev., vol. 33, no. 3, pp. 95–101, 2003.
19. J-Sim :[Online], Available: <http://www.j-sim.org/>
20. The Georgia Tech Network Simulator (GTNetS):[Online], Available: <http://www.ece.gatech.edu/research/labs/MANIACS/GTNetS/>
21. OMNeT++ Network Simulation Framework : [Online], Available: <http://www.omnetpp.org>.
22. S. Keshav:REAL: A Network Simulator In :tech. report 88/472, Univ. California, Berkeley, 1988.
23. Scalable Network Technologies,Qualnetsimulator: Software Package, 2003. [Online].Available: <http://www.qualnet.com>
24. NetSim Simulator: [Online], Available: <http://www.tetcos.com/>
25. Netem.: [Online], Available: <http://www.linuxfoundation.org/collaborate/workgroups/networking/netem>".
26. Nistnet: [Online], Available: <http://snad.ncsl.nist.gov/nistnet/>.