ITERATIVE SYSTEM FOR SIMULATION OF E2E TRANSPORT PROTOCOLS IN HETEROGENEOUS NETWORKS

Biljana Stojcevska[1], Oliver Popov[2], Aleksandar Milenkoski[1]

[1]School of Computer Science and Information Technology, University American College Skopje, Treta makedonska brigada, bb, 1000 Skopje, Macedonia
[2]Department of Computer and Systems Sciences, Stockholm University and KTH, Forum 100, Isafjordsgatan 39, SE- 164 40 Kista, Sweden

stojcevska@uacs.edu.mk (Biljana Stojcevska)

Abstract

The Internet is characterized by the ever-increasing number of connections, nodes and places of presence, kinds of topologies, transmission media technologies, the yearly growth in user communities by magnitudes, the variety and the potential of innovative applications, combined with the economic, social, cultural and political ramifications. While constantly being measured and evaluated, explored and studied, the Internet is arguably impossible to be used for experimentation and investigations in order to test and validate the modifications of the existing network protocols or the introduction of new ones. Studies are commonly done using laboratory conditions through tests beds or simulation environments. The ns-2 network simulator is a well-known open source tool extensively used and extended by the academia, who also needs a standardized set of topologies, traffics, and metrics to validate and verify the results of their efforts. The tcpeval is one of these tools tool designed to provide a standard starting research framework. The goal of the paper is twofold: using ns-2 and tcpeval as tools for creating a simulation platform enriched with data gathering system and evaluation of the performance of an end-to-end transport scheme based on an original TCP modification.

Keywords: Simulation, E2E Protocols, Transport Protocols, ns-2, tcpeval, Heterogeneous Networks

Presenting Author's biography

Biljana Stojcevska is a teaching assistant at the School of Computer Science and IT at the University American College Skopje and a PhD student. She received her MCs degree in area Data Communication Networks and currently is a PhD student. Her academic interests are heterogeneous networking environments, Internet technology with emphasis on the E2E behavior of transport protocols and their role in congestion management, and Operating Systems.

1 Introduction

Since its conception, the Internet has been characterized by continues changes in the transmission technology, modifications in the design and engineering of the protocols across all the layers, new types of applications and services, and extensive worldwide user adoption. The advent of the wireless connectivity and the potential for mobility has had a profound influence on the omnipresence, accessibility and availability, as well as on the performance of the protocols that operate on the transport and network layers.

In general, wireless media is prone to a higher rate of transmission errors. TCP [1] misinterprets the increase in the number of errors as a sign of network congestion. To avoid further degradation in the network performance, the sending speed is reduced, which leads to underutilization of the network resources. The prevalence of TCP as the dominant transport protocol and its effect on both global and local Internet performance has substantially contributed to the need for an intensive research of this problem in the past two decades.

In this paper we present a simulation system for investigation of an end-to-end TCP modifications intended to be deployed in heterogeneous networks. The system is built on the well known ns-2 [2] simulation platform together with the tcpeval tool [3], especially designed for assessment and evaluation of TCP protocol and all its variants.

The TCP SACK PR [4], [5] protocol is an end-to-end TCP variant for operation in heterogeneous networks. It is indeed the SACK modification which improves the performance of TCP SACK [6] in networks that contain links with high random error packet losses.

Researching and developing mechanisms that would eventually become part of the Global Internet is a daunting task. Many methods can be used, such as performance measurements, test-beds, and modeling. But none of them can be used to evaluate the performance of new protocols and algorithms that are not implemented yet. That is why simulation has a very important role in the process of researching and improving the current Internet.

The rest of the paper is organized as follows: In Section 2 we represent ns-2 and tcpeval. Then we describe our methodology of research, the types of simulations, and the confidence levels of the obtained results. In Section 3, we describe the detailed experimental setup in the data gathering system that we have built over ns-2 and tcpeval. The performance of the data gathering system is presented in Section 4. In Section 5 we illustrate the obtained simulation results, while in Section 6 we give our conclusions and plans for future work.

2 Evaluation tools and methodology

2.1 ns-2 and tcpeval

As an open-source discrete event simulator ns-2 is widely used by networking research community. It provides support for simulation of TCP, routing, and multicast protocols over wired and wireless networks and is based on a rich object-oriented programming environment. The platform is subject to continuous upgrade and extensions by its users in the academic and commercial communities, which makes it highly popular and widely used.

One of the most researched issues using ns-2 tool has been the problem of congestion control on the Internet. The efforts lead to defining a set of well-defined standardized test cases and metrics that should be used for a quick initial evaluation of any new proposed mechanism. The idea that has been around for a long time was finally proposed as a common test-suite in [7].

The tcpeval framework is built to help researchers save time by rolling over their research within an existing framework. It includes several standard topologies and traffic models, and measures some of the most relevant metrics present in any TCP evaluation. It has options for automatic generation of simulation statistics and graphs, and is distributed as a patch for ns-2.

2.2 Evaluation methodology

Conducting an investigation by using simulation of the performance of a protocol as complex as TCP asks for a systematic and consistent methodology. We are following the steps described in [8] shown in Figure 1.

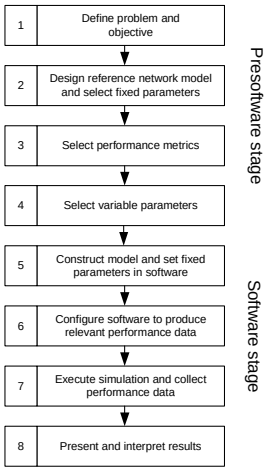

Fig. 1: The Steps of the Evaluation Methodology

The objective of the study was the performance evaluation of an end-to-end TCP variant named SACK PR, which is designed to operate in networks with lossy links. All network models used are provided by tcpeval. The main performance metrics were TCP throughput and the Jain's Fairness Index [9]

of the protocol, while the main variable parameter was Packet Error Rate (PER). The protocol is implemented as an object in ns-2 and then used to perform a large number of experiments. We give a detailed description of this system in Section 3.

2.3 Confidence Level of the Results

The fact that we use simulation as a tool means that the output is an estimate of a certain value. Therefore, we have to provide a level of confidence of the experiments outcome.

For this purpose, we use the Confidence Level Formula [8]. The goal is to estimate the actual value of a random variable X based on a sequence of observations x1, x2, x3,…,xn. The estimate is calculated as a mean of the numbers in the sequence:

$$\overline{X}(n) = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad (1)$$

The confidence in an estimate can be measured by the probability:

$$P(\left|\overline{X}(n) - \mu\right| < \delta = 1 - \alpha, \text{ where } 0 < \alpha < 1 \qquad (2)$$

Where $(1-\alpha)$ is the confidence level of the estimator $\overline{X}(n)$ and δ is the half-width of the confidence interval. The meaning of the above equations is that one is $(1-\alpha)$*100% confident that the true average of the variable is in the interval

$$(\overline{X}(n) - \delta\ ,\ \overline{X}(n) + \delta) \qquad (3)$$

Usually, the value of α is chosen to be 0.05 and in that case the desired confidence level is 95%. The first step is to calculate the half-width δ of the confidence interval, and then, the relative precision ε is obtained as:

$$\varepsilon = \frac{\delta}{\overline{X}(n)} \qquad (4)$$

In most cases the relative-precision is chosen to be 5%, and the series of experiments is continued until the 5% precision is achieved with 95% confidence. The same method is used to provide the statistical level of confidence concerning our results.

3 The data gathering system

3.1 Building a data gathering system by merging ns-2 and tcpeval

We have implemented SACK PR as a TCP Agent in the ns-2 class hierarchy. It inherits all the SACK functionality, so in a way it can be treated as a TCP SACK clone. The modified protocol belongs to the class of the Loss Detection Algorithms: it measures the RTT on each ACK arrival and decides when a receipt of three DACKS is a result of a random loss or a congestion loss. Detailed description of the algorithms included in SACK PR is given in [4], [5].

Our goal in this study was to test the data gathering system by evaluating the protocol performance. The underlying assumption is that tcpeval contains the set of metrics, topologies, traffic generation, and statistics gathering that serves our purpose well. The tcpeval in the specific ns-2.31 installation was patched and created a complex aggregation of simulation scripts that connects the ns-2, tcpeval and the TCP SACK PR Agent. Hence, we have built a data gathering system and provided a means for running the built-in tcpeval scenarios and topologies to generate and execute a comprehensive and systematic set of experiments with the modified protocol.

3.2 The data gathering system architecture

3.2.1 tcpeval structure

The architecture of tcpeval is presented in Figure 2.

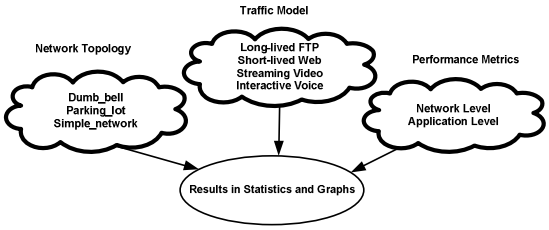

Fig. 2: tcpeval architecture

There are three network topologies included: Dumb-bell (Fig. 3), Parking_Lot (Fig. 4) and Simple Network (Fig. 5).

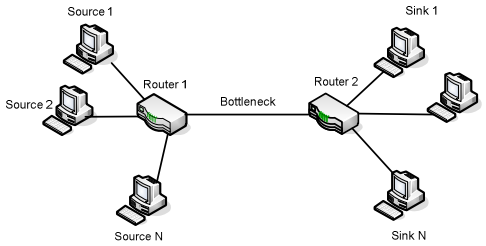

Fig. 3: Dumb-bell Topology

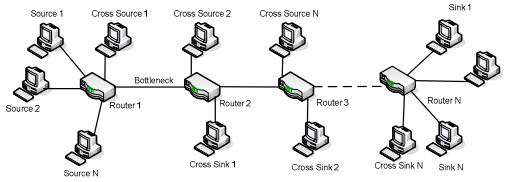

Fig. 4: Parking-lot Topology

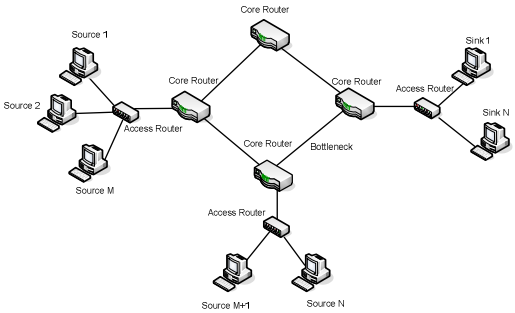

Fig. 5: Simple Network Topology

tcpeval provides generation of various kinds of traffic models. It gathers a large set of metrics: throughput, delay, jitter, and loss rate. Response times and oscillations to sudden network changes, fairness and convergence are measured as well. Testing robustness in challenging environments can be obtained by adding the PER parameter to a simulation scenario.

3.2.2 Architecture details

A set of simulations is conducted as a series of 12 experiments (Section 3.2.3) for variable PER values. When a simulation is started, a basic TCP scheme is selected. That will be the protocol for the long-lived ftp traffic included in the simulation. Besides this traffic, a short-lived HTTP traffic, streaming video traffic and interactive voice traffic can be included. tcpeval contains three example simulations for each of the network topologies:

- test_dumb_bell.tcl,
- test_parking_lot.tcl and
- test_network 1.tcl

Their parameters definitions are in def_dumb_bell.tcl, def_parking_lot.tcl and def_network_1.tcl, respectively. For each of these definition files, we have created a corresponding generic definition file where the parameters that need to be variable are unspecified. The passing of the parameters to the main script through and invoking of the whole system starts with a single command line.

The main script iterates through the set of PER values, parses the generic files and replaces the generic parameters with fixed ones, which produces four versions of def_dumb_bell.tcl, def_parking_lot.tcl and def_network_1.tcl that correspond to the twelve experiments shown in Table 1. During each iteration, the simulation with the parameters defined in one of these scripts is performed. The results of the tcpeval simulation scripts are first stored in */tmp/exp#* folder which is created for each performed simulation. The contents of these folders depend on the parameters defined in the definition files. In our experiments, we choose generation of both data and graphic results. In that case, the *exp#* folder contains two folders: *data* and *figures*. The folder *data* contains the files with the results of the simulations that we are interested to obtain. We can also have a view of the results summarized in an html file. After the simulation is finished, the results are collected from the correspondent folders, averaged, and the text file named *res.txt* is created. This file contains the results in numeric format of the entire set of simulations. The results in the file are grouped and conveniently formatted for parsing. Finally, an archive that contains all the simulation data is created. The whole process is presented in Figure 6.

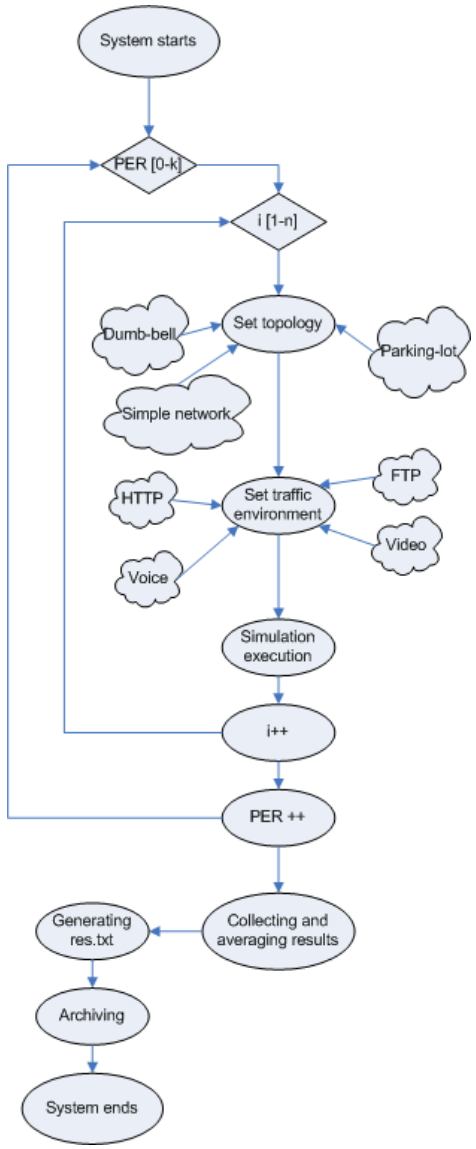

Fig. 6: Flow of operation of the data gathering system

3.2.3 Providing confidence level

The main goal was to investigate the performance of SACK PR compared to SACK for different values of PER, which was the variable parameter in all simulations (as proposed in Figure 1.). We have created a set of 12 different experiments in order to test various features of the protocol performance.

Their structure is given in Table 1. We consider both FIFO and RED [10] queuing discipline.

Tab. 1 Experiments setup

Exp. No.	Protocol	Topology	Queuing Discipline
1	SACK	Dumb-bell	DropTail
2	SACK PR	Dumb-bell	DropTail
3	SACK	Dumb-bell	RED
4	SACK PR	Dumb-bell	RED
5	SACK	Parking-lot	DropTail
6	SACK PR	Parking-lot	DropTail
7	SACK	Parking-lot	RED
8	SACK PR	Parking-lot	RED
9	SACK	Simple Network	DropTail
10	SACK PR	Simple Network	DropTail
11	SACK	Simple Network	RED
12	SACK PR	Simple Network	RED

For each simulation experiment, in addition to the PER interval, the user can choose the number of runs. in such a way that a confidence level of 95% for relative precision of 5% is accomplished. Various simulations assisted us in calculating that 10 runs for the Dumb-bell and 20 runs for the Parking-Lot topologies respectively are sufficient to attain the required statistical significance. In both cases, it took 100 seconds to converge to a steady state.

4 The system performance

4.1 Duration of the simulations

The duration of a simulation depends on the following factors:

- the number of iterations,
- the duration of each simulation,
- the capacity and the complexity of the topology,
- the complexity of the simulation scenario, and
- the complexity of the code describing the objects participating in the scenario.

Naturally, it depends significantly on the hardware platform. We observed the duration of a simulation of the Dumb-bell network topology with the following parameters: 20 iterations, PER varied from 0% to 10%, there were five long lived forward ftp flows, five long lived reverse ftp flows, HTTP generation rate of 15 Kbps, five video streaming flows and five interactive audio flows. The capacity of the network bottleneck was 10 Mbps. On a platform with an Intel Pentium 4 CPU 3.00 GHz with 512MB RAM, the experiment took 12 hours.

4.2 Complexity of the code

The complexity of the code greatly influences the duration of the simulations. When writing the code for a particular network entity we must have in mind how frequent it will be used and optimize it as much as we can. That piece of code, for example, may be executed on each acknowledgement receipt. If the code is inefficient it will slow down the overall network performance immensely.

In the case of SACK PR, the algorithms of the modification are very simple in terms of program complexity. The measurement off the RTTs is passive, based on timestamps. The other operations take just a few comparisons. So this modification does not add to the complexity of the TCP Agent.

5 The results

5.1 Results representation format

At the final stage, the data gathering system summarizes the following results:

- total and average forward throughput
- total and average reverse throughput
- cross traffic throughput, and
- fairness index

Based on the results, a comparison between the original (SACK) and the modified protocol (SACK PR) is calculated and displayed. For that purpose, we designed a spreadsheet document customized to our needs. The document has 12 sheets that contain calculations and charts that show the results of the performed simulations. The only manual action that is needed is importing the content of the *res.txt* file into the first sheet of document.

For each of the network topology there are three groups of sheets. Two kinds of charts are generated for each network topology:

- charts which depict the averaged nominal values of the performance metric as a function of PER (Figure 7),
- charts that show the improvement in the performance for a particular parameter between the original and the modified protocol (Figure 8).

Figure 7 presents the performance of SACK and SACK PR in a Dumb-bell network topology where PER varies from 1% to 10%.

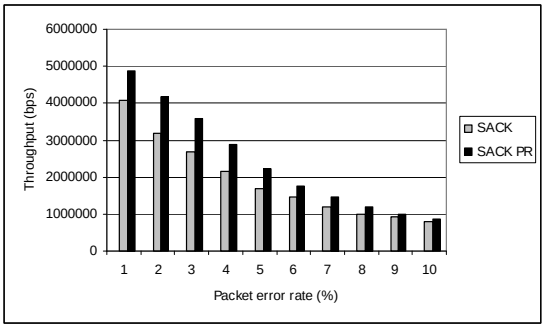

Fig. 7: SACK and SACK PR achieved throughput for Dumb-bell scenario with FIFO queuing discipline

Figure 8 presents the improvement of SACK PR throughput in a Dumb-bell network topology where PER varies from 1% to 10%.

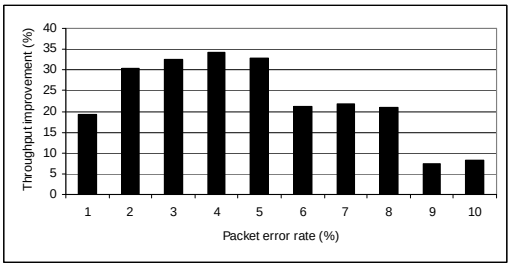

Fig. 8: SACK PR vs. SACK achieved throughput improvement for Dumb-bell scenario with FIFO queuing discipline

Tables and charts are generated for all the experiments represented in Table 1. The charts are represented in worksheets categorized by the simulated network topology. The summary report is structured in a systematic way and can be used to compare the performances of an arbitrary number of protocols (Figure 9).

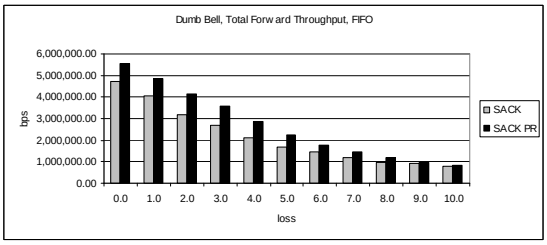

Fig. 9a: An example of a chart sheet

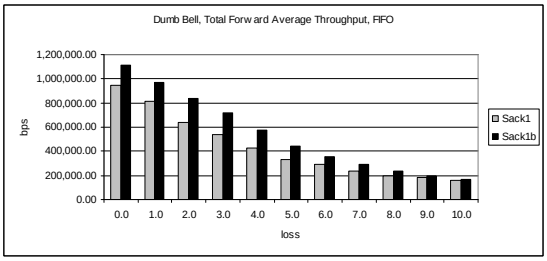

Fig. 9b: An example of a chart sheet

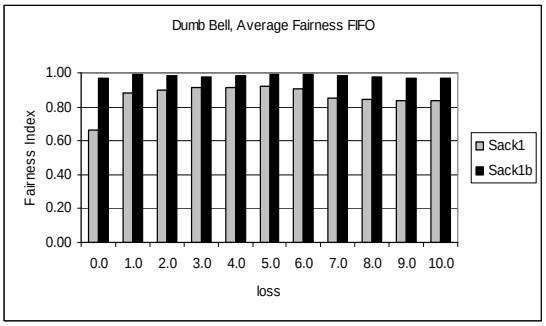

Fig. 9c: An example of a chart sheet

6 Conclusions and future work

In this paper we have presented a simulation environment for investigating e2e transport protocols, which is built on the bases of two popular simulation tools: ns-2 and tcpeval.

An automatic data gathering system that suits the needs of our particular simulation platform has been generated along with a battery of various simulation scenarios. The simulation experiments are run repeatedly until the outcome results are improved and bettered in such a way to confirm to the preset and satisfactory confidence levels.

The major merits and benefits for the researcher from the proposed iterative system for simulation stem from the facts that (1) there is a detailed report on the performance of the examined protocol, (2) the data i statistically verified and is accurate, and (3) the duration of the whole simulation is within a reasonable time (a couple of hours). Naturally, the variations in all three attributes of the system depend on the particular platform and the associated scenarios.

Plans for the future include implementation of a tool that automatically collects a larger and more diverse set of performance metrics.

Further improvements may address the automatic calculation of confidence intervals, which define dynamic bounds on the number of iteration need to attain confidence intervals with preselected relative precisions. That should provide an additional integrity to the results of the simulation and consequently to the validation of the protocols.

Basically, all of the work so far indicates a strong possibility to develop a sound and integrated simulation platform with data gathering and processing capabilities that should create affordances for easier, faster, and more reliable in a sense of statistical significance evaluation of protocols related to heterogeneous networks.

7 References

[1] Postel, J., "Transmission Control Protocol", *RFC 793*, Sep 1981.

[2] The Network Simulator - ns-2, http://www.isi.edu/nsnam/ns/

[3] An NS2 TCP Evaluation Tool, http://labs.nec.com.cn/tcpeval.htm

[4] Stojcevska, B., and Popov, O. B.,: *Loss Pairs for Congestion Detection and Control*. In: Proc. of the 8th World Multi-Conference on Systemics, Cybernetics and Informatics SCI2004, Volume XI, pp. 275-283, July 2004, Orlando, Florida, USA

[5] Stojcevska, B., and Popov, O. B.,: "Packet Pairing for TCP in heterogeneous networks", *ACTA ELECTROTECHNICA ET INFORMATICA*, No.4, Vol. 6: 1- 12, 2006

[6] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow. RFC 2018: TCP Selective Acknowledgment Options, 1996

[7] L. Andrew, C. Marcondes, S. Floyd, L. Dunn, R. Guillier, W. Gang, L. Eggert, S. Ha, and I. Rhee, "Towards a common tcp evaluation suite," in PFLDnet 2008, Manchester, UK.

[8] M. Hassan, and R. Jain, High Performance TCP/IP Networking: Concepts, Issues, and Solutions. Pearson Prentice Hall. 2004.

[9] Chiu, D. M., and Jain, R. "Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks", *Computer Networks and ISDN Systems*, Vol. 17, pp. 1-14, 1991

[10] Floyd, S., and Jacobson, V., "Random Early Detection Gateways for Congestion Avoidance". *IEEE/ACM Transactions on Networkin*g, 1(4):397–413, Aug 1993