Methodologies and Tools for Exploring Transport Protocols in the Context of High-speed Networks

Romaric Guillier and Pascale Vicat-Blanc Primet
ENS-Lyon, INRIA, University of Lyon, LIP Laboratory
Lyon, France

E-mail: romaric.guillier@ens-lyon.fr, pascale.primet@inria.fr

Abstract

This PhD work aims at proposing a methodology and tools for the evaluation of transport protocols in the context of large scale computing environment based on high speed networks. The goal is to define and validate a benchmark for comparing different transport and congestion control approaches. Our contribution concerns the definition of metrics, scenarios, experiments and pilot demonstrations for high demanding distributed applications. After stating the problem of evaluating transport protocols in high speed networks, we present existing methodologies for this purpose. We introduce our proposal of a tool to help users performing network experiments. We illustrate our approach with a few examples of results obtained from experiments in our Grid'5000 testbed which complements other existing approaches.

1. Introduction

Recent months have seen the announcement of the impending construction of "data centers" on the Internet at unprecedented scale. It is foreseen that these data centers will be built around clusters, or around a federation of geographically distributed data centers, connected by long-haul links. The issue of congestion control for high speed transport protocols over such long distance fat networks is still an open issue to be addressed in the large scale distributed systems (grids) as well as in the large scale data centers.

FTTH (Fibber To The Home) is foreseen as the next evolution of networks that will give customers access to a range of new applications like VOD, advanced telecommuting and large-scale computing, that will require massive data transfers in both directions at unpredicted times. Unlike the Internet, such networks have low aggregation level (end-hosts network capacities in the same order of magnitude as the access links capacities) and low multiplexing factor (few flows get mixed in the bottleneck link) [7]. Most of these transfers will be based on TCP protocol.

TCP provides a fully distributed congestion control protocol which statistically share available bandwidth fairly among flows. TCP was designed first and foremost to be robust. When congestion is detected, TCP solves the problem by drastically reducing the output rate, but at the expense of performance. For example, for a standard TCP connection with 1500-byte packets and a 100 ms round-trip time, achieving a steady-state throughput of 10 Gbps would require an average congestion window of 83,333 segments, and a packet drop rate of at most one congestion event every 5,000,000,000 packets (or equivalently, at most one congestion event every 1.66 hours) [9]. This means that in the context of FTTH and large-scale computing enhanced TCP variants will have to be used.

To solve the problem of TCP in very high speed optical networks, protocols enhancements and alternative congestion control mechanisms have been proposed. Most of them are now implemented in current operating systems. But congestion control is the most important and complex part of a transport protocol. All the proposed variants are neither equivalent nor suited for every environment or every application. Moreover they may not cohabit well. Many alternatives are proposed, but they are difficult to compare. The research community recognises that it is important to deploy measurement methods, so that the transport services and protocols can evolve guided by scientific principles.

This PhD work aims at contributing to this methodological effort by proposing a methodology and a tool for these studies, and presenting some steps towards a benchmark design for innovative high speed transport protocols comparison. Our contribution concerns the definition of metrics, scenarios, experiments and pilot demonstration for high demanding distributed applications. The goal is to define and validate a benchmark for comparing different transport and congestion control approaches.

The rest of the paper is organised as follows. In Sec-

tion 2, we survey different methodologies that have been recently used to evaluate transport protocols. Then in Section 3, our proposal, articulated around the Network eXperiment Engine (NXE) is presented. Some results obtained previously are presented in Section 4. We conclude in Section 5, giving an insight into our future works.

2. Evaluation methodologies for high-speed transport protocols

For the last couple of years, several teams have been aiming at developing methodologies and tools providing comprehensive standards-compliant testing of TCP implementations. Several methodologies, scenarios and results have been proposed in [2, 14, 16] to identify characteristics, how they affect experiments' results, and which aspect of evaluation scenario determine them.

In this section, we present initiatives focusing on TCP variants for high-speed networks evaluation. In the IRTF draft [3], Wang *et al.* propose a framework for benchmarking TCP variants based on the NS-2 network simulator. Mascolo in [17] is using NS-2 simulations to observe the impact of reverse traffic on the new TCP congestion control algorithms.

For experimenting and validating their TCP variants (BIC-TCP and CUBIC), Injong Rhee *et al.* use a testbed based on *Dummynet* [14]. With their experiments on TCP [16], Dough Leith *et al.* present too an experimental testbed based on *Dummynet* network emulator.

Few real experiments have been run [8,16] to analyse the behaviour of a range of new protocols. Other recent works focus on shared high speed networks dedicated to high performance distributed applications. In [11, 13], 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 is an experimental networking testbed aimed at developing, testing and evaluating new communications protocols and technologies like FAST or TCP MaxNet.

This brief overview (see Table 1) of recent works on high-speed transport protocol evaluation has highlighted that various instruments can be used for this purpose: simulation, emulated networks (with software or hardware emulators), real networks (Internet or dedicated private networks).

Simulation with NS-2 NS-2 is a reference discrete event network simulator that has been used since the early nineties to analyse and evaluate a range of network protocols, from wireless MAC layers to TCP congestion control methods. Simulators like NS-2 are using mathematical formulae to compute the interactions between different entities involved in the experimental setting (*e.g.* end-hosts, routers, packets).

Emulation Emulation, that is to say the duplication of the properties of a network (*e.g.* latency, capacity) using a limited amount of resources, can be used to reproduce the behaviour of large-scale networks. It can be done either with software or with specific hardware.

The AIST and INRIA teams use hardware emulators combined with network virtualisation software eWAN, to evaluate protocols under different latency and topology conditions [18]. AIST-GtrcNET-10 is a hardware emulator that allows latency emulation up to 858 ms without losses, rate limitation and precise bandwidth measurements at 10 Gbps wire speed. GtrcNET-10p3 is a fully programmable network testbed, which is a 10 Gbps successor of a well-established network testbed, GtrcNET-1.

Real testbeds The real experiment method gives an insight of the real protocol behaviour in very high speed environments (*e.g.* 10 Gbps), explores the interactions with the hardware infrastructure and generally helps debugging the global hardware and software communication chain.

Wan-In-Lab [10] is a testbed of the California Institute of Technology. It is built around a 2400 km optic fibber cable and arrays of optical switches to construct networks with variable length and RTT. Users can upload experimental kernels instrumented with the Web100 tools, and run a set of predefined tests.

Grid'5000 [6] 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 10 Gbps 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 [11, 13].

Simulators like NS-2 can be used to study the internal mechanisms of a transport protocol at packet level, to have a fine-grained control on every point of the network and a fine-grained precision on the metrics we want to compute. But there are some downfalls too. This is very CPU intensive making large experiments last for a long time, and therefore very high-speed experiments are difficult (as in these examples) to set up. The deterministic nature of this kind of tools can also lead to synchronisation effects, that may not appear in real life. Software emulators present similar flaws, as it is difficult to reach high link speeds (at most 400 Mbps) due to software overheads. Real testbeds give researchers access to 10 Gbps links through real networking equipment to understand the behaviour of protocols in the real world, but there are limitations here too. Due to deployment cost, it is often difficult to create complex topologies and some parameters are stuck in a limited range, like the RTT. Combined with hardware emulators, they can gain an extra flexibility without altering the other properties of the

Author	Wang [3]	Mascolo [17]	Rhee [14]	Leith [16]	Grid'5000 [11, 13]
Type of network	NS-2	NS-2	Dummynet	Dummynet	Real
Goal of study	Generic framework	Reverse traffic impact	TCP variant comparison	TCP variant comparison	TCP variant comparison
Topology	Dumbbell, Parking Lot, 4 Domain Network	Dumbbell	Dumbbell	Dumbbell	Dumbbell
Traffic model	FTP, Web, Video streaming, Voice	FTP, Web	FTP, Web	FTP, Web	FTP, Web
Metrics	throughput, queueing delay, jitter, loss rate, response time, fairness, convergence, robustness	link utilisation, goodput, congestion window size, timeout events	link utilisation, coefficient of variance, loss rate, Jain index, convergence	link utilisation, goodput, congestion window size, loss rate, Jain index	throughput, goodput, std dev, Jain Index completion time
Rate max (Mbps)	n/a	250	400	250	10000
RTT range (ms)	n/a	40,80,160	16,64,162,324	16,22,42,82,162	0-200

Table 1: Recap table of different evaluation methodologies

Parameter	**NS-2**	**Hw. emul.**	**Sw. emul.**	**Real testbeds**
RTT	Any	< 858 ms	< 400 ms	limited
C	Any	10 Gbps	< 400 Mbps	10 Gbps

Table 2: Parameter limitation for the different methods

Parameter	**Description**	**Range**
RTT	Round Trip Time	0 to 200 ms
C	Bottleneck capacity	1 or 10 Gbps
K	Aggregation lvl	1 or 10
M	Multiplexing factor	1 to 20
C_g	Congestion lvl	0 to 2.0
N_s	Parallel streams	0 to 10
R	Reverse traffic lvl	0 to 2.0

Table 3: Parametric space used in real experiments

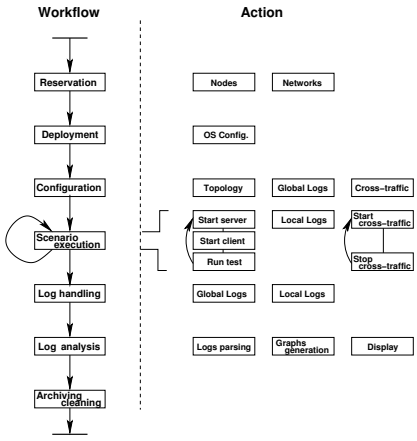


Figure 1: Workflow used by the NXE tool

real testbeds.

As each protocol evaluation tool has its advantages and pitfalls, a mix of several methods is highly required to produce convincing results [4]. Table 2 presents the latency and bottleneck capacity range usable in each instrument.

3. Proposed automation tool

Real experiments give the flexibility to explore a wider range of rates, and the real interaction of the protocol with end-points and network equipment. But real experiments are difficult to deploy and analyse. There are problems related to the hardware (faulty components in the servers) or to the software that is managing the testbed (lack of usable output from the deployment tool) or just improper handle of the configuration of networking stack in the kernel.

But the biggest issue to tackle is linked to the dimension of the parametric space (see Table 3). In this Table, K represents the aggregation level, the ratio between the bottleneck capacity and the access link capacity. M is the multiplexing factor, the number of contributing sources. The background traffic is another factor, which has to be expressed by flows' arrival and size distributions. Even by carefully limiting the range of selected parameters to "interesting" values, you can end up very quickly with hundred of hours worth of planned experiments to run and analyse. It is then vital to have tools to automate these tasks.

The following sections present the tool we have developed to help with the automation of networking experiments in real network testbeds.

3.1. Network experiment definition

A networking experiment is described with a scenario skeleton defined as a succession of dates at which an event occurs. The events corresponds to the starting point of an action (*e.g.* the start of a new bulk data transfer, of a new web session) combined with the parameters relevant to this action (*e.g.* distribution law of file sizes, inter waits) between a set of end-hosts, whose size depends on the kind of application we are trying to model (*e.g.* 2 for data transfers, many for parallel applications).

The end-hosts are organised in a networking abstract topology, that roughly defines sites (*e.g.* aggregation of end-hosts, as in a cluster), aggregation points between them

(*e.g.* switches or routers) and networking links. The "abstract" term refers to the fact that an instantiation of the nodes over this topology is needed at run-time as in real testbeds, and resources allocation mechanisms might be used to accommodate multiple users.

Figure 1 shows the experiment workflow, and the various operations that are done at each stage of the execution of a protocol evaluation scenario. This workflow is a description of an evaluation process. It is composed of a number of tasks which are connected in the form of a directed graph. These tasks have been broken up into elementary operations to explicit what is done precisely at each stage. The tasks were designed so that there is as little interaction as possible between successive tasks. The description of each stage of the workflow is as follows:

Reservation: at this stage, the available resource allocator services are contacted to get the resources needed by the experiment, *e.g.* computing nodes or network links.

Deployment: this configuration phase can be either a reboot of the nodes unto an adequate kernel image, or just the setting of the OS internal variables (*e.g.* TCP buffer size) to the appropriate value.

Configuration: at this stage, the available hardware (*e.g.* hardware latency emulator, routers) are contacted to alter the topology, or to activate the gathering of statistical information (*e.g.* aggregate throughput) according to the needs of the experiment.

Scenario execution: here the actual execution of the scenario is started. The scenario can be run multiple times in a row to ensure that the results are consistent.

Log handling: the logs generated by the nodes and the global logging facility are gathered at a single point for analysis.

Log analysis: the logs are parsed, and metrics are computed from them to generate graphs that can be easily interpreted by the user.

Archiving and cleaning: resources are reset and released.

3.2. Implementation

The actual implementation is made in Python and uses the python Expat XML library, the paramiko SSH2 library and the Gnuplot python bindings. The additional programs *iperf* and *D-ITG* are used to simulate the workloads. Bash scripts are used to wrap the calls to these programs on the end-hosts that are used in the scenarios.

The scenarios are described through XML files that provides a simple and hierarchical description of the topology,

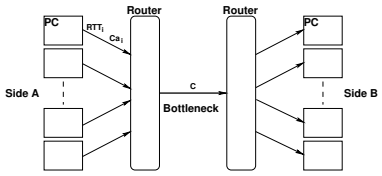


Figure 2: Typical experiment topology: a simple dumbbell

the general configuration and the interactions between the end-hosts.

NXE is an application that scripts the execution of a schedule based on a centralised relative timer, that is generated from the input scenario description. It assumes that the granularity of the scenario execution steps is coarse and in the same order of magnitude as a second. The timers used are much more finely grained (in the order of a few milliseconds), so the tool could be enhanced to be more precise, but currently it doesn't seem relevant to the general user-context. For scalability purposes, it launches a separate thread for every node involved in the scenario, and issues the commands at the appropriate time via an SSH remote command execution. Only one SSH connection is opened per node.

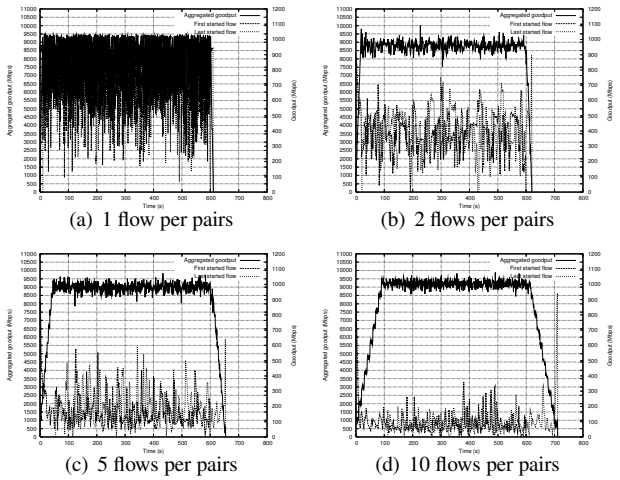
4. Results

In this section, we discuss some results obtained with our methodology and tool. Figure 2 presents the typical topology that has been used to perform experiments in the Grid'5000 testbed. The dumbbell is the typical topology used to evaluate congestion control mechanisms, as it provides a single bottleneck where flows gather, generating congestion. Large-scale computing over FTTH is characterised by massive, symmetrical and sporadic machine to machine data transfers. This requires specific studies in terms of high speed transfer protocol evaluation.

4.1. Parallel streams

To perform massive data transfers, a lot of applications are using GridFTP [1]. GridFTP is based on the parallel streams approach which has been recognised as a powerful technique to increase the global throughput [15]. We have designed an experiment to systemically check the impact of the number of simultaneous parallel streams on the performance of such transfers.

In this experiment, 11 pairs of nodes and a constant number of parallel streams per nodes were used. Up to 110 flows were generated. In this experiment, the nodes and their streams were sequentially started 1 s apart, and were transmitting continuously for 600 s.



(a) 1 flow per pairs (b) 2 flows per pairs

(c) 5 flows per pairs (d) 10 flows per pairs

Figure 3: "Parallel streams" using BIC-TCP in Grid5000, 11 ms RTT

Nb of flows by node	1	2	5	10
Mean total goodput (Mbps)	8353.66	8793.92	8987.49	9207.78
Flow mean (Mbps)	761.70	399.83	163.53	83.71
Jain Index	0.9993	0.9979	0.9960	0.9973

Table 4: Results for "parallel streams in Grid5000" scenario for 11 pairs of nodes

Figure 3 shows the impact of parallel BIC-TCP streams on the utilisation of a 10 Gbps link in Grid'5000. Each subfigure presents the aggregate goodput and two individual flows (the first and the last started) on the same plot. These figures show that individual goodput become more stable when the number of flows increases. This confirms that TCP behaves better in high multiplexing conditions.

The aggregate results for this experiment are also summarised in Table 4. As expected, large number of parallel streams manage to obtain more bandwidth than single streams. This confirms the convergence to an asymptotic value of throughput deficiency as in [5]. Here the asymptotic deficiency is about 700 Mbps (i.e. 7 %).

4.2. Reverse traffic impact

This experiment tries to highlight the impact of the congestion of the reverse path on the forward path. This is important in the context of grids as it is possible that both forward and reverse paths are heavily congested. For this experiment, we used (n_f, n_r) pairs of nodes generating the forward and the reserve traffic consisting in 30 Gb transfers, one transfer per node pair. The transfers were started sequentially 1 s apart.

Figure 4 presents the effect of different levels of reverse

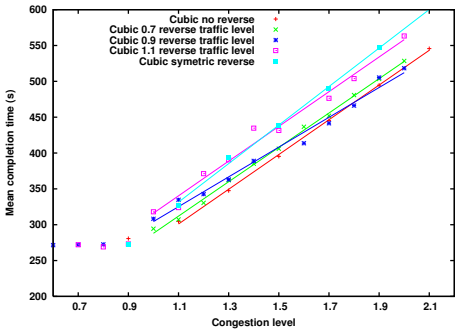


Figure 4: Impact of reverse traffic level on mean completion time for CUBIC, 19.8 ms RTT

traffic on the mean completion time for CUBIC. We can see that for reverse traffic level lower than 1.0, its effect is limited on the mean completion time (about 2.5 %). The fluctuations observed for 0.9 reverse traffic level are mainly due to the fact that we are close to the congestion gap and thus to a very unstable point. When the reverse traffic is congesting, we observe that the difference compared to the case without reverse traffic is much more important (about 10 %). The slopes for 0.7, 1.1 and no reverse traffic level are very similar to each other, which indicate that the impact of the reverse traffic could be seen as a reduction of the available bandwidth.

4.3. Towards a transport protocol benchmark

Using our methodology and NXE tool, we are currently working on the definition of a test suite called HSTTS (High Speed Transport protocol Test Suite) [12]. The goal is to provide users with synthetic metrics to assess the performance of their networking environment with respect to some common application traffic profiles relevant to the context of grid computing. In this benchmark, we aim at exploring the parametric space presented in Table 3 systematically. The metrics used are different from those of Table 1 as they are focusing on a user perspective. Figure 5 gives an example of the benchmark output. It presents the comparison of the performance of some TCP variants for a bulk data transfer done in a dumbbell topology with 19.8 ms RTT. We can observe that in these conditions, BIC-TCP seems to be the best solution as it presents both the smallest average completion time and standard deviation.

5. Conclusion

In this article, we have shown the problem of evaluating transport protocols in high-speed networks. We have

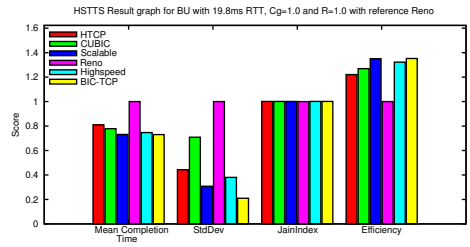


Figure 5: Comparison of TCP variants for bulk data transfer use, with congested forward and reverse path

presented our tool NXE, that allows the automation of experiments in very large scale real testbeds. This type of experiments are mandatory to assess the characteristics and behaviour of transport protocols in a real environment. We have also presented some results obtained with our approach in the Grid'5000 facility, thus verifying experimentally results that had not be tested yet at this scale in real high-speed networks. We are currently working on a tool to easily convert NXE scenarios from and into the language readable by *NS-2* so as to have access to the better of both worlds, simulation and real experiments. We will also design and develop a tool to automatically characterise a networking environment (bottleneck, buffering capacities, load. . .), and to help the user choosing the most appropriate transport protocol for a given need. As part of future works, we will first validate our approach in dedicated grid environments. We will test realistic scenarios in different platforms. We will then extend our work to FTTH networks by replacing the traffic models used in our scenarios by models more relevant to the Future Internet.

Romaric Guillier is a PhD student advised by Pascale Vicat-Blanc Primet. He started his PhD in September, 2006 and is expected to complete in 2009.

6 Acknowledgement

This work has been funded by the French ministry of Education and Research, INRIA, and CNRS, via ACI GRID's Grid'5000 project , ANR HIPCAL grant, the ANR IGTMD grant, IST EC-GIN project, INRIA GridNet-FJ grant, NEGST CNRS-JSP project.

References

[1] GridFTP: Protocol extension to FTP for the Grid. In Allcock W., editor, *Grid Forum Document 20*, April 2003.

[2] Tools for the evaluation of simulation and testbed scenarios. In S. Floyd and E. Kohler, editors, *http://www.icir.org/tmrg/draft-irtf-tmrg-tools-03.txt*, December 2006.

[3] An NS2 TCP Evaluation Tool Suite. In G. Wang, Y. Xia, and D. Harrison, editors, *http://www.icir.org/tmrg/draft-irtf-tmrg-ns2-tcp-tool-00.txt*, April 2007.

[4] A. F. M. Allman. On the effective evaluation of TCP. *ACM Computer Communication Review*, 5(29), 1999.

[5] E. Altman, D. Barman, B. Tuffin, and M. Vojnovic. Parallel TCP Sockets: Simple Model, Throughput and Validation. In *Proceedings of the IEEE INFOCOM*, 2006.

[6] R. Bolze, F. Cappello, E. Caron, M. Daydé , F. Desprez, E. Jeannot, Y. Jégou, S. Lanteri, J. Leduc, N. Melab, G. Mornet, R. Namyst, P. Vicat-Blanc Primet, B. Quetier, O. Richard, E.-G. Talbi, and T. Irena. Grid'5000: a large scale and highly reconfigurable experimental grid testbed. *International Journal of High Performance Computing Applications*, 20(4):481–494, November 2006.

[7] T. Bonald and J. Roberts. Congestion at flow level and the impact of user behaviour. *Elsevier Computer Networks (COMNET) Journal*, 42:521–536, 2003.

[8] R. L. Cottrell, S. Ansari, P. Khandpur, R. Gupta, R. Hughes-Jones, M. Chen, L. McIntosh, and F. Leers. Characterization and Evaluation of TCP and UDP-based Transport on Real Networks. In *PFLDnet'05*, February 2005.

[9] S. Floyd. RFC 3649: HighSpeed TCP for Large Congestion Windows. RFC 3649, December 2003. experimental.

[10] A. T. George S. Lee, Lachlan L. H. Andrew and S. H. Low. Wan-in-lab: Motivation, deployment and experiments. In *PFLDnet'07*, February 2007.

[11] R. Guillier, L. Hablot, Y. Kodama, T. Kudoh, F. Okazaki, R. Takano, P. V.-B. Primet, and S. Soudan. A study of large flow interactions in high-speed shared networks with grid5000 and gtrcnet-1. In *PFLDnet'07*, February 2007.

[12] R. Guillier, L. Hablot, and P. Vicat-Blanc Primet. Towards a user-oriented benchmark for transport protocols comparison in very high speed networks. Research Report 6244, INRIA, 07 2007. Also available as LIP Research Report RR2007-35.

[13] 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.

[14] S. Ha, L. Le, I. Rhee, and L. Xu. A Step toward Realistic Performance Evaluation of High-Speed TCP Variants. *Elsevier Computer Networks (COMNET) Journal, Special issue on "Hot topics in transport protocols for very fast and very long distance networks"*, 2006.

[15] T. Hacker, B. Noble, and B. Athey. Improving throughput and maintaining fairness using parallel TCP. In *Proceedings of the IEEE INFOCOM*, 2004.

[16] Y.-T. Li, D. Leith, and R. N. Shorten. Experimental Evaluation of TCP Protocols for High-Speed Networks. In *Transactions on Networking*, June 2006.

[17] S. Mascolo and F. Vacirca. The effect of reverse traffic on the performance of new TCP congestion control algorithm. In *PFLDnet'06*, February 2006.

[18] P. Vicat-Blanc Primet, R. Takano, Y. Kodama, T. Kudoh, O. Glück, and C. Otal. Large Scale Gigabit Emulated Testbed for Grid Transport Evaluation. In *Proceedings of The Fourth International Workshop on Protocols for Fast Long-Distance Networks, PFLDnet'2006*, February 2006.