A study of large flow interactions in high-speed shared networks with Grid5000 and GtrcNET-10 instruments

Romaric Guillier*, Ludovic Hablot*, Yuetsu Kodama†, Tomohiro Kudoh†, Fumihiro Okazaki†,
Pascale Vicat-blanc Primet*, Sébastien Soudan* and Ryousei Takano†
*LIP, UMR INRIA-CNRS-ENS Lyon-UCB Lyon 5668
École Normale Supérieure de Lyon, France
Email: rguillie@ens-lyon.fr
†National Institute of Advanced Industrial Science and Technology (AIST), Japan

***Abstract*— We consider the problem of huge data transfers and bandwidth sharing in the context of grid infrastructures where transfer delay bounds are required. This article investigates large flow interactions in a real very high-speed network and aim at contributing to high-speed TCP variants evaluation by providing precise measurements. It also gives an insight on the behaviour of protocols under different realistic congestion and long latency conditions in 10 Gbps experimental emulated environments.**

Keywords: bulk data transfers, bandwidth sharing, high speed transport protocol experimentation.

I. Introduction

The data volumes of distributed applications such as data and computing grids, distance visualisation and high-end collaborative environments are in the order of terabytes and will likely reach petabytes in some cases. The transfer of these data have demanding performance requirements such as reliable and predictable delivery [1]generating specific challenges on the transport protocol and its related mechanisms. The enhancement of TCP/IP has been intensively pursued to tackle limits that classical congestion control solutions encounter in long bandwidth-delay product environment [2]. Few studies have measured the performance of TCP proposals in real high speed networks [3], [4] and in grid environments. It is acknowledged that more real and systematic experiments are needed to have a better insight on the relevance of metrics, on representative scenarii for protocol evaluation and on the potential usage of these protocols in particular applications [5].

This paper contributes to this challenge by exploring several high data transfer scenarii in two experimental environments: the Grid5000 testbed and the AIST GtrcNET-10-based testbed. Grid5000 [6] is an experimental grid platform gathering more than 3000 processors over nine geographically distributed sites in France (see figure 1). The particularity of this testbed is to provide researchers with a fully reconfigurability feature to dynamically deploy any OS image or TCP stack on any host of the testbed and with a fully dedicated optical network. The other experimental environment we used is the GtrcNET-10-based emulated and controlled testbed connected within the AIST Super Cluster. Hosts of both testbeds have similar hardware configuration.

This article explores how the transport protocol enhancements could benefit to high-end applications in terms of data transfer efficiency and predictability in these two environments. It focuses on elephant-like bulk data transfers in very high-capacity networks that these grids are supposed to benefit from today. In such cluster interconnection context, hundreds of hosts may generate large flows through their gigabit interfaces. The access links between cluster network and wide area networks currently offer between 1 to 10 Gbps. Consequently they introduce a strong bottleneck that may drastically increase the transfer delays and impact the overall distributed environment performance. In the Internet, the endpoints' access rates are generally much smaller (2 Mbps for DSL lines) than the backbone link's capacity (2.5 Gbps for an OC48 link). According to the law of large number, coexistence of many active flows smooths the variation of load, and a link is not a bottleneck unless the load approaches its full capacity [7]. To curb the load, distributed congestion control protocols such as TCP statistically shares available bandwidth among flows in a "fair" way. In contrast, for high-end grid applications, the bandwidth demand of a single endpoint (1 Gbps, say) is comparable to the capacity of bottleneck link. Considering this context, this paper aims at exploring how elephant flows generated by high-speed TCP-variant may cohabit. The rest of the article is organised as follows. Section II gives some insights on parameter space and metrics. Scenario and experiments are described in section III. Results are discussed in section IV. The article concludes in section V.

II. Methodology

This work has been inspired by the results and methodologies proposed by [5], [8], [9]. [5] identifies several characteristics and describes which aspects of evaluation scenario determine these characteristics and how they can affect the results of the experiments. This helped us in defining workloads and metrics.

A. Traffic characteristics

The aggregated traffic on a link is characterised by the a) distribution of per-packet round-trip time, b) file sizes, c) packet sizes, d) ratio between forward-path and reverse-path traffic, e) distribution of peak flow rates, f) distribution of transport protocols. Despite no extensive study of grid traffic exists, we assume the specific grid context studied here presents the following specificities:

a) The distribution of per-packet round-trip time is multi-modal. Nodes are generally clustered, consequently, several modes may appear ($\frac{N*(N-1)}{2}$ modes for N sites), each mode of the distribution representing the set of given site to site connections.
b) File sizes are not exponentially distributed. For example, in Data Grid like LCG (for LHC) file size and data distribution is defined by the sampling rate of data acquisition. The traffic profile is then highly uniform.
c) Packet sizes are also mostly constant, with a large proportion of packets having the maximum size.
d) The ratio between forward-path and reverse-path traffic is unknown and depends on the location of the storage elements within the global grid.
e) Distribution of peak flow rates may also be uniform.
f) Today, most of grid applications need reliable transport and use TCP-based protocols. The distribution of transport protocols is modal.

In the rest of the paper, we call these specific conditions the "grid context".

B. Scenarii

We examine two types of features that can help users to obtain good performance in such context: parallel streams and TCP variants.

We investigate the four following types of scenarii:

- Parallel streams in the Grid5000 real testbed.
- Range of TCP variants in the Grid5000 real testbed.
- Range of TCP variants with a range of emulated latency in the AIST-GtrcNET-10 testbed.
- Range of TCP variants combined with parallel streams in the AIST-GtrcNET-10 testbed.

The parallel streams approach has been recognised as a powerful approach to increase the global throughput [10]. It is largely used by the Grid community through the GridFTP protocol [11]. [12] demonstrates why few connections are sufficient to make up for the aggregate throughput deficiency due to AIMD adaptation and establishes a formula. In this set of experiments, we explore the number of flows parameter.

Different TCP variants have been proposed to improve the response function of AIMD congestion control algorithm in high bandwidth delay product networks. All these protocols are not equivalent and not suited to every context. We investigate here their behaviours in our "10 Gbps grid context". Combining parallel streams approach and enhanced TCP variants could pull the best from each approach in this context. We run several experiments to have a better insight on this idea. The last set of experiments investigates the behaviour of different TCP variants in a well controlled 10 Gbps emulated environment in which latency parameter is tuned from 0 ms up to 200 ms.

C. Measured parameters and metrics

We designed and configured our experimental testbeds to have a direct access to the following parameter measurements during experiments: a) goodput using *iperf* on the receiver side, b) aggregated throughput via the GtrcNET-10 equipment and c) TCP kernel variables with the *Web100* patch. The parameters are evolving along the three following axis: 1) TCP variant, 2) RTT, 3) congestion level.

Every test for a given RTT, has been repeated for a given congestion control method and for a given number of nodes. We took great care of fine measurement precision: 0.5 s for *iperf*, 20 ms for the GtrcNET-10 and *Web100*. Even though we specify *iperf* to perform *read()/write()* of 8 kB, we still observe burstiness in goodputs due to delay variation betweeen packets arrivals and *read()* returns.

To analyse all the data acquired, several metrics inspired by [13] have been used to synthetically characterise the behaviour of different TCP variants. These metrics are:

- mean goodput: $\overline{g_i} = \frac{1}{T}\sum_{t=0}^{T} g_i(t)$
- aggregate goodput: $G(t) = \sum_{i=1}^{N} g_i(t)$
- standard deviation of goodput:

$$\sigma = \sqrt{\frac{1}{T}\sum_{t=0}^{T}(g_i(t) - \overline{g_i})^2}$$

- goodput distribution:

$$\{p_{i,k} = p(\frac{k}{100} * c \leq g_i(t) < \frac{k+1}{100} * c) | k \in [\![0;100]\!]\}$$

- fairness [14]: $J = \frac{(\sum_{i=1}^{N}\overline{g_i})^2}{N(\sum_{i=1}^{N}\overline{g_i}^2)}$
- aggregate throughput: $X(t) = \sum_{i=1}^{N} x_i(t)$

where N is the number of nodes involved in the experiment, c the capacity of the access link, T the total duration of the experiment (typically 2800 s), $g_i(t)$ the i^{th} node's goodput over time t averaged on the iperf sampling interval, and $x_i(t)$ the i^{th} node's throughput over time t averaged on the GtrcNET-10 sampling interval.

III. EXPERIMENT DESCRIPTION

A. System and service description

We used two similar experimental systems, composed of a classical dumbbell topology with twelve 1 Gbps source workstations connected to a 10 Gbps bottleneck link and twelve sink workstations on the other side as described in figure 2. In the first testbed (*testbed 1*):(Grid5000, France), the backbone of the Grid5000 platform is composed of a private 10 Gbps Ethernet over DWDM dumbbell with a bottleneck at 10 Gbps between Rennes and Nancy hubs (see figure 1). The average RTT is 11.5 ms that gives a bandwidth-delay product of 13.71 Mbytes.

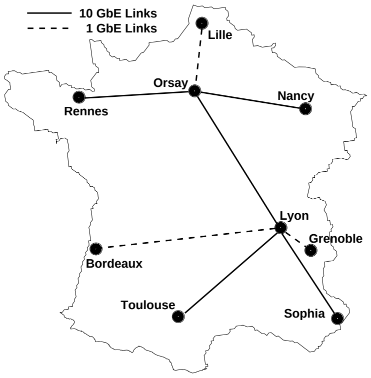


Fig. 1. Grid5000 backbone

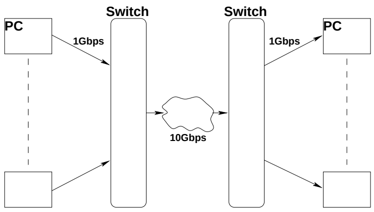
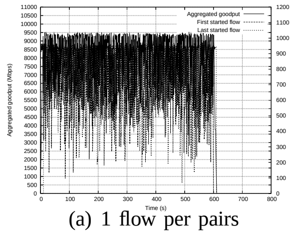


Fig. 2. Topology of the experiments, the cloud represents either the Grid5000 or the AIST-GtrcNET-10 backbone

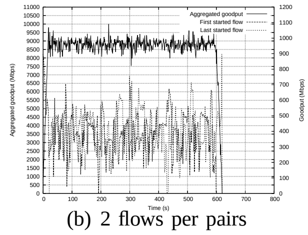
Nb of flows by node	1	2	5	10
Mean total goodput (Mbps)	8353.66	8793.92	8987.49	9207.78
Flow mean (Mbps)	761.70	399.83	163.53	83.71
Jain Index	0.9993	0.9979	0.9960	0.9973

TABLE I

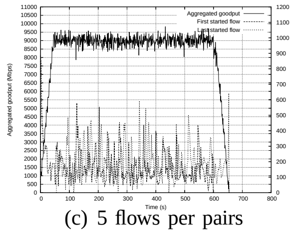
RESULTS FOR "PARALLEL STREAMS IN GRID5000" SCENARIO FOR 11 PAIRS OF NODES



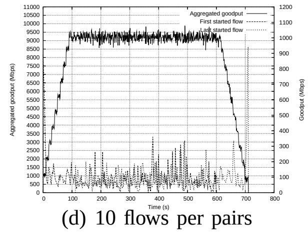
(a) 1 flow per pairs



(b) 2 flows per pairs



(c) 5 flows per pairs



(d) 10 flows per pairs

Fig. 3. "Parallel streams" using Bic in Grid5000, 11

The second testbed [15] (*testbed 2*):(AIST-GtrcNET-10, Japan), is fully controlled. It is built around the GtrcNET-10 equipment 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 for 1 Gbps Ethernet. GtrcNET-10p3 consists of a large-scale Field Programmable Gate Array (FPGA), three 10 Gbps Ethernet XENPAK ports, and three blocks of 1 GB DDR-SDRAM. By re-programing FPGA configuration, its functions are easily added and modified with keeping 10 Gbps wire speed.

Nodes are interconnected by a layer 2 switch. All PCs also have a second Ethernet NIC to which all control traffic is sent so that there is no perturbation on the test traffic. The output port of the switch acts as the bottleneck of the system.

In the *testbed 1*, we used Dell PowerEdge 1950 and HP ProLiant DL145G2 servers, while IBM e-server 325 were used in the *testbed 2*. In both testbed, the nodes were all equipped with 2 AMD64 Opteron on which we deployed GNU/Linux 2.6.17 kernels patched with the *Web100* [16].

IV. RESULTS ANALYSIS

A. Parallel streams in Grid5000 testbed

Figure 3 shows the impact of parallel BIC streams on the utilisation of a 10 Gbps link. We are using 11 pairs of nodes and a constant number of parallel streams per nodes. The nodes and their streams are sequentially started. It accounts for the jitter at the beginning and the end of each figure, as the first (or the last) flow will be able to use most of the 1 Gbps link of its node. 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 I. 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 [12]. Here the asymptotic deficiency is about 700 Mbps (i.e. 7 %).

Figure 4 is a comparison between our measures and Altman's model. Assuming that we can apply the formula to BIC:

$$\overline{x}(N) = C(1 - \frac{1}{1 + \frac{1+\beta}{1-\beta}N})$$

where β is multiplicative decrease factor of an AIMD protocol and C the bottleneck capacity, we can see that the two graphs are very similar and the convergence to an asymptotic value.

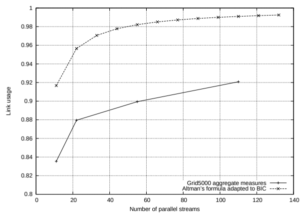
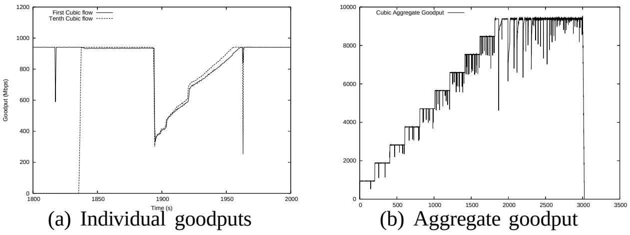
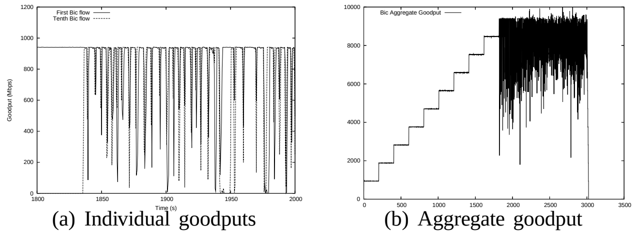


Fig. 4. Comparison of our measures against Altman's model



(a) Individual goodputs (b) Aggregate goodput

Fig. 5. Cubic in Grid5000, 11.5 ms RTT



(a) Individual goodputs (b) Aggregate goodput

Fig. 6. Bic in Grid5000, 11.5 ms RTT

Our graph is below Altman's by 8 %. This is likely the difference between throughput (as considered in the formula) and goodput (considered in our measurements)

As each pair of nodes is using the same number of parallel streams, no fairness problem between connection appears.

B. TCP variants experiment in Grid5000 testbed

In this test, we evaluate different TCP variant protocols (HS-TCP, H-TCP [17], Scalable TCP, BIC TCP [18], Cubic[1]) with the same latency - 11.5 ms - imposed by the Grid5000 testbed configuration. For a given TCP variant and a given RTT, the first tests series were performed as follows:

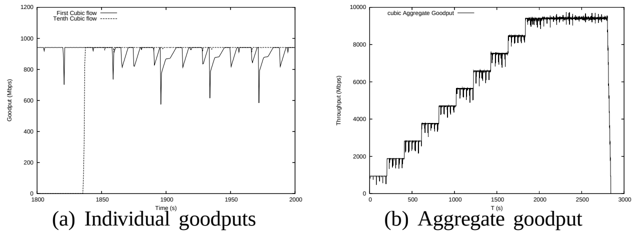
- At time 0, we start the first couple of client-server
- A *iperf* client is started 4 seconds after the corresponding *iperf* server to prevent overlap due to ssh connexion delay.
- Every 200 s, we start a new couple till all twelve nodes are started.
- As each *iperf* client is set to last $max\_duration - nb\_nodes\_started * 200$ s, they gradually stop around time $max\_duration$.

The interval between each flow's start is of importance as flows may interact during their slow start phase. In the previous case, we insure all interactions do not occur during any slow start phase. We give the results obtained for BIC, Cubic and H-TCP. For each protocol, the figure on the left shows the individual flow goodputs for the first and the tenth flow when we start to have congestion in the system. The corresponding figure on the right gives the aggregate goodput value. We do not report here the results for other protocols. Reader can find more details in [19].

At this latency, all the protocols manage to fully use the network. The maximal aggregate throughput is close to 9843 Mbps when all nodes are present. All figures are displaying sharp steps (as far as the 0.5 s mean provided by *iperf* allow us to see), except for Cubic (figure 5) and the H-TCP (figure 8) protocols. These two are starting to display heavy perturbations from the arrival of the fifth node (the system is not congested at this stage).

At the arrival of the tenth node, a change in the behaviour of all the protocols is observed despite the nominal capacity of our network is not reached yet. In this phase, the nodes aren't

[1]Please note that there is a known bug in the CUBIC implementation of the GNU/Linux kernel version we used for our experiments that accounts for the bad behaviour of this TCP variant in high latency conditions.



(a) Individual goodputs (b) Aggregate goodput

Fig. 7. Cubic in AIST GtrcNET-10, 11.5 ms RTT

able to maintain a "stable" goodput, which might be caused by the uncontrolled "background noise" of the Grid5000 testbed generated by other users.

We can also notice that some protocols have huge and quick individual variations in goodput such as Bic (figure 6). This has an impact on the mean aggregate goodput.

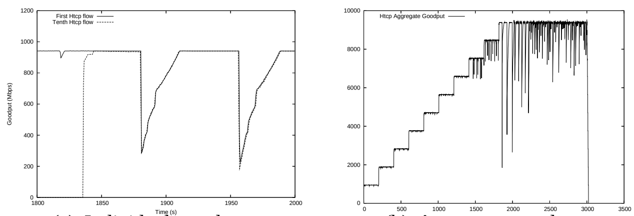
C. Exploration of TCP variants behaviour in various latency conditions

We experiment the various TCP protocols by applying the same experimental procedure we used for Grid5000 in the AIST-GtrcNET-10 testbed. In this emulated testbed we explore the impact of the latency parameter.

a) Impact of the latency: First, we are going to verify the expected impact of an increasing latency on the various TCP variants, which is a deterioration of the performances.

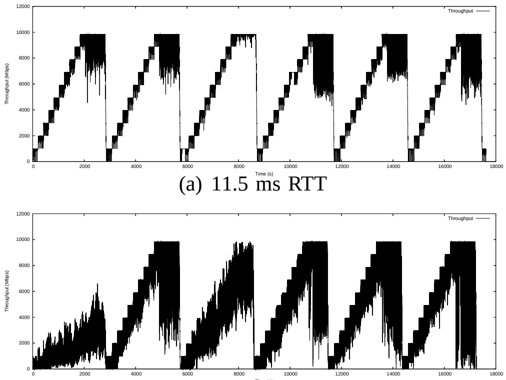
The figures were generated with the GtrcNET-10 logs for 11 ms and 100 ms RTT and so what is displayed in the figure 9 is the throughput measured after the bottleneck of the 10 Gbps link. From left to right, we present Reno, BIC, CUBIC, HighSpeed, H-TCP and Scalable TCP variants.

In our case, we can notice that the steps due to the addition of another couple of nodes get sloppier when we increase the latency. The effect is particularly noticeable on Reno (first



(a) Individual goodputs (b) Aggregate goodput

Fig. 8. H-TCP in Grid5000, 11.5 ms RTT



(a) 11.5 ms RTT

(b) 100 ms RTT

Fig. 9. Reno, BIC, CUBIC, HS-TCP, H-TCP and Scalable with various RTT in AIST-GtrcNET-10

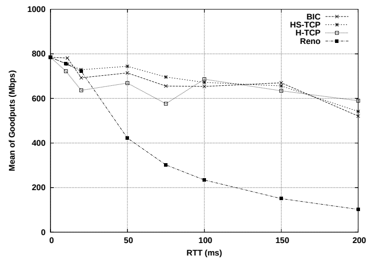
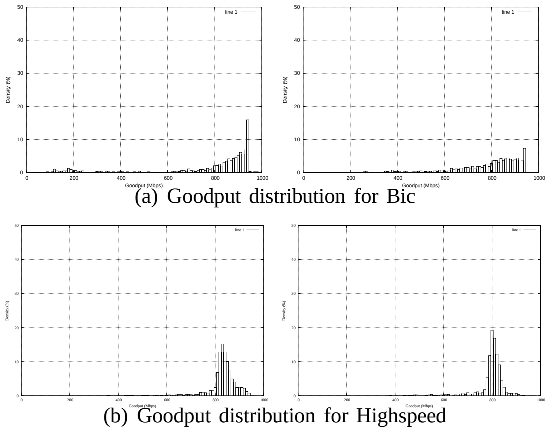


Fig. 10. Mean goodputs for Bic, HS-TCP, H-TCP and Reno when 12 nodes are active in AIST-GtrcNET-10, (1 flow/node case)



(a) Goodput distribution for Bic

(b) Goodput distribution for Highspeed

Fig. 11. Examples of Goodput distribution for 11.5 ms RTT when 12 nodes are active, in AIST-GtrcNET-10 (1 flow/node case)

	Flow mean goodput		Mean fairness		Normalised std deviation	
	11.5 ms	100 ms	11.5 ms	100 ms	11.5 ms	100 ms
Reno	756.0	234.3	0.951	0.918	0.222	0.232
BIC	781.1	653.7	0.969	0.919	0.176	0.306
CUBIC	784.5	534.3	0.974	0.961	0.144	0.140
HS-TCP	753.6	671.9	0.960	0.962	0.069	0.233
H-TCP	722.2	686.1	0.953	0.926	0.230	0.256
Scalable	674.0	540.4	0.870	0.955	0.337	0.317

TABLE II

SUMMARY OF MAIN METRICS USED, AIST-GTRCNET-10 EXPERIMENTS

column) and CUBIC (third column) as these protocols aren't able to fill the link. The deficiency observed for Reno is the well-known fact that Reno congestion control method isn't adapted to networks with high BDP product due to the slow evolution of the congestion windows in this condition.

b) Individual goodputs and fairness as a function of latency and protocol: Figures 5 and 7 show respectively Cubic on Grid5000 and GtrcNET-10 testbed with 11.5 ms RTT. Surprisingly, we can observe that Cubic behaves more fairly in the real testbed whereas we seem to have better efficiency in the emulated testbed.

Web100's log reports more retransmissions than expected in non-congested state. It appears that there was a bad interaction between the Base Board Management controller firmware of the nodes used at the *testbed 2* with the firmware version of the NICs, which caused extra losses in the flows and downgraded the results we might have achieved with this testbed.

Figure 10 presents the mean of the goodput means on the period where 12 flows are active. We can see that RTT has an important effect on the goodput as it can cause a diminution of more than 50 % of the mean goodput.

For small values of RTT, the protocols are providing equivalent results but there are more discrepancies. As the RTT increases, we experience more than 100 Mbps mean goodput difference between the different protocols. This information could help us to choose an adequate TCP variant given a RTT value. If our goal was only to maximise the goodput we can hope to achieve during a bulk data transfer, a good choice in our limited case would be to use BIC for RTTs lower than 20 ms, then switch to Highspeed for the range 20 ms to 150 ms and finally use H-TCP for the higher RTT values.

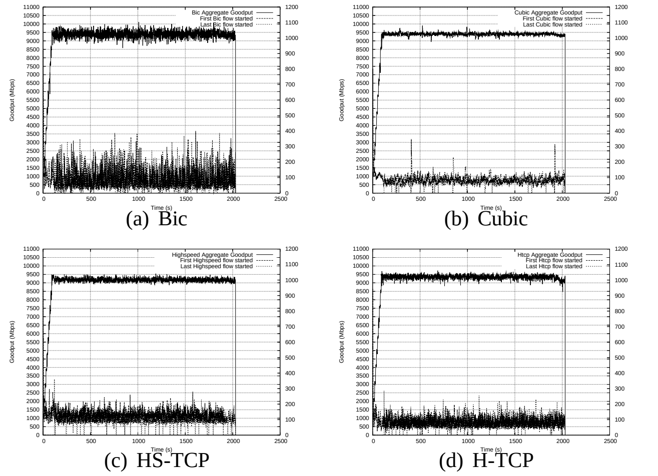
The analysis of goodput distributions shows two flow densities among the twelve. The figure 11 is representative of the different behaviours observed at 11.5 ms. We can see that the Bic distributions show an important mode close to the maximal goodput achievable (941 Mbps) for more than 30 % of the time, but there is a tail larger than 600 Mbps. Highspeed distributions look more like a Gaussian distribution, which shows that the Highspeed goodput tends to be less variable than the one obtained with Bic.

Table II summarises the results obtained during the *testbed 2* experiments with the main metrics we used. It shows their evolution with respect to the RTT. The differences for low latencies are slight. Fairness remains close to 0.95 which is a rather good value. Most protocols, except HS-TCP and BIC, have a stable normalized standard deviation.

D. Parallel streams with TCP variants

In this section, we explore the usage of parallel streams with different TCP variants in the *testbed 2*. It is done with the scenario used in section IV-A with a 11.5 ms RTT, 12 pairs of nodes and a fixed number (10) of parallel streams.

Figure 12 clearly shows that at this latency, the differences in mean aggregated goodput between each TCP variant are very slight, less than 2 %. The real difference between



(a) Bic (b) Cubic (c) HS-TCP (d) H-TCP

Fig. 12. "Parallel streams" in AIST-GtrcNET-10 with 11.5 ms RTT, 12 pairs with 10 streams each

these protocols lies in the stability due to their respective aggressiveness. For BIC, the aggregated goodput variability is more than 500 Mbps, while CUBIC's variability is less than 50 Mbps. CUBIC might be better suited if we want to have precise control of the completion predictability of a transfer.

We can also compare figure 12(a) and figure 3(d) to see that yet again, the AIST-GtrcNET-10 provides a good approximation of a real experimental testbed, as they are displaying similar mean aggregated goodput and variability.

V. Conclusion

In this article, we have explored the behaviour of TCP variants in the context of real high-speed networks of grids. We have proposed a simple methodology that could be easily reproduced everywhere. We have evaluated few metrics that helped to characterise different variants of TCP in various RTT conditions. This work also enabled to ensure that the AIST-GtrcNET-10 testbed is a good approximation of a real testbed like the one we used in Grid5000. It offers interesting extra functionalities like precise bandwidth measurement and latency emulation. Finally, we have provided a set of experimental measurements that give an insight of the performance of several TCP variants.

Our conclusion is that using parallel streams with new TCP protocols like BIC is highly valuable in this context as is increases the multiplexing level. According to the modest RTT value of the grid testbed we use, the various TCP variants we evaluated present comparable results, and Reno still behaves quite well. When the latency increases, H-TCP and HS-TCP performs better than the others in these particular conditions.

In the future, we plan to extend further our work by investigating other aspects. We intend to study the evolution of transfer time according to RTT, protocol variant and congestion level parameters as well as the RTT and interprotocol fairness problems in the "grid context". Moreover, we will measure the impact of reverse and background traffics to be as close as possible to real grid networking conditions.

VI. Acknowledgement

This work has been funded by the French ministry of Education and Research, INRIA (RESO and Gridnets-FJ teams), via ACI GRID's Grid5000 project, the IGTMD ANR grant, NEGST CNRS-JSP project. A part of this research was supported by a grant from the Ministry of Education, Sports, Culture, Science and Technology (MEXT) of Japan through the NAREGI (National Research Grid Initiative) Project and the PAI SAKURA 100000SF with AIST-GTRC.

References

[1] I. Foster, M. Fidler, A. Roy, V. Sander, and L. Winkler, "End-to-end quality of service for high-end applications," *Computer Communications*, vol. 27, no. 14, pp. 1375–1388, 2004.

[2] M. Weltz, E. He, P. Vicat-Blanc Primet, and al., "Survey of protocols other than tcp," OGF Informational Document, Open Grid Forum, Tech. Rep., April 2005, GFD 37.

[3] J. P. Martin-Flatin and P. Vicat-Blanc Primet, Eds., *High Performance Networks and Services for Grid : the IST DataTAG project experience*. Elsevier, dec 2004.

[4] R. L. Cottrell *et al.*, "Characterization and evaluation of tcp and udp-based transport on real networks," presented at 3rd International Workshop on Protocols for Fast Long-distance Networks, Lyon, France, 3-4 Feb 2005.

[5] "Tools for the evaluation of simulation and testbed scenarios," in *http://www.ietf.org/irtf/draft-irtf-tmrg-tools-02.txt*, S. Floyd and E. Kohler, Eds., June 2006.

[6] F. Cappello and al., "Grid5000: A large scale, reconfigurable, controlable and monitorable grid platform," in *GRID2005 worskhop of the IEEE SuperComputing Conference*, Nov. 2005.

[7] J. Roberts, "A survey on statistical bandwidth sharing," *Computer Networks*, Apr. 2004.

[8] Y.-T. Li, D. Leith, and R. N. Shorten, "Experimental evaluation of tcp protocols for high-speed networks," in *Transactions on Networking*, to appear 2006.

[9] 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" Pascale Vicat-Blanc, Joe Touch, Kasuchi Kobayashi Eds.*, 2006.

[10] T. Hacker, B. Noble, and B. Athey, "Improving throughput and maintaining fairness using parallel tcp," in *Proceedings of the IEEE INFOCOM*, 2004.

[11] "Gridftp: Protocol extension to ftp for the grid," in *http://forge.gridforum.org/sf/go/doc8322?nav=1*, Allcock W., Ed., April 2003.

[12] E. Altman, D. Barman, B. Tuffin, and M. Vojnovic, "Parallel tcp sockets: Simple model, throughput and validation," in *Proceedings of the IEEE INFOCOM*, 2006.

[13] D. Wei, P. Cao, and S. Low, "Time for a TCP benchmark Suite?" Caltech, Tech. Rep., 2005.

[14] R. Jain, C. D. M., and H. W., "A quantitative measure of fairness and discrimation for resource allocation in shared systems," Digital Equipment Corporation, Tech. Rep., 1984.

[15] Y. Kodama, T. Kudoh, T. Takano, H. Sato, O. Tatebe, and S. Sekiguchi, "Gnet-1: Gigabit ethernet network testbed," in *In Proceedings of the IEEE International Conference Cluster 2004*, San Diego, California, USA, September 20-23 2003.

[16] M. Mathis, J. Heffner, and R. Reddy, "Web100: extended tcp instrumentation for research, education and diagnosis," *SIGCOMM Comput. Commun. Rev.*, vol. 33, no. 3, pp. 69–79, 2003.

[17] R. Shorten and D. Leith, "H-TCP: TCP for high-speed and long-distance networks," in *PFLDnet'04*, Argonne, Illinois USA, feb. 2004.

[18] L. Xu, K. Harfoush, and I. Rhee, "Binary increase congestion control for fast long-distance networks," in *INFOCOM*, 2004.

[19] R. Guillier, L. Hablot, Y. Kodama, T. Kudoh, F. Okazaki, P. Primet, S. Soudan, and R. Takano, "A study of large flow interactions in high-speed shared networks with grid5000 and gtrcnet-10 instruments," INRIA, Research Report 6034, 11 2006. [Online]. Available: https://hal.inria.fr/inria-00116249