EuroFGI Workshop on IP QoS and Traffic Control
P. Pereira (Ed.)
Lisbon, Portugal, December 6-7, 2007

MODERN TCPS IN THE INTERNET - SURVIVAL OF THE FITTEST?[1]

Dragana Damjanovic, Michael Welzl and Kashif Munir

Institute of Computer Science, University of Innsbruck, Austria
e-mails: dragana.damjanovic@uibk.ac.at, michael.welzl@uibk.ac.at, kashif.munir@uibk.ac.at

Keywords: Traffic Control, congestion control, TCP, fairness.

Abstract. *Since June 2004, the TCP/IP stack in the Linux operating system uses an experimental TCP variant called "BIC" by default. BIC is one out of many protocols which were designed to probe for the available bandwidth more aggressively than TCP under normal circumstances, and to fall back to standard TCP behavior only if packet loss is high. This way, the network is better saturated, but a fair rate allocation between this new protocol and standard TCP is no longer given when packet loss is reasonably small. In this paper, we illustrate this fairness problem with results from real-life tests and present an overview of current proceedings in the IETF and IRTF regarding this issue, which is not only related to BIC but in fact to the general evolution of TCP and its implementation in modern operating systems.*

1 INTRODUCTION

For years TCP and UDP were the only protocols used in the Internet. The TCP specification from RFC 793 [1], published in 1981 is the foundation of the today's TCP. A little later congestion control was added [2]. Since then, standard TCP did not change a lot; it still has the same conservative congestion control mechanism that uses an Additive Increase Multiplicative Decrease (AIMD) algorithm. Over the years technology evolves, and the same is true for the Internet. There are high-speed links (links with high bandwidth-delay products) which were not present in 1981. Also multimedia applications have become more common, which led to wider use and more thorough studies of UDP. It has been shown that even a single UDP (unresponsive) flow can cause severe harm to a large number of responsive flows [3]. Under such conditions it becomes obvious that TCP's congestion control algorithm exhibits its limitations, and this led to some concerns about UDP. This opened new research areas in two distinct topics: TCP-friendly congestion control for multimedia traffic, where a smoothed rate is preferable and somewhat better-than-TCP congestion control.

Concerns about UDP and its severe harm to the TCP traffic led not just to new research developments, but also to new standards. Since TCP was the most widely used protocol on the Internet, it was tried to make UDP TCP-friendly, which means that, in steady state, it must not use more bandwidth than a conforming TCP running under comparable conditions [4]. The IETF recently standardized the DCCP protocol. DCCP enables multimedia application programmers to make use of smooth yet TCP-friendly congestion control without having to implement this function within their applications [5]. Since TCP congestion control made the Internet stable, in order to avoid Internet-wide congestion collapse, TCP-friendliness is regarded as a

[1] The work described in this paper is partially supported by the European Union through the FP6-IST-507613 project Euro-FGI and research funding for young scientists from the University of Innsbruck.

requirement not just for multimedia traffic but for all traffic. This led to a conflict: researchers are developing better-then-TCP protocols, but protocols need to be TCP-friendly to be used in the Internet.

Bandwidths that have been considered "high-speed links" years ago are now readily available to consumers in many countries, yet standard TCP is not able to fully utilize them. This led to a corresponding interest by the operating system community to deliver mechanisms to the users that can efficiently use these kinds of network paths. Linux, for example, offers to its users non-RFC-compliant congestion control mechanisms, which are not properly specified and studied – in particular, the influence of these protocol on other standard protocols may not have been given enough consideration. Linux made BIC [6] the default mechanism in 2004, and recently (since February 2007), keeping in up to date, its successor CUBIC [7]. Microsoft Windows Vista offers to customers Compound TCP [8], which was however only enabled as the default TCP mechanism in a Beta version. As we have already shown with simulations in [9], there is reason to be concerned with this mixture. In this paper we use one simple test to show that it is not just the ns-2 simulator that shows these worrying facts.

In response to these concerns the IETF and IRTF offer to be a venue for community discussion and review, followed by an eventual technical specification, of new congestion control variants. The first developments in the IETF and IRTF in this field will be presented in the section 3.

2 A FAIRNESS TEST

The intention of this paper is to analyze the throughput achieved by standard TCP and one TCP variant – BIC – when they are sharing the same path. We perform a test in an isolated test environment as well as in the real Internet, stressing the issue of how the capacity is shared between these two TCP variants.

2.1 Tests in the small testbed

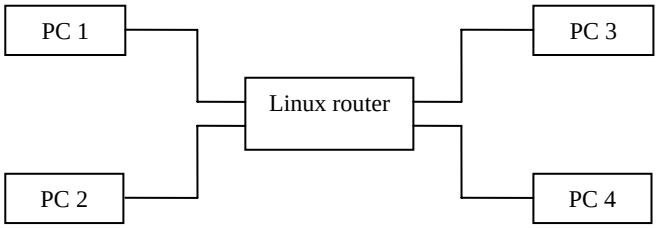

Figure 1: The Testbed.

We set up a simple test environment (figure 1). It interconnects five machines using Fast-Ethernet links (100 Mbps). All PCs run Linux (RedHat 8.0, Kernel v2.4.18). Two PCs on left are two senders and each of them has one receiver on the right. One of the PCs on the left uses standard TCP and the other uses BIC. A PC running Linux (RedHat 8.0, Kernel v2.4.18) with two network interfaces is used as a router. Each test lasted for 240 seconds. First we ran one flow using TCP-Reno from PC 1 and one flow using BIC from PC 2 starting at the same time, and then we ran tests with 5 TCP-Reno flows against 5 BIC flows and 7 standard TCP flows against 7 BIC flows. The sending rate was measured using tcpdump at the sender and running tcptrace version 6.4.2. [10] on the files produced by tcpdump.

Figure 2 shows how the gap between the rate of TCP-Reno and BIC is growing. Clearly, BIC is making much better use of the available bandwidth than TCP-Reno. Considering an Internet browser where a lot of separate connections are open, this means that a connection to a BIC website will be faster than a connection to a TCP-Reno website.

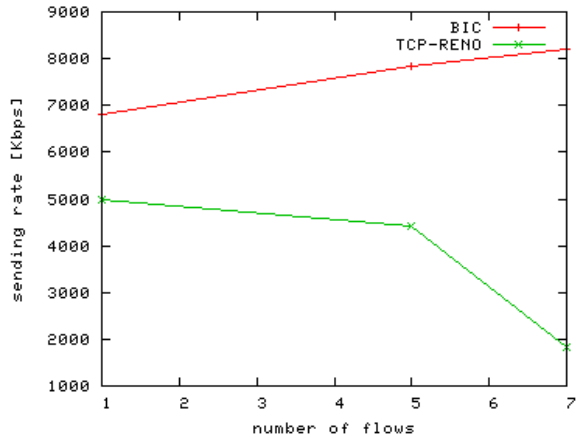

Figure 2: The Testbed results.

2.2 Planet-Lab tests

For the measurements in the Internet we used Planet-Lab [11]. We ran tests between two PCs in Innsbruck and two PCs in Brazil (planetlab1.larc.usp.br – 143.107.111.194 and planetlab2.larc.usp.br – 143.107.111.195). One of the PCs in Innsbruck used TCP-Reno and the other used BIC. Similar to the tests in the testbed, we opened 1 FTP connection from Innsbruck to Brazil from the PC that uses TCP-Reno, and, at the same time, 1 FTP connection between Innsbruck and Brazil from the PC that uses BIC. We ran two more tests with 5 TCP-Reno flows against 5 BIC flows and 10 TCP-Reno flows against 10 BIC flows. The results are shown in Figure 3.

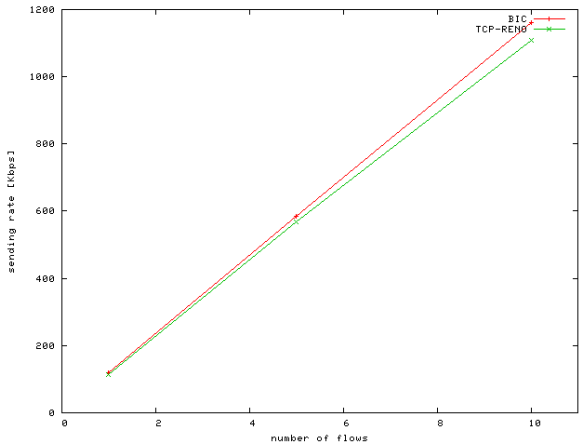

Figure 3: The Planet-Lab results.

Clearly, this behavior is altogether different from what we have seen in our local testbed – but there is a simple explanation for this seemingly strange phenomenon: the rates of both the BIC and the TCP-Reno flows were limited by the receiver windows. We know this because we also ran tests from Innsbruck to the hosts in Brazil as well as several other hosts in PlanetLab with the Web 100 kernel [12] for a different measurement study that we carried out [13]. The Web 100 kernel clearly showed that the receiver limited the sending rate in all of our tests; as a matter of fact, for the measurement study documented in [13], we had to patch our kernel in order to make the TCP sender ignore the receiver window, which at least worked for a certain duration (after which the other side terminated the connection).

In our case, the reason for this receiver window limitation is the lack of use of the window scaling option [14], which would be needed because the size of the receiver window field in the TCP header is too small for high speed networks. In the Internet, it seems that this option is not frequently (or properly) used; in [15], the following report is given:

"Additionally, 7,540 clients (or 26.6% of our dataset) advertised support for TCP's window scaling option, which calls for the advertised window to be scaled by a given factor to allow for larger windows than can naturally be advertised in the given 16 bits in the TCP header. Just over 97% of the clients that indicate support for window scaling advertise a window scale factor of zero — indicating that the client is not scaling its advertised window (but understands window scaling if the server wishes to scale its window). Just over 1% of the clients in our dataset use a scale factor of 1, indicating that the advertised window in the client's segments should be doubled before using. We observed larger window scale factors (as high as 9) in small numbers in our dataset."

We do not have an explanation why the window scaling option is not widely used and [15] do not give any clarification. To the best of our knowledge, this limitation has not been thoroughly investigated yet, and is not being taken into account in the IETF and IRTF, where there is now a major concern about the fairness between experimental TCP variants and standard TCP. These bodies have come up with their own way for addressing the problem, which we will elaborate on in the next section.

3 A PROPOSED IETF/IRTF REVIEWING PROCESS

The whole process of bringing new congestion control mechanisms was defined during the IRTF's Internet Congestion Control Research Group (ICCRG) meeting which was held on 12/13 February 2007 in Marina del Rey, CA, USA, and agreed upon at the TSV Area meeting ("Transport Area Open Meeting") which was held during the 68th IETF meeting in Prague, Czech republic, 18-23 March 2007. The document [16], which describes the outcome of these meetings will be used as a guideline; in what follows, we provide a brief abridgement of this document.

The goal of the newly defined process is to publish documents with statements that indicate that the IETF believes that a certain mechanism is safe for experimentation on the Internet, or safe for experimentation in certain more restricted network environments. The proposal is to use the expertise in the IRTF's Internet Congestion Control Research Group (ICCRG) during the initial phase of this review before bringing it to IETF. The proposers of high-speed congestion control variants will present their mechanism and results to the ICCRG, which will analyze and discuss these variants with the goal of determining whether they can be declared safe for experimentation, and under which conditions.

Document [17] defines models that can be used in the evaluation of transport protocols. These models will be used by ICCRG during reviews of new congestion control mechanisms. In document [17] a set of metrics is defined that should be use for characterizing a congestion control scheme; such a mechanism should be evaluated in terms of tradeoffs between a range of metrics, rather than in terms of optimizing a single metric. The document includes metrics that describe a protocol itself (throughput, delay, loss rates, response times, minimizing oscillations), the metrics that measure stability of a protocol and how safe it is to use it on the

Internet (robustness for challenging environments, robustness to failures and to misbehaving) as well as the metrics that measure behavior of a protocol in the presence of other flows of the same kind or different ones (fairness and deployability).

For any new proposal a serious scientific study of the pros and cons of the proposal needs to have been done such that the IETF has a well rounded set of information to consider. As a guideline the IETF published the document [18] with a set of criteria that should be considered. As defined in this document for each new proposed congestion control variant the following thorough studies should be provided:

1. Differences with Congestion Control Principles [19]
2. Impact on standard TCP, SCTP [20], and DCCP [5]
3. Difficult environments (behavior of a protocol in difficult environments, like wireless environments, environments with multipath routing, with long delay, etc.)
4. Investigating a range of environments (proposals should be investigated across a range of bandwidths, round-trip times, levels of traffic on the reverse path, and levels of statistical multiplexing at the congested link etc.)
5. Protection against congestion collapse (The alternate congestion control mechanism should either stop sending when the packet drop rate exceeds some threshold RFC3714 [21], or should include some notion of "full backoff" and does not require that the full backoff mechanism must be identical to that of TCP)
6. Fairness within the alternate congestion control algorithm
7. Performance with misbehaving nodes and outside attackers
8. Responses to sudden or transient events (The proposal should consider how the alternate congestion control mechanism would perform in the presence of transient events such as sudden congestion, a routing change, or a mobility event.)
9. Incremental deployment (The proposal should discuss whether the alternate congestion control mechanism allows for incremental deployment in the targeted environment.)

The minimum requirements for approval for widespread deployment in the global Internet include guidelines (1) on assessing the impact on standard congestion control, (3) on investigation of the proposed mechanism in a range of environments, guideline (4) on protection against congestion collapse and guideline (8), discussing whether the mechanism allows for incremental deployment. Other guidelines ((2), (5), (6) and (7)) should be evaluated as well, and according to these analyses the ICCRG should declare concerns in approval for widespread deployment in the global Internet. It is unclear if BIC and CUBIC would pass the evaluation

4 CONCLUSION

As future work we plan to do more thorough simulations and measurements and we plan to compare some other TCP versions, but still our simulations and real life test results, presented in this paper, indicate that there is indeed a reason of concern about fairness issues in the Internet. The IRTF/IETF reviewing process will not be enough to ensure fair and efficient operation of the Internet, as neither the IETF nor the IRTF can act as an Internet "police" – anybody is free to implement any kind of congestion control mechanism in his or her end system. Due to the inefficiency of standard TCP over high-speed links there is always incentive for operating system designers not to following the IETF/IRTF recommendations and to come up with aggressive protocols.

Our tests show an alarming situation that can happen in the Internet which for the time being seems to be avoided due to the fact that the window scaling option is not used much. There are a lot of question marks: if it is only this option which prevents the aggressiveness of mechanisms like BIC/CUBIC from acting unfairly towards standard TCP flows, then what will happen if all of a sudden people start to use Window Scaling? Will we then face the same situation in the Internet that we have seen in our testbed? Do we need a policing body/entity which can ensure TCP-friendliness? Who will do

it and how will it be done? While we could not answer these questions in this paper, we hope that it contributes to raising the awareness of this urgent problem that the Internet may face in the future.

REFERENCES

[1] Postel, J., “Transmission Control Protocol”, RFC 793, September 1981.

[2] Jacobson, V., Karels, M. J., “Congestion Avoidance and Control”, In Proceedings of ACM SIGCOMM '88, Stanford, CA, August 1988.

[3] Sally Floyd, Kevin R. Fall, “Promoting the use of end-to-end congestion control in the internet,” IEEE/ACM Transactions on Networking, 7(4):458--472, 1999.

[4] B. Braden, D. Clark, J. Crowcroft, B. Davie, S. Deering, D. Estrin, S. Floyd, V. Jacobson, G. Minshall, C. Partridge, L. Peterson, K. Ramakrishnan, S. Shenker, J. Wroclawski, L. Zhang, “Recommendations on Queue Management and Congestion Avoidance in the Internet,” April 1998, RFC 2309.

[5] E. Kohler, M. Handley, S. Floyd, “Datagram Congestion Control Protocol (DCCP),” March 2006, RFC 4340.

[6] L. Xu, K. Harfoush, and I. Rhee, “Binary Increase Congestion Control (BIC) for Fast Long-Distance Networks,” In Proceedings of IEEE INFOCOM 2004, March 2004

[7] I. Rhee, L. Xu, “CUBIC: A New TCP-Friendly High-Speed TCP Variant, “In Proceedings of Third International Workshop on Protocols for Fast Long-Distance Networks (PFLDnet), February 2005, Lyon, France.

[8] K. Tan, J. Song, Q. Zhang, and M. Sridharan, “Compound TCP: A Scalable and TCP-friendly Congestion Control for High-speed Networks”, in 4th International Workshop on Protocols for Fast Long-Distance Networks (PFLDnet), 2006, Nara, Japan.

[9] Kashif Munir, Michael Welzl, Dragana Damjanovic, "Linux beats Windows! – or the Worrying Evolution of TCP in Common Operating Systems", PFLDnet 2007 (Fifth International Workshop on Protocols for Fast Long-Distance Networks), 7/8 February 2007, Marina Del Rey (Los Angeles), USA.

[10] http://www.tcptrace.org/

[11] http://www.planet-lab.org/

[12] http://www.web100.org/

[13] Dragana Damjanovic, Werner Heiss, Michael Welzl, “An Extension of the TCP Steady-State Throughput Equation for Parallel TCP Flows”, poster, accepted for presentation at ACM SIGCOMM 2007, 27-31 August, Kyoto, Japan.

[14] V. Jacobson, R. Braden and D. Borman, “TCP Extensions for High Performance”, RFC 1323, 1992.

[15] Alberto Medina and Mark Allman and Sally Floyd, “Measuring the Evolution of Transport Protocols in the Internet”, ACM Computer Communications Review 35(2): 37-52, April 2005.

[16] Eggert L., “DRAFT ION: Experimental Specification of New Congestion Control Algorithms”, ion-tsv-alt-cc (work in progress) April 2007

[17] Floyd S., “Metrics for the Evaluation of Congestion Control Mechanisms”, draft-irtf-tmrg-metrics-08 (work in progress), March 2007.

[18] Floyd S. and Allman M., “Specifying New Congestion Control Algorithms”, draft-ietf-tsvwg-cc-alt-00 (work in progress), March 2007.

[19] Floyd S., “Congestion Control Principles”, RFC 2914, Best Current Practice, September 2000.

[20] Stewart R., Xie Q., Morneault K., Sharp C., Schwarzbauer H., Taylor T., Rytina I., Kalla M., Zhang L., and V. Paxson, “Stream Control Transmission Protocol”, RFC 2960, October 2000.

[21] S. Floyd and J. Kempf, “IAB Concerns Regarding Congestion Control for Voice Traffic in the Internet”, RFC 3714, March 2004.