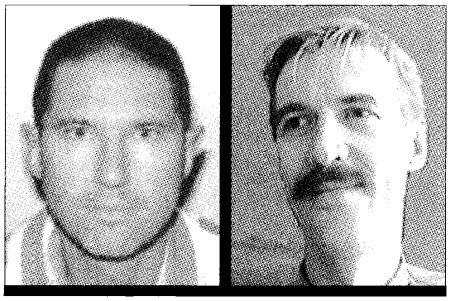
Control of Best Effort Traffic



Christophe Diot Jean-Yves Le Boudec

The Internet is almost exclusively based on the concept of best effort transmission. The simplicity of this model certainly played a key role in the deployment of a ubiquitous Internet service. However, a completely uncontrolled network may suffer from *congestion collapse*, which occurs when the offered load locally exceeds the available bandwidth. Such collapses already occurred in the mid-'80s, when the Internet had no deployed congestion avoidance. Repeated congestion problems triggered the definition and implementation of congestion control functions in TCP (thus, end to end), similar to those found in DECNET. Their original design required avoiding modifications to routers; congestion control was introduced as a set of functions at sources and destinations, whereas routers continued to handle all traffic in an aggregate way. With some minor modifications, these mechanisms are still used to:

- Protect the Internet from congestion collapse
- Make all users share the available bandwidth in a "fair" way

The efficiency of these mechanisms depends heavily on user equipment correctly implementing congestion control functions.

This special issue is dedicated to the control of best effort traffic. We tried to put together a review of the state of the art in concepts, methods, and algorithms that avoid the development of congestion collapse while fairly sharing bandwidth between users, whatever application or transport protocol they are using.

The first article, by Panos Gevros, Jon Crowcroft, Peter Kirstein, and Saleem Bhatti, "Internet Congestion Control: Principles and Mechanisms," gives a global perspective on traffic control functions. It describes the different techniques used to detect and avoid congestion, and to provide some fairness between flows.

With the increasing growth of non-TCP traffic (e.g., media streaming), congestion control had to be extended to non-TCP flows. More precisely, the Internet Engineering Task Force (IETF) mandates that a non-TCP flow not send more than a TCP flow would under similar network conditions. If so, a flow is said to be *TCP-friendly*. The principle of TCP friendliness may seem simple. However, it may not be that simple to implement. First, it requires understanding the performance of TCP. Some simple closed-form formulas have been found; for example, the "square root formula" [1]. To be tractable, these equations rely on crude modeling assumptions. Second, forcing non-TCP flows to adopt TCP-like behavior is not straightforward. A simple idea is equation-based congestion control: a non-TCP source attempts to track an equation such as the square root formula. How to do this in detail may rapidly become complex; furthermore, it is not always guaranteed to work [2]. An overview of methods for implementing TCP friendliness is given in the article by Jörg Widmer, Robert Denda, and Martin Mauve, "A Survey on TCP-Friendly Congestion Control." The article by Chadi Barakat, "TCP/IP Modeling and Validation," explains some of the difficulties in modeling TCP behavior and surveys the existing results.

End-to-end traffic control functions rely on simple router-level mechanisms to detect congestion or the onset of congestion. Packet scheduling and buffer management in network nodes affect the performance perceived by users. Since its origin, the Internet is based on first-in first-out packet scheduling. Thus, the natural congestion indication signal is packet loss (a router is congested when its buffers overflow). Active queue management has been proposed to:

- Send a congestion signal before the router queue is full
- Increase the fairness among users as bursty flows were penalized by packet losses due to buffer overflows

The first incarnation of active queue management was Random Early Detection [2]. The article by Sanjeewa Athuraliya, Victor H. Li, Steven H. Low, and Qinghe Yin, "REM: Active Queue Management," reviews active queue management objectives and propose a novel mechanism that addresses the problem in a more fundamental way than RED.

Posing TCP friendliness as a requirement is somehow arbitrary; indeed, the congestion control functions of TCP were developed in a brilliant but ad hoc fashion, and the resulting fairness is not necessarily what one might desire. The performance of a TCP connection is heavily dependent on its round-trip time (roughly speaking, it is inversely proportional). This is not to be confused with the fact that a source using many hops should receive less throughput [4, 5]. Consider, for example, two users accessing a Web site, one via a satellite link, the other via asynchronous digital subscriber line (ADSL). If the bottleneck for both users is not in their access links, the

satellite user will have less throughput, because its round-trip time is larger, even if both users have the same number of hops to the destination. Is this fair? Certainly not, and fixes have been suggested [6], but they remain ignored by current TCP friendliness practice. Note that TCP friendliness is also made more difficult to implement by strong disparities in flow size and duration that make connections compete unfairly for bandwidth.

Per-flow queuing is an alternative to active queue management; it has a long history: it was found, for example, in SNA. The reference work on that topic remains [7]. It has been unused for many years because of perceived implementation complexity and its potential lack of scalability. However, recent progress in technology has made per-flow queuing feasible on fairly high-bandwidth links. Unfortunately, per-flow queuing does not seem to be a popular approach yet for traffic control in the Internet, and one regrets that the research community is not more aggressive in challenging the existing practice based on aggregate scheduling.

Unlike reservation-based services, the very nature of best effort services implies that the definition of a flow is not clearly defined. A malicious user can open 10 parallel TCP connections to a single Web site to fasten the transmission of a single object. To overcome this problem, it has been suggested that a flow be identified as a [source, destination] pair of IP addresses. This solution is partial since it does not solve the case where a user would download its object through multiple replicated sites (such as Web caches). It would also penalize users that tunnel their data into a unique address pair (i.e., one behind a NAT server, or IPv6 packets encapsulated in IPv4 headers). Other mechanisms such as encryption would make it impossible to demultiplex these flows in intermediate routers. Here again, it is unfair to treat such a tunnel as one flow, but it is not clear whether there is a unique solution to such ambiguities.

Actually, TCP fairness might be an unfriendly notion to Internet service providers (ISPs). Assuming that an ISP can, by adequate provisioning, ensure that losses remain rare and congestion collapses are unlikely to occur in its backbone, any traffic is friendly since it generates revenue. Note that practically all non-TCP traffic today in the Internet is not TCP- friendly.

The last two articles explore new areas in the control of best effort traffic. "Distributed Control and Resource Marking Using Best Effort Routers," by Richard Gibbens and Peter Key, proposes to let users make their own set of services out of a single best effort network, by defining how they react to the feedback (loss or packet marking) received from the network. "ABE: Providing a Low-Delay Service Within Best Effort," by Paul Hurley, Mourad Kara, Jean-Yves Le Boudec, and Patrick Thiran, proposes how a best effort network could offer a low-delay service without any form of access control or differentiated charging.

All articles in this special issue were refereed by at least three independent reviewers. Jörg Liebeherr acted as editor for articles with a conflict of interest. We thank all reviewers for their valued contribution.

References

[1] M. Mathis *et al.*, "The Macroscopic Behavior of the TCP Congestion Avoidance Algorithm," *Comp. Commun. Rev.*, vol. 27, no. 3, July 1997.

[2] M. Vojnovic and J.-Y. Le Boudec, "Some Observations on Equation-Based Rate Control," Tech. rep. DSC200109, EPFL-DSC, http://dscwww.ep .ch./EN/publications/documents/tr01 009.ps, Jan. 2001.

[3] S. Floyd, "Random Early Detection Gateways for Congestion Avoidance," http://www-nrg.ee.lbl.gov/floyd/red.html

[4] S. Floyd, "Connections with Multiple Congested Gateways in Packet Switched Networks, Part 1: One Way Traffic," *ACM Comp. Commun. Rev.*, vol. 22, no. 5, Oct. 1991, pp. 30–47.

[5] M. Vojnovic, J.-Y. Le Boudec, and C. Boutremans, "The Fairness of Additive Increase, Multiplicative Decrease with Heterogeneous Round Trip Times," *Proc. IEEE INFOCOM 2000*, Apr. 2000.

[6] T. R. Henderson *et al.*, "On Improving the Fairness of TCP Congestion Avoidance," *Proc. IEEE GLOBECOM '98*, Sydney, Australia, Nov. 1998.

[7] S. Keshav, "A Control-Theoretic Approach to Flow Control," *ACM SIGCOMM '91*, Aug. 1991, pp. 189–201.

Biographies

Christophe Diot (cdiot@sprintlabs.com) received a Ph.D. degree in computer science from INP Grenoble in 1991. From 1993 to 1998 he was a research scientist at INRIA Sophia Antipolis, working on new Internet architecture and protocols. He moved to Sprint Advanced Technology Laboratory in October 1998 to take the lead of the IP research group (http://www.sprintlabs.com). His current interest is in deployable multicast (SSM) and Internet resource control. The major project at Sprint is passive monitoring of the Sprint IP backbone in order to study IP traffic characteristics. He is a member of ACM, the COST 264 action, and serves as an editor for *ACM/IEEE Transactions on Networking*.

JEAN-YVES LE BOUDEC [M'89] (leboudec@epfl.ch) is full professor at EPFL, Switzerland. He graduated from Ecole Normale Superieure de Saint-Cloud, Paris, France, where he obtained the Agregation in mathematics in 1980. He received his doctorate in 1984 from the University of Rennes, France, and became an assistant professor at INSA/IRISA, Rennes. In 1987 he joined Bell Northern Research, Ottawa, Canada, as a member of scientific staff in the Network and Product Traffic Design Department. In 1988 he joined the IBM Zurich Research Laboratory, Rueshchlikon, Switzerland, where he was manager of the Customer Premises Network Department. He joined EPFL in 1994. His interests are in the architecture and performance of communication systems.