

Computer system automatically generates TCP congestion-control algorithms

July 18 2013, by Larry Hardesty

TCP, the transmission control protocol, is one of the core protocols governing the Internet: If counted as a computer program, it's the most widely used program in the world.

One of TCP's main functions is to prevent [network congestion](#) by regulating the rate at which computers send data. In the last 25 years, engineers have made steady improvements to TCP's congestion-control algorithms, resulting in several competing versions of the protocol: Many Windows computers, for instance, run a version called Compound TCP, while Linux machines run a version called TCP Cubic.

At the annual conference of the Association for Computing Machinery's Special Interest Group on Data Communication this summer, researchers from MIT's Computer Science and Artificial Intelligence Laboratory and Center for Wireless Networks and Mobile Computing will present a computer system, dubbed Remy, that automatically generates TCP congestion-control algorithms. In the researchers' simulations, algorithms produced by Remy significantly outperformed algorithms devised by human engineers.

"I think people can think about what happens to one or two connections in a network and design around that," says Hari Balakrishnan, the Fujitsu Professor in Electrical Engineering and Computer Science, who co-authored the new paper with graduate student Keith Winstein. "When you have even a handful of connections, or more, and a slightly more complicated network, where the [workload](#) is not a constant—a single file

being sent, or 10 files being sent—that's very hard for human beings to reason about. And computers seem to be a lot better about navigating that search space."

Lay of the land

Remy is a machine-learning system, meaning that it arrives at its output by trying lots of different possibilities, and exploring further variations on those that seem to work best. Users specify certain characteristics of the network, such as whether the bandwidth across links fluctuates or the number of users changes, and by how much. They also provide a "traffic profile" that might describe, say, the percentage of users who are browsing static webpages or using high-bandwidth applications like videoconferencing.

Finally, the user also specifies the metrics to be used to evaluate network performance. Standard metrics include throughput, which indicates the total amount of data that can be moved through the network in a fixed amount of time, and delay, which indicates the average amount of time it takes one packet of information to travel from sender to receiver. The user can also assign metrics different weights—say, reducing delay is important, but only one-third as important as increasing throughput.

Remy needs to test each candidate [algorithm](#)'s performance under a wide range of network conditions, which could have been a prohibitively time-consuming task. But Winstein and Balakrishnan developed a clever algorithm that can concentrate Remy's analyses on cases in which small variations in network conditions produce large variations in performance, while spending much less time on cases where network behavior is more predictable.

They also designed Remy to evaluate possible indicators of network congestion that human engineers have not considered. Typically, TCP

congestion-control algorithms look at two main factors: whether individual data packets arrive at their intended destination and, if they do, how long it takes for acknowledgments to arrive. But as it turns out, the ratio between the rates at which packets are sent and received is a rich signal that can dictate a wide range of different behaviors on the sending computer's end.

Down to cases

Indeed, where a typical TCP congestion-control algorithm might consist of a handful of rules—if the percentage of dropped packets crosses some threshold, cut the transmission rate in half—the algorithms that Remy produces can have more than 150 distinct rules.

"It doesn't resemble anything in the 30-year history of TCP," Winstein says. "Traditionally, TCP has relatively simple endpoint rules but complex behavior when you actually use it. With Remy, the opposite is true. We think that's better, because computers are good at dealing with complexity. It's the behavior you want to be simple." Why the algorithms Remy produces work as well as they do is one of the topics the researchers hope to explore going forward.

In the meantime, however, there's little arguing with the results. Balakrishnan and Winstein tested Remy's algorithms on a simulation system called the ns-2, which is standard in the field.

In tests that simulated a high-speed, wired network with consistent transmission rates across physical links, Remy's algorithms roughly doubled network throughput when compared to Compound TCP and TCP Cubic, while reducing delay by two-thirds. In another set of tests, which simulated Verizon's cellular data network, the gains were smaller but still significant: a 20 to 30 percent improvement in throughput, and a 25 to 40 percent reduction in delay.

Provided by Massachusetts Institute of Technology

Citation: Computer system automatically generates TCP congestion-control algorithms (2013, July 18) retrieved 25 April 2024 from <https://phys.org/news/2013-07-automatically-tcp-congestion-control-algorithms.html>

This document is subject to copyright. Apart from any fair dealing for the purpose of private study or research, no part may be reproduced without the written permission. The content is provided for information purposes only.