TCP EX MACHINA : Computer Generated Congestion Control
By : Shrikrishna Bhat

This paper stands out as one of the best papers in End-to-End congestion control on multi-user networks. Here the Researchers at MIT have developed a program called Remy which is used to generate congestion control algorithms to run at endpoints. Mainly they are trying to achieve high throughput and low queuing delay. Here mainly Remy produces distributed algorithms. They have compared this Remy algorithm to its counterparts that is TCP Cubic, Vegas, etc and it produced great results and outperformed it.

The authors have primarily discussed congestion control in this paper and how the algorithm is efficiently developed to prevent congestion control. The basic principle of this paper is that instead of humans computing and preventing congestion, machines will compute and automatically handle congestion. Mainly this paper presents the idea of developing the known TCP and providing more reliable and seamless communication. The authors have also mentioned how TCP designers must think about its dynamic behavior when a packet enters and leaves the network.

Next, the author explains briefly the experiment they conducted and how they designed their network to increase reliability. The authors have mentioned the design specification and how they should define the upper and lower bounds. Next, a traffic model is designed and batch processing is carried out. Thirdly, there is a modeled network scenario to design congestion control algorithms and it is implemented on endpoints. The results of their experiment were successful and Remy provided much reliability and it was completely end-to-end. One thing to note here is that authors have mentioned that end-to-end algorithms calculate the congestion window, as well as Round Trip Time (RTT) using the stream of acknowledgments from the receiver. If there is congestion from packet loss, then the sender reduces the window and when there is no congestion, the sender increases its window. The next aspect of the paper is how Remy creates a congestion control algorithm. In this scenario, they have only evaluated the endpoints, which is sufficient for developing an efficient algorithm. Usually, for any given network there should be an optimal congestion control scheme. Here the author introduces a rule table to improve the optimization of Remy's design. In the case of optimization link rate, network delay, total sources, and conditions like these are considered to develop the efficient algorithm.

The authors used their Remy to construct additional RemyCCs and the results are astonishing. They have also mentioned that the newly developed RemyCC beats the Cube-over-sfqCoDel and this new RemyCC vastly increases performance over a wide variety of networks. Although an efficient algorithm is developed the authors are not sure whether this algorithm will work in all cases in a network and there are many unanswered questions. One of the unanswered questions tests the robustness of the RemyCC system. Even this was not tested on real networks and there could be a chance of this failing catastrophically.

Overall this paper provides details about congestion control and how it is optimized. But the things in this paper are not completely tested in real-world networks but done on an ns2 simulator. This paper gives a brief idea of congestion control but a little difficult to understand how it works correctly. In conclusion, the authors have worked hard and produced a great paper and a great algorithm to increase network reliability and robustness.