

TCP-friendly Rate Control for non-TCP Multipath Flows

Christopher Pluntke
University College London
c.pluntke@ee.ucl.ac.uk

Miguel Rio
University College London
m.rio@ee.ucl.ac.uk

ABSTRACT

In this paper we study the design of a rate controller for TCP-friendly non-TCP flows using multiple end-to-end paths simultaneously. Previous work has shown that it is possible to design a rate controller for multipath TCP that is friendly to standard TCP traffic while improving throughput over single path connections. We argue that this rate controller is not optimal for every type of non-TCP traffic. We propose a new rate controller that adjusts the subflow rates dynamically according to the solution of an optimisation problem. We show that it improves goodput over currently known single-path and multi-path controllers while being fair to standard TCP traffic even in the case of not observable shared bottleneck links in the network.

1. INTRODUCTION

When designing a rate controller for non-TCP traffic, the sharing properties of TCP on bottleneck links are used as a measure of fairness. TCP traffic adapts its sending rate according to the path characteristics it observes. This enables TCP flows to share bottleneck links fairly without direct knowledge about where bottlenecks are. Since most of Internet traffic is TCP, some kind of rate control is desirable for non-TCP traffic since overly aggressive flows can cause competing TCP flows to overly reduce their rates. In this paper we explore the question how to design a TCP-friendly rate controller for non-TCP traffic using multiple end-to-end paths simultaneously.

We found that single-path TCP-friendly rate control [1] does not directly extend to the multipath case. Furthermore, the TCP rate controller for multiple subflows which is currently being standardised by the IETF [2–4] does not cope well with some forms of non-TCP traffic, for example stream-

ing traffic. In this paper we describe our solutions.

2. THE RATE CONTROLLER FOR MULTIPATH TCP CAN GET LOW GOODPUT.

In general, TCP adapts its sending rate, x^{TCP} , depending on roundtrip time, RTT, and loss rate, p , according to the following formula: $x^{TCP} = \frac{\sqrt{2}}{RTT\sqrt{p}}$. The standard way to make a UDP flow TCP friendly is to measure loss rate and RTT and then adapt its rate to the rate TCP would get. This approach is called equation-based rate control and it has been intensely studied for single path connections in [1].

The multipath case is more delicate since there is not one single rate that is considered TCP friendly. For a multipath congestion controller to be considered TCP-friendly means that its subflows cannot be more aggressive than standard TCP even if they share common bottleneck links among themselves. Just think of a video stream connected through a single outgoing interface to 10 access servers simultaneously. It is surely not fair if it got 10 times the throughput on the end-user's access network, draining any competing TCP flow.

Formally, this means that an TCP-friendly multipath flow using a set of paths \mathcal{R} cannot transmit with higher rate than standard TCP would get on the best path of any subset of \mathcal{R} :

$$\text{for all } \mathcal{S} \subseteq \mathcal{R} : \sum_{r \in \mathcal{S}} x_r \leq \max_{r \in \mathcal{S}} x_r^{TCP} \quad (1)$$

This concept of fairness has been studied in [4] where a suitable additive increase/multiplicative decrease style rate controller for multipath TCP (MPTCP) has been introduced.

On our search for a good rate controller for non-TCP multipath traffic, we first go the standard way as we did in the single-path case above. We measure roundtrip time, RTT_r , and loss rate, p_r for path r , and adapt the subflow rates to the rate, x_r , that MPTCP would get [4]:

$$x_r = \sqrt{2a} \frac{1/p_r}{RTT_r \sqrt{\sum_s \frac{1}{p_s}}} \text{ with } a = \sum_r x_r RTT_r \frac{\max_r x_r / RTT_r}{(\sum_r x_r)^2}.$$

However, MPTCP's rate allocation can lead to suboptimal results. For non-TCP traffic throughput does not always equal goodput. For real-time video for example, a high latency packet is as good as lost, and so one of the main challenges in constructing a real-time video service is to cope with variable latency. A suitable rate controller should take

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ACM CoNEXT Student Workshop, December 6, 2011, Tokyo, Japan.

Copyright 2011 ACM 978-1-4503-1042-0/11/0012 ...\$10.00.

RTT distributions into account when choosing the subflow rates. However, the MPTCP controller does not explicitly take any special application requirements into account.

As an illustration for this behaviour, let us assume a multipath connection between sender and receiver using three subflows. RTTs are assumed to be exponentially distributed with given mean for each subflow and losses are independent. Let the mean RTTs and loss rates be as follows: ($p_1 = 10\%$, $RTT_1 = 20ms$), ($p_2 = 2\%$, $RTT_2 = 100ms$), ($p_3 = 1\%$, $RTT_3 = 300ms$). The resulting rates that the MPTCP controller gives are: $x_1 = 88.2$ [packets/s], $x_2 = 88.2$ [packets/s], $x_3 = 47.1$ [packets/s]. We implemented a simulation of a livestreaming application with a deadline constraint of 300 ms. The system sends a fraction of redundant packets as forward error correction (FEC) to protect against losses and late packets and then splices the packets across the three subflows. Running a simulation of the system with the MPTCP rates leads to a receiver goodput of 133 [packets/s] using high FEC redundancy of 40% due to frequent timeouts.

3. AN OPTIMISATION BASED RATE CONTROLLER MAXIMISES GOODPUT.

In general, this suggests the design of a generic TCP-friendly congestion controller for multipath flows that can be described as an optimisation problem that calculates the optimal subflow rates subject to staying TCP-friendly. The objective function of the optimisation problem is maximise a concave utility function. In the case of our streaming example with deadlines the objective is to maximise receiver goodput in the presence of variable RTTs and given loss rates. This means the receiver keeps track of mean loss rates as well as mean and variance of RTTs for each subflow. Using Chebychev's inequality, we bound the timeout rate per subflow given the deadline and can compute the worst case receiver loss assuming that timeout losses and drops are independent. The feasible region of the optimisation problem is the set of all TCP-friendly rates as described by (1). The optimal TCP-friendly rate allocation often forces some subflows to send nothing while other subflows carry most of the traffic. This makes the algorithm blind to any changes on those subflows. Hence, we enforce that a small fixed fraction of the total throughput is used for probing. This optimisation problem is linear and can be solved efficiently in the timescale of milliseconds.

Figure 1 shows traces of a simulation run of the example streaming system with three subflows. We ran it with the standard MPTCP rate controller and switched to our optimisation based controller in the middle of the simulation run. In agreement with (1), the rate controller chooses to send almost all traffic on path 1 in this case. With this choice of rates, the overall throughput is the same, but the receiver goodput is 173 [packets/s] while FEC redundancy stays at 22%. This is an increase in goodput of 30% in comparison to the MPTCP controller.

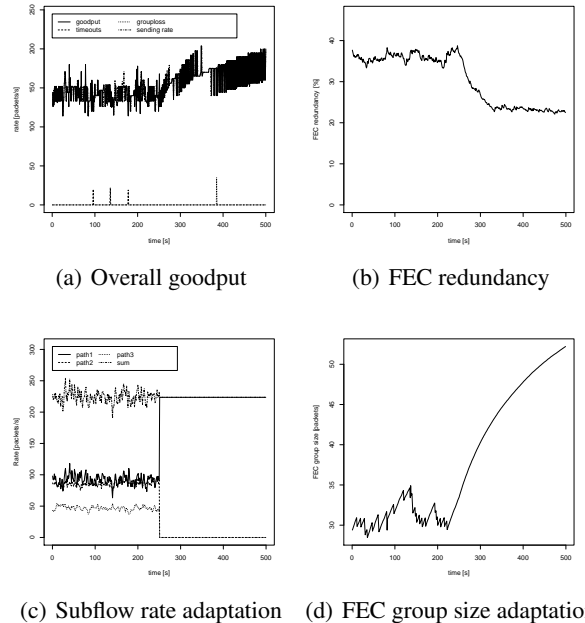


Figure 1: A simulation run starting with the standard MPTCP rate controller and switching to an optimised controller in mid-way at $t=250s$.

4. CONCLUSION

We studied the problem of TCP-friendly rate control non-TCP connections using multiple paths. We argued that standard equation-based rate control to adapt subflow rates can lead to low throughput in the multipath case. Instead, we introduced an alternative rate controller that is able to take application requirements into account when choosing its rate and showed that it gives better performance while staying TCP-friendly.

Acknowledgements

The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) in the ENVISION project, grant agreement 248565.

5. REFERENCES

- [1] S. Floyd, M. Handley, J. Padhye, and J. Widmer. Equation-based Congestion Control for Unicast Applications. In *Proceedings of SIGCOMM*, pages 43–56, New York, NY, USA, 2000. ACM.
- [2] A. Ford, C. Raiciu, M. Handley, S. Barre, and J. Iyengar. Architectural Guidelines for Multipath TCP Development. Internet-Draft draft-ietf-mptcp-architecture-05, Internet Engineering Task Force, Jan. 2011. Work in Progress.
- [3] C. Raiciu, M. Handley, and D. Wischik. Coupled Congestion Control for Multipath Transport Protocols. Internet-Draft draft-ietf-mptcp-congestion-01, Internet Engineering Task Force, Jan. 2011. Work in Progress.
- [4] D. Wischik, C. Raiciu, A. Greenhalgh, and M. Handley. Design, implementation and evaluation of congestion control for multipath TCP. In *Proceedings of NSDI*, 2011.