Decoupling Congestion Control using Traffic Aggregates and Middleboxes

Niklas Neumann, Ralf Lübben, Mayutan Arumaithurai (students) and Xiaoming Fu (faculty)
Computer Networks Group, University of Goettingen
{neumann,luebben,arumaithurai,fu}@cs.uni-goettingen.de
http://www.net.informatik.uni-goettingen.de/research_projects/dcc

I. Introduction

A rise in numbers of large bandwith-delay product links and an increasing heterogeneity of IP networks bring new challenges for the existing congestion control mechanisms. Congestion control mechanisms are traditionally end-to-end oriented. This makes them slow to react on high delay links and inaccurate if the flow traverses heterogeneous network segments that have different jitter, or packet loss rates. Furthermore, the slow-start which TCP uses as part of its congestion control is slow to react on high bandwith-delay product links and makes it hard for short-lived flows to develop fully.

There are a number of approaches to solve those problems. They can be divided in two groups: end-host measurement based mechanisms, for example, TCP Cubic or TCP Compound, and hop-by-hop discovery approaches, for example, eXplicit Control Protocol (XCP) [2] or Rate Control Protocol (RCP) [3]. The first group relies on measurements made by the end-host which can lead to inaccurate rate adjustments in design-deviant scenarios. The second group needs support from every node along the path which raises deployment issues.

However, the existing approaches are still end-host driven and flow-based. This means that the end-host has to compute and maintain a congestion window individually for each flow. Synergetic effects, for example, the continuous usage of a congestion window for a new flow after the previous one has completed, are not realized.

We propose to decouple the end-to-end congestion control mechanisms by introducing middleboxes to create dedicated congestion control segments within the network. Within those segments congestion control mechanisms can be deployed that are specifically adapted to the particular properties of this segment.

II. Our Approach

The basic idea of our approach is to create different congestion control segments within a network which allow to deploy specialized congestion control mechanisms. Figure 1 shows such a deployment of two middleboxes between two sites. The middleboxes aggregate the entire traffic between them, creating three congestion control segments: one for each site and one in the middle. Within each segment a dedicated congestion control mechanism can be employed, controlled by the middleboxes. This can be normal TCP congestion control mechanisms, an existing approach like the TCP Friendly Rate Control (TFRC) protocol [1] or any proprietary protocol that is supported by the nodes within a segment. In fact, we introduce a dedicated local congestion control protocol which we employ in local segments. The middleboxes delimit the segments and translate between the different congestion control mechanisms which effectively decouples the segments.

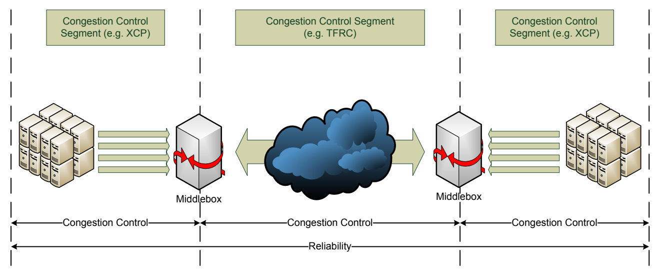

Fig. 1. Creating congestion control segments - Overview

A. End-to-end characteristics

Originally TCP provides end-to-end *reliability* and *congestion control*. In our initial attempts we use a virtual aggregate between the middleboxes which does not introduce additional packet retransmissions. A splitting of TCP flows also does not occur. Therefore, we do not change the principle of end-to-end reliability. However, depending on the congestion control mechanisms employed within the different segments, end-to-end congestion control may no longer be assured.

We didn't notice any negative side-effects due to this, although it is an open issue for future work.

B. Fairness

For congestion control segments that are created between middleboxes fairness can be an issue since the traffic aggregate may compete with other flows. However, we do not consider fairness issues in our work for two reasons. First, the fairness to other flows is dependant on the behavior of the traffic aggregate and, therefore, on the aggregation method that is used. There are a number of traffic aggregation approaches available that have fairness as a goal and can be well used if fairness is important (e.g. PA-MulTCP [5]). The second reason is that fairness is not necessarily an issue in every scenario. For example, a service provider may have an amount of bandwith between two sites that is assured through a service level agreement. In this case, the service provider may expect the middleboxes to utilize this amount of bandwith as aggressively as possible.

C. Dedicated congestion control protocol

We employ a dedicated local congestion control protocol (LCCP) within the local networks. LCCP enables the local middlebox to signal the allowed transmission rate to the end-hosts. It uses a direct feedback channel to communicate the congestion status to a host. Furthermore, it is protocol independent which allows to control protocols that are traditionally agnostic to congestion (e.g. UDP). Since the local congestion control segment is under the administrative authority of the network operator the deployment barriers for new and dedicated congestion control protocols are comparatively low.

III. Evaluation

As our initial scenario we chose two data centers that are connected through the Internet. There are a vast number of flows between the two sites originating from a large number of end hosts. Long-lived and short-lived flows are present and the bandwith usage and number of flows fluctuate erratically. Most of the short-lived flows are assumed to be data queries which are rather time-critical for the data center operations. By introducing middleboxes at the edges of the two datacenters we create three separate congestion control segments, two of which are under the sole control of the local administrative authority. We considered two scenarios for the evaluation of our approach. First, we evaluated the approach using currently available protocols in the local congestion control segments, namely XCP. Second we deployed LCCP which allows much faster and direct feedback to the local hosts.

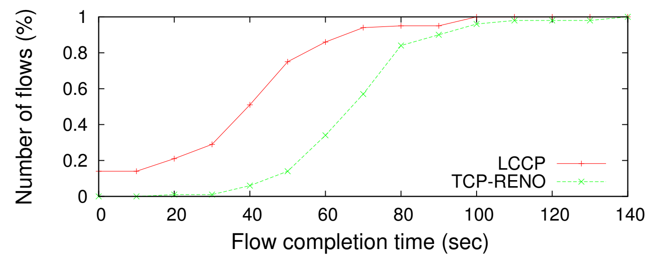

Fig. 2. Comparison of flow completion times

Figure 2 shows the evaluation results for the completion time of short lived flows. The scenario simulates 100 flows which each transports 100 Kbyte of data. The link between the two middleboxes is 1 Mbit/s and the arrival rate of the flows is modelled exponentially with a mean of 1 second. In this scenario, flows using LCCP complete 30% to 40% faster than flows without LCCP and congestion control segments. Since LCCP presets an accurate congestion control value to the local nodes, flows are able to reach their ascertained throughput much faster than with the traditional slow-start and AIMD mechanisms. This also results in more fairness between the local flows because there is no starvation or thwarting of new flows.

Further evaluations will include a scenario where UDP traffic is added to show that LCCP can also effectively control UDP flows. By including UDP traffic control our approach prevents the forwarding of unresponsive, constant bit rate traffic into a congested link. Furthermore, it ensures fairness between UDP and TCP flows.

References

[1] M. Handley, S. Floyd, J. Padhye, and J. Widmer, "TCP Friendly Rate Control (TFRC): Protocol specification, RFC 3448 (Proposed Standard), Jan. 2003

[2] D. Katabi, M. Handley, and C. Rohrs, "Congestion control for high bandwidth-delay product networks," in *SIGCOMM '02: Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications*. New York, NY, USA: ACM, 2002, pp. 89–102.

[3] N. Dukkipati, M Kobayashi, R. Zhang-Shen and N. McKeown, "Processor Sharing Flows in the Internet," in *IWQoS '05: Proceedings of the 13th International Workshop on Quality of Service*. Springer, 2005, pp. 271-285.

[4] J. Crowcroft and P. Oechslin, "Differentiated end-to-end Internet services using a weighted proportional fair sharing TCP," in *ACM SIGCOMM Computer Communication Review archive*, Vol. 28, July 1998, pp. 53–69.

[5] F.-C. Kuo and X. Fu, "Probe-Aided MulTCP: an aggregate congestion control mechanism," in *SIGCOMM Comput. Commun. Rev.*, Vol. 38. New York, NY, USA: ACM, 2008, pp. 17–28.