A New Loss Recovery Algorithm for a Stateless Protocol

Dmitrii Chalyy, Mikhail Nikitinskiy
P.G. Demidov Yaroslavl State University
Yaroslavl, Russia
chaly@uniyar.ac.ru, nikitinskiy87@mail.ru

Abstract

The work consider performance issues of the Trickles protocol [1]. The performance analysis is shown that the protocol behave inefficiently during loss recovery. We propose a modification of the client side algorithm which recover in a more efficient way.

Index Terms: Transport Protocols, Trickles.

In our work we put our attention to loss recovery algorithm of the Trickles protocol [1]. Trickles protocol was proposed in [1] as a transport protocol where the state of the connection is maintained on the one side of the connection. Authors considered that state maintained on the so called client side which is responsible for getting data from the server side which is stateless. It was shown experimentally (see [1]) that Trickles protocol consumes fewer resources and much more resistant for Denial-of-Service attacks. Stateless server side enables different kinds of networking services and is perspective for use in a cloud environement.

We consider performance of the Trickles protocol. Our analysis and experiments in ns-2 simulator shown that Trickles behave inefficiently when data loss occur. In [1] authors propose a loss recovery scheme that may recover in 2RTT (Round-Trip Time) time. In a contrast standard TCP algorithms can recover in RTT time. Our approach can reduce loss recovery time by the stateless protocol to RTT by modifying the client side. The idea of modification is to give the client side the ability to make an early request for data which is considered to be lost. The algorithm consists of the following steps:

1. If client receives three out-of-order packes then the client goes to loss recovery phase and sends a request for the data considered to be lost. This allows us to request the lost data earlier than in the original Trickles protocol.
2. During the loss recovery the number of packets in network is halved.
3. If the client receives the lost data then it ends loss recovery and goes back to normal operation.

Such an approach allot functionality to a client side which is located on server side of the original Trickles protocol. Since the client doesn't make decisions to increase number of requests the proposed protocol doesn't behave more aggressively in competing for network resource.

Acknowledgment

The work was supported by RFBR Grants No 12-01-00281-a and 11-07-00549-a.

References

[1] Shieh, A., Myers, A. C., and Sirer, E. G., "A stateless approach to connection- oriented protocols", *ACM Trans. Comput. Syst*. 26, 3, Article 8 (September 2008), 50 pages.