Robust UDP-based Byzantine Fault Tolerant Consensus

State-machine-replication (SMR) is used to build fault tolerant systems such as airplanes, cars, and industrial control systems. SMR consists of multiple machines (replicas), which agree on a common value even in case of faulty behaviour of single machines. In SMR, there are two main fault models, namely crash fault tolerance (CFT) and byzantine fault tolerance (BFT). For CFT systems, a replica runs as specified or it crashes. In BFT, a replica may behave arbitrarily (drop/send wrong values/messages).

In this thesis, we work on the BFT consensus protocol HotStuff [1]. The original authors provide a freely available C/C++ implementation of the protocol. In previous work, we augmented this implementation to use UDP communication. In this thesis we will analyze the impact of custom retransmissions based on RTT estimations between communication partners. First we will explore the basic structure of the consensus protocol and its implementation. Afterwards you will implement RTT estimations between parties and custom retransmission. Finally you will analyze the performance of the resulting system.

Your Tasks

- Familiarize yourself with the HotStuff protocol and implementation
- Identify and implement the necessary code changes
- Measure the performance impact of your changes
- Evaluate the results

References


Prerequisites

- Experience with Linux-based operating systems and networking
- Experience with C/C++ programming

Contact

Richard von Seck    seck@net.in.tum.de
Johannes Schleger   schleger@net.in.tum.de