TRTP 2 - Return of the Protocol

LINGI1341

Group #138	d'Herbais de Thun Sébastien	28751600
	Heuschling Thomas	28871600

1 Abstract

We created a high performance multithreaded receiver implementation that can saturate a 1GbE connection and greatly utilize a 10GbE connection while providing reliable and repeatable data transfers. In addition, we used advanced Linux syscalls that help reduce their numbers to a minimum, improving performance and reducing CPU usage. It is also documented in great details explaining as the code flows the architectural decisions and algorithms used to make the application work. This means it would be easily modified by another team for any future modifications. Finally, the code is accompanied by a full suite of tests for almost every function in the codebase to ensure that compilation equates to reliable data transmissions.

Contents

2 Introduction

The goal of this project was to implement a receiver based on the TRTP protocol capable of handling multiple clients at the same time. To try and have a more interesting and more in depth analysis we decided to realise an optionally multithreaded version. This means it can use multiple threads if desired or run on a single one. In the first mode we can reach with limited software/hardware tinkering speeds exceeding 2Gb/s and in single threaded mode speeds exceeding 1Gb/s easily saturating a typical 1GbE card. In addition, our implementation can treat hundreds of concurrent connections with an excellent success rate of over 99%[1]. This means our implementation fulfills the requirements and more by providing a high performance and reliable implementation.
In this document the architectural decisions and the techniques employed will be explained followed by in-depth performance analysis in different conditions including stress tests and link simulation. The end goal is to understand why a multithreaded implementation is interesting and why the specifics of our implementation give great performance with reasonable CPU usage. It will also try to explain the hard limit seen at 2.1Gb/s during testing.

[1]Note that all of the failures are due to sender unable to cope with the data rate