

Transparent TCP to SCTP translation shim layer

Daniel Manns

TU Chemnitz

Chair for Communication Networks
Chemnitz, Germany

Daniel.Manns@etit.tu-chemnitz.de

Thomas M. Knoll

TU Chemnitz

Chair for Communication Networks
Chemnitz, Germany

knoll@etit.tu-chemnitz.de

I. INTRODUCTION

A. Motivation

The multi-path communication “Stream Control Transmission Protocol (SCTP)” [1] gains huge attention in the research community, as it has the potential to supplement or even replace the TCP protocol. However, there are currently only a few applications with integrated SCTP support available. The reason for this is that every application would need to be rewritten in order to take advantage of the features of SCTP. In this contribution we outline a method for transparent TCP to SCTP translation, which enables unmodified TCP-based applications to fully exploit the benefits of SCTP.

As outlined in [2], the goal of a transparent TCP-to-SCTP-translation shim consists of an augmentation of TCP-based applications to make use of multipath transport capabilities and other features of an underlying transparent SCTP association. Transparent in this scope means, the TCP-based applications will not notice the SCTP wrap-up at all.

As a result, the following scenarios using SCTP translation will be supported:

- TCP client and TCP server,
- TCP client and SCTP server,
- SCTP client and TCP server.

This functionality could foster adoption of SCTP as new standard communication protocol. Furthermore, the broad usage on fixed and mobile devices (i.e. Android based) becomes possible.

B. SCTP Features

Upon the design of SCTP, some shortcomings of TCP have been eradicated. The biggest advantage is an enhanced resilience against resource-based attacks (i.e. DoS) as well as the support for multi-homed clients. Other advantages include multi-streaming within a SCTP association, as well as optimized ACK processing.

II. CURRENT STATE & FUTURE WORK

A. Approach

To achieve the desired behaviour, a SCTP socket will be generated and hid inside a newly instantiated TCP socket.

The hidden SCTP socket is passed down to the transport-layer to establish the communication. However, a reference to the initial TCP socket will be held, as applications are not

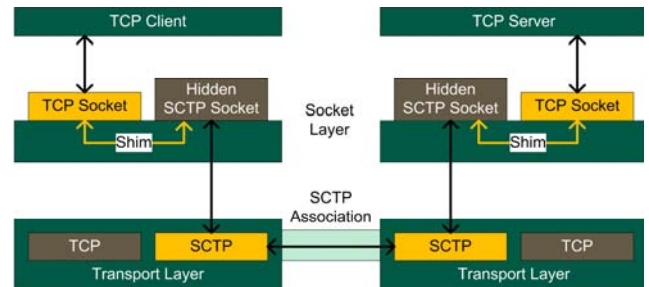


Figure 1. Architecture

aware of the translation process by the shim-layer. In case a SCTP connection is not possible, a fall-back standard TCP connection is provided.

B. Current State

At present, only an implementation for the outdated FreeBSD 4.0 is available. As it is not suitable for on-going research to work with 4.0, a migration to the latest stable FreeBSD version (8.2) has been performed. The verification of the functionality was done through different application scenarios – namely web browsing and content-streaming applications.

C. Use Cases

SCTP augmented TCP can be applied to a broad range of use cases. All communication scenarios, which increase throughput and/or stability by means of SCTP transport across multiple interfaces, will gain from this shim layer technology.

This is in particular true for mobile devices with several communication interfaces. The breakdown of one connection could automatically be repaired by means of the SCTP path failover mechanism. Thus, TCP applications will not even notice the swap between communication paths.

D. Future Work

Once the functionality is verified, the mechanism will be ported to a current Linux-Kernel and afterwards tested on various Android based devices.

III. REFERENCES

- [1] R. Stewart: Stream Control Protocol (SCTP). RFC 4960. IETF. September 2007. URL: <http://tools.ietf.org/html/rfc4960>
- [2] R. Bickhart, P. Amer, R. Stewart. TCP-to-SCTP translation shim layer in the FreeBSD kernel, Euro BSD Con 2007, Copenhagen, May 2007