Contribution ID: **151** Type: **not specified**

Linux SCTP is Catching Up and Going Above!

Tuesday, 13 November 2018 17:15 (35 minutes)

SCTP is a transport protocol, like TCP and UDP, originating from SIGTRAN IETF Working Group in the early 2000's with the initial objective of supporting the transport of PSTN signalling over IP networks. It featured multi-homing and multi-stream from the beginning, and since then there have been a number of improvements that help it serve other purposes too, such as support for Partial Reliability and Stream Scheduling.

Linux SCTP arrived late and was stuck. It wasn't as up to date as the released RFCs, while it was also far behind other systems such as BSD, and also suffered from performance problems. In the past 2 years, we were dedicated to ensuring that these features were addressed and focused on making many improvements. Now all the features from released RFCs have been fully supported in Linux, and some from draft RFCs are already ongoing. Besides, we've seen an obvious improvement in performance in various scenarios.

In this talk we will first do a quick review on SCTP basics, including:

- Background: Why SCTP is used for PSTN Signalling Transport, why other applications are using or will use SCTP.
- Architecture: The general SCTP structures and procedures implemented in Linux kernel.
- VS TCP/UDP: An overview of functions and applicability of SCTP, TCP and UDP.

Then go through the improvements that were made in the past 2 years, including:

- SCTP-related projects in Linux: Other than kernel part, there are also lksctp-tools, sctp-tests, tahi-sctp, etc.
- Features implemented lately: RFC ones like Stream Scheduling, Message Interleaving, Stream Reconfig, Partially Reliable Policy, and many CMSGs, SndInfos, Socket APIs.
- Improvements made recently: Big patchsets like SCTP Offload, Transport Hashtable, SCTP Diag and Full SELinux support.
- VS BSD: We will take a look at the difference between Linux and BSD now regarding SCTP. You will be surprised to see that we've gone further than other systems.

We will finish by reviewing a list of what is on our radar as well as next steps, like:

- Ongoing features: SCTP NAT and SCTP CMT, two big important features are ongoing and already taking form, and more Performance Improvements in kernel have also been started.
- Code refactor: New Congestion Framework will be introduced, which will be more flexible for SCTP to extend more Congestion Algorithms.
- Hardware support: HW CRC Checksum and GSO will definitely make performance better, for which a new segment logic for both .segment and HW that works for SCTP chunks is needed.

- RFC docs improvements: We believe that more extensions and revisions will make SCTP more widespread.

For its powerfulness and complexity, SCTP is destined to face many challenges and threats, but we believe that we have already and will continue to make it better than that on other systems, but also than other transport protocols. Please join us, Linux SCTP needs your help too!

Presenters: LEITNER, Marcelo RIcardo (Red Hat); LONG, Xin (Red Hat)

Session Classification: Networking Track