ULE Decapsulation in Linux

Status of implementation and future developments

67th IETF Meeting - San Diego, CA, USA

Bernhard Collini-Nocker <bnocker@cosy.sbg.ac.at>

Christian Praehauser <cpraehaus@cosy.sbg.ac.at>

Department of Computer Sciences,
University of Salzburg

Current Status

- Updates for RFC 4326 compliance in Linux 2.6.18
  - Corrected check of SNDU length
  - Fixed handling of optional extension headers
  - Added handling of extension header padding
  - Fixed check of destination address in presence of extension headers
  - Fixed/Improved filtering on destination address

Current Limitations

- Currently, everything that has to do with DVB networking in Linux is located in one file, dvb_net.c, which is a part of the dvb-core module. This file is rather involved and hard to read.
- People suggested to “beautify” that code ;-)
  - *“This driver would have to be one of the ugliest-looking things in the kernel. You must have a strong stomach”* (Andrew Morton)
- Some interesting ULE Extensions are at the horizon (e.g. ULE security, header compression). Supporting one in Linux would mean to “statically” hack it into dvb_net.c

Future Development

- Thus we started with re-designing the DVB networking stuff in Linux.
- Encapsulation methods (MPE, ULE) are abstracted behind a common interface and designed as loadable modules.
  - New encapsulations can be added without changing existing code, which might be interesting for GSE.
- Support for ULE Extensions is also provided by modules
  - However, extensions that are defined in the ULE standard are directly implemented in the ULE decaps module (ULE decaps core).

Future Development

- If the ULE decaps core receives an SNDU with an unknown extension type it will search for a module which implements the required extension
  - If none is found, drop the SNDU and print a meaningful message to the kernel log
- Automatic search for ULE Extension modules can be turned off (via module parameter)

Future Development

- We also envisage a proxy module which allows extension modules in userspace.
  - especially interesting for the development of new extensions, proprietary extensions (remove dependency on kernel versions) and extensions which do very complicated things.
  - In userspace, extension implementers have all libraries, networking, etc. at their disposal.
- Of course, the performance of userspace XTs will be modest.
- ADVANTAGE: portability; The module may be used e.g. by ulenet for Windows.

Future Development

- The design is at a very early stage. A first version (in form of a Kernel patch) should become available during Q1 2007
- Everybody is welcome to contribute. Please contact us if interest exists
- Authors of ULE Extensions are encouraged to express their opinion about implementation requirements , i.e. what information and which facilities would be required/useful for an extension at the decaps side.