Modular Link Layer Functions of a Generic Protocol Stack for Future Wireless Systems

L. Berlemann, A. Cassaigne, R. Pabst and B. Walke

Chair of Communication Networks (ComNets)
RWTH Aachen University, Germany

2004 Software Defined Radio Technical Conference and Product Exposition,
November 17th 2004, Phoenix, USA

Overview

- Introduction and Motivation
- Generic Protocol Stack in the Context of Multi-Mode Capable Wireless Networks
- Modular Approach
- Realization of Protocol Layers
- Conclusion and Outlook

Introduction and Motivation

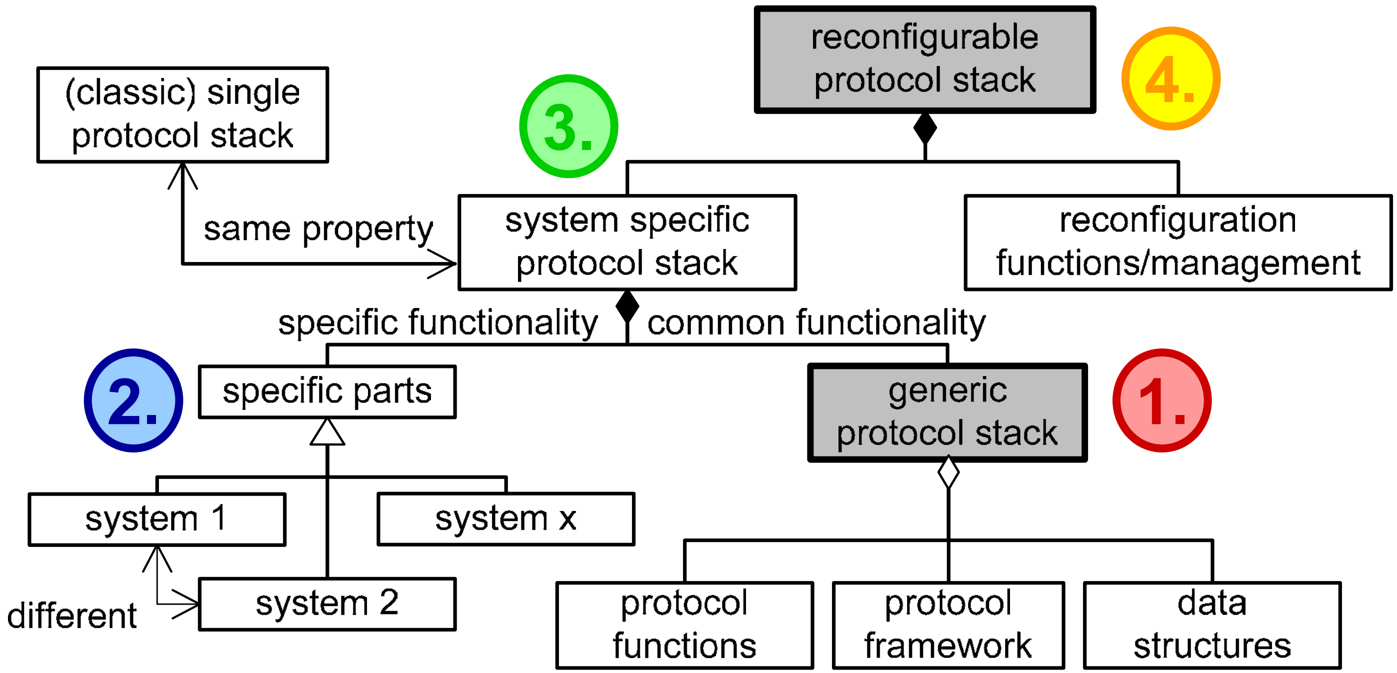
- Idea: Protocols share a lot of communalities, that can be exploited in an efficient reconfigurable wireless system

  → **Generic Protocol Stack**

- Advantages: runtime reconfigurability, maintainability, code/resource sharing and accelerated protocol development through reusability
- Generic part is crucial: Tradeoff - general usability vs. implementation effort
- Two approaches for realization, depending on the abstraction level of identified similarities:
  - Parameterizable modules including fundamental protocol functions
  - Inheritance of generic part(s) [1]

[1] M. Siebert, B. Walke, "Design of Generic and Adaptive Protocol Software (DGAPS)," in Proc. of 3Gwireless '01, San Francisco USA, June 2001

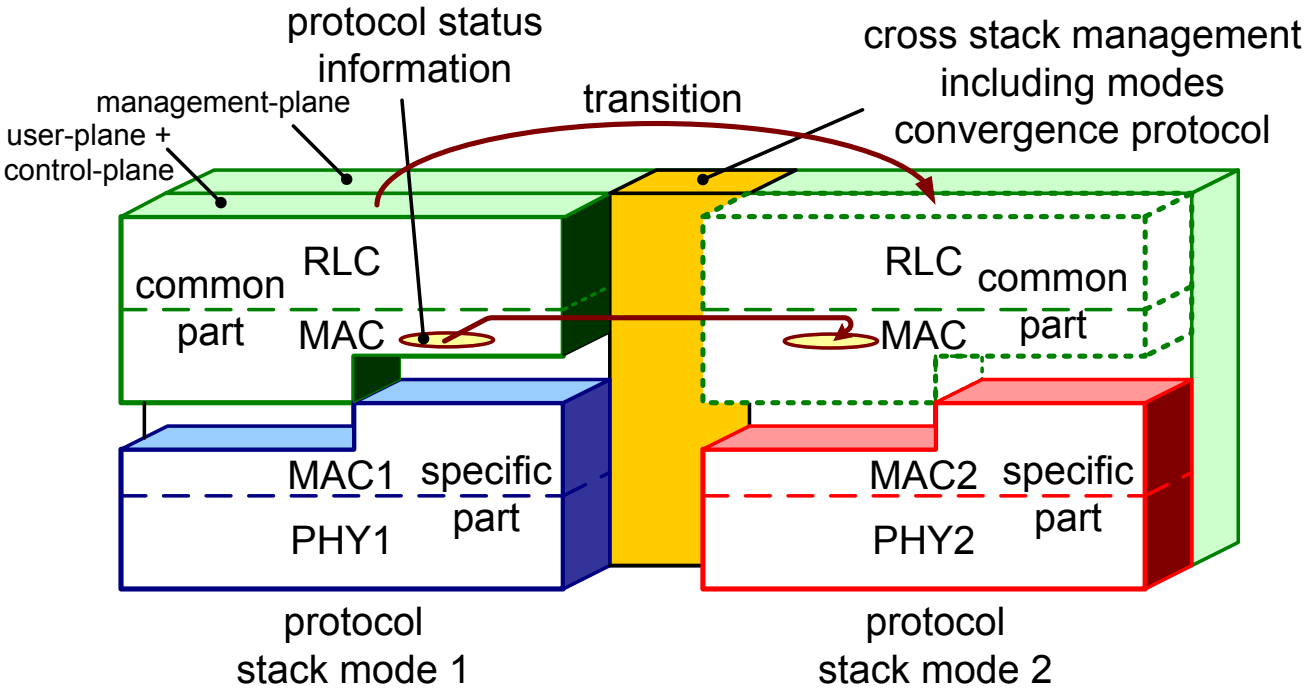
Generic Protocol Stack



- ... (step 4) ... (step 1) ... protocol layers ... (step 3) ... (step 2)
  - ... system
  - ... a flexible as well as efficient realization of a reconfigurable protocol stack
  - ... software design
  - ... of the individual behavior of a system

Enabling Transition between Multiple Modes

… with the help of a **Modes Convergence Protocol**.



- Separation into specific/generic part in each layer
- Management of (parallel existing) protocol stack/layer
- Administration of user data (seamless mode transition)
- Cross layer optimization (preservation of protocol status info)
- Support of network-initiated reconfiguration

Generic Protocol Functions of the Data Link Layer

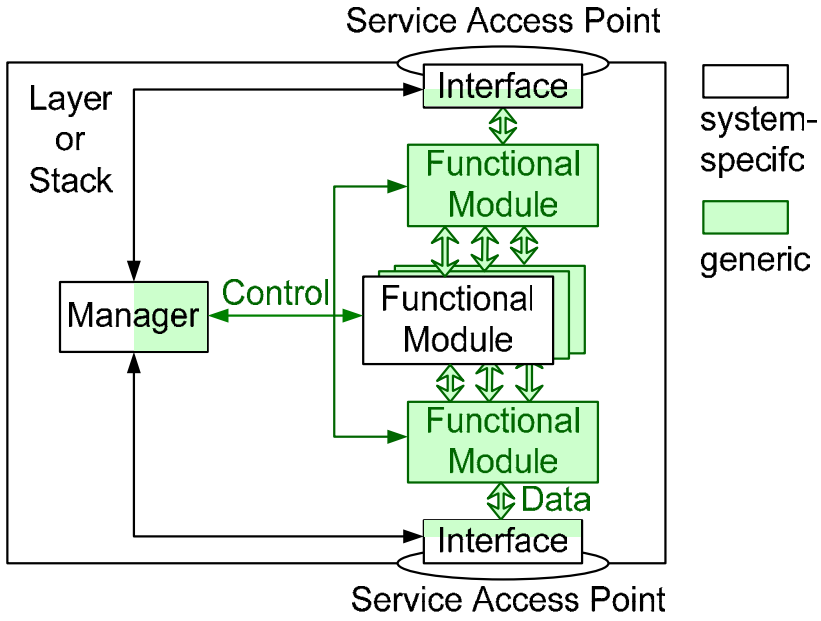
- Modern communication protocols cannot be forced into classical layered architecture of ISO/OSI RM
- Though belonging to Data Link Layer, common fundamental protocol functions can be found in multiple layers (2-4):
  - Error handling - Forward Error Correction or Automatic Repeated reQuest protocols [2]
  - Flow control
  - Segmentation, concatenation and padding of PDUs*
  - Multiplexing and De-Multiplexing
  - Dynamic Scheduling
  - Ciphering
  - Header Compression

[2] L. Berlemann, A. Cassaigne and B. Walke, "Generic Protocol Functions for Design and Simulative Performance Evaluation of the Link-Layer for Reconfigurable Wireless Systems," in Proc. of *WPMC'04*, Abano Terme Italy, September 2004

Modular Approach

- Common protocol functions as parameterizable modules and system-specific modules form a complete protocol layer
- Communication inside: generic service primitives and generic PDUs

- *Functional Module*: Realizes fundamental functionality as black box
- *Manager*: Composition, rearrangement, parameterization and data query of modules; Administration of internal communication
- *Interface*: Translation of generic service primitives to specific ones
- *Service Access Point*: Is needed, if a classical layer is demanded for fitting into ordinary stack



→ Simulation and performance evaluation on several levels: (sub-)layer as well as complete protocol stack

Parameterization of Functional Modules

Parameterization implies:

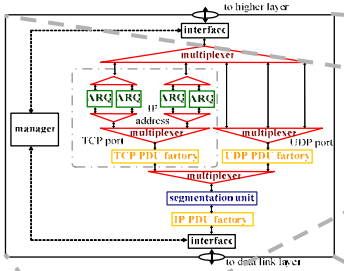
- Specification of a value
- Switching on/off of a behavior or functionality
- Extension of the modules' interface

Segmentation module as example:

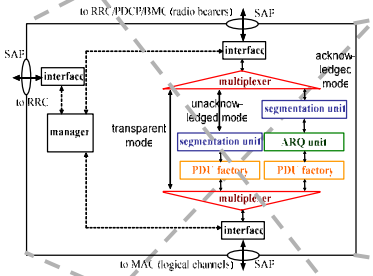
- Use of concatenation
- Targeted PDU size after handling
- Use of Padding, i.e. filling up of a PDU to reach a certain size
- Transmitter/Receiver Role
- Buffer size for SDUs concatenated in a single PDU
- Behavior in case of an error, i.e. interworking with ARQ module

Exemplary Protocol Layers

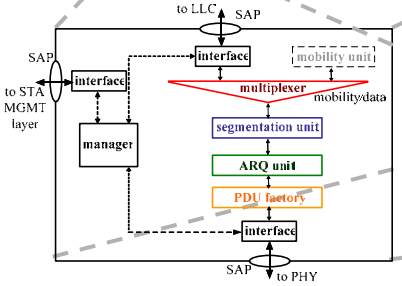
TCP/UDP/IP

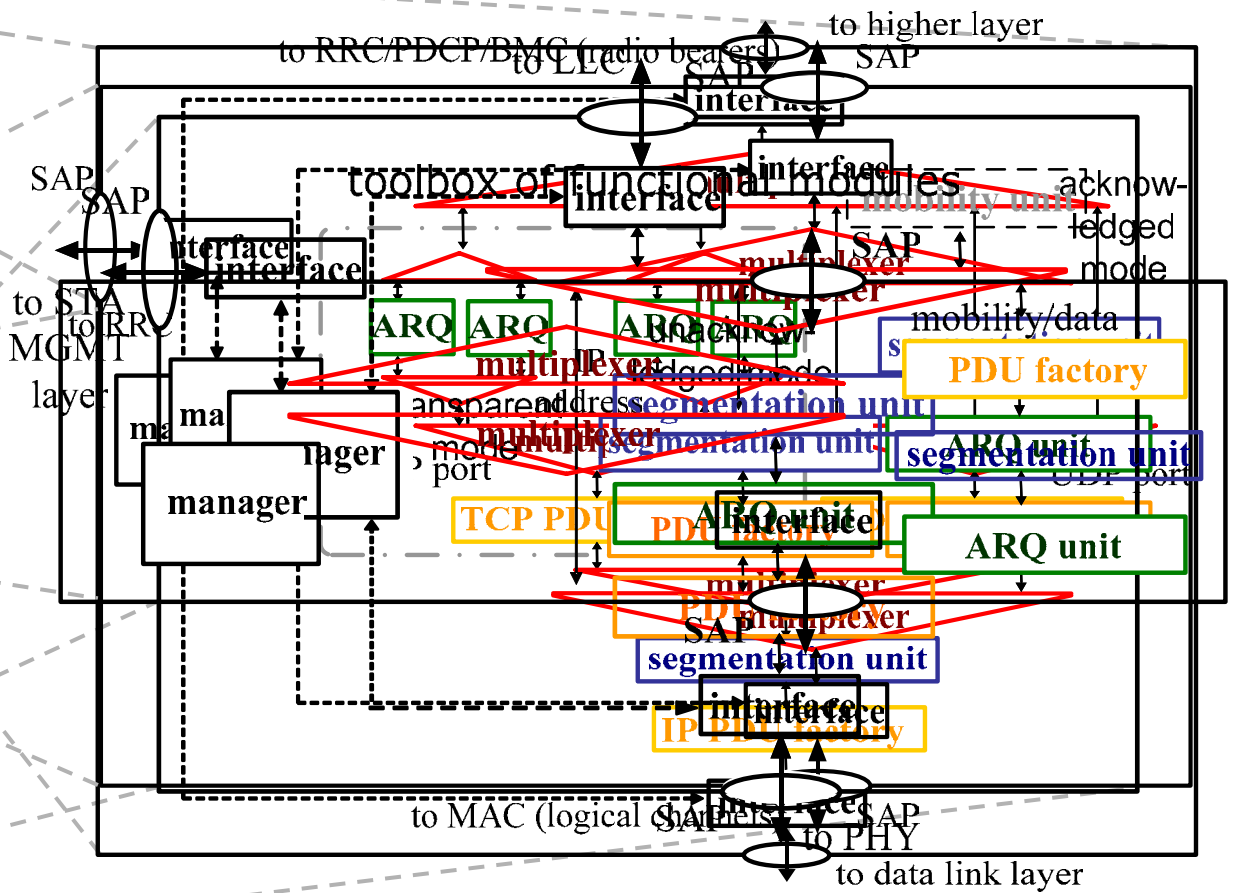


UMTS RLC



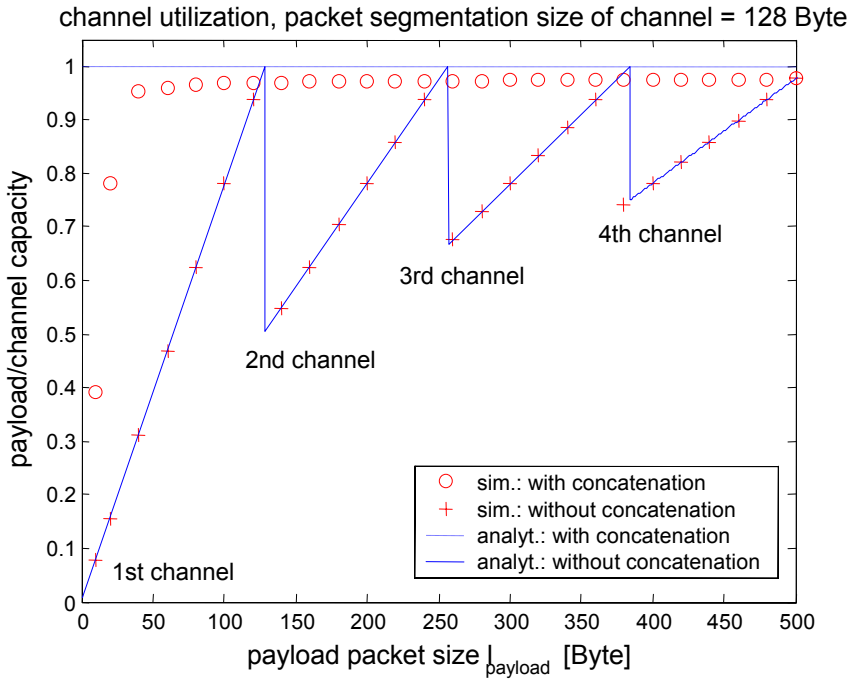
802.11 MAC





Segmentation Module as Example

SDU is separated into PDUs to fit to transport channel
Here: Segmentation aspects of UMTS RLC in Unacknowledged Mode



payload / channel capacity

protocol overhead without concatenation

protocol overhead with concatenation

$$\frac{payload}{channel\ capacity} = \frac{l_{payload}}{\left\lceil \frac{l_{payload}}{packet\ size} \right\rceil \cdot packet\ size}$$

packet size of payload is increased

- Protocol overhead vs. optimized channel utilization
- The segmentation module reflects the known behavior → it can be legitimately used in an multi-mode capable protocol stack

Conclusion and Outlook

- The identified similarities are decisive for success
  → tradeoff of genericity
- Generic protocol stack takes up well-proven and known fundamental protocol functions
- Existing (as shown) and future protocols (4G) can be composed out of adequately parameterized modules
- Library of common functions results in a construction kit for accelerated protocol development
- Efficient protocol reconfigurability through parameterization is enabled on the basis of functional modules

The introduced approach is a first step to an efficient multi-mode capable wireless system

Thank you for your attention !

Lars Berlemann
ber@comnets.rwth-aachen.de

EW'05

11th European Wireless Conference

"Next Generation Wireless and Mobile Communications and Services"

Nicosia, Cyprus
April 10 - 13, 2005
http://www.european-wireless2005.de

PIMRC 2005

The 16th Annual IEEE International Symposium on Personal Indoor and Mobile Radio Communications

Maritim Hotel, Berlin, Germany
September 11 – 14, 2005
http://www.pimrc2005.de