Generic Protocol Software for Reconfigurable Systems

L. Berlemann, M. Siebert and B. Walke

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

PIMRC'04 – E2R Workshop
September 5th 2004, Barcelona Spain

Overview

- Introduction and Motivation
- Generic Protocol Stack in the Context of Reconfigurability
- Modular Approach
- Realization of Protocol Layers
- Functional Validation of ARQ-Module as Part of IEEE 802.11 MAC
- Conclusion and Outlook

Introduction and Motivation

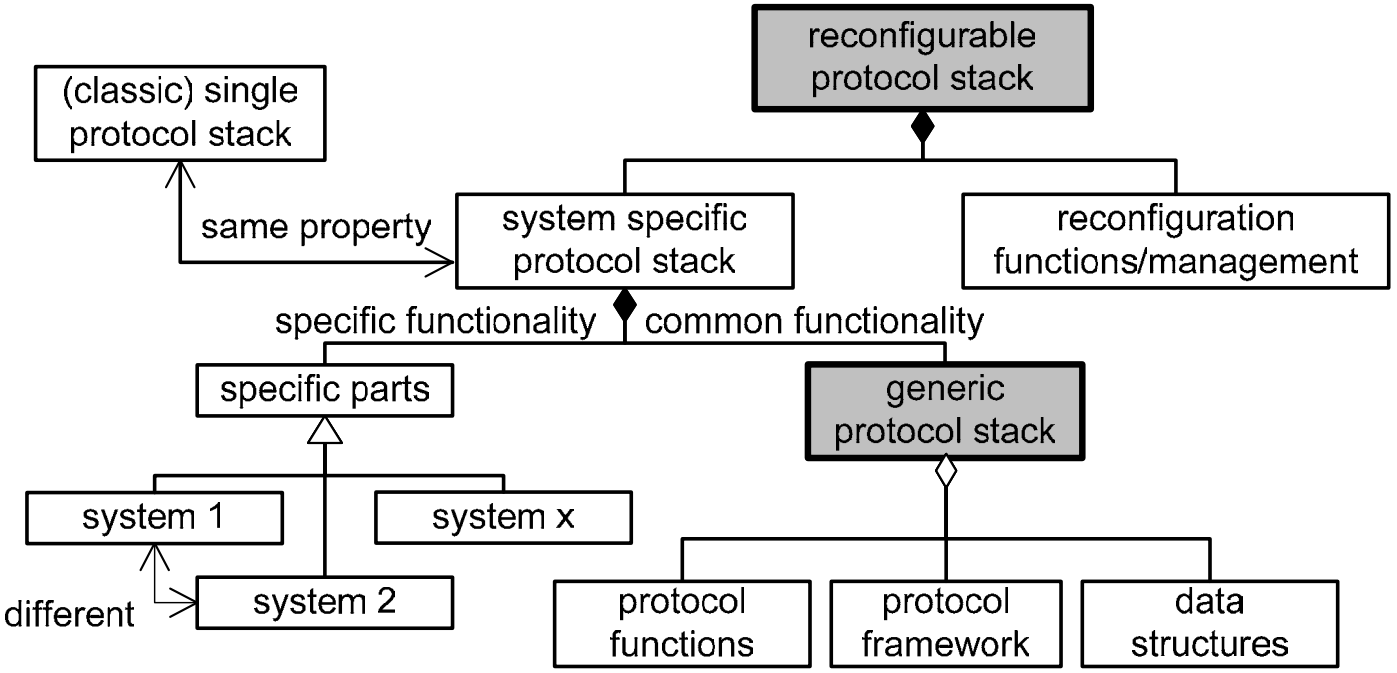
- Idea: Protocols share a lot of communalities, that can be exploited in an efficient reconfigurable wireless system

  → 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

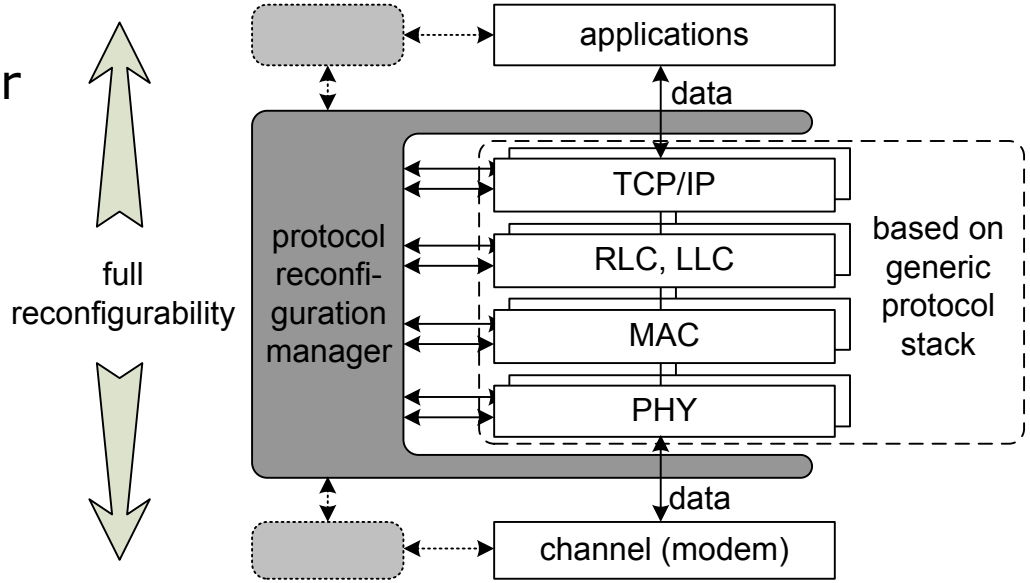


- Generic Protocol Stack
  - provides characteristics for all systems
  - will serve as a basis for future software design (reusability)

Enabling Reconfiguration

… with the help of a **Protocol Reconfiguration Manager**.

- Management of (parallel existing) protocol stack/layer
- Creation + Destruction + Reconfiguration
- Administration of user data
- Cross layer optimization
- Support of reconfiguration functions of the network



Characteristics of Protocol Reconfiguration:

(I.) Creation of new stack/layer based on adequately parameterized modules

(II.) Reconfiguration of existing protocol in exchanging parameterization of corresponding modules

Generic Protocol Functions of the Data Link Layer

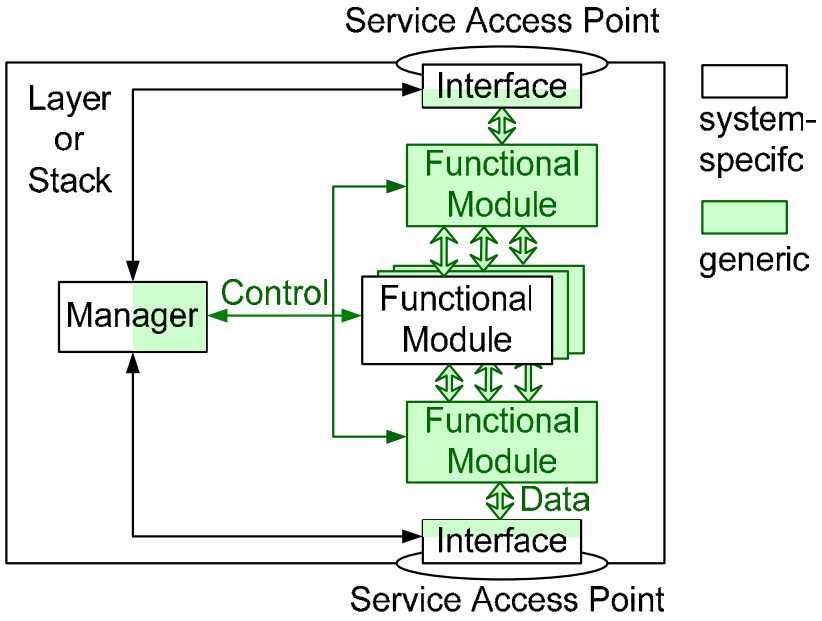
- Modern communication protocols cannot be forced into classical layered architecture of ISO/OSI RM
- Though belonging to Data Link Layer, common fundamental protocol functions can be found in multiple layers (2-4):
  - Error handling - Forward Error Correction or Automatic Repeated reQuest protocols*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

[2] L. Berlemann, A. Cassaigne and B. Walke , "Modular Link Layer Functions of a Generic Protocol Stack for Future Wireless Networks ," to appear in Proc. of *SDRforum04*, Phoenix USA, November 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:

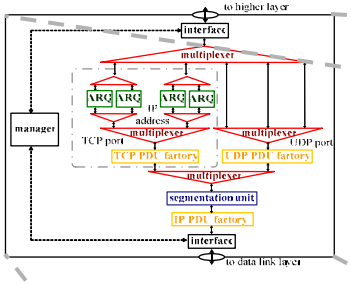
- Specification of a value
- Switching on/off of a behavior or functionality
- Extension of the modules' interface

ARQ module as example:

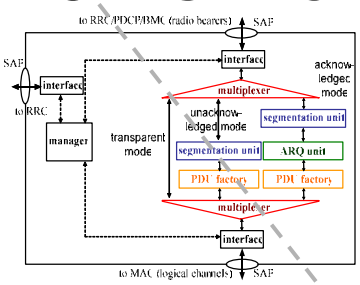
- ARQ characteristics, Go-Back-N or Selective-Reject
- Transmitter and/or receiver role
- Transmission window - size, fixed or variable
- Timer value, after packet is assumed to be lost
- Use of Negative ACKnowledgments
- Connection service: inexistent (UMTS RLC), separated for each direction (802.11 – CSMA/CA with RTS/CTS), 2-way handshake (GSM LLC) or 3-way handshake (TCP)

Exemplary Protocol Layers

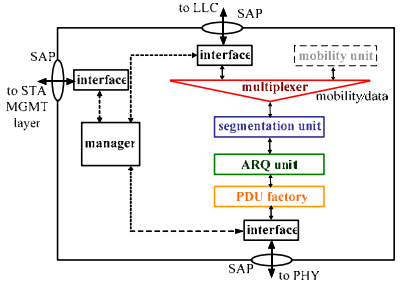
TCP/UDP/IP

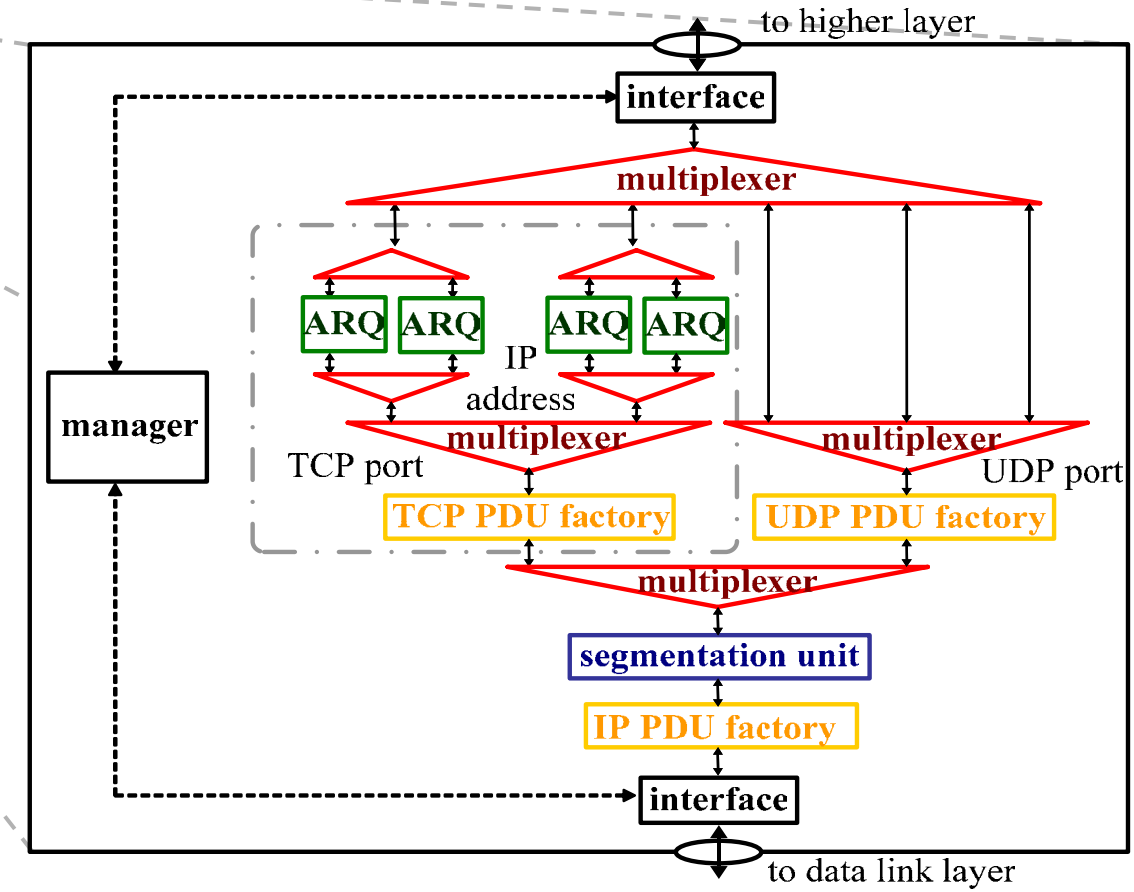


UMTS RLC



802.11 MAC

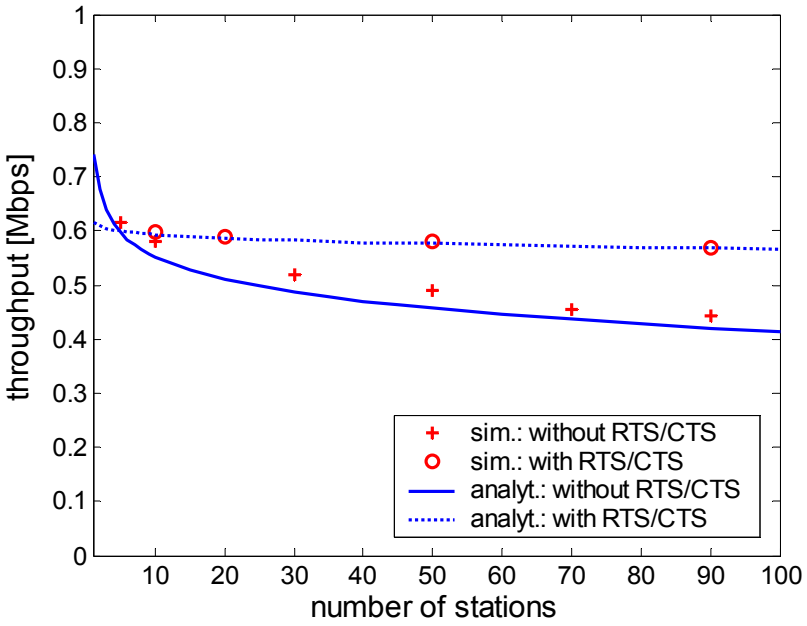




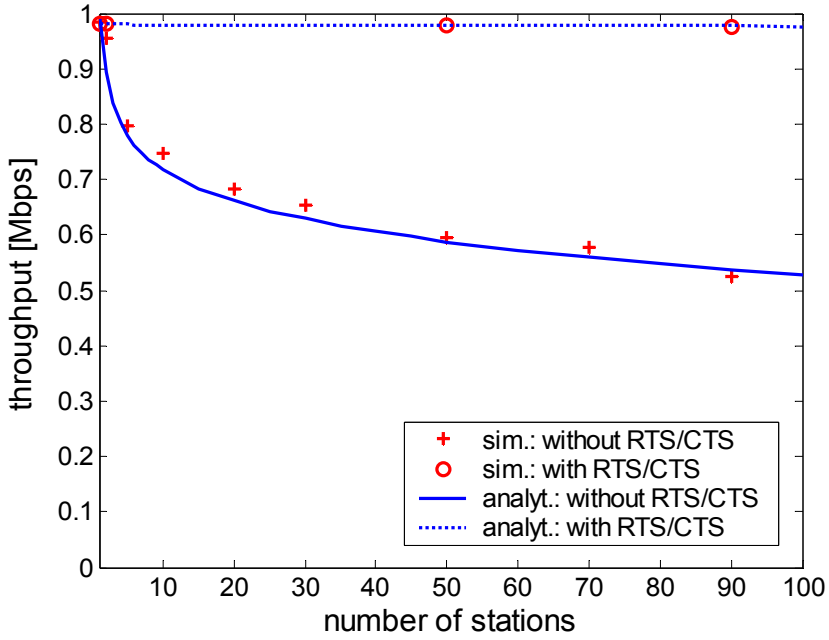
Validation of ARQ module - IEEE 802.11 MAC

Distributed Coordination Function (DCF) of IEEE 802.11 regarded as Send-and-Wait ARQ (= Go-Back-N ARQ with window size 1)

packet size = 128 byte



packet size = 4095 Byte



- RTS/CTS for throughput optimization in case of collisions
- The ARQ module reflects the known behavior of the RTS/CTS mechanism → it can be legitimately used in an IEEE 802.11 MAC

Conclusion and Outlook

- The identified similarities are decisive for success
  → tradeoff of genericity
- Generic protocol stack takes up well-proven and known protocol functions
- Library of common functions results in a construction kit for protocol development
- Efficient protocol reconfigurability through parameterization is enabled on the basis of functional modules
- A limited additional effort for protocol management is clearly identified

The introduced approach is a first step to an efficient end-to-end reconfigurable wireless system

Thank you for your attention !

Lars Berlemann
ber@comnets.rwth-aachen.de