Paper published at

EW 2005
11th European Wireless Conference 2005

10-13 April 2005, Nicosia, Cyprus

Title:

Reconfigurable Multi-Mode Protocol Reference Model Facilitating Modes Convergence

Abstract: The vision for the next generation of wireless communications is a ubiquitous radio system concept, providing wireless access from short-range to wide-area, with one single adaptive system for all envisaged radio environments. To achieve this, a future radio system is envisaged to efficiently adapt to multiple scenarios by using different modes of a common technology basis. By introducing a reference model for multi-mode protocols, we target a harmonization of the various current research efforts in approaching multi-mode capability and protocol adaptability. Based on the idea of identification of commonalities between multiple envisaged modes and a separation of the common and the mode-specific parts of the protocol, this paper defines a reference model for a multi-mode protocol incorporating the ideas of functional partitioning and re-configuration of layers to achieve both horizontal and vertical convergence between multiple modes.

Keywords: Generic Protocol Stack, Modes Convergence Manager, Multi-Mode Protocol, Reconfigurablity, Reference Model.

Authors: Lars Berlemann, Ralf Pabst, Marc Schinnenburg, Bernhard Walke

Affiliations: Chair of Communication Networks, RWTH Aachen University

Addresses: Kopernikusstraße 16, D-52074 Aachen, Germany

Telephone: +49 241 80 27248

E-Mail: {ber|pab|msg|walke}@comnets.rwth-aachen.de

Reconfigurable Multi-Mode Protocol Reference Model Facilitating Modes Convergence

Lars Berlemann, Ralf Pabst, Marc Schinnenburg, Bernhard Walke
Chair of Communication Networks, RWTH Aachen University, Germany
{ber|pab|msg|walke}@comnets.rwth-aachen.de

Abstract — **The vision for the next generation of wireless communications is a ubiquitous radio system concept, providing wireless access from short-range to wide-area, with one single adaptive system for all envisaged radio environments. To achieve this, a future radio system is envisaged to efficiently adapt to multiple scenarios by using different modes of a common technology basis. By introducing a reference model for multi-mode protocols, we target a harmonization of the various current research efforts in approaching multi-mode capability and protocol adaptability. Based on the idea of identification of commonalities between multiple envisaged modes and a separation of the common and the mode-specific parts of the protocol, this paper defines a reference model for a multi-mode protocol incorporating the ideas of functional partitioning and re-configuration of layers to achieve both horizontal and vertical convergence between multiple modes.**

Keywords — **Generic Protocol Stack, Modes Convergence Manager, Multi-Mode Protocol, Reconfigurablity, Reference Model**.

I. INTRODUCTION

This paper introduces a reference model for a multi-mode protocol stack of a flexible, dynamic reconfigurable air-interface for future wireless networks. This future wireless network has the vision of a ubiquitous radio system concept providing wireless access from short-range to wide-area, with one single adaptive system for all envisaged radio environments. It will efficiently adapt to multiple scenarios by using different modes of a common technology basis. We target at a harmonization of the various current research efforts in approaching multi-mode capability and protocol reconfigurability in the integrated projects Wireless World Initiative New Radio (WINNER) and Ambient Networks (AN) of the 6th framework research funding program of the European Union originating from the Wireless World Research Initiative (WWI) [1]. Relay-based wireless mobile broadband systems [2], providing a patchy coverage in densely populated urban areas, play an important role in these projects. In the following, the term "multi-mode" may be equivalently substituted by "multi-standard", "multi-protocol" or "multi-system" and "layer" implicates a "sub-layer".

II. COMMON ANCESTORS – COMMON FUNCTIONALITIES

Technology shows the tendency to develop in an evolutionary fashion. The result of this evolutionary process is that technologies with "common ancestors" tend to exhibit a high degree of similarity and commonalities. As example, we take the High Level Data Link Control Protocol (HDLC), standardized in the year 1978 [3]. HDLC is a bit-oriented synchronous Data Link Layer (DLL) protocol which forms the basis for a large number of protocols, both in the wired and the wireless worlds as depicted in Fig. 1. All these protocols use a standard frame structure for transferring user-data and channel related control information for the signalling on the DLL and they provide them in a very similar fashion [4]-[6].

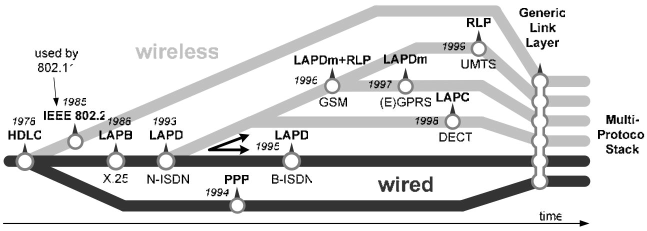


Fig. 1: Concrete example for "generic": High Level Data Link Control Protocol as common origin of multiple protocols used for signalling in the data link layer.

III. WHAT IS GENERIC?

A key issue of the introduced reference model for multi-mode protocols is the separation of a layer into specific and generic parts. The term generic is used by multiple authors with different knowledge backgrounds leading to dissimilar or even contradictory understandings of genericity. In taking the realization of a protocol stack out of generic and specific parts into account this becomes a software engineering problem of generic programming. Generic programming can be defined as

"*programming with concepts, where a concept is defined as a family of abstractions that are all related by a common set of requirements. A large part of the activity of generic programming, particularly in the design of generic software components, consists of concept development - identifying sets of requirements that are general enough to be met by a large family of abstractions but still restrictive enough that programs can be written that work efficiently*" [7]

The balancing of the trade-off between general usability and implementation effort is crucial for the success of the separation of complex protocol software into generic and specific parts.

In general, generic protocol software may be realized through parameterizable modules and/or inheritance of system specific behaviour. Well known programming patterns from computer science [8] provide thereby a suitability-proven fundamental approach to the efficient realization of reconfigurable multi-mode protocol software.
In the context of communication protocols generic parts can be identified on different levels:

- Architecture and composition of a protocol stack, introduced in Section VI
- Functions fulfilled by a layer that imply a certain behaviour, outlined in Section VII
- Data structures, i.e., protocol data units, used for communication between peer-entities of a layer
- Protocol framework: common rules for communication, as for instance the structure of a Medium Access Control (MAC)-frame (sequence and duration of broadcast, downlink and uplink phase)
- Management of a layer and protocol stack

The term "generic" can be substituted in the following by "common" and "general".

IV. Idea of a Generic Protocol Stack

The rationale for approaching a generic protocol stack [9]-[11] is that all communication protocols share much functional commonality in various layers, which can be exploited to build an efficient multi-mode wireless communication system. The aim is to gather these common parts in a single generic stack, i.e., toolbox of generic protocol functions. These common functions are adapted following particular requirements of the targeted mode, as illustrated in Fig. 2. The targeted advantages of this concept are: runtime reconfigurability, maintainability, code/resource sharing and protocol development acceleration through reusability.
The initial step towards a generic protocol stack is a detailed, layer-by-layer analysis of communication protocols to identify their similarities. Their elaborated realization of the generic parts is crucial for the success of the proposed concept as introduced above. As depicted in Fig. 2, the generic protocol stack comprises the above introduced fundamental protocol functions, data structures, protocol architecture, framework and management. They form, together with mode specific parts, a system specific protocol stack. An efficient multi-mode capable stack is

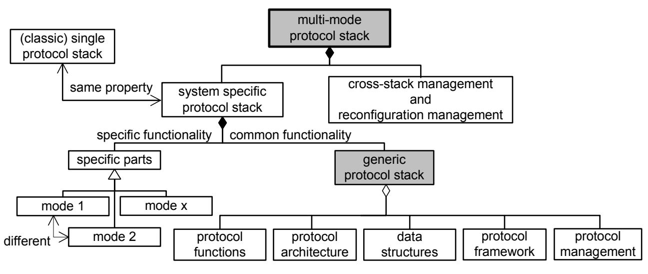


Fig. 2: UML diagram of the generic protocol stack in the context of multi-mode wireless communication systems.

realized in adding cross-stack and reconfiguration management related functions. The reconfiguration management of a stack contains functions beyond mode transition as for instance the support of a network initiated reconfiguration or the administration of a software download of specific parts and updates of new releases for the maintenance of the protocol software.

V. Partitioning Functionality - Why and How?

The HDLC protocol from above should serve here as an example to show how protocols belonging to different air interfaces can exhibit a high degree of commonalities.
This paper presents a reference model for future protocol stacks supporting multiple modes of an air-interface by exploiting their commonalities and by providing an efficient way of re-using generic functions for air-interface-specific purposes.
The reference model presented in this paper is based on the widespread perception that radio interface protocol functions can be divided into two sets of functionalities:

1. **Mode-/System-specific functions**: These are protocol functions that are unique to a certain kind of radio interface mode and can not be found in any other mode of the same or any other radio interface. Examples for such mode-specific functions are the allocation of a dedicated physical resource and parameters for dimensioning a mode related to the local communication environment.
2. **Generic (common) functions**: The view that is usually taken in standardization is that "generic" functions are not fully specified and have to be enriched by specifying missing parts. This view does not apply in our case. It has to be noted that in our context, the "generic" functions are assumed to be the identified set of common (mode independent) functions of a set of air interface modes. This means they are "generic" from the viewpoint of the modes, but **not from the viewpoint of functionality**. It is assumed that these functions can be

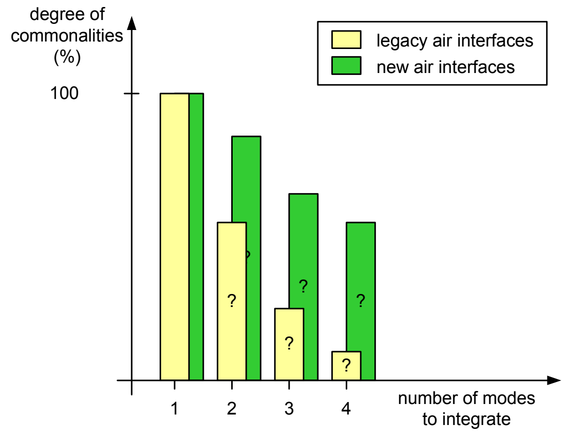


Fig. 3: Estimated degree of commonalities in multi-mode protocol stacks as a function of the number of integrated modes. New air interfaces can benefit from cooperative design taking general usability of the protocol into account.

adapted to the use in any of the targeted modes through proper parameterization. They are generic in the following sense: In most cases, they will have to rely on additional mode-specific functions to provide the full functionality of a certain protocol layer of a certain air interface mode. An ARQ protocol for instance is a generic protocol which can be specified as Go-Back-N, Selective Reject or Hybrid ARQ protocol providing an error-free data transfer. In regarding the generic protocol stack as toolbox of common protocol functions which may be used in a specific protocol implementation the multitude of protocol functions considered as being generic is increased: It suffices that they can be found in at least two protocol modes to be counted as belonging to the toolbox of generic protocol functions.

A general and trivial observation is that the degree of commonalities between a set of different air interface modes is decreasing with the number of modes being integrated, as depicted in Fig. 3. The interesting observation that can be made in this context is that the design processes of new air interfaces can be expected to highly benefit from cooperation, resulting in the potential to increase commonalities between these new air interfaces and thus reduce development and equipment costs. Thus, the degree of similarities can be improved, when considering multi-mode capability based on genericity during the development and standardization of new protocols compared to the common parts of existing 2G and 3G protocols.

VI. The Multi-Mode Reference Model based on Modes Convergence

A. *A Multi-Mode Protocol Architecture*

Fig. 4 illustrates the architecture of a multi-mode protocol stack for a flexible air interface. The layer-by-layer separation into specific and generic parts enables a protocol stack for multiple modes in an efficient way: The separation is the result of a design process that is referred to as cross-stack optimisation, which means the identification and grouping of common (generic) functions. The generic parts, diagonally hatched in Fig. 4, of a layer can be identified on different levels as introduced in Section III. The generic parts are reused in the different modes of the protocol stack. All generic parts together can be regarded as generic protocol stack [9]-[11]. The composition of a layer out of generic and specific parts is exemplarily depicted in Fig. 4. The composition and (re-)configuration of layer is performed by the (N)-Layer Modes Convergence Manager ((N)-MCM). The protocol modules of generic functions are exemplarily introduced: Some of them are reused in a layer and/or additional functions are taken from the toolbox of common protocol functions as part of the generic protocol stack. The RRC on the control-plane and the RLC on the user-plane are generic to the layers located above. A mode specific protocol stack has an individual management-plane. Radio Resource Management (RRM), the Connection Management (CM) and the Mobility Management (MM) are located in the Radio Resource Control (RRC) layer. The cross-stack management of

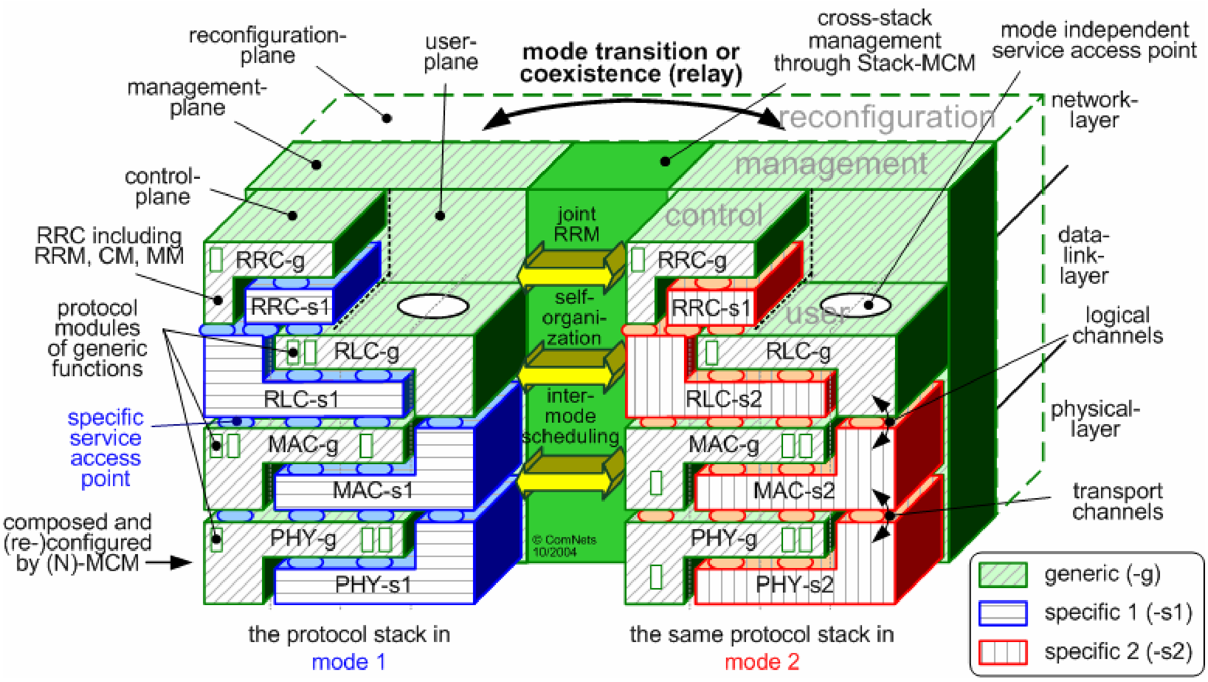


Fig. 4: The Multi-mode protocol architecture, facilitating transition between modes (inter-mode handover) and coexistence of modes (in relay stations connecting different modes) by way of the cross-stack management supported by the modes convergence manager of a layer or stack.

different modes completes the reference model for multi-mode protocols in connecting the management-planes of the device's modes with the help of the Stack Mode Convergence Manager (Stack-MCM). Stack-MCM and (N)-MCM exchange in a hierarchical order data between two modes. The transition between modes and the coexistence of several modes is performed by the Stack-MCM. The (N)-MCM enables composing a layer out of different parts as depicted in Fig. 4 and introduced in the previous section. The split between user and control-plane is limited to the network layer, as known from H/2, and the DLL is used for both, signalling and user-data transfer.

The (N)-MCM is the intermediator between the generic and specific parts of the multi-mode protocol stack's layers: All SAPs, i.e., interfaces, touched during the mode transition of a layer are administrated by the (N)-MCM. In the classical view of protocols as state machines, the (N)-MCM transfers all state variables of the protocol layer between two modes with the help of the Stack-MCM. At our example of the HDLC protocol this would imply the state transfer of being connected from one mode to the other together with a data transfer of received but unconfirmed data frames.

Concrete, the (N)-MCM manages a single layer and has the following tasks and responsibilities which are introduced later in this section:

- Layer composition and reconfiguration, considering all interfaces related to the transition between two modes
- Protocol convergence:
    - horizontally – between "generic" and "specific" parts
    - vertically – mapping of higher layer user data flows for RLC as known for instance from ATM
- Data preservation and context transfer

Furthermore, the convergence between mode specific protocol stacks is realized through the Stack-MCM implying implicitly the following functions:

- Joint Radio Resource Management (Radio resource coordination) between different modes
- Inter-mode scheduling
- Self-organization (frequency allocation of adjacent relays and APs, user data flow routing)

The reconfiguration-plane located behind the management-plane, see Fig. 4, is not considered in this paper. The Stack-MCM realizes the reconfiguration of the protocol stack from one mode to the other in providing services to the reconfiguration-plane. The reconfiguration plane contains all functions related to reconfiguration management [12] as for instance the security aspects of reconfiguration and software download as well as the communication of the reconfiguration capabilities of a device.

B. Composition of a Layer from Specific and Generic Parts

Fig. 5 shows the general structure of a protocol layer conforming to our reference model. It is assumed that the functionality inside the layer is always composed of a generic (common to all modes) part and mode-specific parts, which jointly provide the modes' services of the layer via Service Access Points (SAPs). Through this, layers can be configured for one mode at a time. Modes can also coexist temporarily or permanently. The specific SAPs of a layer are defined via the currently used mode or set of modes[1]. This does not preclude the possibility that SAPs of different modes can be accessed by higher layer entities in a common way as visualized by L(N)-SAP-g.

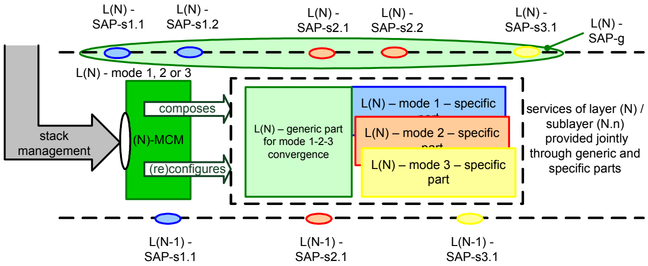


Fig. 5: Composition of a layer (N) from generic and specific functions. The composition and (re-)configuration is handled by the (N)-MCM. The (N)-MCM is controlled by a layer-external stack management entity, namely the Stack-MCM.

The composition and (re-)configuration of the layer is taken care of by a layer-internal instance, the (N)-Layer Modes Convergence Manager ((N)-MCM), which resides in the management plane. The (N)-MCM enables that a (N)-layer provides multiple modes and makes functionality of one mode or common to several modes available. An instance of the MCM serves as a reconfiguration handler in each layer of the air interface protocols.

Fig. 6 depicts two exemplary cases of a (N)-layer: In Fig. 6(a) a completely generic layer configured (if necessary) for mode 1 by the (N)-MCM is shown. The common, generic part is thereby composed by the (N)-MCM. Contrary Fig. 6(b) illustrates a layer, where the different integrated modes exhibit no commonalities. Here, the layer cannot provide generic services. The role of the MCM is restricted to support the Stack-MCM by choosing the demanded mode specific part.

C. "Mode Transition" vs. "Mode Coexistence"

The Stack-MCM administers the protocol stacks of several modes under consideration of the environment, i.e., receivable modes at the location of a device and requirements of the user and its applications. The reconfiguration of the protocol stack from one mode to the other is referred to as mode transition. The mode transition may be done on several levels depending on architectural constraints of the protocol layers and the general treatment of generic parts: In case of a unique permanent existing

[1] As a result of the actual configuration of a protocol layer (N) for a mode X, it makes available mode X's (N)-service number Y via a mode X specific (N)-SAP for the service Y (Notation used: (N)-SAP-sX.Y)

DLL [13] of functions which are reconfigured (re-parameterized, restructured or extended) the mode transition is limited to the layers below – the MAC and PHY. The same stands for the case of a protocol stack of two modes used for relaying as introduced below: The termination of an end-to-end retransmission protocol above the relay limits the considered protocol layers during transition from one relaying-mode to another. Thus, mode transition may be done (i) on MAC level or (ii) on RRC level depending on termination of generic parts of the DLL.
Temporal or permanent in parallel existing modes of a protocol stack are referred to as mode coexistence. Mode coexistence can be reasoned through, (i) a relaying function in case of two simultaneous existing modes or (ii) the simultaneous connection to multiple modes for other reasons (a dedicated mode for broadcasts, applications of the user with different QoS requirements, cost preferences from user, etc.) or (iii) short-term coexistence for inter-mode handover and (iv) seamless mode handover.

D. Multi-Mode Relaying

A relay, either homogeneous or heterogeneous (multi-mode), operates with two coexisting protocol stacks that have to be connected (bridged). The Stack-MCM as intermediator has to provide cross stack management functions to connect the two mode specific protocol stacks of the relay. Thereby the (N)-MCM has to provide mapping functions between the two modes as for instance the mapping of data format, behaviour and status information between the two relaying-modes. Therefore, the (N)-MCM has to transfer the incoming information of one mode to the outgoing link of the other mode. The mapping is related to the control-plane as well as to the user-plane. Multi-mode relaying explicitly implies multiple hops over multiple different modes and requires routing algorithms as multiple paths may exist as addressed in Section F under self-organization. The (N)-MCM provides joint RRM functions to adapt the radio resource allocation of the relaying-modes in order to avoid of buffer overflows, if the relay downlink does not provide as much capacity as the relay uplink for one connection or to avoid the waste of capacity in the case of higher downlink capacity.
Further the status information, e.g. idle connection, has to be mapped from one mode towards the other. The status information can be easily mapped for generic parts and has to be translated for the mode specific parts. The common generic parts enable relaying on different levels.

E. Functions of the (N)-Layer Modes Convergence Manager ((N)-MCM)

Protocol Convergence

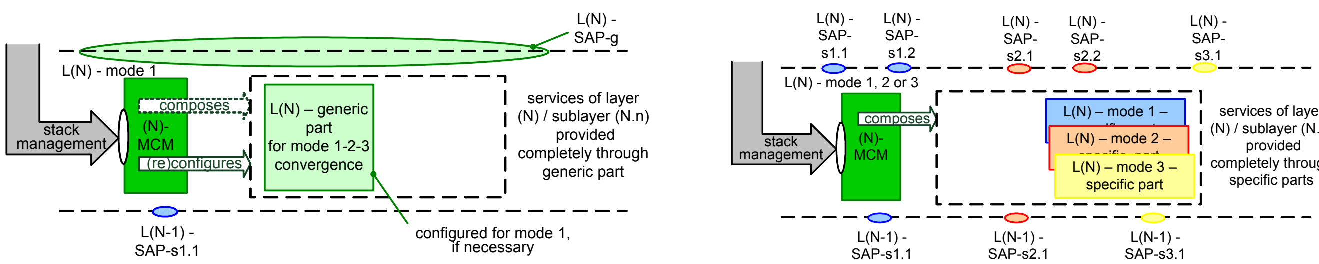
The convergence of multi-mode protocol stacks has two dimensions: First the convergence between two adjacent layers, in the following referred to as vertical convergence as it is known from the user-plane of H/2 protocol stack. Second the convergence between layers located in the different modes of the protocol stack which have the same functions: In the following referred to as horizontal convergence. The generic protocol stack, managed by the (N)-MCM as introduced above, enables both the horizontal as well as vertical protocol convergence.
From the perspective of higher layer protocols the multi-mode protocol stack is transparent on the user- as well as on the control-plane, i.e., generic parts terminate the stack to the layers above, as depicted in Fig. 4. The vertical conversion of the (N)-MCM implies the adaptation of the on the multi-mode protocol stack working packet data protocols to the specific mode. This may be for instance the conversion of an IP datagram in compressing the IP-Header.

Layer Composition and Reconfiguration

The separated approach of generic and specific parts requires an administration when taking the transition between modes into account: The common generic parts of the old mode need to be adopted for being reused in the new mode of the protocol stack. We assume that the generic parts of a layer exist permanently and are to be reconfigured and/or recomposed by the (N)-MCM corresponding to the characteristics of the targeted new mode. This assumption may imply a module-based composition concept of the generic parts as introduced in [10] and [11]. The composition and configuration of the layer out of generic and specific parts, see Section B, is done by the (N)-MCM.

Data Preservation and Context Transfer

The communication between two modes and the mapping



(a) completely generic **(b) mode specific without common part**

Fig. 6: Exemplary composition of a layer (N). Two extremes are depicted.

between generic and specific parts of a mode is done by the (N)-MCM (inside layer) and the Stack-MCM (transfer between modes). The transition between two modes can be optimized in using the data from the old mode to the new mode. The ability of a user-plane protocol to reuse status information in the generic part and protocol data after transition to another mode requires an extension of the protocol into the control-plane though it performs only user-plane tasks. Depending on the status of the related protocol parts, the data transfer is referred to as "data preservation" or "context transfer" as illustrated in Fig. 4: If the generic part is reconfigured and recomposed the data needs to be preserved, i.e., adopted, to the new mode. In the case of a deletion of the old specific/generic part the data transfer is named "context transfer" which implies preservation for the new mode.

F. Functions of the Stack Modes Convergence Manager (Stack-MCM)

Joint Radio Resource Management

The functions of the user and control-plane are administrated by the RRM which may be coordinated centralized or decentralized and the RRM decisions are executed by the RRC of the corresponding modes. The RRM may assign multiple modes to one specific data flow. The RRC provides status information about the mode specific protocol stack in a generic structure to the RRM of the multi-mode protocol. In case of a (semi-)centralized coordination of the radio resource allocation, this generic information structure about the status of the different modes of the protocol stack can be transmitted to enable an adequate decision.

The RRM of a single multi-mode device may also support the coordination across neighbouring operating devices as for instance the coordination across base stations.

Inter-Mode Scheduling

The Stack-MCM as intermediator between modes performs scheduling among different modes, as illustrated in Fig. 4. Contrary the scheduling inside a mode across logical/transport channels: It is done in MAC-g or RLC-g of the specific mode's protocol stack. The inter-mode scheduling considers the dynamic scheduling of different user data flows over multiple modes. The scheduling strategy may for instance be based on the modes' interference situation which requires a provision of necessary information directly from the PHY if the decision is done in the MAC independent from RRC/RRM. This information about the quality of the radio link is again provided in a generic information structure.

Self-Organization

The envisaged communication system is able to autonomously decide about its radio resource allocations in taking the environment into account. This implies the for instance the adequate selection of frequencies used for transmission or the routing of user data packets. The radio resource is selected under consideration of interference avoidance with other radio systems. Further, the optimized spectrum utilization coordinated with neighbouring radio systems of the same technology is taken into account, which may also be related to efficient multi-hop relaying depending on the selected deployment scenario. The self-organisation comprises scenarios of breaking down and installation of additional devices in an operating communication system. The Stack-MCM has to support the addressed functionalities in activating for instance different modes to provide information about the interference situation or the role of devices (if it is acting as relay or access point) in reception range.

VII. Layer-Based Separation into Generic and Specific Parts

Generic Protocol Stack is based on generalized common protocol functions sorted by the corresponding layer: PHY-g, MAC-g, RLC-g and RRC-g. Each layer is composed out of generic and specific parts as depicted in Fig. 4. They are connected through generic or specific SAPs administrated by the (N)-MCM following the definitions from above. Analogous to the data link layer, MAC-g and RLC-g can be regarded as Generic Link Layer (GLL) [13]. Taking up the idea of a toolbox consisting of general protocol functions, the introduced assignment to specific layers is not limiting their utilization to this single layer, as for instance the segmentation function which may be used on various levels of a protocol stack. In the following a first, preliminary separation of layers into generic and specific parts is suggested.

A. Separation of the Radio Resource Control into RRC-g and RRC-s

The RRC, as depicted in Fig. 4, contains all algorithms which are required for configuration and operation of a specific mode of the protocol stack. The RRC is generic to the layers above to enable an efficient RRM of multiple modes independent from the characteristics of a single mode. The RRC establishes on demand a connection through the complete protocol stack for transferring user data. Therefore RLC, MAC and PHY are configured corresponding to required QoS parameters of the users' applications. Further the RRC is responsible for the location management and inter-/intra cell handover, also regarded as MM-functions. It supports the cross-stack RRM in collecting information in the different layers for optimizing radio resource allocation of all modes together. The RRC executes the decisions of the joint RRM as for instance the control of a dynamic allocation of the radio resource. The information collected by the RRC of a specific mode may be provided in a generic information structure to the RRM of the multi-mode protocol as introduced in Section VI.F.

In case of a (semi-)centralized coordination of the radio resource allocation, this generic information structure about the status of the different modes of the protocol stack can be transmitted to enable an adequate decision. The CM can be realized with a high grade of genericity as it provides mode independent functions as the call control.

B. Separation of the Data Link Layer into Specific and Generic Parts

The following functions can be regarded as belonging to MAC-g and RLC-g. They are part of the generic protocol stack as toolbox of common, reusable protocol functions:

- Error handling with the help of (i) ARQ protocols [10] or (ii) Forward Error Correction (FEC)
- Flow control
- Segmentation, concatenation and padding of PDUs [11]
- Discarding of several times received segments
- Reordering of PDUs for in-sequence delivery
- Multiplexing/De-Multiplexing of the data flow, as for instance the mapping of different channels
- Dynamic intra-mode scheduling between user data flows within one mode
- Ciphering
- Header compression

The RLC is used by the control and user-plane, as depicted in Fig. 4. The RLC provides a reliable connection for the signalling of the control-plane. The DLL provides multi-mode macro diversity for the transmission and reception of data depending for instance on the temporal quality of a radio link. This dynamic scheduling of user packets over multiple modes requires accurate information about the state of the physical channels, which has to be provided by the PHY-g in a generic information structure. The MAC layer coordinates the access to the physical medium. Thus the specific resource allocation of a mode is done by the MAC-s.

C. Separating the Physical Layer into PHY-g and PHY-s

Both, the protocol (software) parts of a reconfigurable physical layer and the reconfigurable modem hardware originate from Software Defined Radios (SDR) [14]. Reconfigurable functional modules used for signal processing are for instance the forward error correction or interleaving. The synchronisation, fast power control or the support of soft and softer handover for macro diversity can be regarded as generic functions of PHY-g. The PHY-g supports physical channel related decisions in MAC and RRC: Generic information about status of physical channels is reported to the RRC is for instance the Frame Error Rate, the Signal to Interference Ratio, the measured interference power etc. The specific characterization of the utilization of the physical resource as for instance the modulation, waveforms, codes used for spreading and scrambling or frequencies used for radio transmission are part of PHY-s.

VIII. CONCLUSION

The introduced multi-mode protocol reference model facilitates the structuring of an arbitrary layer into generic and specific parts. In providing guidance for understanding this structuring it marks up optimization potential in questioning the necessity of the indicated differences. In this way, an increased protocol convergence is reached enabling an efficient multi-mode protocol stack for future wireless systems – especially in the context of relay-based wireless mobile broadband systems as envisaged by the WINNER integrated project.

ACKNOWLEDGEMENTS

The work presented in this paper has been funded by the European Commission in the 6^{th} Framework Program as part of the integrated project WINNER (IST-2003-507581).

REFERENCES

[1] http://www.wireless-world-initiative.org, November 2004.

[2] R. Pabst et al., "Relay-Based Deployment Concepts for Wireless and Mobile Broadband Radio," *in IEEE Communications Magazine*, vol. 42, no. 5, pp. 80-89, September 2004.

[3] ISO 4335, "HDLC elements of procedures", September 1978.

[4] GSM 04.06. "Digital Cellular Communication System (Phase 2+); Mobile Station – Base Station System (MS-BSS) Interface; Data Link (DL) Layer Specification," *Technical Report*, ETSI, 1996.

[5] ETSI ETS 300 175-4. "Digital Enhanced Cordless Telephony (DECT); Common Interface (CI); Part 4: Data Link Control (DLC) Layer," *Technical Report*, 1998.

[6] 3GPP TS 24.022. "Radio Link Protocol (RLP) for circuit switched bearer and teleservices (Release 5)," *Technical Specification*, December 2003.

[7] D. R. Musser, "Generic Programming," http://www.cs.rpi.edu/~musser/gp/, November 2004.

[8] E. Gamma, R. Helm, R. Johnson, J. Vlissides, "Design Patterns – Elements of Reusable Object-Oriented Software," *Addison Wesley*, 22nd Printing, July 2001.

[9] M. Siebert, B. Walke, "Design of Generic and Adaptive Protocol Software (DGAPS)," *3Gwireless '01*, San Francisco, US, June 2001.

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

[11] L. Berlemann, A. Cassaigne, R. Pabst, B. Walke, "Modular Link Layer Functions of a Generic Protocol Stack for Future Wireless Networks," *SDRforum'04*, Phoenix USA, November 2004.

[12] N. Alonistioti, A. Glentis, F. Foukalas, A. Kaloxylos, "Reconfiguration Management Plane for the Support of Policy-based Network Reconfiguration," *PIMRC'04*, Barcelona, Spain, September 2004.

[13] J. Sachs, "A Generic Link Layer for Future Generation Wireless Networking," *ICC'03*, Anchorage USA, May 2003.

[14] http://www.sdrforum.org, November 2004.