

Functional Analysis of a SDR Based Bluetooth/HiperLAN Terminal Demonstrator

Fokke Hoeksema, Roel Schiphorst, Kees Slump
 University of Twente, Department of Electrical Engineering,
 Laboratory of Signals and Systems,
 P.O. box 217 - 7500 AE Enschede - The Netherlands
 Phone: +31 53 489 2770 Fax: +31 53 489 1060
 E-mail: f.w.hoeksema@el.utwente.nl

Abstract—In our *Software Defined Radio (SDR)* project we aim at combining two different types of standards, Bluetooth and HiperLAN/2 on one common hardware platform. HiperLAN/2 is a high-speed Wireless LAN (WLAN) standard, whereas Bluetooth is a low-cost and low-speed Personal Area Network (PAN) standard. An SDR system is a flexible radio system that is re-programmable and re-configurable by software in order to cope with its multi-service, multi-standard and multi-band environment.

Goal of our project is to generate knowledge about designing the front end of an SDR system where especially an approach from *both* analog and digital perspective is essential.

To what extent can we use the HiperLAN/2 receiver hardware for our Bluetooth receiver?

In this paper we present a functional architecture that brings the architectural descriptions of both standards to an equal level. This *SDR functional architecture* is used in the sequel of the project for a number of purposes, of which we mention

1. Definition of reference points (for requirements definition purposes).
2. Definition of interfaces (potential alignment with SDR Forum).
3. Delimitation of our demonstrator (what is it that is going to be built).
4. Identification of inter-standard functional integration challenges.

Keywords—Software Defined Radio, Bluetooth, HiperLAN/2, Functional Modelling, Functional Architecture

I. INTRODUCTION

IN THE NEXT year of our *Software Defined Radio (SDR)* project we aim at combining two different types of standards, Bluetooth and HiperLAN/2 on one common hardware platform. HiperLAN/2 is a high-speed Wireless LAN (WLAN) standard (e.g. [5] and [6]), whereas Bluetooth is a low-cost and low-speed Personal Area Network (PAN) standard ([11]). An SDR system is a flexible radio system that is re-

programmable and re-configurable by software in order to cope with a multi-service¹, multi-standard and multi-band environment. As can be seen in Table I the standards differ in several aspects and pose an interesting challenge for an SDR platform.

TABLE I
 BLUETOOTH AND HIPERLAN/2 PARAMETERS

	Bluetooth	HiperLAN/2
System	PAN	WLAN
Frequency Band	2.4-2.4835 GHz	5.150-5.300 GHz, 5.470-5.725 GHz
Access Method	CSMA	TDMA
Duplex Method	TDD	TDD
Modulation Type	GFSK	OFDM
Max. Data Rate	1 Mbps	54 Mbps
Channel Spacing	1 MHz	20 MHz
Max Power Peak	100 mW	200 mW - 1 W

Goal of building a HiperLAN/2-Bluetooth demonstrator is to generate knowledge about designing the front end (say, from antenna to and including demodulation) of an SDR system. Typical SDR aspects of interest are an investigation of integration challenges and identification of flexibility aspects. With "investigation of integration challenges" we mean finding an answer to the question how to use HiperLAN/2 hardware and software for implementation of Bluetooth functionality. HiperLAN/2 is a standard that leads to more complex implementations than Bluetooth (Table I may give an impression) - we don't expect Bluetooth capable hardware and software to be able to implement HiperLAN/2. With "identification of flexibility aspects" we mean identifying possibili-

¹With a *multi-service system* we mean a system that is able to handle different types of data traffic: different with respect to content (email,web,audio,video,speech, ...), different with respect to traffic patterns and different with respect to QoS requirements.

ties to switch between HiperLAN/2 and Bluetooth systems (make the integrated system reconfigurable and reprogrammable). While the aspects above are emphasized in the object-oriented software approach of Mitola (see [7]), our interest is in signal processing aspects and hardware. In our project the focus is on *both* analog and digital (signal processing) functions - the question where to place the Analog Digital Converter (ADC) is of paramount importance for our research. Moreover, the choice of (programmable) hardware platform (COTS, GPP (x86), DSP, FPFA, FPGA, ASIC, ...) is subject of research. A first objective of the project is the realization of a Bluetooth - HiperLAN/2 test-bed within a one year time-frame.

In this paper we first briefly describe our PROGRESS project. In the next year, we aim at building a Bluetooth - HiperLAN/2 demonstrator. In this paper we focus on the first phase of this building activity: the architectural phase. We specify what the objectives are in this phase of the our project.

One of the goals in this phase is to present a functional architecture that aims at bringing the architectural descriptions of both standards to an equal level. We have two reasons for wanting this: The first reason is that we need a tool for specification and requirements engineering purposes. The second reason is that we are interested in a method for investigation of integration challenges and identification of flexibility aspects in an early stage of the design process. We present candidate *Functional Models* that are tested for enabling these objectives within the time-frame of our project.

We decided to use a so-called *Signal Path Functions Model* (SPFM) for further specification and requirements engineering and design of our demonstrator, thereby leaving an elaborate analysis of the complete inter-standard integration challenges and flexibility aspects undone. However, we present a unified Bluetooth and HiperLAN/2 model using our SPFM. The paper concludes with an assessment of the SPFM: Its capability for specification and requirements engineering purposes is assessed and its capability for identifying integration challenges and flexibility aspects is discussed.

II. SDR PROJECT

A. PROGRESS Project

The research of our SDR Project (as proposed in [1], quoted in this section) is directed at the design and realization of an embedded mobile terminal for

speech and data. In the traditional way of designing, mobile terminals such as DECT and GSM telephones are designed and realized with focus on low-cost hardware. Adjustment to the wishes of users or to specific circumstances is then hardly possible. An embedded approach, however, in which a large part of the functionality is realized in software, can provide the desired flexibility and, within boundaries, the capability to expand.

A software radio in the terminal is more flexible than a conventional radio and finds out itself which transmission frequency, modulation scheme and protocol is used. Also updates in software can easily be downloaded from the wireless network. A software radio offers much more functionality and flexibility than today's dual-band or tri-band terminals, in which just 2 or 3 radios are *merged together*. An (ideal) software radio is defined as a transceiver where the signal is digitized at the antenna and all of the processing is performed by software. A more realistic definition of a software radio, is a transceiver in which the digitization is performed at some stage downstream from the antenna and all digital processing is performed by software residing on reconfigurable hardware (a software *defined* radio). At the receiver side the digitization typically takes place after wide-band filtering, low-noise amplification, and down-conversion to a lower frequency in subsequent stages. For the transmit side the reverse occurs.

In this project, we aim at a configurable and programmable front end which can be defined to a large extent by software. This configurable and programmable front end can operate on a continuous scale of performance, being optimal for the circumstances offered. For example, resolution and bandwidth of an A/D converter can be traded off with power consumption, through simple software.

A good analog-to-digital (AD) conversion is essential for the software defined radio concept. The location of this AD converter will be shifting further towards the antenna in the future. Its exact location will however depend on the available technology and design knowledge, and is thus a function of time. Analog and digital are not treated separately, but the optimal solution for the total system is chosen. The AD conversion will be put at the right place. Also the partitioning between digital hardware and software is a moving target. Instead of just building a software radio with components available today, we aim at deeper insight in the trends of the partitioning between analog/digital/software.

The project aims at (overall project objectives):

- Finding a design methodology, in which in a structural way the right architecture (i.e. partitioning in analog/digital/software) is found as function of the available technology.
- A real design example to prove the methodology and to deliver the chosen functionality, build with off-the-shelf components. As mentioned before, the analog front-end will be programmable/configurable and the *software* will be implemented by using programmable hardware.

In a one-year time frame we aim at implementing a demonstrator in which parts of two different types of standards, Bluetooth and HiperLAN/2 are implemented on one common hardware and software platform. This should help us in gaining insight for a second part of the project in which the overall objectives, sketched above, are met.

B. Bluetooth - HiperLAN/2 demonstrator

In order to build, test and evaluate an SDR front end that combines the Bluetooth and HiperLAN/2 standard a demonstrator will be built that consists of both the front end itself and a test-environment for this front end. We intend to demonstrate the software-defined radio concept by building (part of) the physical layer functionality of both standards. Exactly what part we are going to demonstrate is to be determined by studying the architectures of the two standards.

In the architectural phase of the project we aim at providing a description of our front end and its environment in *SDR architectural concepts*. Using these concepts we specify what we are going to build (the subject of research) and what we consider the environment of the front end (the test-system). Moreover, early ideas about what to test, evaluate and demonstrate need to be formulated.

Our intention is not to implement both a transmitter *and* receiver. Two candidate test-bed configurations are under consideration: a test-bed in which existing HiperLAN/2 and Bluetooth transceivers are used to test our multi-standard receiver and a testbed in which the RF antenna signals are replaced by programmable RF function generators. A rough sketch of the Transceiver-Laptop test-bed is given in Figure 1.

In this demonstrator, there *need* to be (one or two) base-stations and a laptop computer, all containing transceiver (transmitter and receiver) functionality.

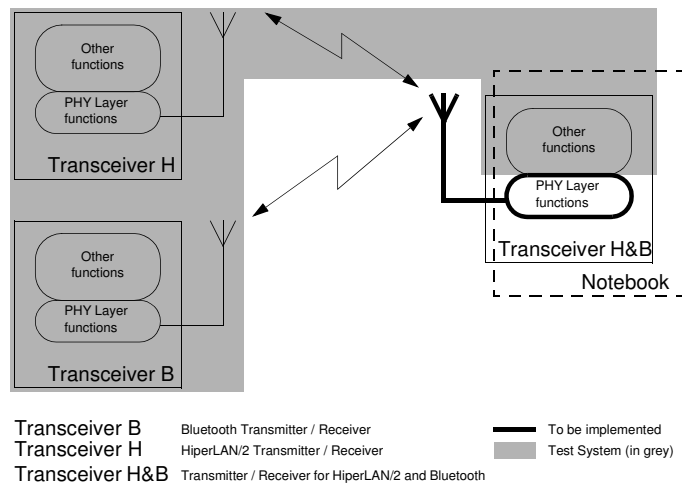


Fig. 1. Rough Testbed Outline.

However we do not plan to *build* all these transceivers. The line of thinking is to use two base-stations (a Bluetooth one, and a HiperLAN/2 one) and prepare their transceivers for transmitting test signals. As we may have to perform some stripping of existing systems (COTS components) and, moreover, the focus of the project is *not* on antenna-design, the Signal Generator-Laptop test-bed appears to be an interesting alternative.

III. SDR FUNCTIONAL ARCHITECTURE

As a first step in our design effort we are interested in finding a so called *SDR Functional Architecture*. This *SDR Functional Architecture* is meant to be used in the sequel of the project for a number of purposes, that can be divided into two categories: 1/. Specification and Requirements engineering purposes (narrow scope objectives) and 2/. Identification of integration challenges and flexibility aspects (wide scope objectives).

For specification and requirements engineering purposes the SDR functional architecture should enable (the narrow scope objectives):

- *The definition of reference points.*

The architecture should enable requirements definition and specification of systems parts.

- *The definition of interfaces.*

As we do not plan to build all system parts ourselves, we need to be able to identify possibilities for usage of existing COTS components, possible re-use of existing software packages and potential alignment with SDR Forum [14] architectures and implementations.

- *The delimitation of our demonstrator.*

We need to be able to specify what it is that is going

to be investigated and built and what the test environment is.

The wide scope objectives are

- *The identification of inter-standard functional integration challenges.*

The assumption made here is that an architectural model can be used as a tool for specifying a particular software defined radio *instance*, e.g. the Bluetooth instance and the HiperLAN/2 instance at a meaningful level of detail. If this assumption is correct, a functional model might be useful in defining possible system parts for functional integration. Moreover, this may help to identify reconfigurability and reprogrammability aspects. Also, such a model might help in assessment of flexibility and re-use of system parts (possibly at the cost of implementation efficiency; however, the latter is *not* the main objective of our project).

- *The identification of system aspects that need to be communicated to allow dynamic reconfiguration of a mobile.*

If a mobile is to change its personality, it needs to be informed on what aspects to change. For this to happen, some sort of interaction-mechanism needs to be developed in which aspects that need to be communicated are identified. Once an architecture is selected it may be possible to identify these aspects and design the necessary protocols.

What is it we mean by an *SDR Functional Architecture*? What type of functional model do we need for the objectives above? Below, a *good* functional model (good choice of concepts) is one that helps us to create an architecture that helps us to satisfy the objectives listed above. We investigated the following candidate models:

- The SDR Forum Architectural Model
- A Layered Protocol Model
- A Signal Processing Functional Model
- A HW/SW Functional Model

A. SDR Forum Architectural Model

In Figure 2 (taken from [14]) a so-called *SDR Forum Interface Chart* is presented. It is a model from an object-oriented software context in which rectangular shaped functional blocks are shown together with the interfaces between these functional blocks. The boxes with rounded edges at the top of the figure represent information-flow formats. The aim is to identify interfaces between software components that can be used in SDR systems.

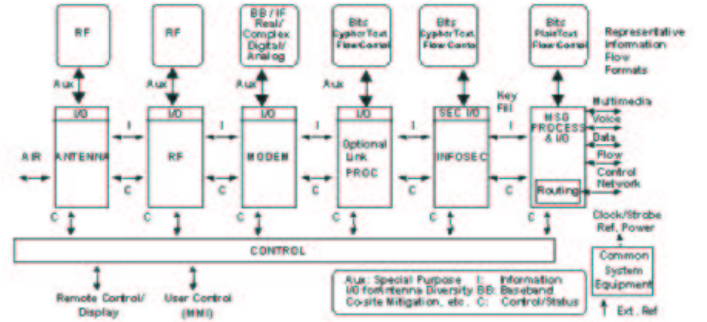


Fig. 2. SDR Forum Interface Chart (taken from the SDR Forum Website, [14]).

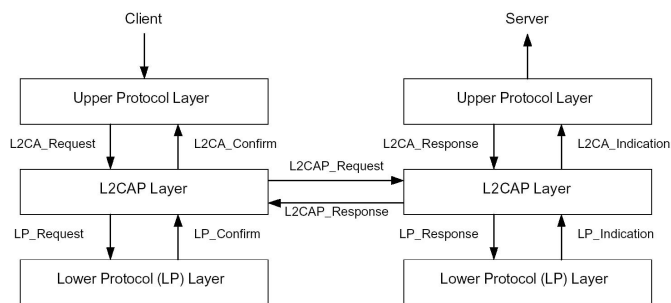
Will this *SDR Forum Architectural Model* enable us to create a good SDR functional architecture? In our project we focus on the RF and MODEM functions, however not as objects in object-oriented software, but as signal processing functions implemented in programmable hardware and software. This model is considered to be either too weak or too general for our (test-bed) specification purposes. Although we could try to investigate whether software modules are available at the ANTENNA \leftrightarrow RF interface or MODEM \leftrightarrow Optional Link PROC interface, we are skeptical about the success of this effort: with respect of availability of software, with respect to intellectual property aspects and with respect to ease and timeliness of integration into our test-bed.

The wide scope objectives may be better served with this model. However, in our project we do not intend to obtain these objectives at an object-oriented software abstraction level, but at a level in which Signal Processing concepts and HardWare/SoftWare (HW/SW) concepts are used. Basically we think that it is useful to look at a system at different levels of abstraction (using different concepts) in order to fully appreciate integration and flexibility options. As the SDR Forum object-oriented model is rather general and needs to be tailored to the standards we are interested in anyway, we choose not to pursue this path.

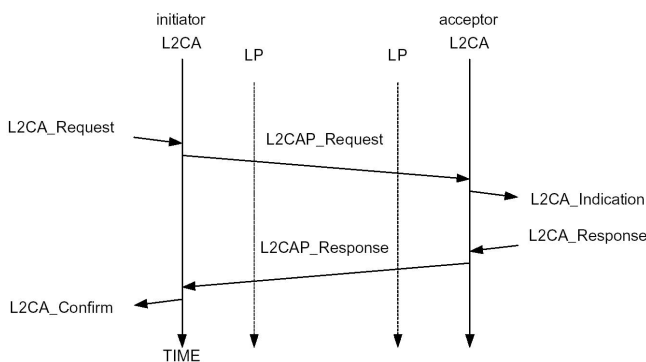
B. Layered Protocol Model

What do we mean with a *Layered Protocol Model*? We will give an example layered protocol model, taken from the Bluetooth specification [11]. In Figure 3 two figures describing Bluetooth's Logical Link Control and Adaptation Protocol (L2CAP) function are depicted. L2CAP enables multiplexing of higher layer protocols and takes the initiative in ordering (German: Gibt eine Anweisung) the Link Manager protocol-entities to set-up and release a connection

between one or more Bluetooth-capable computers.



(a) Protocol Entities and Interactions at Interaction Points.



(b) Example Message Exchange (temporal ordering of interactions).

Fig. 3. L2CAP Protocol Stack (taken from [11]).

A layered protocol model is a description of a communication system in which system-parts that execute functions (actors) and interfaces with other system parts are defined. Moreover, the behaviour of the actors is described: the *interactions* system parts have with one another; the *parameters-and-data* these system parts exchange and the *temporal ordering* of the interactions. Both *system entity-structure* and *system behaviour* need to be described. In Figure 3a we see the system entity-structure of the Bluetooth L2CAP protocol and its environment: the system parts (protocol *entities*, not so much protocol layers are shown). The Interaction Points are implicitly shown as start-point and endpoint of the vertical arrows. At these interaction points, Service Primitives (see Footnote 5 on page 7) are executed. So, the vertical arrows themselves denote the Service Primitives (locally executed procedures). The name of some of the procedures is given in the figure. The horizontal arrows denote *messages* or Protocol data Units (PDUs): the data units

exchanged between peer protocol entities. In Figure 3b an example of system behaviour in the form of a Message-Sequence diagram is shown. We see how a locally initiated procedure (e.g. L2CA-Request) results in a message L2CAP-Request that subsequently results in the execution of another local procedure L2CAP-Indication.

So, does a *Layered Protocol Model* enable to create a good SDR functional architecture? In our project we focus on the physical Layer functionality and not on the entire Bluetooth or HiperLAN/2 standard. Given our objectives (the narrow scope ones), creating a complete Layered Protocol model is not required and will take too much of our time. Moreover, initial attempts in creating a layered protocol model of both standards showed that while the HiperLAN/2 standard is more or less easily modelled, the Bluetooth standard is not. In the latter standard a mix of protocol descriptions, chip-functionality descriptions and interface descriptions are presented that do not translate readily into a layered protocol model of the entire system².

The wide scope objectives are believed to be served by creating a full layered protocol-model of both standards. Whether a layered protocol model offers better insight into the wide scope objectives than an object oriented approach is, to the knowledge of the authors, not known. However, even if integration challenges and flexibility aspects are identified in an integrated layered protocol model, other aspects may exist that can only be found by investigating the system using a signal processing model and/or HW/SW model.

C. Signal Processing Functional Model

With a *Signal Processing Functional Model* we mean a model in which functional blocks like 'Amplifier', 'Filter', 'Mixer', 'Sampler', 'Quantizer', 'Detector' etc. are present. The interactions between functional entities are 'Signal's. These functions can be implemented using either analog or digital hardware or in software. The system is basically analyzable with Signals & Systems Theory.

Does a *Signal Processing Functional Model* result in a good SDR functional architecture? Radio receivers are naturally described using the signal processing concepts listed above. For our demonstrator we think that this abstraction level is the right one to choose. Two problems arise: 1/. where do we stop with the

²There is even a little guide with the title "How to find what you need in the Bluetooth Spec", [10].

description of our demonstrator? What function to include, what to leave out? Especially at the bit-side of the receiver (as opposite to the antenna side) a lot of possibilities exist. This levels seems to be too detailed for the problem at hand and in the project phase we are in. 2/. As finding a good SDR receiver implementation is our main task, and intended result of our effort, starting off with an existing radio architecture does not seem to be a good idea. We believe that for our narrow scope objectives at this stage in the project another level of abstraction is needed. For the wide scope objectives, the signal processing abstraction level is a useful one, and the one we will use in the sequel of the project to assess integration challenges and flexibility options.

D. HW/SW Functional Model

With a *Software/Hardware Functional Model* we mean a model in which high level building-blocks like 'NIC' and 'Host Computer' are used. Other implementation-oriented concepts are hardware related concepts like 'COTS', 'GPP', 'DSP', 'FPGA', 'FPGA', 'ASIC', . . . , and software related concepts like 'Kernel', 'Driver', 'OS', 'User Space',

Will a *HW/SW Functional Model* lead to a good SDR functional architecture? Similar to the reasoning in the previous section we think that in this phase of the project this abstraction level is not well chosen for the problems at hand.

For the wide scope objectives, this is abstraction level is a *necessity*: in two aspects. First, we want to assess the costs of implementing a particular signal processing architecture using current day COTS components; secondly we want to predict what can be integrated at what cost into programmable hardware and software in the future.

E. Relation between Models

What is the relation between the presented models above? Leaving an object-oriented approach out of our discussion, one can argue that a layered protocol model specifies the communication system and that a signal processing functional model and HW/SW functional model are intermediate steps in implementing it.

A *Layered Protocol Architecture* (or *Protocol Architecture*) defines functional entities, their tasks and their behaviour. The mapping of these functions to a *Signal Processing Functional Architecture* and *Software/Hardware Functional Architecture* is completely left open (and, in any case in a design process deferred

to a later moment in time). A graphical interpretation is given in Figure 4.

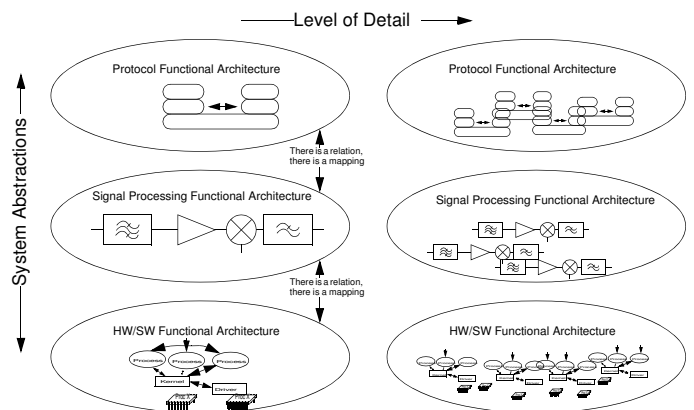


Fig. 4. Architectures: Abstractions and Details.

A systems description can be given using a particular systems-abstraction at a well-chosen level of detail in any phase of the design process. So, also the identification of integration challenges and flexibility options can be investigated using different abstractions and levels of detail. In our project we will pursue our wide scope objectives using signal processing concepts and HW/SW concepts. The possibility of identifying flexibility on different architectural levels (using different architectural concepts) will *not* be attempted. For the narrow scope of our architectural endeavor we still need the right level of abstraction: The Signal Processing Functional Architecture and HW/SW Functional Architecture are considered not sufficient as they are too close to existing implementations. The Layered Protocol Model is considered too involved for specification purposes. Can we find a SDR Functional Model that is useful in this phase of our design process? As a signal processing model and HW/SW model are not applicable in this project phase, it seems that a *simplified* Layered Protocol model is the model to look for.

IV. SIMPLIFIED LAYERED PROTOCOL MODEL

First we present and characterize a functional model for software radio systems that was used before in a software radio project (the SpectrumWare project, [13]). The SpectrumWare project was chosen as the project members designed and implemented an SDR in which the ADC was placed near the antenna and, more interestingly³, *all* the subsequent process-

³More interestingly' when assessing the SpectrumWare project from our perspective. In the SpectrumWare project itself the question where to place the ADC (a focus of *our* project) was not presented as a main research object.

ing was handled in *user space* software. We comment on this functional model and present a new one which we think suitable for our purposes. Before applying it to the relevant 'standards under implementation' we characterize our model: what is it that we actually model (and what is it that we do not model)?

A. The Bose Model

The thesis of Bose [4] provides us with something he coined a *Software Radio Layering Model* that relates the OSI layer-based modelling to functions in the signal path of a software radio system, see Figure 5. Both the sender functionality (top-down) and receiver functionality (bottom-up) are shown. In order to validate the model, he applied it to the IEEE802.11 Wireless LAN (WLAN) specification and the GSM cellular phone specification. Bose (Chapter 5) distinguishes the following functions:

1. Link Framing

Determination of beginning and end of frames, e.g. for error handling functions. "The traditional link layer is preserved in this model", [4, p. 72].

2. Medium Access Control (MAC)

Mediate shared medium access and implement collision avoidance.

3. Coding

All coding necessary between bits and symbols. Two types of coding are discerned: Channel Coding, dealing with error handling; and Line Coding, dealing with spectrum control, e.g. removal of baseline drift or undesirable symbol correlations.

4. Modulation

Transformation between symbols and (baseband) signals.

5. Multiple Access

Sharing of spectrum between different networks and isolation of single network spectrum. Examples: TDMA, FDMA, CDMA, FH-SS and DS-SS.

6. A/D&D/A Conversion

7. Frequency Conversion

Conversion from IF signals to RF signals.

Bose's identification of OSI layers with functionality seems to be confused, especially as there is a mismatch between text and figure. Applying OSI-type of modelling, one has to distinguish two Data Link Layer (DLL) types: a DLL in a point-to-point link between two computers and a DLL in a shared-medium network, in which multiple stations are attached to one physical medium.

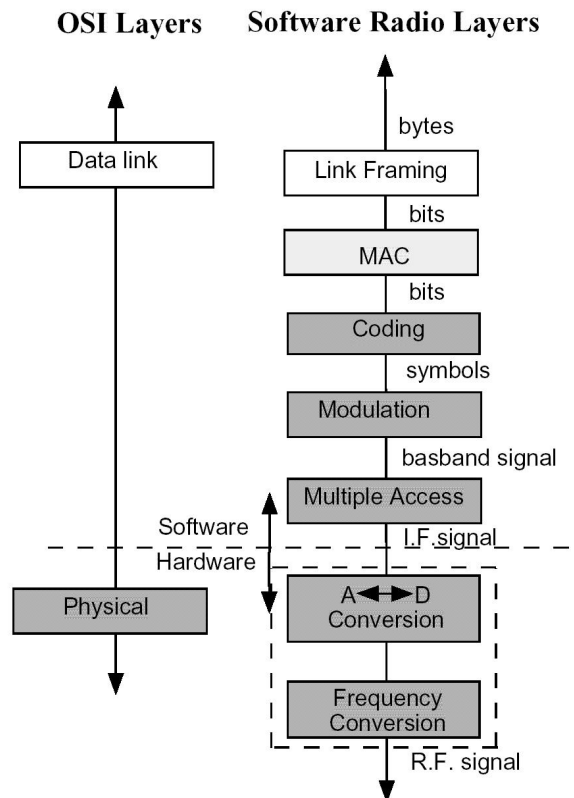


Fig. 5. Software Radio Layering Model (taken from [4]).

In the point-to-point situation the DLL functionality provides *reliable* communication⁴. Protocol Functions of the DLL are: Service Primitive Handling⁵, Error Handling (by application of error correction techniques or by error detection techniques and Automatic Repeat reQuest (ARQ) algorithms), Flow Control and Framing.

In the shared-medium context the DLL not only provides reliable communication but also has to mediate access to the medium. In WLANs and Personal Area networks (PANs) *this* context is applicable (and not the one suggested in "The traditional link layer is preserved in this model" [4, p. 72]). In the shared-medium case the DLL is split up into two sublayers: the Logical Link Control Layer (LLC Layer) and the Medium Access Control Layer (MAC Layer). Proto-

⁴"The Data Link Layer detects and possibly corrects errors which may occur in the Physical Layer" and "The Data Link layer enables the network layer to control the interconnection of data-circuits within the Physical Layer", direct quotes from [8].

⁵A 'Service Primitive' can be considered to be a procedure or function that is (locally) executed between two fully cooperative functional entities *and* has a remote consequence (the execution of a procedure or function elsewhere). One may think of a C-programming-language function, executed on one computer that (miraculously as though it may seem) leads to the execution of another function on a remote computer.

col functions of the LLC layer are Service Primitive Handling, Flow Control and ARQ. Protocol functions of the MAC layer are Framing, Error Detection (CRC-check), Addressing and Access Control, [12].

In case the Link Layer provides a Connection Oriented (CO) service, the LLC entities set up the connection, execute flow control and ARQ during the data phase of the communication and terminate the connection. The sending MAC entity assembles data into frames, adds address fields and CRC fields. Subsequently the Medium Access Control Mechanism is executed (ALOHA, Slotted ALOHA, CSMA/CD, CSMA/CA, ...). The receiving MAC entity disassembles the frame, performs address recognition (checks the address) and executes the error detection algorithm (performs the CRC). A successfully transmitted MAC frame is passed to the LLC entity at the receiving side.

Below the MAC layer, OSI models the functionality as belonging to the Physical (PHY) Layer⁶. The main function of the PHY layer is the creation of signals out of bits (sending side) and bits out of signals (receiving side); so, as transmission-engineers would call it: the Modulation and Demodulation function. Functions as Line Coding and Equalization are also placed here. Another function placed in the physical layer is the function that enables radio spectrum resource sharing and physical channel isolation: a Multiplexing and Demultiplexing function or Channelizer and Channel Selection function (e.g. TDMA, FDMA, CDMA, ...). Bose calls it 'Multiple Access' function (see figure 5). At the Service Access Points (SAPs) between PHY Layer and MAC Layer blocks of (unreliable) bits are the data units that are exchanged, *not* IF Signals (as suggested in Figure 5).

Bose also must have had problems with his model as in *earlier* articles (see articles for INFOCOM [3] and IEEE JSAC [2]) a somewhat different model from the one in his thesis is used, see Figure 6.

In this figure, instead of the MAC sublayer a 'Channel Coding' function is presented as DL layer function. Moreover, instead of the (general notion) 'Coding' function in Figure 6 a 'Line Coding' function is shown. Other functional blocks are identical. However, while in Figure 5 (thesis) the PHY layer - DL layer boundary is unclear (and coincides with the HardWare/SoftWare (HW/SW) boundary), it is clearly and correctly represented in Figure 6 and,

⁶"The Physical Layer provides [.. i.m.h.o. irrelevant stuff deleted ..] physical connections for bit transmission between Data Link entities", direct quote from [8].

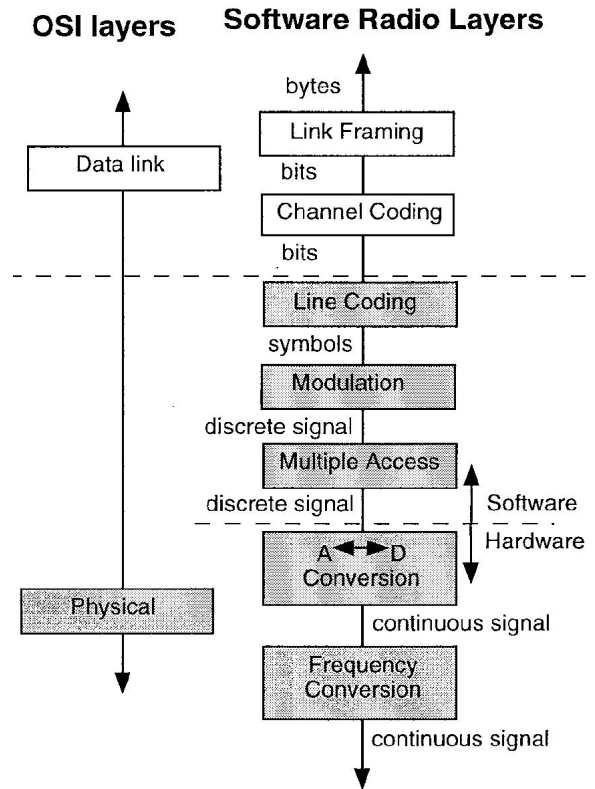


Fig. 6. Software Radio Layering Model (taken from [3], also used in [2]).

moreover, separated from the HW/SW boundary. In the latter figure however, the function 'Channel Coding' clearly under-defines the capabilities of the MAC layer. In both figures 'Link Framing' is not a sufficient description for the LLC functionality.

Bose is interested in a *software architecture* and stops detailing the analog part of the system. Moreover, he places the A/D&D/A converters between IF and RF handling systems. For our project the placement of A/D&D/A systems *is* of importance and a *subject of research*, so we need to identify functional alternatives for placement of the A/D&D/A systems. Also, in a functional model in which we specify what tasks a system has to execute, *we* do not choose to position an A/D&D/A system - it may more hinder our design effort than help it.

In the figures mentioned above we also see the data-units that are transported (exchanged) between the functional blocks. These signals are denoted with a generic name (in Figure 6 e.g. discrete signal, continuous signal, bits, symbols). We will adopt this choice for a generic name as a precise definition of Service

Data Units (SDUs) exchanged between protocol entities is not required for our purposes. What *is* necessary however, is the definition of Reference Points that delimit functionality. These reference points can be used for the specification of interfaces between functions and for the specification of signal requirements (e.g. for a specific test-signal scenario we require a Signal to Distortion Ratio (SDR ⁷) of x dB). Reference points are *not* meant to hamper implementation, but to clarify communication over system aspects.

B. The Simple LP Model

A first version of our functional model (a simple Layered Protocol (LP) model) that repairs the problems sketched above is given in Figure 7. It is a receiver-only model, as the receiver is the focus of our project. The *character* of the model however is equal to the one of Bose and will be discussed in §IV-C.

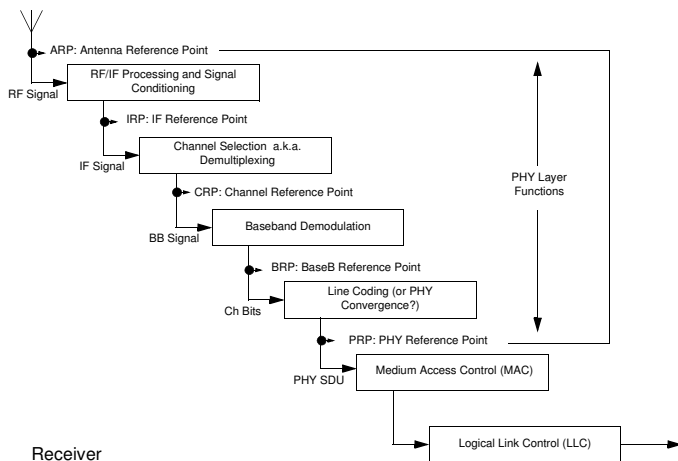


Fig. 7. Simple LP Model - First Version.

A point of discussion was whether to define the IF reference point in this model or not. On the functional level we *want* to use at this stage in the project and for the objectives we *aim at obtaining*, we do not *need* the IF reference point. What functions *do* we need? We do need the (de)modulation function. A modulation function maps symbols (represented by blocks of bits) to a piece of signal (signal element), a demodulator does the reverse⁸. It is possible to deny the existence or necessity of a channelization/channel

⁷Pun intended.

⁸There may be somewhat confusion as to what a *symbol* actually is. Is a 'symbol' a character chosen out of an alphabet of 2^N characters $\alpha_1, \dots, \alpha_{2^N}$ (represented by N bits), or is it used to denote one out of the 2^N signal elements (pieces of signal; signal element $s_i(t)$ is associated with symbol α_i)? For now we assume that the difference will become clear in the context the word is used.

select function, as the channel choice is, at a *signal-processing-functions level*, easily described as a filter operation: in a matched filter receiver, $H(j\omega)$ is the *only* function before the sampler and quantizer *and* an integral part of a demodulator. However, life is not so simple. In our project we will focus on designing the receiver front end and, by consequence, the IF/RF architecture is a focal point. We need a Channel Reference Point (CRP) in order to understand the requirements for the system part that selects and conditions the signal for (de)modulation. Moreover, at a more principle level, in the "ether" there are signals not intended for the particular receiver we are building and these signals should not be received. So, a channellizing/channel select function is necessary in our model.

In a subsequent stage of our design-process we can, for example, adapt some of the so called RF/IF types of an SDR Forum contribution of [9] and asses the efficiency and costs of the channel-selection function when implemented using a particular signal processing architecture. So, the exclusion of the IF reference point does not obstruct the design process in any way. We decided to leave it out.

Between Baseband reference Point (BRP) and Physical Reference Point (PRP) in Figure 7 we defined a 'Line Coding' or 'PHY Convergence?' function. This is rather vague⁹. The problem is in the meaning of 'Line Coding'. In Figure 7 we followed OSI's lead in that Error Handling is a Data Link function. However, if one inspects the HIPERLAN/2 PHY layer standard [6], one sees that both line coding and Forward Error Correction (FEC) coding are applied to a "PDU train from DLC" (see [6], Figure 1). As layering (and modelling for that part) is a slave and not the master, we should adapt. The other side of adaptation is in using the term 'PHY Convergence' which is more or less a garbage-can in which any function operating on symbols and bits can be placed. We propose to follow Bose's thesis approach (in Figure 5) and settle for a 'Coding' function: a mapping of a symbol stream to a bitstream, which we will leave unspecified at this moment.

A remark on the level of detail of functionality. Our project is about the radio front end, so, any function more or less "farther away" from this front end can be modelled/described at a less detailed level. We think therefore that 'MAC Function' and 'LLC func-

⁹In a receiver, of cours, *de*-coding is appropriate.

tion' provide the right description of these functions (in a language the implementers of these functions understand). More detail is not required for our purposes¹⁰.

In our model we do not want to specify in detail what kind of information/data (type of signals, meaning of bits, size of Service Data Units) is transferred from one functional block to another. We intentionally talk loosely about X -signal at Y -reference point, see Table II.

TABLE II
REFERENCE POINTS AND DATA EXCHANGE

Reference Point (RP)	Service Data Unit (SDU) Stream
Antenna RP (ARP)	RF Signal
Channel RP (CRP)	Channel Signal
Demodulation RP (DRP)	Symbol Stream
PHY RP (PRP)	PHY SDU Stream
MAC RP (MRP)	MAC SDU Stream

The final form of our model, in which the points of critique mentioned above are addressed, is presented in Figure 8.

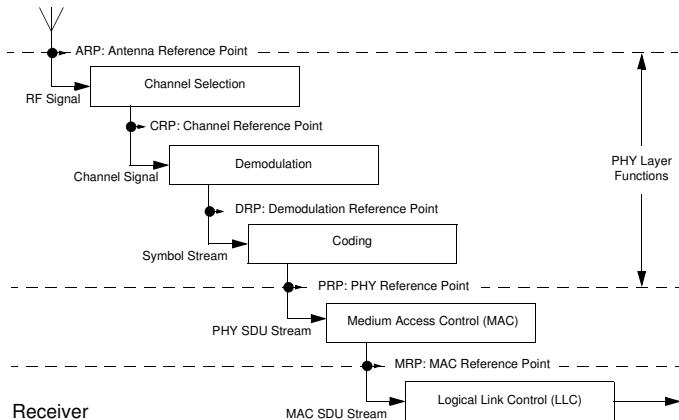


Fig. 8. Simple LP Model - Second Version (= SPFM).

C. Functional Model or Signal Processing Functions Model?

Can we call our model in Figure 8 a functional model? What is the character of this model?

Both in Bose's model and in the Simple LP Model no distinction is made between User Plane Function-

¹⁰There may be a problem here, as we do not intend to research implementation of MAC functions and LLC functions. However, if we want to test our front end and display test-results, we may need a system like a 'MAC Controller' or 'Link Handler' as part of demonstrator.

ality (for the transport of user data), Control Plane Functionality (for the set-up, modification and release of connections) and Management Functionality (for e.g. network start-up, network configuration, network provisioning, network monitoring and fault handling). Moreover, it appears to be valid in the user-data transport-phase only. When applying the models (Bose and Simple PL) all Data Link Connections and Physical Layer Channels are assumed to be set-up! The models show the signal-path processing functions and information mappings for data-link frames¹¹ interpreted as a single stream. So, basically, what we have here is a type of model we will call a *Signal Path Functions Model*.

Below we will investigate the usability and usefulness of the *Signal Path Functions Model* (SPFM, the new interpretation of the Simple LP Model¹²) by deriving model instances for the HiperLAN/2 and Bluetooth standards. So, in our project we refrain from creating a Layered Protocol Model for both standards and focus on the Signal Path functions instead.

The SPFM is believed to fulfill our narrow scope objective in the architectural phase of the project - we are able to specify what is going to be build (e.g. a system that implements the model from ARP to DRP, see Figure 8) and to specify channel selection requirements (the filter requirements between ARP and CRP). In as far as the wide scope objectives are concerned, opportunities are left un-used here, as not the entire protocol stack is analyzed and modelled. For physical layer functionality however, the question remains whether the SPFM is helpful in achieving the wider scope objectives. An example using Bluetooth and HiperLAN/2 may clarify this issue.

V. HIPERLAN/2 AND BLUETOOTH SPFM

In Figure 9 and Figure 10 at the end of this paper we present the SPFMs for HiperLAN/2 and Bluetooth. The demodulation function is detailed into three functional blocks: transformation, decision and mapping to accommodate the OFDM demodulator in HiperLAN/2. The mapping function is a transformation from I/Q symbols to blocks of bits and rather straightforward. The transformation block for Bluetooth (in Figure 10) is a IF-to-BB transformation. In

¹¹The highest level of Service Data Units used in the SPFM are MAC SDUs (MAC frames).

¹²and the term we will use below and in the sequel to refer to it.

Figure 11 the integrated SPFM is shown: the transformation block has become a modulation-scheme specific transformation. Observation of the figure leads naturally to the question whether it is useful to implement the transformation in the Bluetooth system using the HiperLAN/2 infrastructure.

As stated before, the narrow scope objectives can be achieved using an SPFM. The wide scope objectives are achieved only in a limited way and when detailing the model further it appears to be more or less artificial. It is clear that this is the moment to leave the SPFM abstraction level and switch to the Signal Processing Model and HW/SW model. In the sequel of the project we will focus on the channel selection function, the transformation function and the decision function of both standards using signal processing and HW/SW related concepts.

VI. CONCLUSIONS

In this paper we presented our project and formulated the objectives for the architectural phase for our Bluetooth-HiperLAN/2 testbed. We distinguished two types of objectives: narrow-scope objectives mainly pertaining to the design of test bed and wide-scope objectives that deal with identification of integration and flexibility opportunities. The question was addressed what type of model to use for satisfying these objectives.

As an object-oriented software approach is not a subject of research in our project, the SDR Forum Model was rejected. A Layered Protocol Model proved too involved to arrive at within the time-frame of the project and is a form of overkill for the narrow-scope objectives at hand. For achieving the wide-scope objectives this type of modelling is believed to be promising. Also, it may be an interesting option to compare an object-oriented model with a Layered Protocol Model.

As the Signal Processing Model and HW/SW Model are too detailed for our narrow scope objectives we decided to derive a simplified Layered Protocol Model and arrived at the Signal Path Functions Model (SPFM) in Figure 8. This model satisfies the narrow scope objectives of our project rather well. For the wide scope objectives however, the SPFM proved to have only limited capabilities and appears to be artificial when trying to detail it further.

The project will continue by using Signal processing and HW/SW models and concepts in order to satisfy the wide scope objectives for functions identified using the SPFM.

ACKNOWLEDGMENTS

This research is carried out in cooperation with the Integrated Circuit Design Laboratory of the University of Twente. We would like to thank Vincent Arkesteijn, Erik Klumperink and Bram Nauta for their valued and appreciated contributions. This research is supported by the PROGram for Research on Embedded Systems & Software (PROGRESS) of the Dutch organization for Scientific Research NWO, the Dutch Ministry of Economic Affairs and the technology foundation STW.

REFERENCES

- [1] C.H. Slump et al. B. Nauta. Progress proposal: Development of a software-radio based embedded mobile terminal. PROGRESS, 1999.
- [2] V. Bose, M. Ismert, M. Wellborn, and J. Guttag. Virtual radios. *IEEE JSAC*, 17(4):591–602, April 1999.
- [3] V. Bose, A.B. Shah, and M. Ismert. Software radios for wireless networking. *IEEE Infocom, San Francisco*, pages 1030–1036, March/April 1998. http://www.ieee-infocom.org/1998/papers/08d_1.pdf.
- [4] Vanu G. Bose. *Design and Implementation of Software Radios Using a General Purpose Processor*. PhD thesis, Massachusetts Institute of Technology, June 1999.
- [5] ETSI. Broadband radio access networks (bran); hiperlan type 2; system overview. Technical Report ETSI TR 101 683 V1.1.1 (2000-02), ETSI, 650 Route des Lucioles - Sophia Antipolis, Valbonne - FRANCE, February 2000.
- [6] ETSI. Broadband radio access networks (bran); hiperlan type 2; physical (phy) layer. Technical Specification ETSI TS 101 475 V1.2.2 (2001-02), ETSI, 650 Route des Lucioles - Sophia Antipolis, Valbonne - FRANCE, February 2001.
- [7] Joseph Mitola. *Software Radio Architecture*. John Wiley & Sons, 2000.
- [8] OSI.84. Information processing systems. open systems interconnection - basic reference model. International Standard IS 7498, ISO, 1984.
- [9] Jack Rosa. Rf/IF subsystem of a commercial sdr base station. SDR Forum Contribution SDRF-01-I-0012-V0.00, SDR Forum, January 2001. http://www.sdrforum.org/public/i_0022_v0_00_rf_ifsystems_jrosa_04_11_01.pdf.
- [10] Tom Slep. *How to Find What You Need in the Bluetooth Spec*. IEEE Press, 2000.
- [11] Bluetooth SIG. Specification of the bluetooth system - core. Technical Specification Version 1.1, Bluetooth SIG, February 2001.
- [12] William Stallings. *Local Area network Standards*, volume 2 of *Handbook of Computer Communications standards*. Macmillan Publishing Company, 1990.
- [13] D.L. Tennenhouse, T. Turlitti, and V.G. Bose. The spectrumware testbed for atm-based software radios. *IEEE International Conference on Universal Personal Communications*, September/October 1996. <http://tns-www.lcs.mit.edu/publications/icupc96.ps.gz>.
- [14] SDR Forum Website. Sdr primer. http://www.sdrforum.org/sdr_primer.html, 2000.

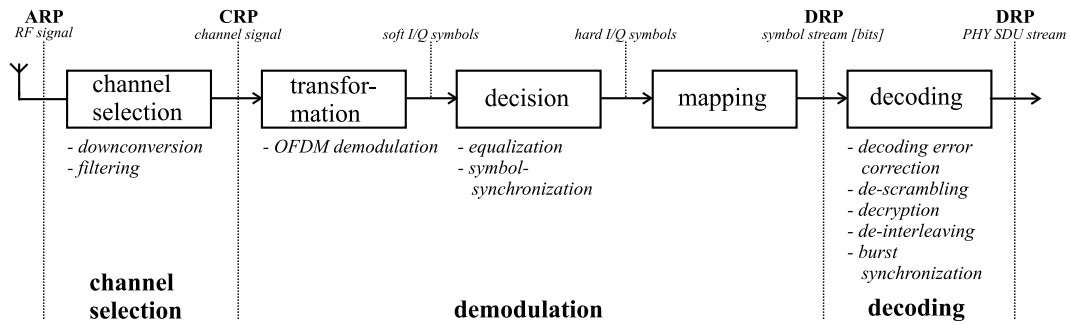


Fig. 9. HiperLAN/2 SPFM.

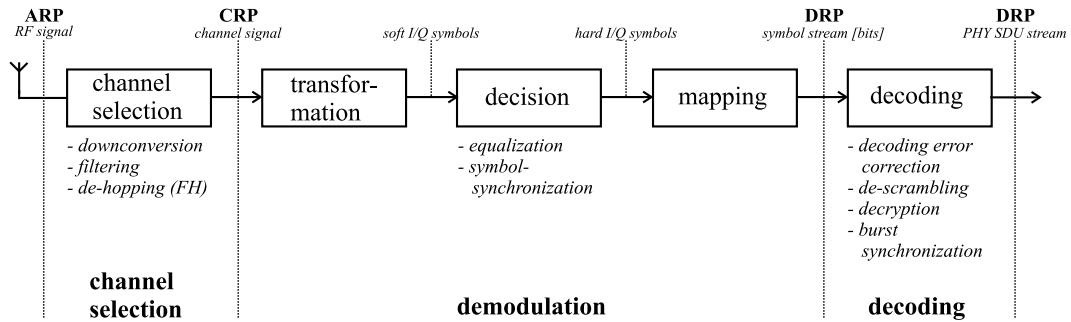


Fig. 10. Bluetooth SPFM.

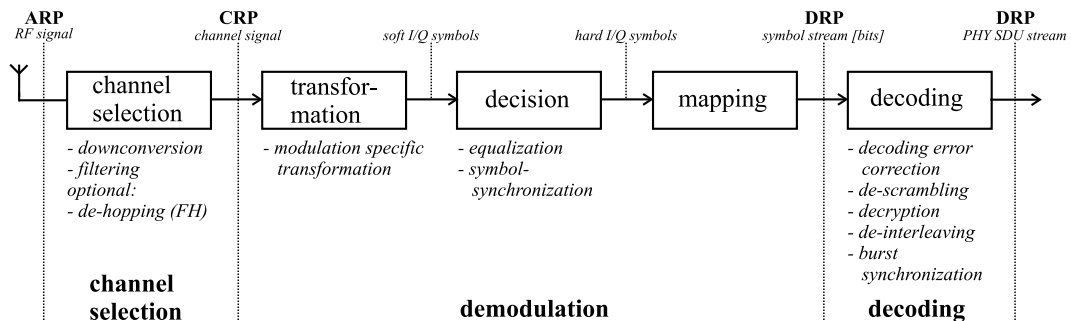


Fig. 11. Integrated SPFM.