Technical University Berlin

Telecommunication Networks Group

PHY-MAC Interface Definitions for Dynamic OFDMA Systems

Mathias Bohge, James Gross, Adam Wolisz

{bohge,gross}@tkn.tu-berlin.de, awo@ieee.org

Berlin, September 2007

TKN Technical Report TKN-07-003

TKN Technical Reports Series

Editor: Prof. Dr.-Ing. Adam Wolisz

Contents

I. Purpose

Recently, numerous research groups have undertaken major efforts to develop experimental platforms that support the physical layer (PHY) of OFDMA systems. Independently, significant research is being done on resource optimization and other medium access control layer (MAC) related issues. The two communities, however, develop their solutions mostly in isolation.

In order to ease the merging of PHY and MAC developments, as well as in order to make experimenting with different PHY/MAC combinations possible, we believe that it is necessary to define a clear interface. This document contains a proposal for such an interface. Its structure largely follows the style of the IEEE 802.11 standard description, since this type of interface is well established and has been defined for numerous instances of system solutions.

II. Background

Orthogonal Frequency Division Multiplexing (OFDM) is a multi-carrier modulation technique that splits a given system bandwidth into a certain number of sub-channels, on which data symbols can be transmitted simultaneously [1]. Since several years, OFDM is being successfully deployed to wired transmission systems (e.g. the Digital Subscriber Line (DSL) [2]), as well as to wireless transmission systems (e.g. as the physical basis of the Terrestrial Digital Video Broadcasting (DVB-T) service [3], the Wireless LAN standards IEEE 802.11a/g [4], and the wireless broadband access technique WiMax [5]). Moreover, Orthogonal Frequency Division Multiple Access (OFDMA) has been chosen for the downlink of the Long Term Evolution (LTE) system concept currently standardized in 3GPP. Finally, OFDMA is considered to be used for 4th generation cellular networks.

Recently, the concept of dynamic OFDMA systems has received a lot of scientific attention [6],[7] as it provides large gains in the exploration of the wireless channel's capacity [8]. In dynamic OFDMA systems each terminal associated to a centralized unit of the system (i.e., the access point) receives a varying amount of system resources (sub-carriers and/or transmission power) for data communications. This assignment of resources can be based on various parameters belonging to different layers of the communication stack, such as the states of the sub-carriers, the amount of data queued for each terminal at the access point, and the required quality of service of each flow (or even each packet of a flow). Given these parameters, an algorithm at the access point generates resource assignments, for example for the next transmission phase (i.e., down-link or up-link). As the assignments are typically based on the quality of the sub-carriers, these assignments have to be regenerated periodically, depending on the coherence time of the channel.

As a consequence of the dynamic cross-layer behaviour, diverse information needs to be exchanged between the MAC and the PHY layer of an according system. This document provides a generic interface definition, as well as detailed PHY service specifications..

III.System Definition and Interface Structure

Let us consider a cellular system, with designated roles of base station (BS) and mobile terminals (MT). We will use further on the classical notion of up-link (UL) and down-link (DL). The interface is designed for supporting both TDD (Time Division Duplex), as well as FDD (Frequency Division Duplex) operation. Time is slotted in transmission time intervals (TTI). While the support of OFDMA in both directions has to be assured by the interface structure, the interface has to support also up-link following a different standard of transmission (e.g. single carrier frequency division multiple access – SC-FDMA).

III.1. Interface Functionality

The basic interface functionality consists of three functional blocks:

1. Data transmission functions (DATA),
2. Signalling functions (SIGN),
3. Control functions (CTRL).

In addition, a system depending management interface needs to be provided.

III.1.1. Data Transmission

In dynamic OFDMA systems the MAC explicitly tells the PHY, which data is to be delivered using which resources. In order to enable dynamic enhancements the following information needs to be exchanged once per frame.

a. Sending side

Information exchange per scheduled sub-carrier:

- Modulation / coding type combination (id),
- Transmission power (quantized value or id)

Information exchange per scheduled terminal:

- Sub-carrier assignments (id – might also be assigned in chunks),
- Data block information / data delivery in MACPDUs to PHY.

Miscellaneous necessary information:

- Allocation done – ready for transmission.

b. Receiving side

Data delivery information:

- Sub-carrier assignments for this terminal,
- Modulation / coding type combination (id) per utilized sub-carrier,
- Data block information / data delivery to local MAC.

III.1.2. Signalling

In order to enable the PHY of the receiving entity to recover data conveyed in a received data frame it has to be provided with resource allocation information. The sending side MAC provides this information to the receiving side MAC via a dedicated signalling channel. Using this knowledge the receiving side MAC configures its PHY to receive the subsequent data on specified sub-carriers / sub-carrier groups using the chosen adaptive modulation and coding configuration.

III.1.3. Control

Dynamic resource allocations mechanisms at the MAC require channel state knowledge that needs to be provided by the PHY. Thus, the PHY must provide an interface for channel state acquisition versus another entity. This is done by tracking the power values per sub-carrier / sub-carrier group upon the reception of data frames, as well as by sending and evaluating probe messages.

In addition means to set up general parameters (e.g. centre frequency, bandwidth, cyclic prefix, signalling channel band definition …) at system start-up need to be provided. However, as this is not special to dynamic OFDMA systems, they are out of the scope of this document.

IV. Detailed PHY service specifications

In this chapter the services to be provided by the PHY layer are described as primitives, which do not imply any particular implementation. The primitives associated with communication between the MAC and the PHY layer fall into two basic categories:

1. Service primitives that support MAC peer-to-peer interactions,
2. Service primitives that have local significance and support sublayer-to-sublayer interactions.

The notation used in this chapter follows the IEEE primitive definition rules (as used in e.g. [9]).

IV.1. Basic service and options

All of the service primitives described here are considered mandatory unless otherwise specified. If a primitive is issued by the PHY sublayer of an entity it is directed towards the MAC sublayer. If it is issued by the MAC sublayer it is directed towards the PHY sublayer.

We distinguish between three different kinds of primitives:

1.	Request	:	direction:	MAC → PHY
2.	Indication	:	direction:	PHY → MAC
3.	Confirmation	:	direction:	PHY → MAC

A request issued by a MAC always requires a confirmation by the PHY. An indication issued by the PHY is never confirmed.

IV.1.1. PHY peer-to-peer service primitives

Table 1 indicates the primitives for peer-to-peer interactions.

Table 1: PHY peer-to-peer service primitives

Primitive	Request	Indication	Confirmation
PHY-DATA	X	X	X
PHY-SIGN	X	X	X
PHY-CTRL	X	X	X

IV.1.2. PHY sublayer-to-sublayer service primitives

Table 2 indicates the primitives for sublayer-to-sublayer interactions.

Table 2: PHY sublayer-to-sublayer service primitives

Primitive	Request	Indication	Confirmation
PHY-TX-MACPDU-START	X		X
PHY-TX-MACPDU-END	X		X
PHY-TX-TRANSMIT	X		X
PHY-RX-DATA-START		X	
PHY-RX-DATA-END		X	
PHY-TX-SIGN-START	X		X
PHY-TX-SIGN-END	X		X
PHY-RX-SIGN-START		X	
PHY-RX-SIGN-END		X	
PHY-TX-SETPOW	X		X
PHY-TX-SETAMC	X		X
PHY-RX-SETAMC	X		X

IV.1.3. PHY service primitive parameters

Table 3 shows the parameters used by one or more of the PHY service primitives.

Table 3: PHY service primitive paramters

Parameter	Associated primitive	Value
DATA	PHY-DATA.request PHY-DATA.indication	Octet value X’00’X’FF’
SIGNALLING	PHY-SIGN.request PHY-SIGN.indication	Octet value X’00’X’FF’
USER	PHY-CTRL.request PHY-CTRL.indication	8, 16 or 24 bit value.
POWERVECTOR	PHY-TX-SETPOW.request PHY-CTRL.indication PHY-RX-DATA-END.indication	A vector that contains one entry per sub-carrier or group of sub-carriers.
AMCVECTOR	PHY-TX-SETAMC.request PHY-RX-SETAMC.request	A vector that contains one entry per sub-carrier or group of sub-carriers.
SUBVECTOR	PHY-TX-MACPDU-START.request	A vector that contains one entry per sub-carrier or group of sub-carriers.

IV.1.4. Vector descriptions

Several service primitives include a parameter vector. Table 4 lists the parameter values required by the MAC or PHY in each of the parameter vectors.

Table 4: Vector descriptions

Parameter	Associated vector	Value
AMC_ID	AMCVECTOR	Depending on the number of available AMC combinations a 1,2,3 or 4 bit value.
POWER	POWVECTOR	Double value.
POW_ID	POWVECTOR	Alternative way to indicate the power to apply to a sub-carrier or group of sub-carriers. Depending on the level of quantization a 1,2,3 or 4 bit value.
SUB_ENABLED	SUBVECTOR	1 bit value that enables the use of the according sub-carrier or group of sub-carriers.

IV.2. PHY detailed service specification

The following sub clause describes the services provided by each PHY sublayer primitive.

IV.2.1. Peer-to-peer service primitives

According to the interface structure presented in Section III, the primitives for peer-to-peer to peer interactions are grouped into the three categories:

1. Data transmission peer-to-peer primitives (DATA),
2. Signalling peer-to-peer primitives (SIGN),
3. Control peer-to-peer primitives (CTRL).

IV.2.1.1. Data transmission peer-to-peer service primitives

IV.2.1.1.1. `PHY-DATA.request`

a. Function

This primitive defines the transfer of an octet of data from the MAC sublayer to the local PHY entity.

b. Semantics of the service primitive

The primitive provides the following parameters:

```
PHY-DATA.request(DATA)
```

The `DATA` parameter is an octet of value `X’00’` through `X’FF’`.

c. When generated

The primitive is generated by the MAC sublayer to transfer an octet of data to the PHY entity. This primitive can only be issued following a data-to-sub-carrier-mapping initialization that results from a preceding `PHY-TX-MACPDU-START.request` and `PHY-TX-MACPDU-START.confirm` primitive exchange between MAC and PHY.

d. Effect of receipt

The receipt of this primitive by the PHY entity causes the PHY to distribute the data contained in `DATA` among the buffers that belong to the sub-carriers that have previously been specified by the MAC using the `PHY-TX-MACPDU-START.request(SUBVECTOR)` primitive. Once the PHY entity has received and buffered the octet, it will issue a `PHY-DATA.confirm` to the MAC sublayer.

IV.2.1.1.2. `PHY-DATA.indication`

a. Function

This primitive defines the transfer of an octet of data from the PHY sublayer to the local MAC entity.

b. Semantics of the service primitive

The primitive provides the following parameters:

```
PHY-DATA.indication(DATA)
```

The `DATA` parameter is an octet of value `X’00’` through `X’FF’`.

c. When generated

The primitive is generated by a receiving PHY entity to transfer the received octet of data to the local MAC sublayer.

d. Effect of receipt

The receipt of this primitive by the MAC is unspecified.

IV.2.1.1.3. `PHY-DATA.confirm`

a. Function

This primitive confirms the transfer of data from the MAC to the PHY sublayer.

b. Semantics of the service primitive

The semantics of the primitive are as follows:

```
PHY-DATA.confirm
```

This primitive has no parameters.

c. When generated

The PHY sublayer will issue this primitive in response to every `PHY-DATA.request` primitive issued by the MAC layer, whenever the transfer of an octet has been completed.

d. Effect of receipt

The receipt of this primitive by the MAC entity causes the MAC to start the next MAC entity request.

IV.2.1.2. Signalling peer-to-peer service primitives

IV.2.1.2.1. `PHY-SIGN.request`

a. Function

This primitive defines the transfer of an octet of signalling information from the MAC sublayer to the local PHY entity.

b. Semantics of the service primitive

The primitive provides the following parameters:

```
PHY-SIGN.request(SIGNALLING)
```

The `SIGNALLING` parameter is an octet of value `X’00’` through `X’FF’`.

c. When generated

The primitive is generated by the MAC sublayer to transfer an octet of signalling information to the PHY entity. This primitive can only be issued following a signalling initialization primitive exchange using `PHY-TX-SIGN-START.request` and `PHY-TX-SIGN-START.confirm` between MAC and PHY layer.

d. Effect of receipt

The receipt of this primitive by the PHY entity causes the PHY to distribute the contents of `SIGNALLING` among the sub-carrier buffers according to a predefined modulation and coding configuration that is valid for signalling information only. Once the PHY entity has received and buffered the octet, it will issue a `PHY-SIGN.confirm` to the MAC sublayer.

IV.2.1.2.2. `PHY-SIGN.indication`

a. Function

This primitive defines the transfer of an octet of signalling information from the PHY sublayer to the local MAC entity.

b. Semantics of the service primitive

The primitive provides the following parameters:

```
PHY-SIGN.indication(SIGNALLING)
```

The `SIGNALLING` parameter is an octet of value `X’00’` through `X’FF’`.

c. When generated

The primitive is generated by a receiving PHY entity to transfer the received octet of data to the local MAC sublayer.

d. Effect of receipt

The receipt of this primitive by the MAC entity causes the MAC to buffer the delivered signalling data for subsequent PHY layer configuration.

IV.2.1.2.3. `PHY-SIGN.confirm`

a. Function

This primitive confirms the transfer of signalling information from the MAC to the PHY sublayer.

b. Semantics of the service primitive

The semantics of the primitive are as follows:

```
PHY-SIGN.confirm
```

This primitive has no parameters.

c. When generated

The PHY sublayer will issue this primitive in response to every `PHY-SIGN.request` primitive issued by the MAC layer, whenever the transfer of an octet has been completed.

d. Effect of receipt

The receipt of this primitive by the MAC entity causes the MAC to start the next MAC entity request.

IV.2.1.3. Control peer-to-peer service primitives

IV.2.1.3.1. PHY-CTRL.request

a. Function

This primitive is a request for channel state acquisition to be used by the MAC sublayer for dynamic resource allocation mechanisms.

b. Semantics of the service primitive

The primitive provides the following parameters:

```
PHY-CTRL.request(USER)
```

The `USER` parameter is an 8, 16 or 24-bit value that represents a user ID.

c. When generated

The primitive is generated by the MAC sublayer to enable the entity that holds the ID `USER` to acquire channel state knowledge.

d. Effect of receipt

The receipt of this primitive by the PHY entity causes the PHY to send a probe message to the specified entity. Once the probe has been sent, the PHY confirms the sending of the probe by issuing the `PHY-CTRL.confirm` primitive.

IV.2.1.3.2. PHY-CTRL.indication

a. Function

This primitive indicates the reception of a probe message.

b. Semantics of the service primitive

The primitive provides the following parameters:

```
PHY-CTRL.indication(POWVECTOR)
```

The `POWVECTOR` parameter is a vector that contains one value per sub-carrier / sub-carrier group.

c. When generated

The primitive is generated by a receiving PHY each time it has received a valid start frame delimiter (SFD) and probe message header, if the header error check has successfully been accomplished.

d. Effect of receipt

The receipt of this primitive by the MAC entity causes the MAC to update its local channel-state vector by utilizing the power values contained in `POWVECTOR`.

IV.2.1.3.3. `PHY-CTRL.confirm`

a. Function

This primitive confirms the transmission of a probe message via the air interface.

b. Semantics of the service primitive

The semantics of the primitive are as follows:

```
PHY-CTRL.confirm
```

This primitive has no parameters.

c. When generated

The PHY sublayer will issue this primitive in response to every `PHY-CTRL.request` primitive issued by the MAC layer when the transmission of the probe message has been completed.

d. Effect of receipt

The receipt of this primitive by the MAC is unspecified.

IV.2.2. Sublayer-to-sublayer service primitives

According to the interface structure presented in Section III, the primitives for sublayer-to-sublayer interactions are grouped into the three categories:

1. Data transmission sublayer-to-sublayer primitives (DATA),
2. Signalling sublayer-to-sublayer primitives (SIGN),
3. Control sublayer-to-sublayer primitives (CTRL).

Note that we differentiate between the primitives that are effective at the transmitter side and those that are effective at the receiver side. Transmitter side primitives contain the token ‘`TX`’, whereas receiver side primitives contain the token ‘`RX`’ in their names.

IV.2.2.1. Data transmission sublayer-to-sublayer service primitives

IV.2.2.1.1. `PHY-TX-MACPDU-START.request`

a. Function

This primitive marks the start of the delivery of a MACPDU and indicates the sub-carriers on which it is to be transmitted.

b. Semantics of the service primitive

The primitive provides the following parameters:

```
PHY-TX-MACPDU-START.request(SUBVECTOR)
```

The `SUBVECTOR` parameter is a vector that contains one value per sub-carrier / sub-carrier group.

c. When generated

The MAC sublayer will issue this primitive after having received the `PHY-TX-SETAMC.confirm` primitive by the PHY layer.

d. Effect of receipt

The receipt of this primitive by the PHY entity causes the PHY prepare for data delivery and enables it to map the data of the subsequently following `PHY-DATA.request` to the appropriate sub-carriers / sub-carrier groups. Once the PHY entity has prepared for MACPDU delivery, it will issue a `PHY-TX-MACPDU-START.confirm` to the MAC sublayer.

IV.2.2.1.2. `PHY-TX-MACPDU-START.confirm`

a. Function

This primitive confirms the beginning of a MACPDU data delivery.

b. Semantics of the service primitive

The semantics of the primitive are as follows:

```
PHY-TX-MACDPU-START.confirm
```

This primitive has no parameters.

c. When generated

The PHY sublayer will issue this primitive in response to every `PHY-TX-MACPDU-START.request` primitive issued by the MAC layer, whenever it has prepared for data delivery.

d. Effect of receipt

The receipt of this primitive by the MAC entity causes the MAC to issue the `PHY-DATA.request` primitive.

IV.2.2.1.3. `PHY-TX-MACPDU-END.request`

a. Function

This primitive marks the end of the delivery of a MACPDU.

b. Semantics of the service primitive

The semantics of the primitive are as follows:

```
PHY-TX-MACDPU-END.request
```

This primitive has no parameters.

c. When generated

The MAC sublayer will issue this primitive after having received the `PHY-DATA.confirm` primitive by the PHY layer, which confirms the delivery of the final data octet of the current MACPDU.

d. Effect of receipt

The receipt of this primitive by the PHY entity causes the PHY prepare to issue a `PHY-TX-MACPDU-END.confirm` to the MAC sublayer.

IV.2.2.1.4. `PHY-TX-MACPDU-END.confirm`

a. Function

This primitive confirms the end of a MACPDU data delivery.

b. Semantics of the service primitive

The semantics of the primitive are as follows:

```
PHY-TX-MACDPU-END.confirm
```

This primitive has no parameters.

c. When generated

The PHY sublayer will issue this primitive in response to every `PHY-TX-MACPDU-END.request` primitive issued by the MAC layer.

d. Effect of receipt

The receipt of this primitive by the MAC entity causes the MAC to issue either a `PHY-TX-MACPDU-START.request` primitive for the delivery of the next MACPDU or to issue the `PHY-TX-START.request` to start the OFDMA data transmission over the air interface, in case there are no more MACPDUs to be delivered to the PHY in this TTI.

IV.2.2.1.5. `PHY-TX-TRANSMIT.request`

a. Function

This primitive is a request to start the transmission via the air interface.

b. Semantics of the service primitive

The semantics of the primitive are as follows:

```
PHY-TX-TRANSMIT.request
```

This primitive has no parameters.

c. When generated

The MAC sublayer will issue this primitive after having received the `PHY-TX-MACPDU-END.confirm` primitive by the PHY layer, which confirms the delivery of the final data octet of the last MACPDU that is to be delivered in this TTI.

d. Effect of receipt

The receipt of this primitive by the PHY entity causes the PHY to create OFDM symbols form the data in the sub-carrier buffers and send them over the air interface. Once the transmission has been finished it issues a `PHY-TX-TRANSMIT.confirm` to the MAC sublayer.

IV.2.2.1.6. `PHY-TX-TRANSMIT.confirm`

a. Function

This primitive confirms the transmission of data via the air interface.

b. Semantics of the service primitive

The semantics of the primitive are as follows:

```
PHY-TX-TRANSMIT.confirm
```

This primitive has no parameters.

c. When generated

The PHY sublayer will issue this primitive in response to every `PHY-TX-TRANSMIT.request` primitive issued by the MAC layer, whenever the creation and transmission of appropriate OFDM symbols via the air interface has been completed.

d. Effect of receipt

The receipt of this primitive by the MAC entity causes the MAC to prepare for the next TTI.

IV.2.2.1.7. PHY-RX-DATA-START.indication

a. Function

This primitive indicates that data has arrived at the PHY sublayer.

b. Semantics of the service primitive

The semantics of the primitive are as follows:

```
PHY-RX-DATA-START.indication
```

This primitive has no parameters.

c. When generated

The primitive is generated and sent to the local MAC entity by a receiving PHY each time that it has received a valid start frame delimiter (SFD) and data header, if the header error check has successfully been accomplished.

d. Effect of receipt

The receipt of this primitive by the MAC entity causes the MAC to configure for data reception.

IV.2.2.1.8. PHY-RX-DATA-END.indication

a. Function

This primitive indicates that the data currently being received is complete and delivers actual channel state values to the MAC.

b. Semantics of the service primitive

The primitive provides the following parameters:

```
PHY-RX-DATA-END.indication(POWVECTOR)
```

The `POWVECTOR` parameter is a vector that contains one value per sub-carrier / sub-carrier group.

c. When generated

The primitive is generated and sent to the local MAC entity by a receiving PHY each time that it has received a valid end-of-frame delimiter (EFD).

d. Effect of receipt

The receipt of this primitive by the MAC entity causes the MAC entity to pass on the delivered data to higher layers, and update its local channel-state vector by utilizing the power values contained in `POWVECTOR`.

IV.2.2.2. Data transmission sublayer-to-sublayer service primitives

IV.2.2.2.1. `PHY-TX-SIGN-START.request`

a. Function

This primitive is a request by the MAC sublayer to the local PHY entity to hold itself ready for signalling information delivery.

b. Semantics of the service primitive

The semantics of the primitive are as follows:

```
PHY-TX-SIGN-START.request
```

This primitive has no parameters.

c. When generated

The MAC sublayer will issue this primitive once it has decided about how to allocate the resources among the system entities during the subsequent TTI. It must be issued before the first `PHY-SIGN.request` primitive is issued.

d. Effect of receipt

The receipt of this primitive by the PHY entity causes the PHY to prepare for signalling information transmission. Once the PHY entity is prepared for signalling information transmission, it will issue a `PHY-TX-SIGN-START.confirm` to the MAC sublayer.

IV.2.2.2.2. `PHY-TX-SIGN-START.confirm`

a. Function

This primitive confirms that the delivery of signalling information is going to start.

b. Semantics of the service primitive

The semantics of the primitive are as follows:

```
PHY-TX-SIGN-START.confirm
```

This primitive has no parameters.

c. When generated

The PHY sublayer will issue this primitive in response to every `PHY-TX-SIGN-START.request` primitive issued by the MAC layer, whenever it is prepared for signalling information delivery.

d. Effect of receipt

The receipt of this primitive by the MAC entity causes the MAC to start issuing the `PHY-SIGN.request` primitive.

IV.2.2.2.3. `PHY-TX-SIGN-END.request`

a. Function

This primitive marks the end of signalling information delivery for this TTI.

b. Semantics of the service primitive

The semantics of the primitive are as follows:

```
PHY-TX-SIGN-END.requets
```

This primitive has no parameters.

c. When generated

The MAC sublayer will issue this primitive after having issued the last `PHY-SIGN.request` primitive in the corresponding TTI.

d. Effect of receipt

The receipt of this primitive by the PHY entity causes the PHY to create OFDM symbols from the contents in the sub-carrier buffers and transmit the symbols via the air interface. Once the PHY entity has sent the signalling information, it will issue a `PHY-TX-SIGN-END.confirm` to the MAC sublayer.

IV.2.2.2.4. `PHY-TX-SIGN-END.confirm`

a. Function

This primitive confirms that the delivery of signalling information has been finished.

b. Semantics of the service primitive

The semantics of the primitive are as follows:

```
PHY-TX-SIGN-END.confirm
```

This primitive has no parameters.

c. When generated

The PHY sublayer will issue this primitive in response to every `PHY-TX-SIGN-END.request` primitive issued by the MAC layer, whenever OFDM symbol creation and transmission has been finished.

d. Effect of receipt

The receipt of this primitive by the MAC entity causes the MAC to start issuing the `PHY-TX-SETPOW.request` primitive.

IV.2.2.2.5. `PHY-RX-SIGN-START.indication`

a. Function

This primitive indicates that signalling information has arrived at the PHY sublayer.

b. Semantics of the service primitive

The semantics of the primitive are as follows:

```
PHY-RX-SIGN-START.indication
```

This primitive has no parameters.

c. When generated

The primitive is generated and sent to the local MAC entity by a receiving PHY each time that it has received a valid start frame delimiter (SFD) and signalling data header, if the header error check has successfully been accomplished.

d. Effect of receipt

The receipt of this primitive by the MAC entity causes the MAC to configure for signalling information processing.

IV.2.2.2.6. `PHY-RX-SIGN-END.indication`

a. Function

This primitive indicates that signalling information currently being received is complete.

b. Semantics of the service primitive

The semantics of the primitive are as follows:

```
PHY-RX-SIGN-END.indication
```

This primitive has no parameters.

c. When generated

The primitive is generated and sent to the local MAC entity by a receiving PHY each time that it has received a valid signalling channel end-of-frame delimiter (EFD).

d. Effect of receipt

The receipt of this primitive by the MAC entity causes the MAC entity to analyze the delivered `SIGNLALLING` data octets. In case the transmitter's channel state information is sent, this information is stored for future use. In case resources for data delivery are assigned to itself, it configures the PHY accordingly by issuing the `PHY-RX-SETAMC.request` primitive.

IV.2.2.3. Control sublayer-to-sublayer service primitives

IV.2.2.3.1. `PHY-TX-SETPOW.request`

a. Function

This primitive is a request to adapt the transmission power on the sub-carriers to the specified levels.

b. Semantics of the service primitive

The primitive provides the following parameters:

```
PHY-TX-SETPOW.request(POWVECTOR)
```

The `POWVECTOR` parameter is a vector that contains one value per sub-carrier / sub-carrier group.

c. When generated

The MAC sublayer will issue this primitive after having received `PHY-TX-SIGN-END.confirm` primitive by the PHY layer.

d. Effect of receipt

The receipt of this primitive by the PHY entity causes the PHY to adapt the power level per sub-carrier / sub-carrier group accordingly. Once the PHY entity has adapted the power levels, it will issue a `PHY-TX-SETPOW.confirm` to the MAC sublayer.

IV.2.2.3.2. `PHY-TX-SETPOW.confirm`

a. Function

This primitive confirms the adaptation of sub-carrier / sub-carrier group power levels.

b. Semantics of the service primitive

The semantics of the primitive are as follows:

```
PHY-TX-SETPOW.confirm
```

This primitive has no parameters.

c. When generated

The PHY sublayer will issue this primitive in response to every `PHY-TX-SETPOW.request` primitive issued by the MAC layer, whenever the sub-carrier power level have been successfully adapted.

d. Effect of receipt

The receipt of this primitive by the MAC entity causes the MAC to issue the `PHY-TX-SETAMC.request` primitive.

IV.2.2.3.3. `PHY-TX-SETAMC.request`

a. Function

This primitive is a request to apply a certain adaptive modulation and coding (AMC) configuration to the individual sub-carriers / sub-carrier groups.

b. Semantics of the service primitive

The primitive provides the following parameters:

```
PHY-TX-SETAMC.request(AMCVECTOR)
```

The `AMCVECTOR` parameter is a vector that contains one value per sub-carrier / sub-carrier group.

c. When generated

The MAC sublayer will issue this primitive after having received the `PHY-TX-SETPOW.confirm` primitive by the PHY layer.

d. Effect of receipt

The receipt of this primitive by the PHY entity causes the PHY to adapt the adaptive modulation and coding (AMC) configuration per sub-carrier / sub-carrier group accordingly. Once the PHY entity has adapted the AMC configurations, it will issue a `PHY-TX-SETAMC.confirm` to the MAC sublayer.

IV.2.2.3.4. `PHY-TX-SETAMC.confirm`

a. Function

This primitive confirms the adaptation of sub-carrier / sub-carrier adaptive modulation and coding (AMC) configurations.

b. Semantics of the service primitive

The semantics of the primitive are as follows:

```
PHY-TX-SETAMC.confirm
```

This primitive has no parameters.

c. When generated

The PHY sublayer will issue this primitive in response to every `PHY-TX-SETAMC.request` primitive issued by the MAC layer, whenever the sub-carrier AMC configurations have been successfully deployed.

d. Effect of receipt

The receipt of this primitive by the MAC entity causes the MAC to issue the `PHY-TX-MACPDU-START.request` primitive.

IV.2.2.3.5. `PHY-RX-SETAMC.request`

a. Function

This primitive is a request to get ready for data reception using specified coding and modulation configurations on a subset of the available sub-carriers / sub-carrier groups.

b. Semantics of the service primitive

The primitive provides the following parameters:

```
PHY-RX-SETAMC.request(AMCVECTOR)
```

The `AMCVECTOR` parameter is a vector that contains one value per sub-carrier / sub-carrier group. If an entry is set to `NULL`, this sub-carrier / sub-carrier group is not used for data reception.

c. When generated

The MAC sublayer will issue this primitive after having received the `PHY-RX-SIGN-END.indication` primitive by the PHY layer and having created the `AMCVECTOR` according to the delivered signalling data.

d. Effect of receipt

The receipt of this primitive by the PHY entity causes the PHY to apply the ACM configurations to the sub-carriers / sub-carrier groups according to the information delivered in `AMCVECTOR`. Once the configuration is set up accordingly, the PHY sublayer issues a `PHY-RX-SETAMC.confirm` to the MAC sublayer.

IV.2.2.3.6. `PHY-RX-SETAMC.confirm`

a. Function

This primitive confirms the adaptation of AMC parameters..

b. Semantics of the service primitive

The semantics of the primitive are as follows:

```
PHY-RX-SETAMC.confirm
```

This primitive has no parameters.

c. When generated

The PHY sublayer will issue this primitive in response to every `PHY-RX-SETAMC.request` primitive issued by the MAC layer, whenever the adaptation of appropriate AMC parameters has been successfully completed.

d. Effect of receipt

The receipt of this primitive by the MAC is unspecified.

IV.3. Flowcharts

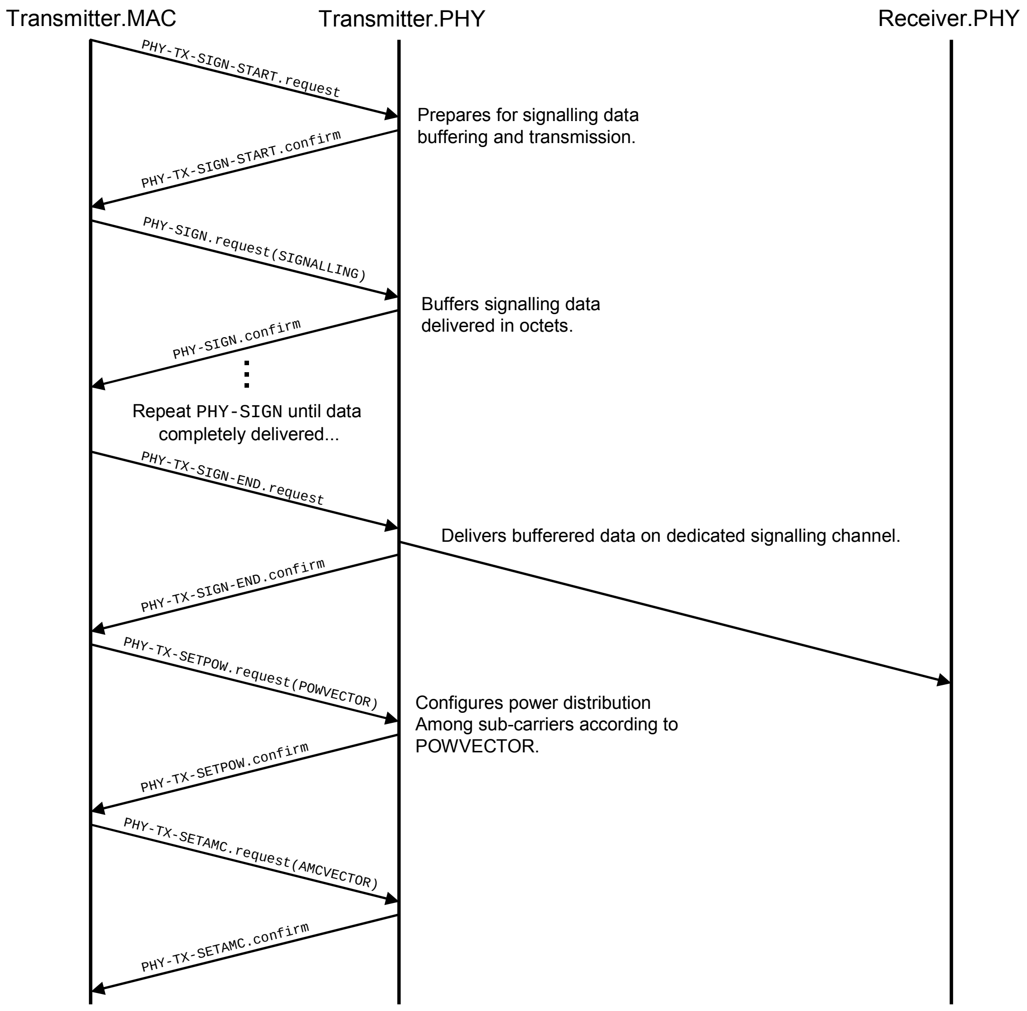
In the following an overview of the connections between the different service primitives is given in five different flow charts:

1. Transmitter setup and signalling information delivery (TX_SIGN Setup).
2. Transmitter data handling and transmission (TX_DATA Delivery).
3. Receiver signalling information reception and setup (RX_SIGN Setup).
4. Receiver data reception and handling (RX_DATA Delivery).
5. A forced channel state update between two entities (Probing).

The flowcharts contain only a few explanations. For a deeper understanding on the primitive functionality refer to the previous section.

IV.3.1. TX_SIGN Setup

In addition to preparing the `Transmitter.PHY` for the data delivery of the next frame, the configuration information is forwarded to the `Receiver.PHY` using the dedicated signalling channel. The signalling data also include channel state information (power values measured during the last frame reception).

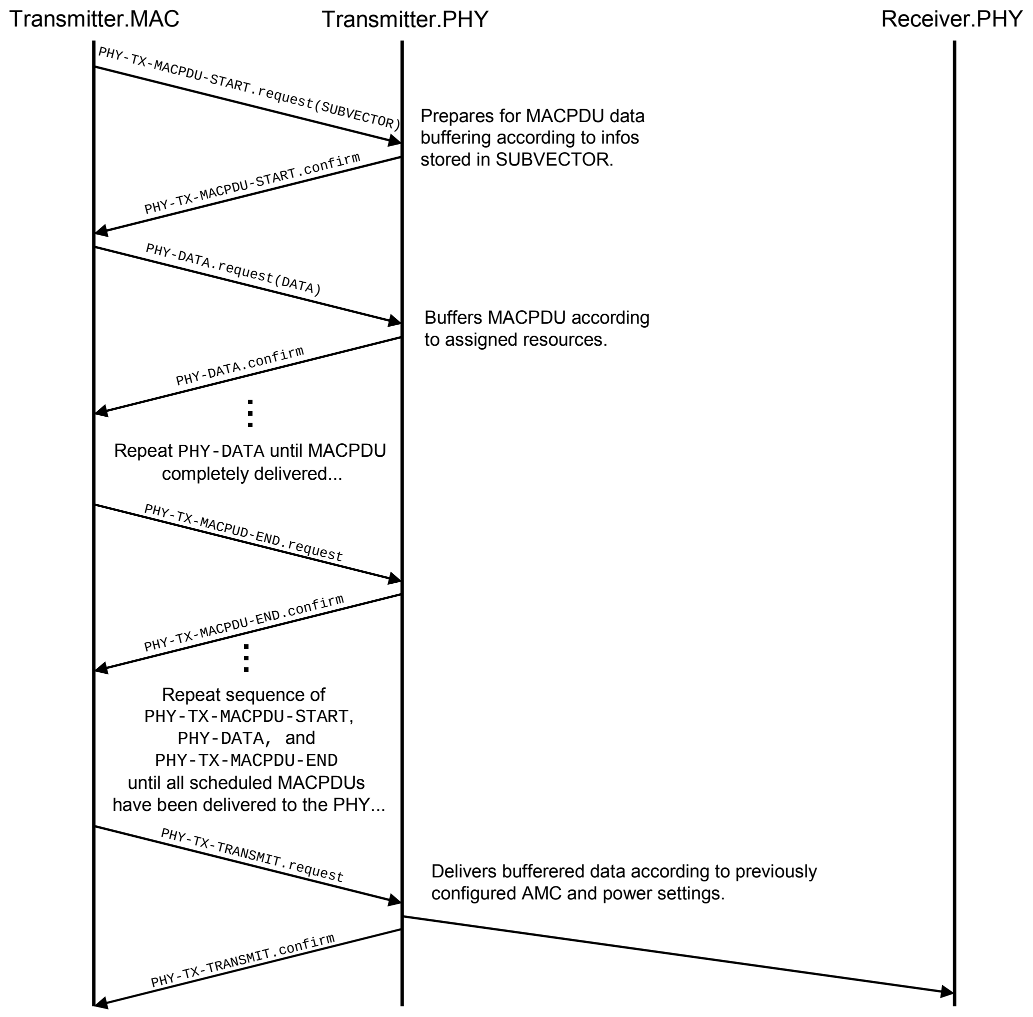

Transmitter.MAC
Transmitter.PHY
Receiver.PHY
PHY-TX-SIGN-START.request
Prepares for signalling data buffering and transmission.
PHY-TX-SIGN-START.confirm
PHY-SIGN.request(SIGNALLING)
Buffers signalling data delivered in octets.
PHY-SIGN.confirm
Repeat PHY-SIGN until data completely delivered...
PHY-TX-SIGN-END.request
Delivers bufferered data on dedicated signalling channel.
PHY-TX-SIGN-END.confirm
PHY-TX-SETPOW.request(POWVECTOR)
Configures power distribution Among sub-carriers according to POWVECTOR.
PHY-TX-SETPOW.confirm
PHY-TX-SETAMC.request(AMCVECTOR)
PHY-TX-SETAMC.confirm

Flowchart 1: The use of service primitives during transmitter setup.

At this point the Transmitter.PHY is configured for data delivery as shown in Flowchart 2. At the receiver side, the signalling information is received and analyzed as shown in Flowchart 3.

IV.3.2. TX_DATA Delivery

The `Transmitter.MAC` delivers data-octets on a per-MACPDU basis. The `Transmitter.PHY` stores them according to the configured settings and waits for the command to start the transmission.

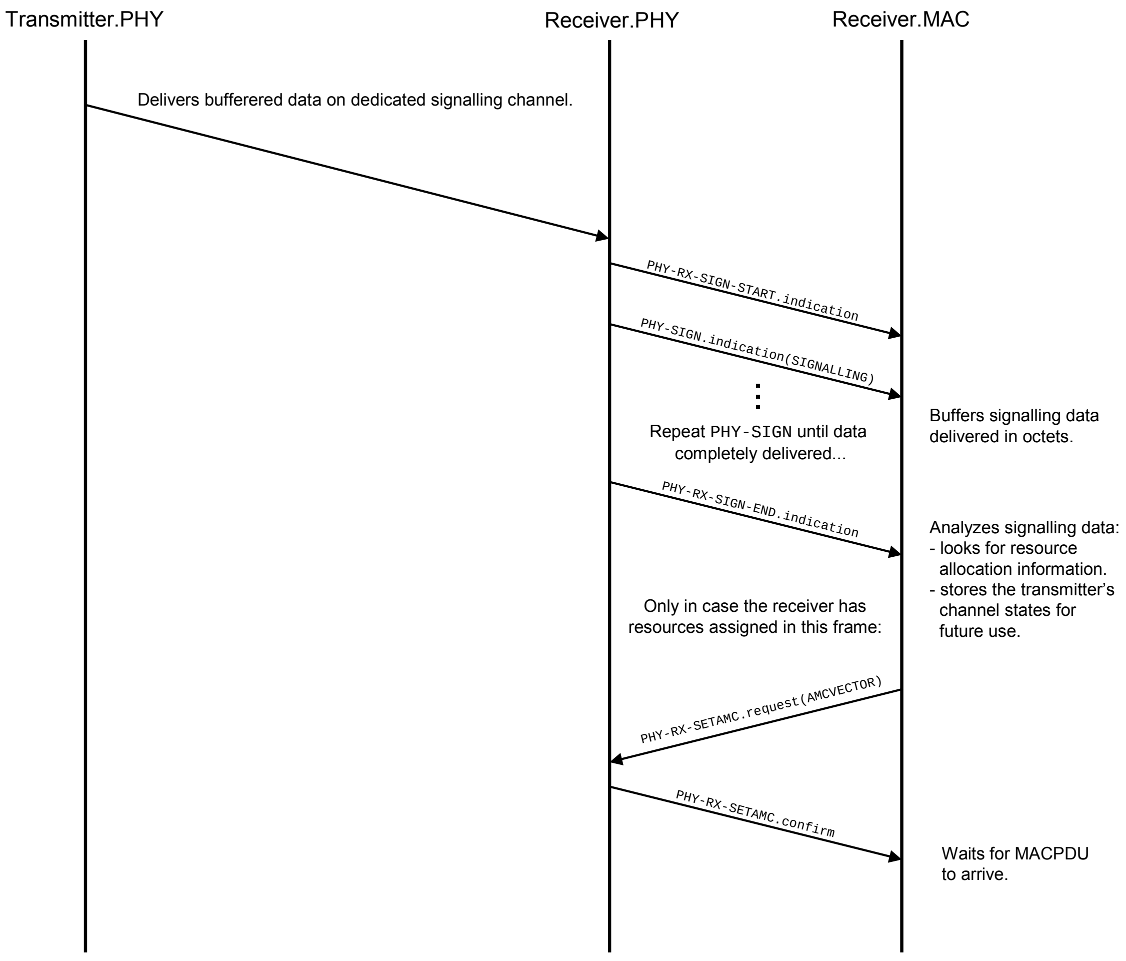

Transmitter.MAC
Transmitter.PHY
Receiver.PHY
PHY-TX-MACPDU-START.request(SUBVECTOR)
Prepares for MACPDU data buffering according to infos stored in SUBVECTOR.
PHY-TX-MACPDU-START.confirm
PHY-DATA.request(DATA)
Buffers MACPDU according to assigned resources.
PHY-DATA.confirm
Repeat PHY-DATA until MACPDU completely delivered...
PHY-TX-MACPUD-END.request
PHY-TX-MACPDU-END.confirm
Repeat sequence of PHY-TX-MACPDU-START, PHY-DATA, and PHY-TX-MACPDU-END until all scheduled MACPDUs have been delivered to the PHY...
PHY-TX-TRANSMIT.request
Delivers bufferered data according to previously configured AMC and power settings.
PHY-TX-TRANSMIT.confirm

Flowchart 2: The use of service primitives during MACPDU delivery.

At this point the transmitter's regular per-frame-time cycle stops and starts all over again with the signalling information delivery part shown in Flowchart 1. The receiver processes the received data as presented in Flowchart 4.

IV.3.3. RX_SIGN Setup

As the `Receiver.PHY` cannot handle the signalling information issued by the `Transmitter.MAC`, the signalling information is forwarded to the `Receiver.MAC` that in turn configures the `Receiver.PHY`.

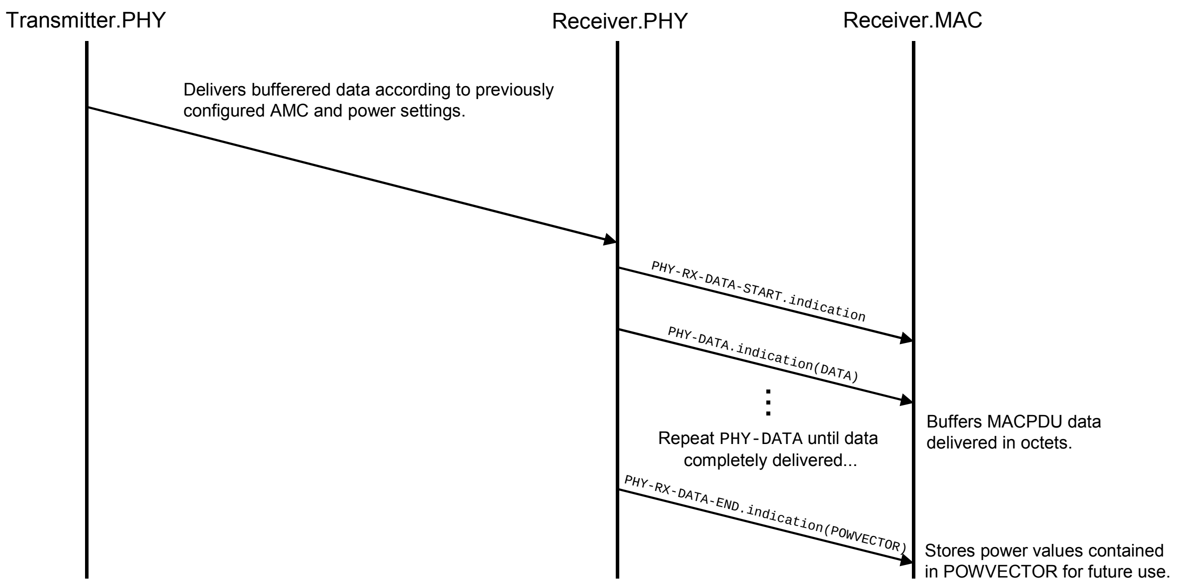

Transmitter.PHY
Receiver.PHY
Receiver.MAC
Delivers bufferered data on dedicated signalling channel.
PHY-RX-SIGN-START.indication
PHY-SIGN.indication(SIGNALLING)
Repeat PHY-SIGN until data completely delivered...
Buffers signalling data delivered in octets.
PHY-RX-SIGN-END.indication
Analyzes signalling data:
- looks for resource allocation information.
- stores the transmitter's channel states for future use.
Only in case the receiver has resources assigned in this frame:
PHY-RX-SETAMC.request(AMCVECTOR)
PHY-RX-SETAMC.confirm
Waits for MACPDU to arrive.

Flowchart 3: The use of service primitives after signalling data reception.

At this point the receiver is configured to receive data according to the sub-carrier allocations and AMC settings that have been chosen for this frame-time.

IV.3.4. RX_DATA Delivery

The `Receiver.PHY` receives data as configured and passes them on to the `Receiver.MAC`. Afterwards the measured power values are passed to the MAC that stores them until they are sent to the entity that is currently acting as transmitter as part of the signalling data.



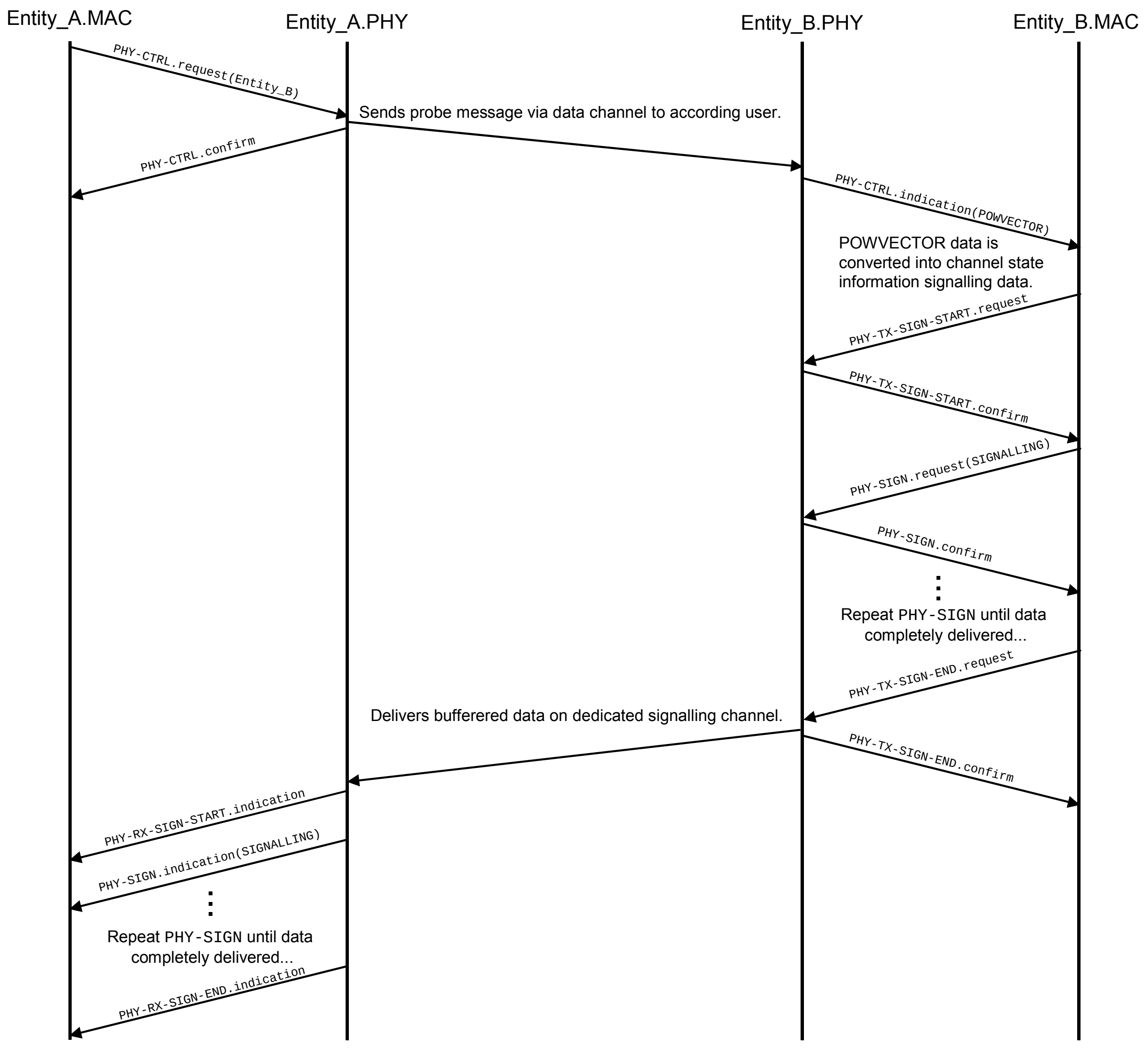
Transmitter.PHY
Receiver.PHY
Receiver.MAC
Delivers bufferered data according to previously configured AMC and power settings.
PHY-RX-DATA-START.indication
PHY-DATA.indication(DATA)
Repeat PHY-DATA until data completely delivered...
Buffers MACPDU data delivered in octets.
PHY-RX-DATA-END.indication(POWVECTOR)
Stores power values contained in POWVECTOR for future use.

Flowchart 4: The use of service primitives after data reception.

At this point the regular per-frame-time cycle stops and starts all over again with the transmitter setup and signalling information delivery shown in Flowchart 1.

IV.3.5. Probing

In case an entity has been idle for some time, another entity that wants to deliver data it might decide to send a probe message in order to update the necessary channel state information.


Entity_A.MAC
Entity_A.PHY
Entity_B.PHY
Entity_B.MAC
PHY-CTRL.request(Entity_B)
Sends probe message via data channel to according user.
PHY-CTRL.confirm
PHY-CTRL.indication(POWVECTOR)
POWVECTOR data is converted into channel state information signalling data.
PHY-TX-SIGN-START.request
PHY-TX-SIGN-START.confirm
PHY-SIGN.request(SIGNALLING)
PHY-SIGN.confirm
Repeat PHY-SIGN until data completely delivered...
PHY-TX-SIGN-END.request
Delivers bufferered data on dedicated signalling channel.
PHY-TX-SIGN-END.confirm
PHY-RX-SIGN-START.indication
PHY-SIGN.indication(SIGNALLING)
Repeat PHY-SIGN until data completely delivered...
PHY-RX-SIGN-END.indication

Flowchart 5: The use of service primitives for entity probing.

V. Generic Application Programming Interface (API)

Based on the primitives introduced in Section IV.2, this section provides pseudo-code functions of a higher level language that resembles those used on several existing platforms. A discussion on the feasibility of such functions is very much appreciated.

V.1. Data Transmission Functions

V.1.1. `boolean tx_phy_setupSubs(double* subPower, int* subConf)`

This function is provided by the transmitter's PHY and called by its MAC.

It sets the transmission parameters per sub-carrier at the PHY for the subsequent transmission of user data.

a. Inputs

- `double* subPower`: a double array holding one power value for each sub-carrier that to be applied in the upcoming frame-time.
- `int* subConf`: an integer array holding one number for each sub-carrier that reflects a special modulation/coding type combination to be applied in the upcoming frame-time.

b. Returns

- A `boolean` value that indicates successful deployment on the PHY side.

c. Primitives involved

- `PHY-TX-SETPOW.request(POWVECTOR),PHY-TX-SETPOW.confirm`
- `PHY-TX-SETAMC.request(AMCVECTOR),PHY-TX-SETAMC.confirm`

V.1.2. `boolean tx_phy_deliverMacpdu(Macpdu* macpdu, Bitmap* subAlloc)`

This function is provided by the transmitter's PHY and called by its MAC.

It delivers a MACPDU from the transmitter's MAC to its PHY and determines the sub-carriers on which it will be transmitted.

a. Inputs

- `Macpdu* macpdu`: an arbitrary Macpdu struct, or char array.
- `Bitmap* subAlloc`: a binary bitmap, the size is determined by the number of sub-carriers in the system. A one indicates that the according sub-carrier is selected for transmission of `macpdu`.

b. Returns

- A `boolean` value that indicates successful deployment on the PHY side.

c. Primitives involved

- `PHY-DATA.request(DATA),PHY-DATA.confirm`
- `PHY-TX-MACPDU-START.request(SUBVECTOR),PHY-TX-MACPDU-START.confirm`
- `PHY-TX-MACPDU-END.request,PHY-TX-MACPDU-END.confirm`

V.1.3. `boolean tx_phy_startTransmission()`

This function is provided by the transmitter's PHY and called by its MAC.

It indicates the start of a new frame-time and thus causes the PHY to transmit data as configured.

a. Inputs

- -

b. Returns

- A `boolean` value that indicates completed data transmission.

c. Primitives involved

- `PHY-TX-TRANSMIT.request, PHY-TX-TRANSMIT.confirm`

V.1.4. `boolean rx_phy_setupSubs(int* subConf)`

This function is provided by the receiver's PHY and called by its MAC.

It sets the transmission parameters per sub-carrier at the PHY for the subsequent reception of user data.

a. Inputs

- `int* subConf`: an integer array holding one number for each sub-carrier that reflects a special modulation/coding type combination to be applied in the upcoming frame-time.

b. Returns

- A `boolean` value that indicates successful deployment on the PHY side.

c. Primitives involved

- `PHY-RX-SETAMC.request(AMCVECTOR),PHY-RX-SETAMC.confirm`

V.1.5. `double** rx_mac_deliverMacpdu(Macpdu* macpdu)`

This function is provided by the receiver's MAC and called by its PHY.

It delivers the received data along with the measured power values per sub-carrier to the MAC upon the completed reception of a frame.

a. Inputs

- `Macpdu* macpdu`: the received MACPDU struct or `char` array.

b. Returns

- A double array that contains one measured power value per sub-carrier.

c. Primitives involved

- `PHY-RX-DATA-START.indication`
- `PHY-DATA.indication(DATA)`
- `PHY-RX-DATA-END.indication`

V.2. Signalling Functions

V.2.1. boolean tx_phy_deliverSignalling(Signalling* signalling)

This function is provided by the transmitters PHY and called by its MAC.

It delivers signalling information from the transmitter's MAC to its PHY to be transmitted on the dedicated control channel following a pre-determined modulation, coding and power setting.

a. Inputs

- `Signalling* signalling`: an arbitrary signalling struct, or char array.

b. Returns

- A `boolean` value that indicates successful deployment on the PHY side.

c. Primitives involved

- `PHY-SIGN.request(SIGNALLING),PHY-SIGN.confirm`
- `PHY-TX-SIGN-START.request,PHY-TX-SIGN-START.confirm`
- `PHY-TX-SIGN-END.request,PHY-TX-SIGN-END.confirm`

V.2.2. boolean rx_mac_deliverSignalling(Signalling* signalling)

This function is provided by the receiver's MAC and called by its PHY.

It delivers the received signalling data along with the measured power values per sub-carrier to the MAC upon the completed reception of a signalling-information frame.

a. Inputs

- `Signalling* signalling`: the received signalling struct, or char array.

b. Returns

- A `boolean` value that indicates successful delivery to MAC side.

c. Primitives involved

- `PHY-SIGN.indication(SIGNALLING)`
- `PHY-RX-SIGN-START.indication`
- `PHY-RX-SIGN-END.indication`

V.3. Control Functions

V.3.1. `boolean tx_phy_probing(Address userId)`

This function is provided by the transmitters PHY and called by its MAC.

It triggers the transmission of a probing message to a certain user in order to update necessary channel state information.

a. Inputs

- `Address userID`: an arbitrary address format uniquely referring to the user with `userId`..

b. Returns

- A `boolean` value that indicates completed transmission of probe message.

c. Primitives involved

- `PHY-CTRL.request(USER),PHY-CTRL.confirm,PHY-CTRL.indication`

VI. Concluding Remarks

The authors consider this document as a proposal for the research community and hope that it will become a basis for a widely used interface enhancing the cooperation within. We will be grateful for any related comment or criticism.

Acknowledgements

We would like to thank the following persons and their groups for their valuable comments on earlier versions of this paper:

- Henning Wiemann (Ericsson, AC/EDD)
- Raghu R. Rao (Xilinx)
- Douglas C. Sicker (University of Colorado)

References

[1] R.W. Chang. *Synthesis of band limited orthogonal signals for multi-channel data transmission.* Bell Systems Technical Journal, 45:1775-1796, December 1966.

[2] J. Bingham. *ADSL, VDSL and Multicarrier Modulation.* John Wiley & Sons, Inc., 2000, New York, USA.

[3] ETSI 300 744. *Digital Video Broadcasting (DVB); Framing structure, channel coding, and modulation for digital terrestrial television,* August 1997.

[4] B. O'Hara and A. Petrick. *IEEE 801.11 Handbook: A Designer's Companion.* IEEE Press, 1999.

[5] IEEE 802.16. *Air Interface for Fixed and Mobile Broadband Wireless Access Systems*., Dec.2001.

[6] P.S. Chow, et al. *A practical discrete multi-tone transceiver loading algorithm for data transmission over spectrally shaped channels.* IEEE Transactions on Communications, 43(234): 773-775, Feb.-April 1995.

[7] M. Bohge, J. Gross, M. Meyer, and A. Wolisz. Dynamic Resource Allocation in OFDM Systems: An Overview of Cross-Layer Optimization Principles and Techniques", IEEE Network Magazine, Special Issue: "Evolution toward 4G wireless networking", vol. 21, no. 1, pp. 53-59, Jan./Feb. 2007.

[8] W. Rhee and J.M. Cioffi*. Increase in capacity of multi-user OFDM systems using dynamic sub-channel allocation.* Proc. of the IEEE Vehicular Technology Conference VTC, vol. 2, pp. 1085-1089, May 2000.

[9] ANSI/IEEE Std 802.11. *Increase Part 11; Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.* IEEE-SA Standards Board, 1999 Edition (R2003), Jun 2003.