UMTS Layer 2

UMTS Layer 2

1. OVERVIEW OF LAYER 2 ARCHITECTURE

1.1 General

Layer 2 is comprised of Medium Access Control (MAC), Radio Link Control (RLC), Packet Data Convergence Protocol (PDCP) and Broadcast Multicast Control (BMC). Transport Channels form the service access points between MAC and the physical layer. Logical channels form the service access points between MAC and RLC. Radio Bearers or Radio Signalling Bearers form service access points between L2 and higher layers (e.g. Radio Resource Control or RRC).

Peer RRC, RLC and MAC entities exist in the UE and either the Node B (idle mode) or SRNC (connected mode).

RRC controls the L2 protocols to set up, maintain and clear down connections with appropriate QoS.

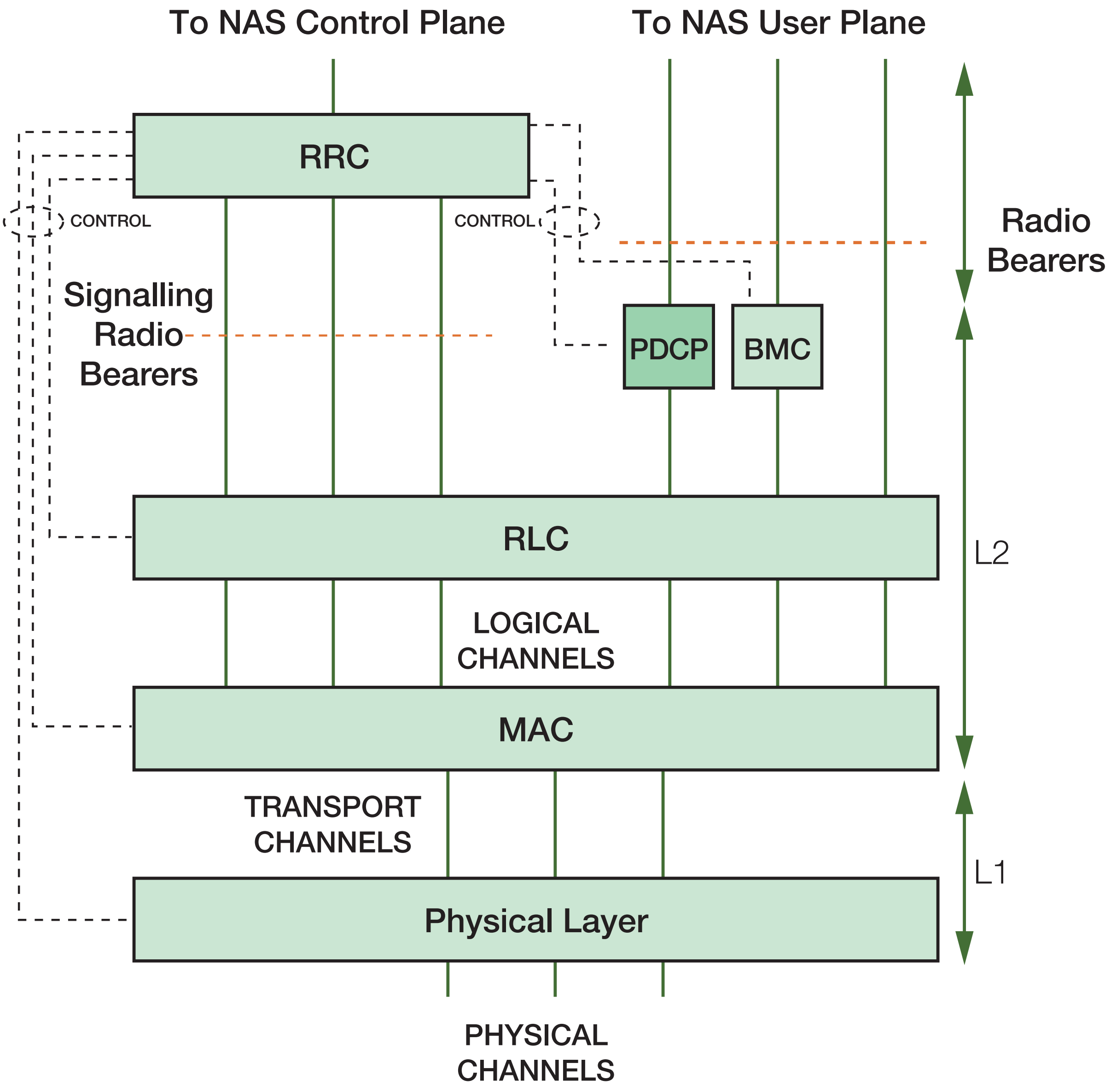


Fig. 1 – General Protocol Architecture

2. LOGICAL CHANNELS

Each logical channel type is therefore defined by the type of information transferred, and fall into one of two basic groups. These are:

- Control Channels, for control plane information
- Traffic Channels, for user plane information

2.1 Logical Traffic Channels:

There are just two channels defined for user plane information, as follows:

CTCH – Common Traffic Channel
This is a point-to-multipoint channel, and hence is relevant to communication on the downlink only. It is used for transferring dedicated user data intended for all or a group of specified terminals. (For R99 providing cell broadcast only)

DTCH – Dedicated Traffic Channel
The DTCH channel contrasts with CTCH in being a point-to-point channel – it is dedicated to just one mobile for the transfer of user information. This channel can exist in both downlink and the uplink directions.

	FDD	TDD	uplink	downlink
CTCH Common Traffic Channel	✓	✓	✗	✓
DTCH Dedicated Traffic Channel	✓	✓	✓	✓

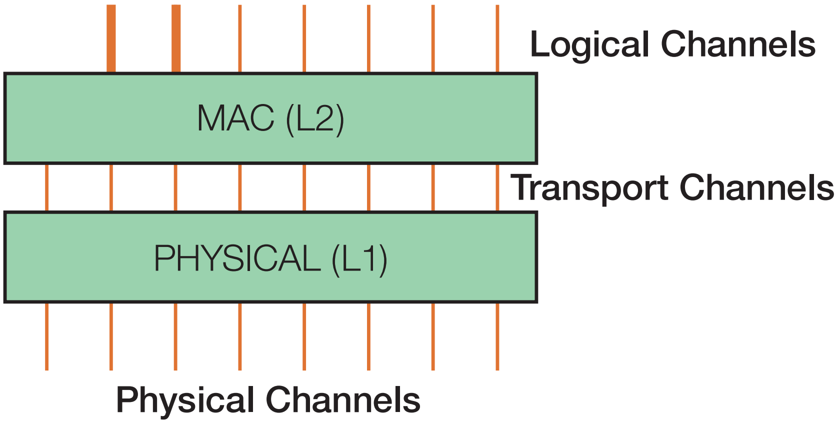


Fig. 2 – Logical Traffic Channels

2.2 Logical Control Channels:

In the Control Plane, there are five logical channels, as follows:

BCCH – Broadcast Control Channel
The BCCH is used to carry control and signalling information which is to be broadcast, and therefore is only applicable in the downlink direction.

When mapped through the lower layers, it will eventually be carried on a physical channel which uses the same channelisation code in all cells (specifically a channel known as the Primary Common Control Physical Channel, PCCPH). This means that its messages can always be read by a mobile terminal, once the terminal has detected a base station’s unique scrambling code, which it does during its initial cell search.

PCCH – Paging Control Channel
This channel is used to carry paging requests. It is therefore a downlink-only channel and is used either when the network does not know the location cell of the mobile equipment, or when the mobile is in the RRC connected state Cell_PCH, utilising “sleep mode” procedures to preserve battery power. (Also applies to URA_PCH, which is similar to the Cell_PCH state except that location updates to the UTRAN are performed on an UTRAN Routing Area (URA) basis, rather than a cell basis).

CCCH – Common Control Channel
CCCH is a channel used for transmitting control information between the network and mobiles, and is applicable in both the uplink and downlink directions. As a common channel, it is a resource which carries control information to and from a number of different mobiles. It is commonly used by mobiles which currently have no RRC connection with the network, (idle mode) and by those accessing a new cell after cell re-selection.

DCCH – Dedicated Control Channel
By contrast with CCCH, DCCH is a multi-purpose, point-to-point channel which is used to carry dedicated control information, i.e. information specific to a single mobile. It is established when a RRC connection is set-up (connected mode), and is applicable in both uplink and downlink directions.

	FDD	TDD	uplink	downlink	content
BCCH	✓	✓	✗	✓	broadcast
Broadcast Control					information
PCCH	✓	✓	✗	✓	paging
Paging Control					requests
CCCH	✓	✓	✓	✓	control
Common Control					information
DCCH	✓	✓	✓	✓	control info for
Dedicated Control					a single mobile

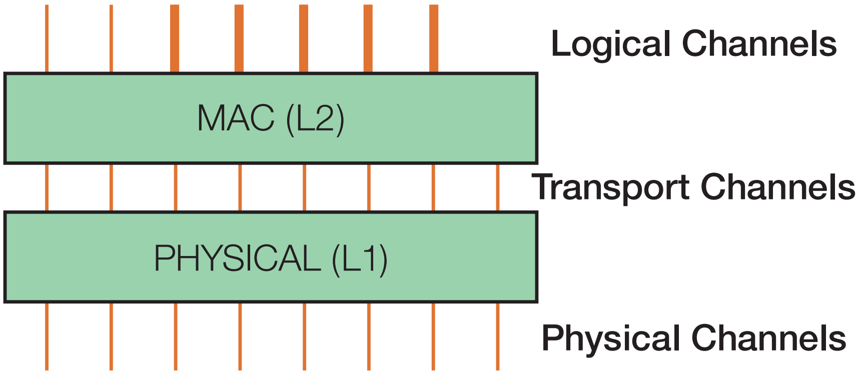


Fig. 3 – Logical Control Channels

2.3 Logical Channels for ODMA Mode

ODMA (Opportunity Driven Multiple Access) is another possible access scheme which can be applied in UMTS, although not fully specified in R’99 and unlikely to be used in the early deployments.

It is really just a relaying protocol rather than a pure access scheme, whereby a terminal which lies outside cell coverage can use another mobile terminal as a relay to transmit to the base station. It is only likely to prove feasible in the TDD scheme, where reception and transmission are in the same frequency band – if implemented in FDD, it would require terminals to be able to receive in their normal transmission band and vice versa, which is impractical to implement.

There are a number of logical channels which can be defined for future ODMA operation.

These are a single traffic channel for user data, ODTCH (ODMA Dedicated traffic channel), and two control channels: OCCCH (ODMA Common Control Channel) and ODCCH (ODMA Dedicated control channel).

Both OCCCH & ODCCH are used for transmitting control information between terminals, the difference being that OCCCH carries information common to a number of terminals, whereas ODCCH is “point-to-point”, intended for a specific terminal.

	Traffic (user data)	Control	point-to-point
ODTCH	✓	✗	✓
(ODMA Dedicated Traffic)			
OCCCH	✗	✓	✗
(ODMA Common Control)			
ODCCH	✗	✓	✓
(ODMA Dedicated Control)			

- Only feasible in TDD Mode.

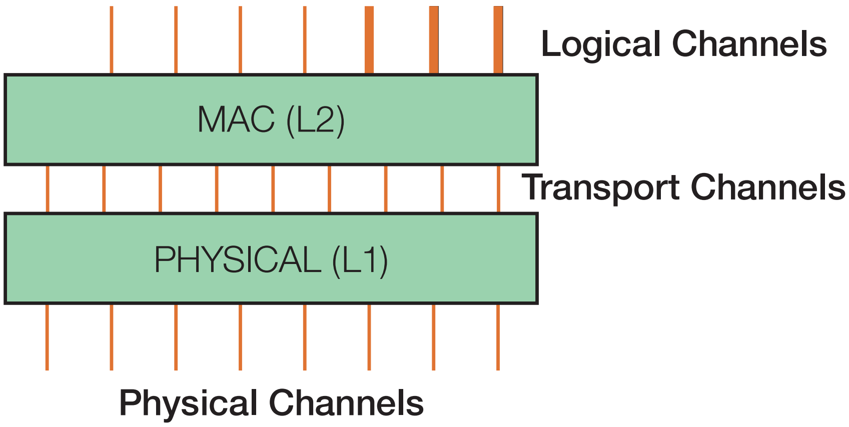


Fig. 4 – Logical Channels – ODMA Mode

3. MEDIUM ACCESS CONTROL (MAC)

3.1 MAC Functions

MAC performs the following functions:

- mapping between the logical and transport channels.

- selection of Transport Formats for each transport Channel.

- priority handling between data flows related to one user terminal, achieved by selecting high or low bit rate transport formats for the different data flows.

- priority handling between different user terminals for common or shared downlink transport channels. For dedicated transport channels, such priority handling has already been performed by RRC as part of the reconfiguration function.

- identification of different user terminals, on occasions when dedicated-type data from logical channels is carried over common transport channels. To do this, the C-RNTI or UTRAN RNTI (U-RNTI) is included in the MAC header. This process is relevant to actions such as paging or random access attempts, for example.

- Ciphering is performed within MAC if a logical channel is using the transparent RLC mode.

- multiplexing/demultiplexing of higher layer data units into/from transport blocks delivered to/from the physical layer on *common* transport channels. Service multiplexing for these common channels cannot be done in the physical layer, hence this function falls within MAC.

- multiplexing/demultiplexing of higher layer data units into/from sets of transport blocks delivered to/from the physical layer on *dedicated* transport channels. Although the physical layer makes it possible to multiplex any type of service, multiplexing within MAC is only possible for services with the same QoS parameters.

- Traffic volume monitoring, reporting to the RRC. Measurements reported to the RRC may be used to trigger reconfiguration of radio bearers and transport channels if the amounts of data being transmitted are too high or too low to make most efficient use of the assigned bearers/channels.

- dynamic transport channel type switching, which involves switching between common and dedicated transport channels, based on decisions derived from RRC.

- Access service class selection, used to prioritise usage of the Random Access channel.

MAC

- Logical ⟷ Transport Channel Mapping
- Selection of Transport Format
- Multiplexing of PDUs into Transport Blocks
- Dynamic Transport Channel Type Switching
- Identification of UEs on common traffic channels
- Priority handling
- Access Class Selection for RACH and CPCH
- Traffic Volume Monitoring
- Ciphering (RLC TrM only)

Fig. 5 – Overall Functions of MAC

3.2 Mapping of Logical Channels onto Transport Channels through MAC

In the downlink direction in both FDD and TDD modes, the paging and notification logical channel PCCH maps directly onto the transport channel PCH.

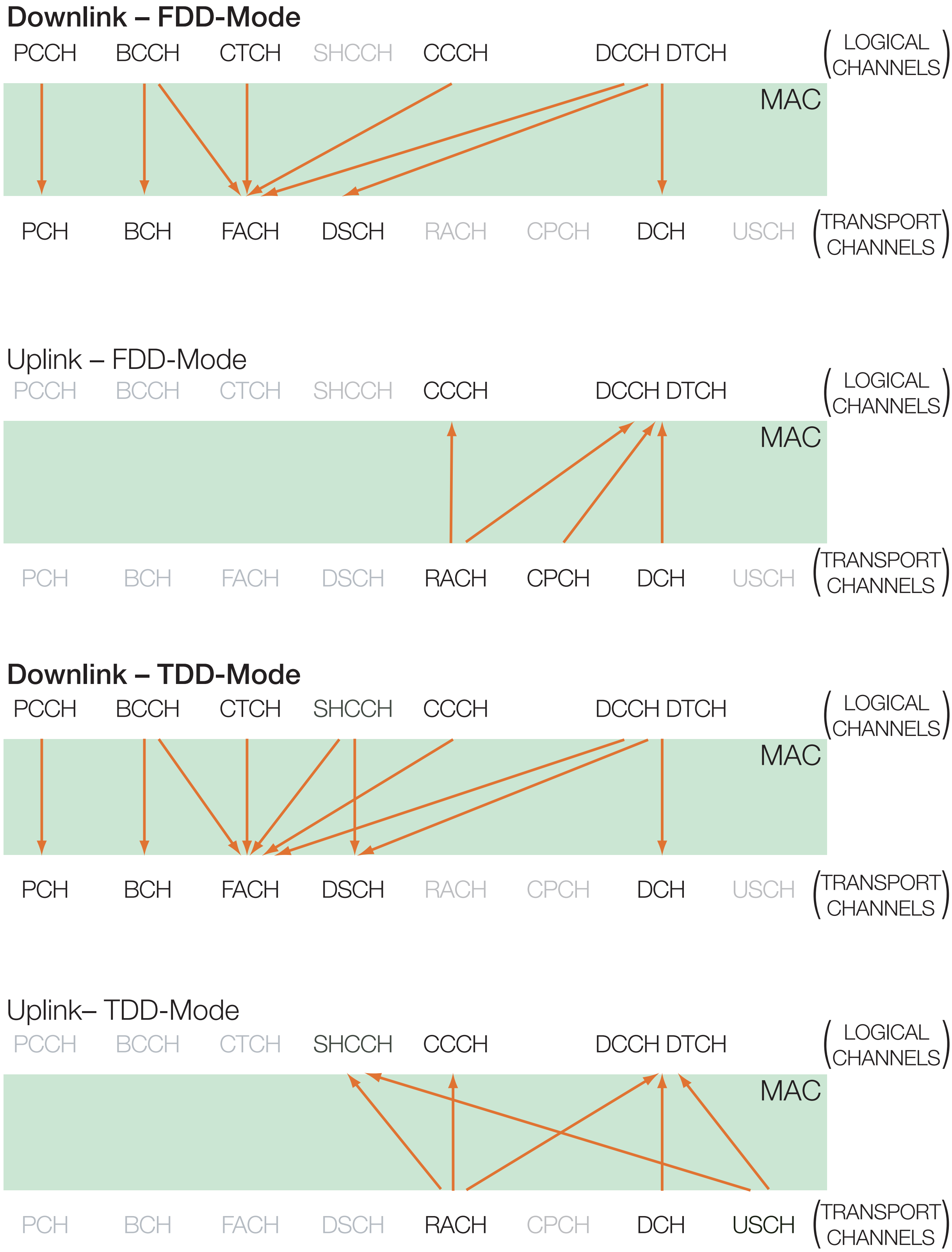
Similarly, the downlink logical channel for broadcast information, BCCH maps directly onto the transport channel BCH, in both TDD and FDD modes. However it is also possible to map BCCH onto the transport channel FACH, for small amounts of broadcast information.

The logical channels CTCH and CCCH, the common traffic and control channels, both map onto the FACH transport channel in the downlink direction, for both FDD and TDD modes. The logical channel CTCH and the transport channel FACH are not applicable on the uplink, so in this case CCCH maps solely onto RACH, again in both FDD and TDD modes.

The dedicated logical control and traffic channels DCCH and DTCH map onto transport channels DCH and, optionally, DSCH and FACH in the downlink of both FDD and TDD. In the uplink, FACH and DSCH are not applicable transport channels, and so mapping is to DCH once again, but in this case also to RACH. In the case of FDD, additional mapping is optional to CPCH and, in the case of TDD, to USCH.

Applicable in TDD mode only, the logical shared control channel SHCCH, is mapped to RACH on the uplink and to FACH or DSCH on the downlink.

ODMA channel mapping follows an equivalent pattern as the TDD mode, with the ORACH transport channel equivalent to the RACH transport channel in FDD and TDD, and ODCCH, OCCCH and ODTCH equivalent to DCCH, CCCH and DTCH respectively.



Downlink – FDD-Mode

Uplink – FDD-Mode

Downlink – TDD-Mode

Uplink– TDD-Mode

Fig. 6 – Mapping of Logical Channels onto Transport Channels

3.3 MAC Architecture

3.3.1 MAC Functional Entities

Within the MAC sublayer there are 3 functional entities (FE), each connecting to RLC via logical channels and to the physical layer via transport channels. All FEs are under the control of RRC via a control SAP.

3.3.2 MAC for the Broadcast Channel (MAC-b)

MAC-b maps information from the BCCH logical channel to BCH transport channel. The UTRAN must support one instance of MAC-b per cell. The UE supports one instance.

3.3.3 MAC for the Common and Shared Channels (MAC-c/sh)

All common and shared logical channels are mapped to appropriate transport channels by MAC-c/sh, which involves multiplexing and the addition of MAC header information. Once again, the UTRAN must support one instance of MAC-c/sh per cell and the UE must support a single instance.

3.3.4 MAC for Dedicated Channels (MAC-d)

MAC-d handles mapping between the DTCH and DCCH logical channels, with the appropriate DCH at the transport level. MAC-d may also need to map information to common transport channels, depending upon the nature of the connection and QoS. MAC-d carries out multiplexing of channels and adds header information. One instance of MAC-d must be supported in the UTRAN for every connected UE. A single instance resides in the UE.

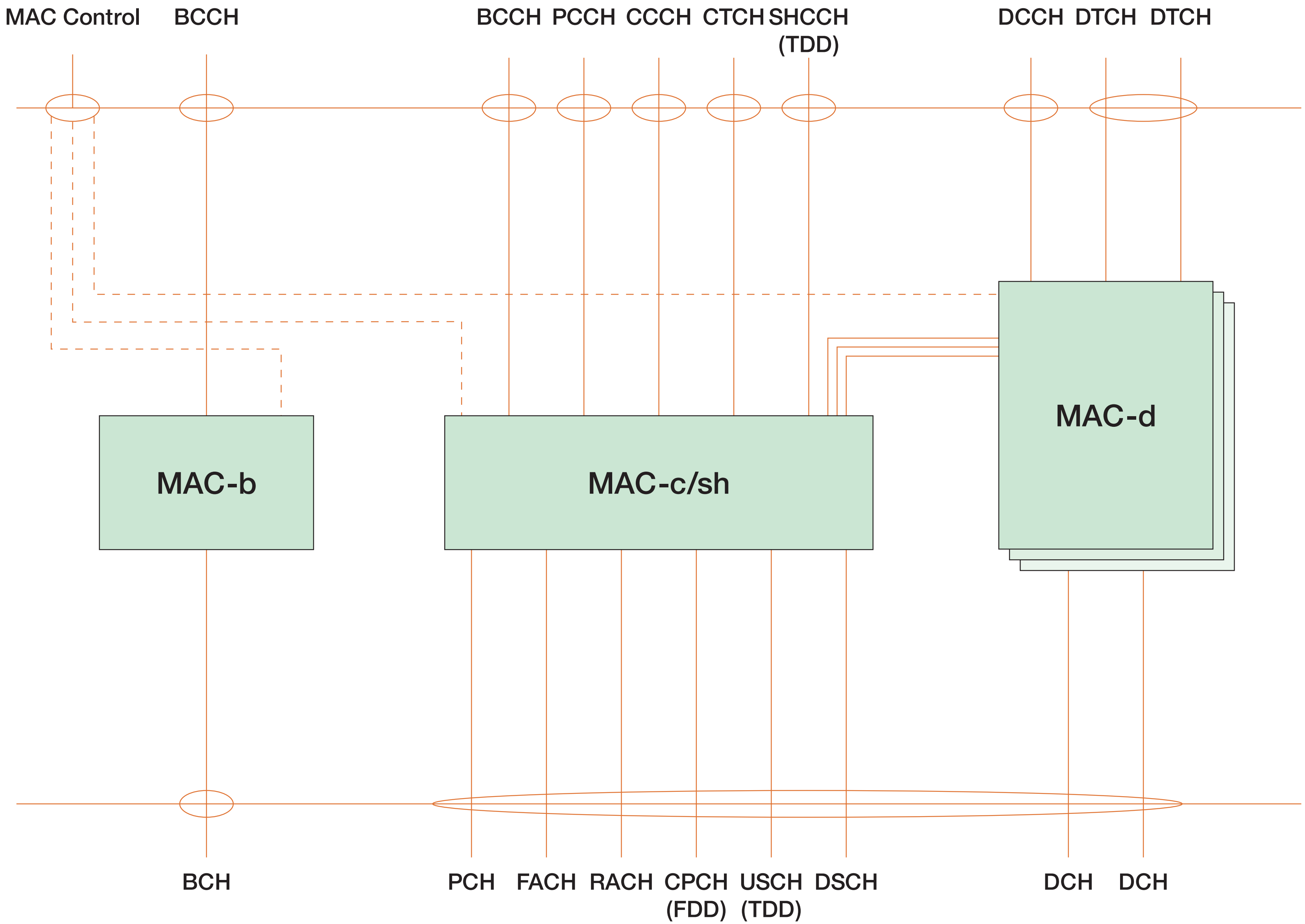


Fig. 7 – MAC Architecture

3.4 MAC Primitives

Figure 8 shows the main primitives defined for MAC protocol. MAC-DATA, the main primitive, is used for transporting data packets between peer MACs in the UTRAN and UE. Other primitives support measurement, status, and monitoring of the MAC and also radio resources.

Most MAC primitives are only of the type REQUEST and INDICATION, due mainly to the fact that MAC provides an unacknowledged service.

PRIMITIVE	Request	Indication	Response	Confirm
MAC – DATA	✓	✓		
MAC – STATUS		✓	✓	
CMAC – CONFIG	✓			
CMAC – MEASUREMENT	✓	✓		
CMAC – STATUS		✓		

Fig. 8 – MAC Primitives

3.5 MAC Protocol Data Units (PDU)

Higher layer protocol units (RLC PDUs) become the payload or service data unit (SDU) within the MAC PDU which includes the MAC header.

The contents of the header vary according to the services being provided by MAC. MAC PDUs are passed to the physical layer using transport blocks which ensure the correct rate of transfer. One transport block is transferred over the TTI defined.

3.5.1 MAC Header

The Target Channel Type Field (TCTF) is used only in association with the RACH and FACH channels and in TDD mode for the USCH and DSCH. It defines which logical channels are being mapped to the RACH/FACH. For FDD, this can be BCCH, CCCH, CTCH or DTCH/DCCH. For TDD mode additional mappings can be SHCCH over RACH/FACH, SHCCH over USCH/DSCH and DTCH/DCCH over USCH/DSCH.

The Channel Type (C/T) field is a 4 bit field identifying one particular logical channel from a maximum of 15 where multiple logical channels are being mapped to one transport channel.

The UE-ID is used when a dedicated logical channel is being carried on a common transport channel. The ID could be either a U-RNTI or C-RNTI, which is defined by the UE-ID type field.

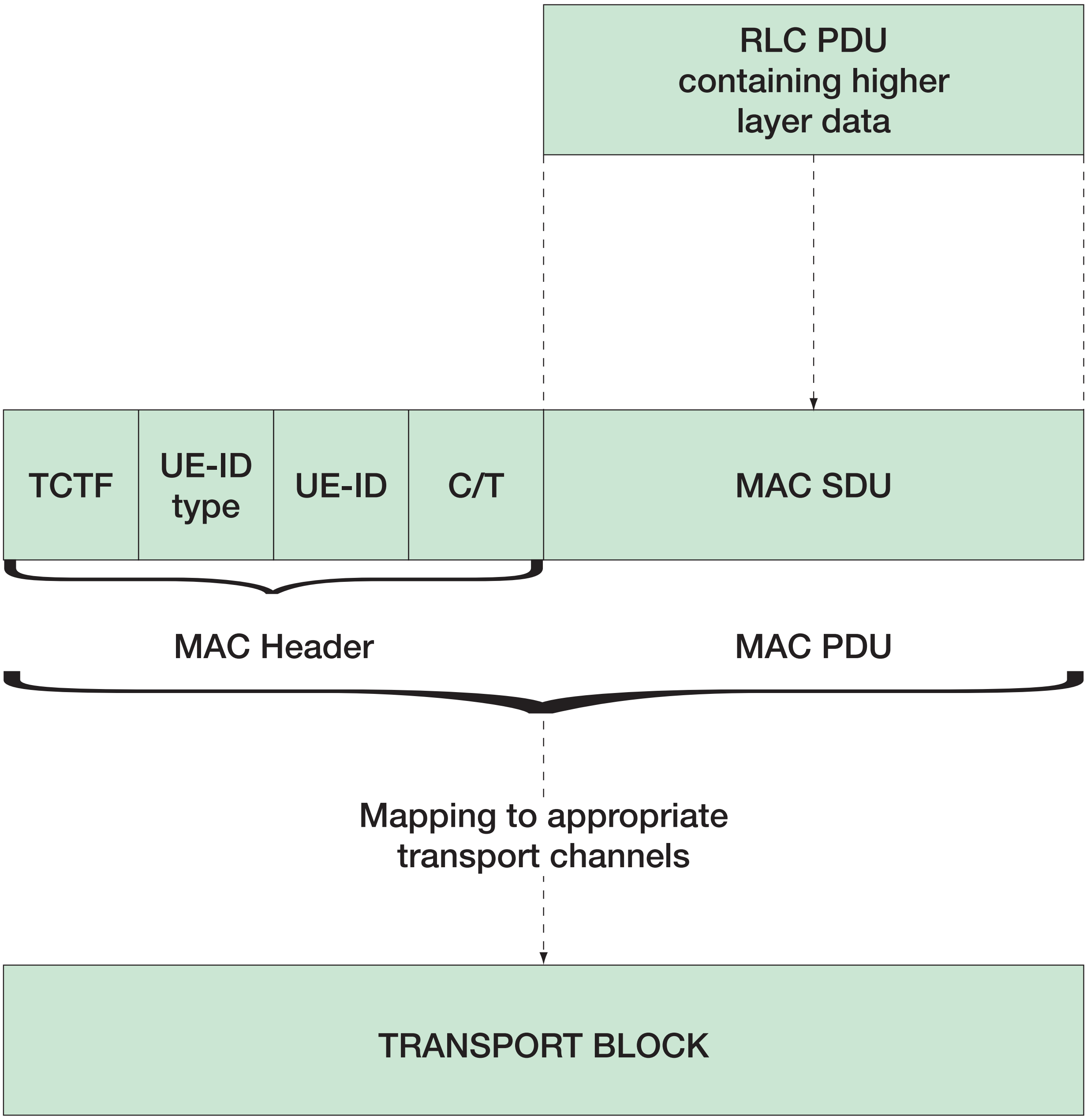


Fig. 9 – MAC PDU Processing

4. RADIO LINK CONTROL (RLC)

4.1 Radio Link Control (RLC) Services

The RLC is responsible for connection management and control of radio links, providing segmentation & retransmission services for both user and control data. In the control plane, the services provided to higher layers are known as Signalling Radio Bearers, used as they are by the RRC for signalling transport.

In the User plane, services provided by RLC are known simply as Radio Bearers. For packet user data, or broadcast, the Radio Bearer would include the service-specific protocol layers (PDCP or BMC). But for circuit switched type user data the Radio Bearer service is provided directly by RLC for other higher-layer user plane functions, such as speech codec.

Each RLC instance is configured by RRC to operate in one of three modes of data transfer. These are:

- Transparent
- Unacknowledged
- Acknowledged

Each mode provides a different set of services defining the use of that mode by the higher layers. Transfer of user data is a service which is common to all three modes.

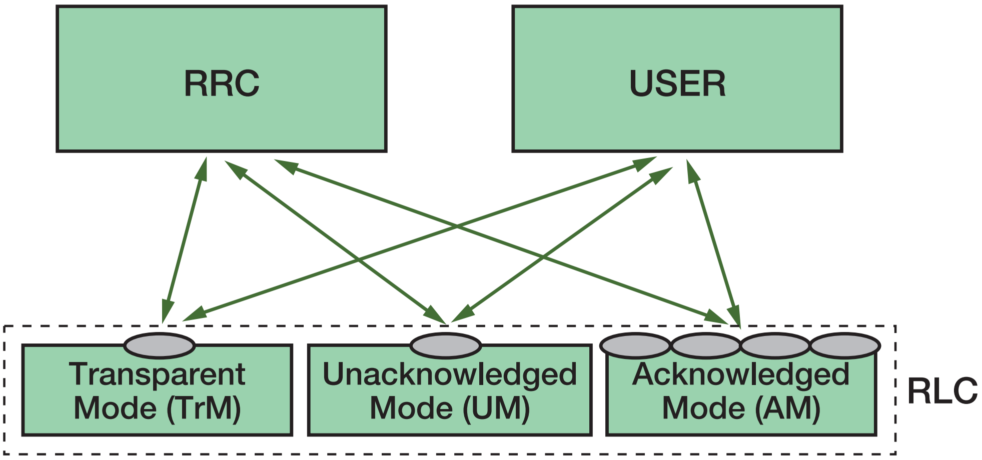
Transparent mode is defined for “quick and dirty” data transfer across the radio interface, and is the only one of the three modes which does not involve the addition of any header information onto the data unit. Erroneous data units are discarded or marked as erroneous.

Transparent mode is the mode normally used by both the PNFE and BCFE entities within RRC, for paging/notification and cell broadcast messaging.

In **Unacknowledged mode**, as in transparent mode, no retransmission protocol is used, and so data delivery is not guaranteed. Received erroneous data can be either marked or discarded, depending on configuration.

For both Transparent mode data transfer & unacknowledged mode data transfer, RLC provides a function for the segmentation of large data units into smaller ones (and re-assembly at the receive end). The segment lengths are defined when the channel is established. In unacknowledged mode, segment lengths are given by a length indicator which is within the header added to the data unit.

Unacknowledged mode additionally provides a service whereby small packet data units can be concatenated together (again indicated within a header field), a ciphering service, and a sequence number check which allows the receiver to check whether or not data has been lost.



Header	✗	✓	✓
Retransmission	✗	✗	✓
Segmentation	✓	✓	✓
Concatenation	✗	✓	✓
Ciphering	✗ (MAC)	✓	✓
Missing Data Check	✗	✓	✓
CRC Check	✓	✓	✓
In-sequence Delivery	✗	✗	✓
Duplication Detection	✗	✗	✓
Error Correction	✗	✗	✓
QoS Setting	✗	✗	✓

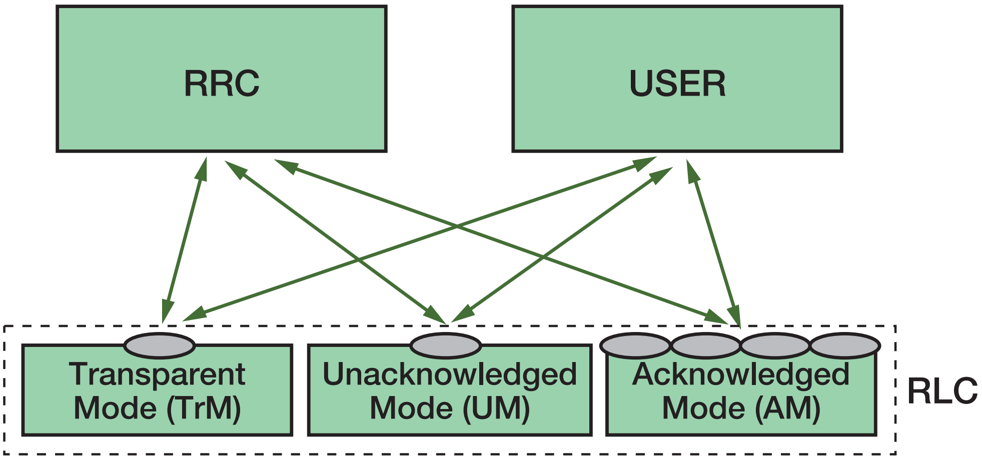
- User data uses AM (e.g. Packet based services), UM (e.g. VoIP), or TrM (e.g. streaming)

Fig. 10 – RLC Data Transfer Modes

Acknowledged mode provides a much more reliable mechanism for transferring data between two RLC layer entities, by including further services. These include in-sequence delivery of data units, detection of duplicate data units, error correction and flow control. The RLC can also set QoS levels and notify higher layers of unrecoverable errors.

Acknowledged mode is the mode used mainly by the DCFE entity within RRC, for dedicated control functions, although in some cases the other modes can be used, for example unacknowledged mode for RRC release, or transparent mode for cell update or RRC connection re-establishment requests.

For all three modes, CRC (cyclic redundancy check) error detection is performed on the physical layer, and the result is delivered to RLC along with the actual data.



Header	✗	✓	✓
Retransmission	✗	✗	✓
Segmentation	✓	✓	✓
Concatenation	✗	✓	✓
Ciphering	✗ (MAC)	✓	✓
Missing Data Check	✗	✓	✓
CRC Check	✓	✓	✓
In-sequence Delivery	✗	✗	✓
Duplication Detection	✗	✗	✓
Error Correction	✗	✗	✓
QoS Setting	✗	✗	✓

- User data uses AM (e.g. Packet based services), UM (e.g. VoIP), or TrM (e.g. streaming)

Fig. 10 – RLC Data Transfer Modes

4.2 RLC Functions

In order to provide the necessary services, a number of functions are performed within RLC. These are as follows:

- segmentation & reassembly of variable length higher layer data units into (or from) smaller RLC units. The size of these is set according to the smallest possible bit-rate for the service which is using the RLC entity. For variable bit-rate services, for a time interval in which the bit-rate is higher than the smallest one, several RLC units will be transmitted.

- concatenation, in the case where higher layer data units do not fill a whole number of RLC units. In this case the first segment of the next higher layer unit may be added to the RLC unit containing the last segment of a previous higher layer unit.

- padding, used where concatenation is not applicable (transparent mode) yet higher layer units again don't fill the RLC units. This simply involves adding padding bits to the remainder of the RLC data field.

- transfer of user data, supporting the transparent, unacknowledged and acknowledged modes, and controlled by a QoS setting.

- error correction, relevant to acknowledged mode, where retransmission can occur.

- in-sequence delivery, preserving the order of higher layer data units which are to be transferred using acknowledged mode data transfer.

- detection of duplicated received RLC data units, and making sure that only one is delivered on to the higher layer.

- flow control, which allows an RLC receiver to control the rate at which the peer RLC entity at the transmission end can send information.

- sequence number checking, in unacknowledged mode, which makes sure that reassembled data units are not corrupted. If they are, then they will be discarded.

- detection of, and recovery from, errors which occur during operation of the RLC protocol.

- ciphering is performed in the RLC for acknowledged and unacknowledged mode data transfer. (For transparent mode transfer, ciphering is performed in the MAC.)

- suspend/resume of data transfer, used during the security procedure, and commanded by the RRC via the control interface.

- Segmentation and re-assembly
- Concatenation
- Padding
- Data transfer
- Error correction
- In-sequence delivery
- Duplicate detection
- Flow control
- Sequence number check
- Error recovery
- Ciphering
- Suspend/resume data transfer

Fig. 11 – RLC Functions

4.3 RLC Architecture

RLC can be considered as two sides, transmit and receive. Both sides exist to support both Radio Bearers (User plane) and also Signalling Radio Bearers (Control plane).

Corresponding with the 3 modes of RLC, there exist 3 types of functional entity (FE).

The Transparent functional entity (TrFE) supports transparent mode, the unacknowledged functional entity (UMFE) supports unacknowledged mode and the AMFE supports acknowledged mode operation, requiring close co-operation between the transmit and receive sides.

The use of transparent, unacknowledged or acknowledged mode operation depends upon the logical channel and type / quality of service concerned.

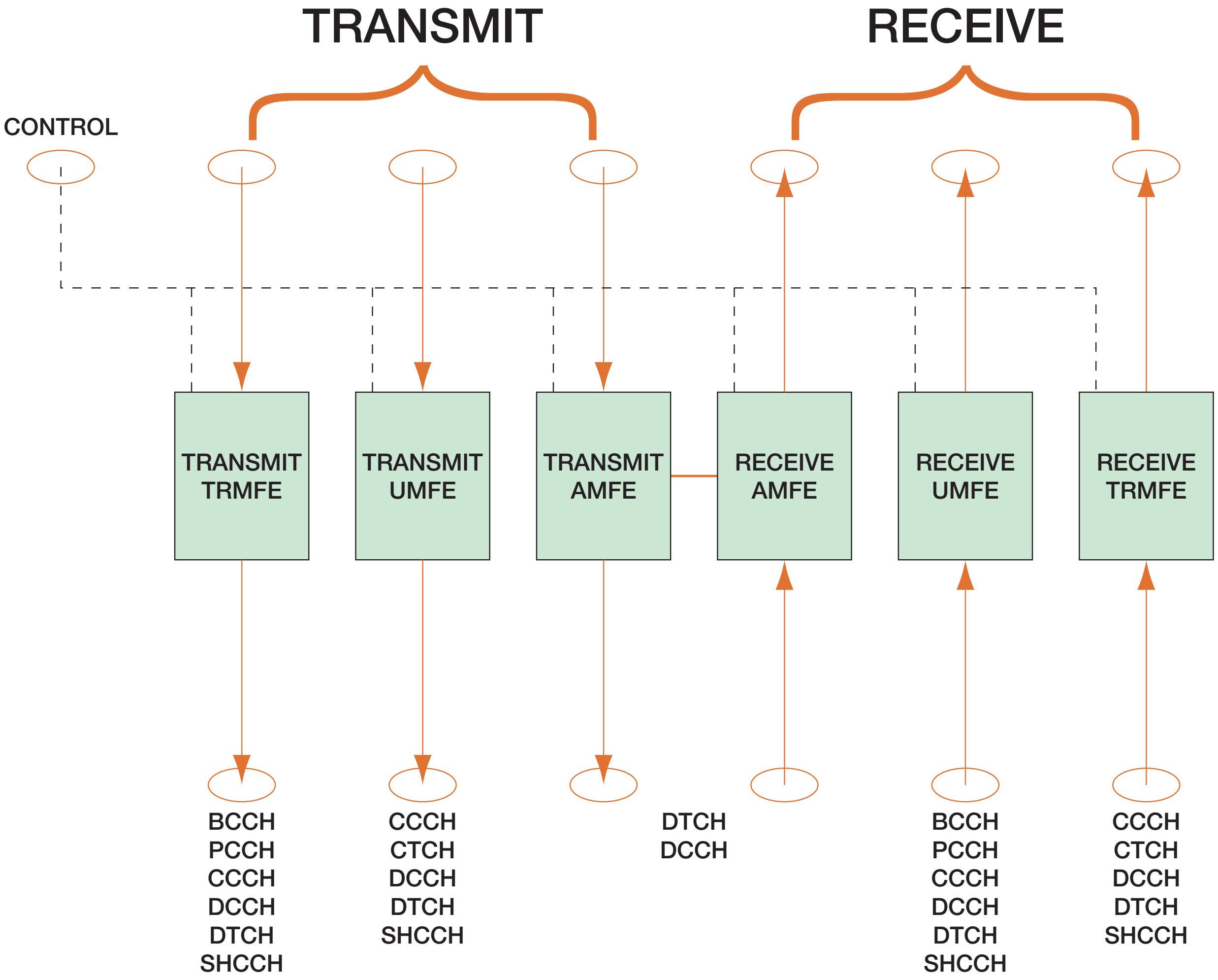


Fig. 12 – RLC Functional Entities

4.4 RLC Transparent Mode Operation

In this mode, no RLC header is added, but higher layer SDUs may be segmented for transmission and re-assembled on reception.

The BCCH, PCCH, DCCH, CCCH and SHCCH control plane logical channels make use of this mode, together with the user plane logical channel DTCH.

Note that CCCH does not use this mode for uplink.

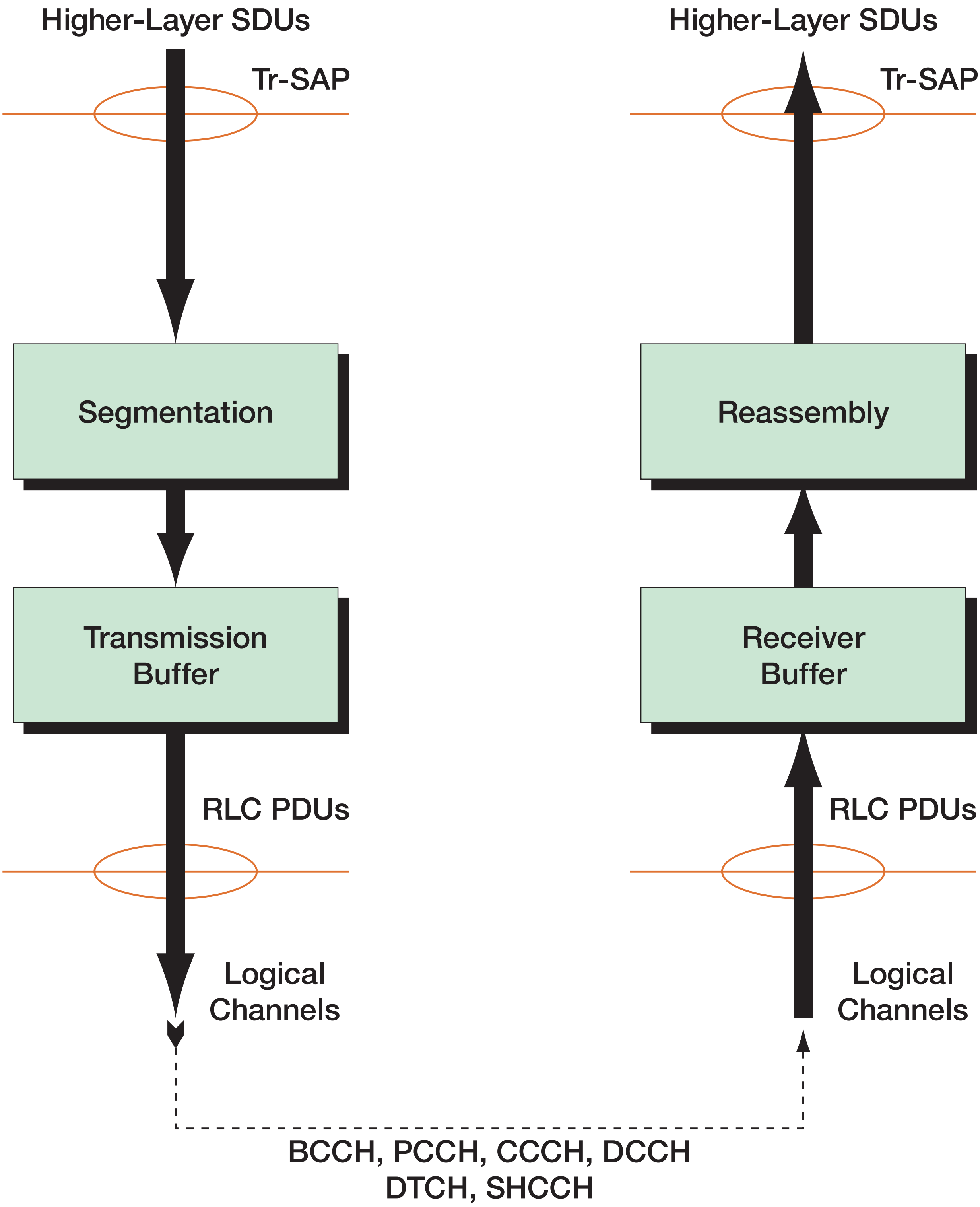


Fig. 13 – RLC TM Operation

4.5 RLC Unacknowledged Mode Operation

In this mode, higher layer SDUs can be either segmented (if large) or concatenated (if small) into an RLC PDU payload area. Optionally, ciphering can be implemented before the addition of the RLC header which provides an indication of segmentation or concatenation plus sequence numbers to aid more reliable re-assembly at the receiving end.

In the control plane, the logical channels CCCH, DCCH and SHCCH use this mode, together with user plane logical channels DTCH and CTCH.

Note that CCCH uses this mode for downlink only.

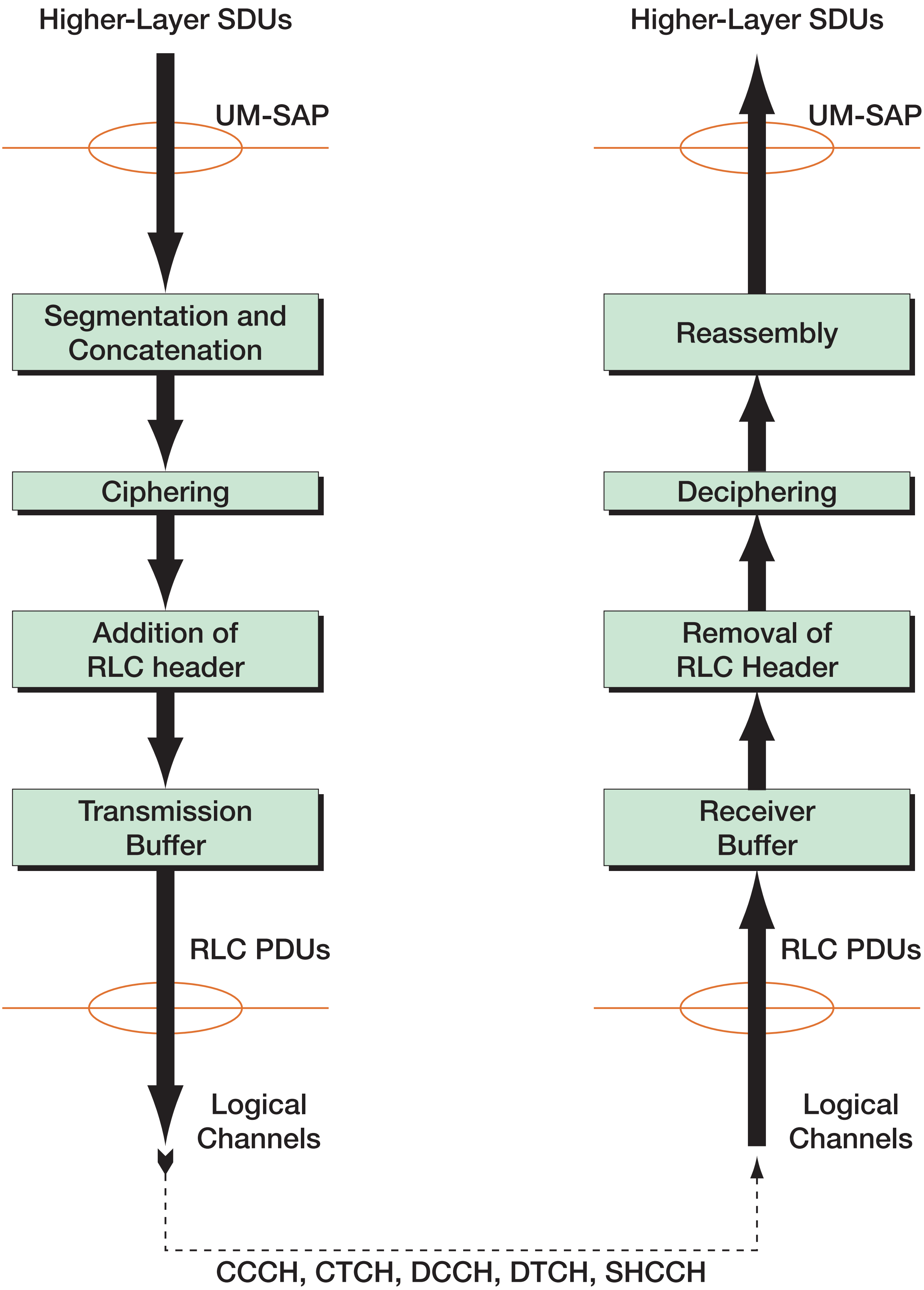


Fig. 14 – RLC UM Operation

4.6 RLC Acknowledged Mode Operation

4.6.1 AM Transmission

Higher layer SDUs are segmented, or concatenated (as required) into the payload arc of an RLC PDU as a payload unit (PU). A PU can have variable length as determined by RRC at bearer set-up. The addition of an appropriate header completes the RLC PDU. RLC PDUs are passed simultaneously to the re-transmission buffer (to be stored for possible re-transmission) and to the multiplexer. The multiplexer determines how data PDUs and status PDUs are multiplexed. Status PDUs can be formed in their own right or included in data PDUs using a process call piggybacking.

PDUs may be ciphered but this excludes the header, which requires final setting of some fields before being passed to MAC via a logical channel.

4.6.2 AM Reception

Incoming MAC PDUs are stored in the receive buffer until a complete RLC SDU has been received. Status PDUs and any piggybacked status information is extracted and retransmissions are triggered as required. Finally, the RLC header is removed before SDUs are passed to higher layers.

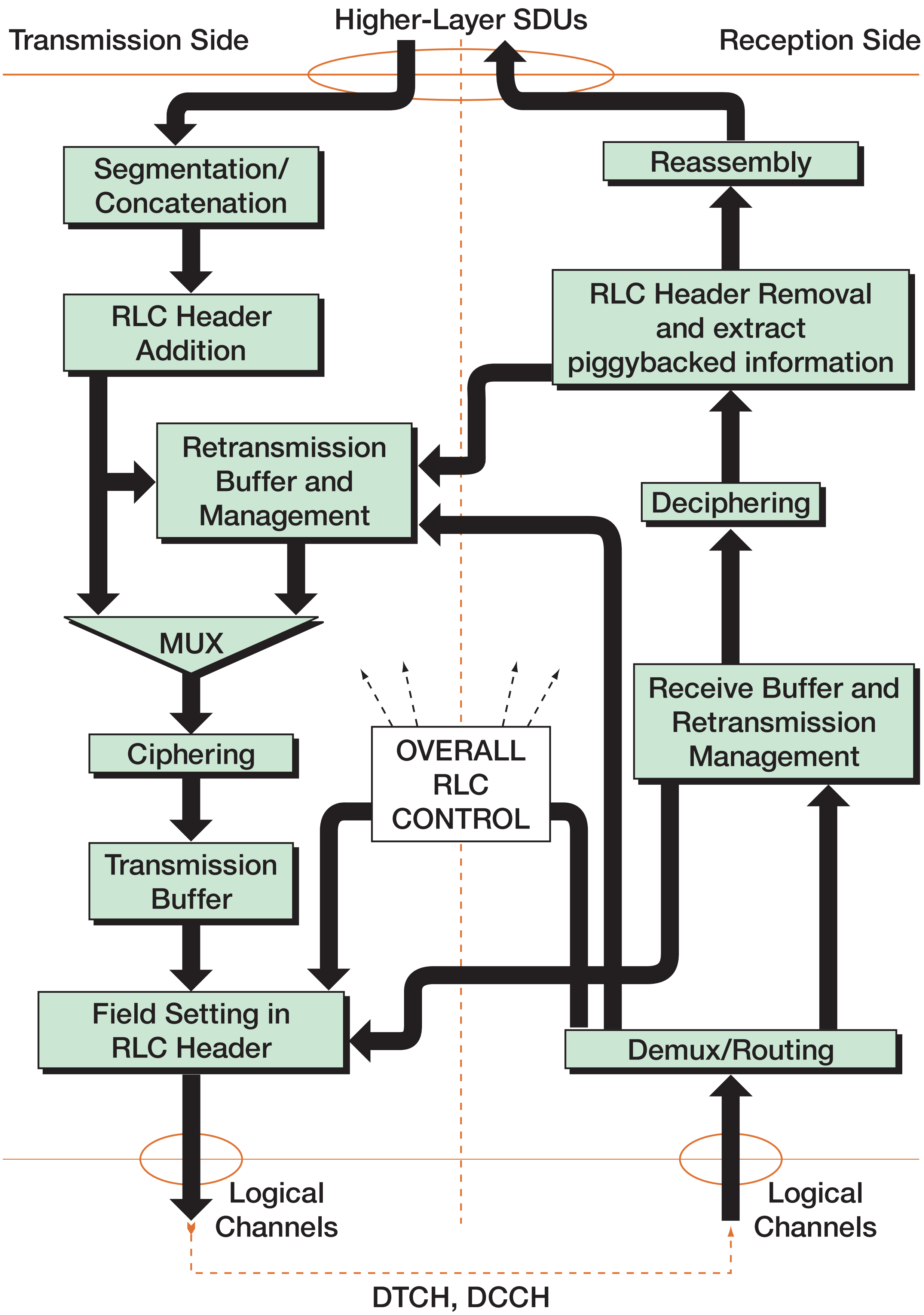


Fig. 15 – RLC AM Operation

4.7 RLC Primitives

Figure 16 summarises primitives used between RLC and higher layers. The response primitive is not required because acknowledgement of delivery is generated within RLC.

For acknowledged mode, RLC-AM-DATA is used to pass data to and from higher layers. It is also used as a confirm primitive for this mode.

Data is passed in unacknowledged mode using RLC-UM-DATA and in transparent mode using RLC-TR-DATA.

CRLC-CONFIG is used for establishment, release or re-configuration of RLC operation.

For acknowledged mode or unacknowledged mode operation, the flow of PDUs can be suspended using CRLC-SUSPEND and later resumed using CRLC-RESUME. For suspension, the parameter N is used to determine the sequence number from which to suspend transfer.

CRLC-STATUS is used by RLC to inform RRC of an unexpected event, defined by an event code (EVC).

Generic Name	Request	Indication	Response	Confirm
RLC-AM-DATA	Data, CNF, MUI	Data, Discard info	–	MUI
RLC-UM-DATA	Data	Data	–	–
RLC-TR-DATA	Data	Data	–	–
CRLC-CONFIG	E/R, Ciphering Elements (AM/UM) AM Parameters (AM)	–	–	–
CRLC-SUSPEND	N	–	–	–
CRLC-Resume (UM/AM only)	–	–	–	–
CRLC-STATUS	–	EVC	-	–

KEY

CNF	=	Confirmation Request
Data	=	Higher-Layer SDUs
EVC	=	Event Code
E/R	=	Establishment/Re-establishment
MUI	=	Message Unit Identifier
N	=	An Integer value

Fig. 16 – RLC Primitives

4.8 RLC PDU Types

For TRM and UM there is a single PDU type for each mode. AM uses several different types.

4.8.1 AM PDU

In AM, one PDU type is used for transfer of higher layer SDUs and four other types provide peer-to-peer communication as well as control of RLC itself.

The PDU used to carry higher layer SDUs can also carry piggybacked status information as shown in figure 17.

The first field in the RLC header (D/C) indicates whether this is a data or control PDU. This is followed by a sequence number. For AMD this is of length 12 bits and used for both re-assembly and re-transmission purposes. The polling (P) bit can be used to solicit a status report from a peer entity. The header extension field of 2 bits indicates whether any length indicator bits are present. If present, length indicators of length 7 to 15 bits indicate the position of boundaries between SDUs concatenated into the data field (PU). The PU is padded if necessary to be a whole number of octets. (Unless a piggybacked STATUS PDU is present).

The RESET and RESET ACK PDUs are used to reset all protocol states, times and other variables such as sequence number in order to synchronise two peer entities.

RLC Header

D/C	Sequence Number	P	HE	Length Indicator(s)	Data (Payload Unit)	Padding / Piggybacked Status PDU

D/C = data or control PDU

P = polling bit

HE = header extension

AM PDU	Purpose
AMD	Sequenced AM data
STATUS	Status report (solicited or unsolicited)
Piggybacked STATUS	Piggybacked version of above
RESET	Reset (sequence number) command
RESET ACK	Acknowedgement of RESET

Fig. 17 – RLC AM PDU Types

4.8.1.1 Status PDU

The structure of a STATUS PDU is virtually the same for either the stand-alone or piggybacked case. A standalone STATUS PDU will have a D/C bit set to indicate its control status. A piggybacked STATUS PDU does not use this bit currently. The 3 bit PDU type field is set to all zeros to indicate a STATUS PDU.

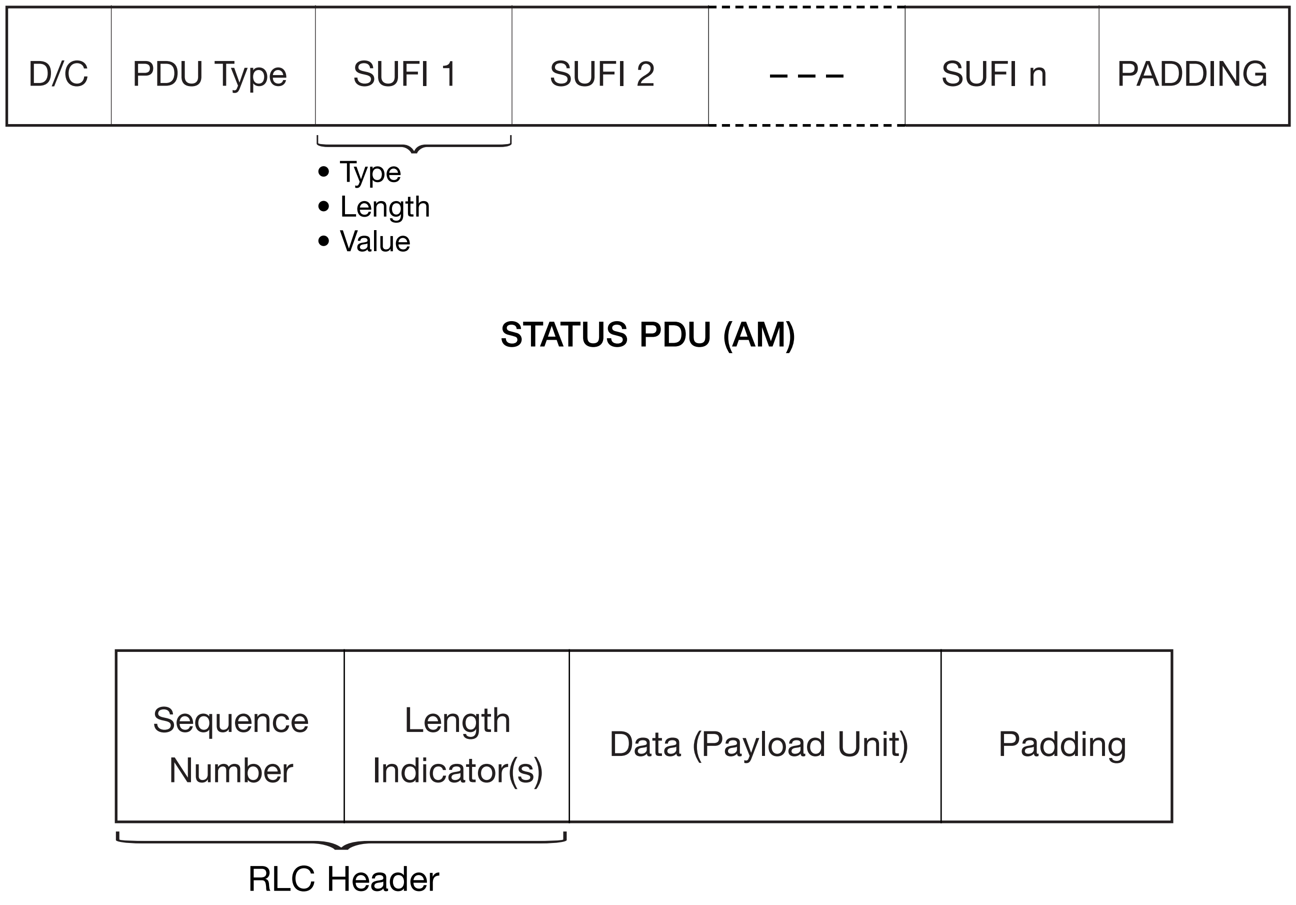
One or more Super Fields (SUFI) form the main body of the PDU. SUFIs carry information about successfully and unsuccessfully received PDUs and window sizes. Each SUFI is divided into a Type field, Length field and Value field. The length field describes the length of the value field which contains status information defined by the type field.

4.8.2 UM PDU

A UM PDU carries a header consisting simply of a sequence number of 7 bits and optional length indicators used in the same way as for AM mode. Similarly it can be padded if required to a whole number of octets.

4.8.3 TRM PDU

A TRM PDU has no RLC header and consists of a segment of a higher layer SDU or a complete higher layer SDU.



STATUS PDU (AM)

UM PDU

Fig. 18 – Status (AM) and UM PDUs

5. PACKET DATA CONVERGENCE PROTOCOL (PDCP)

PDCP exists in the user plane for packet data transfer. Its functions are compression of packet data headers and PDU numbering.

Compression of packet headers improves efficiency over the radio interface. This is especially important for packet protocols with long headers, such as IPv6.

Header compression can be carried out in accordance with a range of different compression protocols.

PDU numbering for RLC AM ensures no loss of PDUs when SRNS relocation or hard handover (within UMTS) occurs.

5.1 PDCP Architecture

PDCP supports entities for each mode of RLC. For R99, the relationship is one to one as drawn, but multiplexing is likely under later releases.

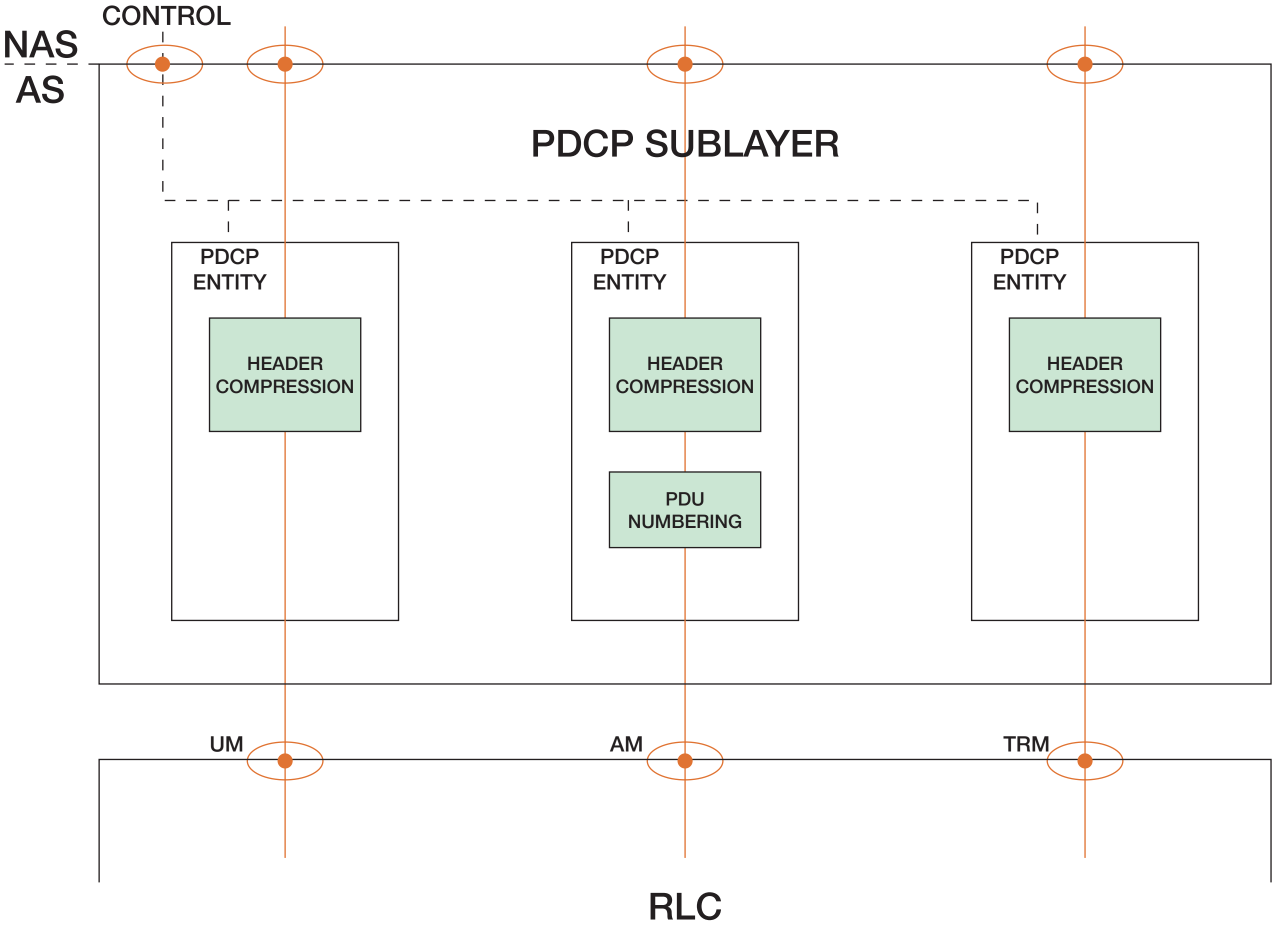
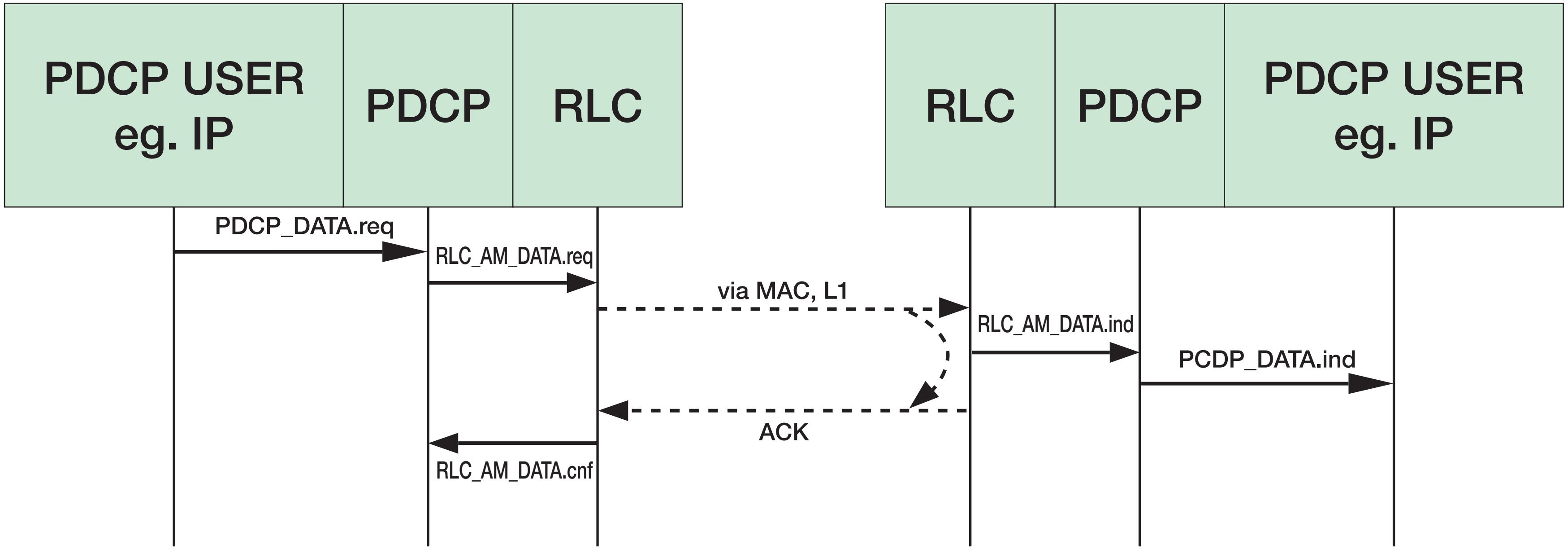


Fig. 19 – PDCP Architecture

5.2 PDCP Transfer in RLC Acknowledged Mode

Referring to figure 20, the process beings with the user of PDCP using the primitive PDCP-DATA.req. PDCP will now carry out compression of the datagram's header according to an agreed protocol and then add PDCP header information (which may include a sequence number). This now constitutes a PDCP PDU. The PDCP PDU is passed to RLC using the primitive RLC-AM-DATA.req, which becomes an RLC SDU. RLC adds an appropriate header and passes the RLC PDU to its peer at the receiving end via MAC and the sending RLC expects an acknowledgement from the receiving end. Once this is received, the sending RLC confirms delivery to the sending PDCP using the primitive RLC-AM-DATA.cnf. At the receiving end, PDCP receives the RLC PDU using RLC-AM-DATA.ind. The PCDP header is then processed before removal and the remaining IP datagram passed to IP using PDCP-DATA.ind.

Fig. 20 – PDCP Data Transfer in RLC AM Mode



5.3 PDCP PDU Structure

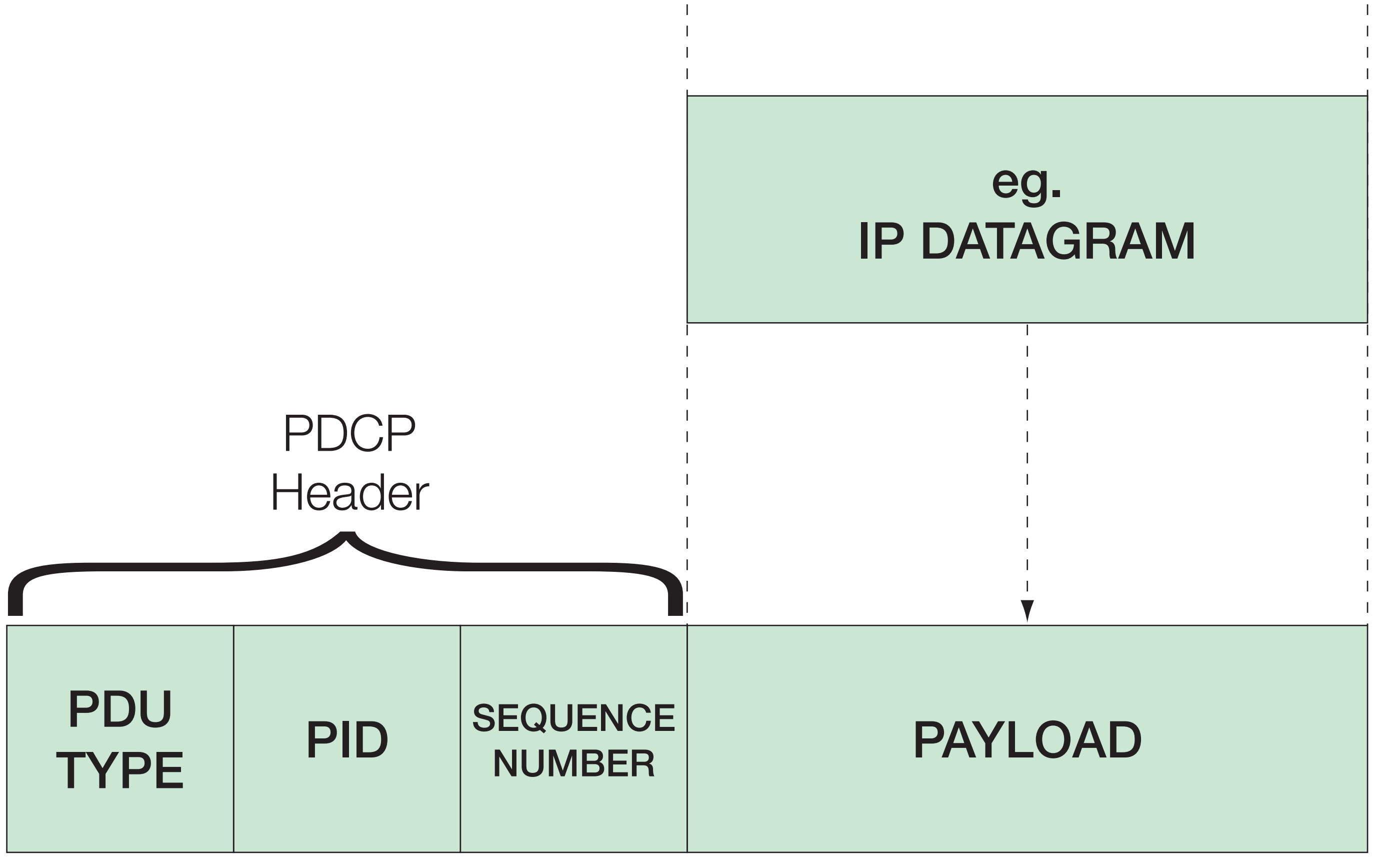
The PDCP PDU consists of a payload carrying the user data (e.g. IP datagram) and a header.

The form of header depends on the exact PDCP PDU type but will have 0 to 3 elements which are all shown in figure 21.

The PDU type field can have only two values to indicate either the presence of a packet identifier (PID) alone, or the presence of both a PID and sequence number.

The PID field indicates the header compression protocol employed and the sequence number can be used to avoid packet loss when performing SRNC relocation or hard handover.

Fig. 21 – PDCP PDU Structure



PID = Packet Identifier

6. BROADCAST MULTICAST CONTROL (BMC)

BMC is a L2 sublayer managing broadcasting and multicasting of downlink messages to UEs. For UMTS R99 this will be limited to the SMS Cell Broadcast protocol already defined in GSM but future UMTS releases will make more use of BMCs capabilities.

On the UTRAN side, there is one BMC entity per cell. There is a single BMC entity in the UE. There is a two-way exchange of control information between RRC and BMC to ensure efficient scheduling of cell broadcast messages.

6.1 Cell Broadcasting (CB) and BMC

Cell broadcast messages originate from a variety of possible sources which are aggregated in a Cell Broadcast Centre (which may belong to a third party or the network operator). CB messages will be transferred via the core network (CN) to the relevant RNC which passes them to BMC in appropriate cells where the CB message should be transmitted. BMC analyses the control information supplied with each CB message to determine scheduling. This is communicated to RRC which responds with an appropriate configuration for the CTCH logical channel. BMC also generates a scheduling message which is ultimately transmitted along with the CB message to which is relates.

At the UE, the BMC sublayer analyses the scheduling message and uses this to indicate to RRC the CB messages that should be received.

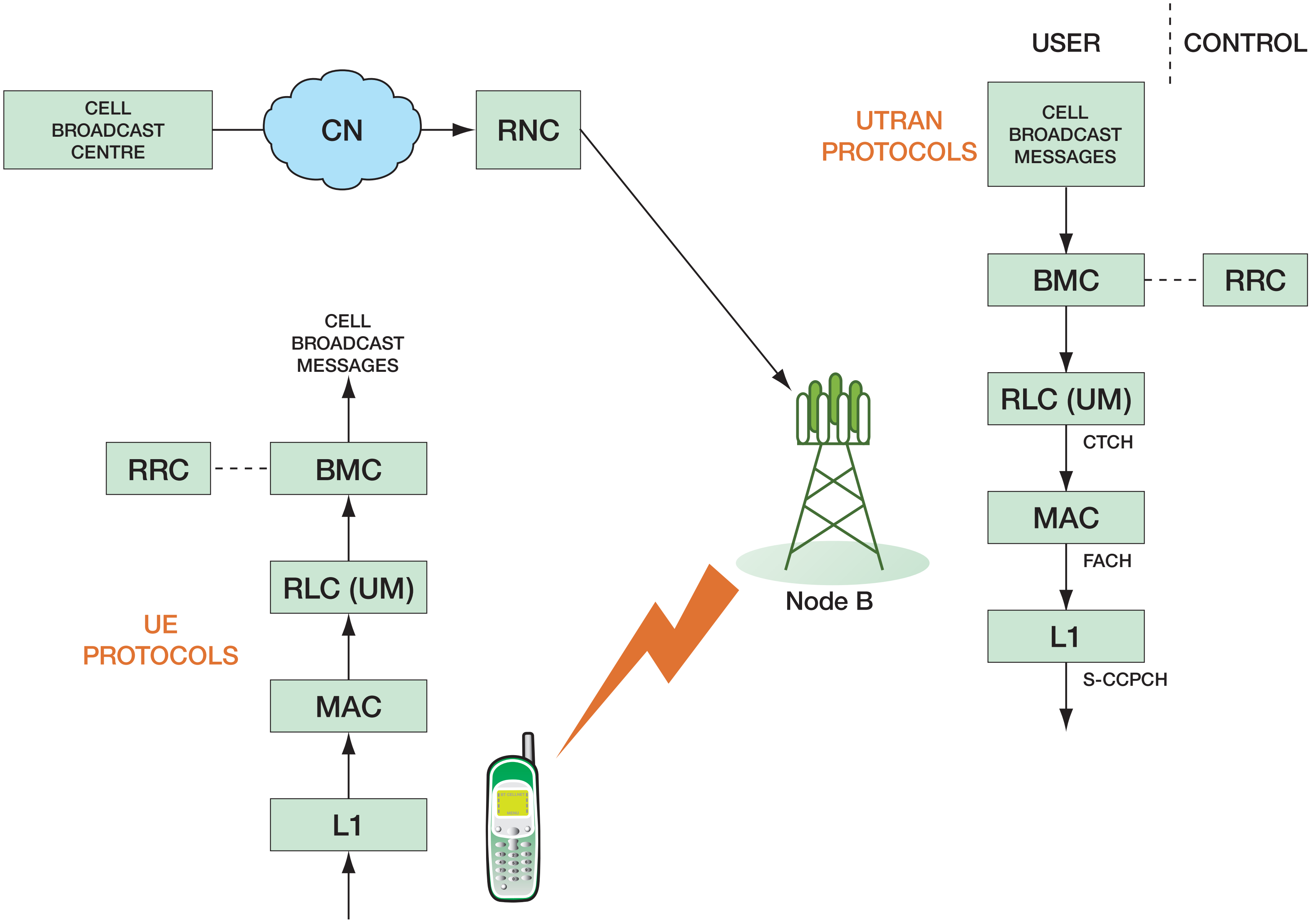
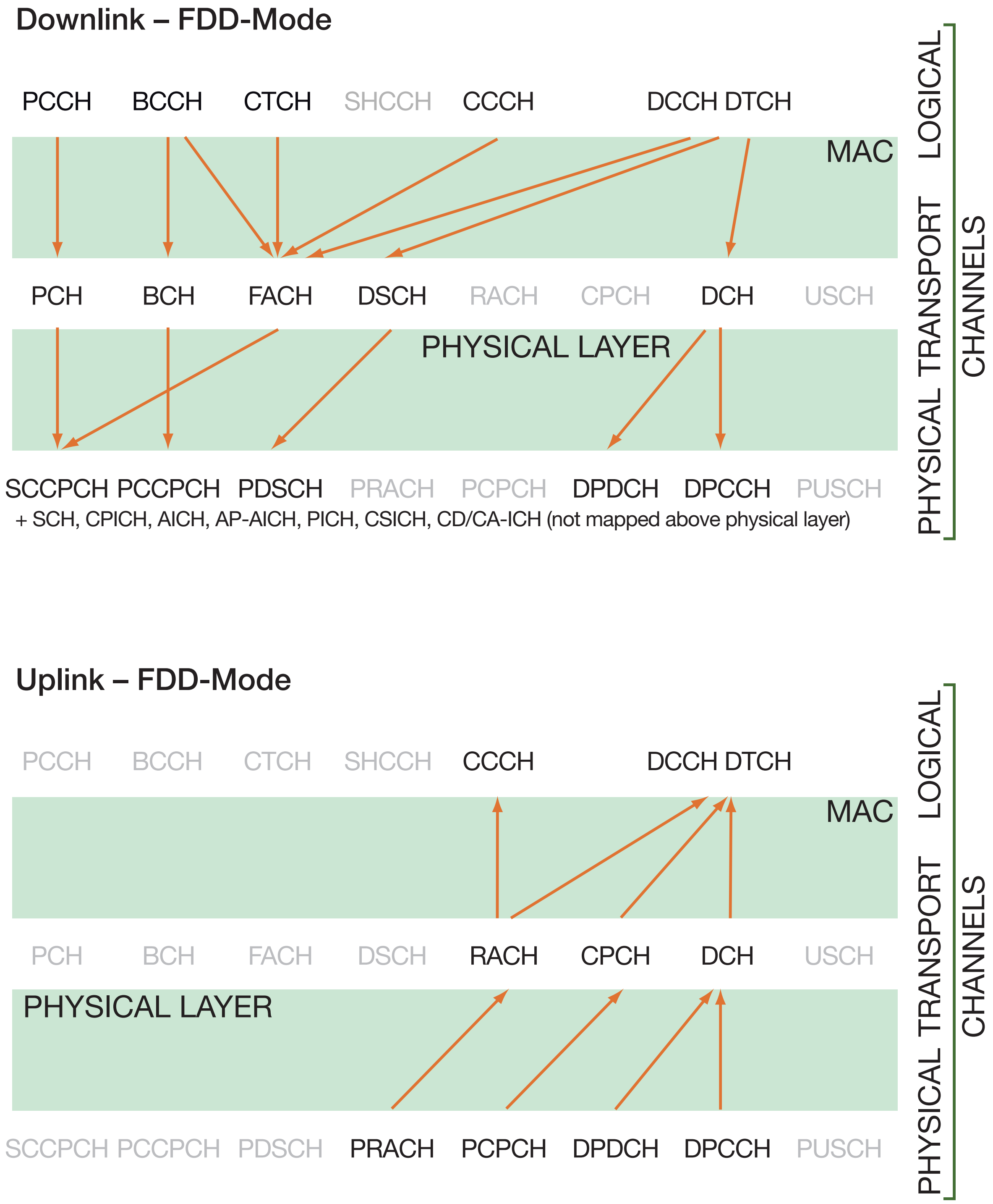


Fig. 22 – Cell Broadcast and BMC

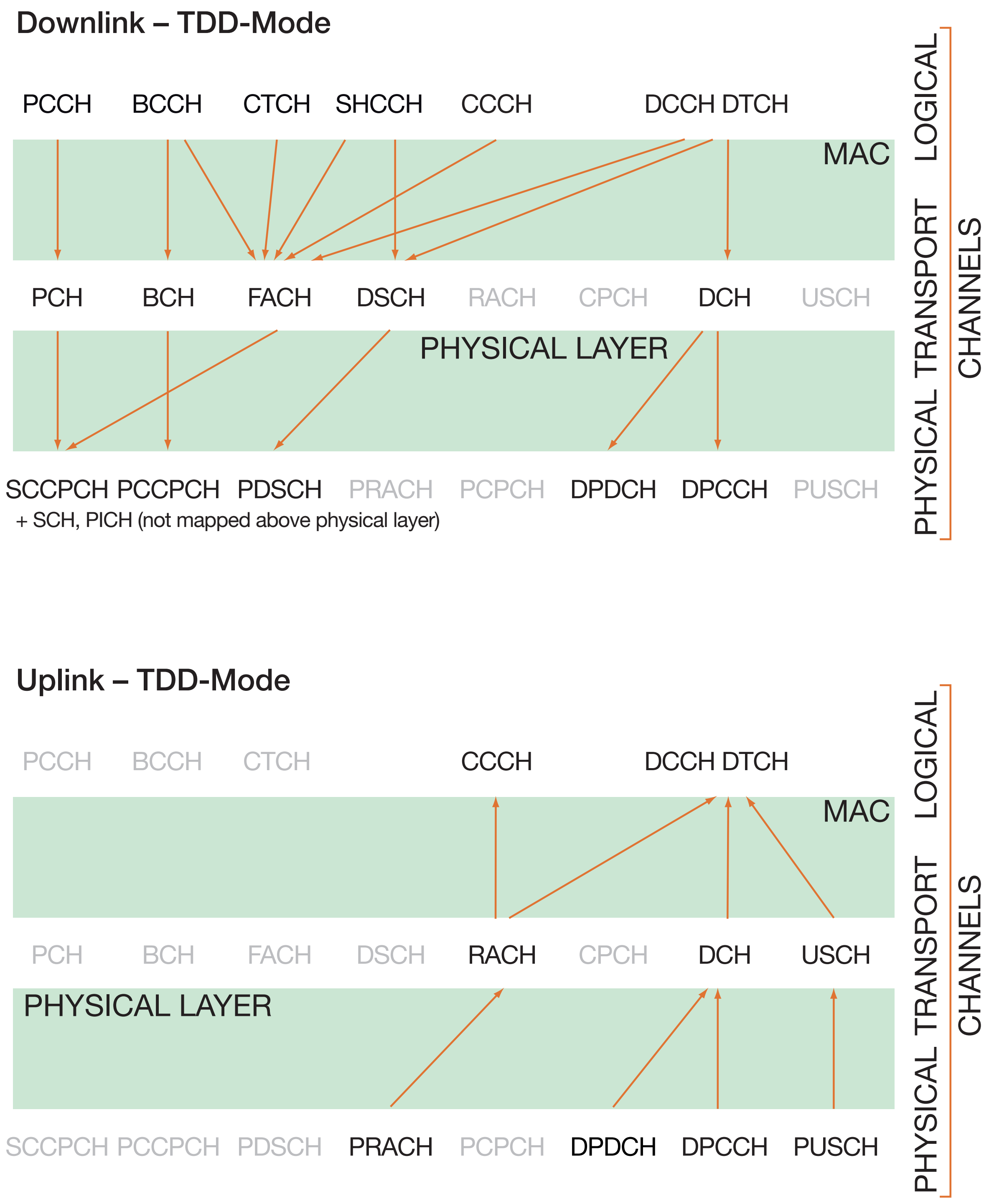
APPENDIX
Summary of channel mapping



Downlink – FDD-Mode

Uplink – FDD-Mode

Fig. A1 – Channel Mapping – A Summary



Downlink – TDD-Mode

Uplink – TDD-Mode

Fig. A2 – Channel Mapping – A Summary