FFI-rapport 2008/01921

The design of an initial NBWF network simulator

Tore J. Berg

Forsvarets forskningsinstitutt/Norwegian Defence Research Establishment (FFI)

November 24th, 2008

FFI-rapport 2008/01921

1088

P: ISBN 978-82-464-1461-4
E: ISBN 978-82-464-1462-1

Keywords

Modellering og simulering

Mobile datanett

Trådløse lokalnett

Approved by

Vivianne Jodalen	Project Manager
Vidar S. Andersen	Director

English summary

The NATO sub-committee 6 ad-hoc working group 2 has an ongoing activity with the objective to develop a narrowband waveform (NBWF) standard. This is a single-channel mobile ad-hoc network (MANET) which shall serve both voice traffic and data traffic over a 25 kHz radio channel. This document is a contribution to the NATO NBWF activity and is the first step towards the development of a simulator by which we can study the performance of a simplified NBWF network in different operating scenarios.

The simulator is primarily designed to study network protocols in a network that shall serve multicast voice traffic and data traffic. A protocol stack for the NBWF has not yet been specified and this document specifies a reference model and a simplified protocol stack covering layers 2 to 7 suitable for implementation in a simulator. The research focus is layer 2 and layer 3 protocol functions but the simulator must, of course, include the other layers. The last part of this document describes how a network of nodes can be modelled in a simulator and also gives the current status for the simulator implementation activity.

Sammendrag

Dette dokumentet er en leveranse fra modellerings- og simuleringsaktiviteten under TIPPER-prosjektet. Prosjektet deltar aktivt i et standardiseringsarbeid i NATO som skal ta fram en smalbånds bølgeformstandard. Hensikten med dette dokumentet er å designe en simulator som kan modellere et antall smalbånds radionoder i et nettverk. Da det per i dag ikke er spesifisert noen protokollstakk for et slikt nettverk, spesifiserer første del av dokumentet et sett med forenklede protokoller. Deretter beskrives en overordnet datastruktur for en simulator som kan implementeres med det simulatorverktøyet prosjektet skal benytte.

Contents

1 Introduction

The NATO sub-committee 6 ad-hoc working group 2 has an ongoing activity with the objective to develop a narrowband waveform (NBWF) standard [1]. This is a single-channel mobile ad-hoc network (MANET) which shall serve both voice and data traffic over a 25 kHz radio channel.

This document is a contribution to the NATO NBWF activity and is the first step towards developing a simulator by which we can study the performance of a simplified NBWF network. A very interesting research question is the capacity left for data traffic in a network with one active multicast voice stream when using a radio channel of size 25 kHz.

The simulator shall be designed to model a network of any size, and shall implement traffic generators[1] that model multicast voice and data traffic. The simulator shall implement estimators to give performance measures of throughput and delay of data traffic under different operating scenarios. A set of estimators shall also quantify the quality of the voice service (the loss rate of voice packets, call setup delay, etc.).

The purpose of this document is to develop a simplified protocol stack for serving TCP/UDP traffic and Multicast Voice (MV) traffic detailed enough to facilitate implementation of a radio node in a simulator. The stack shall be implemented in a simulator covering OSI layer 1 to layer 7 [2], and we shall use an object oriented modelling technique. The latter means the node structure should have a structure similar to a real radio node.

The radio node will be a very complex unit and we must have a good reference model [2] before the protocol design can start. The reference model presented is based on the reference models from two existing systems, the Multi-Role Radio (MRR) [15] and the Universal Mobile Telecommunication System (UMTS) [5]. The air interface between the UMTS mobile terminal and the base station serves both data traffic and voice traffic over a Time Division Multiple Access (TDMA) based Medium Access Control (MAC) protocol. Note that the UMTS protocol functions are not relevant for NBWF since the system environments are different[2].

The NBWF MANET shall serve real-time traffic (voice) and the MAC must be designed accordingly. This document does not discuss different MAC protocols but assumes that the MAC protocol is based on TDMA. The initial MAC protocol design is described in [10].

This document starts with a general discussion of data traffic and voice traffic. Here we decide the principles to use for establishing voice connections and introduce some simplifications to reduce the simulator's software complexity. Chapter 3 presents the reference model which splits the radio node into horizontal and vertical sections.

---

[1] The generators shall be located above OSI layer 7 and emulate the usage of the services.

[2] For example, the base station and the mobile stations work in complementary operation and are not identical. The radio nodes in NBWF are identical.

The purpose of chapter 4 is to design the signalling system for setting up multicast voice (MV) connections. MV-traffic must use reserved TDMA slots on each relaying node on the end-to-end path to be able to fulfil the delay requirements. The procedures for connection setup are very complex in contrast to the procedures for handling the MV-traffic during the data transfer phase as outlined in chapter 5. Unicast voice is not addressed since this traffic type is less demanding.

Packet data traffic requires no connection setup phase[3]; a fact that reduces the protocol complexity and chapter 6 describes the protocol stack for serving best-effort data traffic.

Chapter 10 takes a “real” radio network and the reference model as a starting point and explains how a network can be modelled in a simulator. The radio node is split into atomic models and these models are implemented in a programming language under the simulation framework named OMNeT++ [7, 12]. This chapter also introduces some modelling simplifications to ease the protocol design and reduce the simulator’s software complexity. Note that these simplifications lead to optimistic performance results compared to a real system.

Chapter 11 “Conclusions and Remarks” presents some observations made when using the reference model both as a reference for protocol design and implementation. The chapter also presents the status of the implementation activity.

1.1 Limitations and Development Strategy

The development of the simulator is based on an existing MANET simulator framework developed at FFI, which is a set of software components developed to model radio networks and conduct simulation experiments. The software supports the entire “life-cycle” of modelling and simulation – functions to implement models of real objects, functions to configure a network, functions to conduct debugging and functions to produce simulation reports. Our strategy is to make changes and enhancements on this software to meet the requirements from the NBWF-project. Note that the goal is to develop a simulator for studying network protocols and we do not intend to model the radio at “bit level” [14].

The first phase of the project is to specify an initial radio solution and a TDMA based MAC protocol. These are ongoing activities, partly described in this document, and only **multicast voice and data traffic** are considered. We have foreseen the three development stages for the simulator:

Stage 1: No node mobility[4] with simplified data protocols

Stage 2: No node mobility with enhanced data protocols

Stage 3: Node mobility

[3] The use of reserved or unreserved TDMA slots is decided by the MAC entity itself on a per packet basis.
[4] The simulator uses a fixed routing table calculated at time instance zero.

Stage 1 shall implement the physical layer (radio) in [14], the TDMA protocol in [10] and the protocol stack specified in this document. We have an operating model after stage 1 which is suitable for performance study under fixed network topology. The network dynamics are caused by data and voice traffic, and the intra network protocols[5]. Because the nodes are kept at fixed locations within the playground, the simulator needs only support static routing. Performance analysis using the stage 1 simulator will give us an indication of the network throughput capacity in different static scenarios.

The protocol stack[6] for handling data traffic used in stage 1 is implemented by an existing software package not intended for a TDMA based MAC protocol. Based on the simulation experiments using the stage 1 simulator, we will make better adaptation to an NBWF scenario by developing an improved version in stage 2.

The introduction of mobile nodes in stage 3 demands implementation of dynamic routing if we shall route packets between end-destinations in a multihop network. Mobility also increases the complexity of the simulator, even in modules not dealing with routing.

The remainder of this document encompasses stage 1 only.

2 Data and Voice Traffic

The simulator shall model **multicast voice** and **best-effort unicast data** services at OSI layer 3 in the end-systems shown in Figure 2.1 and the usage of those services. The following supplementary services shall be provided:

- Multi-Level Precedence and Preemption (MLPP).
- A service coverage area larger than the radio coverage area: the intra network layer protocols shall implement relaying.
- Enhanced resilience against packet loss across radio links: the intra network layer protocols shall implement ARQ for best-effort data traffic.

Multicast voice traffic needs other protocol functions than best-effort unicast data traffic, and this chapter discusses some important aspects we must have in mind before the two traffic types can be integrated in a single network. We consider data traffic in section 2.1 while voice traffic is discussed in sections 2.2 and 2.3.

[5] Fx retransmissions of packets
[6] Layer 3a and LLC

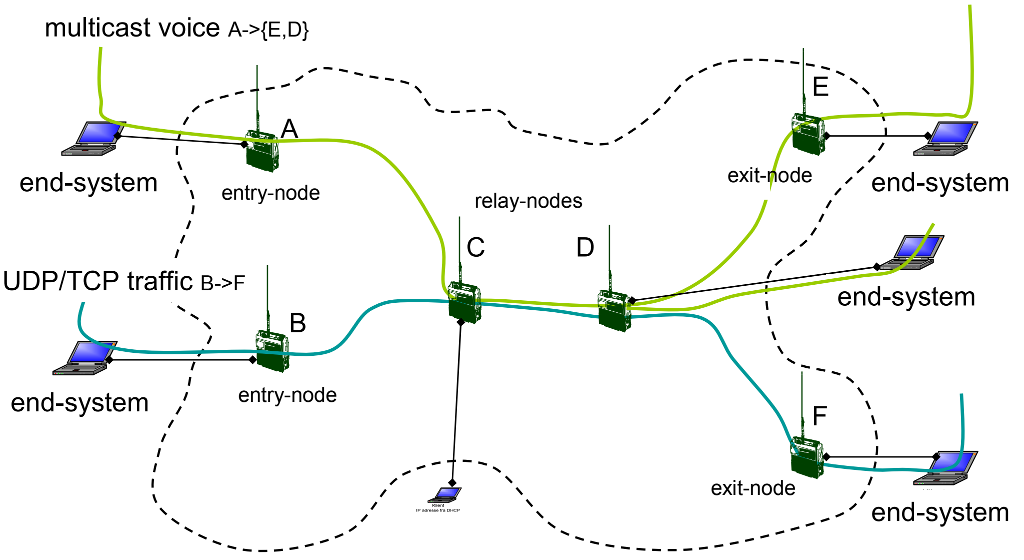

Figure 2.1 The simulator shall model a multihop network serving different traffic types.

2.1 Data Traffic

The simulator shall model the scenario depicted in Figure 2.1 with a varying number of nodes. An *entry-node* in the figure is a radio node which serves traffic from its local terminal equipment while an *exit-node* is a radio node which delivers traffic to the local terminal equipment. A common name for an entry-node or an exit-node is *edge-node*. A node that relays traffic from an adjacent node is named a *relay-node*. A node may take the role as a relay-node and an edge-node simultaneously. For example, node D in the figure operates as a relay-node and an exit-node.

A general network node has two interfaces; a radio based interface and an interface towards the terminal equipment, see Figure 2.2. The figure depicts an IP based access protocol between the network and the end-systems. The wire based interface towards the terminal equipment is not an issue for this document since this interface has infinite capacity compared to the air interface. IP traffic[7] is bursty in the sense that no connection setup and disconnect signalling is received from the end-system, and layer 3 entities are unable to determine the upper layer protocol state[8].

[7] We use the term "IP traffic" to mean a stream of TCP packets or UDP packets *without* real-time requirements.
[8] If the edge-node implements a TCP proxy, we could determine the start and stop of TCP sessions.

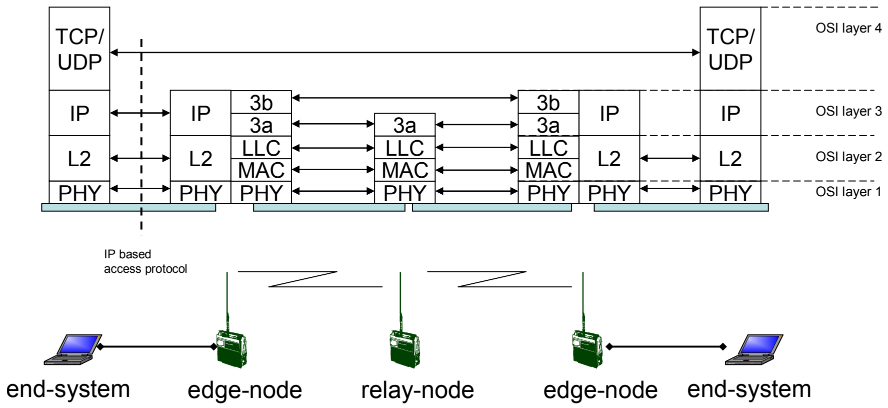

Figure 2.2 Protocol architecture.

We have an existing simulator for a MANET [15] and want to reuse the existing software as much as possible during the initial design. This protocol stack cannot serve voice traffic. NBWF shall use another radio modem [14] and a TDMA protocol. The protocol stack may serve data traffic but is not optimised for these conditions.

The 3b layer protocol may perform end-end-to signalling of intra network information (e.g., end-to-end flow control) and is not included in the simulator. The 3a layer protocol performs store and forwarding operation in a multihop network and the following protocol functions are implemented for **best-effort unicast data** traffic:

- Data transmission using ARQ and passive/implicit acknowledgement
- Data transmission without ARQ
- Duplicate filtering
- Lifetime control
- Precedence and preemption
- Segmentation and reassembly
- Relaying
- Flow control

The LLC layer delivers data across a single link and the following protocol functions are implemented for **best-effort unicast data** traffic:

- Data transmission with ARQ using a selective repeat protocol with window size 2
- Data transmission without ARQ
- Lifetime control
- Precedence and preemption

The MAC layer protocol is a TDMA based protocol since the system shall serve delay sensitive applications. MAC uses dynamic TDMA which means that TDMA slots are reserved by the nodes only when they have traffic. [13] gives a general review of TDMA while [10] proposes a

TDMA protocol for NBWF. Real-time traffic must be sent over reserved TDMA slots to meet the delay requirements. Non real-time data traffic may be sent on unreserved slots as well as reserved slots. The benefits of using unreserved TDMA slots instead of reserved slots, seen from a traffic engineering point of view, is less signalling and a higher multiplexing gain. The drawback is no guaranteed end-to-end transit delay[9].

2.2 Multicast Voice

The broadcast voice service is similar to a CNR-voice[10] application with the extension that voice traffic shall be received even by users outside the radio coverage area of the originator. This implies the network must relay this type of voice traffic. A multicast voice service is a specialisation of the broadcast voice service - the traffic shall be received by a specified subset of the radio nodes. Figure 2.1 illustrates how different traffic types appear at the edge nodes and are relayed by the relay nodes. To deliver the voice traffic at constant delay, the MAC layer uses reserved TDMA slots. The MAC layer must assign dedicated slots for the end-to-end traffic streams, and the relay nodes must be able to differentiate between the end-to-end streams such that each stream can be sent over the correct slot.

Multicast voice is a connection-mode service where the originating edge node initialises a call setup upon a push-to-talk (PTT) event. The multicast voice service is characterised by the following properties:

- It is implemented by means of the MAC connection mode service [3, section 5.5.1].
- It is unreliable in the sense that voice packets are not retransmitted in case of loss.
- It has a one-to-many traffic pattern.
- The implementation shall utilise the broadcast[11] nature of the radio channel. That is, the originating edge-node does not send multiple copies, but any MAC layer entity on the path uses a one-to-multipoint connection[12] if the network topology demands (the MAC layer in node D in Figure 2.1 does exactly this).
- A call setup fails on a particular link when the serving node has no free resources for serving the call.
- There is no guarantee that all the addressed end-destinations receive the call.

The signalling for setting up connections is much more complex than the protocol functions used in the data transfer phase and therefore we look at the data transfer phase first. When the connection setup has been completed successfully as outlined in section 2.3, the multicast voice service enters the data transfer phase. Consider the two different cases in Figure 2.3 a) and b), where the reservation phase has determined that the voice shall be sent along the paths marked as thick blue lines.

[9] The jitter may be high.
[10] Combat Net Radio (CNR).
[11] A single transmission may reach all nodes within the radio coverage area of the sender.
[12] Named multi-endpoint-connection in [2, section 5.3.1.4].

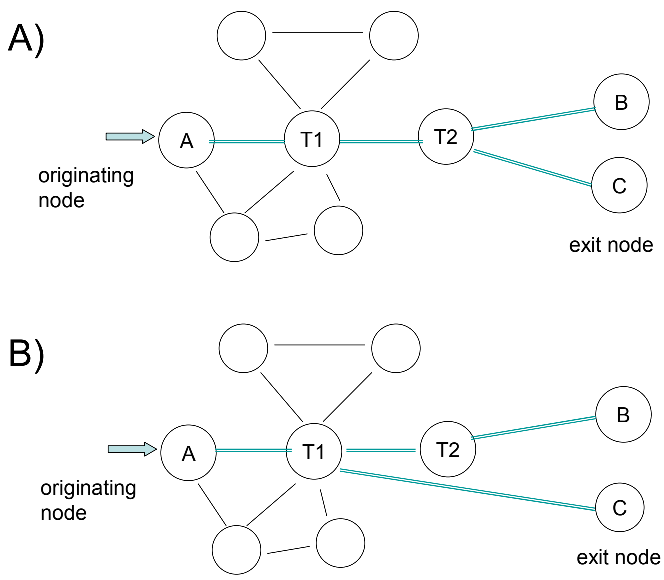

Figure 2.3 The transit-nodes T1 and T2 shall relay traffic to B and C.

In figure a) the transit node T2 is RF-connected[13] to B and C, and T2 can therefore deliver data to many destinations in a single transmission on the same TDMA slot. This is accomplished by using the one-to-multipoint MAC service (Figure 2.4)[14]. This is not possible in figure b) where the exit nodes are served by different transit nodes. Here T1 establishes a one-to-multipoint connection to T2 and C (Figure 2.5), while T2 sets up a one-to-one MAC connection to B.

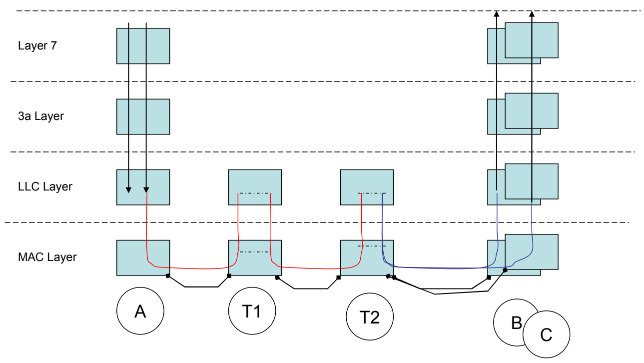

Figure 2.4 LLC relaying in Figure 2.3 a). T2 uses LLC relaying by means of a MAC one-to-multipoint connection (blue lines). T1 uses LLC relaying over a MAC one-to-one connection (red lines).

[13] RF-connected means the signal-to-noise radio (SNR) is sufficiently high for successful transfer of packets.
[14] Chapter 8 deals with one-to-multipoint connections.

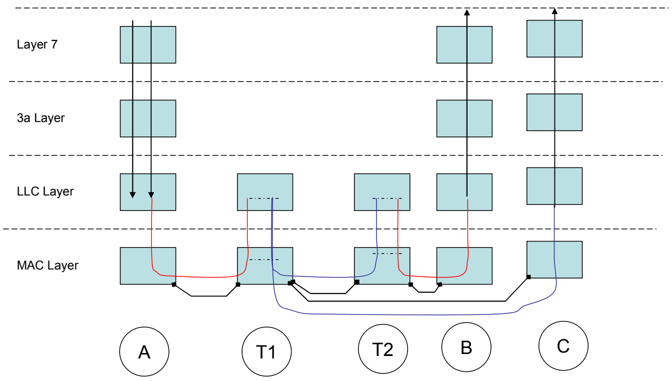

Figure 2.5 LLC relaying in Figure 2.3 b). T1 uses LLC relaying over a one-to-multipoint MAC connection, while T2 uses LLC relaying over a one-to-one MAC connection.

Relaying within the LLC layer is easy. Just establish a map between incoming and outgoing connections during the connection setup phase. Based on the connection endpoint identifier on incoming traffic, the LLC entity finds the connection identifier to use for outgoing traffic by a simple table lookup[15] as shown in Figure 2.6.

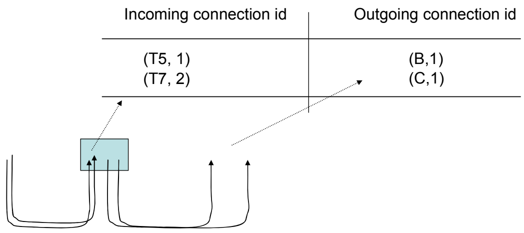

Figure 2.6 A table describes the mapping between incoming and outgoing connections. An outgoing connection can also be the upper layer (i.e., traffic to the local terminal).

2.3 Principles of MV Call Setup

The most complex part of the multicast voice service is the connection setup phase. We have a short time limit from the incoming PTT event until the first voice packet arrives at the entry node. With the use of the hop-by-hop connection setup strategy in Figure 2.7 a), the relay nodes inspect their local resource situation and give a positive response before the entire path towards the end-destination(s) is ready. When the voice packets start to flow, some nodes on the path may have to discard the packets caused by lack of resources, or time to complete the setup. The opposite strategy is to set up the complete end-to-end path, as shown in Figure 2.7 b), before voice packets are accepted. We select the hop-by-hop strategy since the other strategy introduces excessive connection establishment delay. The drawback of the strategy selected is that the originating node gets no feedback upon call setup failure beyond the local neighbourhood.

[15] A none existing entry is interpreted as ”to upper layer”.

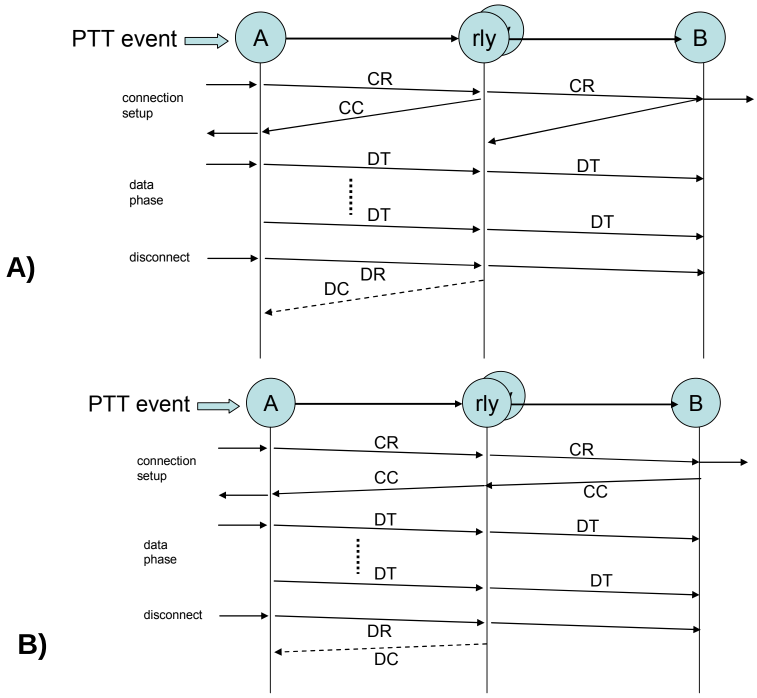

Figure 2.7 Illustration of two connection setup strategies: A) Hop-by-hop signalling B) End-to-end signalling

Consider the situation depicted in Figure 2.8 where a terminal attached to the edge-node A invokes a multicast voice setup to the exit-nodes {B,E,H}. When the call request reaches E with this multicast group, node E cannot know if another node already has started (or will start) a forwarding of the call setup to the other exit-nodes. One possible strategy is to let many nodes do connection setup simultaneously and then release the resources that became redundant. However, this generally leads to much signalling traffic and we have to find a better solution.

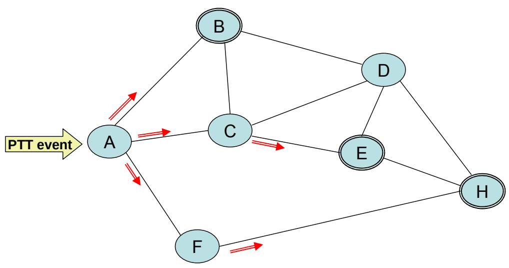

Figure 2.8 A multicast PTT event addressing the multicast group {B,E,H}.

If the link cost for all links is one in the topology shown, node A's shortest path spanning tree for the multicast group is {AB,ACE,AFH}. The second strategy we outline is based on this spanning tree and a source routing scheme. Node A initialises the call setup by sending a connection setup packet containing the following elements:

Multicast group: {B,E,H}
Relay node set: {CE, FH}
CC node set: {C,F}

This packet is sent as a broadcast packet within the MAC layer so all the neighbours can receive it. The relay node set tells which nodes shall take the responsibility of relaying the request as well as the path to use. The Connect Confirm (CC) node set determines which nodes shall return a confirmation packet back to the sender. This signalling informs the neighbourhood about a successful reservation while the local issuer can determine the outcome of the call request sent.

The processing at node B is as follows:

- I shall not relay
- I am an exit-node so allocate local resources
- I shall not issue a CC

The processing at node C is as follows:

- I shall relay, schedule a connection setup packet to node E with the following elements:
  Multicast group: {B,E,H}
  Relay node set: {}
  CC node set: {E}
- I am not in the multicast group so operate as a pure relay-node
- I am in the CC set, schedule a confirmation packet for transmission

The processing at node F is as follows:

- I shall relay, schedule a connection setup packet to node H with the following elements:
  Multicast group: {B,E,H}
  Relay node set: {}
  CC node set: {H}
- I am not in the multicast group so operate as a pure relay-node
- I am in the CC set, schedule a confirmation packet for transmission

The processing at node E and H is as follows:

- I shall not relay
- I am an exit-node so allocate local resources
- I am in the CC set, schedule a confirmation packet for transmission

The processing at originator A depends on the feedback from the neighbourhood. If all the CC packets are lost then we should have an error recovery mechanism that initiates a new setup. However, node A can overhear the transmissions within its neighbourhood. If the relaying nodes tag their outgoing CR with an identifier known by A, A can take this as a confirmation of receipt when none of the CC packets are received. The short time limit available to recover from call setup errors and the potentially large channel load introduced by an error recovery mechanism makes error recovery a difficult subject. The description of the error recovery mechanism is insufficiently specified in this document and is a subject for further study. However, we can give some conclusions about the call setup procedure proposed.

The benefits of this strategy:

- No other schemes can offer less signalling traffic measured in number of packets
- Circumvents the problem of looping by letting the entry-node set the full relay path in the relay node set

The drawbacks of this strategy:

- The connection setup packet becomes large since the relay node set must be included
- The shortest path spanning tree may be outdated (or need fast updates) in case of node mobility

Possible improvements:

- The multicast group set can be predefined and identified by a few digits

The call setup scheme must be developed in close coordination with a routing protocol. However, we have yet not started work on routing. Therefore the first version of the simulator shall implement the strategy outlined above (remember that this version does not support node mobility). The simulator calculates the minimum spanning tree[16] for all the radio links at time zero, and the tree is valid during the entire run-time since we have fixed topology.

If the TDMA protocol fails to prevent collisions on the radio channel, the hidden node problem may degrade the network performance. A countermeasure to this situation might be to let the originating node build the hidden node neighbour set for each possible relay and then select the CC-set based on the relays with the minimum number of common nodes to maximise the number of nodes that get information about a reserved TDMA slot.

3 The Reference Model

This chapter proposes a node architecture that is able to provide best-effort unicast data and multicast voice services. All entry-nodes must be able to differentiate between the traffic streams coming from the terminals. This is illustrated by Figure 3.1 where the 3a layer offers a separate Service Access Point (SAP) [2] for each type of service; best-effort data or multicast voice.

[16] Based on a link cost graph where the edge cost function is based on the SNR level given one single transmission in the network.

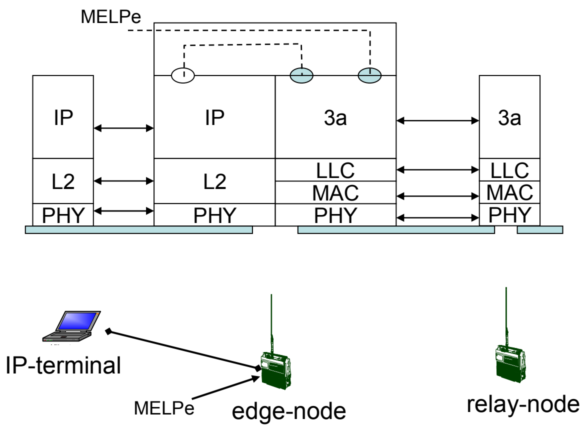

Figure 3.1 The 3a layer provides an SAP for each type of service.

The network shall serve voice and data over a TDMA based MAC protocol. A communication system that does this is UMTS [5]. We have studied the UE-BaseStation interface, see Figure 3.2, and the proposed reference model and our terminology is based on UMTS.

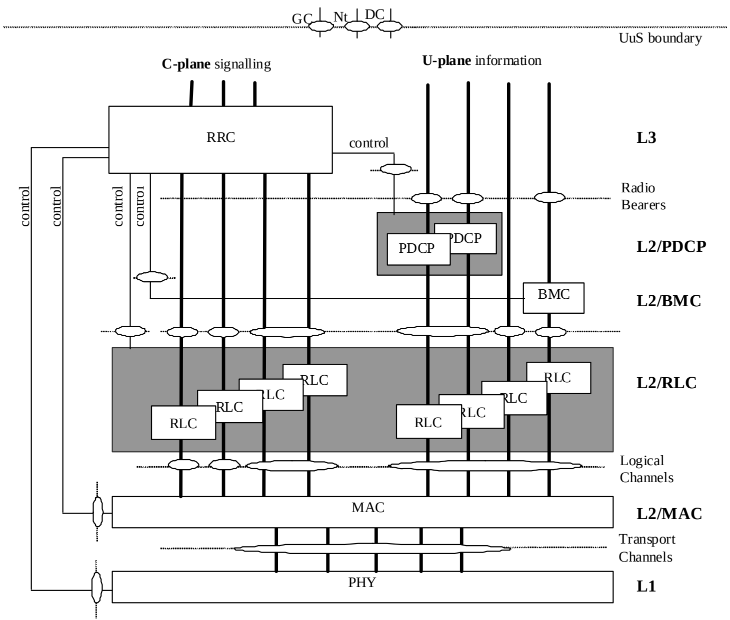

Figure 3.2 UMTS UE protocol stack for the radio interface (Uu) to the base station [5].

This document has focus on the intra network layer protocols and we skip the 3b layer in Figure 2.2. Moreover, to simplify the presentation layer 4 is placed on top of layer 3a as shown in Figure 3.3.

The OSI Reference Model [2] divides a system into horizontal layers. UMTS, and some other standards (e.g. [6]), introduces a vertical "layering" named control plane (C-plane) and user plane (U-plane). The control plane transfer information for the user plane connections. This includes call setup and disconnect as well as resource reservation/de-allocation. The C-plane establishes connections for the U-plane, and the Connection Oriented (CO) mode MAC service is, of course, not available in the C-plane. A significant point to notice is that the separation of the C-plane traffic and the U-plane traffic takes place above layer 3.

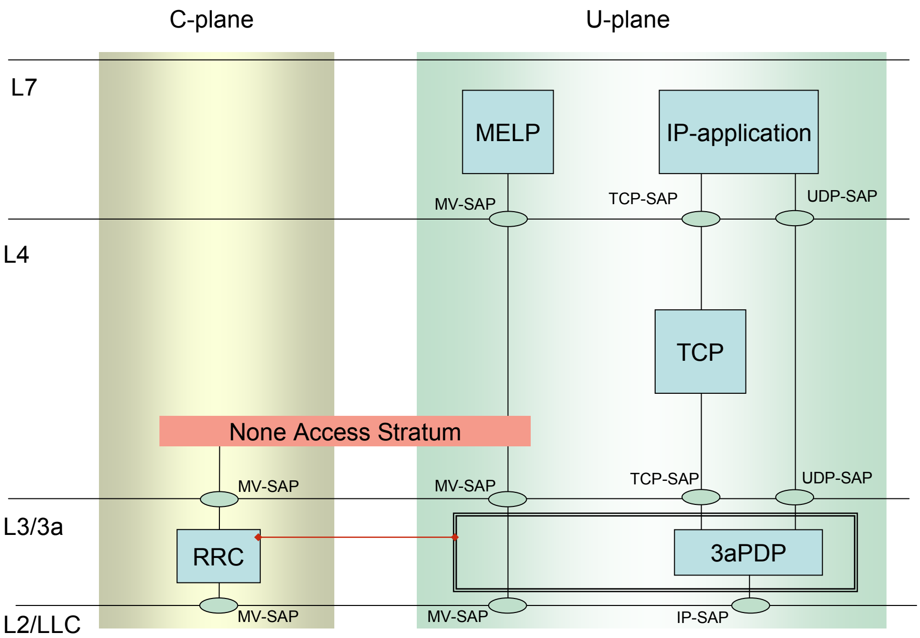

Figure 3.3 The protocol architecture for the upper layers in the radio node. The red lines indicate local control channels for the coordination of the two vertical planes. The None Access Stratum (NAS) splits/combines signalling traffic and data traffic.

The 3a sublayer provides a set of SAPs, referred to as *data bearers*, each designed for a particular service. Data bearers are provided by the 3a service provider for the transport of traffic from a radio node to one or more remote radio nodes. A CO-mode [3, section 5.5.2] bearer may be unidirectional or bidirectional. If a CO-mode bearer is unidirectional then only the initiator is allowed to send data on the bearer.

Currently, we have identified the need for the following data bearers:

Multicast Voice Data Bearer
A CO-mode bearer for transporting multicast voice traffic from the local terminal to two or more remote terminals. This bearer is unidirectional and only the entry-node (the owner of the PTT event) is allowed to send data.

TCP and *UDP* Data Bearers
A CL-mode [3, sec 5.5.1] bearer for transport of IP traffic between the terminal equipment.

Unicast Voice Data Bearer
A CO-mode bearer for transporting point-to-point voice traffic between two terminals.

Network Management Data Bearer
A CL-mode bearer for transport of intra network management traffic.

The two last data bearers are not shown in the figure above. A CL-mode bearer is available when a radio node has become a member of the network - no connection setup in the C-plane is needed.

The Radio Resource Control (RRC) protocol is the most important component for signalling. The RRC entity operates in the C-plane, has no counterpart in the U-plane, and executes the following procedures:

- Connection setup and disconnection
- Congestion and admission control
- Precedence and preemption handling
- Relaying of call setup messages

These procedures are applied to CO-mode signalling bearers only (MV-SAP, UV-SAP). The CL-mode data bearers (TCP/UDP-SAP, NM-SAP) demand no signalling in the C-plane because connection setup and disconnect are not required.

The functions of the 3a Packet Data Protocol (3aPDP) is identical with the 3a protocol outlined in section 2.1.

Figure 3.4 shows the architecture for the lower layer protocols. The LLC layer contains a protocol which operates in one of three modes: Transparent Mode (TM), Unacknowledged Mode (UM) or Acknowledged Mode (AM). The LLC-TM is designed to serve voice traffic and tolerates bit errors since no LLC Protocol Control Information (PCI) is added. This in contrast to the LLC-UM/AM which fails if MAC delivers a payload containing bit errors.

The MAC layer operates a set of logical channels with certain properties, determined by the requirements needed by the upper layer protocols. The C-plane must have a Common Control channel (CCCH) for signalling. The U-plane needs two channels to serve voice traffic - a one-to-One Traffic channel (OTCH) and a one-to-Multipoint Traffic channel (MTCH)[17]. IP traffic is served on a Random-Access Traffic channel (RATCH).

The MAC protocol divides the TDMA frame into a set of transport channels. The traffic on the logical channels is multiplexed down to the transport channels but this operation is invisible to the MAC service user.

[17] Subject for further study: Consider to merge these two channels into one.

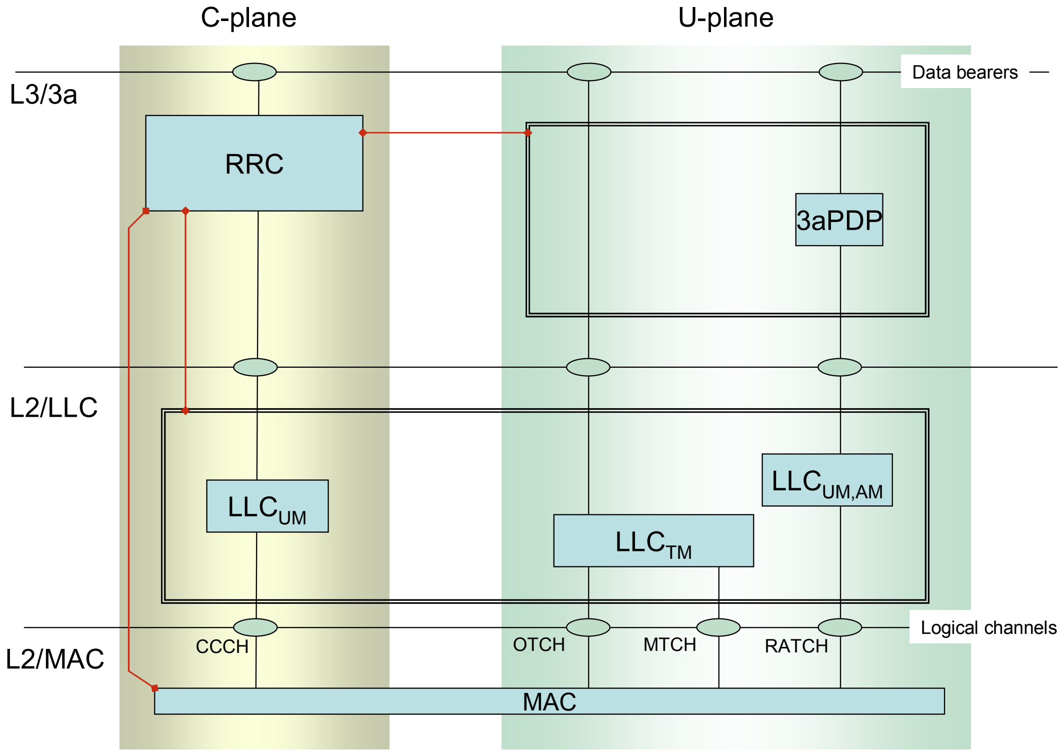

Figure 3.4 The protocol architecture for the lower layers in the radio node.

The LLC entities access the logical channels via Service Access Points (SAPs) which are assigned identical names (identifiers) with the MAC logical channels. MAC then knows the logical channel type in use, and by including the SAP-ID in the MAC PCI, the receiving MAC entity is able to determine the destination MAC SAP to use. Traffic from different logical channels must never be mixed.

We define the following MAC SAPs:

OTCH-SAP, MTCH-SAP
A CO-mode SAP in the U-plane for transport of voice traffic.

CCCH-SAP
A CL-mode SAP which is used in the C-plane to transport signalling information on the behalf of all the upper layer protocol entities in the U-plane.

RATCH-SAP
A CL-mode SAP in the U-plane for sending delay-insensitive data traffic.

Figure 3.5 presents the resulting addressing scheme for the network. All radio nodes are assigned a unique global identifier - a MAC address which identifies the PHY-SAP (the physical radio equipment). The protocols above the physical layer include SAP identifiers in their protocol control information and the incoming traffic can then be routed to the correct layer 3 SAP.

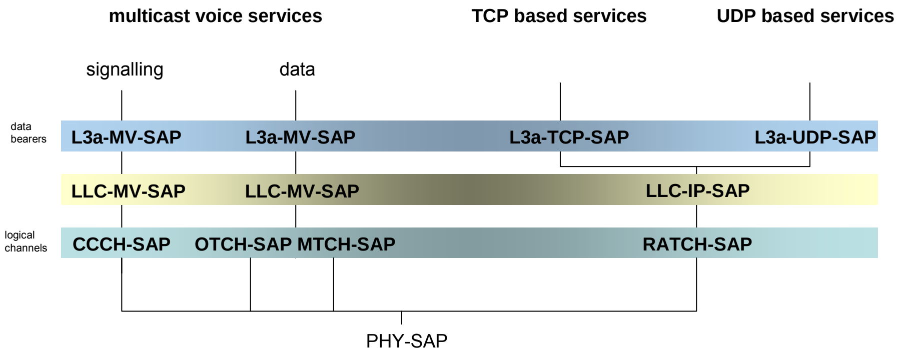

Figure 3.5 SAP address mapping. The PHY-SAP is identified by a unique global address assigned to each network node.

4 Multicast Voice and RRC

The Radio Resource Control (RRC) protocol entity operates in the control plane and deals mainly with resource allocation and release. The RRC entity has a very central position since it executes admission control for fresh traffic as well as transit traffic with regard to connection setup. The RRC entity may perform functions such as:

- Establishment, maintenance and release for the CO-mode services for the U-plane
- Precedence and preemption
- Affiliation to a new network
- Congestion control
- Admission control
- Error recovery
- Inactivity control

This chapter specifies the protocol functions applied to traffic passing through the signalling data bearers. Currently, we have defined a single signalling data bearer for multicast voice that shall be mapped to the logical channel named Common Control CHannel (CCCH).

This chapter is structured as follows. The two first sections deal with the connection setup phase and the disconnection phase, respectively. The last section states some principles with regard to error recovery procedures.

4.1 MV Connection Setup

The purpose of this function is to establish a connection in the U-plane MV-SAP. Below we consider the interlayer signalling instead of an isolated specification of the RRC protocol because this presentation form makes it easier to understand the coordination between the different layers as well as the intention of the functions introduced.

Connections do not exist in the C-plane but in the U-plane. Therefore we cannot speak about RRC connections but we instead introduce the term “RRC association” to identify the agreement between peer RRC entities of an established LLC connection. An RRC association unambiguously identifies an LLC connection.

The sequence diagram for establishing an RRC peer-to-peer association is shown in Figure 4.1, and the forthcoming presentation is based on this figure. Recall that the 3a layer protocol is empty for voice traffic.

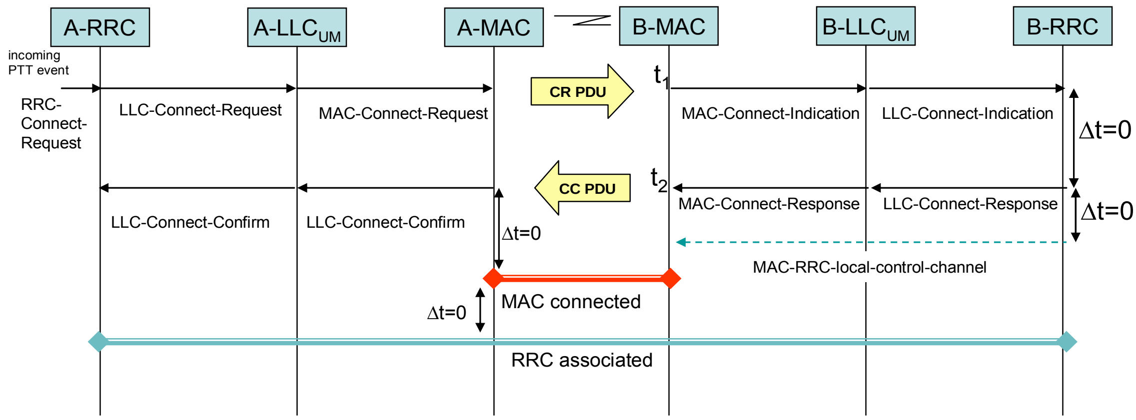

Figure 4.1 *Interlayer service primitive sequence diagram. If the B-RRC entity is not included in the CC-set, a CC-PDU shall not be sent, and the local MAC entity is ordered to allocate the resources through the internal node control channel.*

The signalling starts when the RRC-user issues an RRC-Connect-Request, and the table below presents the parameters in the RRC connect service primitives provided at the signalling MV-SAP.

Parameter	request	indication	response	confirm
Destination Multicast Group	M	M(=)		
Priority	M	M(=)		
Lifetime	M	M	M	M
Payload	O	O(=)	O	O(=)
Interface Control Information (ICI)				
Connection Endpoint Identifier (CEID)	M	M(=)[18]	M(=)	M(=)

Table 4.1 *RRC-Connect primitives and ICI. M means mandatory while O means optional, cf. [3]. “(=)” means the value of the parameter is semantically identical to the corresponding parameter in the preceding related primitive.*

The *destination multicast group* identifies the destination MV-SAPs in the exit-nodes to be included in the call setup procedure. *Lifetime* specifies the maximum lifetime in seconds of the

[18] They are semantically identical but may have different numerical values

signalling packets within the network. *Priority* specifies the importance of the call in the connection setup phase, the data transfer phase and the disconnection phase.

The RRC-Connect-Request results in an RRC-CR-PDU carrying the following elements:

```
message RRC-CR-PDU
{
fields:
  // Layer PCI
      int constPduTypeBit;
      int dstSAP = L3a-MV-SAP;
      set mGroup  = {...};          // multicast group
      set ccSet = {...}             // connect confirm
      set relaySet = {...}          // relay node set
}
```

An RRC-CC-PDU is redundant since the MAC service provider shall issue MAC-Connect-Confirm for each of the MAC-CC-PDU received over the air interface.

The RRC entity performs no error recovery of lost RRC-CR-PDUs since retransmission is a problem due to the call setup delay requirement [1]. However, if the performance study discovers degradation of the voice service caused by lost CR-PDUs then we have to find a solution. For example, replace the LLC-UM by the LLC-AM.

This section describes the connection setup phase as if the LLC-layer is empty[19], and RRC interacts directly with the MAC layer. The MAC-Connect primitives support transmission of a payload, see the table below, and the RRC-CR-PDU is carried by the MAC-Connect-Request primitive. This is beneficial since only one radio transmission is needed to deliver two CR PDUs to the remote site(s).

Parameter	request	indication	response	confirm
Destination address	M	M(=)	M	M(=)
Source address	M	M(=)	M	M(=)
Priority	M	M(=)	M(=)	O(=)
Lifetime	M	M	M	M
Resource	M	M(=)	M(=)	O(=)
Payload	O	O(=)	O	O(=)
Interface Control Information				
Connection Endpoint Identifier (CEID)	M	M(=)	M(=)	M(=)

Table 4.2 MAC-Connect primitives. The Resource parameter signals the logical channel type (e.g., OTCH) requested by the MAC service user.

When requesting an MTCH, the *destination address* is set to broadcast in the request primitive since all adjacent RRC entities shall have the payload. The *source address* is the global address of

[19] The LLC service primitives are mapped one-to-one to the MAC service primitives. We get away from describing this by assuming that the LLC layer is empty.

the originator and the *resource* parameter specifies the type of U-plane logical channel requested {OTCH, MTCH}.

This document specifies the layout of some of the MAC PDUs only to exemplify the MAC PCI needed within the scope of this document. The MAC protocol will therefore include more PCI fields than shown in this document. The MAC-CR-PDU has the following layout:

```
message MAC-CR-PDU
{
fields:
  // Layer PCI
      int destAddr  = *;            // broadcast
      int srcAddr;                  // global source address
      int destSAP   = MAC-CCCH-SAP;
      int connectionId;

      double remainingLifetime;
      int priority;
      int payload;
}
```

The destination address signals broadcast and all the adjacent MAC entities shall issue a MAC-Connect-Indication. The B-MAC entity receives the MAC-CR-PDU at t_1 in the Figure 4.1, issues an indication primitive built from this CR PDU and takes no further action. The B-RRC entity processes the request, builds an RRC-CC-PDU on a positive outcome, and issues a MAC-Connect-Response at time instance t_2. This primitive contains enough information so the MAC entity does not need to save any information at t_1. The time delay between t_2 and t_1 should be zero (exactly zero in the simulator). The MAC-CC-PDU has the following layout:

```
message MAC-CC-PDU
{
fields:
  // Layer PCI
      int destAddr;                 // originator’s global address
      int srcAddr;                  // responder’s global address
      int destSAP   = MAC-CCCH-SAP;
      int connectionId;

      double remainingLifetime;
      int priority;
      int payload;
}
```

Nodes not included in the RRC-CC-set, cf section 2.3, shall not generate a MAC-Connect-Response at t_2. However, the local MAC entity must be informed if the call setup shall be accepted, and this signalling is conducted in the C-plane through the local control channel. Only positive outcomes shall be reported.

A receiving RRC entity acts as a relay-node or as an exit-node, or both. We introduce the term passive-node to identify events where an RRC entity is not addressed by the RRC-CR-PDU.

A passive-node (RRC entity) shall register the event and take no further action. If the receiving RRC entity is included in the multicast group then it shall act as an exit-node for this call setup. If the receiving RRC entity is included in the relay set then it shall act as a relay-node for this call setup. If the RRC entity is included in the CC-set, it shall issue an RRC-CC-PDU.

When the RRC entity operates as an exit-node, it shall:

- Always issue an RRC-Connect-Indication

When the RRC entity operates as a relay-node, it shall:

- Send an RRC-Disconnect-PDU if the local resource situation does not permit a new call. (This is a subject for further study since it may introduce too much signalling in a narrow band system. Especially if the node is located many hops away from the A-terminal).
- Otherwise determine the logical channel type (OTCH,MTCH) to use on the next hop and send an RRC-CR-PDU on the next hop.

When the RRC entity operates as an entry-node, it shall:

- If the local resource situation does not permit a new call, issue an RRC-Disconnect-Indication.
- Otherwise determine the logical channel type to use and send an RRC-CR-PDU.

4.1.1 C-plane to U-plane Signalling

The RRC entity establishes MV connections for the U-plane entities and must supply sufficient information to the U-plane entities so they can serve the traffic streams. The figure below depicts an example scene where node *i* serves three MV connections. In the U-plane there exists one and only one LLC entity for voice traffic and the entity is able to serve any number of MV connections. The LLC entity operates above two MAC SAPs and the binding to the SAPs is static, that is, the data structure is valid from time zero (power on).

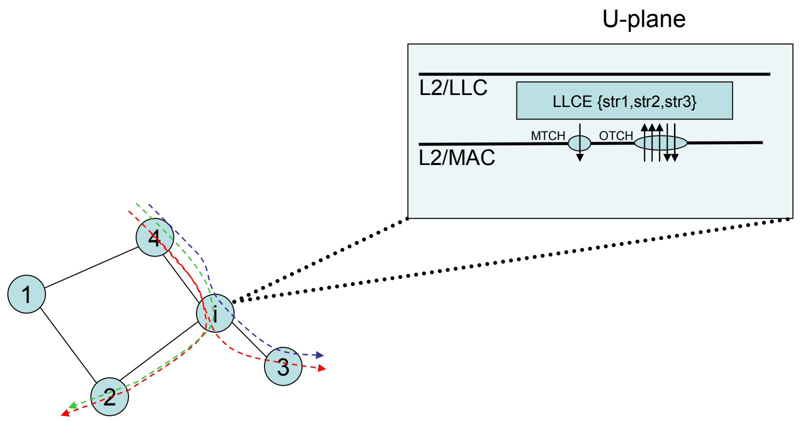

Figure 4.2 An example of multicast voice connections where node i operates as a relay node. Here the LLC entity must maintain state information on a per connection basis and handle three MAC connections concurrently.

The U-plane LLC entity stays inactive if no connections exist, and is informed by the local RRC about a new connection. The MAC entity must maintain state information on a per connection basis as defined by the following data structures:

```
class MacGlobalConnectionId
{
      int globalNodeAddress;
      int sapId;        // identifies the user plane SAP
      int serialNumber; // a unqiue number within an MAC-SAP
};

class MacConnectionCache
{
  MacGlobalConnectionId cid;
  int localCeId; // a unique number over the LLC/MAC interface
  bool isOutgoing;
};
```

The *serialNumber* is a unique number for both the MAC-CCCH-SAP and the corresponding U-plane SAP. This cannot lead to ambiguity since the U-plane does not allocate numbers. Connections are just established over MAC-CCCH-SAPs and then created in the U-plane. There is a one-to-one relationship between MAC connections and L3a-connections, and the upper layer protocols (below layer 4) need not add additional information to identify connections.

To clarify the usage of data structures and the local signalling scheme, we consider the scene above when the blue connection from node *i* to node 3 shall be established. Assume that node-i-RRC just has established the blue MV connection from node 4 to *i*. Then the local MAC entity holds the following information for incoming traffic from node 4:

```
MacConnectionCache c1;

c1.cid.globalNodeAddress = node4;
c1.cid.sapId = MAC-OTCH-SAP;
c1.cid.serialNumber = 15;     // just a number not in use
c1.localCeId = 55;            // a unique number over the LLC/MAC interface
c1.isOutgoing = false;
```

RRC issues a connect request at time instance t_1 in Figure 4.3 with CeId = 7 (say). The MAC entity builds a CR-PDU and adds the new cache entry:

```
MacConnectionCache c2;

c2.cid.globalNodeAddress = node3;
c2.cid.sapId = MAC-OTCH-SAP;
c2.cid.serialNumber = 13;     // just a number not in use
c2.localCeId = 7;             // a unique number over the LLC/MAC interface
c2.isOutgoing = true;
```

At t_3 MAC receives a CC PDU, finds the matching fingerprint in the cache (c2), and then issues a connect confirm primitive using CEID = c2.localCeId (= 7). RRC recognises this number and sends a new connection event (t_4) to the U-plane LLC on the local control channel:

```
class RRC-LLC-newMVconnection
{
  int incomingCeId;
  int outgoingCeId;
};

RRC-LLC-newMVconnection c;
c.incomingCeId = 55;    // the LLC is aware of this before t1
c.outgoingCeId = 7;
emitLocalSignal(c,LLC); // the LLC receives c with zero delay
```

Now the U-plane LLC entity has enough information to use the MAC connections to relay the traffic. Node *i* received a voice packet from node 4 at t_5 and generates an MAC-CO-Data-Request[20] primitive using ceid=7. MAC recognises through the cached data that this is a packet destined for node 3 and builds a MAC Connection Oriented Data (CODT) PDU.

At t_6 the local RRC invokes preemption due to a higher priority call (just an example), decides that the blue MV connection must be disconnected, and issues two MAC-Disconnect-Requests using ceid=7 and ceid=55. The cause is set to “preemption due to higher priority call”. Then MAC builds two MAC-DR-PDUs to signal release of reserved slots, and cleans up the cached data.

[20] On the OTCH or the MTCH SAP.

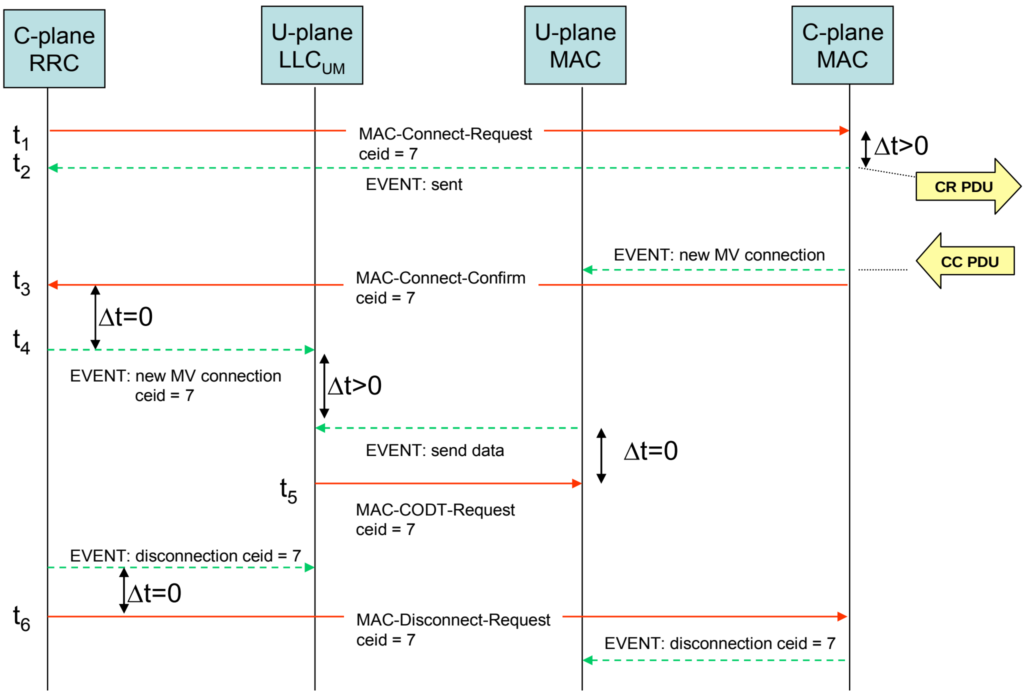

Figure 4.3 Time-sequence diagram for node i when node i completes a full communication session with node 3. The green dotted arrows indicate node internal signalling on the internal control channel.

4.2 Disconnect

Procedures for disconnection belong to signalling and are therefore done in the C-plane. The disconnection phase may be initiated by the L3a service user (NAS), by the MAC service provider or by the RRC entity itself. The sequence diagram for the disconnection phase is shown in Figure 4.4.

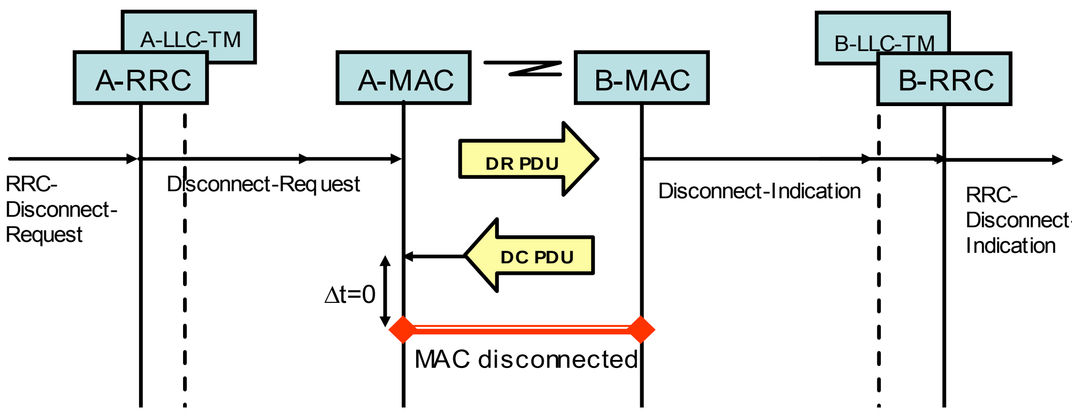

Figure 4.4 RRC initiated disconnection. The RRC-DR-PDU is embedded in the MAC-DR-PDU.

The signalling starts when the RRC-user issues an RRC-Disconnect-Request primitive and the table below presents the parameters in the RRC disconnect service primitives provided at the signalling MV-SAP.

Parameter	request	indication		
Cause	M	M(=)		
Payload	O	O(=)		
Interface Control Information				
Connection Endpoint Identifier (CEID)	M	M		

Table 4.3 RRC-Disconnect primitives and ICI. A priority parameter is not needed since disconnection shall be handled at the same priority level as the corresponding RRC-Connect.

Cause specifies the reason for the disconnection (e.g., “User initiated” - a normal case which means that the user does not need the connection anymore). The priority parameter is superfluous since a disconnection shall be handled at the same priority level as the connection it disconnects.

Note that the B-MAC entity in Figure 4.4 shall send a Disconnect Confirm (DC) PDU on the DR PDU received from A. The signalling adds insignificant overhead since node B is allowed to use the TDMA slot reserved by its peer entity (node A) to deliver DC PDUs. The error recovery procedure for handling lost DC PDUs is described in [10].

The MAC-DR-PDU and the RRC-DR-PDU have the following layouts:

```
message MAC-DR-PDU
{
fields:
  // Layer PCI
      int destAddr;                 // global destination address
      int srcAddr;                  // global source address
      int destSAP   = MAC-CCCH-SAP;
      int connectionId;
      int payload;
}

message RRC-DR-PDU
{
fields:
  // Layer PCI
      int cause;                    // the reason for sending a disconnect
      int payload;
}
```

4.3 Error Recovery

The error recovery procedures shall bring the system into a normal state after any irregular events. Error recovery involves local actions such as cleaning buffers and release of resources as well as signalling of abnormal states over the air interface.

The throughput capacity is very limited, and the protocol design must be very careful to introduce error recovery procedures based on information exchange across the air interface. The following paragraphs identify some events which must be solved before a simulator can be built, and propose a preliminary solution.

MAC-DR-PDU losses
When a MAC-DR-PDU is lost, the corresponding MAC-connection continues to exist at the remote side. The RRC takes no actions to tear down this connection, but the MAC entity shall implement an inactivity control which automatically marks reserved slots as free when no traffic has been available for a period of time.

MAC-CR-PDU losses
The consequence of MAC-CR-PDU losses depends on the number of nodes which misses the CR-PDU and where on the path between the multicast end-points the loss occurs. The effect will be observed as degraded QoS for multicast voice. A multicast voice dialog is short and must be served nearly at real-time speed which makes it questionable to apply ARQ on CR-PDU traffic. The first version of the simulator shall not implement error recovery of lost CR-PDUs and we suggest a performance study to look at the effect on the QoS for multicast voice traffic as the CR-PDU loss rate changes in different operating scenarios.

5 Multicast Voice (U-plane)

The LLC-TM entity operates in one of three different modes according to the role of the node, see Figure 5.1. Note that buffering of the voice packets is done at the LLC level. Depending on the role, the following functions shall be applied:

- Buffering of PDUs
- PDU Relaying
- Inactivity control (inform RRC about unsignalled disconnection)[21]

Voice packets have a deterministic service time in each node and they follow a fixed path across connections. Therefore the lifetime control function is not needed on the U-plane voice traffic.

The following sections consider the LLC-TM functions in detail.

5.1 Buffering and Relaying

Only the LLC-TM shall buffer outgoing voice traffic as shown in the figure below. The figure also illustrates the internal node traffic path under different operating roles: entry-node, exit-node or relay-node.

[21] Inactivity control has also been specified for the MAC layer. How inactivity control shall be distributed across the layers needs further study.

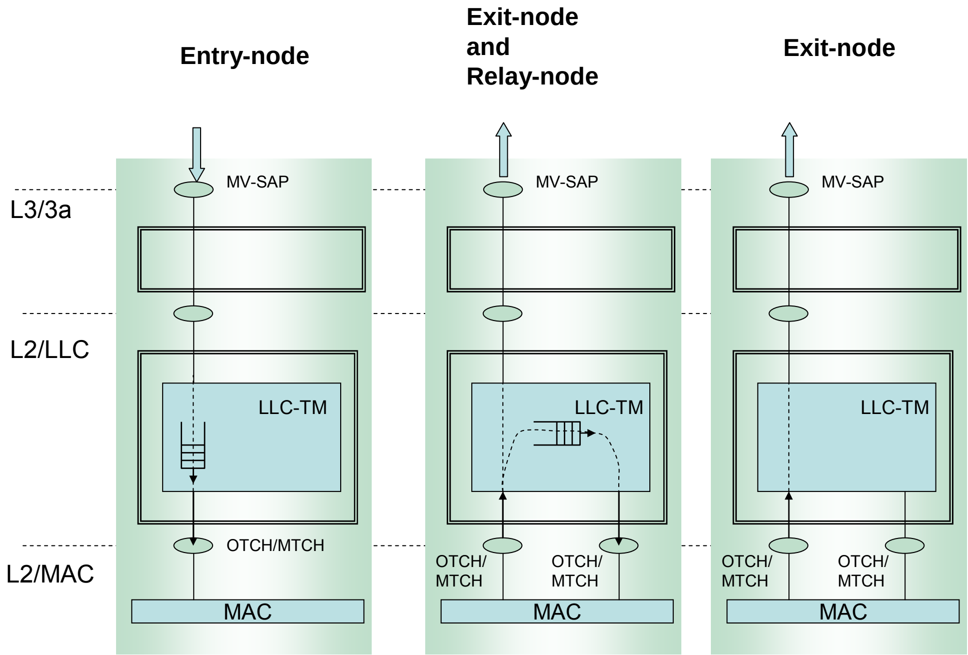

Figure 5.1   The operating modes of the U-plane LLC-TM protocol entity.

PDU relaying is done by using the MAC-CO-Data service primitives in Table 5.1.

Parameter	request	indication
Payload	M	M(=)
Interface Control Information		
Connection Endpoint Identifier	X	X

Table 5.1   MAC-CO-Data service primitives

The MAC-CO-Data primitives are carried by the following MAC-CODT-PDU (remember we have MAC-CLDT-PDU for CL-mode services):

```
message MAC-CODT-PDU
{
fields:
  // Layer PCI
      int destSAP = ...;{OTCH,MTCH}
      int srcAddr;                   // global source node address
      int connectionId;

      int payload;
}
```

Figure 5.2   The layout of the MAC-CODT-PDU. A destination address is not required since the PDU shall be sent with a broadcast address. From the {srcAddr, connectionId} pair all the receiving nodes can determine if they have a connection associated with this PDU. If not, they simply discard the PDU.

The RRC initialises the LLC-TM entity when the call setup phase ends successfully. The call setup is done on a hop-by-hop basis and therefore the incoming connection may be established before the outgoing. The RRC initialises the LLC-TM entity in two steps by sending the data structures outlined in section 4.1.1. When the incoming connection is ready, the LLC entity starts to buffer incoming data, but the entity is unable to do relaying until the outgoing connection is ready.

5.2 Inactivity Control

The MAC entity reserves slots for multicast connections and expects timely deliver of data upon request (Figure 5.3). Consider a situation where a relay node B misses a disconnect request PDU from its peer node A. Random packet loss due to bit-error(s) on the channel is a relative frequent event in a radio network, and a similar situation occurs when node mobility leads to loss of radio connectivity. The consequence is that the LLC entity in node B is unable to deliver data when its local MAC entity requests it. The inactivity control procedure shall cope with unsignalled termination of MAC connections.

The inactivity control can be implemented by means of an inactivity timer. The data rate of a voice connection is known from the connection setup phase and the timer's expiration time is set n-times larger than the voice packet inter-arrival time. The inactivity timer is restarted each time a CO-Data-Indication primitive is received. The inactivity control can either be implemented by the LLC entity or the MAC entity. Upon timeout the entity shall send a failure event to the local RRC entity on the internal control channel, and the RRC entity starts immediately to disconnect the corresponding connection. The entity sends dummy packets in the period of time the voice buffer is empty.

5.3 LLC/MAC Interface Flow Control

The delivery of data from the LLC layer to the MAC layer shall be synchronised with the TDMA frame, that is, the MAC entity requests a new MAC SDU by sending a local control signal to the local LLC entity as shown in Figure 5.3. Upon receiving the following signal, the LLC entity shall issue a MAC-Data-Request without any delay:

```
class MacLLcInterfaceControl
{
  typedef RequestType ::= {NewSDU,...};

  RequestType rType = NewSDU;
  Integer connectionIdentifier = ...;
}
```

All buffering is then above the MAC level. This principle is important when serving IP traffic since the protocol entities above the MAC layer then have the possibility to alter the PDU fields immediately before they are sent on the air.

No flow control is applied to incoming traffic over the air interface; the LLC entity is always able to handle the MAC-Data-Indication as seen from the MAC entity point of view.

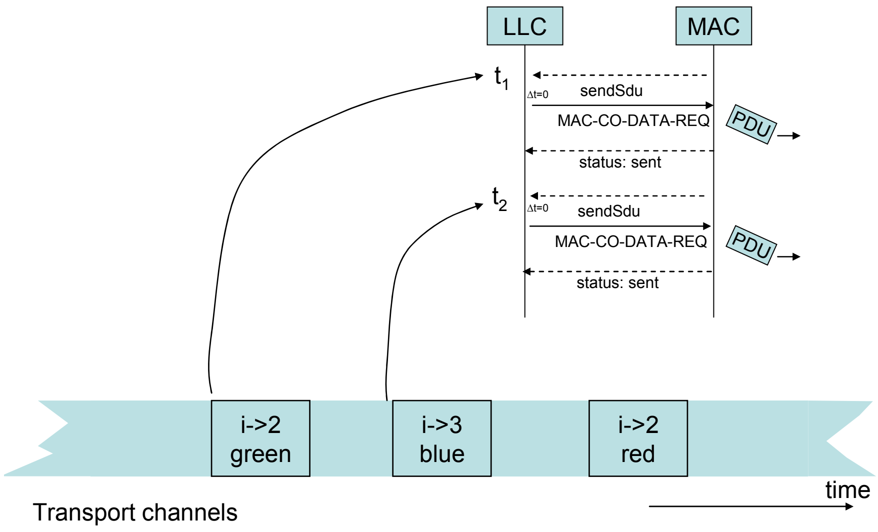

Figure 5.3 The MAC layer requests multicast voice SDUs from the LLC layer just before the corresponding TDMA slot is reached. The SDU size fits exactly in a single TDMA slot after MAC PCI is added.

6 IP Traffic (U-plane)

The first version of the simulator shall implement best-effort unicast data traffic only and therefore this chapter restricts the discussion to unicast traffic. However, a superficial discussion of multicast IP traffic may be found in section 6.3.

Unicast data traffic is handled by complex protocols both at the 3a layer and the LLC layer, see section 6.1 and section 6.2, respectively. The IP-SAP[22] exists in the U-plane only and is entirely based on CL-mode services. These protocol entities shall use the complex queue structure at layer 3a and LLC illustrated in Figure 6.1 and Figure 6.2. Queuing of relay traffic is done within the 3a layer, and this layer determines the serving policy of fresh traffic and transit traffic.

[22] The term IP-SAP refers to a TCP-SAP or a UDP-SAP.

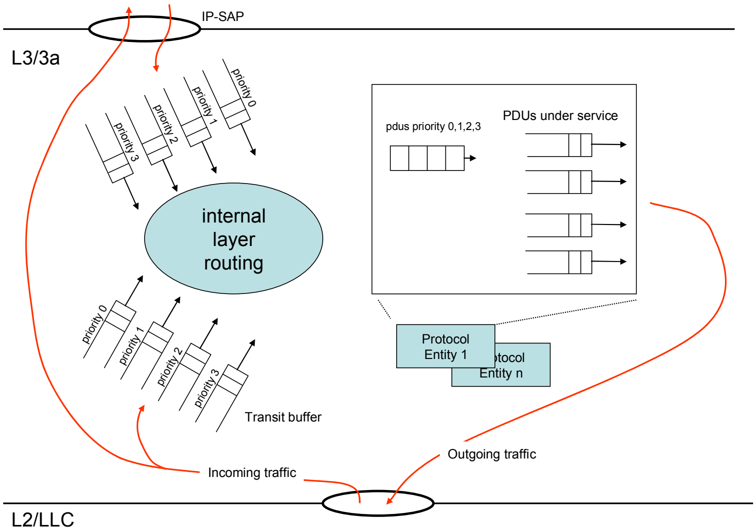

Figure 6.1 The queue structure within layer 3a for IP traffic. The 3aPDP entity operates in CL-mode and queuing is done on a per PDU basis according to the priority level.

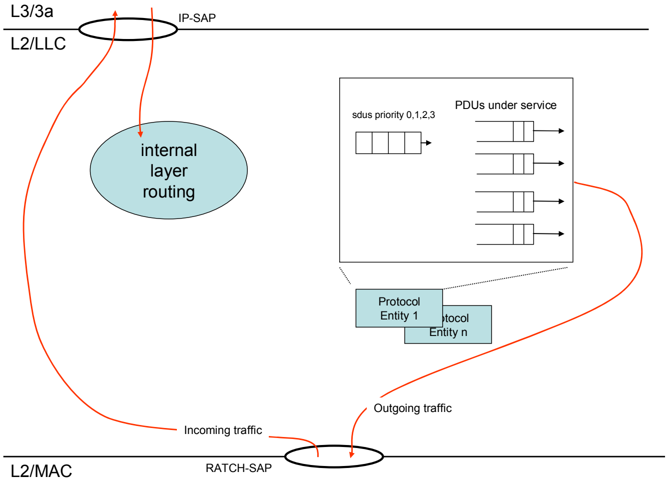

Figure 6.2 The queue structure within the LLC layer used for buffering IP traffic. The LLC entity operates in CL-mode and queuing is done on a per PDU basis according to the priority level. Each entity can store only one (1) fresh PDU per priority. PDUs under service are PDUs ready to be sent (acknowledgments or retransmissions), or PDUs awaiting acknowledgment.

Queuing of data traffic is done above the MAC layer and we introduce the following signal by which the LLC entity informs the MAC layer about pending data:

```
class MacLLcInterfaceControl
{
  class SDUinfo
  {
    Integer referenceNumber; // Unique local identifier
    Integer sduSize;
    MacAddress destination;
    Time remainingLifetime;
  };

  typedef RequestType ::= {DoScheduling,...};
  RequestType rType = DoScheduling;

  ListOf<SDUinfo> infoList = {...};
  Priority priority = ...;
}
```

The MAC entity gets sufficient information about the pending traffic, and the upper layer entities can keep their PDUs in the buffer system and may easily update the PCI fields. (For example, updating LLC piggyback acknowledgement is a problem if a PDU is physically placed in the MAC layer). Based on the *sduSize* MAC can determine the most efficient slot type to use; reserved or unreserved. The *remainingLifetime* gives the time delay until the LLC SDU shall be deleted and no scheduling is needed.

The local coordination of the MAC entity and the LLC entity is based on the two local signals named *DoScheduling* and *sendSdu* where the former carries an instance of the class *MacLlcInterfaceControl*. When the MAC entity receives a *DoScheduling*, the entity shall abort any lower priority ongoing scheduling, and immediately start a new scheduling according to the new data received. Figure 6.3 exemplifies an interface control sequence where the MAC entity requests a particular MAC SDU at time instance t_1 identified by the value *ref1*. Note that each MAC SDU contains a complete LLC PDU and its size is generally larger than the size of a TDMA slot. MAC must therefore store the MAC SDU, apply segmentation and transfer the data over a number of slots. MAC needs four buffer locations in order to store one MAC SDU for each priority level.

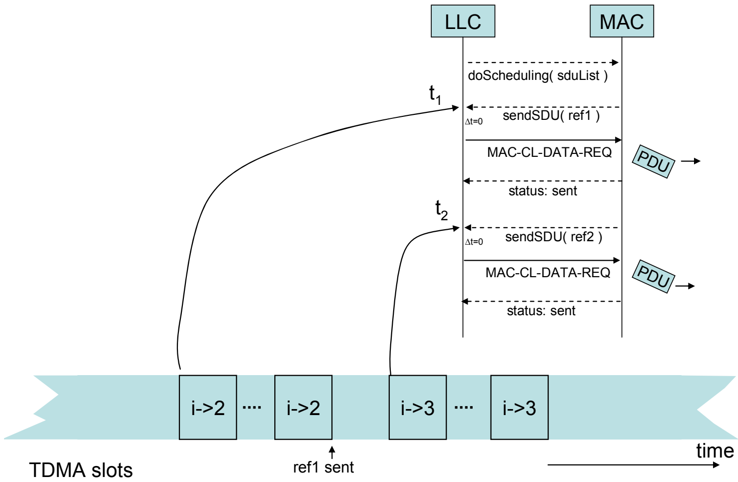

Figure 6.3 The LLC/MAC scheduling process for RATCH traffic. LLC informs MAC about the traffic conditions, MAC starts the scheduling and requests the MAC SDU when the TDMA slot starts. A MAC SDU is generally too large to be sent on a single slot.

The MAC-CLDT-PDU carries IP traffic and has the following elements:

```
message MAC-CLDT-PDU
{
fields:
  // Layer PCI
      int destAddr;                 // global destination node address
      int destSAP = MAC-RATCH-SAP;
      int srcAddr;                  // global source node address

      double remainingLifetime;
      int priority;
      int payload;
}
```

Figure 6.4 CL-mode traffic is carried by MAC-CLDT-PDUs.

6.1 The 3a Protocol

The 3a layer protocol performs store and forwarding operation in multihop networks. The following protocol functions are proposed for **best-effort unicast data** traffic:

- Data transmission using ARQ and passive acknowledgement
- Data transmission without ARQ
- Duplicate filtering
- Lifetime control (is described in chapter 7)
- Precedence and preemption (is described in chapter 7)
- Segmentation and reassembly
- Relaying
- Flow control

The paragraphs below give a short description of these protocol functions.

ARQ and Passive ACK

A “last hop” *PDU* is a PDU which has only one hop left to its end-destination while a multihop *PDU* is a PDU that has more than one hop left to its end-destination. Figure 6.5 depicts a chain of three nodes where the multihop PDU uses passive/implicit acknowledgement on the A->B link, while the “last hop” PDU uses unacknowledged transmission at the 3a level and acknowledged transmission at the LLC level. The implementation of an implicit acknowledgement scheme demands a *Global Identifier* (GId) field in the 3a PCI: a unique identifier that identifies a PDU during its lifetime in the subnetwork.

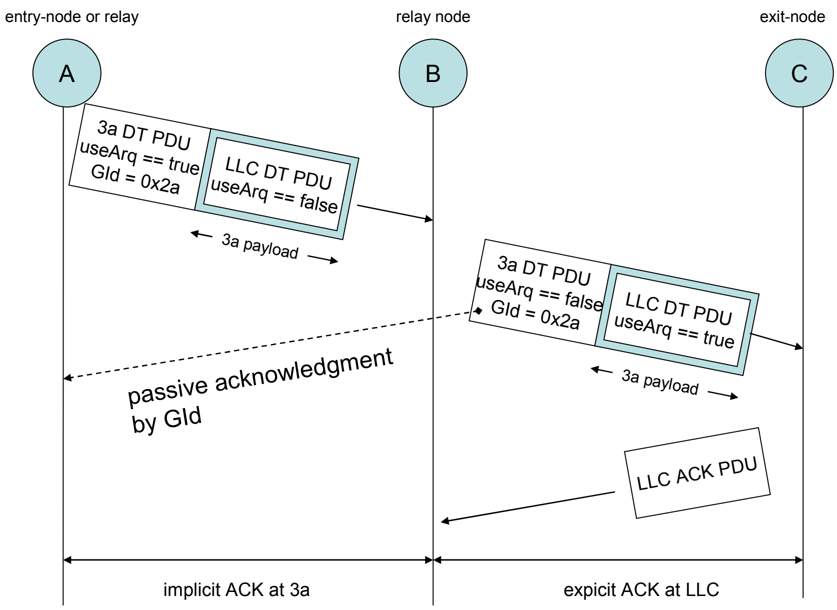

Figure 6.5 The 3a protocol utilises the semi-broadcast feature of radio communications by using passive acknowledgement on the intermediate hops and explicit acknowledgement on the last hop.

GId = <endDest><endSrc><dataUnitId> and is included in all 3a-DT-PDUs, see Figure 6.6. The GId is also utilised by other protocol functions within the 3a layer.

```
message 3a-DT-PDU
{
fields:
  // Layer PCI
      int endDest;      // terminal destination address
      int endSrc;       // terminal source address
      int dataUnitId;   // Unit identifier
      int priority;
      bool useArq;
      int noOfSegments;
      int segmentSeqNo;
}
```

Figure 6.6 Format of the 3a-DT-PDU carrying IP traffic.

```
message 3a-ACK-PDU
{
fields:
  // Layer PCI
      int endDest;      // terminal destination address
      int endSrc;       // terminal source address
      int dataUnitId;   // Unit identifier
}
```

Figure 6.7 Format of the explicit 3a-ACK-PDU (needed when the 3a source misses the passive ACK and retransmits).

Duplicate Filtering

PDUs may be duplicated within the network due to loss of 3a level acknowledgements, retransmissions and selection of alternative routes. The entry-node assigns a GId to each PDU and all relay nodes store the GId in a cash/database. If a new PDU with the same GId arrives, this PDU will be deleted since it is regarded as a duplicate.

Segmentation and Reassembly

The entry-node splits packets received from the terminal into a packet size acceptable for the LLC layer. The exit-node reassembles the packet to its original size before sending it to the terminal. All segments are relayed as independent packets.

Flow Control

The node buffer system is scaled to have a small buffer space below layer 3a and the 3a layer entity can therefore effectively choke the outgoing traffic. The flow control mechanism implemented is described in [9] and an overview is given here by means of Figure 6.8. As the first rule (single-threading), the 3a protocol does not allow more than one outstanding data packet to each of its neighbours. This is achieved by adding a forced idle period (pacing) after transmitting packet 1 in the figure. B starts to relay A's data packet at t_2 and A should obviously defer further transmissions until the passive ACK B->A is received. If A sends in the interval $\langle t_4, t_5 \rangle$ then A interferes with the ACK C->B and reduces the likelihood of successful forwarding of its own packet. Node A shall sustain from further transmissions to B until a pacing period has elapsed. The 3a protocol entity measures the forwarding delay to each of its neighbours and uses this estimate to set a pacing interval. 3a DT PDUs not requesting ARQ are not subject to this flow control mechanism.

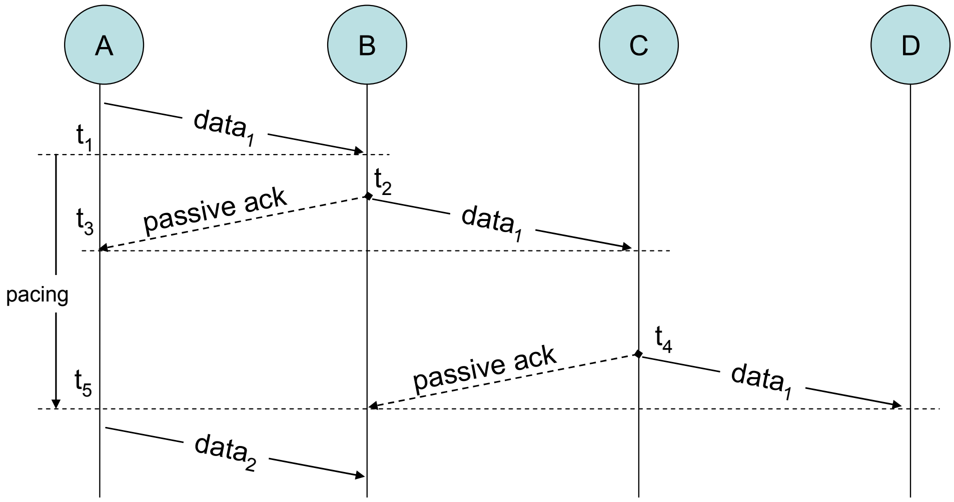

Figure 6.8 Time-sequence diagram for packet forwarding. Related data packets are tagged with the same number (GId).

6.2 The LLC Protocol

The following protocol functions are proposed for **best-effort unicast data** traffic:

- Data transmission with ARQ using a selective repeat protocol window size 2
- Data transmission without ARQ
- Lifetime control (is described in chapter 7)
- Precedence and preemption (is described in chapter 7)

The LLC protocol uses a DT PDU (Figure 6.9) to carry data traffic and the ARQ function uses the ACK PDU in Figure 6.10.

```
message LLC-CLDT-PDU      // (n)-layer peer-to-peer data
{
fields:
  // Layer PCI
      int constTypeBit   = 0;
      int nS;                 // 2 bits Send sequence number
      int nR;                 // 2 bits Receive sequence number
      bool outOfSeqIndicator;
      bool useArq;
}
```

Figure 6.9 The format of the LLC PDU which serves IP traffic.

```
message LLC-ACK-PDU        // (n)-layer peer-to-peer data
{
fields:
  // Layer PCI
      int constTypeBit = 1;
      int nR;                 // 2 bits Receive sequence number
}
```

Figure 6.10 The format of the explicit acknowledgement packet used by the LLC protocol.

6.3 Multicast Traffic

The simulator shall not implement multicast data traffic and the purpose of this section is only to explain how a multicast data protocol stack can be incorporated in our reference model. The 3aPDP we have specified cannot serve multicast traffic and we must implement a new 3a layer protocol for multicast. To separate unicast IP traffic streams from multicast IP traffic streams, we define a new data bearer (layer 3a SAP) and an accompanying logical channel (MAC SAP):

Multicast IP Data Bearer (MIP-SAP)
A CL-mode bearer for transporting multicast IP traffic from the local terminal to two or more remote terminals.

Multicast RATCH (MRATCH)
A multicast random access traffic channel used by the U-plane to send CL-mode multicast data.

The reference model is extended with a new protocol stack as shown in Figure 6.11. The 3a layer multicast data protocol (3aMDP) handles the multicast IP traffic while the LLC layer is empty (LLC-TM). The 3aPDP for unicast IP traffic splits the packets into smaller segments at the entry-node and reassembles at the exit-node. This in contrast to the 3aMDP which does not implement a segmentation function. Segmentation and reassembly is proposed done by the MAC protocol and not by the 3aMDP. The traffic flows through a dedicated logical channel (MRATCH) and the MAC entity knows that this is multicast and should select the data rate and FEC combination best suited for this type of traffic. A multicast IP packet is segmented and reassembled on a hop-by-hop basis since segmentation is done by the MAC protocol. Whether MAC uses reserved or unreserved slots for this type of traffic is invisible to the MAC service user.

The 3aMDP meets the same challenge as the multicast voice call setup protocol. The protocol architecture opens for ARQ and passive acknowledgement in the 3aMDP. The first version of the simulator shall not model multicast IP and we close further discussion in this document.

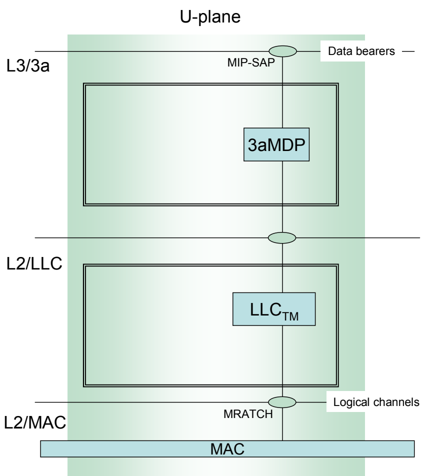

Figure 6.11 Protocol stack for multicast data traffic.

7 Crosslayer Aspects

Certain subjects are better explained if all protocol layers are considered simultaneously. This chapter discusses some subjects that come under this category.

7.1 Priority Handling

In the past, Multi-Level Precedence and Preemption (MLPP) has been a mandatory service in connection oriented military networks. With the introduction of IP terminals, we have no connections to disconnect towards the terminal equipment so preemption in this case must mean to stop a traffic stream. We assume that MLPP also in the future will be a mandatory service, and we implement priority handling in the simulator from the very beginning.

MLPP shall specify the importance of the **information content** and **not the application type** [16]. This implies that under times of resource overload, the network shall pre-empt a voice connection to release resources for serving a higher priority IP traffic stream. The MLPP level shall be set by the users[23] of the network.

In the simulator, an MLPP value is assigned per call basis, or per packet basis at OSI layer 7, and four levels are supported. The MLPP function handles the traffic strictly after rank and not by the application type. If we have a resource overload situation and the two applications multicast voice and data with **identical** priority levels, multicast voice is given precedence over data traffic.

Precedence handling on U-plane MV-traffic shall not be applied since this stack uses CO-mode services only. The serving sequence is completely determined by the reserved TDMA slot. However, precedence on C-plane MV-traffic shall be applied within all layers. Preemption of MV-connections is generally needed and is a task for the RRC entity both for incoming and outgoing calls.

Precedence is needed on data traffic within all layers. Layer 3a and LLC precedence is simply to serve the priority queues in the correct order, see Figure 6.1. The simulator has a very large buffer space and never needs delete data packets to allocate buffers for higher priority traffic. The competition between data traffic and voice traffic occurs at the MAC level since this is the only shared resource with limited capacity.

The MAC preemption process is to interrupt the ongoing service of a lower priority data packet when a higher priority data packet arrives. The cost of releasing a reserved TDMA slot and then allocate a new TDMA slot to a pre-empted data packet is high. Preemption is therefore not recommended on data traffic. The maximum MAC SDU size should be short[24] and we assume the blocking delay of high priority data becomes acceptable.

[23] A user is a person sending a message or a computer process.
[24] 500 to 1500 bytes

The MLPP process at the MAC level has the highest impact on the performance of the MLPP since it is the MAC protocol that assigns transmission capacity. Implementation of MLPP internally in a node is easy. The difficult part is to construct a mechanism that operates efficiently between network nodes because nodes have no exact and timely information about the queue status in the adjacent nodes. The signalling of the traffic load level between the network nodes need further study.

7.2 Queuing of MV Traffic

An MV-SAP exists both in the C-plane and the U-plane, and the discussion must be separated accordingly. The U-plane protocol stack is entirely based on CO-mode services and **all** buffering shall be done within the LLC layer, see section 5.1. The buffer system operates on a per connection basis.

The C-plane 3a-MV-SAP is served by the RRC entity and the RRC/LLC entity uses the CCCH which offers a CL-mode service only. We have a similar situation as described for the U-plane data traffic in Figure 6.3 and select the same solution for the C-plane signalling traffic. Then the U-plane LLC and the C-plane LLC can be implemented by the same software package.

7.3 Connection Endpoint Identifiers

The purpose of connection endpoint identifiers (CEIDs) is to identify a connection within a service access point (SAP). For the reference model, we need CEIDs in the MAC-OTCH-SAP, MAC-MTCH-SAP, LLC-MV-SAP(U-plane) and L3a-MV-SAP(U-plane). The rest of the SAPs supports only CL-mode services and needs no CEIDs. CEIDs must be carried by PCI fields and therefore the number range is an issue since CEIDs cannot be reused during the maximum SDU lifetime in the underlying layer. Therefore a real implementation must consider these aspects carefully. However, a simulator may take some shortcuts.

We decide to allocate 32-bits for CEIDs and let the entry-node (L3-MV-SAP user) select the identifier. The same number shall be reused by the lower layer protocols in the entry node. The MAC entity assigns CEIDs to connection setups received over the air interface. The same number shall be reused by the upper layer protocols in the exit/relay node.

7.4 Lifetime Control

One purpose of packet lifetime control is to stop serving data which have expired, that is, the data is not useful for the recipient(s). Another purpose is to end looping of packets if the routing protocol fails. We also need to have a maximum packet lifetime in the network to be able to reuse unique identifiers. For example, connection endpoint identifiers must be frozen until all packets belonging to the connection have been removed in the subnetwork. If not then the system may deliver corrupted data. Note the relationship between the range of identifiers and the maximum lifetime. Lifetime is not intended to be used as a service to take delay measurements!

The principle of the lifetime control function is that the upper layer in entry-node (RRC-Connect-Req/CL-Data-Req/CO-Data-Req) sets a maximum lifetime value. The 3a and LLC layer entities in the entry-node measure their internal queuing delays. When the SDU is sent down to the MAC layer, the lifetime and the internal node delay are also sent over. Based on these two values the MAC entity calculates the remaining lifetime and this value is carried by the outgoing MAC PDU. The receiving nodes execute the lifetime control function according to the same procedure. When the remaining lifetime reaches a predefined threshold, a specific threshold for each layer, the corresponding packet shall be deleted without further actions. The exception is if a receiving node is an exit-node. In this case the packet shall always be sent to the local terminal.

Lifetime can be given as the expiry time referred to the global time, or as a relative time as outlined above. The simulator uses relative time with perfect precision (a variable of type double in each layer makes delay measurements and the MAC PCI lifetime field is also a double).

The MAC-User decides if lifetime control and signalling should be applied on a per SDU basis since the lifetime field of the MAC PCI is redundant for certain types of control traffic. For example, lifetime control of an LLC acknowledgement packet is not needed.

8 The MAC/LLC Interface

The RRC entity operates at layer 3 and has a better view of the resource situation than any lower layer entities. As distinct from the MAC entity, the RRC entity has knowledge of the network topology and can determine the type of MV-connection needed - one-to-one or one-to-multipoint. The MAC entity's main responsibility is to manage the TDMA frame structure both with regard to local housekeeping and synchronisation with remote nodes. A MAC protocol for NBWF is proposed in [10] and below we give a summary of the basic MAC protocol characteristics assumed in this document.

Common functions for all logical channels:

- MAC does not implement an ARQ protocol
- MAC executes a lifetime control function as the other protocol entities
- MAC does only buffer MAC SDUs which are under service.

Traffic on OTCH and MTCH

- MAC **does not** implement a segmentation and reassembly protocol
- MAC **does not** implement precedence and preemption function; which stream to serve is determined by the TDMA slot. (Preemption is handled by RRC)
- Support for one-to-multipoint connections on the MTCH, see Figure 8.1.

Traffic on RATCH and CCCH:

- MAC implements a segmentation and reassembly protocol
- MAC accepts SDUs up to 1500 bytes

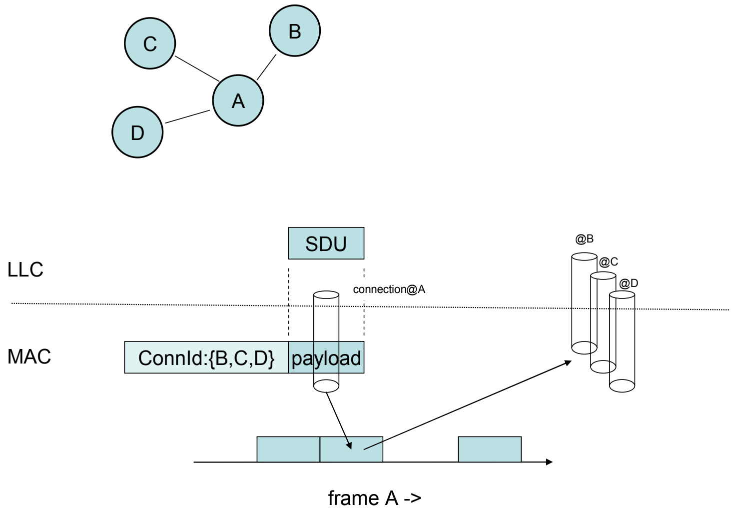

Figure 8.1 *A one-to-multipoint connection facilitates delivery of one MV voice packet to many destinations in a single transmission.*

9 Network Management

Earlier chapters have outlined how the terminal traffic shall be served by the user plane and how the signalling traffic is processed in the control pane. A real network needs many management functions to be able to provide the basic communications services. For example, a function for probing terminals is needed - “where is terminal x? I want to talk”. Emission of beacon/Hello packets are needed under low traffic conditions to maintain the network topology information. A routing protocol must supply the RRC entity and the 3aPDP entity with routing information when requested. The first version of the simulator shall not support node mobility and we can postpone management to a later time. However, management is a very important issue and we want to indicate how management protocols can be incorporated into the reference model.

One solution is to introduce a management plane as shown in Figure 9.1, and an accompanying logical channel named Management RATCH (MRATCH). Note the similarity with the logical channel RATCH which serves IP traffic. Management traffic has no real-time requirement and can be processed by the same protocol stack as the IP traffic. The overhead introduced by the M-plane is one extra bit in the MAC PCI. The management traffic shall be buffered in the same queuing systems as ordinary traffic. If the MAC layer does not need an explicit management protocol, the M-plane traffic could use the RATCH while the LLC layer performs the one-to-many SAP splitting.

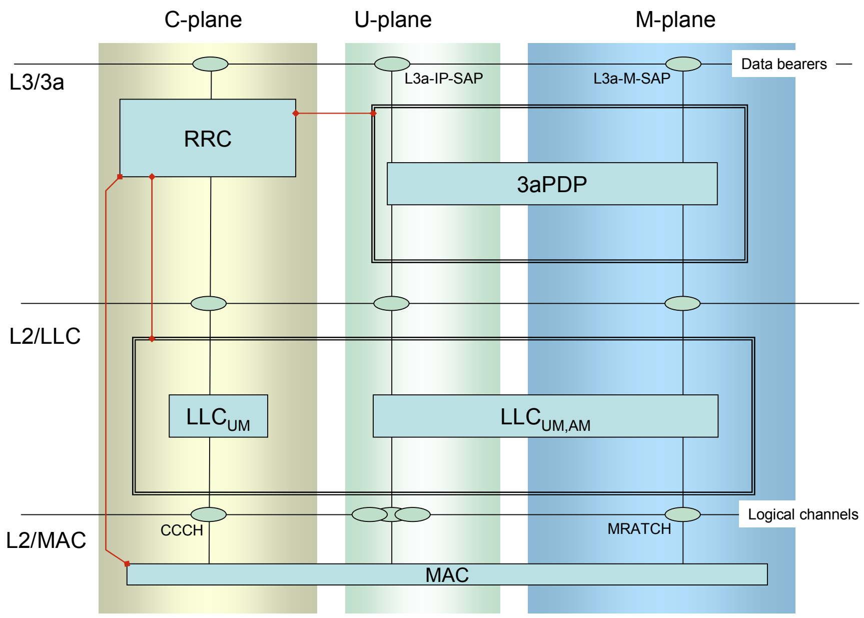

Figure 9.1   The reference model extended with a management plane.

10 Modelling a Network of Radios

Modelling and Simulation (M&S) are two distinct, yet complementary activities. *Modelling* is the process of creating a model, while a model is anything to which experiments can be applied in order to answer questions about the system modelled. A *simulator* is any object that implements the model and *simulation* is the process of running the simulator. The objective of this chapter is to **model** a network of radios that can predict throughput-delay performance and multicast voice quality under different conditions.

A model needs to support different operating environments with respect to the usage of the communication services and radio coverage areas (network topology). Thus the model must facilitate deployment of radio nodes within a deployment area, referred to as the playground, and provide functions to set up different traffic generators. An example scene is illustrated by Figure 10.1. A number of radio nodes deployed on the playground serves terminal traffic on a shared radio channel.

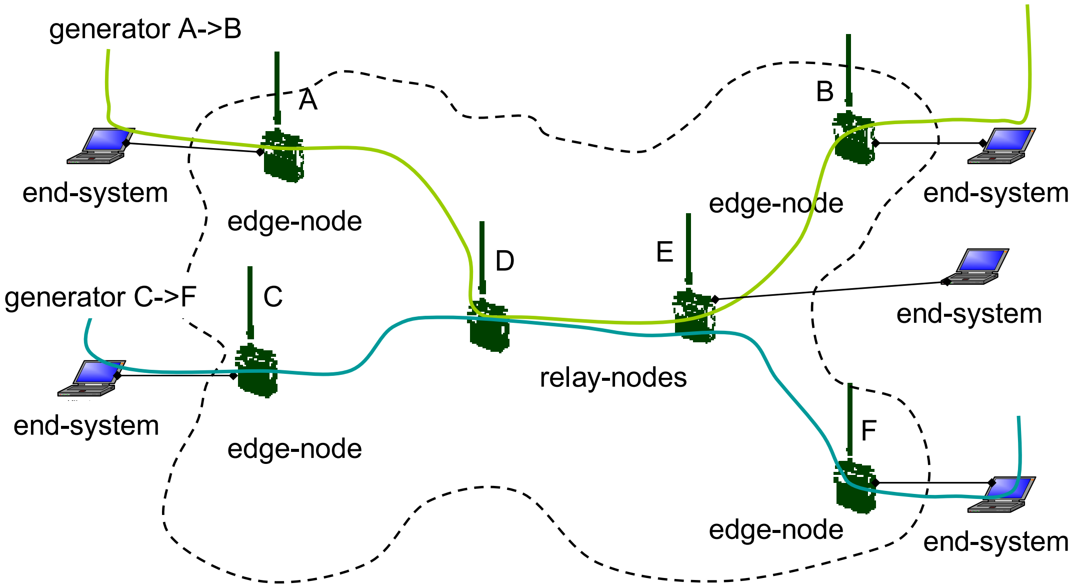

Figure 10.1 An example MANET scene with two traffic generators representing the terminal traffic.

The top level components introduced to model the network scene are shown in Figure 10.2 below. The user environment, that is, the usage of the communication services, is modelled by the *User Environment* (UE) box. The arrows signify message passing and the figure illustrates how traffic generators send messages to layer 7 within the hosts. A *Host* is an abstraction of a radio (OSI layer 1 to 3) with an attached user terminal (OSI layer 4 to 7). No messages flow back to the UE and the traffic streams terminate at layer 7 within the exit-nodes. Notice that no direct connections exist between the hosts. They communicate through a physical transmission medium, which models the RF conditions on the radio channel.

The model models a set of radio nodes operating on a single shared channel. The model does not handle a scenario where two separate radio networks are interconnected by a gateway.

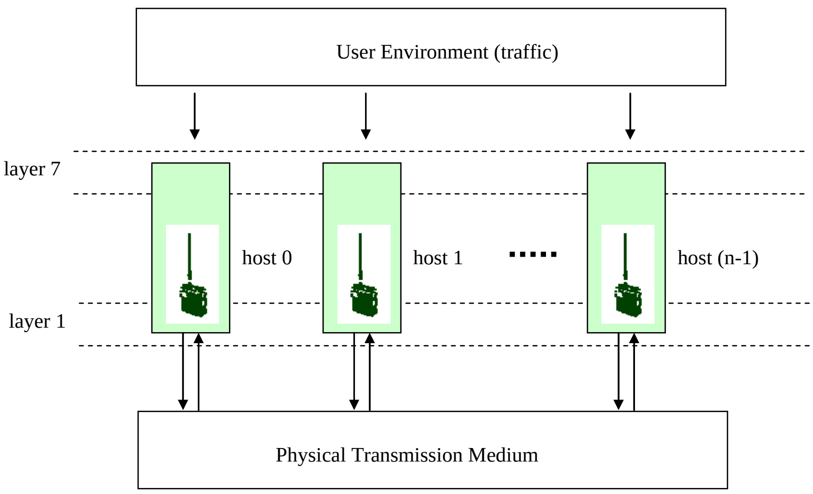

Figure 10.2 The basic components of the model that shall model the system exemplified by Figure 10.1.

The remaining part of this chapter breaks down this basic structure in a hierarchical manner to end up with a set of sub models of reduced complexity suitable for implementation by a computer program.

The model of a network of radios is a **stochastic discrete event continuous time** model, and has a hierarchical architecture composed of two elementary model types: *atomic* models and *coupled* models[25]. Atomic models are, as the name says, the smallest element of building blocks that are coded in a programming language. Coupled models are either composed of other coupled models and/or atomic models.

The top level is a coupled model, identified by the dotted box in Figure 10.3, named *Sim*. *Sim* contains four atomic models and one coupled model, and encapsulates all the other models. *Sim* makes a complete system with any number of hosts.

The coupled model *Host* models a single network node with one radio and one user terminal, that is, a *Host* encompasses layer 1 to 7 of the OSI Reference Model. The model must handle any number of *Host* instances where each host is assigned a unique address (number range is zero to network size minus one).

The atomic model *ChannelControl* models the radio frequency (RF) environment of the real world, the physical transmission medium in our basic structure[26]. The functions of the *ChannelControl* is to determine the RF pathloss according to the pathloss model in use, set the receiving power at the destination end, and copy the RF signals from the *coaxOut* port of a transmitting host to the *coaxIn* port on all the hosts within the radio coverage area of the transmitting host. The *Sim* shall have one instance only of the *ChannelControl.*

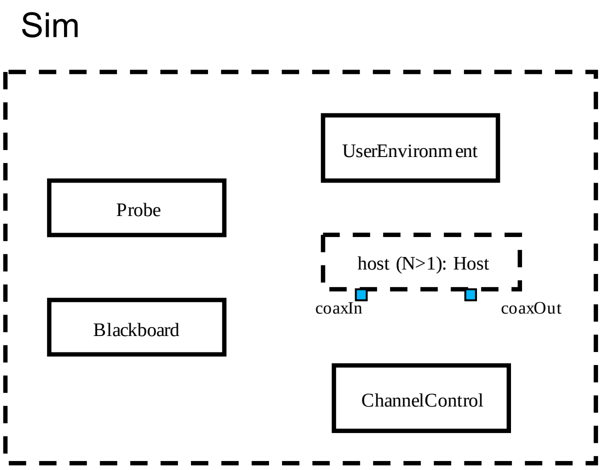

Figure 10.3 The top level of the model is a coupled model containing both atomic and coupled models.

---

[25] The term ”atomic model” is in this document implemented by a ”OMNeT++ simple module”. Cf OMNeT++ user manual [7].
[26] The OMNeT++ class ChannelControl is taken as the basis for implementing the ChannelControl module.

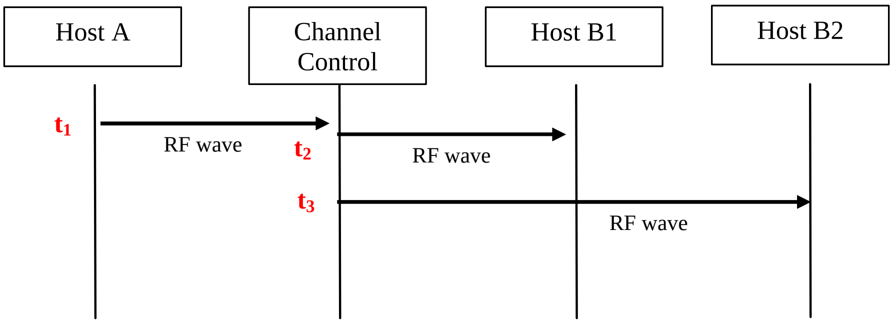

Figure 10.4 Sequence diagram for radio wave propagation when host A transmits. The ChannelControl distributes the wave to all hosts within the radio coverage area of host A.

```
message RFwave
{
fields:
  double txPowerDbm;       // Radiated power in dBm with which this packet is transmitted.
  double rxPowerDbm;       // Power in dBm at the receiver antenna input
  double rxPowerW;         // rxPowerDbm converted to W
  double duration;         // Time it takes to transmit the packet, in seconds!
  double startedAt;        // The time instance the wave reached the receiver antenna.
  int srcNode;             // The host identifier (>= 0) that sends this packet.
};
```

Figure 10.5 The attributes of the RF wave.

The atomic model *Blackboard* does not exist in the real world, but is included in the model for publishing of global network information. Information published on the *Blackboard* does not traverse the radio channel, does not affect the network performance and *Sim* shall have one instance only of the *Blackboard.*

One usage of the *Blackboard* is in conjunction with routing. The *ChannelControl* publishes the link cost matrix on the blackboard at time instance zero (the model does not support mobility yet). By reading this set, the network routing algorithm gets information about radio link connectivity, which is needed to route traffic over multiple radio hops. Another usage of the *Blackboard* may be to implement an address map between internal addresses, range is 0...(n-1), to external addresses (e.g. IP addresses, map[host0].ipAddress gives 127.0.0.8).

The atomic model *Probe* collects samples from network distributions (packet loss, queuing delays etc.) and performs data analysis in run-time. Such functionality would also be needed if network statistics should be measured in a real system. *Sim* shall have one instance only of the *Probe* which is specified in a separate document [7, 12].

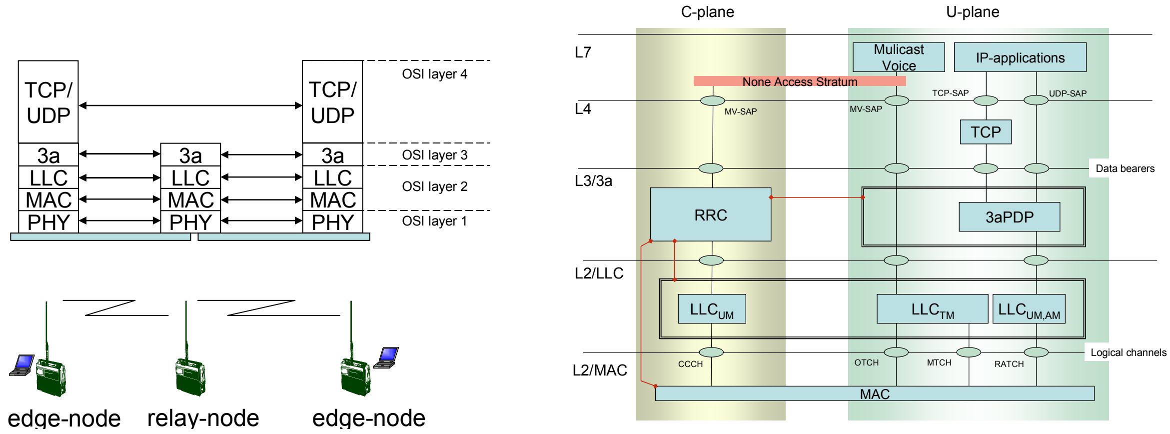

Figure 10.6 The upper layer protocol architecture for a host object.

An objected oriented approach is also followed in the design phase of the simulator - there is a one-to-one mapping between the real-life components and the components in the model. The simulator must create one host object for each radio node specified in a network scene. Figure 10.6 presents the upper layer protocol stack for the host. Layer 7 contains a voice application based on MELPe [11], and an IP application which demands TCP and UDP. Layer 4 entities are able to determine the traffic type since the traffic is served on different SAPs. Only one protocol (TCP) is needed because voice and UDP traffic can pass transparently through layer 4. The 3a packet data protocol (PDP) fulfils all UDP traffic requirements.

No control plane functionality is defined above layer 3, and the host must have a function for routing the layer 7 signalling traffic (connect/disconnect requests) to the control plane. Splitting of signalling and data traffic is done within the *None Access Stratum* (NAS). The RRC is, of course, not involved in TCP call establishments. Layer 3 does not differentiate between TCP control traffic, TCP data traffic and UDP data traffic[27] and serves the traffic over the LLC-IP-SAP.

Most readers are familiar with the term *Network Interface Card* (NIC). When you plug in a WLAN (IEEE 802.11) NIC, you "implement" layer 1 and layer 2/MAC functionality in your PC. Layer 2/LLC usually runs in the OS kernel space, not on the external card, and you "implement" LLC when you install the driver. We extend the NIC functionality to also include the LLC sublayer and define the term *RadioNic* as the notation of an entity which implements the physical radio hardware, the MAC and the LLC protocols. The simulator shall handle any number of hosts, but **a host can contain one NIC only**. Motivated by these observations, we introduce a set of atomic and coupled models as shown in Figure 10.7.

[27] However, users may use QoS parameters to signal special needs

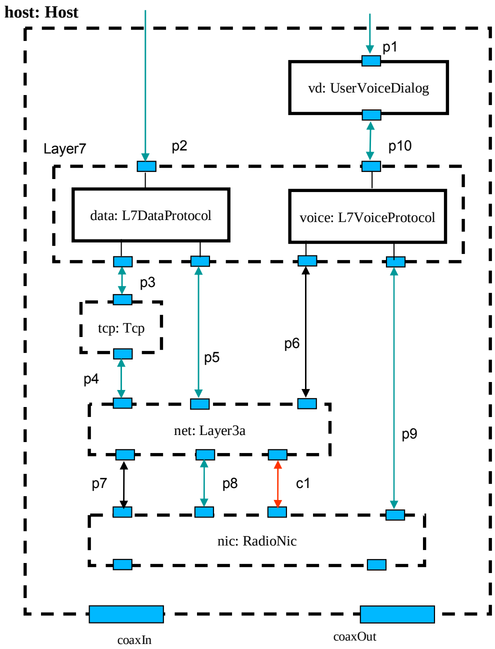

Figure 10.7 The interior of the coupled model Host. Coupled models are marked as dotted rectangles.

The two atomic models *UserVoiceDialog* and *L7VoiceProtocol* implement user behaviour during a multicast voice dialog and a MELPe protocol, respectively. These models are described in [11]. The atomic model *L7DataProtocol* is a simple application layer protocol which handles TCP and UDP traffic. TCP is implemented in the atomic model named *TCP* and is specified in [8].

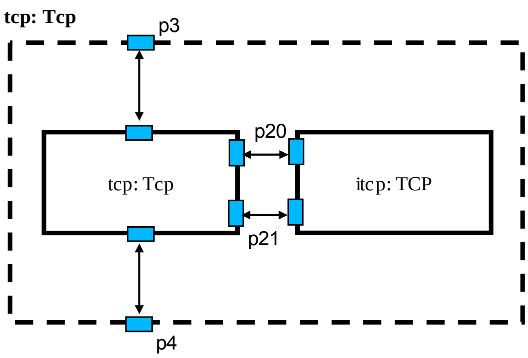

Figure 10.8 The coupled model Tcp contains the atomic model TCP which is implemented by the INET-project [7]. The atomic model Tcp is designed to be a wrapper between the INET software and our software.

The blue rectangles in the figure are gates which facilitate message exchange between models. A gate is not directly linked to the term SAP as specified by the reference model. Remember that SAPs, as defined by the OSI Reference Model, is a utility to address protocol entities. Also note the control plane specified in our reference model is not visible in the model of a *Host*.

The coupled model *Layer3a* models the 3a layer and contains the three atomic models shown in Figure 10.9. The *RRC* and *L3aPDP* model the RRC entity and the 3aPDP entity defined by the reference model. The atomic model *Routing* provides the routing functions needed by RRC and L3aPDP.

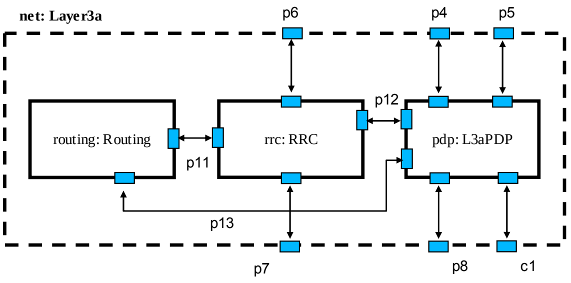

Figure 10.9 The coupled model Layer3a.

All gates are assigned unique identifiers and the following paragraphs give a summary of the most important message types passing these ports.

Gate p1 and p2
Messages from the user environment model enter a host through ports 1 and 2. These messages represent terminal generated traffic. A message on *p2* activates either the TCP entity or the UDP entity. A message on *p1* initiates a multicast voice call setup.

Gate p4 and p5
This is data traffic in the U-plane severed by two layer 3a data bearers (L3a-TCP/UDP-SAP) in our reference model.

Gate p6 and p9
Multicast voice signalling traffic (C-plane) goes via gate 6 while the multicast voice data traffic (U-plane) passes through gate 9. A message directed downwards on *p9* is sent directly to the *RadioNic*, and the *RadioNic* model must internally multiplex the traffic on the OTCH or the MTCH.

Gate p7
Control plane CCCH traffic is served on gate 7.

Gate p8
User plane traffic passing L3a-IP data bearers (gate 4 and 5) is sent or received via port 8 at the L3a/LLC interface.

Gate c1
Internal node control channel used for flow control between layer 3a and LLC.

Above we introduced the coupled model *RadioNic,* and Figure 10.10 below expands the model to three new sub models.

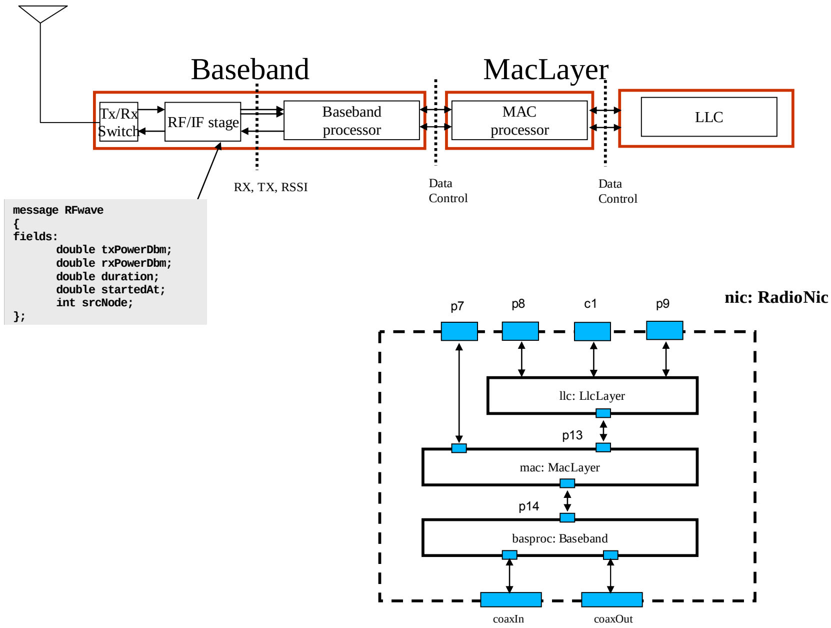

Figure 10.10 The topmost figure shows the internal structure of a radio card. Functional split of the RadioNic into sub models is shown at the bottom. An incoming RF wave (class RFwave) arrives at the coaxIn port and is processed by the base band processor. A radio transmission leads to an outgoing RFwave on the coaxOut port.

The atomic model[28] *Baseband* models the radio while the coupled model *MacLayer* implements the MAC protocol. The *Baseband* model is based on the radio proposal in [Phil] and how the radio shall be modelled in the simulator is described by [14]. The *LlcLayer* implements the LLC protocol. Of course, the *Baseband* and the *MacLayer* must also communicate - both data and local control signalling are needed. This is done over a number of gates as shown in the figure.

The modelling hierarchy is now completed since all coupled models are described by atomic models. The network model is a model of models organised in a hierarchical manner as illustrated by the tree structure in Figure 10.11. The leaf nodes are the atomic models. The top node is the *Sim*, which is a coupled model, and the model may have any number of the nodes named *Host*.

[28] The term ”atomic model” is in this document implemented by a ”OMNeT++ simple module”. Cfr OMNeT++ user manual.

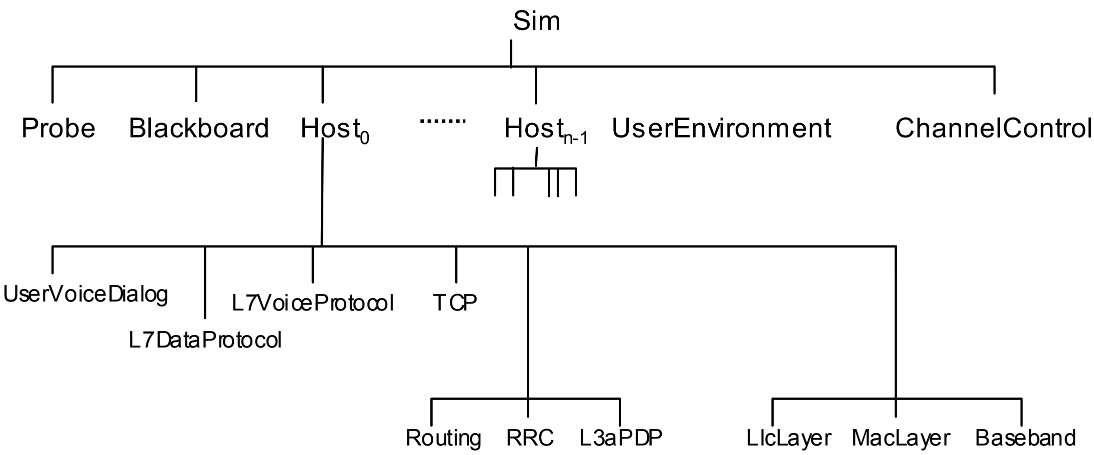

Figure 10.11 The network model is a model of models organised in a hierarchical manner.

Before leaving this chapter, we will give a more detailed description of *ChannelControl* by means of Figure 10.12. When a transmitting radio sends, *Baseband* creates an *RFwave* and fills in the parameters: *srcNode*, *txPower* [dBm], *duration* [sec], *startedAt* [sec]. The time values are referred to the simulator's time axis. The *RFwave* message is sent to *ChannelControl*, which calculates the signal level for each possible destination, makes an explicit copy to each destination of the *RFwave*, fills in the *rxPower* and sends the messages to all the receivers. Further processing of the *RFwave* is done by the *Baseband* processor in the receiving hosts.

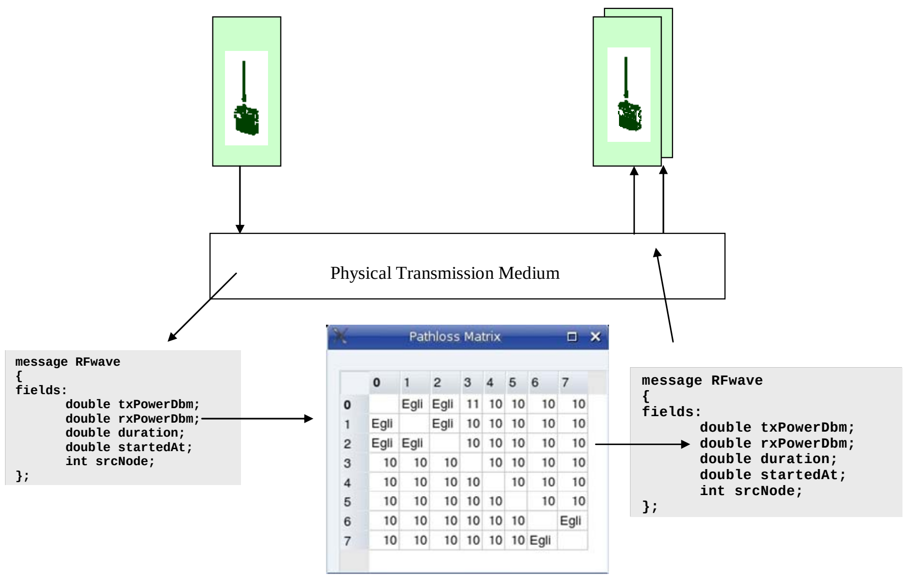

Figure 10.12 The ChannelControl model models the physical transmission medium.

10.1 Global Objects

The first version of the simulator assumes perfect knowledge of the TDMA slot states - busy or idle. This means that all the nodes in the network have identical and correct information about the slot states when the slot reservation phase starts. A global object in the simulator named "TDMA monitor" maintains this information. The hosts access the TDMA monitor over an idealised "channel"[29] (i.e., the radio channel is not used).

The TDMA metronome emits an event to each node when a new TDMA slot starts, including the slot number. All hosts have exact timing information without using the radio channel.

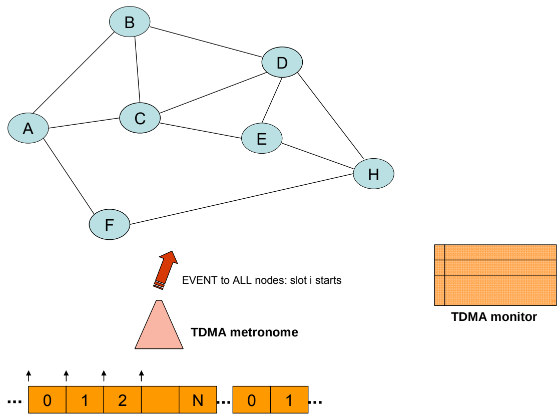

Figure 10.13 The simulator uses two global objects, TDMA metronome and TDMA monitor, by which every radio node gets exact and correct information about the TDMA slot states and timing regardless of the network topology.

11 Conclusions and Remarks

The aim of this document is to make an initial design of a simulator that models a simplified NBWF network. A protocol stack for the NBWF network has not yet been specified and we had to specify a reference model for a radio node and an initial protocol stack that is able to serve TCP traffic, UDP traffic and multicast voice traffic.

The reference model presented in chapter 3 is a result of a number of iterations between different models and protocol designs as well as studying OSI reference models specified by other communications systems. A radio node with the complexity studied must be structured into functional units in one way or another. The reference model proposed appeared to be very useful both during the protocol specification and the design of a radio node.

The first phase of any performance study should focus on protocol behaviour where the dynamics are caused by the user traffic only. Introduction of node mobility complicates traffic analysis and should be deferred until we get a better understanding of the basic protocol characteristics. Therefore, routing is not a big issue in this document since we can use fixed routing tables. Moreover, our main focus is protocol specification for building a simulator and we have the obligation to introduce some simplifications regarding TDMA slot state information (reserved/unreserved) and slot synchronisation. Both must be considered in a real system and will

[29] Simply a procedure call in the simulator.

add signalling traffic, and thus lead to performance degradation upon failure. The first version of the simulator models perfect conditions.

The layer 2 and layer 3 protocols serving data traffic are not optimised for a TDMA based access protocol. These protocols are used since we have the software package that implements these protocols. We emphasise that they shall be used in stage 1 only and the protocol stack for data traffic shall be redesigned in stage 2.

The object oriented design technique used and the detailed modelling of the NBWF node result in a considerable number of code lines. The benefits are less modelling error; the node structure implemented is close to a real node, and the quality assurance of the protocol specification we get by actually implementing it.

The simulator designed in chapter 10 is under implementation and Table 11.1 presents the current status of this work, as of November 2008.

<table>
<tr><th></th><th>UDP traffic</th><th>TCP traffic</th><th>Voice traffic</th></tr>
<tr><td>Traffic generators and L7 protocols</td><td>ready</td><td>ready</td><td>ready</td></tr>
<tr><td>L4: TCP</td><td></td><td>Implementation started in September</td><td></td></tr>
<tr><td>L3a: RRC</td><td colspan="2"></td><td>ToDo</td></tr>
<tr><td>L3a: 3aPDP</td><td colspan="2">ready</td><td></td></tr>
<tr><td>L2b: LLC</td><td colspan="2">ready</td><td>ToDo</td></tr>
<tr><td>L2a: MAC</td><td colspan="3">Design starts in December</td></tr>
<tr><td>L1: Radio</td><td colspan="3">Design ready and implementation starts in December</td></tr>
<tr><td>Propagation</td><td colspan="3">ready</td></tr>
<tr><td>Statistics and measurements</td><td colspan="3">ready</td></tr>
</table>

Table 11.1   Status of the simulator stage 1 development November 2008.

References

[1] "Requirements for a Narrowband Waveform", AC/322(SC/6-AHWG/2)M(2008)0003, August 2008.

[2] Open system interconnection - Basic reference model, ISO/IEC 7498.

[3] Conventions for the definition of OSI Services, ITU-T Recommendation X.210.

[4] Radio Link Control(RLC) protocol specification, 3GPP TS 25.322 v4.12.0, June 2004.

[5] Radio interface protocol architecture, 3GPP TS 25.301 v4.3.0, June 2002.

[6] TETRA part 1: Overview, technical description and radio aspects, ETR 300-1, May 1997.

[7] OMNeT++, www.omnetpp.org

[8] "A TCP module for the NBWF simulator", in preparation, Bjørnar Libæk.

[9] John Jubin and Janet D. Tornow, "The DARPA Packet Radio Network Protocols", Proceedings of the IEEE, January 1987.

[10] Svein Haavik, "Initial MAC Protocol Design for NBWF", FFI-rapport 2008/02172.

[11] Jan Erik Voldhaug, " Modelling and performance estimators for multicast voice services", FFI-notat 2008/02168.

[12] Tore J Berg, "oProbe - an OMNeT++ Extension Module", http:://sourceforge/projects/oprobe.

[13] Lars Bråten et.al., "Medium access for narrowband wireless ad-hoc networks; requirements and initial approaches", FFI-rapport 2008/01313, FFI June 2008.

[14] Vivianne Jodalen, "Modelling the NBWF radio", TIPPER/FFI project document, FFI June 2008.

[15] www.ericsson.com/hr/enterprise/special/mrr.pdf.

[16] R. G. Cole and B. S. Farroha, "Implications of Precedence and Preemption Requirements on Packet Based Transport Architectures", MILCOM 2007.

[17] B. S. Farroha et.al, "Requirements and Architectural Analysis for Precedence Capabilities in the Global Information Grid", MILCOM 2006.

Abbreviations

ARQ	Automatic Repeat Request
CC	Connect Confirm
CCCH	Common Control Channel
CC-PDU	Connect Confirm PDU
CEID	Connection Endpoint Identifier
CL	ConnectionLess
CNR	Combat Net Radio
CO	Connection Oriented
CR-PDU	Connect Request PDU
DOM	Document Object Model
DR-PDU	Disconnect Request PDU
DSSS	Direct Sequence Spread Spectrum
DT-PDU	Data PDU
GiD	Global identifier
GUI	Graphical User Interface
GUIA	GUI Automatic
ICI	Interface Control Information
IP	Internet Protocol
IP-SAP	Internet Protocol SAP
LLC	Logical Link Control
LLC-AM	LLC Acknowledged Mode
LLCE	LLC Entity
LLCP	LLC Protocol
LLC-TM	LLC Transparent Mode
LLC-UM	LLC Unacknowledged Mode
MAC	Medium Access Control
MAC-E	MAC Entity
MAC-SP	MAC Service Provider
MANET	Mobile Ad-hoc NETwork
MIP-SAP	Multicast IP SAP
MRATCH	Multicast Random Access CHannel
MTCH	one-to-Multipoint Traffic CHannel
MV	Multicast Voice
MV-SAP	Multicast Voice SAP
NBWF	Narrow Band Wave Form
NIC	Network Interface Card
NM-SAP	Network Management SAP
OS	Operating System
OSI	Open System Interconnection

OTCH	one-to-One Traffic CHannel
PCI	Protocol Control Information
PDP	Packet Data Protocol
PDU	Protocol Data Unit
PHY	Physical
PTT	Push To Talk
RATCH	Random Access Traffic CHannel
RF	Radio Frequency
RLC	Radio Link Control
RRC	Radio Resource Control
SAP	Service Access Point
SDU	Service Data Unit
SQL	Structured Query Language
TDMA	Time Division Multiple Access
UE	User Environment or User Equipment
UI	User Interface
UTL	Utility
UV	Unicast Voice
UV-SAP	Unicast Voice SAP
XML	Extensible Mark-up Language
xxx-E	xxx Entity (e.g., LLC-E)
xxx-SAP	xxx Service Access Point (e.g., LLC-SAP)
xxx-SP	xxx Service Provider (e.g. MAC-SP)