QoS and Flow Management for Future Multi-Hop Mobile Radio Networks

Rainer Schoenen[1], Arif Otyakmaz
Communication Networks (ComNets), Faculty 6, RWTH Aachen University, Germany
{rs|aoz}@comnets.rwth-aachen.de

***Abstract*— Mobile radio networks of the IMT-Advanced systems family promise ubiquitous broadband access and high area coverage, with rates of several 100 MBit/s. They claim to guarantee QoS support in terms of low delay and guaranteed throughput. However, with the availability of flat rate plans and bandwidth-hungry applications of future mobile devices the systems are facing a hard challenge in satisfying all demands at the same time in a traffic load situation which can best be characterized as total overload or full buffer. In this situation the basic voice service must still be operational to full Erlang capacity, no matter what load the data traffic offers. In this paper the hierarchical static priority scheduling scheme is used to accomplish the required separation. Candidate technologies like LTE-Advanced, WiMAX are based on OFDMA access which allows flexible radio resource allocation, but has an inherent near-far heterogeneity which leads to unfairness if it is optimized for spectral efficiency. Multi-hop relaying is one way to diminish the near-far problem, which is why it is an important part of the system concepts. The scheduling scheme is designed with this multihop capability in mind and results show that in heterogeneous scenarios a proportional fair substrategy achieves the desired fairness. To achieve QoS distinction a flow management must be provided in OSI layer two, so that flows of different classes can be treated separately. This paper treats the flow management concept for multi-hop mobile radio systems, a key enabling technology for QoS aware resource scheduling. It features the cross-layer signaling of QoS requirements by the Application Layer to the Data Link Layer.**

***Index Terms*— Flows, Relaying, Multi-hop, QoS, Scheduling**

I. Introduction

THE data rates of future cellular systems will increase up to aggregated $1Gbit/s$ in microcells and $100MBit/s$ in rural areas. However, these numbers cannot be increased arbitrarily due to physical and complexity limitations. On the other side the applications which are used wirelessly have just begun to keep up in demand with the offer. One killer application is a laptop running huge operating system and application updates in an unstoppable way while the user has subscribed to a mobile flatrate plan. Wasteful offers like traffic flatrates subserve the tendency of exponential traffic increase. This dubious demand takes more and more financial and technical effort to support. On the income side the market is saturated and more cash flow cannot be expected. The economic result is that the telecommunication market becomes less and less profitable. The technical result is that competing companies spend an enormous effort to come up with new algorithms to better utilize the wireless channel.

[1]Dr. Schoenen is currently at http://www.sce.carleton.ca/

We must realize that the overprovisioning of capacity is an illusion in the long term. Transport capacity growth will come to an end or slow down significantly. In the short term traffic is always bursty in nature and therefore produces short term overload situations in orders of $10ms$ to several seconds. The concept of Quality-of-Service QoS support is very important in this context because it can provide good performance to sensitive traffic while the total traffic is in an overload situation. So the differentiation of services is a key to avoid frustration due to failing service in future radio networks.

On the Data Link Layer (DLL), more precisely in the Medium Access Control (MAC) layer, there is usually no knowledge of applications, services and their demand. Decisions of resource and packet scheduling are taken in this layer 2. Therefore in order to support QoS, packet schedulers and queues must be aware of QoS classes. QoS classes are mapped to priorities which can be easily treated with static priority scheduling in $O(1)$. Flow identification and handling is responsible for providing the knowledge into layer 2. Note that resource scheduling and packet scheduling are two distinct tasks [1]. Related work has identified these issues recently [2].

In this paper the concept of flows [3] is discussed as a new function in the DLL of a cellular mobile radio system. It supports fixed relay stations [4], [5] and is used to support QoS scheduling [1]. The aspects of flow establishment, management and handover signaling are treated. Finally the static priority scheduling is applied to multiple flows and simulation results show the separation in single and multihop scenarios.

II. The Concept of DLL Flows

The support of QoS requirements works basically by prioritizing flows belonging to QoS classes with stringent delay requirements upon those with relaxed requirements or best-effort as last priority. In order to be able to distinguish different flows there must be a mechanism to uniquely identify flows. The concept of flows is not limited to the use of priorities in the scheduler. There can be any scheduler and a flow manager maintains the specification of parameters for each flow to support individual scheduling goals. A flow can be understood as a DLL connection, defined it in the following way [3]:
A flow is a logical group of packets which have a common attribute. This attribute may be the QoS class or the application the packets belong to.
Flows cannot be distinguished with the information available in the DLL or the Physical Layer (PHY), so information from higher layers is needed in order to be able to

decide when a new flow shall be established and released respectively. Usually different flows are identified uniquely by the quadruple of source IP address, destination IP address, source port number and destination port number. In this approach an IP Convergence Layer ($IPCL$) reads the TCP/IP- and UDP/IP-headers. Furthermore a cross-layer interface for QoS aware requests by e.g. the application on top of the $TCP/UDP/IP$ protocols is necessary. Even if it is decided how to distinguish the different flows, it is still a challenge to handle, i.e. establish and release the flows, especially in the case of supporting multiple hops. Packets belonging to the same flow are labeled with the same DLL flow ID. Figure 1 shows the protocol layers which are aware of flow IDs. The flow IDs are valid in the hatched protocol layers. The figure shows a relay node (RN) and the Radio Access Network Gateway (GW) [6]. The introduced concept is directly applicable to LTE-Advanced [7] and WiMAX [8], since in all these systems $decode\text{-}and\text{-}forward$ Layer 2 relays are a part of the specification [9]. Besides the DLL also the

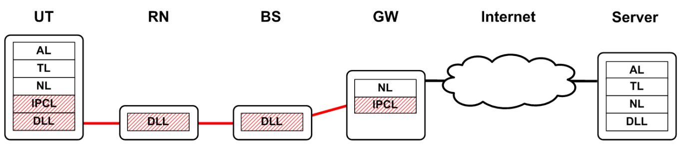


Fig. 1: Protocol layers with knowledge about DLL flow IDs

$IPCL$ has knowledge about DLL flow ID. Therefore the User Terminal (UT) and the GW are the endpoints of the flow management concept, because these two types of stations have an $IPCL$. The $IPCL$ has the ability to analyze TCP-/UDP-/IP headers and is so able to map IP packets to DLL flows and vice versa. Through the ability to identify the flow a packet belongs to, packets of higher layer applications can be identified and mapped to their QoS needs and handled accordingly, e.g. by different ARQ instances for different flows or prioritized resource scheduling which is illustrated in figure 2.

Here hop-wise valid flow IDs are assigned which are stored in tables and are switched from incoming to outgoing flow IDs. For a discussion of global versus local flow IDs see [3]. The hop-wise change of the color indicates that the flow ID

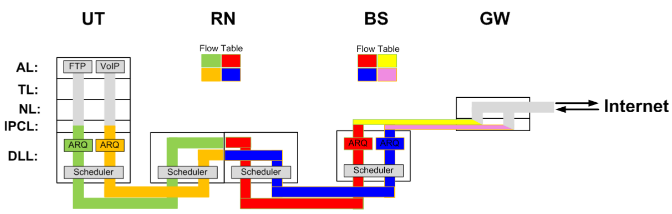
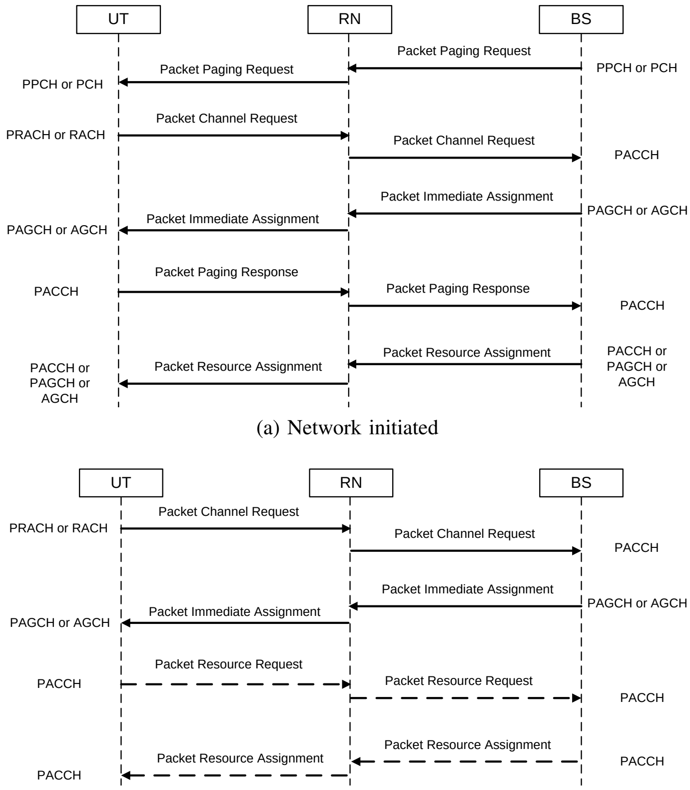


Fig. 2: Hop-wise valid DLL flow IDs

is switched in the RN and in the BS. This approach implies that the responsibility of assigning flow IDs is borne by the RAPs and the GW respectively for their own domain. The flow IDs are symmetric, i.e. they are valid both in DL and UL, since there is no reason to have different IDs. So, the complexity can be reduced. However, everything which was introduced up to now is only valid for the user plane. User plane data has an end-to-end validity between UTs and the GW. Therefore the flow IDs have to be switched hop-wise in the whole RAN. This is not valid for the control plane data. Control plane data is only valid per hop [3]. It is assigned during the initial access to the network which is described in section V.

III. Flow Establishment

The assignment of flow IDs is done during the flow establishment signaling which is shown in the MSC in figure 3. The LTE channels used for these messages are also shown. It can either be network initiated (incoming call) or UT initiated (outgoing call). The flow release signaling can be found in [3].



(a) Network initiated

(b) UT initiated

Fig. 3: Multihop flow establishment signaling

Not only the RAPs have to administer flow ID tables, but also UTs and GWs. However, the aim of these tables is slightly different. Their purpose is to map higher layer packets to the correct DLL flow ID and vice versa. The mapping is done in the $IPCL$, since it is necessary to read the TCP-, UDP- and IP-headers to be able to distinguish packets from different applications.

IV. Cross-Layer Flow Setup Sequence

Each session of an application can uniquely be distinguished by the quadruple of source and destination IP addresses and source and destination TCP or UDP port numbers.

The triggering of the flow establishment originates in the application layer (AL). This can be triggered by the UT (Figure 3b) or network initiated. Figure 4 shows only the first case, since applications are generally started by the user. The cross-layer signaling consists of these steps:

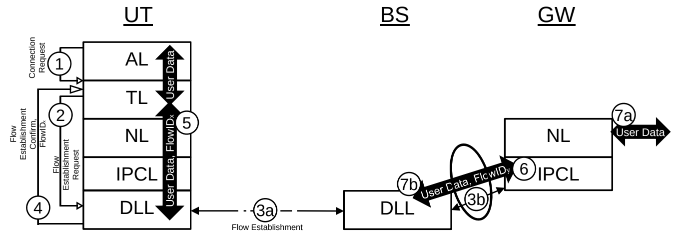


Fig. 4: Cross-layer flow establishment triggered by the AL

1) When an application session starts, the application requests a TCP or UDP socket respectively, in order to be able to send its user data.
2) The TL requests a DLL flow ID from the DLL.
3) In the DLL the flow establishment process with the signaling shown in figure 3 is started.
4) This process provides a new flow ID which is delivered to the TL. Now the AL can be answered that the requested socket is ready to receive user data, at least in case of UDP as TL protocol.

5-7) In case of TCP after having established the DLL flow, the TCP connection has to be established. The TCP *three-way handshake* is already sent over the newly created flow.

From now on all user data written into the new TL socket can be sent using this new DLL flow. The flow ID can either be appended to a data packet explicitly or it can be determined by the $IPCL$ implicitly.

- The flow ID is handed over to the lower neighboring layer explicitly.
- The flow ID is stored in the IP header explicitly, e.g. in the *flow label* field of the $IPv6$ header or in the *options* field of the $IPv4$ header.
- On demand establishment of a new flow: The $IPCL$ maps the quadruple of source and destination IP addresses and source and destination TCP/UDP port numbers to the new flow ID.

Figure 5 shows the the ISO layers and their understanding of a connection. In all these cases the determination of the correct flow ID is handled locally in the same protocol stack. After determining the correct flow ID the packets are transferred to the GW like shown in figure 2. As soon as a packet arrives at the $IPCL$ of the GW in the UL, again a mapping between the quadruple of source and destination IP addresses and source and destination TCP/UDP port numbers to the DLL flow ID is done. So, packets in the DL can be assigned the correct DLL flow ID. The flow release is triggered by the AL in the same way the flow establishment was done [3].

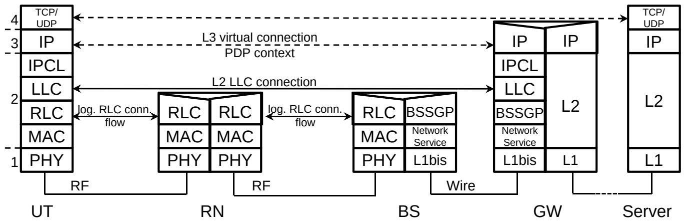
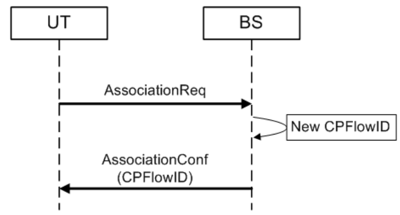
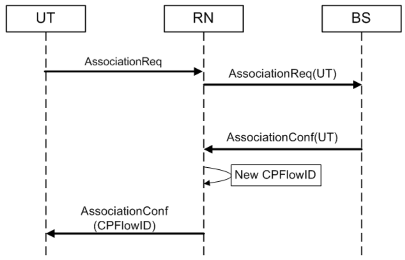


Fig. 5: Protocol Stack for Multihop Flow ID support



(a) Single-hop



(b) Multi-hop

Fig. 6: Association signalling

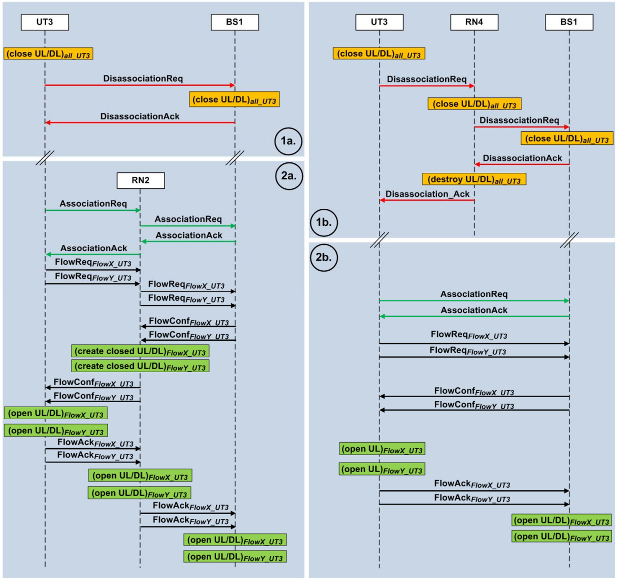
V. Handover and Association

The control plane flow ID is assigned during the initial access to the RAN (see in section II). This *association* is shown in Figure 6 as message sequence chart for the single and multihop case. Handover is the process of changing the association. We distinguish the following cases:

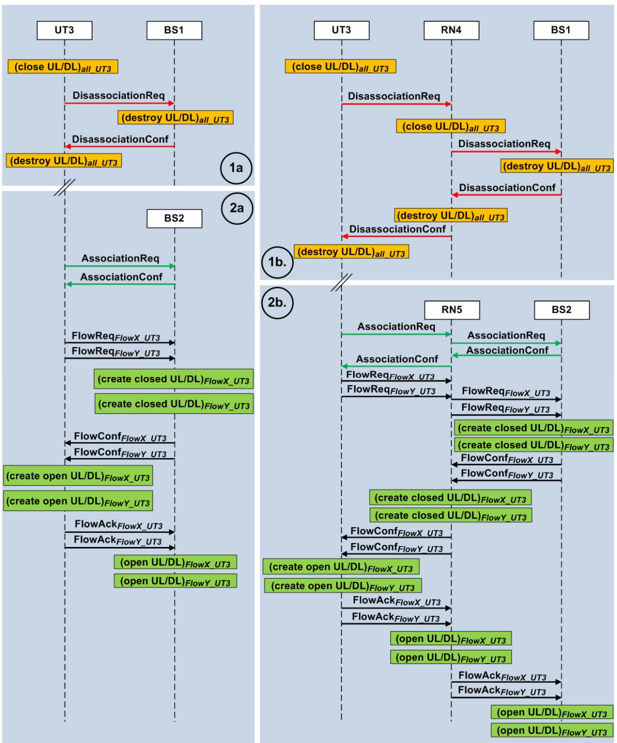
- Intra-REC handover is a handover between a BS and a connected RN. The UT still remains in the same Relay Enhanced Cell (REC).
  - $BS_x \leftrightarrow BS_xRN_{x1}$
  - $BS_xRN_{x1} \leftrightarrow BS_xRN_{x2}$
- Inter-REC handover is a handover between RAPs, either BS or RN belonging to different RECs.
  - $BS_x \leftrightarrow BS_y$
  - $BS_x \leftrightarrow BS_yRN_{y1}$
  - $BS_xRN_{x1} \leftrightarrow BS_yRN_{y1}$

VI. Context Transfer in Handover

In order to make the handover seamless from an application point of view, it is necessary to transfer the current context from the old to the new RAP during a handover. The context to be transferred are the existing flows, more precisely the user plane flows, since the control plane flow has to be newly established during the association to the new RAP as shown in figure 6. In an inter-REC handover case the flows have to



(a) Intra-REC handover



(b) Inter-REC handover

Fig. 7: MSC of Handover Signalling

be transferred via the backbone over the GW. However, in an intra-REC handover case there is no need for a context transfer, since flows are already existent. A detailed description of the context transfer and preservation respectively and a quantitative analysis of the different handover scenarios are given in [10], however for only one user plane flow which is directly established during the association. Therefore in figure 7 the extension of the handover signalling for the case of multiple flows is shown.

In each handover case first the UT disassociates from the old RAP and afterwards associates to the new RAP. After the association to the new RAP the flow signalling is started. Since the signalling has already been described in section III, now only the differences between the inter- and intra-REC handovers are explained. In the inter-REC handover all protocol instances, like e.g. ARQ instances, belonging to a flow are deleted in all stations. They are built up again by means of the flow establishment procedure in the new cell, because in the inter-REC handover case a new flow ID is assigned by the new BS. However, in the intra-REC handover case indeed the next-hop RAP changes, but the BS is still the same. Therefore the BS and the UT can keep the old protocol instances, because they will still be involved in the data transmission after the handover. The only thing which has to be adapted is the last-hop flow ID at the UT, since it definitely will be changed. If the RN was involved before the handover it just has to delete its context, if it involved after the handover it just has to create new instances for the new flow.

Since keeping a handover seamless is a very sensitive issue, in figure 7 not only the creating and deleting of flow contexts occurs, but also the intermediate interruption of flows in terms of closing existent, i.e. already created, flows. The purpose is to have a mechanism to identify packets which are in general valid, i.e. to be delivered finally but not currently, because the flow they belong to will be valid after the handover.

VII. Scheduling on Flows

Once the flows are established and known in the layer 2 together with their QoS profile (kept in a table a flowmanager functional unit), packets are queued separately per flow which allows to access the head-of-line (HOL) packet out of each queue independently. This allows for all scheduling strategies to be implemented. In contrast, having only one queue per UT would inhibit all except FCFS scheduling which doesn't support QoS at all [11].

Separating real-time from best effort traffic is very easily achieved in O(1) by a static priority scheduler which only needs to know the priority classes of each $flowID$. This concept allows to accept VoIP traffic up to the same limit as if the network is free, no matter how high the offered load of an inferior traffic class is. Figure 8 shows the scheme and an exemplary result in a frequency/time resource grid. The resource allocation scheduling must only know the number of bits per priority level and UT, which is available by the handling per $flowID$. Then the packet scheduler can determine the order of packets inside a priority queue (*subscheduler*). It is assumed that best effort (background) traffic is always available (overload situation). Therefore the subscheduler needs to provide fairness. The most simple assumption is Round Robin (RR, $O(1)$), but Proportional Fair (PF, $O(\#Flows)$) has proven to be fair even in heterogeneous scenarios [12].

A. Scheduling Results

Simulations have been performed using the OpenWNS simulator [13] with the parameters of a SISO LTE system in FDD mode using $20MHz$ DL bandwidth [14] and a realistic

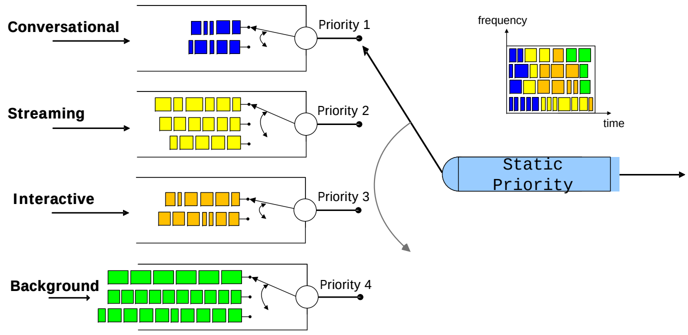


Fig. 8: Hierarchical static priority scheduling

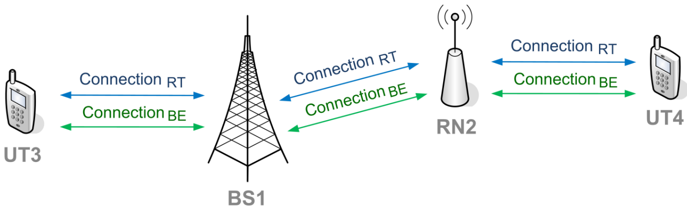
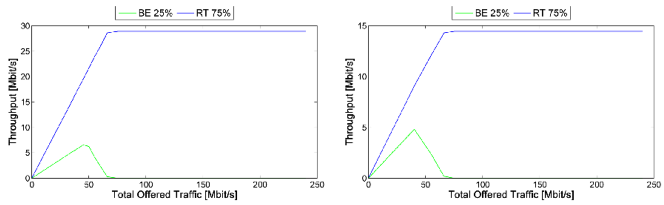


Fig. 9: Single- and Multihop Scenario with two QoS classes

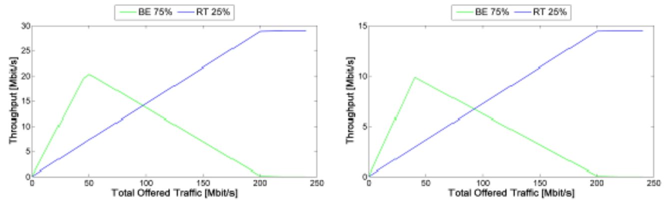
layer 2 implementation. Most important is the scenario setup, as shown in Figure 9. It captures the single and multi hop cases and two QoS classes, namely Real Time (RT=Conversional) and Best Effort (BE=background). The single hop transmission to UT3 and the multihop transmission to UT4 together cover the cases in a relay-enhanced scenario. Without loss of generality the UTs are located on positions such that they achieve the highest PhyMode $QAM64\frac{5}{6}$. The total offered load is $R_{tot} = R_{3,RT} + R_{4,RT} + R_{3,BE} + R_{4,BE} = v \cdot R_{RT} + (1-v) \cdot R_{BE}$. The results shown in Figure 10 show the received data rate (net throughput above layer 2) when the total offered load is increased, while the ratio v between RT and BE traffic is kept constant. As can be seen, there is no influence of the BE traffic on the RT traffic (blue), only in the opposite way (green). RT traffic can be carried up to the maximum cell throughput, while BE traffic is successfully suppressed but can still fill up the remaining capacity left over by the RT traffic. This is the desired behaviour for singlehop terminals as can be seen on the left. It works hassle-free in the multihop case (on the right). In this case the total throughput is limit due to twice the resources needed.

VIII. Conclusion

This paper presents the flow management framework for future multi-hop (*decode-and-forward* layer 2 relays) mobile radio systems and applies it to QoS scheduling with static priorities. The flow concept allows to distinguish application sessions in the Data Link Layer. Flow establishment, signalling and handovers in the Data Link Layer are discussed. The flow management concept is the key enabling method for the support of Quality-of-Service. By distinguishing application sessions on the Data Link Layer, flows are assigned priorities to distinguish QoS requirements. The simulation results show good realtime from best effort traffic separation in a single- and



(a) BE:25%, RT:75%



(b) BE:75%, RT:25%

Fig. 10: Throughput results of singlehop(l) and multihop(r) scenarios with rising offered load (R_{tot}) and two classes of traffic priorities.

multihop scenario. All the described signaling is implemented in the simulator [13] and validated by simulation.

Future research should investigate heterogeneous traffic and QoS requirements that stress the fairness and delay capabilities of substrategies below the static priority scheme.

References

[1] R. Schoenen and F. Qin, "A Closed Loop Control Model separating Resource and Packet Scheduling in Multihop Cellular Networks," in *Proc. of the European Wireless*, Lucca, Italy, Apr 2010, p. 6.

[2] M. Marchese and M. Mongelli, "Vertical QoS Mapping over Wireless Interfaces," *Wireless Communications, IEEE*, vol. 16, no. 2, pp. 37–43, April 2009.

[3] A. Otyakmaz, D. Bültmann, R. Schoenen, and I. Durmaz, "On Flow Management for Future Multi-Hop Mobile Radio Networks," in *IEEE WiCom*, Beijing, China, Sep 2009, p. 6.

[4] R. Pabst, B. Walke, D. C. Schultz, H. Yanikomeroglu, and et al, "Relay-Based Deployment Concepts for Wireless and Mobile Broadband Radio," *IEEE Communications Magazine*, pp. 80–89, Sep 2004.

[5] H. Yanikomeroglu, "Fixed and mobile relaying technologies for cellular networks," in *Proc. of ASWN'02*, Jul 2002, pp. 75–81.

[6] http://www.ist-winner.org/.

[7] http://www.3gpp.org/Highlights/LTE/LTE.htm.

[8] http://www.wimaxforum.org/.

[9] J. Monserrat, P. Sroka, G. Auer, J. Cabrejas, D. Martin, A. Mihovska, R. Rossi, A. Saul, and R. Schoenen, "Advanced Radio Resource Management for IMT-Advanced in WINNER+ (II)," in *Proceedings of the ICT-MobileSummit 2010*, Florence, Italy, Jun 2010, p. 9.

[10] A. Otyakmaz, L. Ding, and R. Pabst, "A Multi-Mode MAC Protocol Integrating FDD and TDD for Next Generation Multi-Hop Mobile Radio Networks," in *Proceedings of 17th ICT Mobile Summit 2008*, Stockholm, Sweden, Jun 2008, p. 8.

[11] J. Sanjay and H. Mahbub, *Engineering Internet QoS*. Artech House, 2002, ISBN 1580533418.

[12] R. Schoenen, A. Otyakmaz, and B. Walke, "Concurrent Operation of Half- and Full-Duplex Terminals in Future Multi-Hop FDD Based Cellular Networks," in *Proceedings of the IEEE WiCom'2008*, Dalian, China, Oct 2008, p. 6.

[13] D. Bültmann, M. Muehleisen, K. Klagges, and M. Schinnenburg, "openWNS - open Wireless Network Simulator," in *15th European Wireless Conference 2009 Electronic Proceedings*. Aalborg, Denmark: VDE VERLAG GMBH, May 2009, pp. 205–210.

[14] R. Schoenen, R. Halfmann, and B. Walke, "An FDD Multihop Cellular Network for 3GPP-LTE," in *Proceedings of the IEEE Vehicular Technology Conference (VTC'spring'08)*, Signapore, May 2008, p. 6.