On Flow Management for Future Multi-Hop Mobile Radio Networks

Arif Otyakmaz, Daniel Bueltmann, Rainer Schoenen and Ismail Durmaz
Department of Communication Networks at RWTH Aachen University, Faculty 6, Germany
{aoz|dbn|rs|isd}@comnets.rwth-aachen.de

***Abstract*— Next generation mobile radio networks of IMT-Advanced systems family will offer ubiquitous broadband high area coverage, at up to 1 GBit/s in cities and 100 MBit/s in rural areas and QoS support in terms of throughput and low delay. Candidate technologies like 3GPP-LTE, WiMAX, as well as the WINNER system design are based on OFDMA transmission for flexible radio resource allocation, scalable and adaptable to both short range and wide area scenarios. Multi-hop relaying is part of all system concepts, in order to cover huge radio cell areas with a reduced number of base stations.**
Moreover, no future mobile radio system can afford to neglect supporting the increasing QoS demand of multimedis services, like high quality video streaming or VoIP. These services may have different QoS need. So, mechanisms have to be provided by future mobile radio systems to distinguish packets of different applications. Therefore in this paper a flow management concept for multi-hop mobile radio systems is introduced which can map packets of applications to reserved DLL connections, so called flows.

***Index Terms*— Flows, Relaying Multi-hop, QoS, WINNER, LTE, Protocol**

I. Introduction

MOBILE radio networks in the future will usually try to be one step ahead of the customers' demand. So the data rates a cellular system can offer will increase up to aggregated $1Gbit/s$ in microcells. In the same time applications and user demand increase more and more. Wasteful offers like traffic flatrates subserve this ever-ongoing tendency. Operating systems and application programs freely update themselves frequently over any network these times. It takes more and more financial and technical effort to support this dubious demand. 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 and the technical result is that competing companies spend an enormous effort to come up with algorithms to better utilize the wireless channel.

We must realize that capacity cannot always be overprovisioned. In the long term the growth will come to an end and in the short term traffic is always bursty in nature and therefore procuces overload situations in orders of $10ms$ to several seconds. The concept of quality-of-service (QoS) support is 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 key to avoid frustration due to failing service in future radio networks.

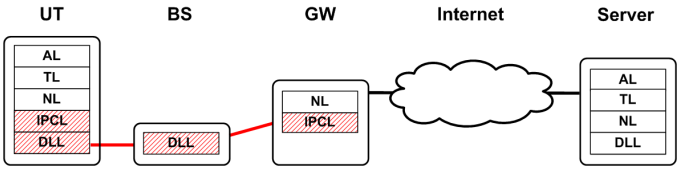
On the data link layer (DLL) and medium access control (MAC) layer there is usually no knowledge of applications, services and their demand. In most cases it is even connectionless in the sense of ISO-OSI. But decisions of resource and packet scheduling are taken in this layer 2. If QoS support must work in the future, it is unavoidable to have schedulers and queues being aware of that. In order to differentiate QoS, the queued packets must be distinguished somehow. That is what a flow identification and handling is responsible for.

In this paper the concept of flows in the DLL is introduced. Aspects of flow establishment, release and management are discussed. The handover procedure is also extended to support the change of controlplane flows. Finally the important cross-layer aspects are discussed.

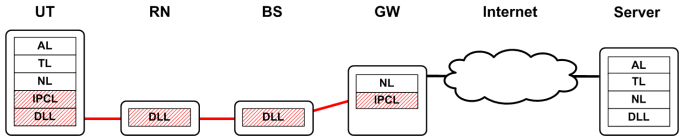
II. Assignment, Validity and Usage of DLL Flows

In order to be able to distinguish different flows and hence support Quality-of-Service (QoS) requirements by prioritising flows belonging to QoS classes with higher priorities, a mechanism to uniquely identify flows must be used. A flow can be understood as a Data Link Layer (DLL) connection. We define it in the following way:
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.
Unfortunately flows cannot be distinguished with the information available in the DLL not to mention in the Physical Layer (PHY). Information from higher layers is needed, in order to be able to decide when a new flow or synonymously connection shall be established and released respectively. One possibility to identify different flows uniquely is the quadruple of source IP address, destination IP address, source port number and destination port number. In this approach an IP Convergence Layer ($IPCL$) would read 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 would be 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 the Radio Access Network Gateway (GW) of the Wireless World Initiative New Radio ($WINNER$)

system concept [1]. However The introduced concept is also directly applicable to 3GPP-LTE [2] or WiMAX [3], since in all these systems it is envisaged to integrate the usage of $decode\text{-}and\text{-}forward$ Layer 2 relays. Besides the DLL also



(a) Single-hop



(b) Multi-hop

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

the $IPCL$ 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 analyse 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 prioritised resource scheduling which is illustrated in figure 2.

Adding a flow ID to the packets, obviously increases the

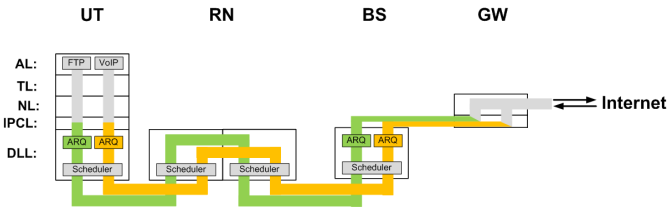


Fig. 2: Flow based ARQ and resource scheduling

signalling overhead and of course it is desired to minimise the length of the flow ID field in the protocol header. However, the length depends on the area of validity of the flow ID. The larger the area of validity in the Radio Access Network (RAN) is, the larger the flow ID field has to be, because its length determines the number of parallel flows in a domain. Figure 3 illustrates the different domains of flow IDs. Now flow IDs can either be unique in the whole GW domain or only in the Base Station (BS) domain or even only in the RAP (BS or Relay Node (RN)) domain, i.e. only in the RN domain in case of multi-hop UTs. The first option would have the highest signalling overhead, because the flow IDs needed to be unique within the whole GW domain, but

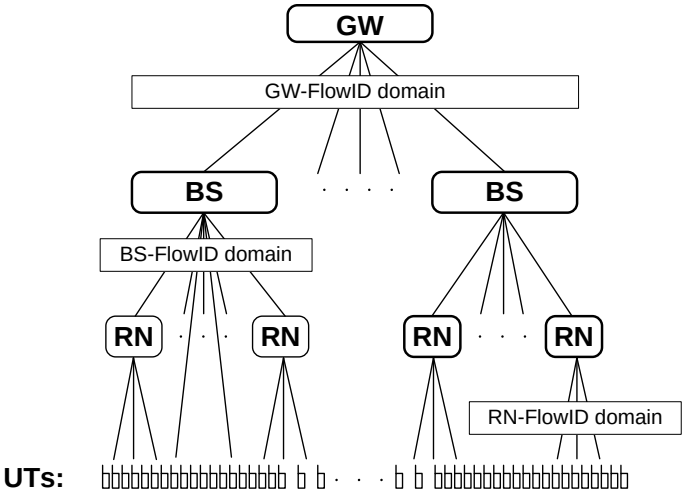


Fig. 3: Flow ID domains in the RAN

would require the lowest hardware costs for RAPs. In this case RAPs would not need to administrate flow ID tables, in order to be able to switch incoming to outgoing flow IDs. The alternative approach, which is preferred and is taken as a basis in the following, is to assign only locally, say hop-wise valid flow IDs which are stored in tables and are switched from incoming to outgoing flow IDs. This approach would minimise the signalling overhead, but increase the hardware costs in terms of required memory. However, these hardware costs can be neglected nowadays. The latter approach is illustrated in figure 4. In contrast to figure 2 the hop-wise

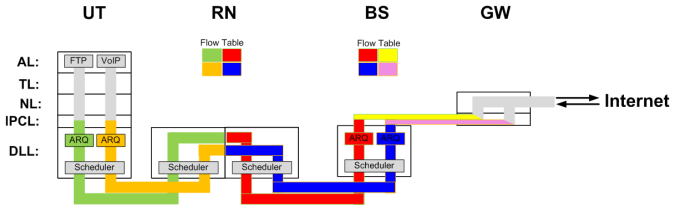


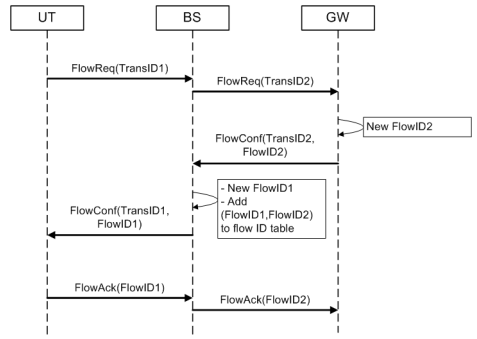
Fig. 4: Hop-wise valid DLL flow IDs

change of the colour indicates that the flow ID 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 as illustrated in figure 3. 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, but it is necessary directly when initially entering the radio cell. Therefore it is assigned during the initial access to the network which is described in section V. Also the RNs get a control plane flow when they initially connect to their BS. They use it for example for the UL resource requests for the first hop

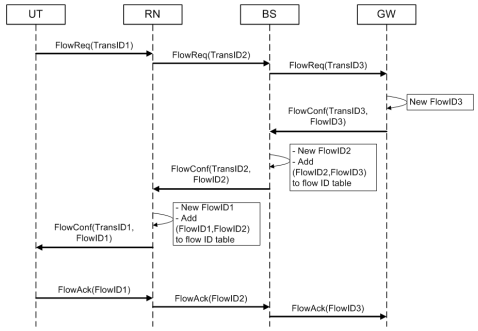
where they accumulate all resource requests of their multi-hop UTs sent to them on the second hop. Therefore the RNs again have to switch the flow based resource requests to the correct next-hop flow like it is also done for the user plane data. Obviously the control plane flow has to be treated with the highest priority during the radio resource scheduling.

III. Flow Signalling

The assignment of flow IDs is done during the flow establishment signalling which is shown in the MSC in figure 5, whereas the signalling for the flow release can be seen in the MSC in figure 6.



(a) Single-hop



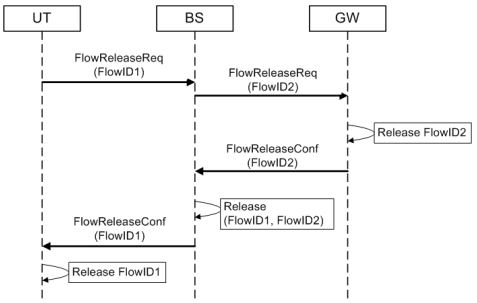
(b) Multi-hop

Fig. 5: Flow establishment signalling

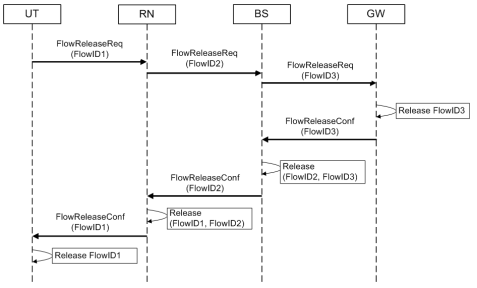
In contrary to the flow release, which is a two-way-handshake, the flow establishment is a three-way-handshake. The reason is that it has to be ensured that the UT definitely has received the $FlowConf$. Otherwise it could be possible that packets in the DL are sent to the UT over the new flow, but the UT still has not received the $FlowConf$.

Furthermore, 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.

Each session of an application can uniquely be distinguished by the quadruple of source and destination IP addresses (see figure 8) and source and destination TCP or UDP port numbers (see figure 7). After having this information the mapping can be done in two different ways. Either for each

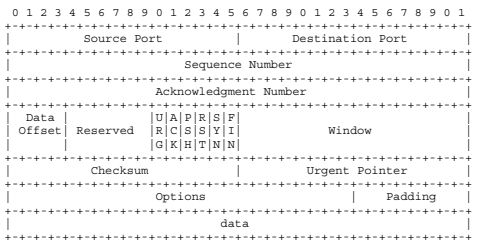


(a) Single-hop

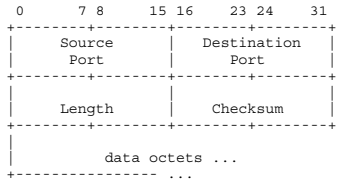


(b) Multi-hop

Fig. 6: Flow release signalling



(a) TCP header [4]



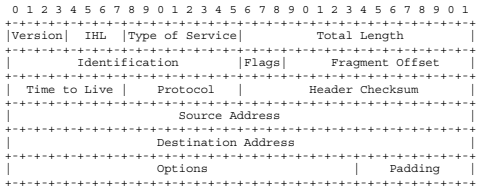
(b) UDP header [5]

Fig. 7: TCP and UDP header formats

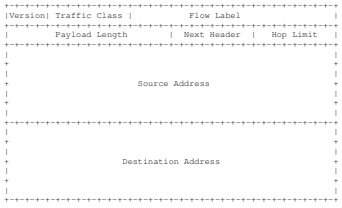
session a new flow is built and so only packets of one and the same session are mapped to the same flow or several sessions are multiplexed to one flow, e.g. sessions with the same QoS requirements. The two different approaches are illustrated in figure 9. Since the first approach is more flexible and allows a more accurate support of the different QoS needs of different applications, in the following the first approach is taken as a basis.

IV. Cross-Layer Aspects

Having described the flow management aspects in the DLL, like flow establishment signalling, flow ID assignment, usage and mapping, still the cross-layer triggering of the flow establishment and release by the Application Layer (AL)

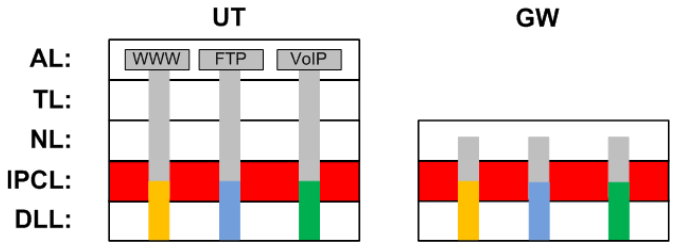


(a) $IPv4$ header [6]

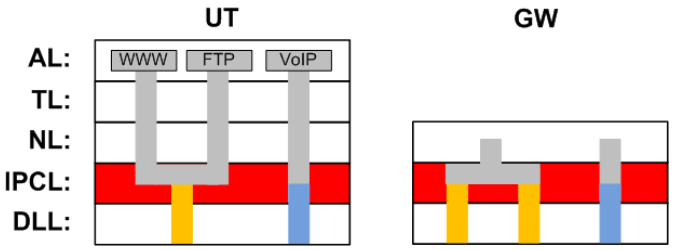


(b) $IPv6$ header [7]

Fig. 8: IP header formats



(a) Flow ID per application session



(b) Multiplexing of multiple application sessions onto one flow ID

Fig. 9: Different types of DLL flow IDs

has to be described. The sequence of the cross-layer flow establishment triggered by the AL is shown in figure 10. In this concept the flow establishment is triggered by the UT, since applications are generally started by the user. However, the concept could easily be adapted to allow also network initiated flow establishments. When an application session starts, the application requests a TCP or UDP socket respectively, in order to be able to send its user data. Hereupon the TL requests a DLL flow ID from the DLL. In the DLL the flow establishment process with the signalling shown in figure 5 is started. 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. In case of TCP before the TCP connection establishment is done 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. Therefore the flow ID

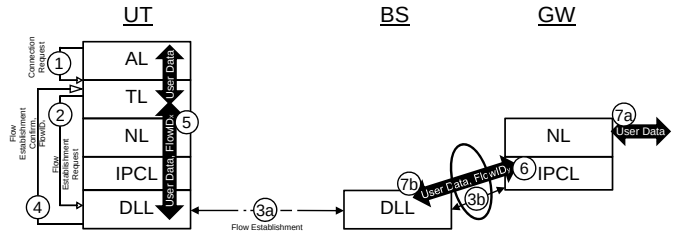


Fig. 10: Sequence of cross-layer flow establishment triggered by the AL

can either be appended to a data packet explicitly or it can be determined by the $IPCL$ implicitly.

- The flow ID can be handed over to the lower neighbouring layer explicitly.
- The flow ID can be stored in the IP header explicitly, e.g. in the *flow label* field of the $IPv6$ header (see figure 8b) or in the *options* field of the $IPv4$ header (see figure 8a).
- When establishing a new flow on demand of the TL, the $IPCL$ can map the quadruple of source and destination IP addresses and source and destination TCP/UDP port numbers to the new flow ID. So, each IP packet to be sent can be uniquely mapped to its DLL flow ID.

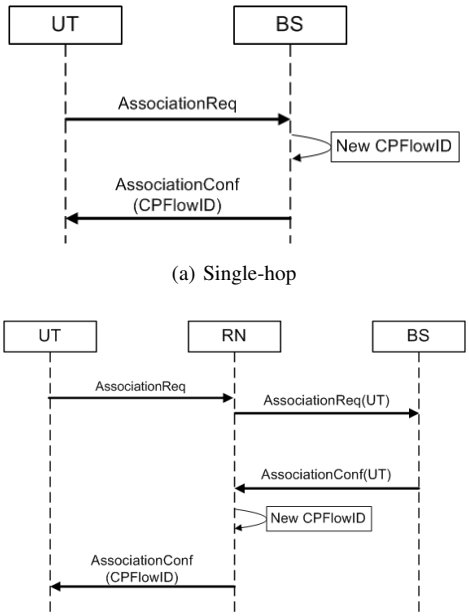
However, in all three cases the determination of the correct flow ID is handled locally in the same protocol stack. After determining the correct flow ID the packets are transferred to the GW like shown in figure 2 or figure 4. As soon as a packet arrives at the $IPCL$ of the GW in the UL, again a mapping 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.

After an application session is finished, the flow release can be triggered by the AL in the same way the flow establishment was done. The signalling of the flow release was shown in the MSC of figure 6. However, the flow release is only possible due to the explicit cross-layer signalling. If the cross-layer signalling were not explicit, but a flow establishment would be started implicitly every time a TCP/IP or UDP/IP packet with an unknown above mentioned quadruple arrives at the DLL, for each new flow a timer would have to be set which would determine the invalidity of a flow [8], [9]. Moreover, the explicit triggering of the flow establishment allows the signalling of QoS requirements of the application.

V. Handover Aspects

As mentioned in section II the control plane flow ID is assigned during the initial access to the RAN. The according signalling is shown in the MSC of figure 11. In cellular multi-hop radio system various types of handovers are possible. In general they can be divided into two main categories:

- Intra-REC handover is a handover between a BS and a connected RN. The UT still remains in the same Relay Enhanced Cell (REC).



(a) Single-hop

(b) Multi-hop

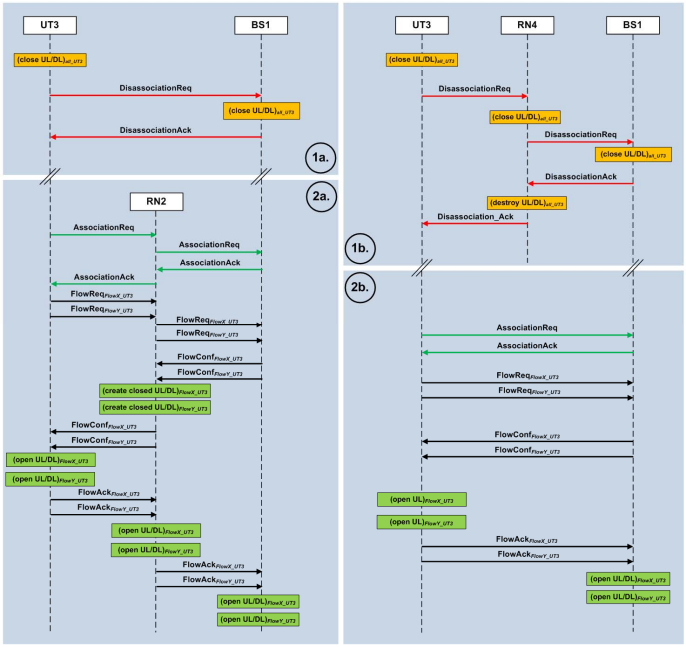
Fig. 11: Association signalling

- Inter-REC handover is a handover between RAPs, either BS or RN belonging to different RECs.

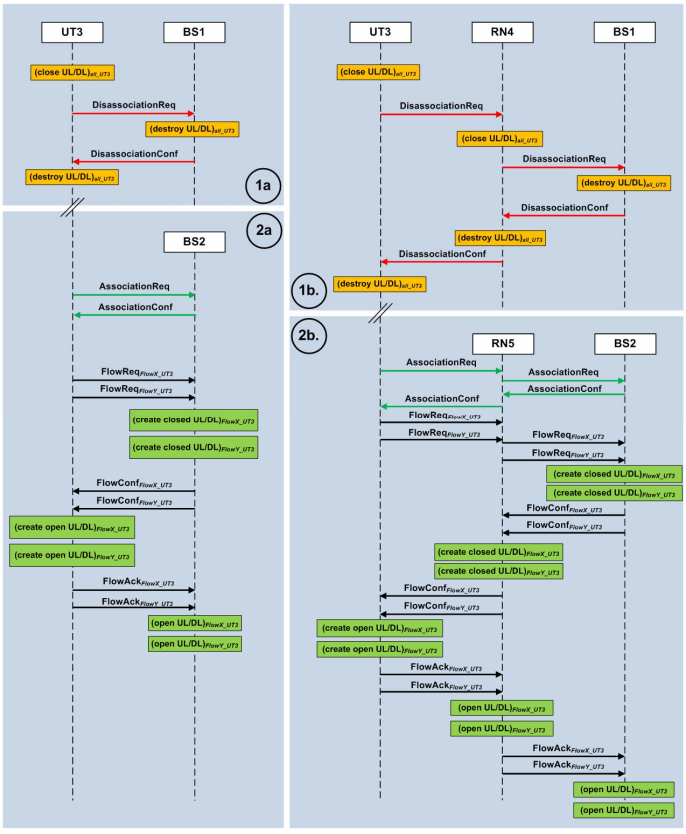
The different handover types are:

- Inter-REC handovers
  - $BS_x \leftrightarrow BS_y$
  - $BS_x \leftrightarrow BS_y RN_{y1}$
  - $BS_x RN_{x1} \leftrightarrow BS_y RN_{y1}$
- Intra-REC handovers
  - $BS_x \leftrightarrow BS_x RN_{x1}$
  - $BS_x RN_{x1} \leftrightarrow BS_x RN_{x2}$

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 plan flows, since the control plane flow has to be newly established during the association to the new RAP as shown in figure 11. In an inter-REC handover case the flows have to 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. So, only a context preservation is necessary meaning that e.g. existent ARQ instances (see figure 2) can be kept and only the UT has to adapt the flow ID of the existent flow, because the BS can still use the old flow ID. A detailed descripiton 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 12 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



(a) Intra-REC handover



(b) Inter-REC handover

Fig. 12: Handover signalling

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 senitive issue, in figure 12 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.

VI. Conclusion

This paper presents a concept for a flow management for future multi-hop mobile radio systems. The flow concept allows to distinguish application sessions in the Data Link Layer. The concept comprises the cross-layer triggering of flow establishment and release, the respective signalling in the Data Link Layer, the hierarchical assignment of flow IDs and their usage in the different hops of the Radio Access Network. For all aspects of the introduced concept the focus is especially on integrating $decode\text{-}and\text{-}forward$ Layer 2 relays. Furthermore, the influence of the flow concept on Layer 2 handovers is discussed in various handover scenarios. The flow management concept is the basis for the support of Quality-of-Service. By the possibility of distinguishing application sessions on the Data Link Layer a possibility is given to prioritise flows with higher Quality-of-Service requirements.

Future research should investigate methods which have Quality-of-Service-aware scheduling goals that meet traffic contract requirements.

References

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

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

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

[4] Jon Postel, "RFC 793: Transmission Control Protocol," Internet Engineering Task Force, Tech. Rep., 1981. [Online]. Available: http://www.ietf.org/rfc/rfc793.txt

[5] J. Postel, "RFC 768: User Datagram Protocol," Internet Engineering Task Force, Tech. Rep., 1980. [Online]. Available: http://www.ietf.org/rfc/rfc768.txt

[6] Jon Postel, "RFC 791: Internet Protocol," Internet Engineering Task Force, Tech. Rep., 1981, available at http://www.ietf.org/rfc/rfc791.txt. [Online]. Available: http://www.ietf.org/rfc/rfc791.txt

[7] S. Deering and R. Hinden, "RFC 2460: Internet Protocol, Version 6 (IPv6) Specification," Internet Engineering Task Force, Tech. Rep., 1998, available at http://www.ietf.org/rfc/rfc2460.txt. [Online]. Available: http://www.ietf.org/rfc/rfc2460.txt

[8] R. Kreula and H. Haapasalo, "Transfer Delay at ATM LAN Emulation and Classical IP over ATM," in *ICC*, 1997.

[9] M. Laubach, "RFC 1577: Classical IP and ARP over ATM," Internet Engineering Task Force, Tech. Rep., 1994. [Online]. Available: http://www.ietf.org/rfc/rfc1577.txt

[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. [Online]. Available: http://www.comnets.rwth-aachen.de