Reliable File Distribution over Mobile Broadcast Systems

Von der Fakultät für Elektrotechnik und Informationstechnik der
Rheinisch-Westfälischen Technischen Hochschule Aachen zur Erlangung
des akademischen Grades eines Doktors der Ingenieurwissenschaften
genehmigte Dissertation

vorgelegt von

Diplom-Ingenieur

Thorsten Lohmar
aus Monheim, Deutschland

Berichter: Univ.-Prof. Dr. Petri Mähönen
Univ.-Prof. Dr. Rudolf Mathar

Tag der mündlichen Prüfung: 17. Oktober 2011

*Diese Dissertation ist auf den Internetseiten
der Hochschulbibliothek online verfügbar.*

Abstract

The 3GPP Multimedia Broadcast and Multicast Service (MBMS) provides new bearer services and procedures for efficient transmissions to large user groups. When the group is large, MBMS distributes content by using broadcast on the air-interface. This thesis evaluates the use of MBMS for reliable file distribution services. One important requirement for file distribution is that the files contain no transmission errors.

The MBMS file distribution process is subdivided into two phases in this thesis: During the first phase, the radio access network sends the IP packets in each cell either using one broadcast channel or several ptp channels depending on the number of receivers. During the second phase, the file repair service is executed when needed. The file repair service uses either HSPA bearers or MBMS bearers. It is mandatory when minimizing the needed resources for reliable file transmission.

In order to understand the transmission characteristics of the first phase, we analyze the packet transmission over the MBMS traffic channel (MTCH). The use of shorter IP packets leads to a lower IP packet error probability on the MTCH. When using shorter IP packets, a larger share of bits is spent on packet headers. To evaluate the information throughput over MTCH, we define the goodput as the fraction between received information bits and sent data bits. IP packets smaller than 500 Byte lead in case of block error rates larger than 10% to a higher goodput.

We evaluate different optimization targets for MBMS file delivery. The most important evaluation target is to balance both transmission phases. The resource usage for the MBMS transmission is balanced with the resource needs for the file repair in order to increase the system efficiency of the file distribution of a certain size to all receivers. It is possible to trade the transmit power with the amount of application layer FEC redundancy at same load for the file repair service. The Raptor FEC is used for MBMS. Additional FEC redundancy increases the needed transmission energy, since the system resources are used for a longer time.

The point-to-point file repair uses unicast HTTP connections and spreads the repair requests in a time window. The receivers draw randomly a start time out of a given *wait-time window*. The link between the file repair server and the system limits significantly the serving time and may even lead to an under utilization of the radio resources. The PTP file repair is well dimensioned when the radio links of all active file repair receivers and the link to the file repair server are just fully utilized. The smallest file repair service duration takes approximately 1.2 times the Sequential Delivery Time of all missing data over the link between the file repair server and the system.

Kurzfassung

Die im 3GPP definierten Multimedia Broadcast and Multicast Services (MBMS) stellen neue Übertragungsdienste und Prozeduren für die effiziente Übertragung an große Benutzergruppen bereit. Wenn die Gruppe groß genug ist ,werden die Inhalte per Broadcast auf der Luftschnittstelle an Gruppen verteilt. Diese Arbeit untersucht die Nutzung von MBMS für skalierbare Push-Datei-Übertragung an große Benutzergruppen. Eine Anforderung bei Dateiübertragungen ist, dass die empfangenen Dateien frei von Übertragungsfehlern sind.

In dieser Arbeit wird der MBMS Übertragungsdienst in zwei Phasen unterteilt. In der ersten Phase sendet das Radio-Zugangssystem die IP Pakete in jeder Zelle in Abhängigkeit der Anzahl der Empfänger über einen Broadcast Kanal oder mehrere ptp Radio-Kanäle. In der zweiten Phase wird bei Bedarf der File Repair Dienst ausgeführt. Der File Repair Dienst kann entweder HSPA (ptp) oder wieder MBMS (ptm) Übertragungsdienste nutzen. Wenn die für zuverlässige Dateiübertragungen genutzten Systemressourcen minimiert werden sollen, dann ist die Nutzung des File Repair Dienstes zwingend notwendig.

Um die Übertragungscharakteristika während der ersten Phase besser zu verstehen, analysieren wir die Paketübertragung über den MBMS Traffic Channel (MTCH). Dabei führt die Benutzung von kurzen IP Paketen zu einer geringeren IP Paketfehlerwahrscheinlichkeit auf dem MTCH. Der Nutzdatendurchsatz ist bei kleineren IP Paketen geringer, da der prozentuale Anteil des IP Headers steigt. Um den Nutzdatendurchsatz über den MTCH zu bewerten, definieren wir das Goodput als Verhältnis zwischen den empfangenen Nutzdaten und den gesendeten Daten. Bei Radioblockfehlerraten größer 10% führen IP Pakete kleiner als 500 Byte zu einem höheren Goodput.

Für die Dateiübertragungen über MBMS untersuchten wir unterschiedliche Optimierungsziele. Das wesentliche Ziel ist das Ausbalancieren der genutzen Ressourcen beider Übertragungsphasen. Dabei wird die genutzen Ressourcen der MBMS Übertragung mit der Ressourcennutzung für den File Repair Dienst ausbalanciert, um die Systemeffizienz einer Dateiübertragung gegebener Größe an alle Empfänger zu steigern. Es ist möglich, die Sendeleistung mit FEC auf Applikationsschicht (AL-FEC) bei gleich bleibender Last für den File Repair Dienst abzuwägen. Für MBMS ist der Raptor FEC vorgesehen. Der Zusatz von AL-FEC erhöht den Energiebedarf, da die Systemressourcen länger benutzt werden.

Der ptp File Repair Dienst nutzt unicast HTTP Verbindungen und verteilt die Anfragen in einem vorgegebenen Zeitfenster. Die Empfänger wählen zufällig eine Startzeit aus dem Fenster aus. Die Verbindung zwischen File Repair Server

und Netz limitiert maßgeblich die Bediendauer und kann auch zu einer ineffektiven Nutzung der Luftschnittelle führen. Der ptp File Repair Dienst ist korrekt konfiguriert, wenn die Radio Übertragung aller aktiver File Repair Sitzungen und auch die Verbindung zum File Repair Server gerade voll ausgelastet sind. Die kleinste File Repair Dienst Dauer ist ungefähr das 1.2 fache der sequentiellen Übertragung aller Daten über die Verbindung zwischen File Repair Server und Netz ist.

Contents

1

Introduction

This thesis evaluates the Push File Delivery over MBMS Point-to-Multipoint UTRAN bearers. A Point-to-Multipoint bearer uses *true* broadcast on the radio interface. The motivation for the work is given in Section 1.1. Then, a summary of the thesis contributions is given in Section 1.2 followed by a description of the structure of this thesis in Section 1.3.

1.1 Purpose and Objectives

Today's mobile communication systems are more and more used to deliver multimedia services like video clips to receivers. The mobile systems are also apable of transporting bandwidth demanding TV services for mobile phones and for laptops. The interest for receiving mobile video services is increasing, boosted by the introduction of a new generation of SmartPhones. Those SmartPhones come with very good multimedia capabilities like large screens and high computing power and *invite* to multimedia consumption.

The number of users is increasing continuously. In some countries, the number of available mobile phone subscriptions is even larger than the population size of the country. The traffic volume in the 3^{rd} Generation mobile communication systems is increasing faster than new spectrum becomes allocated and usable. These trends require the introduction of new, scalable content distribution techniques.

The Multimedia Broadcast and Multicast Service (MBMS) defined by the 3^{rd} Generation Partnership Project (3GPP) for the Universal Mobile Telecommunications System (UMTS) is one way to address these issues. But so far, MBMS mostly has been discussed in combination with *Mobile TV* services. Mobile TV is associated with *linear TV* and scheduled broadcast distribution. This typically means to *re-broadcasting* a sub-set of the available TV channels also to mobile receivers. However, this type of services has so far not been accepted by end-users and not taken off. One possible reason could be that the concept of *linear* content reception, as used for TV, is not suitable for mobile end-users.

Consumer studies have shown that users utilize their phone for *killing time*, e.g. while waiting for the bus. Mobile users do not wait for the start of a program, like expected from linear TV users. Instead, the program or video clips should start when the user wants to start it.

This type of consumption fits more to the *video on-demand* concept, where the mobile end-user controls the start and progress of the content consumption.

The on-demand consumption does not necessarily mean to have an on-going communication session to a server. The content may also be side-loaded onto the mobile device via e.g. USB and served from the local file system.

The MBMS system defines features suitable for pushing content into mobile devices so that the content consumption is on-demand from the file system. A content provider can thus trigger pushing content items into the memory of mobile receivers. The content consumption may then happen on-demand similar to side-loaded content. In that sense, the consumption of the content is decoupled from the delivery of the content and different delivery optimization approaches become possible. We call this concept *MBMS Push File Delivery* in the following.

The main topic of this thesis is performance evaluations of different features of the MBMS specification. File reception in general must be 100% error free, since most file formats such as *.mp4* or *.jpg* are designed for error free environments. A set of tools are defined for MBMS Push File Delivery to handle transmission errors, namely variations of transmit power on the physical channel, Forward Error Correction on Application Layer (AL-FEC) and two mode of *Post Delivery File Repair*. Since MBMS terminals are not required to support simultaneous reception of MBMS broadcast traffic and unicast traffic on interactive channels, the file repair is typically executed after the data transfer on MBMS bearers is finished.

The system efficiency of a broadcast stream relies on the fact, that an arbitrary number of independent receivers are capable of receiving the stream. When a certain function is not supported by *all* broadcast receivers, the client-side function cannot be addressed by the network or multiple streams must be provided. Both situations are not beneficial for any broadcast system. Therefore, the amount of *optional* client side features should be limited as much as possible.

3GPP SA4 has selected the Raptor FEC as mandatory forward error correction for MBMS out of three candidate FEC codes. During the selection process, also LDPC and Reed-Solomon codes were proposed as MBMS FEC codes. Raptor codes provide a better decoding performance than Reed-Solomon codes. Raptor codes can handle longer source blocks (in terms of independent source symbols). Consequentially, the produced FEC symbols cover more source symbols. The decoding performance is very important for the efficiency of a mobile broadcast system.

Another advantage of Raptor FEC compared to Reed Solomon codes is that the decoding procedure requires a lower amount of operations per recovered source symbol. This behavior is very important for the battery efficiency of the FEC code. The reception of mobile broadcast content should not drain the battery too much.

A further general advantage of Raptor codes compared to LDPC and Reed-Solomon codes is the ability to produce an arbitrary high number of FEC symbols from the same source block. This simplifies the configuration of MBMS transmissions very much. A certain amount of FEC redundancy can even be pre-produced and stored with the source files and the service provider may still decide the amount of FEC redundancy at time of transmission.

Energy-efficient and reliable reconstruction of source data cannot be ensured by any FEC code. Mobile receivers may enter weak coverage spots like in elevators or in sub-terrain garages at any time even in densely populated and densely covered areas. Therefore the *post delivery file repair* is crucial to ensure a system-efficient and reliable file delivery procedure over mobile broadcast channels.

The performance and behavior of mobile push file delivery using MBMS UTRAN bearers is evaluated in the body of this thesis. The *system efficiency* is regarded as very important and is always subject for optimization. The used transmission power has a direct impact on the radio block error rate, thus also on the packet error rate. The Application Layer FEC is employed to handle the existing packet error rate and the needed amount of FEC redundancy depends on the packet error rate. Missing data after the MBMS PTM transmission is possible and should be handled by the post delivery file repair. It is not reasonable for an energy efficient system to handle all error with FEC only.

The MBMS service layer specification defines a set of error recovery schemes. One intention of this thesis is to clarify the usage and the behavior of the different schemes. There is always an error for quantitative performance evaluations due to the modeling of the system in a simulator. The error increases with the number of assumptions and simplifications. However, typically the behavior and the qualitative results of the evaluations are still applicable and provide very interesting insights in the configuration of the communication system.

1.2 Thesis Contributions

The author was deeply involved in the standardization of the MBMS Service Layer and the MBMS system. For MBMS, he started early with performance and scalability evaluations of the MBMS file repair. The Georgia Tech Network Simulator (GTNetS) was used as bases for file repair and the error propagation evaluations. Several new modules were developed which model different aspects of the system. A MATLAB radio network simulator is used to create traces of radio transmission block loss probabilities. The trace files are used as input for a GTNetS *MBMS link* model, which implements the fragmentation of IP packets into radio transmission blocks.

To get a better understanding of the Raptor FEC and its theory, the systematic Raptor FEC has been implemented in Python. It was used to determine the decoding failure probability depending on the numbers of source symbols.

A list of the authors publication is attached on Page 163.

In our paper **C1**, we reviewed the MBMS features for File Delivery, with particular focus on scalability and resource efficiency. MBMS is intended for data distribution to a large audience. We presented in collaboration with the co-authors the first time the idea of balancing amount of AL-FEC with the missing data for file repair considering the total Push File Delivery session duration (i.e. MBMS delivery plus post delivery file repair) as measure. Further we discussed and evaluated the concept of *Combined PTP and PTM file repair* in order to shorten the file repair duration and to cope with overloads. The paper introduced the measure of *used resources* in order to compare the system efficiency of different configurations.

In our paper **C2**, we extensively evaluated in collaboration with the other author the different factors influencing the MBMS system efficiency. Tuning the MBMS system for a certain target of *satisfied receivers* after the MBMS transfer does not lead to optimal usage of system resources. *Unsatisfied receivers* will use file repair in order to get the needed missing data to recover the file(s) and the file repair also influences the system efficiency. Therefore, we started to use the Missing Data after the MBMS data transfer as evaluation target.

We evaluated in **C2** the influence of key factors like the the IP packet size, the used transmission power and the amount of Application Layer FEC on the missing data after the MBMS transmission. We showed the dependency of the remaining missing data ratio on the energy. We also proposed balancing the used MBMS transmission resources with the needed resources for HSPA PTP File Repair.

In our publication **C3**, we evaluated the performance of the point-to-point (PTP) File Repair feature. The input into the evaluation is a distribution of missing data to be repaired by the PTP file repair. We used the *Sequential Delivery Time* of the missing data as an indication to configure the wait-time window of the PTP file repair.

In our publication **C4**, we evaluated the need for MBMS PTM bearers to distribute Mobile TV services. We showed that when considering the popularity distribution of the Mobile TV channels only a very low number of Broadcast Bearers are needed for an efficient system configuration. It is basically the same principle as when balancing the remaining missing data for file repair with the used resources during the MBMS data transfer.

The papers **C5** and **C6** contain a more general description of the MBMS feature. In **C6**, we describe in particular the advantages and use-cases for Push File Delivery using Mobile Broadcast bearers.

During very early stages of the thesis, the author was also working on the subject of Mobile Broadcast in several public funded research projects (e.g. **A7**), which evaluated the combination of digital broadcast such as DVB-T or DAB with cellular systems (e.g. **A10**, **A19**, **A20** and **A21**). The need for further research on the subject of mobile broadcast solutions was outlined in **A9** and **A15**. The particular focus was on the scalability issues to avoid overloads, optimizing the overall system efficiency (e.g. **A9**) .

1.3 Structure of the Thesis

The thesis is structured as following: We first introduce the UMTS system with its new MBMS feature in detail in Chapter 2. The intention is to provide sufficient background about the evaluated MBMS features as necessary. The reader finds there references to more detailed information about the evaluated system.

We then review extensively the relevant state of the art for reliable IP Multicast communication in Chapter 3. The use of IP Multicast for reliable communication has been subject for research for fixed IP networks for many years. However, the wide usage of IP Multicast has not yet taken off, since IP Multicast traffic is typically not routed between autonomous IP networks. However, within autonomous networks, it is quite often used. The most prominent example in these days is IPTV, where IP Multicast is used for linear TV distribution. The use of Application Layer FEC (AL-FEC) is also reviewed in Chapter 3 with a particular focus on Raptor FEC.

After the review of the relevant background, we evaluate the transmission capacity and the *goodput* of MBMS traffic channels. Already the radio protocols have an influence on the system efficiency. We evaluate in particular the error propagation from radio block losses to IP packet losses and its dependency on the IP packet size. After that, we evaluate the impact of the transmission power and the cell planning in Chapter 5. The lower the transmission power, the higher the radio block error rate. The remaining packet error rate can be decreased by AL-FEC, which increases the MBMS transmission time, thus also increases the system resource usage. The PTP file repair reduces the need for low packet error rates to some extent, since it allows recovering of missing data after the MBMS transmission. The used transmission power and the employed amount of application layer FEC must be balanced with the post delivery file repair in order to optimize the system resource usage for delivering the content to *all* receivers.

Chapter 6 and Chapter 7 evaluate the file repair procedure in detail. First, we start in Chapter 6 with the evaluation of the file repair method, which uses unicast (PTP) mobile radio bearers. Any mobile communication system and any server farm is only capable to handle a limited number of simultaneous unicast connections. Thus, we must carefully dimension the PTP file repair in order to allow a timely recovery of the data. Chapter 7 evaluates further the use of MBMS bearers for file repair.

Finally, we summarize the thesis in Chapter 8 and also give an outlook to further work.

2

3G Mobile Systems and MBMS

2.1 Introduction

The Universal Mobile Telecommunications System (UMTS) provides global mobility with a wide range of services including telephony, paging, messaging, public Internet access and broadband data [1][2]. Both connection-oriented and connection-less services are possible. UMTS is part of the IMT-2000 system family and defined by the 3^{rd} Generation Partnership Project. The specifications are endorsed by the participating organizational partners like the European Telecommunications Standards Institute (ETSI) as European Standard.

The 3^{rd} Generation Partnership Project (3GPP) is a collaboration agreement between various telecommunication standardization bodies that was create in December 1998 by ETSI SMG, T1P1, ARIB TTC and TTA in Copenhagen, Denmark. 3GPP produces globally applicable Technical Specifications (TS) and Technical Reports (TR) for 3^{rd} Generation Mobile Systems. The systems are based on evolved GSM and support the Universal Terrestrial Radio Access (UTRA) with Frequency Division Duplex (FDD) and / or Time Division Duplex (TDD) modes. 3GPP is also responsible for the development of Enhanced Data rates for GSM Evolution (EDGE) and the GSM/EDGE Radio Access Network (GERAN) standards.

The 3GPP Technical Specification Groups (TSG) prepare, maintain and approve Technical Specifications (TS) and Technical Reports (TR). The different groups are responsible for different parts of the system and different stages of the system design. Every 3GPP member may submit change request proposals to 3GPP specification in form of contributions. Those change requests are handled during one of the TSG meetings. The change requests approved, if there is consensus among the participating partners.

The 3GPP specifications are progressed in so-called *Releases.* Each release contains a set of feature definitions for all involved nodes and interfaces. All specification of a certain release are *functional frozen* after some time. Only *corrections* and *editorial improvements* are allowed after a release is functionally frozen. The procedure enables a step-wise development of the 3GPP features according to the release structure and to ensure a smooth deployment of new functions and features in the Live-Systems.

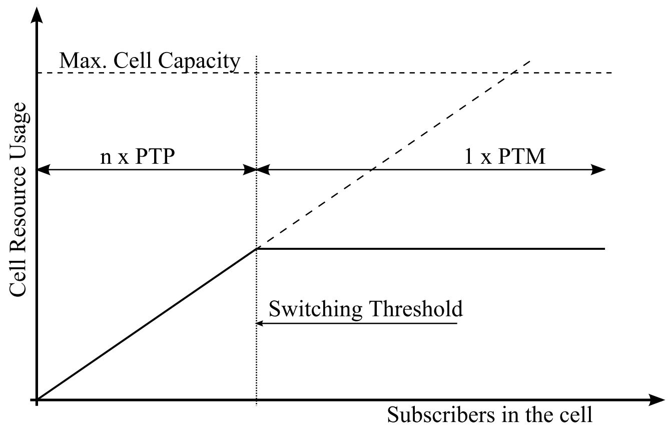


Figure 2.1: Radio bearer switching principle for of efficient group transmissions.

2.2 MBMS Principles and Architecture

This chapter gives an introduction to the Multimedia Broadcast and Multicast Services (MBMS) and its key features [3][4]. MBMS was defined first for 3GPP Release 6 and then evolved in later releases. MBMS efficiency and functionality is currently evolving with High Speed Packet Access evolution (HSPAe) and Long Term Evolution (LTE) projects of 3GPP. The chapter explains the principles around efficiency considerations, counting and the PTP/PTM switching. Further, it will introduce the overall MBMS architecture and the new MBMS specific functions.

2.2.1 Principles of efficient transmissions to groups

The key motivation for integrating multicast and broadcast extensions into mobile communication systems is to enable efficient group related data distribution services, especially on the radio interface level. The basic idea is to use radio broadcast transmissions (PTM) within geographical areas of high density of group members and to use unicast radio bearers (PTP) in areas of low user density.

Figure 2.1 schematically depicts the principle of an efficient transmission to a group. If there is no subscribed receiver in a cell, the radio network can decide to not transmit anything. If only a low number of receivers is located in a cell, the radio network may decide to use several PTP radio bearers, otherwise, the radio network serves using one PTM bearer.

The 3G radio network checks the interested receiver population for a particular service in each area and select the best suitable radio bearers, using the so-called *Counting Procedure*. The realization of PTP and PTM radio bearers depend on the access network and the used technology.

3GPP MBMS allows usage of *true broadcast* on the radio interface which allows addressing a transmission to multiple receivers. The MBMS Traffic Channel (MTCH) is used to realized the PTM radio bearer. The Radio Link Control (RLC) and Media Access Control (MAC) layers of mobile communication systems are designed for bi-directional communication and MBMS re-use certain functionality from the unicast communication.

MBMS realizes the PTP radio bearers either using HSPA or dedicated radio bearers. If several PTP radio bearers are used, the incoming IP packets are replicated by the RNCs to all PTP radio bearers. The PTP radio bearers may use acknowledgments on the radio to recover lost packets. The average bitrate is the same for all PTP radio bearers.

The difference between point-to-point and point-to-multipoint radio bearers is the power control and radio bearer loss recovery. UMTS PTP radio bearers use the fast power control to use the lowest possible transmit power. The UTRAN Radio bearers also use an acknowledged mode to recover losses on the radio bearer using retransmission techniques. The MTCH does not adapt the transmission power levels to individual needs. It is assumed that some receivers are located in rather bad radio conditions or at cell edges. The MTCH does also not implement a retransmission techniques, since UTRAN ARQ schemes are designed for unicast only. These facts generate research and development challenges with regards to efficient radio multicast deployments.

2.2.2 MBMS Architecture

The MBMS feature is split into the MBMS bearer service [3] and the MBMS user service [4]. The MBMS bearer service provides a new point-to-multipoint transmission bearer service, which may use common radio resources (i.e. broadcast) in areas of high receiver density. With 3GPP Release 6, the MBMS bearer service is introduced into both, the UMTS Terrestrial Radio Access Network (UTRAN) [5] and the GSM/EDGE Radio Access Network (GERAN) [6]. It offers a range of bit-rates for different services. The MBMS bearer service addresses MBMS transmission procedures below the IP layer, whereas the MBMS user services addresses service layer protocols and procedures.

The MBMS service layer [4],[7] is a toolbox, which includes a streaming and download delivery method. The MBMS security procedures offer service access protection for download and streaming traffic. The security procedures are bearer agnostic, thus do not differentiate between multicast and broadcast modes. It is possible to transmit protected content via MBMS, e.g. using OMA DRM.

Figure 2.2 shows which nodes of the 3GPP architecture are affected by MBMS. It also depicts the new Broadcast / Multicast - Service Centre (BM-SC) function, which is responsible for providing and delivering cellular broadcast services. The BM-SC serves as the entry point for content-delivery services that want to use MBMS. The BM-SC will be described in Section 2.2.2 in more detail. Towards the mobile core network it sets up and control MBMS transport bearers and it

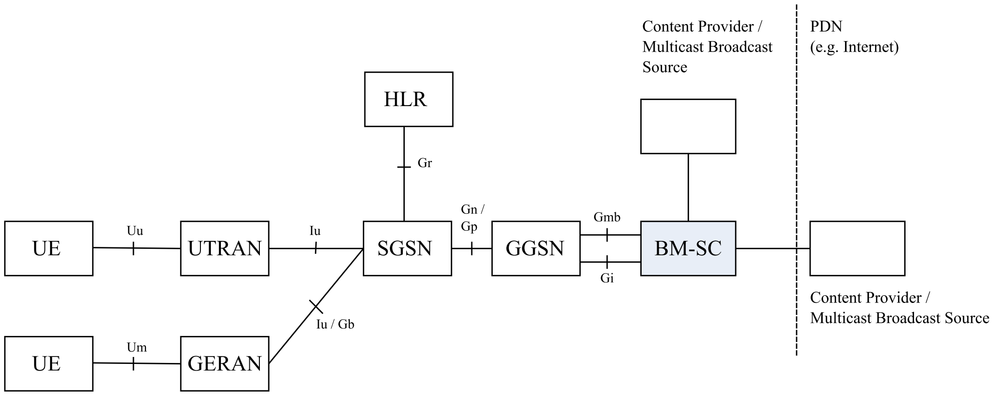


Figure 2.2: Reference architecture of network that supports MBMS.

can be used to schedule and deliver MBMS transmissions.

The Gateway GPRS Support Node (GGSN) is the ingress gateway towards the packet switched part of mobile networks (cf. [8]). It uses the GPRS Tunneling Protocol (GTP) [9] to transport the user level IP packets between the GGSN and the mobile terminals. The Serving GPRS Support Node (SGSN) handles the location and session management for the end-user. The user level IP address is not used for routing between the GGSN and the terminal. This allows to offer mobility services to the end-users.

The Gateway GPRS Support Node (GGSN) serves also as an entry point for IP Multicast traffic into the mobile network. The GGSN establishes a point-to-multipoint distribution tree within the mobile network upon notification from the BM-SC. The GGSN releases the MBMS bearer when triggered by the BM-SC. In case of the MBMS Multicast mode, the GGSN includes only those SGSNs into the distribution tree, which serve subscribers for that bearer. In case of the MBMS Broadcast Mode, the BM-SC provides a list of SGSNs, which shall be included in the distribution tree.

The GGSN receives IP Multicast traffic from BM-SC and forwards this data using the GTP. IP unicast is used to tunnel the MBMS traffic from the GGSN via the SGSN to the radio access network. The use of IP Multicast on transport level between the gateway (evolved GGSN) and the radio network nodes is first introduced with the Evolved Packet System (EPS) in 3GPP Release 9.

MBMS uses the IP Multicast framework. Only the protocol between the terminals and the first designated IP Multicast router is used. Multicast routing protocols such as PIM or MOSPF are not used inside of the Mobile Core network. The BM-SC may inject the multicast IP traffic using IPsec into the GGSN. A *Temporary Mobile Group Identifier (TMGI)* is used as a group identifier within the Mobile Network.

The radio networks (GERAN and UTRAN in 3GPP Release 6 networks) are responsible for efficiently delivering MBMS data to the designated MBMS

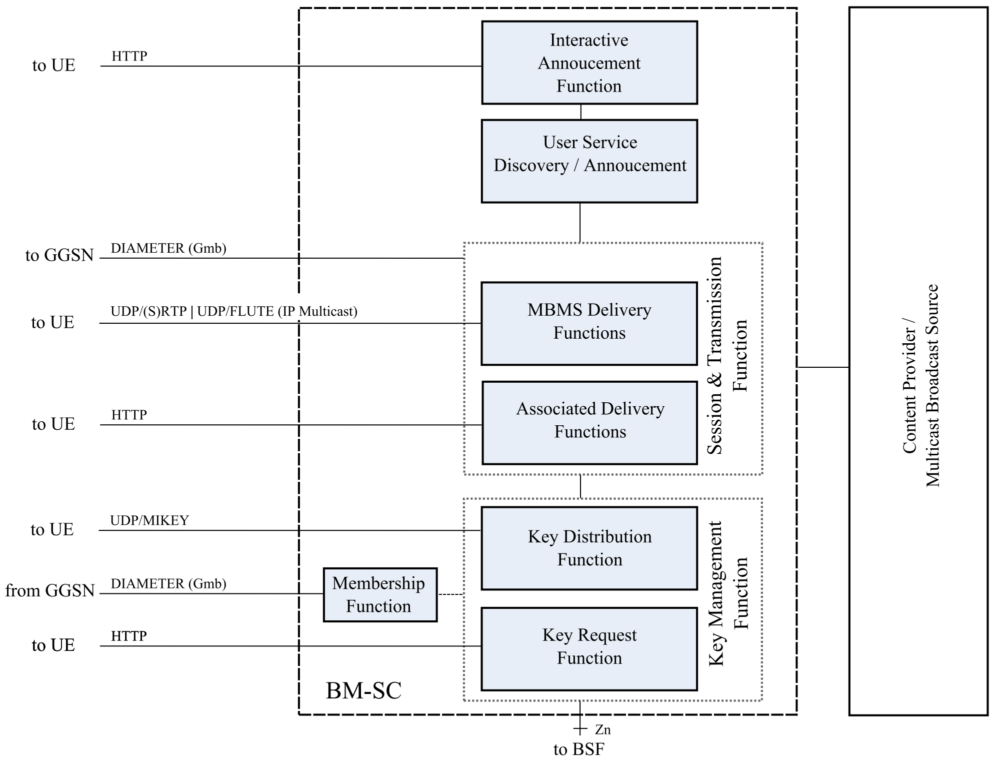


Figure 2.3: BM-SC sub-structure.

service area. MBMS in GSM/EDGE Radio Access Network (GERAN) may use up to 5 timeslots in downlink for a single MBMS Channel. Depending on the modulation scheme and the network dimensioning, a channel capacity between 32 kbps and 128 kbps can be achieved. The total cell capacity depends on the number of supported frequencies of that cell. MBMS in UMTS Terrestrial Radio Access Network (UTRAN) may use up to 256 kbps per FDD bearer. Several bearers may be active in a single UTRAN FDD carrier.

The Broadcast Multicast -Service Center (BM-SC)

The BM-SC is a new function in the MBMS reference architecture and offers a number of services for MBMS transmissions. The sub-functions of the BM-SC and their relations are depicted in Figure 2.3. It is not necessary to provide all BM-SC sub-functions via a single physical node. It is possible to host the Service Announcement, the Session and Transmission and the Key-Management functions on different physical nodes.

The **User Service Discovery / Announcement function** provides announcements of available services to end-devices. These announcements contain all necessary information, such as a service identifier, IP Multicast addresses and media descriptions which a terminal needs in order to join an MBMS service. The Service Announcement information is encoded in XML [10][11] and SDP [12] format, and is structured into information units, so-called fragments. MBMS supports several ways for delivering Service Announcements:

- Interactive retrieval of Service Announcements: HTTP interactions are used to fetch the XML service announcement fragment from the User Service Discovery / Announcement function
- Service Announcement using MBMS bearers: The Service Announcement fragments are distributed using the MBMS Download delivery method
- Push announcement using OMA PUSH [13]: the Service Announcement fragment is pushed using OTA-WSP [13] or OTA-HTTP [13] to the terminal. In the simplest way, that Service Announcement fragment may be encapsulated in one or more SMS messages.

The **Session and Transmission Function** includes all content transmission related functions. The function is further subdivided into the *MBMS Delivery Function* and the *Associated Delivery Function*. The MBMS Delivery Function basically includes the Content Sender, which either delivers files via the MBMS Download services (see Section 2.4.2) or streams via the MBMS Streaming service (see Section 2.4.1).

The Associated Delivery Function adds auxiliary procedures such as file repair or reception reporting. The purpose of the file repair Procedure is to assure error-free reception of files delivered over MBMS. File Repair Procedure will be explained in more detail in Section 2.4.3. The Reception Reporting Procedure allows the BM-SC to collect reception statistics.

The **Membership function** is only necessary for MBMS Multicast Bearers. It is used to authenticate the *join* request from a mobile receiver. The authentication request is send by the GGSN using the Gmb interface (see Figure 2.3) to the BM-SC. The membership function may optionally be connected to the key management function.

MBMS supports encryption of the files and streams. Encryption itself is done within the Transmission function. The MBMS **Key Management function** is used to provide authorized terminals (e.g. terminals subscribed to a particular MBMS service) with the necessary keys to decrypt the received files or streams.

The key management uses a Bootstrapping Server Function (BSF) to used to authenticate the user and to exchange shared keys, called MBMS User Key (MUK) between terminal and BM-SC. The keys are either stored in the UMTS Integrated Circuit Card (UICC) or in the terminal software. The UICC is understood as the hardware smartcard containing either a SIM or a USIM card. The BSF is part of the Generic Bootstrapping Architecture (GBA) [14], which is a general security framework for different services.

The Generic Bootstrapping Architecture (GBA) offers two variations to of cryptographic key storage. GBA_U defines the storage of the keys in the UICC and GBA_ME in the terminal. GBA_U is often regarded as more secure, since the keys are stored on the UICC. But since the keys are stored on the UICC, the GBA-U variant requires 3GPP Release 6 compliant UICCs. GBA_ME stores the keys in the terminal software and works also with legacy UICC cards.

The MBMS User Key is used to decrypt the MBMS Service Keys (MSK), which are transmitted individually per user in MIKEY (Multimedia Internet KEYing) messages [15]. MSKs can be either pushed, or requested from the terminal. The MBMS Service Keys are used to decrypt MBMS Traffic Keys (MTKs), which are also transmitted in-band with the actual MBMS data in MIKEY messages. The MTKs are finally the keys which protect the content streams and/or files.

2.2.3 Phases and procedures

The general phase of an MBMS services are depicted in Figure 2.4. The phases related to the MBMS user service are depicted on the left (receiver) and right (sender). Phases related to the MBMS bearer are depicted in the middle. Note, some of the MBMS bearer phases are identical for both: the broadcast mode and the Multicast mode. The differences between functions that are utilized for different modes are highlighted.

MBMS User Services use the MBMS Bearers in combination with unicast bearers. 3GPP Release 6 defined a content transmissions are centered using primarily the MBMS Bearer service for transmission. 3GPP Release 7 includes procedures to also use PSS or OMA Push for the actual content transmission.

Subscription (Only Bearer Service and Multicast Mode): Service subscription is intended to establish a relationship between the user and the service provider. It allows the user to receive the related MBMS multicast service. Service subscription is the basic agreement of a user to receive the service(s) offered by the operator. Subscription information is recorded in the BM-SC. The subscription phase is not further specified in 3GPP specifications.

Service Announcement: The service announcement phase describes the phase, during which the terminal acquires all needed information for a specific service. The MBMS specification allows service announcement over interactive, MBMS and also OMA-Push bearers. Service announcement fragments include all necessary information to activate the reception of a service. In particular IP Multicast address(es), ports, used codecs and codec configurations is included.

Service Initiation: The receiving device initiates the reception of a certain MBMS User Service. The MBMS User Service may use one or several MBMS Bearer Services. The receiver activates the reception of the corresponding bearer services. In case of Multicast Mode bearers, the receiver even triggers the "Joining" Bearer service phase. This phase corresponds to the process by which a terminal becomes a member of the multicast group, e.g., the user indicates to the network that he is willing to receive Multicast mode data of a specific MBMS bearer service.

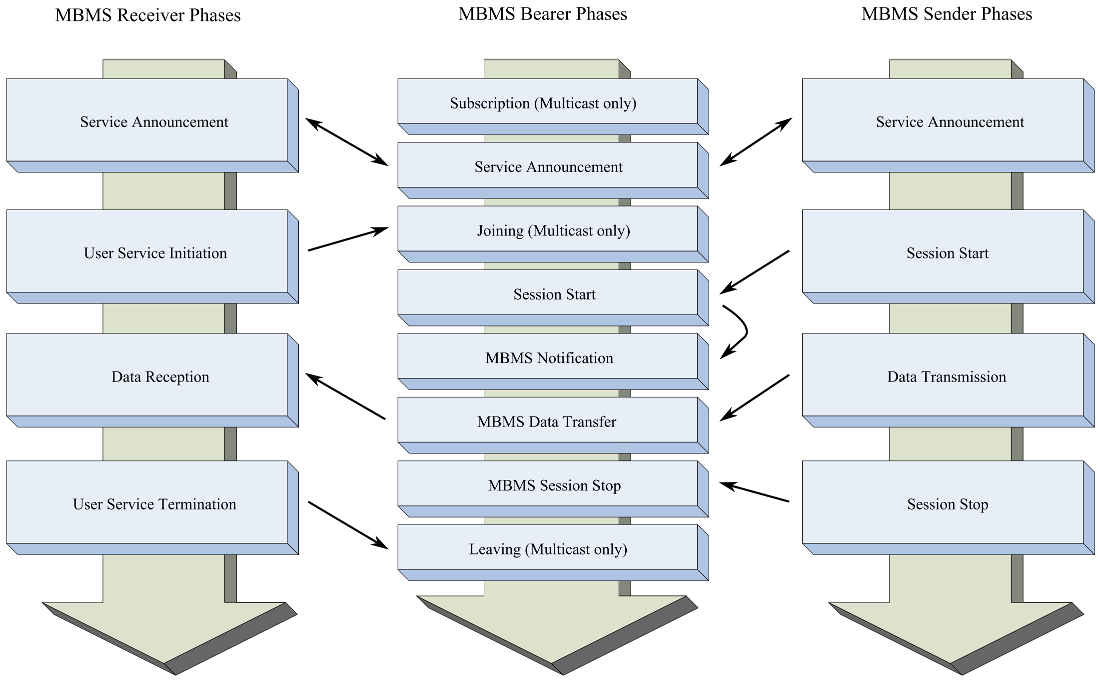


Figure 2.4: MBMS Phases Model for User Services and Bearer Services.

Session Start: Session Start is the indication by the MBMS Server (BM-SC) to the network to establish the MBMS user plane (putting the state of the MBMS bearer to active). Session Start occurs independently of initiation or termination of the service by the user - e.g. a given user may activate the service before or after Session Start. Session Start is the trigger for bearer resource establishment for MBMS data transfer.

MBMS Notification: MBMS Notification phase is used to inform the User Equipment about forthcoming and potentially about ongoing MBMS data transfers. The MBMS Notification is triggered by the Session Start phase. The radio network may determine the terminal density per-cell during the notification phase to select the radio bearers in a system efficient way.

Data Transfer: During the phase called Data Transfer MBMS data is transferring to the terminal.

Session Stop: Session Stop is the point at which the BM-SC determines that there will be no more data to send for some period of time - this period being long enough to justify removal of bearer resources associated with the session. At Session Stop, the bearer resources are released.

Service Termination: The receiver terminates the reception of a certain MBMS User Service. Since the MBMS User Service may use several MBMS Bearer Services, this may result in deactivating several MBMS Bearer Services. A bearer level leaving phase is triggered only when the MBMS Multicast Mode

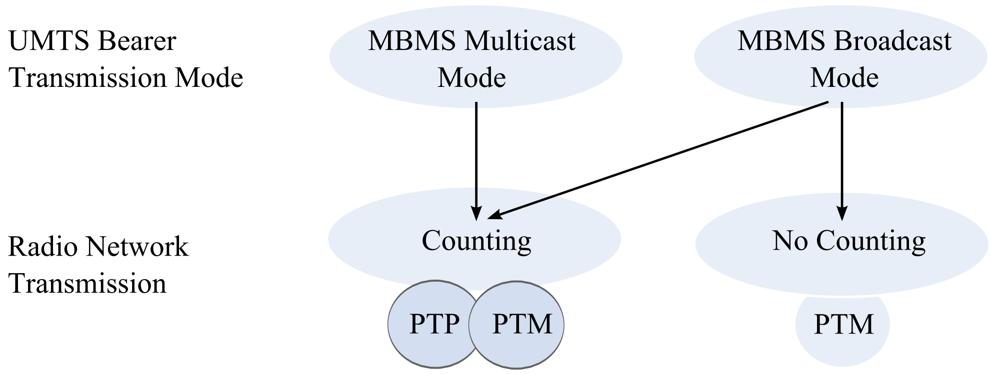


Figure 2.5: MBMS transmission modes and Radio Connection Types.

is used. The receiver leaves the multicast group.

2.3 MBMS Bearer Services

MBMS offers three different data transmission modes: broadcast mode, broadcast with counting (also called Enhanced Broadcast Mode) and Multicast mode[3]. The enhanced broadcast mode was added to Release 6 very late and intends to optimize MBMS for Mobile TV services. The relation of the three different modes is depicted in Figure 2.5.

The Multicast and the Enhanced Broadcast Mode allow for *counting* of the terminals in the given cell. The result of the counting procedure is used to set-up either one PTM or several PTP radio bearer for the data transmission.

MBMS transmissions are only offered within so-called *MBMS Service Areas*. An MBMS service area is associated with each MBMS Bearer. The MBMS Service Area is a group of radio cells, within the MBMS transmission is offered. The largest possible MBMS Service area is typically the entire PLMN. The smaller MBMS Service Area is a single cell. Also set of cell islands are allowed as an MBMS Service area definition. A single cell may belong to several MBMS Service Areas. Two example MBMS Service areas are depicted in Figure 2.6. The MBMS Service Area X consists of seven cells, the MBMS Service Area Y of four cells. The two *sub-areas* of MBMS Service Area X are not connected.

Figure 2.7 depicts an example MBMS bearer transmission. The content source is located in the BM-SC. The BM-SC sends an IP Multicast flow towards the mobile network. The GGSN receives the traffic and forwards it using GTP tunnels downstream to SGSNs. The MBMS Bearer Context includes a list of downstream SGSN addresses. IP unicast is used between GGSN and SGSN for forwarding the traffic on the transport IP level. The SGSN forwards the GTP traffic according to the MBMS Bearer context similar to the GGSN. IP Unicast is also used between SGSNs and RNCs.

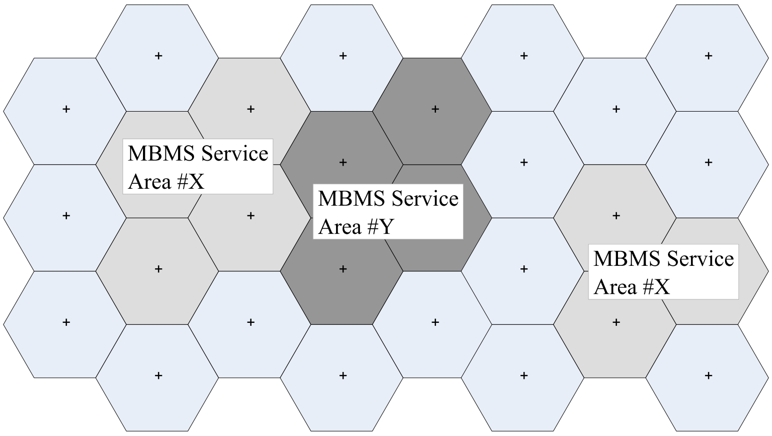


Figure 2.6: MBMS Service Areas.

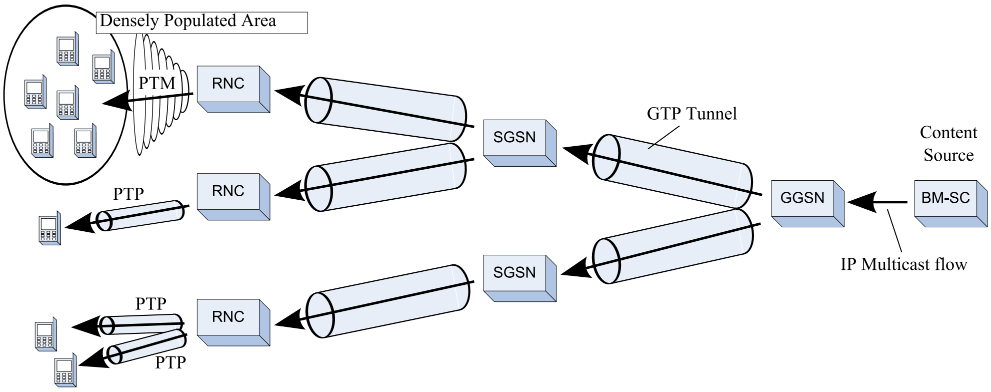


Figure 2.7: Example of an MBMS transmission.

2.3.1 Bearer Transmission Modes

The MBMS Multicast, enhanced Broadcast (i.e. Broadcast with Counting) and Broadcast Mode are introduced in the following section. The three mode differ in the way, how the MBMS bearer context is maintained. The MBMS bearer context basically determines a traffic distribution similar to an IP Multicast distribution tree.

Broadcast Mode

The MBMS Broadcast Mode offers a semi-static Point-to Multipoint distribution system. The BM-SC determines the broadcast area when activating the distribution bearers. The network has no information about active receivers in the Broadcast Area and cannot optimize any resource usage. Broadcast services may be received by all users who have activated the reception of the specific broadcast service locally on their UE and who are in the broadcast area defined for the service. The reception of the traffic in the broadcast mode is not guaranteed. The receiver may be able to recognize data loss. The MBMS Broadcast Mode is very similar to existing broadcast systems like DVB-T/DVB-H.

Enhanced Broadcast Mode

The Enhanced Broadcast Mode allows a more resource efficient delivery than the Broadcast Mode. Terminals indicate *service joining* (cf. Figure 2.4) up to the radio network. The radio network may perform the so-called *counting* or *re-counting* procedures to determine the number of terminals in each cell. It is more efficient to use dedicated channels or shared channels of HSDPA if only a low number of terminals shall be served in a cell, because the transmission on these channels can be tailored to the radio reception conditions of the respective terminals.

A point-to-multipoint MBMS Traffic Channel (MTCH) is only efficient if a higher number of terminals are located in a cell. The switching threshold between point-to-point and point-to-multipoint bearers depends on the terminal capabilities. The switching point is between 1 and 2 terminals per cell in case of soft-combining that means the combination, in the terminal, of radio signals received from several transmitters in adjacent cells, and between 5 and 10 otherwise.

Multicast Mode

The terminals indicate the *service joining* (cf. Figure 2.4) up to the core network when the Multicast mode is used. The network keeps state about service joining with the mobility management context in the Gateway GPRS Support Node (GGSN), Serving GPRS Support Node (SGSN) and Radio Network Controllers (RNCs). When the terminal moves from one area to another, the joining state for all activated services is also transferred to the new serving nodes. The RNCs may use the *counting* or *re-counting* procedures to determine the actual number of terminals in each cell. Thus, the network keeps track of each individual service member and can establish the distribution tree for the MBMS user plane very efficiently.

2.3.2 MBMS UTRAN Radio Bearers

MBMS Specific Channels

One design goal for MBMS in UTRAN was to reuse as much as possible of existing logical and physical channels. Three new logical channels and one physical channel was added to UTRAN specifications in the end. The new logical channels are MBMS Point-to-multipoint (PTM) control channel (MCCH), MBMS PTM traffic channel (MTCH) and MBMS PTM scheduling channel (MSCH). The new physical channel is the MBMS Notification Indicator Channel (MICH), which is used to efficiently notify the MBMS terminals about changes on the MCCH. As long as the information on the MCCH is not changed, the terminal does not need to receive the MCCH and can therefore shutdown the receiver to save battery power.

The MCCH provides access information about ongoing and upcoming MBMS transmissions for mobile receivers. In the first place, terminals understand from the MCCH, which MBMS bearers are or will become active in the cell. Each MBMS Bearer is identified by a TMGI. If the terminal is registered to one of the *announced* MBMS bearers, the terminal reads the MCCH to get access information to the MTCH. The MTCH carries the actual MBMS application data.

The MTCH can be configured with 40 ms or 80 ms transmission time interleaving depth (TTI). The selection of a longer TTI provides greater diversity in the time domain by spreading user data over the fading variations. This, in turn, yields improved MBMS capacity.

The MSCH provides information on the data scheduled on MTCH.

MCCH, MSCH and MTCH reuse the forward access channel (FACH) transport and secondary common control physical channel (S-CCPCH) in UTRAN. The RLC and MAC layer reuse much of the existing protocol stacks.

PTP transmission within MBMS reuses the existing UTRAN shared or dedicated channel types and will be discussed in the following subsection.

A Point-to-Multipoint (PTM) bearer does not employ feedback. Since the enhancements provided by HSDPA rely on feedback from the terminal, HSDPA cannot be used for PTM bearers. The PTM transmission parameters need to be statically configured to provide the desired coverage in a given cell. The transmitted signal is lowest at the cell border and therefore the PTM bearer can greatly benefit from exploiting also the signals from adjacent cells transmitting the same service, i.e. from soft-combining. For soft combining, it is required that transmissions are synchronized with an accuracy of 1 TTI plus 1 slot.

Capacity results have been compiled by 3GPP [16] for radio bearers of 64 kbps for the case without and with soft combining assuming the terminal has capabilities to combine 2 or 3 radio links. The network deployment assumptions are similar to those given in Table 2.1. The results are reproduced in 2.8, which shows the percentage of the total transmit power ($P_{max} = 20\,W$) of a base station that is required to achieve a certain coverage percentage. The important system assumptions are gathered in Table 2.1. The remaining parameters are listed in Tables 4.3.9 and 4.5.1 of TS 25.803 [16].

Soft combining with 3 radio links significantly reduces the transmit power requirement by 6.5 dB [16]. The system capacity can be further increased by receiver antenna diversity in the terminals. Simulations for a different channel model (so called *3GPP case 3*) have shown that terminals with antenna diversity require 3-5 dB lower signal to interference plus noise ratio. The required power scales approximately linearly with the bitrate up to the maximal rate of 256 kbps supported per bearer. The total cell capacity can therefore be used for a flexible combination 64, 128 and 256 kbps radio bearers.

Parameter	Value
MBMS data rate	64kbps
BLER target	1%
Transmission time interval (TTI)	80 ms
CPICH Ec/Ior	-10 dB (10%)
Power Control	Disabled
Channel Model	Vehicular A, 3km/h
Handover margin for no soft combining	0 dB
sectorisation	yes, 3 sectors/site
site to site distance	1000 m
NodeB antenna gain + cable loss	14 dBi

Table 2.1: Selected Link Level Assumptions.

Notification and Counting

The MBMS notification mechanism is used to inform receivers about an upcoming MBMS transmission for a specific MBMS Bearer. Any modification of the data of the MBMS Control Channel (MCCH) is indicated using the MBMS Indication Channel (MICH). Terminals monitor the MICH and MCCH channels for the associated TMGIs of the regsitered MBMS Services. The MCCH also provide access information to *tuning-in* to the according MBMS Traffic Channel (MTCH).

The MBMS Counting and Re-counting procedures are used to determine the density of registered MBMS terminals in a cell. PTM transmissions are only efficient, in case of high receiver populations per cell. If the terminal density is below a certain threshold, it is more efficient to replicate the traffic to one or more PTP radio bearers. Multiple point-to-point radio bearers with its fast power control schemes lead to a much better system utilization for small registered terminal populations. The counting concept is also visualized in Figure 2.1.

The RNC indicate the counting or re-counting procedure during the notification phase on the MCCH channel. Registered terminals send a counting response to the RNC. *Access probability factors* are used to avoid overload situations on the Random Access Channel (RACH), if too many receivers react on the counting procedure simultaneously.

2.3.3 Minimum Terminal Capability Requirements

The UTRAN specification [5] defines a set of *Minimum Terminal capabilities*, which must be at least fulfilled by *MBMS compliant terminals*. This is to ensure, that terminals offer the right set of capabilities to receive very basic MBMS transmissions.

The most important terminal capabilities are the capabilities for selective or soft combining of several radio links. This enables a terminals to receive the data

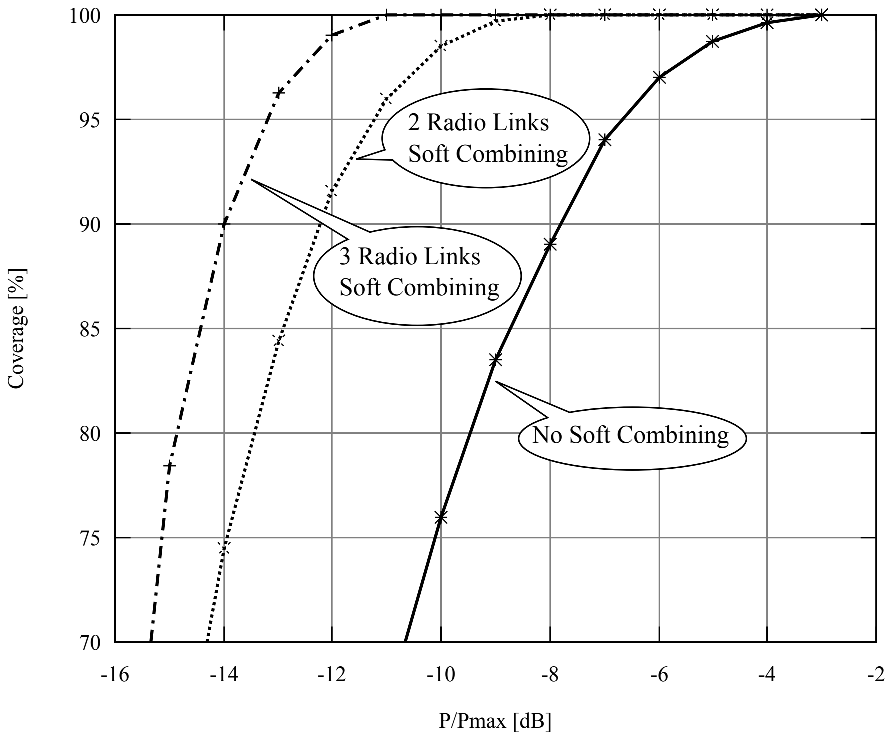


Figure 2.8: Estimated coverage vs. fraction of total Tx power with Soft Combining (64 kbps, 80ms TTI, STTD off, 1% BLER target) [16].

stream from more than one NodeB and to improve the reliability by combining the radio links.

The minimum terminal capabilities do not require the simultaneous usage of PTP radio bearers (i.e. Dedicated Packet or High-Speed Channels) and point-to-multipoint radio bearers (e.g. MBMS Traffic Channel).

There is also no dedicated requirement to support multiple different MBMS receptions at a time. It is expected, that first terminals may only receive MBMS data from a single MBMS bearer service.

2.4 MBMS Service Layer (User Services)

The MBMS user service framework [4], [7] defines the service layer for MBMS. Services like Mobile TV or Push File Delivery may use different components from the MBMS toolbox. The BM-SC (see Section 2.2.2) hosts all MBMS User Service Functions. 3GPP Release 6 offers two different delivery methods: MBMS Download and MBMS streaming. Both, MBMS download and MBMS Streaming may be FEC protected to increase transmission reliability. A set of associated delivery procedures are defined to support and enhanced the basic MBMS type of data transfer. Most associated delivery procedure use point-to-point UMTS

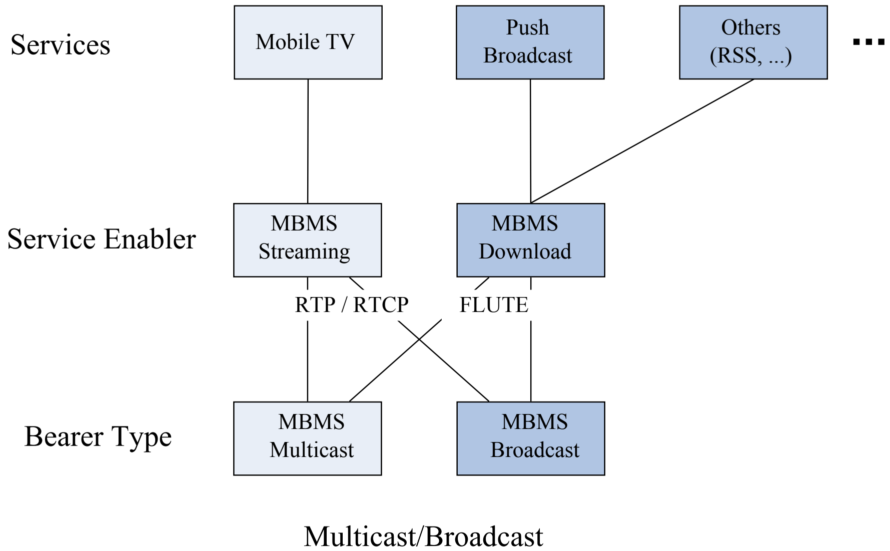


Figure 2.9: Delivery methods and transmission modes.

bearers with interactive traffic class (i.e. unicast bearers).

Note that all associated delivery procedure are *Post-Delivery Procedures*, thus are executed after the MBMS transmission has finished. This is mainly due to the Minimal Terminal Capability requirements (see Section 2.3.3. Terminals are not required to support PTP radio bearers simultaneously to PTM reception. This means, that a terminal with the minimum terminal capabilities cannot handle PTP file repair simultaneously with PTM reception.

The Real-Time Transport Protocol [17] is used for MBMS Streaming. The usage of the RTP protocol is aligned with Packet Switched Streaming [18]. Audio, Video and other elementary streams are not multiplexed into the same RTP session. Instead, the different session components are sent using different RTP sessions, thus using different UDP port pairs.

The IETF File Delivery over Unidirectional Transport (FLUTE) [19] protocol is used to delivery one or more files over MBMS bearers.

MBMS terminals are recommended to offer a set of multimedia codecs. Set of multimedia codes is aligned with the set for Packet Switched Streaming [18]. The most important codecs are the video and audio codecs. Here, H.264 [20] is recommended for video transmissions. Recommended codecs for audio transmissions are the Enhanced aacPlus [21][22][23] and the Extended AMR-WB [24][25][26] codecs.

MBMS Service Layer specification includes the definition of an *MBMS FEC* code. The Raptor FEC [4] code is mandatory for all MBMS terminals to implement. The usage of the MBMS FEC is optional. Whether or not the content is FEC protected is indicated either prior to the transmission. The MBMS FEC code is identical to the Raptor FEC code. The IETF version of the Raptor FEC code [27] was standardized some time after the 3GPP specifications. The MBMS FEC is described in Section 3.5.3 in more detail.

The Raptor FEC code is selected as the mandatory correction code first for MBMS Download and later also for MBMS Streaming (referenced as the MBMS FEC code). The Raptor FEC code is a low complexity code and allows the generation of a large number of independent FEC symbols from a single source block. Generally, Raptor codes can handle even large source block. But since mobile phones have a limited amount of fast memory for decoding, a single source block for 3GPP Release 6 receivers may only contain up to 4100 kByte of data [4]. Thus, larger data objects are subdivided into a number of source blocks and the FEC repair symbols are generated for each source block.

2.4.1 Streaming Delivery Method

The streaming delivery method is intended for the continuous reception and play-out of continuous media in Mobile TV applications. This delivery method complements the download delivery method which consists of the delivery of files. The streaming delivery method is particularly useful for multicast and broadcast of scheduled streaming content. Like digital video broadcasting, information like text and/or still images (static media) is also important can be added.

The Real-Time Transport Protocol (RTP) [17] on top of UDP is used for the MBMS streaming delivery method. RTCP sender reports are distributed mostly to provide synchronization between the different content flows. To avoid overloading the uplink, RTCP Receiver Report shall not be use. RTCP receiver report are disabled using the according SDP fields.

MBMS Streaming delivery defines also a framework for increasing transmission reliability. A FEC (shim)-layer may be added between UDP and RTP. The RTP header information are included in the FEC source block construction. The FEC framework allows *bundling* of several UDP flows for the FEC source block construction. The advantage of the FEC stream bundling concept, as shown in Figure 2.10 is that the FEC efficiency is increased when protecting several data flows together, because the FEC code works on a larger portion of data. Allowed flows are RTP, RTCP and MIKEY [15].

The FEC encoder adds a Source Block identifier and an Encoding Symbols identifier to the end of the UDP packets. This is mostly to re-use RoHC header compression profiles [28]. One important profile is the *RTP profile*, which allows the efficient compression of IP/UDP/RTP flows.

Figure 2.10 depicts the principle of the FEC stream bundling. The FEC stream bundling is about protecting a set of media flows jointly to increase the FEC efficiency. It becomes for instance possible to protect the audio and video flows of a single Mobile TV channel together. Between four and six UDP flows are needed of one TV channel. These are one RTP and RTCP flow for each video and audio components. Optionally the MIKEY key flows (separate flows for audio and video) may be added in to the FEC bundle. Note, if the MBMS traffic keys,

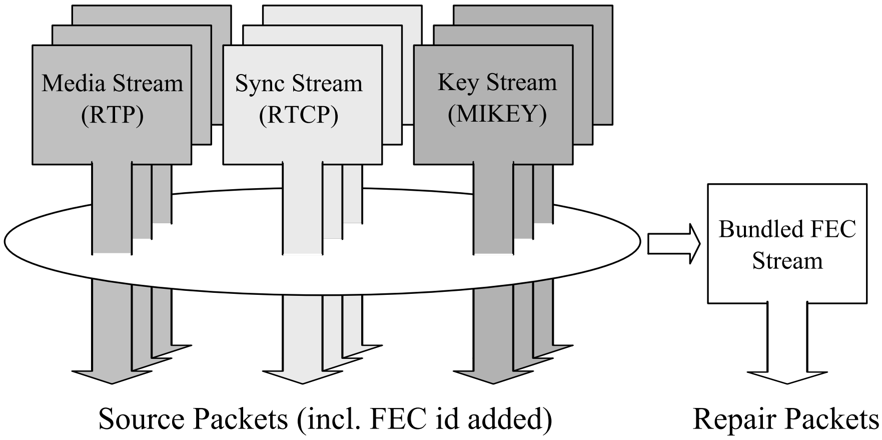


Figure 2.10: FEC stream bundling concept.

which are interleaved with the other data traffic on the MBMS bearer, get lost then the receiver cannot decrypt and render the actual content stream.

2.4.2 Download Delivery Method

MBMS download can be used to deliver an arbitrary number of files from a single source to many receivers. Existing content-to-person MMS services [29], for instance a service, which delivers short video clips of a sports event via MMS, will greatly benefit from this feature. Today, those services use point-to-point connections for MMS delivery. An SMS is sent to the client to trigger a PDP context activation and the retrieval of the actual content using HTTP [30] or WSP ref. In the future the existing MMS sub-system can be interfaced with a BM-SC which then distributes the clip via MBMS download.

MBMS Download uses the FLUTE [19] (File Transport over Unidirectional Transport) protocol for file delivery. FLUTE was designed for massive file delivery over unidirectional links such as for digital broadcasts. Since HTTP [30] and TCP [31] are not feasible for point-to-multipoint communication, a newly developed protocol is used. FLUTE is described in Section 3.4 in more detail.

An example sequence of the MBMS download procedure is shown in Figure 2.11. The Broadcast/Multicast-Service Centre (BM-SC) establishes the MBMS bearer using the MBMS session start procedure. This procedure triggers a new *group* notification procedure which wakes-up all MBMS group members. Section 2.3.2 includes a description of the MCCH channel and the notification. The BM-SC should leave the Radio and Core Network between 20 sec and 40 sec for the notification and MBMS Bearer establishment. After the MBMS bearer is successfully established, the BM-SC (MBMS Sender) starts sending the actual MBMS download data. FLUTE is used to send the files via UDP. The BM-SC releases the MBMS bearer after all files of the MBMS transmission including the

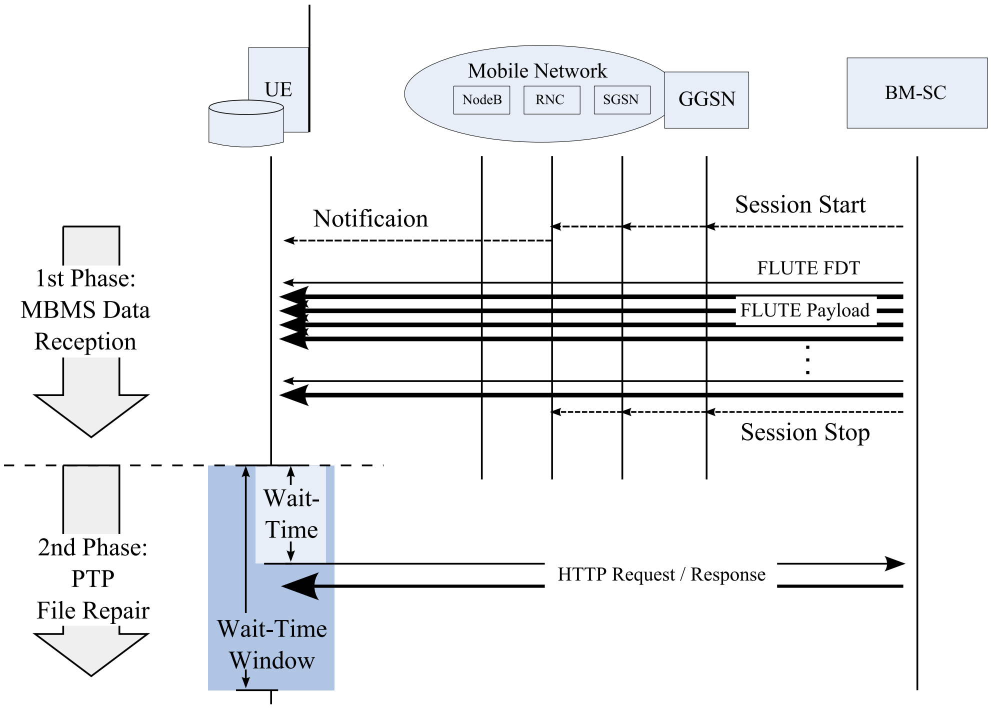


Figure 2.11: MBMS Push File Delivery Principle.

FEC overhead is transmitted.

The *Associated Delivery Procedures (ADP)* are optionally executed after the MBMS data transfer phase. The definition of the associated delivery procedure(s) may be provided or updated during the MBMS data transfer in a so-called Associated Delivery Procedure Description fragment. In such a case, an associated delivery procedure description fragment is transmitted as an own object in the FLUTE session.

Two types of associated delivery procedures are defined by the MBMS Service Layer [4]: The file repair procedure ensures the reliability of transmissions and is described in Section 2.4.3. The reception reporting procedure is used to collect reception statistics and is described in Section 2.4.4.

Figure 2.11 depicts the point-to-point File Repair procedures as an example associated delivery procedure. A terminal may request the repair of missing data from the file repair server of the BM-SC.

MBMS User Services can use MBMS bearer transmissions only or use interactive bearers and MBMS bearers in combination. Particularly in the case of combining MBMS and interactive bearers, the interactive bearer usage must be protected against overload situations.

2.4.3 Reliability Methods for MBMS Download

FLUTE enables file delivery on unidirectional MBMS bearers and improves reliability with FEC technology. However, it cannot guarantee 100% reliability for file delivery. If for example a receiving terminal gets into bad MBMS coverage, there is no way to guarantee a successful reception via MBMS. In order to complement the forward error correction scheme, the file repair mechanism is added to MBMS ([4]) as another mechanism to increase transmission reliability. Terminals that are not able to reconstruct a MBMS data delivered in FLUTE session during the MBMS data transfer phase continue with this file repair procedure if the file repair procedure was defined by the network. The File repair procedure could be regarded as "Backward Error Correction" or "Automatic Repeat Request" (ARQ) scheme.

Three packet error recovery schemes are defined for MBMS download. One important one is the use of *packet level FEC coding* or an Application Level FEC code (AL-FEC). The Forward Error Correction (FEC) allows recovering lost packets without any server interaction. FEC schemes are in particular needed for scalability of reliable Point-to-Multipoint distributions.

The Raptor FEC code was selected in 3GPP as the single MBMS FEC code. Raptor is mandatory to be implemented on the client side, so that the Operator is able to add FEC redundancy when needed or regarded appropriate. An evaluation of the Raptor FEC is given in Section 3.5.

However, even the best FEC code cannot guarantee 100% reliability of file delivery, if the transmission has a deterministic end due to resource savings. It may simply happen that a terminal leaves the coverage and enters again after the transmission has ended.

Therefore, two other type error recovery methods are defined and optionally used after the MBMS transmission has ended. With the *Point-to-Point (PTP) File Repair* method, an unsatisfied receiver can fetch missing data using HTTP. The *Point-to-Multipoint (PTM) File Repair* method allows the BM-SC to send further MBMS data after the actual MBMS data transfer. The MBMS Service Layer standard does not define any combinations or sequence of file repair methods. It does also not define, whether source or repair (FEC) symbols should be used during file repair, although it seems natural to send repair symbols.

The PTP File Repair procedure offers an overload protection scheme. The file repair load may be spread in time and across different servers. The sender (i.e. BM-SC) defines a *Wait-Time Window* in an associated delivery procedure description fragment, which is sent in-band with the actual file delivery session. A terminal, which needs to use the PTP File Repair service draws first a random "Wait-Time" out of the *Wait-Time Window* and defers the actual File Repair operation by this time. The *Wait-Time* and the *Wait-Time Window* are also depicted in Figure 2.11.

If the BM-SC lists a set of PTP File Repair servers in the Associated Delivery Procedure Description fragment, then the PTP File Repair load is even spread across several network elements. Terminals, which want to use the PTP File Repair service randomly select on server out of the list of servers. If one server is busy or not reachable, then the terminal can choose the next server and re-try. In order to protect also the DNS system against overload (and to save even this transaction), the PTP File Repair servers may be described with IP Addresses. Note that the BM-SC may change the List of PTP File Repair server with every transmission. Thus the use of IP addresses does not lead to any loss of flexibility.

Generally, the MBMS receiver waits until the end of the transmission and, if file repair is defined as associated delivery procedure, then identifies the missing data from the MBMS download session. In case of the PTP file repair procedure, the terminal describes the missing data as query arguments in the HTTP `GET` method [30].

```
GET /path/repair.cgi?fileURI=www.example.com/latest.3gp&
    SBN=2;ESI=12&SBN=5;ESI=25-27 HTTP/1.1
Host: bm-sc.example.com
...
```

This example PTP File Repair request line informs the BM-SC, that one Encoding Symbol (ESI) from a first Source Block (SBN) and a range of encoding symbols from a second source block. It is then up to the BM-SC to decide, whether to send the actual source symbols or additional repair symbols (only if FEC is in use for the file). Since the PTP File Repair uses individual HTTP connections, the repair data is independently sent to different receivers and can be tailored to the actual losses of that receiver.

The BM-SC may redirect to a PTM File Repair session depending on various reasons. The PTM File Repair uses the same or a different MBMS Bearer to send the File Repair data. If the MBMS bearer is used, then the File Repair data should be beneficial to as many receivers as possible. Due to the *rateless* property of the Raptor FEC code, it is possible to generate a large number of FEC repair symbols from a single source block and to use them during the PTM File Repair procedure.

The point-to-multipoint File Repair procedure uses an MBMS Bearer to send additional data to terminals. The BM-SC may use the same or a different MBMS bearer for the file repair data. Note, if the BM-SC re-uses the already established MBMS bearer, it may use the MBMS Session identity concept to improve efficiency.

The 3GPP user service specification does not provide any reference configurations for the file repair procedure. Several realization combinations of PTP and PTM file repair are possible are possible.

2.4.4 Reception Reporting Procedures

The reception reporting procedure gives the BM-SC the opportunity to request reception feedback from the receivers. The BM-SC may ask for *reception ack-*

nowledges or for *statistical reporting*.

The reception reporting procedure is a part of the associated delivery procedure. Since the reception reporting procedure uses HTTP to give feedback, it is also protected against overload situations. The reporting load may be spread in time and across network elements like the PTP File Repair procedure (see 2.4.3).

2.5 Hybrid Unicast / Multicast Integration

MBMS is introduced as a *multicast/broadcast* system extension to the 3^{rd} generation mobile network. One prime idea is to enable TV-like broadcast transmissions to large receiver populations. MBMS itself does not define any uplink channel. It re-uses the already defined 3G bearers and assumes, that those bearers are available in mobile terminals. Services may be offered by combining a Multicast/Broadcast downlink with an Interactive channel.

In that sense, MBMS is very comparable to DVB-T or DAB based hybrid systems, except that it is runs in spectrum allocated for mobile services. Keller et al. [32] and Walsh et al. [33] discuss the hybrid combination of Digital Broadcast with 3G systems. One aspect was the combination of the positive properties of the available systems: Broadcast Systems are designed to disseminate data, whereas mobile communication systems are designed to person-to-person communication.

The strength of broadcast systems is the efficient dissemination of content to an unlimited number of receivers in a certain area [34]. The efficiency increases with increasing receiver population size. The weakness of broadcast systems is the lack of interactivity. The system is not aware of the actual number of listening receivers. If the receiver population decreases, also the efficiency is decreasing. No receiver is present in the worse case.

Also Acharya et al. [35] discuss the balancing of "Push" and "Pull" paradigms. The approach is to disseminate web-pages using *Broadcast Disks*, i.e. Web-Pages are periodically broadcasted. To reduce the duration of the broadcast cycle and to save broadcast bandwidth, pages with a limited popularity are not broadcasted. Those pages are requested via the pull paradigm.

The strength of 3G mobile communication systems is the high flexibility of using the available spectrum. 3G is designed to carry unicast voice or data services. The disadvantage of 3G systems is the lack of support for broadcast transmission schemes. The system efficiency decreases, if a high number of terminals receive the same content. In a worst case Mobile TV scenario, the entire cell capacity is consumed by users, watching the same Mobile TV channel.

Since the usage of *broadcast* is only efficient in case of popular content, it seems natural to combine *broadcast* with *unicast* transmission capabilities. The popular content is offered using *broadcast* transmission bearers and the *long-tail* content is offered on-demand using unicast bearers. The terms *popular* and *long-tail* are related to the popularity of a content item. Content items, which are frequently accessed by users are *popular* whereas very unpopular content belong

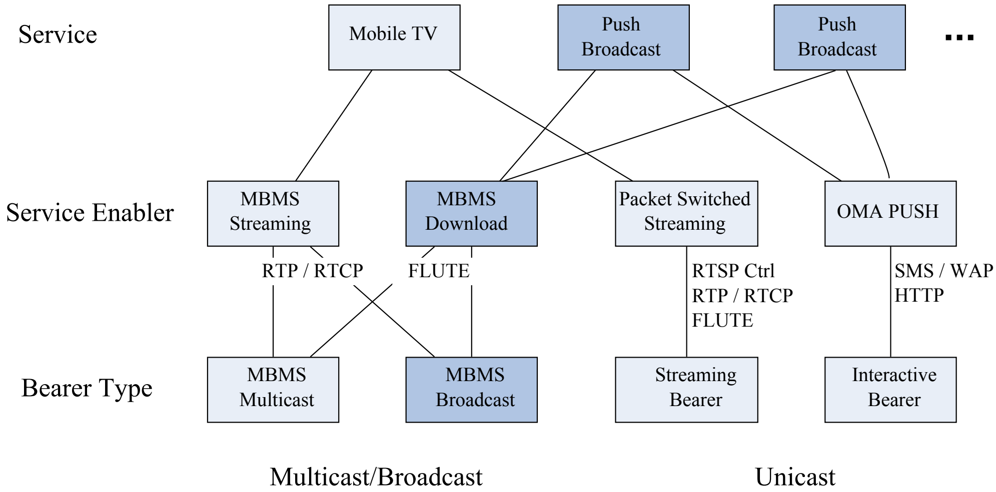


Figure 2.12: Delivery methods and transmission modes.

into the *long-tail* sections the popularity distribution. Since the long-tail content is not consumed frequently, it is more efficient to allocate transmission resources only if necessary.

Such a combination is presented and evaluated by Lohmar et al. [36] and partly also by Hartung et al. [37]. It evaluates the system efficiency of a Mobile TV service, where the popular channels are broadcasted and the long-tail channels offered via unicast.

Note the end-user is typically only interested to consume the service and is not interested in the underlying technology. IP allows a flexible allocation of services to unicast or broadcast transmission schemes. Only network operators are interested in the efficient usage of their resources.

With the 3GPP Release 7 versions of the specification it becomes possible to offer the services also using *normal* unicast bearer services. Figure 2.12 depicts the new combination possibilities. This is of particular interest to offer PLMN wide services, independently from any MBMS Service Area. Broadcast transmission is typically only efficient in densely populated areas. Unicast bearers are usable outside of those MBMS Service Areas or even in roaming situations.

The RTP flows for the multimedia components can be established using Real-Time Streaming Control Protocol (RTSP) [38] in case of unicast delivery according to the 3GPP PSS specification [18]. The FLUTE unicast session may also be established with RTSP, in particular when used in combination with streaming sessions. But the MBMS download method may also use OMA Push [13]. Since OMA Push may use SMS to notify a terminal about newly available files, it is in particular interest of MBMS Download-only service offerings.

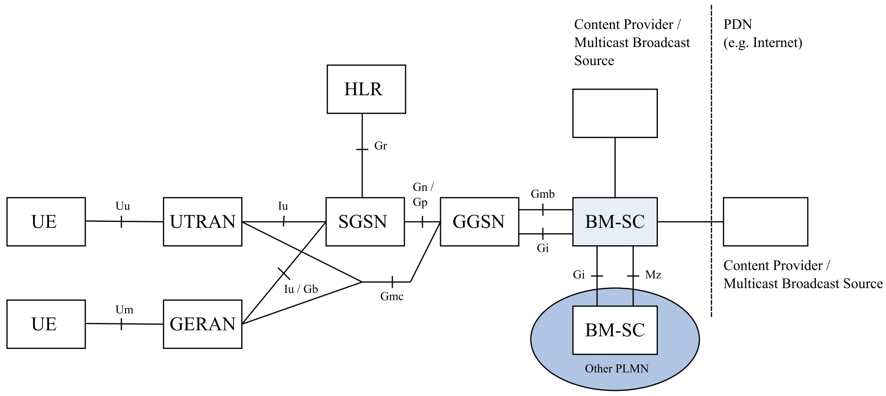


Figure 2.13: MBMS Release 7 Architecture.

2.6 Evolution of MBMS in 3GPP

MBMS evolved already with 3GPP Release 7. The MBMS specifications were updated with several architectural improvements and with a MBMS Single Frequency Network (MBSFN) support mode for UTRAN radio networks. The improved Release 7 architecture is depicted in Figure 2.13. One major architectural improvement compared to Release 6 is the introduction of a *direct tunnel* between the GGSN and the radio network nodes. This direct tunnel allows the GGSN to directly send the MBMS user plan traffic to the RNC or BSC in the network, bypassing the SGSN. It still uses IP unicast on the lower layers. The SGSN stays in the control plane path between GGSN and radio network.

Another new feature is the introduction of a new roaming interface between a BM-SC in a home network and a BM-SC in a foreign network. The new Mz interface is basically the roaming variant of the Gmb interface. It allows MBMS specific user signaling like service authorization, when UEs roam in foreign networks.

A very interesting radio transmission mode introduced with 3GPP Release 8 is called Integrated Mobile Broadcast (IMB). This radio network feature uses the unpaired TDD spectrum bands, thus reliefs the FDD spectrum from multimedia traffic. IMB is re-uses techniques and procedures from existing FDD UTRAN, but also allowing deployment in TDD spectrum. To date the TDD spectrum is widely unused by operators.

The TDD band is subdivided into multiple 5 MHz carriers. Each of them can be used solely for Mobile Broadcast services. Multiple carriers can be aggregated to increase the transmission capacity. GSMA expects [39] that around 20 broadcast channels (MTCH) with 256 kbps per channel can be provided from using 5 MHz of unpaired TDD spectrum.

The MBMS architecture evolved even further in 3GPP Release 9 as part of

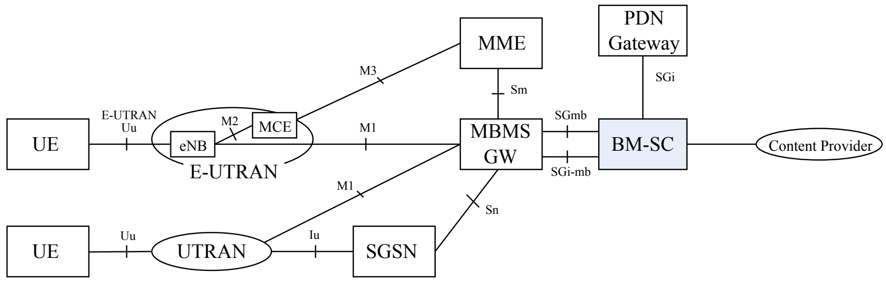


Figure 2.14: MBMS Architecture in Evolved Packet System (EPS).

the Evolved Packet System (EPS) as depicted in Figure 2.14. One key difference of the evolved MBMS architecture is the introduction of IP Multicast on the lower IP layer between MBMS Gateway (MBMS GW) and all the eNBs. The new M1 Interface uses IP Multicast to forward the IP Multicast user plane traffic from the MBMS Gateway to all eNBs and RNCs (IP Multicast-in-IP Multicast encapsulation with some tunneling layer in between). Due to the high number of eNBs, it was important to use efficient delivery techniques. The MBMS-Gateway separates the core-network addressing domain from the service layer domain. This separation is in particular important, when the traffic come from an external provider. The MBMS Gateway may implement monitoring and also policy enforcement functions.

The BM-SC functionality remains the same in EPS. The procedure on the SGmb interface are in principle the same as on the Gmb interface. The MBMS GW function separate the core network domain of the operator from the service layer.

For MBMS in LTE, the RAN group has defined a new Multi-Cell / Multicast Coordination Entity (MCE) in the control plane path, which controls the resources when using MBMS Single Frequency Network (MBSFN) operations. There is a very strong emphasis on Single Frequency Network (SFN) support for MBMS in LTE due to the higher spectral efficiency when transmitting to larger areas.

3

Related work from Reliable IP transmissions

This chapter first provides a very brief introduction into the Transmission Control Protocols (TCP) and the Hyper Text Transport Protocols (HTTP) as most relevant protocols for any PTP communication. It gives the necessary background for the point-to-point file repair (see Section 2.4.3) and the dimensioning on the mobile network nodes.

Afterwards, the chapter contains a review of reliable multicast protocols from the literature. We present different reliable multicast protocols, which are developed by the IETF RMT working group. We extensively review the concept of Application Layer Forward Error Correction as one way to increase reliability of IP Multicast distribution. In particular the LDPC and the Raptor FEC codes are evaluated in detail.

Finally, we discuss the relevance of the existing work from the Internet for mobile broadcast service offerings.

3.1 TCP and HTTP in a Nutshell

This section contains a very brief summary about the important features of TCP and HTTP in order to understand the differences to reliable multicast. Both protocols are widely spread in today's Internet for data consumption.

The Hypertext Transfer Protocol (HTTP) [30] is used to fetch data objects from the Internet. Today, it is commonly used in the Internet to fetch web-pages, but also other information objects such as RSS feeds.

Although HTTP is called a "Hypertext Transfer Protocol", it is not limited to just "hypertext". Any type of binary data may be transported using HTTP. The HTTP header field contains information about the type of the included object. It may be a simple plain text object or any other binary data object. It is also possible to add multiple objects into a single HTTP message.

The HTTP protocol itself is a stateless, transaction oriented protocol. A transaction is a single request-response sequence. An HTTP request message is sent from the communication client to the server. The server responds with an HTTP status message. The structure of HTTP messages is depicted in Figure 3.1. Both, request and response messages are constructed in the same way. The first line of the message determines whether the message is a request or response

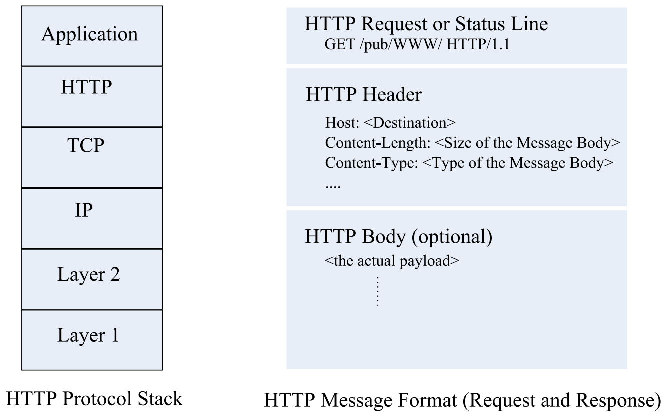


Figure 3.1: HTTP protocols stack and message construction.

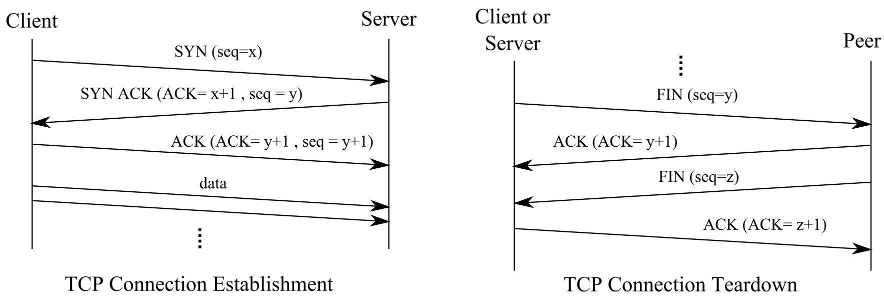


Figure 3.2: TCP handshakes for Connection establishment and Teardown.

message. Both message types contain an HTTP header section and optionally an HTTP body section. One or several data objects such as web pages may be contained in an HTTP message.

There is no session related information, which associates a first transaction with a later transaction. The communication connections between the peers may be released after the transaction. However, this also means that the transport, e.g. TCP session, must be established again, to retrieve further objects from the peer, e.g. if a retrieved web-object contains embedded images.

The Transmission Control Protocol (TCP) [31] is a well-known protocol, which is widely used in today's Internet as a transport protocol for HTTP or other application level protocols such as email. TCP offers a reliable, connection oriented communication service between the network interfaces of two nodes.

TCP offers a reliable, byte-stream oriented transport service between the two communication peers. The data is send in TCP segments from one communication peer to the other. A TCP segment consists of the TCP header and some payload. TCP segment size is smaller or equal to the maximal transfer unit (MTU) of IP, which is typically determined by the layer 2 transmission capabilities.

The TCP connection is established from a client to a server using a three-way handshake (cf. Figure 3.2). To establish a TCP connection, the client sends a TCP segment with a SYN-flag (== Synchronize) set. The server responds with a TCP segment, where the SYN and the ACK flags (== Acknowledge) are set. Once established, the connection can be used in both directions. Clients and servers can tear down the TCP connection, not necessarily in both directions. Even if one peer has torn down the TCP connection, the other peer may still send data. The TCP connection is in such cases "half-closed".

The performance of the TCP connection heavily depends on the intermediate network nodes and on the queue sizes. In particular wireless links have a higher transfer delay. The queues and the queuing principle should be dimensioned according to the "pipe-capacity" [40]. Sågfors *et al.* [40] give a good overview about the issues with TCP in 3G environments and guidelines to dimension the network queues.

3.2 IP Multicast Basics

Group Communication and Point-to-Multipoint communication is a field of research for a long time now. IP Multicast was tested for a long time in the Multicast Backbone (MBONE) [41]. The MBONE was a multicast capable overlay network, which tunneled the Multicast traffic between different mostly research institutions. IP Multicast is until now to commercially available on an end-to-end basis. Different Internet Service Providers (ISP) do not route the multicast traffic.

IP Multicast uses so-called group addresses. An application can register the group address with the local IP stack, which then starts fetching Multicast IP packets from this group from the local network. The Internet Assigned Numbers Authority (IANA) has allocated the address range between 224.0.0.0 and 239.255.255.255 to multicast communication [42]. This range was formerly called "Class D".

The sender transmits a single datagram (from the sender's unicast address) to the multicast address, and the multicast routers take care of replicating the packets and sending them to all receivers that have registered their interest in data from that sender. This is also illustrated in Figure 3.3.

Figure 3.3 depicts an example topology with different Multicast protocols. There are two different protocol categories involved: The first type of protocols is used between *hosts* and *Designated Multicast Routers*. The Internet Group Management Protocol (IGMP) [43][44][45] provides the methods for a host to join a multicast group and for the designated multicast routers to maintain multicast groups. The same function for IPv6 is provided by the Multicast Listener Discovery (MLD) [46][47] protocol.

The second category of protocols comprises the multicast routing protocols. The main task for the multicast routing protocol is to establish and maintain

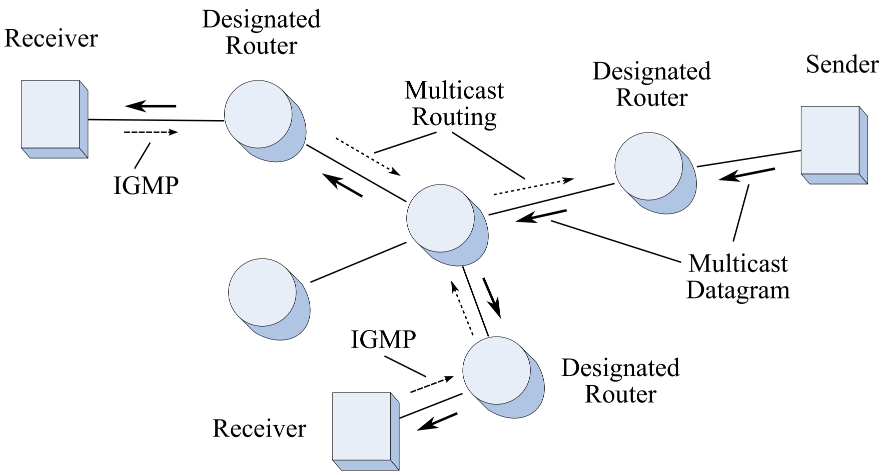


Figure 3.3: Multicast Protocols.

the *multicast distribution tree* in the system. The packets are forwarded and replicated according to the distribution tree. Multicast receivers are the leaves of the tree. A number of different Multicast routing protocols exists. The most important ones are PIM-SM [48], CBR [49][50][51] and MOSPF [52].

3.3 Categorization of File Transfer Protocols

This section gives an overview about different reliable multicast protocols. There are different use-cases for IP Multicast. A large research community focuses on Multimedia Content Distribution over the public Internet. A number of research papers were published in the late '90s, which propose and evaluate different techniques. One prime design goal of IP Multicast was the scalability, since it was intended for group communication services. Today, it is quite often used for data dissemination services: A single server sends out files or stream and a large population of hosts receive the content.

The goal of reliable multicast transmission research is to distribute content to a very large receiver population. IP Multicast provides a good transmission vehicle to disseminate IP packets efficiently. But IP offers only an unreliable, best effort transmission service, thus a transport layer protocol is necessary to ensure reliability. In the following, we focus only on the file delivery protocols. Stream delivery e.g. for TV transmissions or multiparty conferencing systems are also investigated in the literature, but not considered here.

One requirement of reliable multicast protocols is the scalability towards large receiver populations. IP Multicast offers an efficient and scalable packet replication scheme and the transport layer should not limit the efficiency too much.

Another important aspect, which is discussed in the literature is the the fairness and the ability to adapt to network congestion. Reliable multicast transmission protocols should not contribute to the congestion or make it even worse.

A good survey and a general categorization of different reliable multicast protocols is given by Roca *et al.* [53]. A summary about the Quality of Service issues, raised by dynamic changes to the multicast distribution tree and inhomogeneous transmission bandwidth in the multicast distribution tree is given by Striegel and Manimaran [54]. Also Lohmar *et al.* [55] describe the issues of Group Partitioning due to different bandwidth in the tree branches and due to different receiver capabilities.

The background about different reliable multicast protocols is necessary to understand its difference to Mobile Multicast and Broadcast Services.

3.3.1 Acknowledgment based Protocols

Acknowledgment (ACK) based reliable multicast protocols use a positive feedback about received data chunks to the sender. The sender keeps track of the reception status of each client. Unacknowledged data chunks are retransmitted. ACK based protocols are also categorized as "sender oriented" protocols, since the sender must identify the loss of data chunks.

The "Acknowledgment implosion problem" was identified early in the literature as one challenge to overcome. Reliable Multicast Transport Protocol (RMTP) [56] subdivides the receiver population into local regions, and Designated Receivers (DR) represent all receivers of the local region towards the sender. All receivers receive, but only the dedicated receiver acknowledges its own packets towards the sender. Other receivers also acknowledge their packets, but towards the designated receiver. The designated receivers are allocated manually. A dynamic scheme to automatically cluster the receivers into local groups and to assign the role of a designated receiver is missing.

Another solution to overcome the "feedback implosion problem" are tree based protocols [57]. Acknowledgment aggregation points are located at the branches of the multicast distribution tree. Only aggregated acknowledgments are sent upstream (basically reverse replication points) to the sender.

3.3.2 Negative acknowledgment based Protocols

Another set of protocols are based on negative acknowledgments (NACK). A NACK informs the sender about a missing packet or data chunk. The NACK based scheme elaborates on the knowledge that the IP packet loss rate is rather small (e.g. $< 10\%$). Thus, NACKs are sent much more seldom than ACKs and make the scheme more robust again the "feedback implosion" problem.

Further, the reception state information is kept by the receivers. Each receiver itself is responsible of comparing the actually received information with the sent information and sending a NACK if needed. The processing is done more decentralized.

However, the probability of receiving NACKs rises with increasing receiver population size. Also the probability of receiving two or more NACKs for the same data chunk rises with increasing population size.

Each receiver waits for a random "backoff-time" before sending a NACK in case of "NACK Suppression". NACKs are generally sent by multicast to the entire group. If a receiver sees a NACK for the same data chunk on the network, the receiver cancels the NACK timer and does not send a NACK. If the sender processes the NACK, it would send the requested data chunk to the entire group. This could be regarded as a downside of the protocol, since these retransmissions may be pure overhead, if only a very limited number of receivers need this data chunk.

The NACK suppression scheme only works fine with a well chosen back-off time window. If the window is too large, then the reliable multicast transmission is slowed down. If the back-off window is chosen too small or if the transmission delay between the receivers is too large, then the NACK suppression scheme looses efficiency: NACK suppression timer expires before another NACK is received.

The Scalable Reliable Multicast (SRM) protocol [58] evolves the NACK based protocols and adds a so-called "NACK suppression" technique. Receivers, which should send a NACK, wait for a random short time before sending the NACK. During this random time, the receiver listens for NACKs from other receivers. If a NACK for the same data chunk is received, the receiver does not send the NACK. By using the NACK suppression technique, the NACKs become local repair request messages. The closest receiver, which has successfully received the data chunk, will replay and serve the unsatisfied receiver.

XTP [59] is designed for either unicast or one-to-many multicast communication. Reliable communication is based on negative acknowledgments. The sender may also initiate a synchronizing handshake, to determine the status of the receivers. In this case, each receiver uses a "slotting" technique and introduces a random delay before sending their control packet, to reduce a control packet implosion. The combined slotting and damping techniques proposed in [59] to reduce NACK suppression have been described earlier in this paper. In XTP receivers or routers can impose a maximum data rate and maximum burst size on the sender.

MFTP [60] supports both ACK and NACK (preferred) based operation. A "product" is periodically announced during an announcement phase. The announcement message contains the ACK/NACK address. The server uses status request messages to query the status of the clients and transmits the file data in "passes". That is, the entire file is sent initially (i.e. pass 1). A file is subdivided into blocks and each block is further subdivided into packets. If any retransmissions are required, the server makes another pass (i.e. pass 2) through the file, but sends only those packets that were reported as missed by the Clients. Additional passes may be required to successfully deliver all packets to all clients. After the first sending phase, the server stops and waits for incoming feedback. It tries to resend only needed packets. The server queries all clients. NACKs are suppressed. The upper limit of DTUs per block is determined by the MTU size. A bit represents the reception status of a DTU. Otherwise, the block-size is determined by the status interval and the transmission rate. The error rate in the announcement indicates at which error rate a client shall stop listening and responding.

3.3.3 Use of Forward Error Correction

Huitema [61] has already very early studied the performance of *packet level FEC* as an alternative to ARQ. FEC has the advantage that the repair packets may correct any of the lost source packets. Thus, sending FEC repair to the entire group has benefits compared to sending source packets to the entire group.

Different usage method for Forward Error Correction (FEC) schemes are also evaluated in the literature. Jörg Nonnenmacher [62] evaluates in his PhD Thesis different combinations of FEC protection methods with retransmissions (both ACK and NACK). Beside others, Nonnenmacher showed that an integrated FEC scheme performs much better than an independent FEC layer. Integrated FEC means, that the FEC repair packets are also used for source block recovery during the feedback (ACK or NACK) phase.

Nonnenmacher *et al.* [63] aim to reduce the number of sent copies of the same information by using FEC. They propose a *Gain Definition* as a ratio between sent packets without FEC and sent packets with FEC.

Rizzo *et al.* [64][65] discusses the need and advantages of FEC in file transmission. They have shown that legacy digital processors are powerful enough to decode Reed-Solomon erasure codes. They developed and evaluated the Reliable Multicast data Distribution Protocol (RMDP), which is basically a combination of FEC and NACK. The Reed-Solomon FEC code uses 8 bit symbols ($m = 8$) and an encoding block size of $n = 255$.

Byers *et al.* [66] presents the general idea of a "Digital Fountain", thus a way to send an arbitrary number of packets. They introduce a new code family called *Fountain codes*, which have the interesting property of being *rateless*. This means that an arbitrary number of repair symbols can be created from a single source block. The paper evaluates "Tornado" codes, an unsystematic FEC code which belongs to the family of Fountain Codes and compares them against Reed-Solomon codes. Byers concludes that Tornado codes have a much better reception efficiency than Reed-Solomon codes. Construction and performance of Fountain codes including Digital Fountain's Raptor code are described in Sections 3.5.2 and 3.5.3 in further details.

Schooler *et al.* [67] want to solve the so-called *Midnight Madness* problem with reliable multicast transmissions (including FEC). Microsoft experienced severe network problems when releasing the Internet Explorer 3.0. The server and line capacity were exceeded due to too many simultaneous downloads of the new software. Schooler *et al.* first reviewed extensively the state of the art and then proposed a set of design goals for their protocol. FCAST starts with *cyclic* retransmissions of the software packet on a given IP Multicast address. A Reed-Solomon $(255, 32)$ code is used to improve reliability. The first version of FCAST called protocol does not include any client feedback. A client joins the multicast group and stays until all data chunks have been successfully received.

3.4 File delivery over unidirectional links

The IETF Reliable Multicast (RMT) Working Group defines one-to-many transport protocols for delivery of large data chunks. A large variety of applications may use reliable point-to-multipoint distribution of files. Unfortunately, those applications put very different, sometimes even contradicting requirements on the reliable transport. Due to the huge variety of different requirements [68], the RMT group decided to define a set of "building blocks" and also some *protocol instantiations*. Figure 3.4 depicts the basic building blocks for the Asynchronous Layered Coding (ALC) protocol that may use different Forward Error Correction codes. More details are provided in the next section. Note that using no FEC code is also defined as code (the Compact No-Code FEC), since the input object is partitioned in order to fit UDP packet payloads.

Building blocks form the least common dominator between the different application needs. The principles and structure of such building blocks are defined in [69]. Protocol instances may use different combinations of building blocks to get the desired protocol behavior. The specifications of such protocol instantiations define mostly the necessary *glue* between the different building blocks. The ALC protocol, which is described in detail in the next section, is an example of such a protocol instantiation.

The FLUTE protocol has become important in the definition of Mobile Broadcast Services in DVB, MBMS and OMA BCAST. The protocol is used to delivery files from a single sender to multiple receivers using the IP Multicast paradigm. Note that TCP requires a bi-directional channel and is not usable of unicast-only communication channels.

3.4.1 Asynchronous Layered Coding

Asynchronous Layered Coding (ALC) protocol [70] combines the Layered Coding Transport (LCT) protocol [71] with an FEC building block [72] and optionally a Congestion Control (CC) building block and defines a protocol. ALC is carried using UDP/IP packets and is independent of the IP version and the underlying link layers. The ALC packet format is depicted in Figure 3.7. The ALC header consists at least of the LCT header and the FEC payload id.

The relation of the different building blocks is depicted in Figure 3.4. ALC is intended to transmit one or more *objects* over IP Multicast to multiple receivers. Each object is uniquely identified in the scope of the session by a LCT Transport Object Identifier (TOI). ALC inherits multiple features from the LCT protocol.

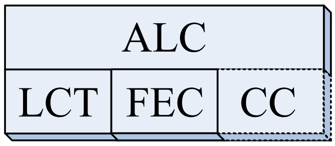


Figure 3.4: ALC Building Blocks.

LCT defines a transport session concept for multicast, which allows the simultaneous usage of several multicast groups. Although it is primarily designed for IP Multicast transport, it may still use IP unicast transmission. LCT is designed to offer an objected oriented delivery scheme. Each object is identified by a Transport Object Identifier (TOI) and has no limitation on size. The interpretation of the objects is not in scope of LCT.

An LCT session comprises multiple channels originating from a single sender that are used for some period of time. One reason for defining multiple channels within the same session is the use for congestion control. *Receiver-Driven* congestion control schemes as evaluated in [73] rely on the data organization in several multicast groups (so called *layers*). Receivers drop the highest order layer, if packet losses are measured.

Each transport session is identified and differentiated using a Transport Session Identifier (TSI). The transport session identifier is scoped with the IP address of the sender. The transport session identifier is a numerical value, which is available in each packet header. The size of the TSI and also the TOI values in the packet header are configurable using LCT header flags. The TOI field in the packet header may vary between 16 bit and 112 bit, depending on the value of the O and H flags. The TSI field may vary between 16 bit and 48 bit, depending on the S and H flags (cf. Figure 3.5).

Figure 3.5 depicts the LCT header field. The FEC header of the Compact No-Code FEC [74] is also depicted. The first four octets of the LCT header determine the total size of the LCT header. The flags determine the availability and also the size of various header fields like the TOI and the TSI. The Sender Current Time (SCT) and the Expected Residual Time (ERT) header fields shall not be used anymore (Flags setting: $T = 0$ and $R = 0$). A new more general timing extension header `EXT_TIME` is defined.

It is possible to add extension headers to the LCT default header. Two types of extension headers are defined: The fixed length header format is of 32 bit size and can carry 24 bit content. The variable length header extension uses multiple of 32 bit.

The Forward Error Correction (FEC) building block (cf. Fig 3.4) defines the Forward Error Correction code [72] and also an object partitioning scheme. The object partitioning algorithm describes the split of the object data into UDP packet payloads. Most of today's FEC codes operate on finite length source blocks and subdivide each source block into encoding symbols of an equal length,

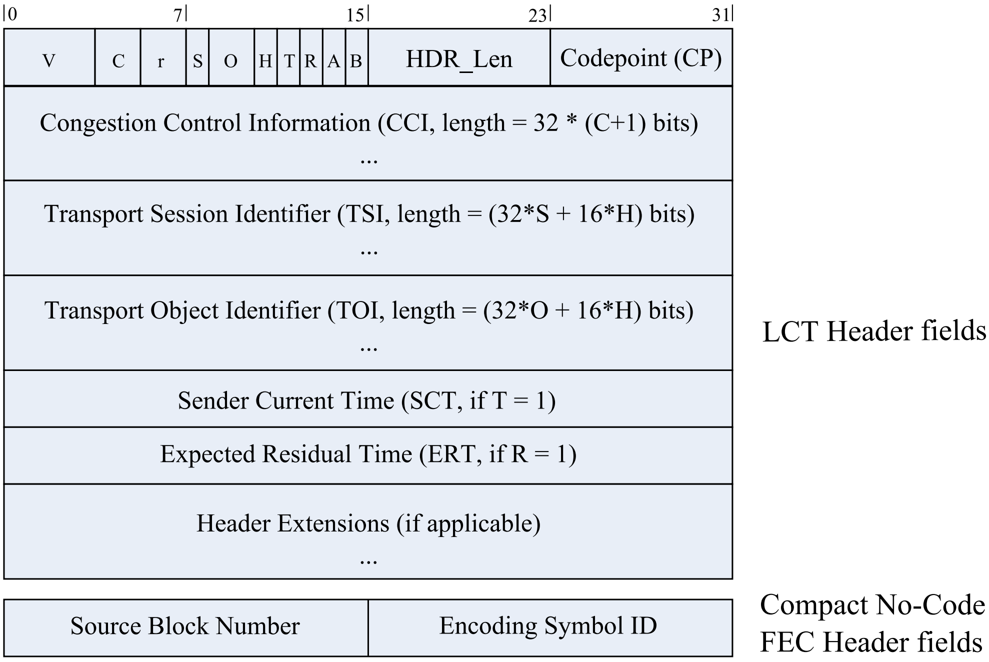


Figure 3.5: LCT and Compact No-Code FEC header Fields.

except the last symbol of a source block. One or more encoding symbols plus some header information fit into a UDP packet.

The so-called "Compact No Code FEC Scheme" [74] defines only a file partitioning algorithm and no FEC protection of the data. It uses the same terminology as real FEC codes. A file is partitioned, mainly depending on the target IP packet size, into one or more source blocks. Each source block is then partitioned into "Encoding Symbols", which fit into the IP packet. This file partitioning is also depicted in Figure 3.6. The FEC Payload ID (also depicted in Figure 3.5) describes the position of the encoding symbol within the source block. The FEC payload id can be regarded as a 32-bit sequence number. It consists of the Source Block Number (SBN) and the Encoding Symbol Id (ESI). Source Block Number and Encoding Symbol Id uniquely describe the position of a symbol or a packet payload in the memory for re-constructing the file. In case of the "Compact No Code FEC", each IP Packet carries exactly one encoding symbol.

Congestion Control (CC) building block (cf. Fig 3.4) is important for scalable and also a TCP-friendly use on a global Internet. MBMS [4] and DVB-H [75] do not define or require any congestion control. Instead, both systems offer path provisioning of the desired bandwidth.

The minimum ALC header is of 16 Bytes size and contains only the mandatory header fields at minimal possible length. The length of the Transport Object Identifier (TOI) and Transport Session Identifier are 16 bit each. The maximum specified ALC header depends on the used header extensions and on certain field

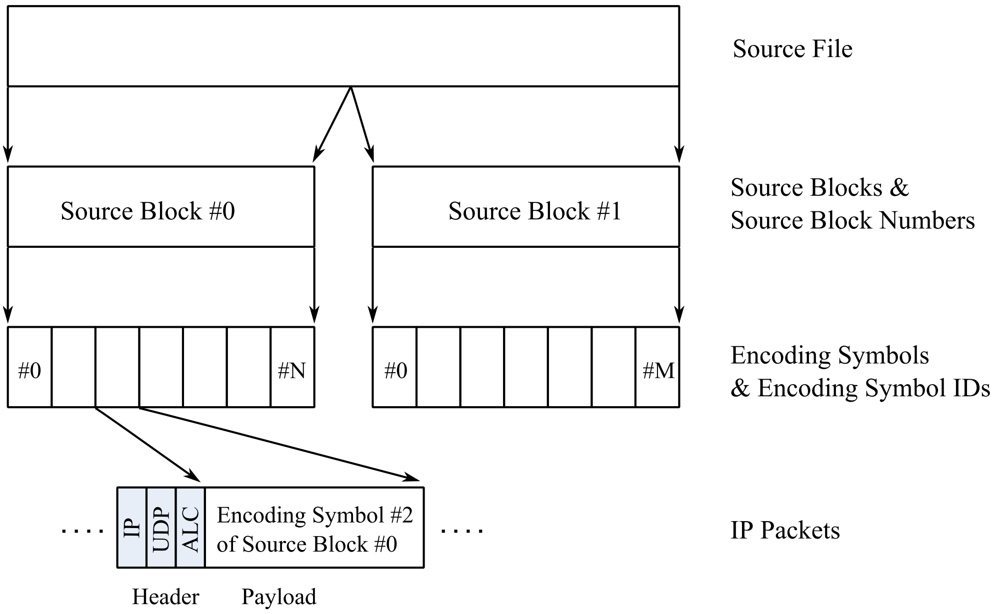


Figure 3.6: File Partitioning with *Compact No-Code FEC Scheme.*

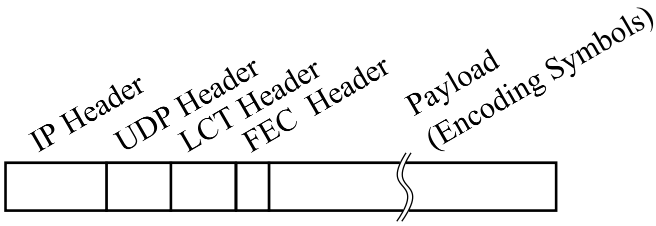


Figure 3.7: ALC Packet Format.

length.

3.4.2 The IETF FLUTE Protocol

The FLUTE protocol uses the object oriented transport service of ALC with all its features and adds functionality to transmit *files* instead of *objects*. FLUTE basically associates file properties such as a filename and a content-location to the ALC objects.

The FLUTE protocol is selected by MBMS [4], DVB-H IP Datacast [75] and OMA Mobile Broadcast Services [76] as file delivery protocol. MBMS [4] also defines a method to establish FLUTE sessions on unicast using the Real Time Streaming Protocol (RTSP) [18].

Files are ALC objects with additional properties: Files have a filename and also a content location associated. Each file also has a content type, which is

Transport Object Id (TOI)	Expires	Content Location (host/path/name)	Content-MD5 (optional)
5	A	www.ex.com/p/clip1.mpg	ODZiYTU1OTFkZGY2NWY5ODh==
6	A	www.ex.com/p/clip2.mpg	
9	B	www.ex.com/p/page1.html	
10	B	www.ex.com/p/page2.html	

Figure 3.8: FLUTE File Delivery Table (FDT).

derived from the file extension or described by the MIME Type [77][78]. ALC identifies the object using the Transport Object Identifier (TOI). FLUTE allows associating TOI values with file properties such as file name, content type, etc.

Each FLUTE receiver maintains the so-called *File Delivery Table (FDT)*, which is depicted as example in Figure 3.8. The receiver associates each received object with the according file properties. This association is deleted after some time, so that the TOI value range can be re-used. The Content-MD5 may be used to differentiate different versions of the same file.

These associations between TOI values and file properties are communicated with special XML files, which are also transported by ALC. The Transport Object Id $TOI = 0$ is reserved for FLUTE FDT instance objects. Each FDT Instance updates or extends the File Delivery Table in the receiver. The FDT expiration time determines the end-time of the TOI to file property association.

The FDT Instance contains similar information as the HTTP header fields, for instance the content-type information. With 3GPP Release 7, the FLUTE FDT is extended with cache control directive very similar to the HTTP cache control directives. This allows a receiver to do some active memory management and delete clearly outdated information objects.

The IETF FLUTE protocol allows the usage of Forward Error Corrections codes to improve the reliability of the data distribution. Those FEC codes are applied on application layer, above UDP protocol level. The IETF RMT group has at least defined the following codes:

- Compact No Code FEC ([74], [79])
- Reed Solomon FEC ([80])
- Different *Low Density Partity Check* (LDPC) codes ([81])
- Raptor FEC ([82])

At least the Compact No-Code FEC ([74], [79]) must be associated with each file entry. The receiver knows from the *FECEncodingId* element in the FDT

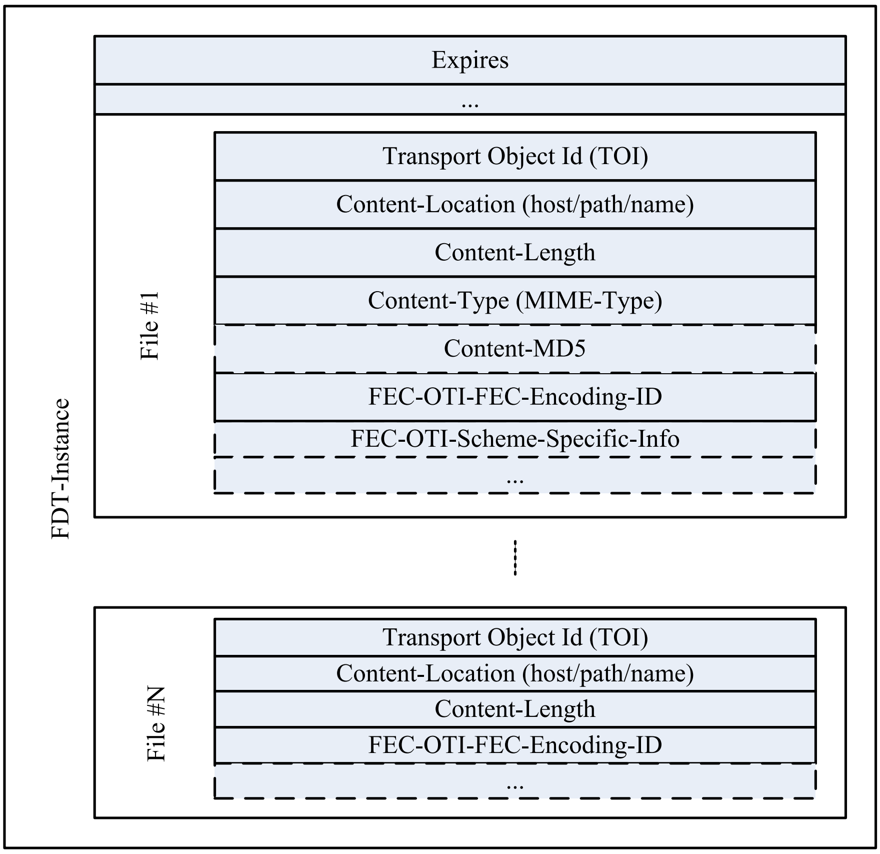


Figure 3.9: FLUTE FDT Instance.

about the used FEC code and the used partitioning algorithm. The FLUTE FDT contains also parameter values for the file partitioning algorithm so that the receiver can determine the encoding symbol size and the number of expected symbols.

The FEC Payload ID header for the Compact No-Code and for the Raptor FEC code has the same format, but other FEC Payload Id formats are defined by IETF as well. The FEC related information may also be provided using the LCT extension header `EXT_FTI`. This extension header must be used for FLUTE FDT octets (that one with $TOI = 0$). The FEC information for all other files can be described in the FLUTE FDT. The `EXT_FTI` extension header may help terminals to start reconstruction of the objects, if the FDT file was not received at the beginning.

FLUTE also defines a FLUTE instance id (`EXT_FDT`) extension header, which gives each FDT instance a unique identifier.

MBMS and DVB-H IPDC extend the FLUTE FDT with a *File Grouping* mechanism. This may be used to gather a number of files to a logical group.

3.5 LDPC and Fountain Codes

Low Density Parity Check (LDPC) codes and Fountain codes are important Forward Error Correcting (FEC) codes for binary erasure channels. Both codes have the advantage that very long codewords can be used, much longer than Reed-Solomon codewords at low computational complexity. Both code types use eXclusive OR (XOR) operations to produce parity check or encoding symbols. LDPC codes have a fixed parity check matrix, while Raptor and LT codes select one or more input symbols randomly.

Today, LDPC codes are widely used in many different areas, for instance in DVB-S2 or WLAN networks (IEEE802.11n). A form of LDPC code is also used as precode for Digital Fountains Raptor codes, which are described later in this section. Raptor codes are defined for mobile broadcast systems, including MBMS [3][4] and DVB-H [75][83]. Raptor codes are also defined for the DVB IPTV system [84].

3.5.1 LDPC Codes

Low Density Parity Check (LDPC) codes are first described by Galager [85] in 1963. LDPC codes are linear block codes which uses the principle of parity checks $\mathbf{H} \cdot b^T = \mathbf{0}$. The Matrix $\mathbf{H}$ is the so-called *Parity Check* Matrix and b is the sequence of received symbols (as row vector). The parity check matrix $\mathbf{H}$ is a sparse matrix, which is an important property for LDPC codes. In case of *regular* LDPC codes, all columns of the parity check matrix $\mathbf{H}$ have the same weight j. Further, all rows of the parity check matrix of regular LDPC codes have the same weight k. Thus, each source symbols is present in j check symbols. Each check symbol is the sum of k source symbols.

The construction of LDPC codes are of low computational complexity. However, finding a parity check matrix following the $\mathbf{H} \cdot b^T = \mathbf{0}$ constraint might not always be easy to find. LDPC codes are linear codes and can be obtained from a sparse bipartite graph. A bipartite graph $\mathcal{G}$ consists out of nodes, which can be subdivided into two disjoint sets of nodes N and R. The sets N and R are independent sets. Every edge of the graph $\mathcal{G}$ connects one node of the subset N with one node of the subset R. Thus, there are no edges between elements of the same subset N or R.

Figure 3.10 depicts the bipartite graph of an LDPC code. All n nodes on the left side $(x_1, .., x_8)$ are message nodes and r nodes on the right side are check nodes. The n coordinates of the codewords are associated with the n message nodes. The codewords are those vectors $(x_1, .., x_8)$ such that for all check nodes the sum of the neighboring positions among the message nodes is zero [86].

Effective decoding algorithms for LDPC codes are message passing algorithms, like *belief propagation* algorithms. The belief propagation algorithm is described in the next chapter in further detail.

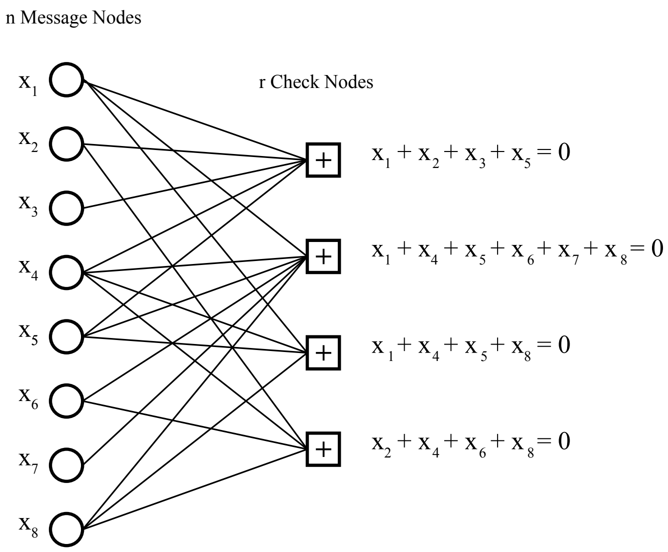


Figure 3.10: Bipartite graph of an LDPC Code.

3.5.2 LT Codes

The *digital fountain* concept was first introduced by Byers *et al.* in [66] as part of the concept of an efficient, scalable and reliable multicast and broadcast protocol. A digital fountain, similarly to a water fountain producing an endless stream of water drops, injects a continuous stream of encoding packets into the network, from which a receiver can reconstruct the source data. A receiver should be able to recover the files from any of the received symbols. Byers *et al.* introduced the class of *fountain* FEC codes, but did not reveal any code construction. LT codes [87] and Raptor codes [88][89] belong to the family of fountain codes.

One design goal of a fountain code is that an arbitrary large number of encoding symbols can be generated from a source block of fixed length. This behavior is also called *rateless*, since the coding rate becomes very small, if the number of encoding symbols is much larger than the actual number of source symbols. The fountain encoder must be able to generate on the fly as many encoding symbols from source data as needed.

An important property of fountain codes is that the number of encoding symbols is not an input parameter to the encoding process. It is always possible to request additional FEC encoding symbols from the encoder. In case of a traditional Reed-Solomon or LDPC codes, the source block length (k) and the number of desired FEC redundancy symbols ($n-k$) are input parameters for the encoding process. The structure of the code is determined before it is used for transmitting data. The FEC code with its encoding matrix is statically chosen before the encoding process is started. Of course, the sender does not need to immediately send all data (puncturing), but the number of repair symbols is limited and depends on the selected FEC code. With fountain codes, it is not

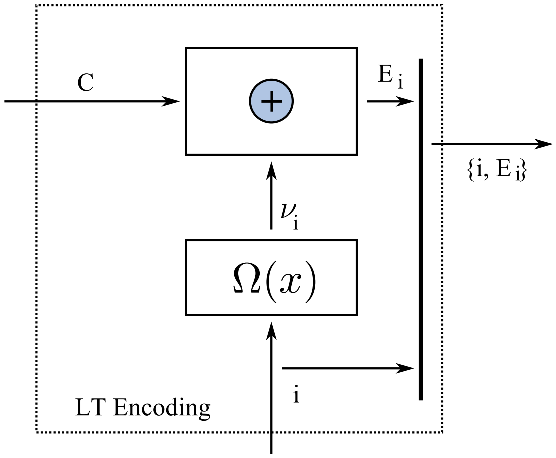


Figure 3.11: LT Encoding Block diagram.

required to provide the number of encoding symbols as input parameter into the encoding process.

LT codes were invented by Michael Luby and described in [87] in detail. LT codes (*Luby Transform* codes) are introduced as first fountain codes, which can produce an (almost) unlimited amount of new encoding symbols. An encoding symbol is generated by eXclusive ORing (XOR) two or more input symbols into the encoding symbol. The *degree* of an encoding symbol is the number of input symbols XORed into the encoding symbol. The number of different encoding symbols depends on the number of possible combinations to XOR input symbols and the used degree distribution. The higher the number of source symbols the more different encoding symbols are possible.

Figure 3.11 depicts the block diagram for LT encoding. $\mathbf{C}$ is the vector of k input symbols. $\Omega(x)$ is the degree distribution function to determine the degree of the encoding symbols $\mathbf{E_i}$. An encoding symbol is produced by first sampling a vector ν_i from the degree distribution function $\Omega(x)$. Any output vector ν_i of the distribution function $\Omega(x)$ belongs in the linear subspace of $\mathbb{F}_2^k$ and is used as generator matrix for the encoding symbol E_i. $\mathbb{F}_2$ is the Galois Field with 2 elements. The LT encoding algorithm is described below in detail.

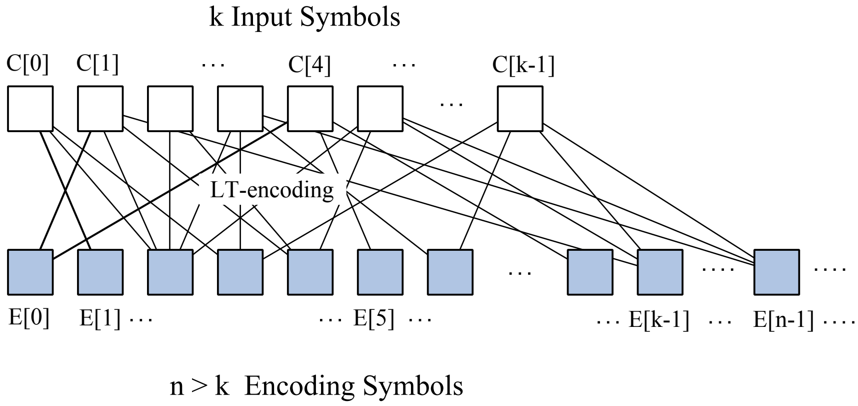


Figure 3.12: LT Encoding.

Algorithm 3.1 LT Encoding

---

1. Choose randomly the degree d with $d \in [1, k]$ for the encoding symbol from a degree distribution. The degree of an encoding symbol describes, how many input symbols are XORed into the encoding symbol
2. Choose randomly d input symbol indexes ω_j ($j \in [0, d-1]$) out of the set of input symbols $C[\cdot]$.
3. XOR the d randomly chosen input symbols $C[\omega_j]$ into the encoding symbol: $E[i] = C[\omega_0] \oplus C[\omega_1] \oplus \cdots \oplus C[\omega_{d-1}]$.
4. Provide the indexes $\omega_0, ..., \omega_{d-1}$ of the contributing input symbols $C[\omega_i]$ with the encoding symbol to the receiver.

Figure 3.12 depicts an example of LT Encoding. The encoding symbol $E[0]$ is for instance of degree 2 and the two input symbols $C[1]$ and $C[4]$ are eXclusive-ORed (XORed) to $E[0] = C[1] \oplus C[4]$ with $\omega_0 = 1$ and $\omega_1 = 4$. The encoding symbol $E[1]$ is of degree 1, since it only containing the value of the input symbol $C[0]$.

Eq. (3.1) describes the encoding procedure as matrix operation over $\mathbb{F}_2^k$ for the first m encoding symbols $E[0]$, .., $E[m-1]$. Vector $\mathbf{C}$ is a column vector containing the input symbols $C[0]$, ..., $C[k-1]$. Matrix $\mathbf{G}$ is the generator matrix, with dimension $(k \times m)$. Each row of $\mathbf{G}$ is sampled from the degree distribution $\Omega(x)$. The Hamming Weight of each row is equal to the degree of the according encoding symbol. Matrix $\mathbf{G}$ is a binary matrix, thus each element is either 0 or 1.

$$\mathbf{E_0}, .., \mathbf{E_{(m-1)}} = \mathbf{G} \cdot \mathbf{C} \tag{3.1}$$

The principle of this type of encoding is of course very nice. The crux is to find a proper distribution function $\Omega(x)$, which minimizes the number of needed encoding symbols to recover the source symbols. The decoding performance of the LT codes depend very much on the selection of a good degree distribution function.

Luby discusses in [87] two different degree distribution functions: The *Ideal Soliton* function and the *Robust Soliton* function. The Ideal Soliton function $\rho(d)$ is defined in Eq. (3.2), where k is the number of available input symbols. Then, $\rho(d)$ is the probability, that the chosen degree for an encoding is d with $1 \leq d \leq k$. k is the number of source symbols.

$$\rho(d) = \begin{cases} \dfrac{1}{k} & , \text{ for } d = 1 \\ \dfrac{1}{d(d-1)} & , \forall\, d \in [2, 3, .., k] \end{cases} \tag{3.2}$$

The expected degree $E(d)$ for the encoding symbols using the Ideal Soliton distribution is according to Eq. (3.3) roughly $(\ln k)$.

$$E(d) = \sum_{d=1}^{k} d \cdot \rho(d) = \frac{1}{k} \sum_{d=2}^{k} \frac{1}{d-1} \approx \ln k \tag{3.3}$$

The Ideal Soliton distribution is designed in such a way, that k encoding symbols are sufficient to recover each of the k source symbols with a very high probability. For each iteration of the decoding process, there is exactly one encoding symbol with a degree of one. However, this is not the case in practice due to the differences between the expected behavior and the actual behavior of the distribution, caused by variance. This variance easily leads to a situation where there are no encoding symbols of degree one in an iteration, thereby causing the decoding to fail. Luby concluded in [87] that the performance of the *Ideal Soliton* distribution is *poor* and has therefore designed and (described in [87]) the *Robust Soliton* Distribution $\mu(i)$.

$$\tau(d) = \begin{cases} \dfrac{S}{dk} & \text{for } d \in \left[1, \dfrac{k}{S-1}\right] \\ \dfrac{S}{k}\ln(S/\delta) & \text{for } d = \dfrac{k}{S} \\ 0 & \text{for } d \in \left[\dfrac{k}{S+1}, \dfrac{k}{S+k}\right] \end{cases} \tag{3.4}$$

δ is the allowable failure probability of the decoder for a given number of k of encoding symbols. S is defined in Eq. (3.5).

$$S = c \cdot \ln(k/\delta)\sqrt{k}\ \text{, with } c > 0 \text{ a suitable constant} \tag{3.5}$$

$$\mu(i) = \frac{\rho(i) + \tau(i)}{\sum_{n=1}^{k} \rho(n) + \tau(n)}\ ,\ \forall\, i \in [1, k] \tag{3.6}$$

The *Robust Soliton function* is defined in Eq. (3.6). It is the normalized sum of the two functions $\rho(i)$ of Eq. (3.2) and $\tau(i)$ of Eq. (3.4). It secures that the probability of having one encoding symbols of degree one in each decoding iteration is increased.

The *encoding cost* of an LT code is the expected number of operations sufficient to calculate one encoding symbol. For LT codes at most $(d-1)$ operations are needed, where d is the expected degree (or Hamming weight) of the encoding symbol. The average encoding cost for LT codes is then $(E(d) - 1)$ XOR operations to calculate one output symbol.

There are several methods for decoding LT codes: One very obvious way is first construct on the receiver side the generator matrix $\mathbf{G}$ and then to invert the matrix for instance with Gaussian Elimination. The receiver knows, which input symbols are XORed into the encoding symbol. Each received symbol results in one row in the generator matrix. The element $\mathbf{G}_{(i,j)}$ of the generator matrix is 1, when the jth input symbols is XORed into the ith encoding symbol.

Another method is using the *Belief Propagation* algorithm [90][89][87], which is an iterative method. The decoder first looks for one encoding symbol of degree one. Note that an encoding symbol with degree one is equal to the input symbol. Then the decoder XORs the input symbol into other encoding symbols, which contain this input symbol. By doing this, the decoder decreases the degree of the other encoding symbols hopefully to one, so that the decoder can repeat the process in the next iteration. If the decoder reduce the degree of any dependent encoding symbol to one, then the decoding procedure fails.

Let $C[i]$ with $i \in [0, k-1]$ be the k input symbols, which should be recovered by the decoding process. Let $E[l]$ with $l \in [0, n-1], n \geq k$ be the n encoding symbols, entering the decoding process. Let $\Psi(x)$ be the vector of contributing input symbol indexes ω to encoding symbol $E[x]$. The Hamming weight of $\Psi(x)$ is the degree of the Encoding symbol $E[x]$. There are several ways to obtain $\Psi(x)$.Then the $C[\cdot]$ input symbols can be recovered using the algorithm described by Algorithm 3.2.

Algorithm 3.2 LT Decoding with *Belief Propagation*

Decoding Process: while not done do:

1. Create a list $D[\cdot]$ containing all encoding symbols from $E[\cdot]$ with degree $d = 1$. Each entry of $D[\cdot]$ is a tuple of symbol value and corresponding source symbol id. Note that the list of encoding symbols $E[\cdot]$ is continuously modified in each iteration.

2. Remove all elements in $D[\cdot]$ from the remaining list of encoding symbols $E[\cdot]$. If $E[\cdot]$ is empty, then the **decoding is successful**.

3. If the list $D[\cdot]$ is empty, then the **decoding is failed**. There must be at least one encoding symbol of degree $d = 1$ in each decoding iteration.

4. For remaining encoding symbol $E[j]$ in $E[\cdot]$ do:

   - Does $\Psi(j)$ of $E[j]$ contain any source symbol id (ω) of $D[\cdot]$. Meaning, was any of the encoding symbols with degree $d = 1$ used to generate the encoding symbol $E(j)$
   - If no, then increase j by one and skip the remaining steps of the iteration.
   - If yes, then $E[j] = E[j] \oplus D[i]$. XOR the found source symbol $D[i]$ into the encoding symbol $E[j]$.
   - Remove the source symbol index (ω) of $D[i]$ from $\Psi(j)$.

To visualize the decoding process, we use the LT code from Figure 3.12. The decoder starts with selecting an encoding symbol of degree one, which is at least $E[1]$ and $E[5]$ in Figure 3.12. The encoding symbol $E[5]$ is of degree one, thus $E[5]$ is equal to the input symbols $C[4]$. The input symbols $C[4]$ is XORed into some more encoding symbols. One of those is the encoding symbol $E[0]$ and the decoder decreases the degree of encoding symbol $E[0]$ to one by XORing the input symbol $C[4]$ with $E[0]$. The decoder can continue with this process until all input symbols have been recovered. If there are no symbols with degree one anymore before all input symbols have been recovered, the decoding process fails. The decoder needs to receive at least one more encoding symbol and repeat the decoding attempt.

According to Shokrollahi [89], the described Belief-Propagation algorithm fails miserably for random LT code even when a large number of encoding symbols is collected. The Raptor FEC code addresses these disadvantages and also ensures good asymptotic performance at low overhead.

The average encoding and decoding cost of LT codes using the Robust Soliton distribution is $O(\log(L/\delta))$. δ is the failure probability of the decoding procedure.

3.5.3 The Raptor FEC Code

3GPP [4] adopted the Raptor FEC Code as the only MBMS FEC code to optionally improve the transmission reliability on MBMS bearers. 3GPP made the Raptor FEC code *mandatory* to be provided by MBMS receivers. Very early in the standardization process was the requirement, that only a single FEC code should be selected. MBMS is a broadcast system, thus proving the broadcast flow in multiple different transmission formats (e.g. with different FEC codes) reduces the system efficiency and increases the complexity. Further, this single FEC code should allow for battery efficient decoding while providing a very good decoding performance (making the most out of the received FEC redundancy). The property of the Raptor code to produce a high number of additional FEC symbols (*rateless* code) is very beneficial for mobile broadcast transmissions, but not required.

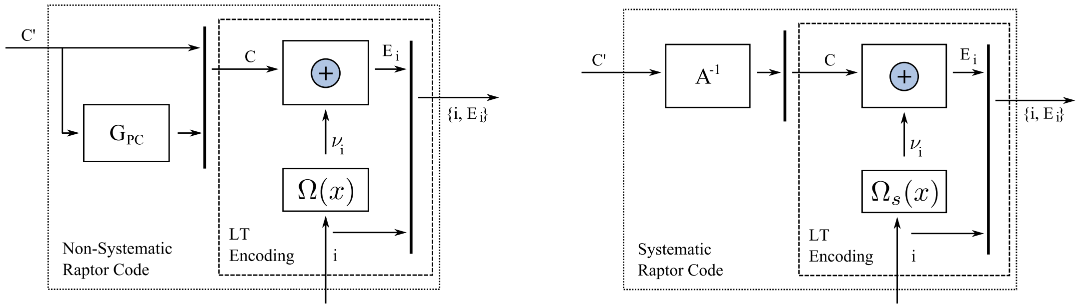
The Raptor FEC code is an evolution of LT codes and are invented and first described by Amin Shokrollahi [89]. Since the Digital Fountain's Raptor FEC code was adopted by 3GPP in 2006 as the only FEC code for MBMS download and streaming [4], Digital Fountain has also started an activity to get endorse the Raptor FEC by IETF [82].

One key motivation for introducing Raptor codes is to relax the decoding requirements for LT codes by introducing a second FEC code. This second FEC Code is called the *Pre-Code*. The decoding graph of LT code needs to have an order of $k\log(k)$ edges in order to make sure that all input nodes are recovered with a high probability. When an LT code needs to recover only a constant fraction of input symbols, then its decoding graph needs only have $O(k)$ edges.

A Raptor FEC Code is defined by parameters $(K, \mathcal{C}, \Omega(x))$, where $\mathcal{C}$ is a linear code of block length L and dimension K. $\Omega(x)$ is the degree distribution of the LT code. The pre-code $\mathcal{C}$ takes K input symbols and generates L intermediate symbols. The output symbols or encoding symbols of the Raptor code are generated from the L intermediate symbols using an appropriate LT code.

This means in principle that the LT decoding process of Raptor FEC need to recovery any K of the L intermediate symbols. The K source symbols can then be recovered with a certain decoding success probability by the pre-code. If the decoding with the pre-code fails, then the receiver needs to gather more encoding symbols to recover a larger number of intermediate symbols.

Main advantage of Raptor codes compared to LT codes is that Raptor can encode and decode with constant costs when the number of collected encoding symbols is close to the number of source symbols [89]. For a given integer k and any real $\epsilon > 0$, a receiver can reconstruct a Raptor transmission at high probability when any $k(1+\epsilon)$ symbols are received. ϵ is also called the *overhead* [89]



(a) Non-Systematic (b) Systematic

Figure 3.13: Block diagrams of Raptor Codes. The precode matrix $\mathbf{A^{-1}}$ of the systematic Raptor FEC corresponds to a Raptor decoding operation of the intermediate symbols $\mathbf{C}$ from the source symbols $\mathbf{C}'$.

or *inefficiency* [91] of the code.

Figure 3.13 depicts block diagrams for a systematic and non-systematic Raptor Code, similar to the block diagram in [91]. The encoding is a two step process, where a fixed number of intermediate symbols are calculated in a first step. LT encoding is applied in a second step using the intermediate symbols as input. The decoding of the non-systematic Raptor FEC can be done in a single step by inverting the generator matrix. The generator matrix contains the LT code and the pre-code symbol relationships and is discussed later in this section in detail. The decoding of the systematic Raptor is a two step process, since the intermediate symbols are not systematic.

Figure 3.13a depicts the non-systematic Raptor Code. $\mathbf{G_{PC}}$ is the generator matrix for the Raptor precode $\mathcal{C}$. Figure 3.13b depicts the systematic Raptor code. The matrix $\mathbf{A^{-1}}$ is the generator matrix for the intermediate symbols. The systematic Raptor FEC is discussed later in this section.

The LT encoding is in both Raptor forms the same, but possibly with different degree distribution functions $\Omega(x)$. ν_i is a vector in $\mathbb{F}_2^k$, which is sampled from the degree distribution $\Omega(x)$ in order to produce the *ith* encoding symbol. The Hamming Weight of ν_i is the degree of the output symbols. The result of the operation $\nu_i \cdot C[\cdot]$ is the encoding symbol $E[i]$.

There are two extreme codes in the Raptor code family: When we select a trivial degree distribution $\Omega(x) = x$, then the Raptor code $(k, \mathcal{C}, x)$ is a *pre-code only* Raptor code (PCO Raptor). On the other side, an LT code with k input symbols and an output distribution $\Omega(x)$ is a Raptor code with parameters $(k, \mathbb{F}_2^k, \Omega(x))$.

The encoding cost of a Raptor code is defined slightly different than of an LT code, since the operations of the pre-code needs to be considered as well. The encoding cost for Raptor is defined as $E(\mathcal{C})/k + \Omega'(1)$, where $E(\mathcal{C})$ is the number of operations to generate a intermediate symbols from the k input symbols.

A Raptor code with good asymptotic performance is presented by Shokrol-

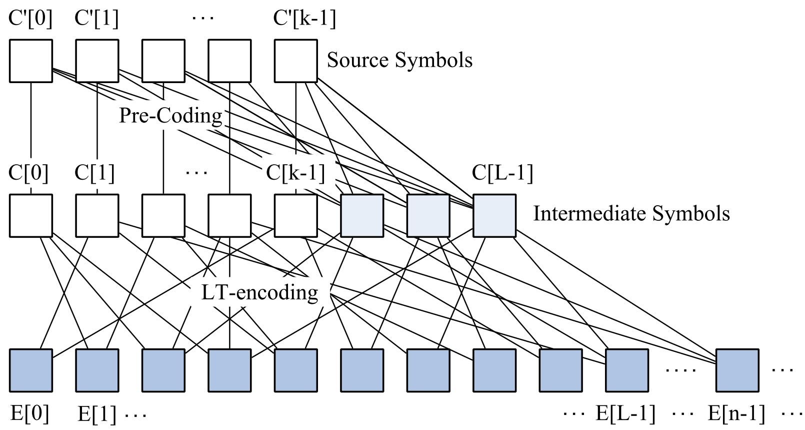


Figure 3.14: Non-Systematic Raptor Encoding.

lahi [89]. The degree distribution is a modified Soliton distribution, which is capped at a maximum degree of D. The overhead ϵ is a real number larger zero and, D is an integer and set to $D = \lceil 4(1+\epsilon)/\epsilon \rceil$.

$$\Omega_D(x) = \frac{1}{\mu+1}\left(\mu x + \sum_{i=2}^{D} \frac{x^i}{(i-1)i} + \frac{x^D+1}{D}\right) \tag{3.7}$$

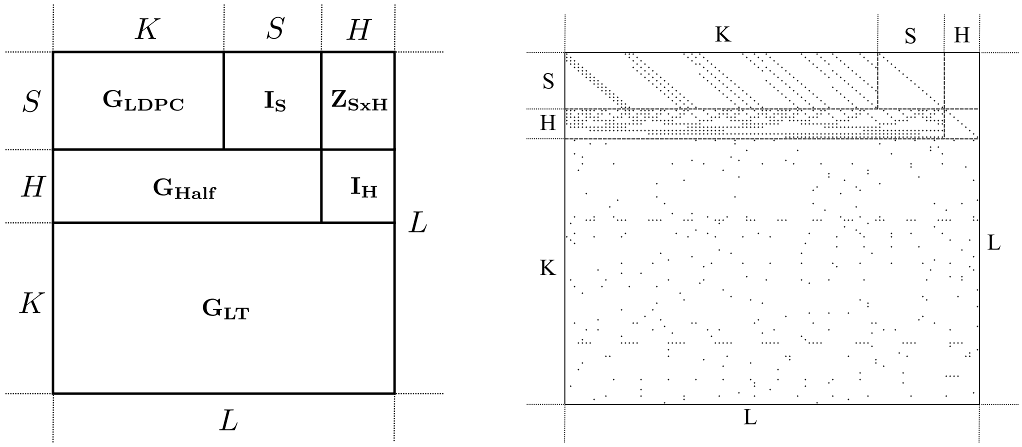
where $\mu = (\epsilon/2) + (\epsilon/2)^2$. Eq. (3.7) defines the degree distribution for the LT code of the Raptor code. The LT code with the parameter (n, Ω_D) is capable of recovering at least $(1-\delta)n$ input symbols with a belief propagation decoder at a high probability. δ is defined in Eq. (3.8).

For the pre-code, Shokrollahi suggests a linear block code $\mathcal{C}_n$ with code rate $R = (1+\epsilon/2)/(1+\epsilon)$. A belief propagation decodes can decode $\mathcal{C}_n$ on a Binary Erasure Channel with erasure probability δ as given in Eq. (3.8).

$$\delta = (\epsilon/4)/(1+\epsilon) = (1-R)/2 \tag{3.8}$$

The belief propagation decoder needs $O(n \log(1/\epsilon))$ operations for the decoding procedure. The encoding and decoding cost for the above described Raptor Code $(k, \mathcal{C}_n, \Omega_D(x))$ is $O(\log(1/\epsilon))$

The non-systematic Raptor is defined in TDoc S4-040230 [92] and takes a sequence $C'[\cdot]$ of K source symbols as input. The parameters L, S and H for the pre-code are derived from K. The encoding scheme of a non-systematic Raptor code is depicted in the Figure 3.14. The lines between input and intermediate symbols and intermediate and encoding symbols depict exclusive-OR operations (XOR) of symbols. The non-systematic Raptor uses a systematic pre-code. An LDPC code is used for S intermediate symbols. An extended Hamming code is used for the remaining H symbols. LDPC codes are constructed in such a way that $\mathbf{H} \cdot r = 0$ for any codeword r. $\mathbf{H}$ is the parity check matrix of the



(a) Raptor FEC generator submatrixes to generate the intermediate symbols.

(b) Example of Matrix **A** with $K = 80$, $S = 17$ and $H = 9$. Each *dot* represents a 1 in the generator matrix

Figure 3.15: Raptor FEC Generator Matrix **A** to generate Intermediate Symbols.

LDPC code. Thus, the receiver *knows* already the values of $(S + H)$ symbols, needed for decoding.

The encoding symbols are generated using $(L, \Omega(x))$ LT code. To generate the encoding symbols, the encoder samples a degree d from the degree distribution $\Omega(x)$. Then the encoder calculates the encoding symbol by randomly selecting d intermediate symbols and XORing them into the encoding symbol.

Figure 3.13b depicts the block diagram of the Systematic Raptor. The systematic Raptor code uses a non-systematic pre-code. The matrix $\mathbf{A^{-1}}$ is the generator matrix for the intermediate symbols. Matrix $\mathbf{A^{-1}}$ is the inverse of generator the matrix **A** for the Raptor FEC code. The intermediate symbols are calculated by decoding the source symbols from a known generator matrix.

The systematic Raptor is designed is such a way that the LT encoding of the K first encoding symbols have the same value as the source symbols. This puts certain requirements on the intermediate symbols. The intermediate symbols are created by a decoding process during the pre-coding step. A set of pre-coding relationships hold within the intermediate symbols themselves. A set of *systematic indexes* are defined in [4][82] so that the requirements on the intermediate symbols are fulfilled. The matrix **A** is constructed to generate the intermediate symbols for the source symbols. Matrix **A** is created using the LT Encoding vectors ν_i of the first K encoding symbols and using the generator matrix for the pre-code.

The construction of generator matrix **A** for the intermediate symbols is depicted in Figure 3.15a. Figure 3.15b depict an example instantiation of Matrix **A** with $K = 80$ source symbols. According to [4][82], the number of LDPC and Half symbols are calculated to $S = 17$ LDPC symbols and $H = 9$ Half symbols.

Each dot represents a one in the matrix.

Matrix $\mathbf{A}$ is a binary matrix over $\mathbb{F}_2$. Each element of the matrix is either 0 or 1. Each 1 (or *dot* in Figure 3.15b) in the matrix corresponds to the exclusive-OR of the corresponding input symbol into the output symbol.

The submatrix $\mathbf{G_{LDPC}}$ is used to generate *Low-Density Parity Check* (LDPC) symbols [85] and is of dimension SxH. The LDPC *staircase* [90] like generator submatrix is good visible Figure 3.15b. Each source symbol contributes here exactly to three intermediate symbols. Together with the identity matrix $\mathbf{I_S}$, the result of each row of this part of the matrix is zero.

The submatrix $\mathbf{G_{Half}}$ contain Half Symbols, which are derived from a *filtered* Gray sequence. The Gray sequence $g(i)$ was developed during times of physical switches. The sequence has the property, that always only a single bit position changes from $g(n)$ to $g(n+1)$. The Gray sequence is filtered in such a way, that always exactly k bit positions are different in binary representation of $g(i)$. For the example in Figure 3.15b, a filtered Gray codes with always 5 Non-Zero bits in the binary representation is used. Always *5 dots* are in each column of the submatix $\mathbf{G_{Half}}$, meaning that each source symbol contributes five times to intermediate symbols. Together with the identity matrix $\mathbf{I_H}$, the result of each row of this part of the matrix is zero.

The Matrix $\mathbf{G_{LT}}$ is a submatrix corresponding to LT encoding. The number of dots per row are according to a degree distribution.

The submatrix $\mathbf{I_S}$ and $\mathbf{I_H}$ are identity matrices of dimension SxS and HxH, respectively. The submatrix $\mathbf{Z_{SxH}}$ is zero matrix, where all elements are zero. Calculating the intermediate symbols is equal to solving the following set of linear equations for the vector $\mathbf{C}$ of intermediate symbols.

$$\mathbf{A} \cdot \mathbf{C} = \mathbf{C}' \tag{3.9}$$

$\mathbf{C}'$ is a column vector with the source symbols of length L. The first $S+H$ symbols of vector $\mathbf{C}'$ are zero symbols, followed by the K source symbols. The first $S+H$ symbols are the results of the LDPC pre-code which is always $\mathbf{H}\cdot c = 0$. Vector $\mathbf{C}$ is a column vector with the L intermediate symbols.

The Eq. (3.9) can be solved for instance using Gaussian Elimination. However, Gaussian Elimination can be very computational demanding. An efficient decoding procedure for Raptor FEC is described in the Raptor specification [27][4].

The actual encoding symbols are generated using LT encoding from the L Intermediate symbols. Figure 3.12 and 3.14 depicts the generation of the encoding symbols from the L Intermediate Symbols. A number of intermediate symbols are XORed into one encoding symbol as defined by the degree distribution $\Omega(x)$. The values of the first K encoding symbols are the same as the source symbols.

The decoding of the systematic Raptor FEC is done in two steps: The intermediate symbols are recovered in a first step. Since the intermediate symbols of the systematic Raptor FEC are generated by *decoding* the source symbols, the

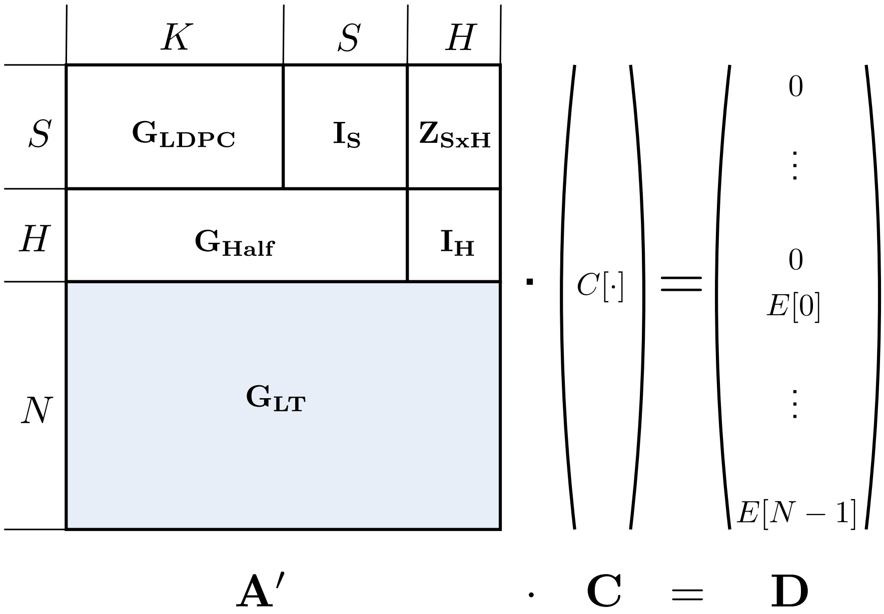


Figure 3.16: Decoding Matrix Construction.

intermediate symbols are multiplied with the generator matrix $\mathbf{A}$ as defined in Eq. (3.9).

For recovery of intermediate symbols from the encoding symbols, the receiver must first collect $N \geq K$ encoding symbols. The decoding Matrix $\mathbf{A}'$ is depicted in Figure 3.16. The only difference to the encoding Matrix $\mathbf{A}$ is the construction of $\mathbf{G_{LT}}$: The matrix is of dimension $(L \times M)$, with $M = N + S + H$. The receiver may collect more than K encoding symbols. The structure of the submatrix is different, since the receiver has collected a different set of symbols than the first K. The receiver generates each row of the matrix from the encoding symbol id and the random generator. For example, the ith row of G_{LT} is the vector ν_i, which is sampled from the degree distribution $\Omega(x)$.

Typically, invertible matrices are square matrices with dimension $(n \times n)$. Raptor FEC decoding is in principle about inverting a matrix $\mathbf{A}'$ with dimension $(M \times L)$ and $M \geq L$. The receiver must be prepared to delete linear dependent rows. When Gaussian Elimination is used for inverting the matrix, the decoding is successful when the matrix $\mathbf{A}'$ is either converted into a $L \times L$ identity matrix or contains a $L \times L$ identity matrix.

Algorithm 3.3 Systematic Raptor Decoding

1. Collect $N \geq K$ encoding symbols $E[\cdot]$ from the transmission

2. Create the decoding vector $\mathbf{D}$ of length $M = N{+}S{+}H$ from the N received encoding symbols. The first $S+H$ symbols of the decoding vector are zero symbols, corresponding to the LDPC and Half symbols. The later K symbols are the N received encoding symbols $E[\cdot]$.

3. Create decoding Matrix $\mathbf{A'}$:

   - The upper part of Matrix $\mathbf{A'}$ are the generator submatrixes $\mathbf{G_{LDPC}}$, $\mathbf{G_{Half}}$, $\mathbf{I_S}$, $\mathbf{I_H}$ and $\mathbf{Z_{SxH}}$.
   - The lower part of Matrix $\mathbf{A'}$ are the corresponding XOR operations for the LT encoding used to generate the encoding symbol.
   - Matrix $\mathbf{A'}$ is of dimension MxL with $M \geq L$.

4. Solve the equation $\mathbf{A'} \cdot \mathbf{C} = \mathbf{D}$ for instance using Gaussian Elimination, belief propagation or the optimized decoding algorithm from [82] or [4]. $\mathbf{D}$ and $\mathbf{C}$ are column vectors with M and L, respectively, symbols. Note that not all intermediate symbols needs to be recovered in order to recover the source symbols.

   - Decoding fails, when Matrix $\mathbf{A'}$ is not invertible. Then, a larger number of encoding symbols (e.g. $N+1$) is needed. The decoding process is restarted.

5. Vector $\mathbf{D}$ is the vector of non-systematic Intermediate symbols. The source symbols $\mathbf{C'}$ are calculated according $\mathbf{C'} = \mathbf{A} \cdot \mathbf{D}$. Note that the submatrix $\mathbf{G_{LT}}$ of $\mathbf{A}$ is different than $\mathbf{A'}$

Since the degree of the encoding symbols and the input symbols itself are randomly chosen, it is not always secured that the decoding with $N = K$ encoding symbols is successful. Instead, the receiver needs to collect $N = (1+\epsilon) \cdot K$ encoding symbols to recover the source block.

Figure 3.17 depicts the CDF of the decoding failure probability (δ) over the overhead ϵ for different number of source symbols K. The Raptor specification recommends to use at least $K = 1024$ source symbols. The Raptor specification [4][82] allows to use between $K = 4$ and $K = 8192$ source symbols. Larger

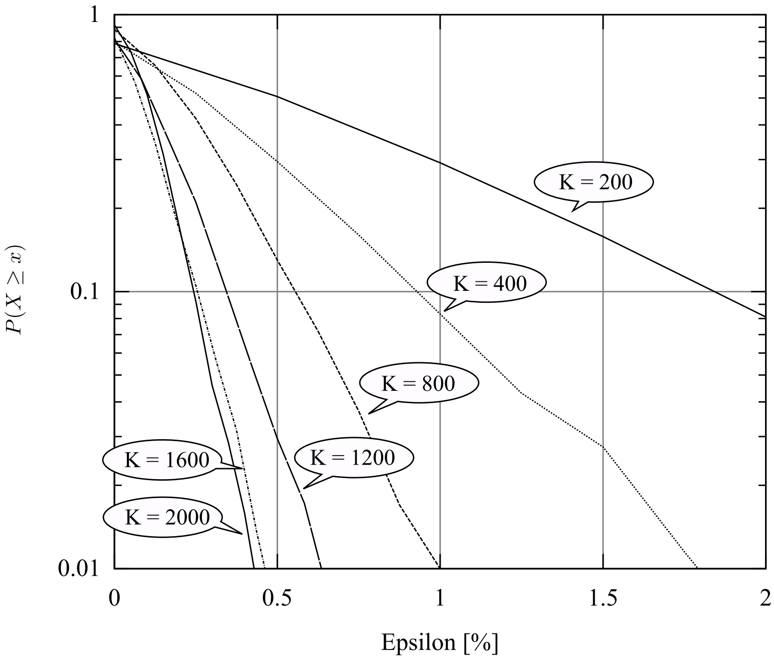


Figure 3.17: Raptor Decoding Performance.

numbers of source symbols are technically possible, but the specification only supports up to $K = 8192$ source symbols. The specification recommends to partition a file into at least $K = 1024$ source symbols. It is not necessary to use the packet payload size as symbol size. It is also possible to have multiple symbols by packet.

The decoding overhead ϵ decreases with increasing source symbol numbers K. Thus, with increasing number of source symbols K the probability increases to recover the source block from $N = K$ received encoding symbols. For instance, the decoder can successfully recover the source block in $\approx 90\%$ of the cases, when the receiver has collected 1.8% extra encoding symbols for $K = 1200$ source symbols. When the source block is partitioned into 1600 source symbols, then the receiver needs only $\epsilon = 0.25\%$ to recover the file with a 90% probability.

We have not yet discussed the partitioning of an input file into source symbols. Source symbols and encoding symbols must have the same size, since encoding symbols are generated from other symbols using XOR operations.

Figure 3.18 depicts the file partitioning principle to create source symbols from a source file. A source file may be first subdivided into two or more source blocks of roughly equal size. In particular mobile phones are limited in memory. Therefore, 3GPP has limited the maximal source block size. The source blocks are encoded and decoded independent from each other. The Raptor payload format allows to identify different source blocks using a source block number.

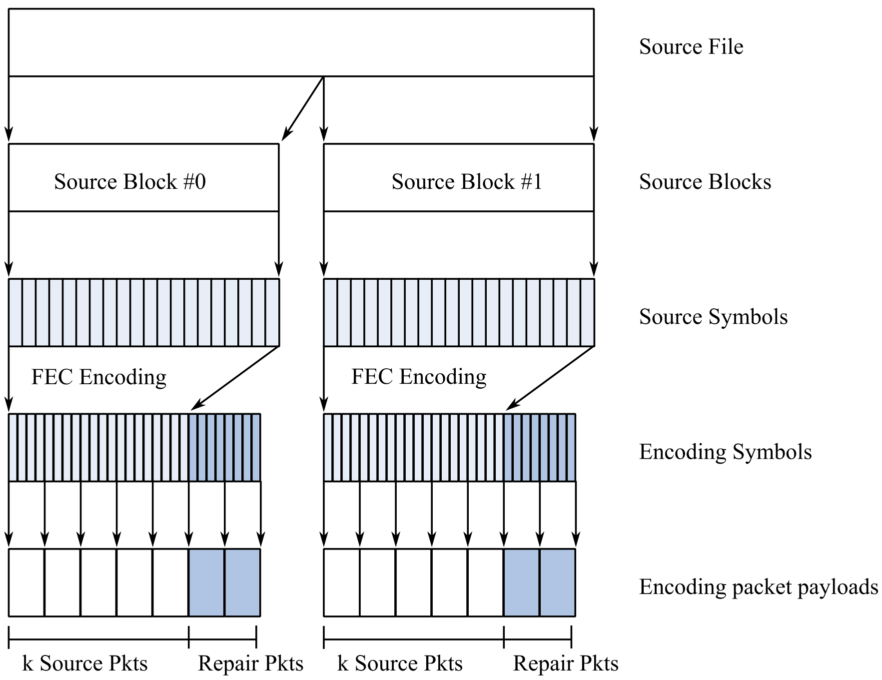


Figure 3.18: File partitioning into encoding packets.

Each source block is then subdivided into source symbols, which are the input symbols for the FEC encoder. The file partitioning algorithm describes the construction of encoding packet payload. An encoding packet is either a source packet with only source symbol or an repair packet, with only repair symbols. One encoding packet may contain one or more encoding symbols. Note, source and repair symbols are not mixed in the same encoding packet.

The Raptor FEC specification put some small limits on the symbol sizes. The maximal symbol size is equal to the maximal IP packet payload size. The payload format for Raptor FEC symbols does not support fragmentation of symbols into multiple IP packets. When using for example IPv4 with packet size of 1460 Byte (MTU of Ethernet), then the maximal symbol size is 1416 Byte (assuming a minimal header size of 44 Byte).

The Raptor FEC specification recommends to have at least 1024 symbols per source block, since the decoding gain is higher with more source symbols. The Raptor FEC payload format allows unique identification of multiple symbols per IP packet, thus smaller files should use smaller symbol sizes than the maximal packet payload size. When large files are transmitted and the maximal supported number of symbols ($K_{max} = 8192$) is reached, the file is divided into multiple source blocks.

A receiver of a Raptor FEC transmission must understand the file partitioning. The receiver must know in particular the symbol size and the number of symbols per source block in order to create the different generator matrices.

All necessary information is either provided in the FLUTE FDT or as an LCT extension header (cf. Section 3.4).

3.6 Conclusions of this Chapter

This section reviews the State-of-the-Art in reliable multicast distribution. A number of different approaches using explicit or negative acknowledgments are described and evaluated. Also approaches using application layer FEC codes are describes and evaluated.

Most sender and receiver initiated protocols generate feedback messages during the actual reception process. Acknowledgments (cf. Section 3.3.1) and negative acknowledgments (cf. Section 3.3.2) are generated on a *per-packet* basis and not on a *per-file* basis. The minimum receiver capabilities of MBMS do not foresee any end-to-end ACKs while receiving an MBMS transmission. DVB-H does not require an integrated 3G terminal in their receivers. Thus, acknowledgments and recovery of missing packets is done after the MBMS transmission has finished. Thus, the published reliable multicast protocols are not applicable for MBMS type of transmissions.

Even if a simultaneous usage of the interactive channel becomes possible (e.g. due to changes of terminal capability requirements), the uplink system must be carefully used. Per-packet feedback might still cause heavy uplink load. *NACK suppression* schemes will not work properly due to the high end-to-end delay. Uplink packets are tunneled up to the network gateway. The radio network can not differentiate between *normal* IP traffic and IP Multicast related traffic.

The FLUTE protocol (Section 3.4) was developed by the IETF RMT working group for different mobile broadcast standards, including MBMS [4], DVB-H [75] and OMA BCAST [93]. All three service layer specifications assume a path-provisioned network, which does not require any congestion control or rate adaptation mechanism. MBMS further mandates the implementation of Digital Fountain's Raptor code for both, the streaming and the download delivery method. It is up to the sender to decide on the usage of FEC. Raptor FEC is optional in DVB-H and OMA-BCAST.

The DVB-H [83] and MBMS [3] carriers offer unidirectional delivery channels. DVB-H may optionally use a 3G interactive channel [8], if desired. MBMS may also use other UMTS bearer services. However, the minimum terminal capabilities (Section 2.3.3) restrict the simultaneous usage of MBMS bearers with other bearer services.

The Raptor FEC code is an evolution of LT codes and both belong into the family of *Fountain Codes*. Fountains codes can produce an arbitrary large number of FEC redundancy symbols from a source block of fixed size. Other FEC codes like LDPC or Reed-Solomon codes have a fixed code rate, thus a fixed number of FEC symbols, which is selected before the encoding process.

The main advantage of Raptor FEC codes compared to LT codes is that

Raptor codes can encode and decode with constant costs when the number of collected encoding symbols is close to the number of source symbols [89]. The number of decoding operations shrinks, when LT codes are only used to recover a constant fraction of input symbols.

The performance of the Raptor FEC code is close to an Ideal FEC code. However, it is not always possible to reconstruct the input source block, when exactly the number of source symbols are received. The receiver requires $k(1+\epsilon)$ encoding symbols to reconstruct the source block at a high probability.

In the following, we have assumed an *Ideal* FEC code, which gives always the best code gain. A Raptor FEC code with a decoding failure probability of $\delta = 0$ at an overhead of $\epsilon = 0$ can be regarded as a Ideal erasure FEC code.

4

Capacity of the MBMS Traffic Channel

This section evaluates the throughput and channel capacity of the MBMS traffic channel for IP packet transmissions. The channel capacity is the description of the error free transmission rate of information over a communication channel.

The theoretic upper bound of the channel capacity for an additive white Gaussian noise (AWGN) channel is described by Shannon-Hartley theorem. It describes the maximal possible *information rate*, excluding any overhead for channel or source coding, which is provided at transmission power S over a communication channel. The communication channels adds white Gaussian noise with power N. Then the channel capacity C is defined as

$$C \text{ [bps]} = B \text{ [Hz]} \log_2 \left(1 + \frac{S}{N}\right), \tag{4.1}$$

where B is the bandwidth of the communication channel. Thus, any type of fragmentation effects from different protocol layers are not considered by this original definition of the channel capacity.

In the following, we would like to compare the effects of using different IP packet sizes on the information bitrate. Therefore, we extend the definition of the channel capacity and include here also the IP packet sizes and the structure of the Radio Link Control protocol in our channel capacity definition. Both have an impact on the information bitrate of the channel.

The Internet can be seen as a *Binary Erasure Channel* (BEC), because IP packets are either dropped by the system or received correctly. The same principle applies for IP communication over mobile links: The underlying radio link protocols ensure at least a reliable detection of corrupted data. IP packets are discarded by the receiving end of the radio link protocol, when not all parts are received correctly.

The Binary Erasure Channels was first defined by Elias [94] and initially regarded as a theoretical channel. When Elias published the BEC model, there was no physical channel with that characteristics. There is no state *nothing received* in radio or wireline communication. The receiver is always receiving something.

With the birth of the Internet and the layered protocol stacks like the OSI Reference Model, the Binary Erasure Channel became important. IP Routers and other IP nodes discard the complete packet, when parts of the packets were incorrectly received. Checksums and sometimes even Cyclic Redundancy Check (CRC) values are part of IP headers and lower layer protocols, which allow the identification of corrupted packets. Also the radio link protocol of mobile networks actively discards incompletely received packets.

On application layer, missing packets are identified using sequence numbers. IP packet payloads are continuously numbered and the receiver monitors the progress of the numbers. TCP's automatic repeat request (ARQ) scheme works with byte counters. RTP and the protocols developed by the IETF RMT working group allow the identification of lost packets based on sequence numbers. Thus, by using protocols from IETF RMT working group, the communication becomes the communication over a binary erasure channel.

In MBMS, IP Packets are fragmented into radio transport blocks at the radio link control layer. A set of radio blocks are interleaved within the duration of one Transmission Time Interval (TTI). Radio block losses within the same TTI are strongly correlated, thus we simplify and handle all radio blocks of the same TTI as a radio transport block. The radio transport blocks are of constant size and several (or parts) IP packets may be fragmented into such a transmission block. Consequently, a single radio block loss may cause one or more IP packet losses. Short IP packets have a higher packet header overhead, but (as we will see in this chapter) have also a lower packet loss probability.

We will show, that the IP packet error rate and thus the capacity of the MBMS links depends on the IP packet sizes. Shorter IP packets lead to an overall higher MBMS traffic channel capacity than larger packets due to fragmentation effects. Therefore, we also introduce the measure of goodput, which describes the fraction of received information bits and sent data bits.

The section is structured as following: First, we describe the principles of the IP packet fragmentation into radio transport blocks. After that, we describe the evaluation environment for measuring the capacity of MBMS traffic channels using a simulator and present the measurement results. We also derive a formula for calculating the MTCH capacity depending on the IP packet size, on the radio block error rate and the radio transmission block size. After that, packet boundary alignment effects between IP packet boundaries and radio transport block boundaries are evaluated. We see, that the capacity can be improved when aligning the IP packets with radio transport blocks where possible.

The goodput is defined as the fraction of received information bits and the sent data bits. The IP packet headers and the channel capacity reduce the information bits. A summary of the main conclusions of this chapter is given in Section 4.5.

4.1 About data losses during MBMS Data Transfer

The Radio Link Control (RLC) layer is responsible to fragment IP packets of variable length into the constant length radio blocks. The RLC Acknowledge Mode is typically used for unicast communication to immediately recover block losses

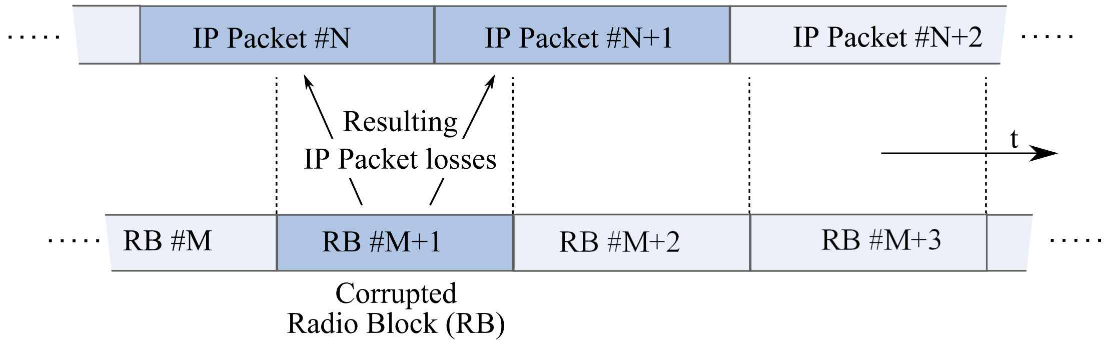


Figure 4.1: Error Propagation from corrupted radio blocks to IP packets losses.

using ARQ. The MBMS Traffic channel can only use the RLC unacknowledged transmission mode, since multiple receivers are targeted with MBMS. MBMS does not employ a multicast ARQ scheme, thus packet losses must be recovered on application layer.

The RLC layer interleaves a fixed number of radio block within one Time Transmission Interval (TTI). 3GPP recommends very long TTIs of 40 ms or 80 ms duration for MBMS, since the turbo codes of the physical layer and the time diversity provide the best correction performance for long TTIs. Depending on the radio bearer transmission capacity, a different amount of data bits can be put into a RLC/MAC transmission frame resulting in different bearer data rates.

Radio blocks get lost depending on the radio conditions. The loss ratio is also called block error rate (BLER). Each block loss may cause one or more IP Packet losses, depending on the IP packet and the radio block length. One radio transmission block may carry fragments of several IP packets. We define a radio transmission block here as the sum of all radio blocks of the same TTI on RLC layer. The block error rates of radio blocks in the same TTI are highly correlated, since the radio blocks are interleaved within the same TTI. Thus, we simplify and apply the radio block error rate (BLER) to the complete TTI.

Figure 4.1 depicts schematically the fragmentation of IP packets into radio transmission blocks and also the propagation of radio transmission block losses to packet losses. In Figure 4.1, the radio transmission block *RB #M+1* is corrupted and causes corruption of *IP Packet #N* and *IP Packet #N+1*, although most parts of both IP packets are received correctly. The IP packet loss rate increases with larger IP packet sizes, since the packets span over several radio blocks, thus, multiple radio blocks have the chance to corrupt the IP packet.

Note, several attempts were done in 3GPP to allow *smart* receivers to re-use the correctly received parts of the IP packets. When the IP packet header is still intact, the receiving application could be identified, so that the correct payload data could be evaluated by the receiving application. But these approaches were

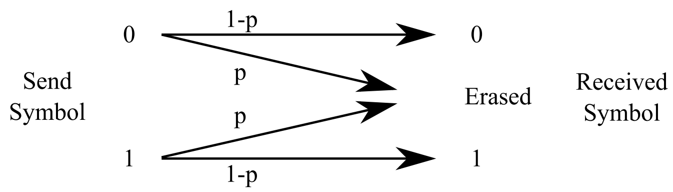


Figure 4.2: Binary Erasure Channel (BEC).

rejected in 3GPP, since receiving applications expect only complete packets. The Checksum in the UDP headers it too weak to identify corruptions reliably. Thus, receiving applications cannot reliably determine, whether or not a part of the payload was correct. Further, complexity of such approaches increases because of backward compatibility issues with deployed receiving applications. Existing UDP receivers may not check the UDP checksum.

In the following, we selected only different constant block error rates to evaluate the impact of the radio block errors to the IP packet error rate.

4.2 Capacity of an MBMS Traffic Channel

The MBMS receiver discards partly received IP packets. Accordingly, the MBMS traffic channel acts like a *Binary Erasure Channel* (BEC). In the following we focus first on a more practical evaluation using a simulator and then more theoretical evaluations.

The binary erasure channel is depicted in Figure 4.2 for single bit inputs (binary input alphabet). Each input value to the binary erasure channel (here *0* and *1*) is either correctly received or is not received by the receiver. In case of IP communication, each input symbol is a complete IP packet, which is either received correctly or dropped by the underlying protocols. The capacity of the binary erasure channel is given by $C = (1-p)$ [94], where p is the packet loss probability. In the following, we develop a formula for the channel capacity of MBMS traffic channels. As described in the previous section (Section 4.1), one or more IP packets may be discarded based on the fragmentation and the block error rates.

Let X be a discrete random variable with

$$P(X = n) = \begin{cases} \frac{1}{l_b} & , \forall n \in \{0, 1, .., (l_b - 1)\} \\ 0 & , \text{else.} \end{cases} \tag{4.2}$$

Then, the probability that the packet of length l_{ps} is fragmented into k(l_{ps})

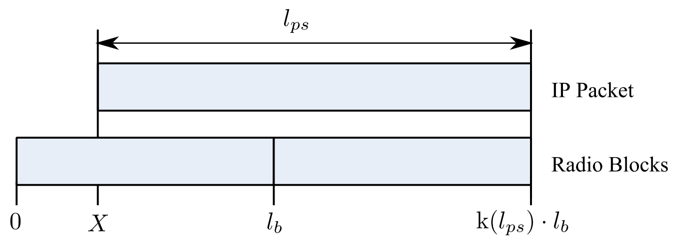


Figure 4.3: Visualization of Eq. (4.3) with Random Variable $X \in \{0, 1, .., (l_b - 1)\}$.

radio blocks, each of length l_b, is

$$P(X + l_{ps} \leq \mathrm{k}(l_{ps}) \cdot l_b) = P(X \leq \mathrm{k}(l_{ps}) \cdot l_b - l_{ps}) \tag{4.3}$$

$$\Rightarrow p_{pkt}(l_{ps}) = \frac{\mathrm{k}(l_{ps}) \cdot l_b - l_{ps} + 1}{l_b} \tag{4.4}$$

with

$$\mathrm{k}(l_{ps}) = \left\lceil \frac{l_{ps}}{l_b} \right\rceil . \tag{4.5}$$

Figure 4.3 visualizes Eq. (4.3). The function $\mathrm{k}(l_{ps})$ (4.5) gives the minimal number of RLC blocks of length l_b, into which an IP packet of length l_{ps} is fragmented. Note that the IP packet is either fragmented into $\mathrm{k}(l_{ps})$ or into $\mathrm{k}(l_{ps}) + 1$ RLC blocks.

$p_{pkt}(l_{ps})$ (4.4) gives the probability, that an IP packet of length l_{ps} is fragmented into $\mathrm{k}(l_{ps})$ fragments. The probability that the IP packet is fragmented into $(\mathrm{k}(l_{ps}) + 1)$ fragments becomes $(1 - p_{pkt}(l_{ps}))$. The resulting IP packet error rate depend on the probability, that an IP packet is spanning multiple RLC blocks. Thus, the IP packet loss rate is a multiple of the RLC block error rate.

The formula is a corrected version of the formula presented by Nortel in TD S4-040120 [95] in 3GPP SA4. The formula allows to calculate the IP packet error rate as a function of the IP packet length l_{ps} and the RLC block error rate p_b. The radio transport block length $l_b$is constant during transmissions. The IP packet error rate increases with increasing packet size. Main reason for this almost linear increase is, that the IP packets are *fragmented* to one or more RLC blocks.

The IP packet error rate as a function of the IP packet length l_{ps} and the radio block error probability p_b is then

$$p_{ip}(l_{ps}, p_b) = 1 - \Big(p_{pkt}(l_{ps}) \cdot (1 - p_b)^{\mathrm{k}(l_{ps})} + \big(1 - p_{pkt}(l_{ps})\big) \cdot (1 - p_b)^{\mathrm{k}(l_{ps}))+1} \Big). \tag{4.6}$$

The capacity of an MBMS traffic channel, in alignment with the capacity of the binary erasure channel, is given as

$$
\begin{aligned}
C_{\mathrm{MTCH}}(l_{ps}, p_b) &= p_{pkt}(l_{ps}) \cdot (1 - p_b)^{\mathrm{k}(l_{ps})} + \\
& \quad \big(1 - p_{pkt}(l_{ps})\big) \cdot (1 - p_b)^{\mathrm{k}(l_{ps})+1}
\end{aligned} \tag{4.7}
$$

It takes IP packets as input alphabet. l_{ps} is the IP packet size and p_b is the radio block error rate (BLER). The radio block size l_b is assumed to be constant during the transmission.

Figure 4.4 depicts the MTCH channel capacity according to Eq. (4.7). MTCH channel capacity of 1 means that all IP packets are successfully received, thus the full Layer 2 bitrate of the channel. Eq. 4.5 causes the division of the curve into segments with different gradients. Eq.(4.7) is not differentiable at multitude of l_b, which are here 640 Byte and 1280 Byte.

The segment borders are visualized with vertical dashed lines in Figures 4.4. Note that 1500 Byte is the maximal transfer unit for Ethernet, which is very often used for IP networks.

Having constant block error rates on a real wireless channel is not realistic, but this simplification helps to understand the influence of the radio link protocol on the IP packet error rate. It becomes obvious, that there is approximately a linear dependency between the block error rate and the packet error rate until about 20% block error rate.

When IP packets are smaller than 640 Byte, two or more packets fit into a single radio block. A single radio block error cause at least two IP packet errors. IP packets larger than 640 Byte but smaller than 1280 Byte span more than one radio block. A single radio block error may cause one or two IP packet errors, which causes a lower curve gradient. IP packet may be fragmented into two and more radio block losses for IP packets of more than 1280 Bytes. A single radio block error may cause one or two IP packet errors. Since the IP packets are fragmented into two or more radio blocks, two consecutive block errors do not cause more damage than a single. That is the reason, why the curves for higher BLERs and IP packet sizes of more than 1280 Byte flattens.

We use a simulation environment for the more complex file delivery scenarios. The algorithm for the IP packet loss pattern is described in Annex A. The radio block size for all simulations is 640 Byte. This corresponds to a 64 kbps MBMS bearer with a 80 ms TTI. The IP packet size varies between 200 Bytes and 1500 Byte depending on different constant block error rates.

4.3 Fragmentation Alignment Effects

IP packets are fragmented into radio transmission blocks. If the packet length is frequently changing like for video transmissions in RTP, the number of IP packets per radio block is also changing. It is not possible to align IP packet boundaries with radio block boundaries without frequently adding of padding octets. Recall that Figure 4.1 depicts the fragmentation of IP packets into radio blocks.

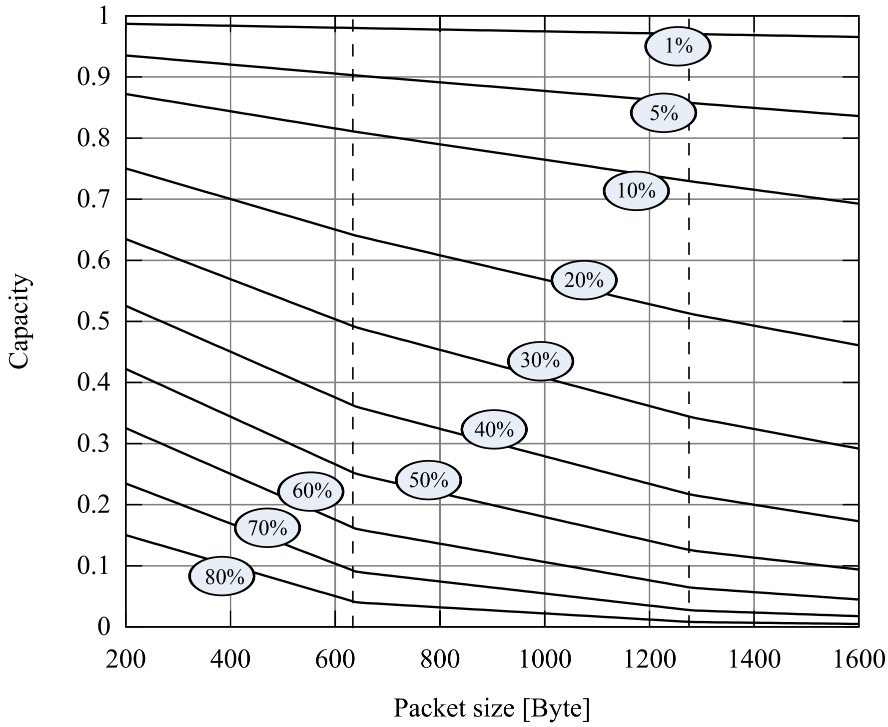


Figure 4.4: MTCH capacity depending on the IP packet size.

A sequence of IP packets of an MBMS download transmission may have the same packet size. With some care and little padding, all packets of an MBMS download session may have the same packet size. In such a situation, it becomes possible to secure alignment of IP packet boundaries with radio transmission block boundaries. The target packet payload length (l_{pp}) is one input parameter to the file partitioning algorithm (cf. Figure 3.18). For Raptor FEC and No-Code FEC, all encoding symbols (source and redundancy) are of equal length. Only the last source symbol can have a different length than the remaining symbols, which needs to be compensated with padding.

The typical FLUTE header (i.e. the LCT and FEC payload ID headers) are also of fixed length for at least one file. There is no imminent need to change the FLUTE header field size from file to file. Some FLUTE packets may carry one or more additional extension headers, for instance the object transmission information (EXT_FTI) header. The EXT_FTI header contains the file partitioning parameters, the used FEC code and required FEC parameters for object recovery. This gives the receiver the opportunity to start object recovery, even when the FLUTE FDT file entry for a specific file is not yet received. It is to be decided by the service provider, where an improved system performance is more important than this *earlier decoding advantage*.

In the following, we evaluate the channel capacity and the effective media rate under the assumption, that all IP packets of an MBMS transmission are

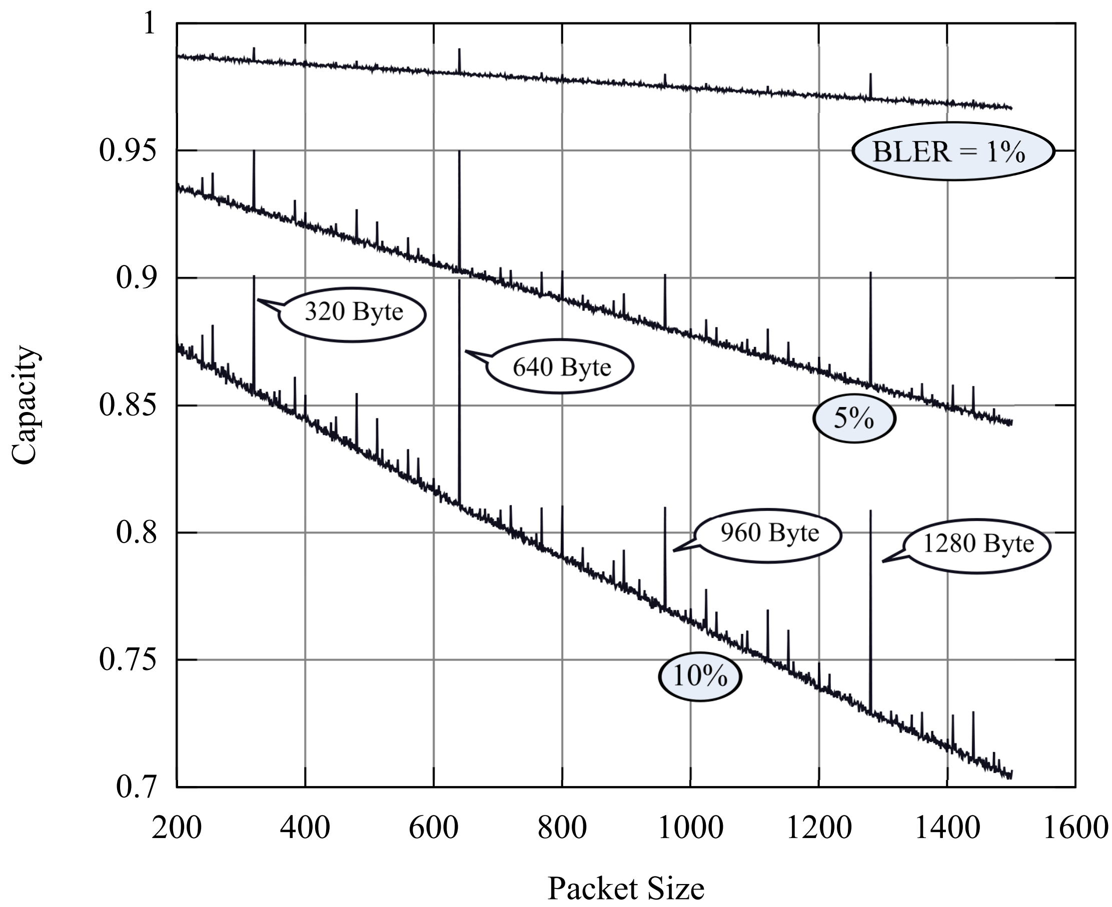


Figure 4.5: MTCH capacity depending on the IP packet size.

of same length. We also assume, that the RLC layer is able to synchronize IP packet boundaries with radio block structure. The first byte of the first IP packet becomes the first byte of a radio transmission block.

Figure 4.4 and Eq. (4.7) do not depict alignment of IP packets to radio blocks. For instance, when the IP packets have the same length as the radio blocks and if the start of an IP packet is aligned with the start of an radio block, then the IP packet error rate and the block error rate are the same. One block error causes exactly one packet loss. Otherwise, the packet error rate is twice the block error rate. A single block error causes exactly two IP packet losses.

All evaluations in this section are carried out with a $l_b = 640$ Byte radio block size (80 ms TTI at 64 kbps). In case of IP packets of $l_{ps} = 640$ Byte length, which are aligned with the radio block boundaries, the packet error rate becomes the same as the radio block error rate. Figure 4.5 depicts the similar evaluation as shown in Figure 4.4, but with radio block aligned IP packets. The very first IP packet and the very first radio block are aligned.

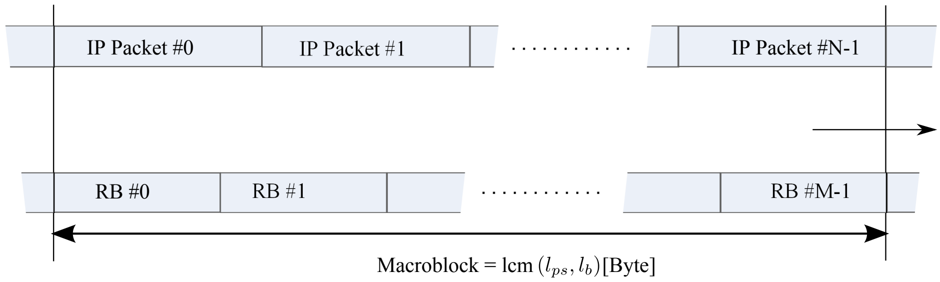


Figure 4.6: Aligned IP Packet and radio block boundaries (Eq. 4.8).

The main visible difference between Figure 4.5 (RLC block boundary aligned packet) and Figure 4.4 (non aligned) are the *spikes* in the graph. There are very strong spikes of the channel capacity at 320 Byte and 640 Byte and some further strong peaks at 960 Byte and 1280 Byte IP packet length. In case of a radio block error rate of 10% (lowest curve, the channel capacity goes up to 90% for 320 Byte and 640 Byte packets. We see a similar behavior for the 960 Byte and the 1280 Byte packets, where the channel capacity goes up to roughly 82%. Other packet sizes like 800 Byte packets lead to a very similar channel capacity.

The reason for the packet error *spikes* in the graph is the alignment of IP packets to radio block boundaries. IP packet are every now and then aligned with the radio block boundaries, leading to a reduction of the packet error rate, thus an improved capacity.

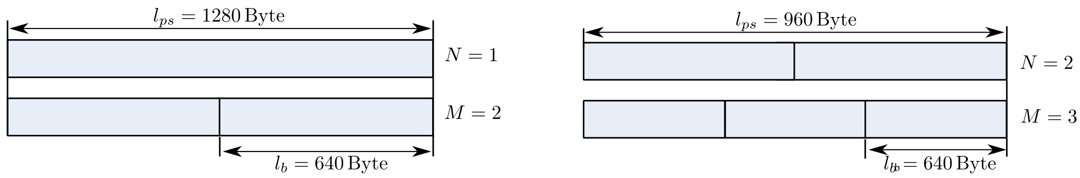
The most significant *spikes* are at 320 Byte and 640 Byte IP packet size. The packet error rate becomes in those two cases equal to the block error rate. If the IP packet size is 640 Byte, every IP packet exactly fits into a single radio block. The packet error rate p almost halves and becomes the same as the radio block error rate ($p_{ip,640} = p_b$). In case of a shift, each IP packet spans two radio blocks and the IP packet error rate is twice the block error rate.

In case 320 Byte IP packets, two IP packets fit into a single radio block and the resulting IP packet rate is also the same as the radio BLER. One erroneous radio block may cause two IP packet losses. But the successful reception of an radio block give also two successfully received IP packets. ($p_{ip,320} = p_b$)

The reason, why Eq. 4.6 gives only the upper limit of the IP packet error rate is that it does not consider the possible positive impact to two consecutive radio block error: A second radio block loss does not increase the IP packet error rate, since the IP packet was anyhow corrupted due to a first radio block error. Figure 4.6 shows an example combination of IP packet length l_{ps} and radio block size l_b, assuming the alignment of the IP packets with the radio block boundary.

$$N \cdot l_{ps} = M \cdot l_b, \text{ with } N = \frac{\text{lcm}(l_{ps}, l_b)}{l_{ps}} \text{ and } M = \frac{\text{lcm}(l_{ps}, l_b)}{l_b} \tag{4.8}$$

Figure 4.6 shows the block alignment of N IP packets and M radio blocks.



(a) 1280 Byte packets (b) 960 Byte packets

Figure 4.7: Alignment for 960 Byte and 1280 Byte packets.

N and M are chosen in such a way, that $N \cdot l_{ps} = M \cdot l_b$ (cf. Eq. 4.8) is fulfilled. In other words: The size of a block in Byte is the least common multiple (lcm) of l_b and l_{ps}. Every Nth IP packet starts aligned with an radio block boundary. The loss events repeat on a per-macroblock basis. The macroblock losses are statistically independent from each other.

Let the random variable X denote the number of erroneous received packets in a macroblock of size N. The average ratio of erroneous received packets is given by

$$p_{ip,s} = \mathrm{E}\left(\frac{X}{N}\right) = \frac{1}{N}\sum_{x=1}^{N} x P_s(X = x). \tag{4.9}$$

Figure 4.7a depicts the alignment of 1280 Byte IP packets with 640 Byte radio transmission blocks. $N = 1$ IP packet fits into $M = 2$ radio transmission blocks of $l_b = 640$ byte size. Figure 4.7b depicts the same for 960 Byte IP packets. Here, $N = 2$ IP packets fit into $M = 3$ radio transmission blocks of 640 Byte size. Corruption of the outer two radio blocks cause only a single IP packet loss. Corruption of the middle radio block lead to corruption and loss of both IP packets.

$$\begin{aligned} P_{1280}(X = 1) &= 2 \cdot p_b(1 - p_b) + p_b^2 && (4.10)\\ &= 2\, p_b(1 - p_b) + p_b^2 \\ &= 2p_b - p_b^2 && (4.11)\end{aligned}$$

Eq. (4.10) describes the probability that the one IP packet (of the sequence of one) is corrupted. This may happen, when either one of the two radio blocks are corrupted $(2 \cdot p_b(1 - p_b))$ or when both radio blocks (p_b^2) are corrupted. Eq. (4.11) provides the IP packet error rate $p_{ip,1280}$ when 1280 Byte IP packets are used.

$$\begin{aligned} P_{960}(X = 1) &= 2 \cdot p_b(1 - p_b)^2 && (4.12)\\ P_{960}(X = 2) &= p_b(1 - p_b)^2 + 3 \cdot p_b^2(1 - p_b) + p_b^3 && (4.13)\\ p_{ip,960} &= \frac{1}{2}\Big(P_{960}(X = 1) + 2 \cdot P_{960}(X = 2)\Big) \\ p_{ip,960} &= 2p_b - p_b^2 && (4.14)\end{aligned}$$

l_{ps} [Byte]	Macroblock Size (Byte)	$N = \frac{\mathrm{lcm}(l_{ps},l_b)}{l_{ps}}$	$M = \frac{\mathrm{lcm}(l_{ps},l_b)}{l_b}$	Packer Error Rate (Aligned)
640	640	1	1	p_b
320	640	2	1	p_b
1280	1280	1	2	$2p_b - p_b^2$
960	1920	2	3	$2p_b - p_b^2$
800	3200	4	5	

Table 4.1: Different Frames for radio block size of $l_b = 640$ Byte.

Eq. (4.12) and Eq. (4.13) describe the probability that one or two (respectively) IP packets are lost. As visible from Figure 4.7b, one IP packet is lost when either the left or the right radio block is corrupted, thus $(2 \cdot p_b(1-p_b)^2)$. Both IP packet are lost when either the middle block is corrupted or any two of the three blocks are corrupted or all three blocks are corrupted.

Table 4.1 summarizes the packet error rates for the strongest spikes, which are at packet sizes 320 Byte, 640 Byte, 960 Byte and 1280 Bytes. The significance of the spikes decrease with greatest common divisor (gcd) of l_b and l_{ps}. Note the relation $\mathrm{gcd}(l_{ps}, l_b) = \frac{l_{ps} l_b}{\mathrm{lcm}(l_{ps},l_b)}$ between least common multiple (lcm) and greatest common divisor (gcd).

The curves in Figure 4.5 have different gradients for different block error rates. Therefore, also the gain in packet error loss reduction might be different for different block error rates.

Figure 4.8 evaluates the capacity gain, due to aligning the IP packets to radio block boundaries. The *capacity gain* is the ratio of the channel capacity of radio block aligned packets and not aligned packets. It becomes obvious again, that certain packet sizes lead to a better IP packet loss rate, if the packet size does not vary. It can be observed that the capacity gain depends on the radio block error rate.

IP packets, which have the same size as the radio transmission blocks (here $l_{ps} = l_b = 640$ Byte show the highest capacity gain. The curves of packet size $P = 640$ Byte and $P = 320$ Byte have both the maximum at a BLER of $p_b = 0.5$ and a similar shape. The curves with packet size $P = 1280$ Byte and $P = 960$ Byte have also a similar shape, but different maximal gain points. It must be noted that the curves with the same maximum in packet error rate reduction have also the same greatest common divisor of the frame (cf. Figure 4.6).

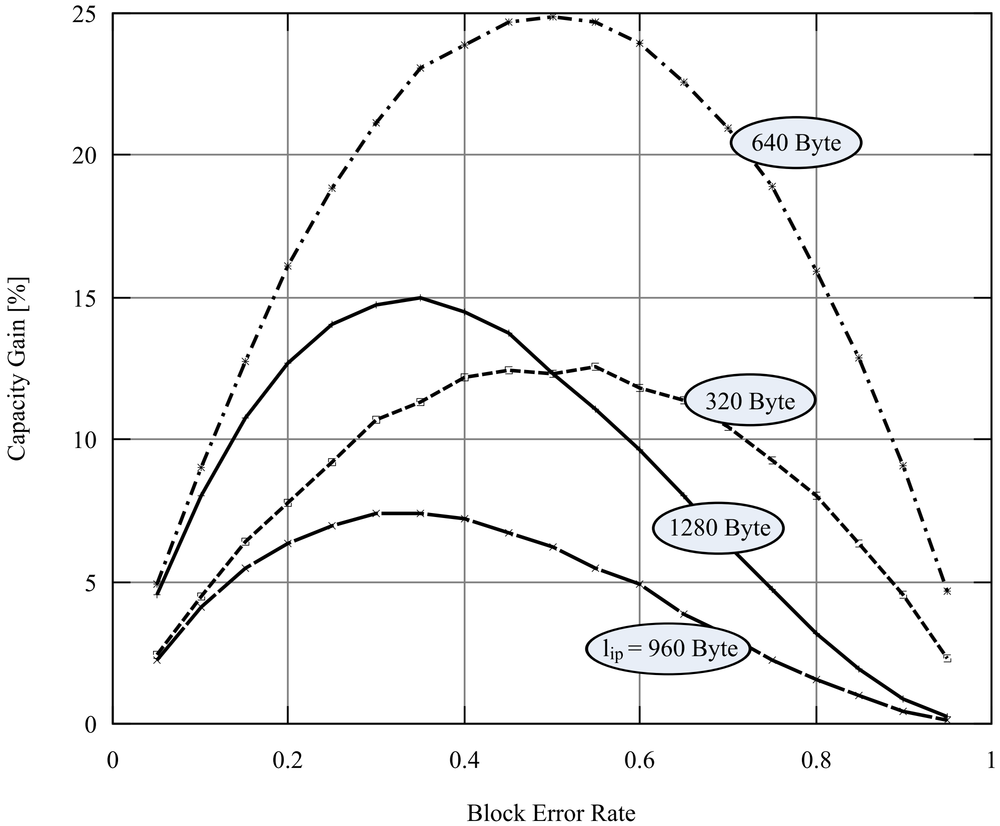


Figure 4.8: Capacity gain due to aligning IP packets to radio blocks.

4.4 Goodput Evaluation

One conclusion from the previous section is that the packet error rate depend on the radio block losses (BLER) and also on the IP packet size. Radio blocks get lost depending on the radio conditions. Each block loss may cause one or more IP packet losses, since an IP packet may be fragmented into one or more RLC blocks. The IP Packet fragmentation is described in 4.1 and the resulting error rate is discussed in 4.2. The resulting IP packet loss rate depends on the RLC block size and also on the IP packet size (l_{ps}). The packet loss rate becomes lower for shorter IP packets.

But, the shorter the packets the higher the packet header overhead. Eq. 4.15 shows the packet header overhead (OH_{hdr}). The packet overhead is an important aspect in the overall system utilization evaluation since it has a direct impact on the the resource efficiency. Since the packet header (l_{hdr}) is constant, the packet header overhead increases with decreasing packet size (l_{ps}), thus the overhead increases with decreasing packet payload length (l_{pp}).

$$OH_{hdr} = \frac{l_{hdr}}{(l_{ps} - l_{hdr})} = \frac{l_{hdr}}{l_{pp}} \tag{4.15}$$

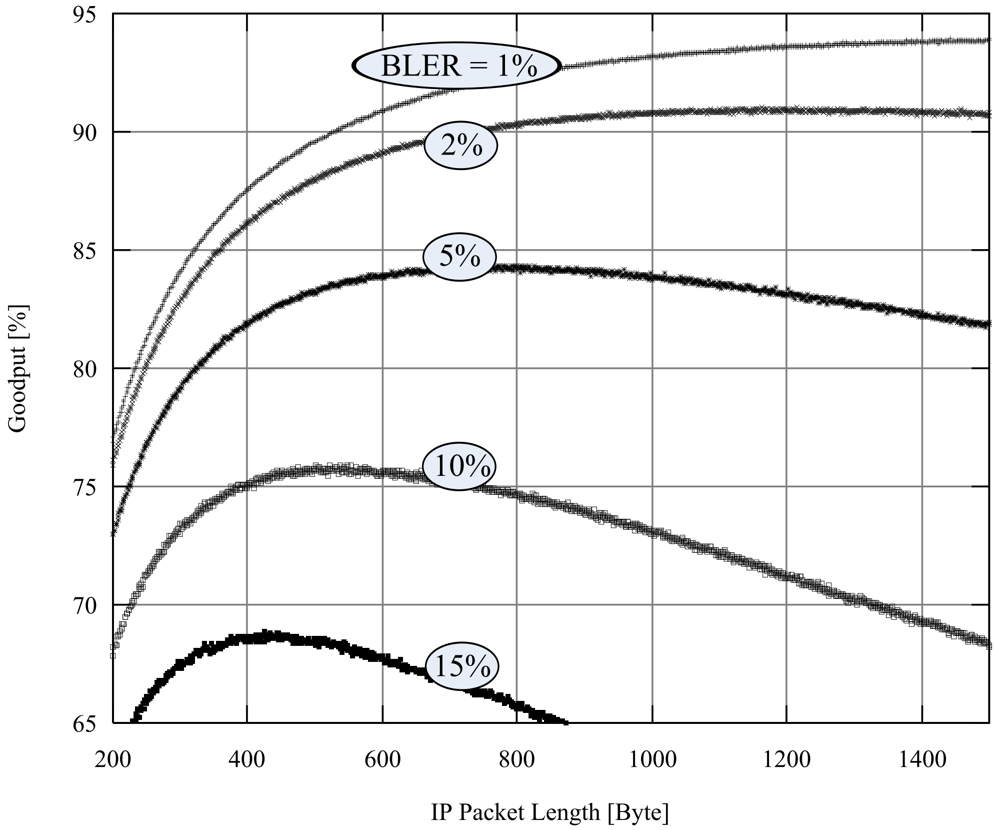


Figure 4.9: Goodput (Effective Media Rate) for different radio block error rates.

If we consider IP version 4 and a minimal FLUTE header of 16 Byte, then the header overhead is $OH_{hdr} = 44\ Byte/l_{pp}$ (IPv4: 20 Byte, UDP: 8 Byte and FLUTE: 16 Byte). Note, the FLUTE header is a variable length header as described in Section 3.4. It has the minimal header size of 16 Byte. If the service requires larger Transport Object Identifier (TOI) or Transport Session Identifiers (TSI), then the packet header may increase. The maximal TOI field size is 14 Byte.

The *Goodput*

$$Goodput(p_{ip}, l_{ps}) = (1 - p_{ip}) \cdot \frac{l_{ps} - 44\ Byte}{l_{ps}} \tag{4.16}$$

is defined as the fraction of received information bits and sent data bits and is function of the IP packet size (l_{ps}) and the IP packet error rate (p_{ip}). The left part of the product Eq. (4.16) $(1 - p_{ip})$ describes the amount of packets, which are received on the remote side. The right side of the product is actually the packet header overhead (assuming here a minimal header length of 44 Byte).

Figure 4.9 shows the resulting goodput over the IP packet length parametrized with the radio block error rate (BLER). As expected, Figure 4.9 shows

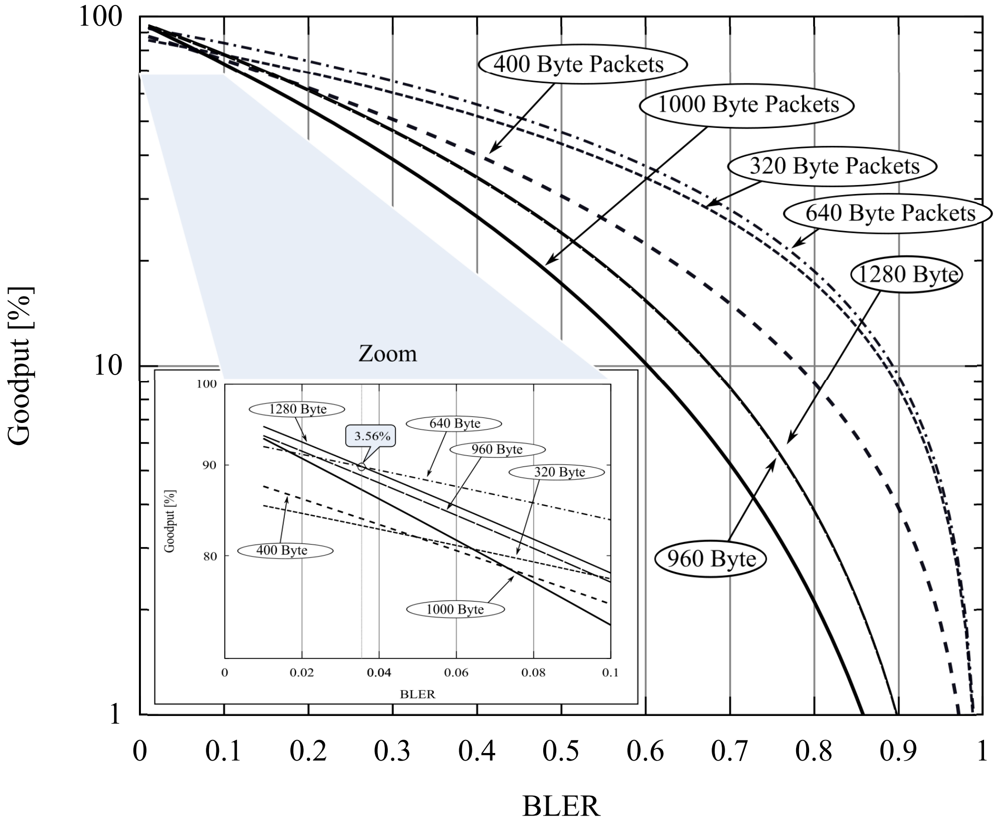


Figure 4.10: Goodput over BLER for different IP Packet sizes (incl. radio block boundary alignment).

the goodput generally decreasing with increasing BLER, regardless of the IP packet length. Figure 4.9 also shows the goodput decreasing with IP packet length for short IP packets for all block loss rates. The packet overhead as defined in eq. 4.15 becomes the dominating factor. The goodput is also decreasing for large IP packets under increasing block error rates (for $BLER > 2\%$). The reason here is that a single IP packet is fragmented into more than one radio blocks. Thus, the IP packet is only received correctly, if several radio blocks are also received correctly. Conclusion out of this graph is, that a lower IP packet length should be used in case of higher block error rates.

One can observe, that the goodput shows a clear maximum for higher radio block error rates. For low BLER rates, large packets are still better to use, although the IP packet loss rate is higher.

We have evaluated packet alignment effects in the previous section and seen a gain is some cases and for certain IP packet sizes. Thus, we expect also a goodput increase as result of packet error rate reduction due to the alignment effects. The goodput for block aligned packets over the BLER for different IP packet sizes is shown in Figure 4.10 for the earlier evaluated packet sizes of 320 Byte, 640 Byte, 960 Byte and 1280 Byte (all aligned with radio block boundaries). Further, 1000 Byte and 400 Byte IP packet sizes are added to show the differences to random chosen IP packet sizes, which show no alignment gains. The BLER range from 0% to 10% shows a different behavior than the range between 10% and 100%. Thus, when operating on bearers with low radio block error rates (e.g. due to automatic repeat request (ARQ), different packet sizes may be selected. The range between 0% and 10% is magnified to increase clarity.

For the large BLER range between 10% and 100% BLER, IP packets of length 640 Byte show the highest goodput than other IP packets sizes. Of course, 640 Byte packets, which are aligned to the radio block boundaries have also the highest alignment gain compared to other IP packet sizes. IP packets of 320 Byte sizes show at 50% BLER a goodput reduction of 7.4% compared to 640 Byte packets.

Interestingly, IP packets of 400 Byte length have a higher goodput than aligned packets with 960 Byte and 1280 Byte. At 50% BLER, the 400 Byte packets show 22.1% higher goodput than 960 Byte packets. However, the 400 Byte packets show a 34.2% lower goodput at 50% BLER than 640 Byte packets. Thus, alignment of IP Packet with the radio block boundaries are not always better than other IP packet sizes.

The range between 0% and 10% BLER is different. The goodput losses due to the packet header overhead is more significant than actual packet losses in this BLER range. Larger packet sizes of 1280 Byte (aligned) and even 960 Byte (aligned) packets show a higher goodput than the 640 Byte packet. The 1280 Byte curve crosses the 640 Byte already at a BLER of 3.56%. Even the not -aligned 1000 Byte packets show a higher goodput than the 640 Byte packets in case of really low BLERs below 2%.

4.5 Conclusion of this chapter

In this chapter, we have evaluated the channel capacity of the MBMS traffic channels for different IP packets. The MBMS Traffic Channel can be seen as a *Binary Erasure Channel* (BEC) like the Internet, since packets are either lost or correctly received. The system ensures that only error free packets are received.

The radio link control protocol (RLC) fragments the variable length IP packets into constant size radio transmission blocks. The duration of a radio transmission block is one Time Transmission Interval (TTI). Radio blocks sent in the same transmission time interval show a strongly correlated loss probability so that we use the simplification, that all radio blocks of the same time transmission interval (thus the radio transmission block) are lost with the loss probability of a radio block (BLER).

IP packets are fragmented into radio transport blocks. The IP packet size has two effects: First, the larger the packet size the smaller is the relative packet header overhead and therefore the larger the goodput. Secondly, larger packets are more vulnerable to radio block errors, because the likelihood that one IP packet is fragmented into several blocks increases. This has the effect of decreasing the goodput because of increased packet error rate. Without AL-FEC, we found that for $BLER < 5\%$ the goodput-optimum IP packet size is larger than the maximum size of IP packets used in the Internet today, so no further restriction of the size is necessary. For larger BLER, the optimum size decreases with increasing BLER and is e.g. around 500 byte for $BLER = 10\%$.

MBMS download uses FLUTE, which allows compared to video transmission with RTP the generation of IP packets with constant size. An input file is partitioned into FLUTE packets of same size. With some carefully selected FLUTE header options, it is possible to have one and even more file partitioned into exactly the same IP packet sizes.

When aligning the first IP packet with the first radio block boundary, we see an improvement in the channel capacity. The highest gain is visible of course, when all IP packets are aligned to radio block boundaries. This is only the case, when the packet size (l_{ps}) is equal to the radio block size (l_b), which is 640 Byte in our evaluations. Here we found a capacity improvement of 25% compared to not aligned packets of same size.

We also see capacity gains for other IP packet sizes, like 960 Byte packets, which do not exactly fit into a single radio block. Here we see that the error rate decreases per macroblock. In case of 960 Byte IP packets, the macroblock size is 1920 Byte, where two IP packets fit into three radio blocks.

The channel capacity and the packet error rate does not consider the IP packet header overhead. The header overhead is larger for shorter IP packets, thus, the effective media rate is lower for shorter IP packets. Therefore, we have introduced a *goodput* definition, which considers the information loss due to the packet overhead. We see in particular for block error rates larger than 10% clear benefit for smaller packet sizes, even when considering the header overhead. The improved capacity is more important than the losses due to headers.

5

Resource usage for reliable file delivery on the MTCH

In the previous chapter, we have evaluated the MTCH channel capacity and the error propagation from radio block losses to packet losses of the MBMS Traffic Channel (MTCH). We develop a basic understanding of the MTCH transmission characteristics using different IP packet sizes with constant radio block error rates.

This chapter evaluates how to maximize the system efficiency for reliable file delivery using the MBMS MTCH channel. There are several challenges with *broadcasting* of files to large groups. Existing and well-understood content delivery protocols such as HTTP cannot operate on MBMS, since TCP is used as transmission protocol. Received files must be generally free of transmission errors, because most file formats cannot handle missing or corrupted sectors. The entire file might not be usable even if only small parts are corrupted.

The MBMS service layer defines several mechanisms for reliable file delivery over MBMS, namely Application Layer Forward Error Correction (AL-FEC) and post delivery file repair. When the MBMS traffic channel is used during the first phase of the file distribution, the tuning and configuration of the reliability mechanisms depend on the used transmit power and the selected IP packet size. The transmit power influences together with the selected IP packet size of the file transmission the IP packet error rate of the transmission.

We evaluate two different optimization targets for file delivery over MBMS: One target is to maximize the number of receivers with a transmission error free file after the first transmission phase at constant resource usage. The second evaluation target is to balance the resource usage for the MTCH transmission with the resource needs for the file repair in order to increase the system efficiency of the file distribution of a certain size to all receivers. It is possible to trade the transmit power with the amount of AL-FEC redundancy at the same load for the file repair service.

Each receiver in a mobile broadcast scenario has its own radio Block Error Rate (BLER), depending on the individual radio conditions and distance to the base station. In this chapter, we model the radio reception conditions of the MBMS Traffic Channel (MTCH) for different transmit power settings. The block error rates of a large number of different locations in the cells are used to evaluate the reception of a 1 MByte file over MTCH.

The Raptor FEC is selected for MBMS as Application Layer FEC (AL-FEC). Raptor FEC recovers the source block from any $k(1+\epsilon)$ received encoding symbol with a certain decode failure probability δ. The overhead ϵ decreases for a given δ with increasing number of source symbols k. The performance of Raptor FEC is close to an Ideal AL-FEC, when the source block is partitioned into a large number of source symbols. We used in the following an Ideal FEC code for the evaluation.

Parts of this chapter has been presented at the IEEE International Symposium on Broadband Multimedia Systems and Broadcasting 2009 in Bilbao [96].

This chapter is structured as following: First, we describe the MBMS radio network model (Section 5.1.1) and the evaluation methodology (Section 5.1.2). Then, we evaluate the MBMS PTM transmission performance on MTCH using different IP packet sizes in Section 5.2. We base our evaluations on the knowledge about the error propagation from the radio block error rate (BLER) to IP packet error rate (PER) (Chapter 4). We discuss balancing the needed resources for the MTCH transmission with the needed resources for File Repair to optimize the system efficiency in Section 5.3. The gains from aligning the IP packets to radio block boundaries are evaluated separately in Section 5.4. It may not always be possible to ensure such a boundary alignment. The conclusions of this chapter are presented in Section 5.6.

5.1 Evaluation Methodology and Simulation Environment

We consider here the goal of minimizing the total transmission energy required to deliver the file error free to all users. Figure 5.1 shows a dependency chart of the different configuration parameters and evaluation targets.

The Block Error Rate (BLER) depends on the used output power of the MBMS Transport Channel (MTCH). The radio link control protocol (RLC) fragments the IP packets into constant size blocks, which determines the relation between BLER and the Packet Error Rate (PER). Short IP packets show a higher goodput than larger IP packets in case of high BLER as shown in Section 4.4. The PER and the amount of added AL-FEC redundancy (FEC Overhead) leads to the Missing Data Ratio (MDR) and the Satisfied User Ratio (SUR). The ptp file repair is used to fetch missing data. The load on the file repair (FR) server is limited by the available FR Server Capacity and the maximal FR Service Duration. Lohmar *et al.* [97] show, that the file repair server may also become a bottleneck in case of large audiences, depending on the file size. In order to optimize the system, we need to balance transmission energy of the MTCH transmissions with the transmission energy of the file repair. The file repair system must be dimensioned to provide the required capacity.

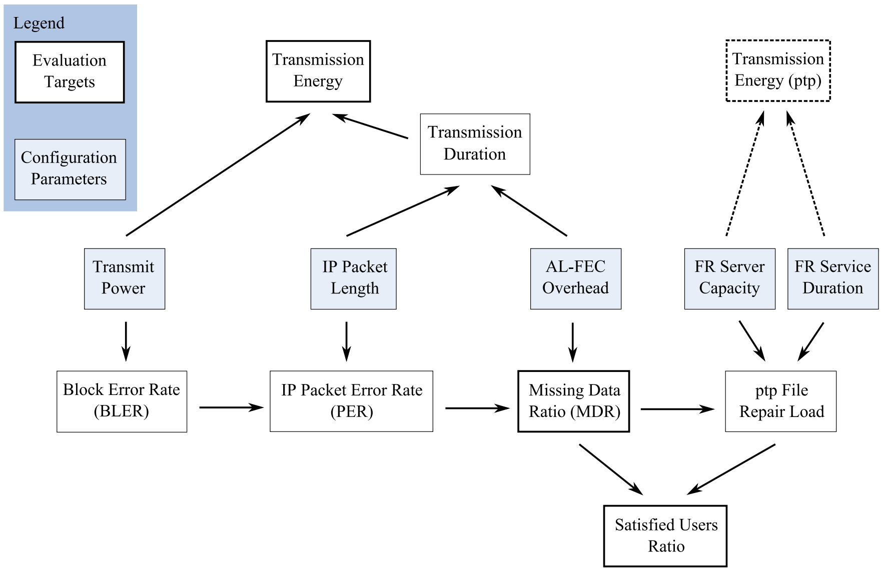


Figure 5.1: Dependency Chart of configuration parameters and evaluation targets.

5.1.1 MBMS Radio Network Model

To evaluate the characteristics of file delivery over the MTCH channel under more realistic conditions, a trace of block error probabilities of the wireless channel is generated for each receiver. 3GPP has defined a number of reference scenarios in TS 30.03 [98] to allow objective comparisons of different radio scenarios. We have chosen the vehicular test environment for the MBMS evaluations, which is characterized by large cells and higher transmit power. Using large cell sizes increases the probability to serve larger receiver populations in a single cell. It also decreases handover probability. The log-normal shadow fading with 10 dB standard deviation is appropriate for urban areas. The according path loss model for vehicular environments for a carrier frequency of $f_c = 2000\,\mathrm{MHz}$ and a base station antenna height of $h_b = 15\,\mathrm{m}$ above the average root top is give in by

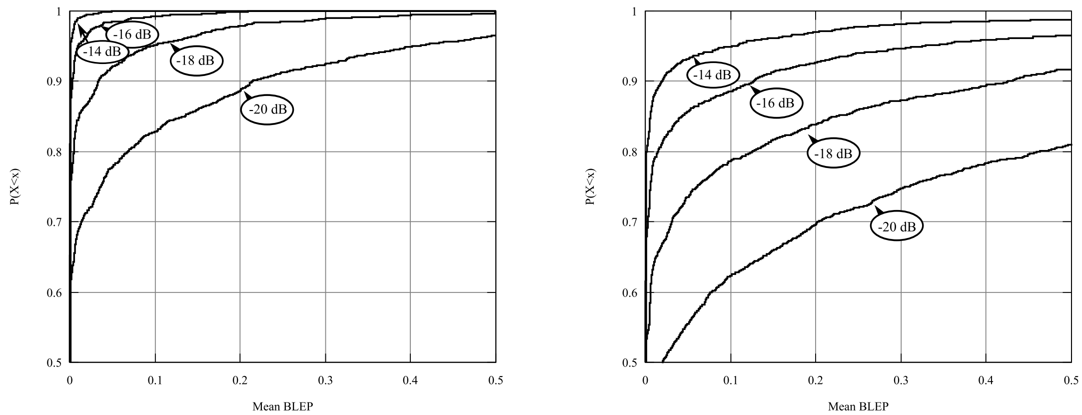
$$L\ [\mathrm{dB}] = 128.1 + 37.6\ \log_{10}(R\ [\mathrm{km}]). \tag{5.1}$$

Eq. (5.1) describes the path loss in dB for urban and also sub-urban test environments, where the buildings are of nearly uniform height. R is the distance between the base station and the receiver in km.

The modeled radio network consists of 21 base stations, with 3 sectors each with a fictive wrap around to avoid simulation area border effects. 1500 receivers are uniformly distributed inside of the coverage area. The receivers at the cell borders use soft-combining to decrease the error probability, therefore also the surrounding cells contribute to the reception. The cell radius is either 500 m for service data rates above 144 kbps or 2000 m otherwise as described in

Parameter	Value
MBMS MTCH data rate	64kbps
CPICH Ec/Ior	-10 dB (10%)
Channel	Vehicular A, $3km/h$
Cell Radius	$r = 500m$ and $r = 2000m$
Propagation model	$L = 128.1 + 37.6 \cdot \log_{10}(R)$
Transmission Time Interval (TTI)	80 ms
Maximal Transmit Power	20 W
Receivers per cell	1500
Trace Capture Duration	320 sec or 4000 TTIs
Sectors per cell	3
Soft-Combining	Yes, up to 3 radio links
Mobility	No, stationary scenario

Table 5.1: Parameters of the Radio Network Model.



(a) Cell Radius 500 m (b) Cell Radius 2000 m

Figure 5.2: Cumulative Distribution of Mean BLEP (-14 dB to -20 dB P/P_{max}).

3GPP TS 30.03 [98]. We have modeled both deployment scenarios although only 64 kbps service data rates are used for the MBMS Download evaluations. It is expected, that cell sizes in urban areas are much smaller than 2000 m due to other mobile broadband services. Operators might have a more dense network to provide sufficient voice and unicast data capacity.

Figures 5.2a and 5.2b depict Cumulative Distribution Function (CDF) of the mean Block Error Probability (BLEP) for deployment scenarios with cell radius of $r = 500$ m and $r = 2000$ m. Only the range between 0% and 50% mean BLEP is depicted. A block error probability higher than 50% will definitely not lead to energy optimal transmission and overload the FR server due to to many required retransmission. It can be observed that an approx 3.5 dB higher transmit power is needed for the $r = 2000$ m deployment to achieve a similar BLEP distribution.

P/P_{max}	Mean BLEP		Estimated Coverage (3.56% target BLEP)	
	$r = 500\,m$	$r = 2000\,m$	$r = 500\,m$	$r = 2000\,m$
−14 dB	0.10%	1.70%	99.6%	92%
−15 dB	0.18%	2.80%	98.8%	88.9%
−16 dB	0.38%	4.47%	97.5%	83%
−17 dB	0.81%	6.80%	93.5%	77.1%
−18 dB	1.73%	10.00%	89.0%	67.9%
−19 dB	3.51%	14.51%	80.6%	59.5%
−20 dB	6.65%	19.99%	71.9%	49.9%

Table 5.2: Expected Mean Block Error Probabilities.

Comparing Figures 5.2a and 5.2b with the Figure 4.10 from the previous chapter, there seems to be no single optimal IP packet size: In case of urban areas with cell radius of $R = 500\,\mathrm{m}$, more than 95% of the receivers have 10% BLER and lower. Thus, the selection of long IP packet sizes with low packet header goodput losses seems to be correct. Of course, receivers with higher BLERs have a reduced goodput and receive also less FEC correction packets.

Table 5.2 lists the mean BLEP values for the different transmit power configurations and the different cell sizes. The table also contains the estimated coverage for a 3.56% BLEP target. We highlight the 3.56% BLEP target because it is the crossing of the goodput of 1280 Byte packets with 640 Byte packets as shown in Figure 4.10. In case of a transmit power of −15 dB, 98.8% of the receivers in the cell of radius $r = 500\,\mathrm{m}$ experience a BLEP of less than 3% and 88.9% in cells with $r = 2000\,\mathrm{m}$ radius.

The radio network evaluation environment is used to create sequences of radio block error probabilities (BLEP) for each simulated user in the cell using different transmit power configurations. The sequences are stored in trace files, which are then used as input to a protocol simulator. This second simulation environment models the transmissions of files and different levels of FEC protection. The link between sender and receiver implements the RLC framing of IP packets into radio blocks as described in Section 4.1. The radio blocks are lost according to the block error probability values from the BLEP trace file. Since the radio network simulator produces a block error *probability* as output, the file transmissions must be repeated several times to get a statistical result.

There is only a finite number of BLEP values in the trace file. The usage of the BLEP values is realized considering a ring buffer. As soon as the last value of the trace file is read, the BLEP value usage continues with a random value in the first part of the trace file. After that, the BLEP value are sequentially consumed from the ring buffer again.

5.1.2 Evaluation Methodology for file delivery over MTCH

The MBMS download method is a combination of the FLUTE data delivery and the file repair procedures. The FLUTE data delivery uses the MBMS bearer and is intended to serve the majority of receivers with sufficient data. The ptp file repair is intended to serve the still unsatisfied receivers with the remaining data. A number of different parameters influence the performance of the MBMS Download delivery method. All steps in this delivery sequence must be considered together for the parameter evaluation and optimization. It is of course possible to optimize the MBMS download procedure for different target measures. One is of course the duration of the file delivery and another one the radio resource consumption for the delivery. If the target is to serve all members of the MBMS download group, then the file repair procedure must be considered as well.

This chapter investigates the *radio resource usage* as one prime optimization target. MBMS download transmissions happens in the background. The end-user is not involved in the actual download process, only the terminal of the end-user. The terminal is informed about an eminent transmission and starts the reception. The end-user may become aware about the MBMS reception, when the transmission is completely received. This means, that the actual transmission duration is of lower importance.

The MBMS transmission sequence starts with the transmission of the source packets with one or more source symbols using the MBMS bearer. A set of repair packets containing one or more FEC repair symbols may be transmitted after the source packets, if the MBMS transmission is FEC protected. The loss rate depends on network deployment parameters such as the site-to-site distance, but also on the transmission power and the general radio conditions. As shown in the previous chapter, the loss rate also depend on the packet sizes. Smaller packets have a lower loss rate but on the expense of a higher transmission overhead. The following two equations are needed to compare the different parameter configurations. The equations describe the overhead due to FEC (OH_{fec}) and the total transmission overhead (OH_{tx}).

$$OH_{fec} = \frac{N_{fec} \cdot l_{pp}}{l_{file}} \tag{5.2}$$

Eq. (5.2) describes the FEC overhead (OH_{fec}) relative to source file length. Typically, the repair packets containing additional FEC packets are sent after the source packet, so that receivers in very good reception conditions that do not need all repair packets can stop receiving earlier. In case of missing data, the receiver uses a FEC packet to repair any other source packet (in case of an *ideal* FEC code).

N_{fec} is the number of FEC packets, which are sent additionally to the source packets. It is assumed that each packet is always completely filled up with repair symbols. l_{pp} is the length of the packet payload in Byte. A packet payload may contain one or more FEC encoding symbols. l_{file} is the file length in Byte.

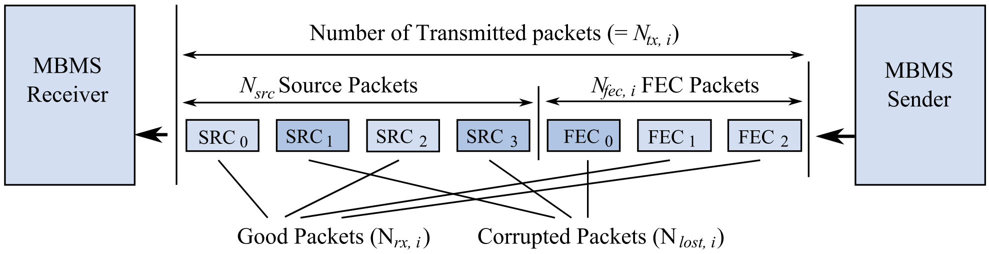


Figure 5.3: Client Reception Procedure (Simplified).

$$OH_{tx} = \frac{1}{l_{file}}\Big(\sum_{n=1}^{N_{src}}(l_{src,n} - l_{pp}) + \sum_{m=1}^{N_{fec}} l_{fec,m}\Big) \tag{5.3}$$

with

$$N_{src} = \left\lceil \frac{l_{file}}{l_{pp}} \right\rceil \tag{5.4}$$

Eq. (5.3) defines the transmission overhead OH_{tx} in its general form. The transmission overhead considers beside the FEC overhead of the additional repair packets also the packet header overhead. Transmission with smaller IP packets has the advantage of a lower packet loss rate by expense of a higher transmission overhead (cf. Section 4). N_{fec} is the number of FEC packets, N_{src} is the number of source packets with packet payload length l_{pp} (according to Eq. (5.4)). $l_{src,n}$ is the length of the *nth* source packet. The packet payload l_{pp} is the same for all source packets. Only the packet headers length may differ from packet to packet depending on the used FLUTE header extensions. $l_{fec,m}$ is the length of the *mth* FEC packet (header and FEC payload). The payload of all FEC packets is of the same size as the source packets. Remember that the encoding symbols size of Raptor and other FEC codes is the same. Thus, the left summand of Eq. (5.3) is the transmission overhead of the source packet headers and the right summand is the overhead caused by the full repair packets.

Eq. (5.5) is a simplification of Eq.(5.3) assuming that all source packet headers are of length l_{hdr} and all FEC packets (header and payload) are all of length l_{ip}.

$$OH_{tx} = \frac{N_{src} \cdot l_{hdr} + N_{fec} \cdot l_{ip}}{l_{file}} \tag{5.5}$$

For IPv4 and the minimal FLUTE packet length l_{hdr} is 44 Byte. The packet header length may be increased by adding FLUTE extension headers.

Figure 5.3 depicts the principle evaluation of the FEC transmission. It also contains the simplified client reception procedure as pseudo code. For each client in the cell (each with different reception conditions) and each iteration i, the client receives $N_{rx,i}$ packets of data until it is possible for the client to reconstruct the

Algorithm 5.1 Reception Procedure

done := **false**
while not *done* **do**
  call $waitForNextPacket()$
  $N_{rx} = N_{rx} + 1$
  if $(N_{rx} \geq N_{src})$ **then**
    {Sufficient packet for reconstruction received}
    call $store(N_{tx})$ {Result for this iteration}
    done := **true**
  end if
end while

file. Then the number of sent packets ($N_{tx,i}$) are stored for later evaluation. This relation is captured in Eq. (5.6).

$$N_{rx,i} = N_{tx,i} - N_{lost,i} \tag{5.6}$$

$$N_{tx,i} = N_{fec,i} + N_{src} \tag{5.7}$$

The number of sent packets $N_{tx,i}$ of the *ith* iteration consists of source packets N_{src} (calculated according to Eq. 5.4) and FEC packets $N_{fec,i}$. This relation is described in Eq. (5.7)). Since we assume an *Ideal* FEC code, the client can reconstruct the source file when $N_{rx,i} = N_{src}$ are received (see also the *IF* condition in the pseudo code). That also means that at least $N_{lost,i}$ additional FEC packets must be sent in order to get at least N_{src} packets through the lossy channel. So, the reconstruction of *ith* iteration fails unless the sender has sent $N_{tx,i} = N_{lost,i} + N_{src}$ packets.

Packet losses are derived from Radio Block losses on the radio link. The radio Block Error Probability (BLEP) is read from a trace file, which is generated by a radio network simulation (see previous Chapter 4). Each BLEP value is used as a threshold for the random discarding of radio blocks. Each corrupted radio block causes again one or more IP packet losses. The mapping algorithm from radio block losses to IP packet losses is described in Section 4.1.

As results we know for each receiver position and each evaluation iteration the number of packets missing to successfully reconstruct the file. For the evaluations later on, we can check whether or not the reception was successful, when we send only a certain amount of redundancy data.

For non-*ideal* FEC codes, more than N_{src} packets may be needed, since some source symbols may not be covered by any of the N_{src} received packets. For instance, the Raptor FEC code requires between almost 0% and up to 5% more packets than N_{src}, i.e. the receiver needs to receive $N_{src} \cdot (1 + \epsilon)$ with $(0 < \epsilon < 0.05)$. We have evaluated the performance of Raptor code in Section 3.5.3. We have seen that we need less additional FEC symbols for larger number of

l_{file}	Size (in Byte) of the source file
l_{ps}	IP packet sized (in Byte) $l_{ps} = l_{hdr} + l_{pp}$
l_{pp}	Length of Packet Payload
$l_{miss,y}$	Missing data of *yth* receiver
l_{hdr}	Length of packet header
N_{src}	Required packets for successful source block reconstruction (assuming an *Ideal* FEC code, here)
$N_{fec,i}$	Number of transmitted repair Packets of the *ith* iteration ($N_{tx,i} = N_{fec,i} + N_{src}$)
$N_{tx,i}$	Number of transmitted packets
$N_{rx,i}$	Number of received packets
$N_{lost,i}$	Number of received packets
N_{UE}	target receiver groups size
N_{UEmd}	Receivers with missing data
OH_{tx}	Transmission Overhead
OH_{fec}	FEC overhead
c_{bearer}	Bitrate of MBMS radio bearer
c_{fr}	Bitrate of the file repair server link
t_{mbms}	MBMS transmission session duration

Table 5.3: Abbreviations of this chapter.

source symbols. Luby et al. [91] discuss and evaluate the performance of the Raptor code and describes also the Raptor inefficiency.

Only the average overhead values are considered in the following evaluations.

$$t_{mbms}\ [\text{sec}] = \frac{l_{file}\ [\text{bit}]}{c_{bearer}\ [\text{bps}]} \cdot \Big(1 + OH_{tx}\Big) \qquad (5.8)$$

The transmission overhead has a direct impact on the MBMS procedure duration (t_{mbms}). Transmissions with a higher transmission overhead must use a constant bit-rate channel for a longer time to transmit the same amount of user data (source of FEC packets). Thus we can say that a higher transmission overhead may lead to a higher resource consumption, since the transmission bearer is longer used for the transmission. Eq. (5.8) describes this relation. c_{bearer} is the bitrate of the MBMS bearer. The right summand of the last bracket described the transmission of all overhead. The transmission duration increase due to the packet headers is included in the right term.

All introduced variables are listed in Table 5.3.

The MBMS MTCH transmission is a *downlink-only* transmission. The RNCs selects the MTCH radio bearer in case of large terminal populations per cell for the file delivery. In order to gain from soft-combining of several radio links, the radio network must switch-on PTM radio bearers for the transmission in the adjacent cells. *Downlink-only* transmission also means, that there is no uplink

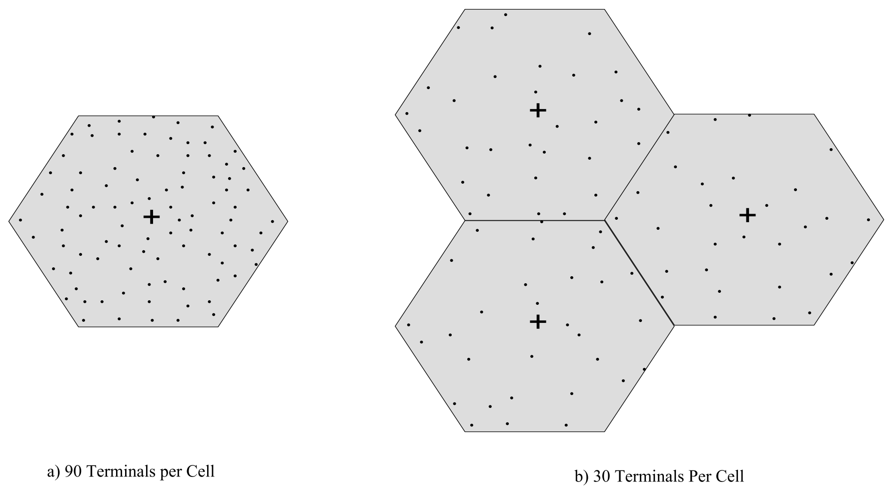


Figure 5.4: Comparison of 90 Terminals per cell with $3 \cdot 30$ Terminals per cell.

e.g. for power control associated to the MBMS downlink. The reception condition of the different terminals in the cell are independent from each other. This is only valid when using PTM radio bearers. In case of PTP radio bearers, the terminals send link quality measurements back to the network. The network adjusts the transmission power on the link quality measurements. This means, that the used transmission power in downlink depends on the terminal reception conditions.

The actual reception condition of a terminal depend on the radio propagation loss (distance to the base station) and the different fading conditions (fast and slow fading). Almost every location in the cell has a different block error probability. However, receiving MBMS data at location A has no impact on reception of MBMS data on location B. We therefore assume, that the MBMS transmission system follows the rules of an *Ergodic System* [99]. The *Ergodic hypothesis* says, that the average over time and the averages over the statistical ensemble are the same. The Ergodic assumption is well justified and for large numbers of receivers likely to be proven by standard algorithms.

There is also an intuitive explanation: Figure 5.4a shows a single cell with 90 terminals and Figure 5.4b shows three cells with 30 terminals each. Note that there are 90 terminals in both scenarios. Averaging the data loss over e.g. 120 reception repetitions with a system configuration according to Figure 5.4a is the same as the average data losses of 40 reception repetitions with a system configuration according to Figure 5.4b. The MBMS data reception of 10800 receivers ($= 120 \cdot 90\,\text{receivers} = 40 \cdot 3\,\text{cells} \cdot 30\,\text{receivers}$) contribute to the average. The receivers in both Figure 5.4b must further be randomly placed in the cell, due to the low number of receivers. Each location in the cell is different with respect to the radio conditions.

File Size	$l_{file} = 1$ MByte
l_{hdr}	44 Byte
Packet Size (l_{ip})	400, 600, 800 and 1000 Byte
Packet Size (l_{ip})	640 and 1280 Byte
Reception Iterations	$Rep = 1000$

Table 5.4: Simulation Assumptions.

5.2 MTCH Transmission Performance for Groups

In this section, we evaluate the dependency between additional AL-FEC and unsatisfied terminals and shows the effect on the actual effective media rate (goodput). A 1 MByte file is sent to all terminals in the simulation area. The packet header size is 44 Byte for all packets (IP, UDP and FLUTE headers). All results are averaged over 1000 iteration. The simulation assumptions for the radio network model are summarized in Table 5.1. Although we have shown BLEP graphs for deployments with of $r = 500$ m and $r = 2000$ m cell sizes in the previous sections, we focus here on $r = 2000$ m cell sizes only. All specific parameters for this evaluation are summarized in Table 5.4.

The file is partitioned into packets and then fragmented into radio blocks. We evaluate the general case of non-radio block aligned IP packets in this sections. Gains from aligning IP packets with radio block boundaries are discussed in Section 5.4.

The MBMS enhanced broadcast and the multicast mode allows usage of point-to-point radio bearers, if the interest in the target area does not justify the use of point-to-multipoint transmission resources. MBMS point-to-point radio links use either dedicated (DCH) or high speed downlink shared channels (HSDPA), which adapt to the link to keep a target BLER. It is expected, that the resulting block error probability distribution is much better compared to point- to-multipoint reception. This should be considered, when deciding on the added amount of FEC protection. It decreases the overall system performance, when the radio channel is allocated for unnecessary FEC transmission.

The dependency between packet size, radio block error rate and resulting effective media rate was explained in Chapter 4. We have chosen the packet sizes $l_{ip} = 400$, 600, 800 and 1000 Byte for the evaluation. A number of small packet sizes is chosen, because we expect high BLER rates per users (cf. Figure 4.4). Table 5.5 lists the header overhead of the four choices.

Figure 5.5 shows the percentage of satisfied terminals over the transmission overhead (OH_{tx}, cf. Eq. 5.3) for 400, 600, 800 and 1000 Byte packet sizes. The file partitioning into 400 Byte IP packet sizes result in a much higher number of packets than the partitioning for 1000 Byte. *Satisfied Terminals* have received sufficient MBMS data to recover the file after the MTCH transmission session successfully. Thus, *Satisfied Terminals* do not need to run ptp file repair.

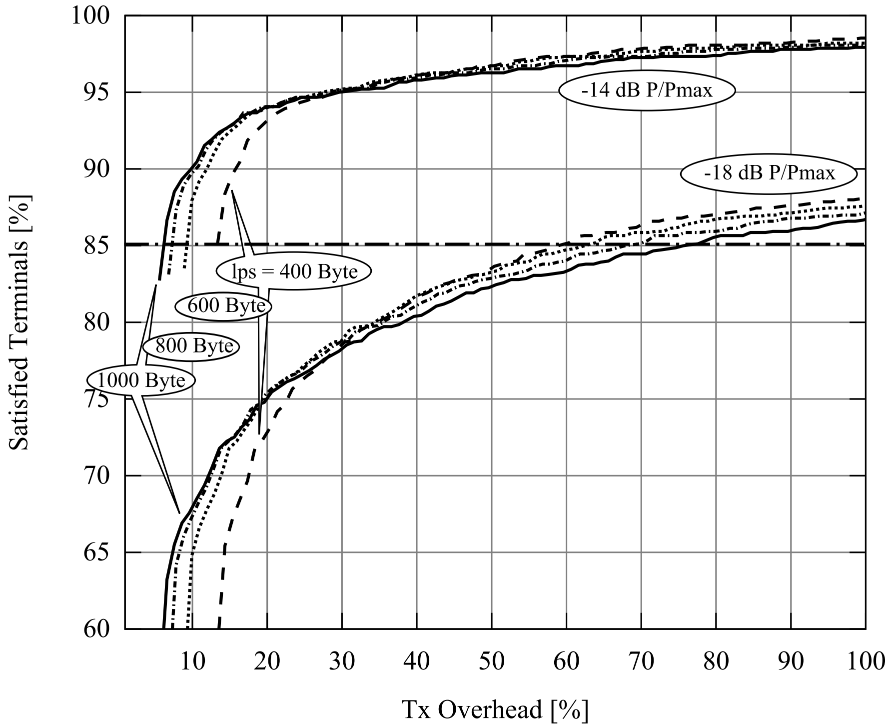


Figure 5.5: Satisfied Terminals vs. Transmission Overhead (-14dB and -18dB).

l_{ip}	OH_{hdr}	OH_{tx} for 85% coverage	
		-14 dB P/P_{max}	-18 dB P/P_{max}
400 Byte	12.4 %	13.4 %	59.4 %
600 Byte	7.9 %	9.9 %	62.9 %
800 Byte	5.8 %	7.8 %	68.8 %
1000 Byte	4.6 %	6.6 %	76.6 %

Table 5.5: Header overhead and required transmission overhead for 85% coverage.

Figure 5.5 contains the results for -14 dB and for -18 dB P/P_{max} transmission powers. The 400 Byte curves start at a higher transmission overhead than the 1000 Byte curves. The required transmission overhead to achieve 85% satisfied receivers is decreasing with increasing packet sizes for -14 dB transmission power, while it is increasing for the -18 dB case.

It is interesting to see that the curves of different packet sizes cross each other (at separate points for -14 dB and for -18 dB). For $OH_{tx} < 10\%$, the larger packet size of 1000 Byte gives the highest percentage of satisfied terminals. Main reason is here the smaller header overhead, which allows to send more FEC repair data at the same cost of transmission overhead. Table 5.5 contains the overhead values for 85 % coverage.

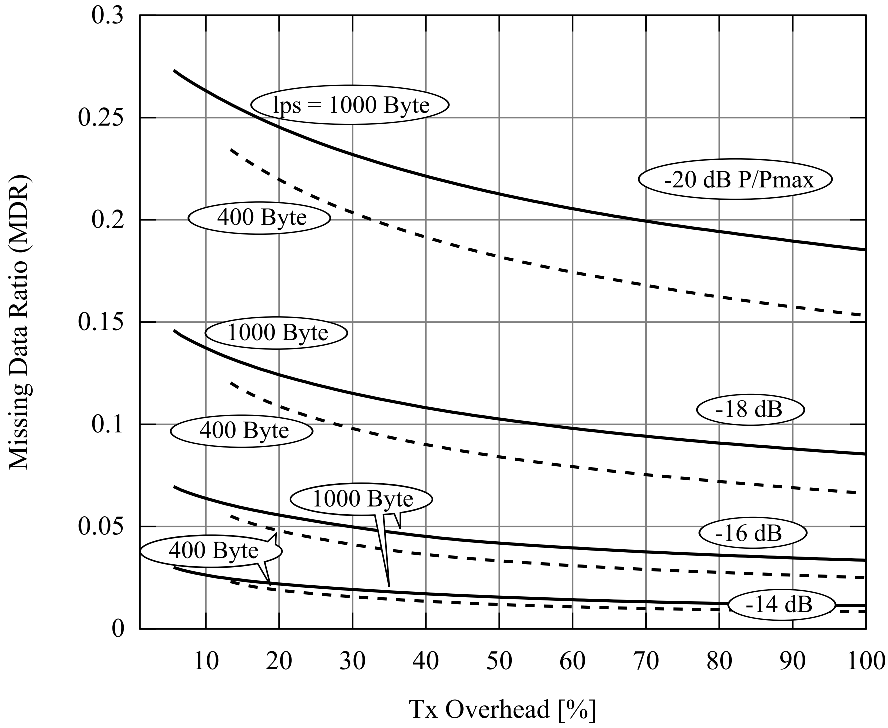


Figure 5.6: Average of normalized Missing payload Data Ratio (MDR).

$$MDR = \frac{\sum_{i=1}^{N_{UEmd}} l_{miss,i}}{(N_{rec} \cdot l_{file})} \tag{5.9}$$

Figure 5.6 shows the average Missing payload Data Ratio (MDR) for the different packet sizes and for different transmission power settings over the transmission overhead (OH_{tx}). The MDR is defined by Eq. 5.9 as the fraction of the sum of all missing data and the total received file size ($N_{rec} \cdot l_{file}$). The parameter $l_{miss,i}$ is missing data of the ith receiver. The MDR is independent of the file and group size, because larger l_{file} also leads to proportionally larger $l_{miss,i}$. Generally, smaller packet sizes result in less missing payload data after the MTCH transmission and should be therefore preferred. The influence of the packet size is decreasing for increasing transmission powers. The amount of missing payload is an important dimensioning factor for the ptp file repair server. The server and the server link towards the network must be sufficiently dimensioned to handle the repair load (cf. [97] and [100] for details). Increasing the transmission power from -20 dB to -16 dB at a transmission overhead of 20% decreases the MDR by a factor of 4.4. Increasing the transmission power from -18 dB to -14 dB at a transmission overhead of 20% decreases the MDR by a factor of 8.3. Note that more than 93% of the terminals are already satisfied and do not need FEC redundancy anymore.

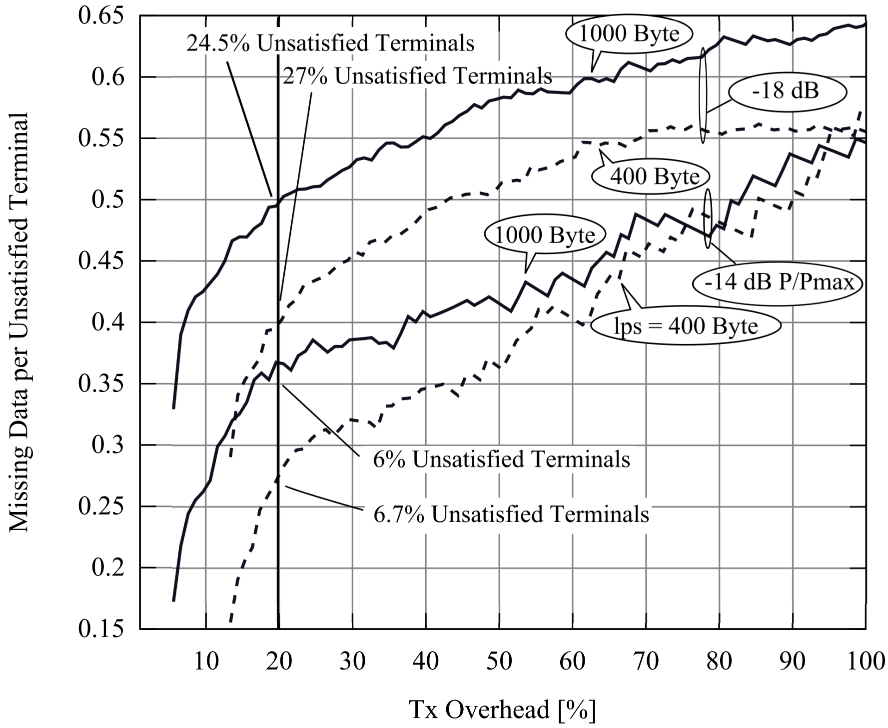


Figure 5.7: Average missing payload data per unsatisfied terminal (-14 dB and -18 dB).

Figure 5.7 shows the amount of missing data per *unsatisfied terminals* (MDRpUT), which is defined as

$$MDRpUT = \frac{\sum_{i=1}^{N_{UEmd}} l_{miss,i}}{(N_{UEmd} \cdot l_{file})}. \tag{5.10}$$

The difference to the MDR, Eq. (5.9), is that the number of unsatisfied receivers N_{UEmd} is used for normalization. The MDRpUT increases with the transmission overhead despite the fact that more AL-FEC redundancy is added so that a given terminal has lower missing data. With increasing transmission overhead more and more users that had only little missing data become satisfied and are then not counted anymore in this MDRpUT definition. It is interesting to see that the -14 dB and the -18 dB curves converge for higher transmission overhead (thus higher FEC overhead) although the transmission power differs by 4 dB. The other interesting aspect is the number of unsatisfied terminals: At 20% transmission overhead, around 24% of the receiver group need additional data at -18 dB transmission power, but only around 6% at -14 dB. Thus, file repair would be needed by a much lower number of terminals in case of -14 dB transmission power.

5.3 Balancing AL-FEC with PTP File Repair

So far, we have only considered Application Layer FEC to increase the transmission reliability, but MBMS Download also provide file repair mechanism using Point-to-Point and Point-to-Multipoint bearers. It is clear that adding too much FEC overhead will decrease the system efficiency. It is also clear, that we should not try to recover all packet losses with AL-FEC.

When the file repair is anyhow necessary, then we should balance the needed resources of the MTCH transmission with the needed resources for the file repair. In particular HSPA type of bearers are designed to use the currently available link capacity.

In this section, we transfer the findings from the previous sections on the entire MBMS file delivery procedure, without modeling the file repair procedure in detail. A detailed evaluation of the file repair procedure is described in Chapter 6.

The resource consumption for the PTP file repair procedure is proportional with the simultaneous users of the file repair service, thus depending on the distribution of receivers per cell and the available cell resource budget. This distribution of receivers was not important until now, since we evaluated only the downlink direction of the file transmission. For the MTCH evaluation, there is no difference whether we serve 1500 receivers in a single cell or averaged over multiple cells (e.g. 15 cells with 100 receivers each). The MBMS MTCH channel takes the same radio resources independent from the number of receivers.

Figure 5.8 shows the duration of the MBMS MTCH transmission and the successive ptp file repair over the transmission overhead for a target group density of 40 receivers per cell (see also Figure 2.11 in Chapter 2.4.2 for the two phases). The ptp file repair is assumed to use HSPA and the available cell HSPA capacity is shared among all HS users. Other services need also resources of the cell so that the available capacity for file repair is lower than the cell capacity. Figure 5.8 assumes 0.5 Mbps on average for all ptp file repair. Note, the HSPA performance is not evaluated by the radio network simulator described above.

The earlier discussed Figure 5.6 shows a decreasing remaining missing data after the MBMS MTCH transmission with increasing transmission overhead. A lower amount of missing data need to be repaired using ptp file repair with increasing transmission overhead. But Figure 5.7 shows that the remaining data per unsatisfied terminal is increasing. These are the terminals in the really bad coverage situation. This means that the *unnecessary* FEC data is the factor contributing most for the high transmission duration towards increasing transmission overhead. The BLER curve shows a low number of terminals with high BLERs and a high number of terminals with low BLERs. The terminals with high BLER should use the ptp file repair. Otherwise, too much radio resources for PTM bearers are used.

Figure 5.8 shows that only a low percentage of FEC overhead is needed to minimizes the transmission duration. The amount of AL-FEC increases with

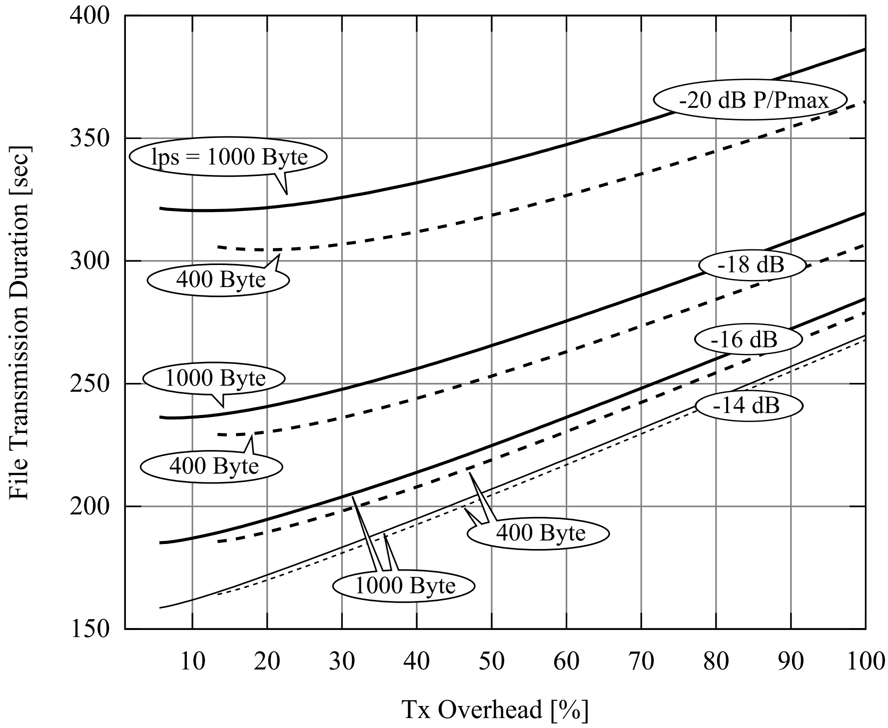


Figure 5.8: Average total file transmission duration assuming 0.5 Mbps (av.) HSPA capacity for ptp file repair.

increasing transmission overhead (OH_{tx}). The amount of missing data after the MTCH is decreasing with increasing AL-FEC. Also the duration for the PTP file repair is decreasing with increasing transmission overhead, since less data need to be repaired. The MBMS file transmission durations increase for transmission overhead larger than 30% in Figure 5.8. The dominating factor for the MBMS file transmission duration is for transmission overhead larger than 30% the transmission duration of the additional FEC.

Almost no additional AL-FEC actually is necessary when the system is operated with a rather high transmission power of $-14\,dBP/P_{max}$. If the system is operated at a lower transmission power setting of -18 dB or even -20 dB, then only moderate FEC overhead of 10% to 20% should be added. The remaining data should be repaired using ptp file repair.

We have seen in [97], that the file repair server is likely the bottleneck in case of large groups, depending on the file size. Therefore, and in contrast to Luby [91] and Gomez [101], we use the MDR as optimization target. The transmission energy per cell is defined as $E = P \cdot t$, with P as used power resource and t as resource usage duration. Eq. (5.8) describes the transmission durations for the MBMS PTM phase.

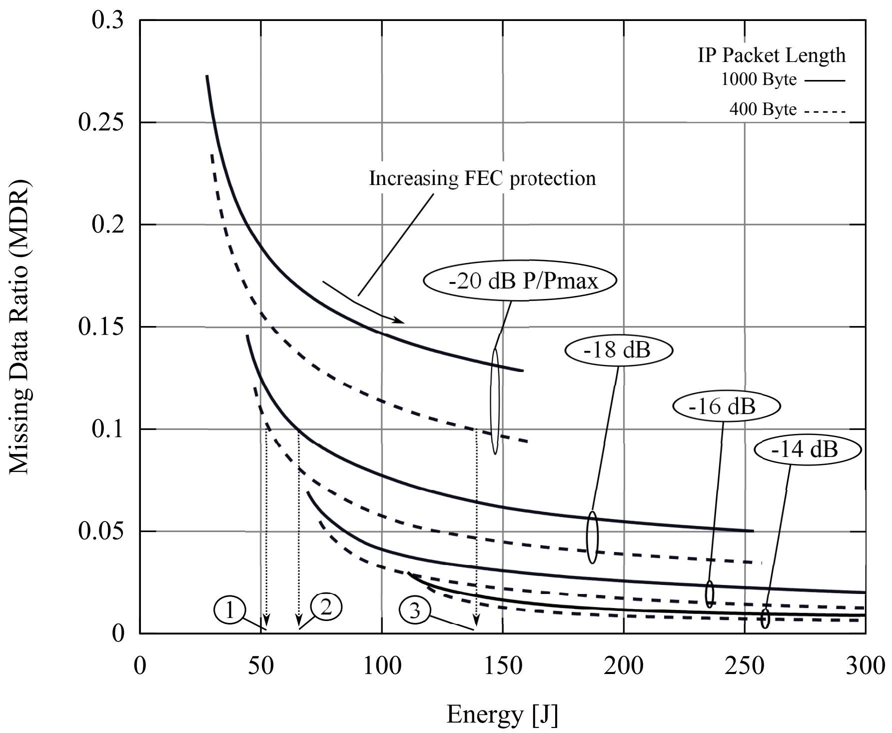


Figure 5.9: Used Energy versus the Missing Data Ratio.

$$E = P \cdot \frac{l_{file}}{c_{bearer}} + P \cdot \frac{OH_{tx} \cdot l_{file}}{c_{bearer}} \tag{5.11}$$

Eq. 5.11 defines the transmission energy for a given transmission overhead with c_{bearer} as MBMS ptm bearer bitrate. The left summand gives the needed energy to transmit the actual content and the right summand the required energy for the FEC and packet header overhead. The absolute values of the energy E is not of actual importance here, since we use the energy only for comparison reasons. Even if there is an error in the energy calculations, this has no effects on the qualitative results of the following discussions.

Figure 5.9 shows the required energy over the missing data ratio (MDR). The missing data after the MTCH transmission is actually the load for the file repair service. Different combinations of transmission power, AL-FEC protection and IP packet sizes result to the same missing data. Increasing the FEC protection is equivalent with increasing the energy, since the transmission of additional FEC packets requires energy (see indication in figure). One immediate observation: Larger IP packets require a higher energy to achieve the same missing data ratio.

The graph in Figure 5.9 is read as following: A target of $MDR = 0.1$ means that the amount of missing data after the MTCH transmission should be 10% of total amount of data (thus $0.1 \cdot (N_{rec} \cdot l_{file})$). Then the optimal transmission power is -18 dB with a packet size of 400 Byte, since it requires the lowest energy to

Tx Power	l_{ip}		Required OH_{tx}	Required Energy	Energy Increase
-20 dB	1000 Byte	$\rightarrow$	$> 500\%$	> 160 J	$> 200\%$
-20 dB	400 Byte	$\rightarrow$	423.4%	137.2 J	156.9%
-18 dB	1000 Byte	$\rightarrow$	54.6%	64.8 J	21.35%
-18 dB	400 Byte	$\rightarrow$	27.36%	53.4 J	Reference

Table 5.6: Required energy for the target of $MDR = 0.1$.

achieve this target. It requires only an energy of 53.4 J for the transmission. This point of operation is marked with (1) in Figure 5.9). The corresponding transmission overhead is with 27.3% rather low (not shown in the figure). A transmission power of -20 dB would require a 156.9% higher transmission energy to archive the same target of $MDR = 0.1$ target. This point of operation is marked with (3) in the figure. Generalizing, the optimal trade-off combination of missing data ration, transmission energy and IP packet sizes lies on the lower envelop of all the graphs in Figure 5.9. An overview of the required energy and the required transmission overhead to achieve the Missing Data Ration target of $MDR = 0.1$ is given in Table 5.6.

The ptp file repair server capacity depends on the used server hardware and the available server link capacity. Both, server hardware and link capacity is assumed to be constant. The upper limit of the ptp file repair service duration is assumed to be a full day (24h), but expected to be much lower in practice. When we require for example that the ptp file repair procedure is finished after maximal $t_{fr-max} = 60$ sec and that the server is capable of sending $c_{fr} = 1$ Gbps of file repair data, then we can provide 55.9 GByte of file repair data. The transmission power, the IP packet length and the amount of FEC overhead must be selected, so that the resulting amount of missing data is lower than the available file repair service capacity. At the time of starting the MBMS file delivery session, the file size and the target audience size can be known. Eq. 5.12 defines the File Repair service Ratio (FRR) so that $MDR \leq FRR$ can become the optimization target for the MBMS file delivery session. N_{rec} is the (expected) total number of users and l_{file} the file size.

$$FRR = \frac{c_{fr} \cdot t_{fr-max}}{(N_{rec} \cdot l_{file})} \tag{5.12}$$

Thus, in case of $c_{fr} = 1$ Gbps and $t_{fr-max} = 60$ sec, we should aims for $MDR \leq 0.57$ if the expected target group size is $N_{rec} = 100000$ receivers and the file length is $l_{file} = 1$ MByte. The MDR should be decreased, in case of larger audience sizes or larger file sizes. In the next chapter we see that the minimum for the file repair service is the Sequential Delivery Time (*SeqDelT*) of the missing data over the file repair link.

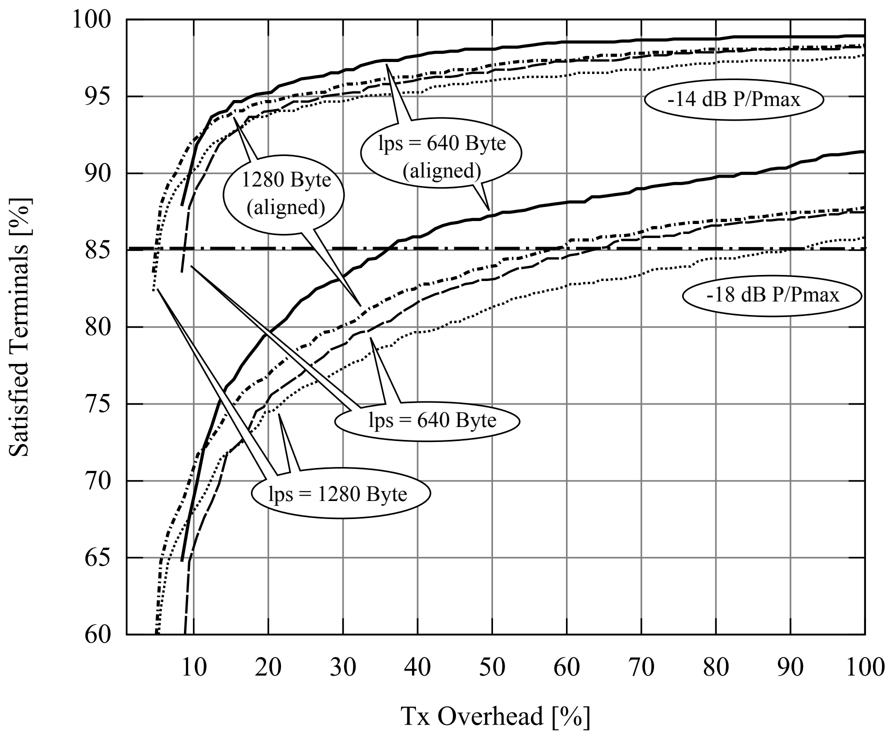


Figure 5.10: Satisfied Terminals vs. Transmission Overhead for radio block aligned IP Packets (-14dB and -18dB).

5.4 Improvements due to packet alignment

We have seen in Section 4.3 that the alignment of IP packets to radio block boundaries brings improvements of up to 25% of the channel capacity for certain IP packet sizes and radio block sizes. We also saw that the capacity improvement is not constant over all block error rates. Until now, we avoided aligning the IP packets with radio block boundaries. On the contrary, we have selected IP packet sizes, which are not a fraction or multiplier of the radio block size. We have further ensured, that IP packets and radio blocks are aligned at the start of the evaluation runs.

Main reason for the capacity improvement of radio block aligned IP packets is that the number of radio blocks that impact the packet error rate is reduced. In case of aligned 1280 Byte packets with 640 Byte radio block sizes, the block error probability of two radio blocks influence the packet error rate. In case of un-aligned packets, three radio blocks may cause the loss of a packet.

In the following, the alignment gains for 1280 Byte and 640 Byte IP packets sizes (l_{ip}) are evaluated. We compare the satisfied terminals and also missing data rations when aligning the IP Packets with radio blocks against un-aligned scenarios. The radio model parameters from previous section (Section 5.2) are also used for this evaluations.

Figure 5.10 shows the percentage of satisfied terminals over used Transmission Overhead (OH_{tx}) parameterized with two packet sizes (1280 Byte and 640 Byte packets) and two transmission power settings (P/P_{max} = -14dB and -18dB) for the evaluation. It shows the same kind of results as Figure 5.5, but with sightly different packet sizes and in particular with radio block aligned IP packets. Remember, the transmission overhead includes FEC redundancy and also all header overheads and we use it to ensure a fair performance comparison when using different packet sizes.

As expected, when using a high transmission power of -14 dB, the percentage of the satisfied terminals after the transmission with a certain value of transmission overhead (OH_{tx}) is much better. The upper two curves (solid -640 Byte and dash-dotted -1280 Bytes) are the results when aligning the IP packets are radio blocks for both power settings. In case of -14dB P/P_{max} transmission power, the amount of satisfied terminals is around 1.5% higher when aligning the IP packets to radio blocks. In case of -18dB P/P_{max} transmission power, aligning the packet to block boundaries lead to an increase of 4% to 5% of satisfied terminals.

The 1280 Byte crosses the 640 Byte curve at a Transmission Overhead of approximately 12% for block aligned packets for both transmission power settings. The crossing is at around 15% transmission overhead for the non aligned packets.

Using 1280 Byte packets provides the best results for transmission overheads up to 12%. Up to 12% transmission overhead, most receivers with rather low block error rates are satisfied. For larger transmission overheads, 640 Byte packets show the better results. We have seen during goodput evaluations (see Figure 4.10 on Page 76) that the 1280 Byte packets show a better goodput for lower block error rates.

So, we see out of this experiments that the alignment of IP packet to radio block boundaries show generally a better performance than not aligned packets. However, we cannot conclude whether we should better use 640 Byte or 1280 Byte packets. This depends on the selected transmission overhead, thus also on the added FEC redundancy. The used transmission energy as evaluated in next section may clarify this.

Figure 5.11 depicts the normalized missing data ratio (MDR) over the transmission overhead, similar to Figure 5.6. Here, the curves with 640 Byte and 1280 Byte IP packet sizes do not cross each other; the 640 Byte packets perform always better than the 1280 Byte packet. In case of radio block aligned IP packets, exactly one IP packet is fit into one radio block, thus the IP packet error rate is the same as the radio block error rate. Table 5.7 summarizes the reduction of the missing data ration when aligning the IP packets with radio blocks for different transmission overhead percentages. For the 640 Byte packets and the transmission power -14dB P/P_{max}, we find a reduction of around 40% for the missing data ratio. With a lower transmission power of -18dB and larger packets of 1280 Byte, the reduction of MDR is still around 18%. Thus, the file repair server needs to provide between 18% and 40% less file repair data when aligning the IP packets with the radio block boundaries.

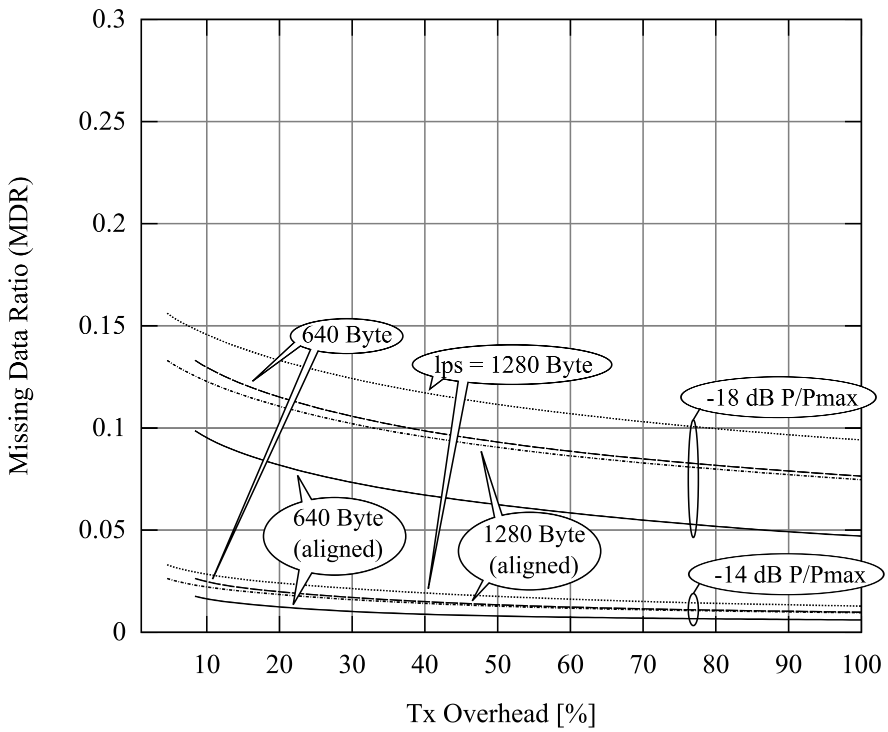


Figure 5.11: Normalized Missing Payload Data Ratio for radio block aligned IP Packets.

	P/P_{max} = -14dB		P/P_{max} = -18 dB	
OH_{tx}	l_{ip} = 1280 Byte	640 Byte	1280 Byte	640 Byte
20%	23.26%	37.57%	16.84%	29.03 %
40%	26.37%	41.29%	18.37%	31.88 %
60%	27.16%	40.32%	19.28%	34.1 %
80%	26.21%	39.38%	19.84%	36.6 %

Table 5.7: Reduction of Missing Data Ratio due to IP packet alignment to radio blocks.

We have seen in previous sections, that aligning the IP packets to radio block boundaries brings some gains. It is of course not for all scenarios possible to create IP packets of same packet size (incl. all IP packet headers), however, with FLUTE and a little care it is very well possible. In the following, we evaluate the gains from boundary aligned packets further.

Figure 5.12 depicts the normalized missing data ratio over the needed energy to transmit the content to all users. The energy is calculated according to Eq. (5.11). Table 5.8 summaries the needed energy and needed transmission overhead to achieve a missing data ratio target of $MDR = 0.1$. Note the intention of the values is the comparison between different configurations. The

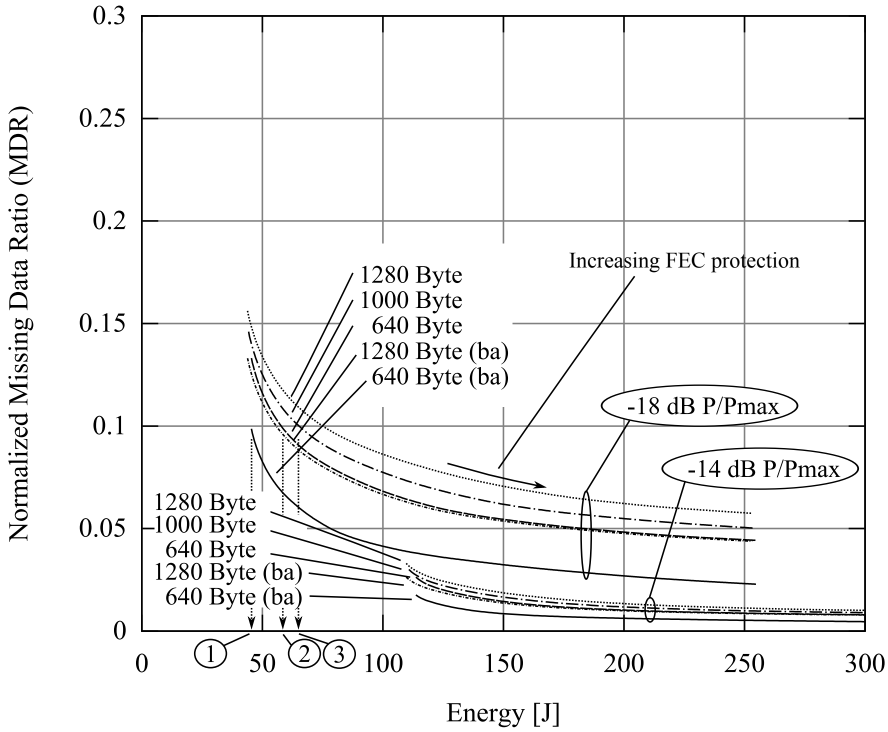


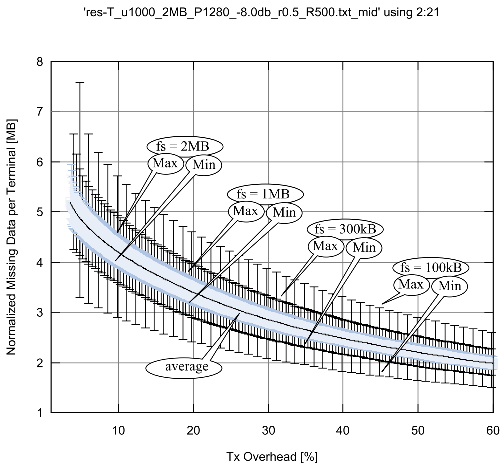
Figure 5.12: Used Energy vs. File Repair Load with radio block aligned packets.

Tx Power	l_{ip}		Required OH_{tx}	Required Energy	Energy Reduction
-18 dB	1280 Byte	→	78.56%	74.9 J	
-18 dB	640 Byte	→	37.4%	57.62 J	
-18 dB	1280 Byte (ba)	→	32.56%	55.6 J	25.8%
-18 dB	640 Byte (ba)	→	8.4%	45.46 J	21.1%

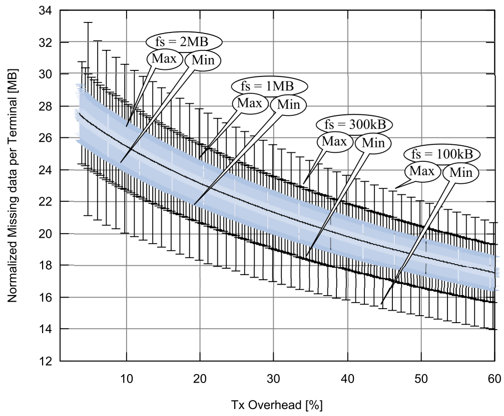
Table 5.8: Needed energy to an MDR target of $MDR = 0.1$ with radio block aligned packets.

boundary aligned (ba) transmission with 640 Byte packets required 21.1% less energy to achieve the same missing data ratio target of $MDR = 0.1$. The used energy is even reduced by 25.7% in case of 1280 Byte packets.

The larger IP packet require a higher transmission energy to achieve a Missing Data Ration of 0.1. Aligning the IP packets to radio blocks allows to reduce the transmission energy by 25.8% (1280 Byte packets) or 21.1% (640 Byte packets) and still achieve the same missing data ratio as summarized in Table 5.8. The needed energy for 640 Byte packets is indicated by points (1) and (2) in Figure 5.12. Figure 5.12 depicts also the performance using 1000 Byte packets. Note that point (3) in Figure 5.12 is the same as point (2) in Figure 5.9.



(a) Tx Power -18 dB P/P_{max}



(b) Tx Power -20 dB P/P_{max}

Figure 5.13: Missing Data Ratio for different file sizes; Cell Radius 500 m.

5.5 Dependency on the File and Group Size

The MBMS file delivery method is capable of transporting any type of files. We have only evaluated files of 1 MByte size until now. The intention of this section is to discuss the impact on the results when using larger or smaller files.

Files of 1 MByte size can contain approximately 1 minute music (encoded at 128 kbps) or 15 seconds of video (encoded at 500 kbps) for a Smartphone. The used file sizes depend on the type of application. Web pages without large images are typically only of around several 100 kByte size, video clips are easily become several Megabytes to Gigabyte large, depending on the video quality and resolution.

Figures 5.13a and 5.13b depict the missing data per terminal for files of 100 kByte, 300 kByte, 1 MByte and 2 Mbyte file size. The transmit power is set to $-18\,dB$ in Figure 5.13a and to $-20\,dB$ in Figure 5.13b.

Like for the missing data ratio in Eq. (5.9), the total amount of received data ($N_{rec} \cdot l_{file}$) is used for the normalization. Main reason to normalize the missing data to total amount of received data for normalization is to stay independent from the actual used file and the receiver group size. For each transmission overhead setting, the minimal, the maximal and the average amount of missing data is depicted.

The average normalized missing data is the same for all four file sizes. Figure 5.13 shows that the best and the worst data losses get *more extreme*. In particular for small file sizes, the amount of missing data varies very much from transmission to transmission. For small file sizes, a short time of bad radio coverage has a high impact on the data loss average. During long file transmission, more terminals may experience in relation bad radio reception.

5.6 Conclusion of this chapter

This chapter evaluates the system resource usage of reliable file delivery over the MBMS Traffic Channel (MTCH). The MTCH transmit power, the IP packet sizes, the Application Layer FEC (AL-FEC) overhead and also the ptp File Repair impact the system resource usage. With AL-FEC, the IP packet error rate can be traded with AL-FEC redundancy, which incurs transmission overhead. A receiver is considered to be *satisfied* when all AL-FEC source blocks of the files can be decoded successfully, thus when the file is free from any transmission error.

The dependency of the percentage of satisfied receivers on the transmission overhead is evaluated. When the transmit power and transmission overhead is set high enough to $-14\,dB$ of the total power and 30% of transmission overhead is used, then the classical design goal of 95% satisfied terminals is achieved. Beyond this power and overhead, the satisfied terminal rate increases only gradually. The remaining unsatisfied receivers have increasingly higher BLER.

This increasingly higher BLER is also the reason why smaller IP packets show a higher percentage of satisfied terminals when a transmission overhead larger than 30% is used. The larger IP packets of 1000 Byte show a better performance only up to a transmission overhead of roughly 30%.

Shorter IP packets show always a lower amount of missing data with increasing transmission overhead. The normalized Missing payload Data Ratio (MDR) is decreasing slowly with increasing transmission overhead, because the average MDR per unsatisfied receiver increases as the remaining unsatisfied UEs have increasingly higher BLER.

The amount of missing data is proportional to the required transmission energy for PTP file repair and is therefore one of our major evaluation metrics for the MTCH transmission of the first phase. The acceptable amount of total missing data increases with the acceptable duration for the repair phase and also with increased repair server capacity. We assume here the bitrate of the link to the file repair servers as the dominating server related constraint.

The MTCH transmission duration and transmission power are both radio resource costs that can be traded off with each other. The total resource consumption can be measured as a product of both, i.e. by the transmission energy. Different combinations of transmit power, AL-FEC and IP Packet size should be compared by using the required transmission energy for a given missing data target and the required transmission energy should be minimized. Generally, the increase of transmit power and the use of smaller IP packets lead to a more system efficient reliable file delivery for a given amount of missing data after the MTCH transmission. Only a small amount of AL-FEC should be added in particular when a low transmit power is used.

The radio network uses the MTCH only when the interest in the cell is very high. Regular unicast radio bearers are used when the interest is low (cf. Section 2.2.1). Unicast radio bearers typically employ a local retransmission scheme (ARQ) to quickly recover lost data segments and AL-FEC is actually not needed. This conclusion about the needed amount of AL-FEC for the MTCH is in particular interesting, when the system uses both, MTCH and regular unicast bearers to distribute files in the different regions. When only a low amount of AL-FEC is added, the transmission is still system efficient when unicast radio bearers are used.

Using the missing data ratio (MDR) as defined by Eq. (5.9), the measure is independent from the transmitted file size and also the target group size. It enables comparing the performance of different system configurations against each other. It is also useful to balance the needed resources of the MTCH transmission phase and the file repair phase. However, the file repair service duration increases with increasing file size or increasing group size since more data is transmitted over the same link. This dependency is described by Eq. (5.12).

Alignment with the radio block boundaries is advantageous when all IP packets of the MBMS transmission have always the same packet size. This is possible when care is taken when determining the encoding symbol size, thus the packet payload and also the IP headers. Highest alignment gains can be expected, when the IP packet sizes is equal to the radio block size. Here we have used 640 Byte radio block sizes.

6

Point to Point File Repair

The performance and behavior of the point-to-point (PTP) file repair is evaluated in this chapter. The reliable MBMS file delivery is subdivided into two phases. The data is transmitted using MBMS bearer services during the first phase. Additional redundancy packets (FEC packets) may be added to the MBMS data transfer session to increase reliability. We have evaluated the MBMS MTCH performance in Chapters 4 and 5. We have seen in the last chapter that we should consider the file repair procedure in order to minimize the overall energy for the transmission to *all* users.

The file repair service is provided during the second phase. The Point-to-Point (PTP) file repair service uses unicast bearers. Receiver may fetch additional data in order to successfully recover the data. The Point-to-Multipoint (PTM) file repair service uses the MBMS bearers to distribute additional data. The PTM file repair service is discussed in Chapter 7 in detail.

A general introduction to the file repair procedure, which belongs to the associated delivery procedures, is given in Section 2.4.2. The two phases are shown in Figure 2.11.

Parts of the following discussion and results have been published in [100] and [97]. There are two possible bottleneck links in the system, which impact the performance and duration of the PTP file repair. The maximal throughput of the individual users is limited by the available bitrate of the radio interface. The throughput of all users is limited by the available link bitrate between the file repair server and the mobile system.

Lohmar *et al.* [100][97] assume Reed-Solomon and Raptor FEC codes for the MBMS download session to create missing data per receiver. The Reed Solomon code has a very short source block length, which results into a more inefficient usage of the redundancy packets. An FEC symbol can only be used to recover data from the belonging source block and not for any of the source symbols. The Raptor FEC code has a superior performance than the Reed Solomon code for long files, since the redundancy is created for very large source blocks.

For the evaluations in this chapter, only the amount of missing data per receiver and the distribution of missing data of the all receivers is relevant. We have introduced the *Missing Data Ratio* (Eq. (5.9)) in the last chapter as a measure, which is independent from the target group size, the selected packet sizes and the transmission overhead. We have shown that multiple combinations of MTCH transmit power, IP packet sizes and additional AL-FEC redundancy may lead to the same missing data after the MTCH transmission. The missing data after the MTCH transmission and the distribution of missing data over receivers form the load on the file repair service.

This chapter starts with a theoretical discussion about potential bottlenecks in the system in Section 6.1. All receivers of the MBMS download transmission sessions typically start the file repair procedure at the same time. This may lead to a synchronization of file repair procedures, thus potentially to an overload of the system. Then, we describe the set-up of the evaluation environment and present the results in Section 6.3. Conclusions of the PTP file repair evaluations are summarized in Section 6.4.

6.1 Using and dimensioning the ptp File Repair

The PTP file repair uses Interactive 3G bearers for requesting additional data chunks in order to reconstruct the file(s). The file repair belongs to the associated delivery procedures and is performed after the MBMS data transfer phase.

The 3GPP specification [4] foresees two types of overload prevention schemes for the point-to-point file repair: Spreading the file repair load in time and across network elements. The first scheme is realized by defining a so-called back-off window by the server. Each client which needs to use the file repair server, randomly selects a waiting time (δ_i) from the wait-time window (Δ_{wtWnd}) and delays the procedure start time. To spread the load across network elements, the server may configure a set of file repair servers. If more than one server is announced, clients randomly select one of the servers from the list. The specification just defines the procedures, but does not give any configuration or combination hints.

The file repair procedure is configured with the MBMS transfer. A separate associate delivery description file is sent with the rest of the user data to all receivers. This description file contains (among others) the wait-time window duration (Δ_{wtWnd}) and a list of available PTP file repair servers (see Figure 2.11 in Chapter 2.4.2). Each receiver, who must run the file repair procedure to fetch additional data picks randomly a delay time out of the range, given by the wait-time window. Each client also picks randomly one file repair server, when the configuration description contains a list of servers. The configuration of the file repair procedure can be updated with each new MBMS data transfer.

Figure 6.1 shows a general topology for such systems. It is very much an aggregation tree like network topology, where only a very few number of file repair service function are available for all receivers. Please note that there is only a single GGSN network element depicted. The GGSN terminates the point-to-point tunnels from the mobile phones in mobile networks. Thus, all traffic is routed through the same GGSN unless the network provides different GGSN addresses at session start-up. Several file repair service functions may be reached from the same GGSN. We focus only on a single GGSN and a single file repair

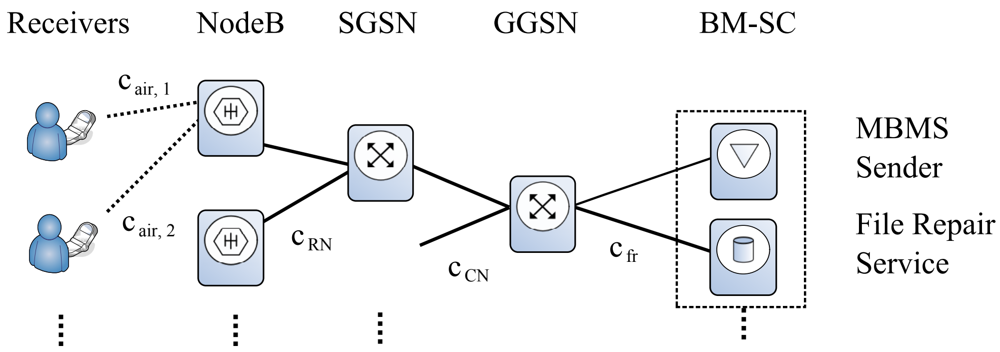


Figure 6.1: Topology for file repair services.

server link in this chapter.

The network topology applicable for the file repair services is actually an aggregation network, where all traffic from the receivers are aggregated to the tree root (here the PTP file repair server). As in all aggregation networks, the upstream links must be carefully dimensioned. In the worst case, the upstream leading link of a node must provide as much bitrate as the sum of the incoming links, so that all downstream nodes may utilize simultaneously the full bitrate of the links. This network dimensioning strategy is described by

$$c_{CN}\ [\text{bps}] \ = \sum_{i=1}^{N} c_{RN,i}\ [\text{bps}]\ . \tag{6.1}$$

c_{CN} is a link towards the core network and c_{RN} are the links from the radio network. In principle, the sum of the links from the radio network ($c_{RN,i}$) should be the outgoing link c_{CN}. However, this dimensioning strategy will lead to a pretty much overprovisioned network, since typically not all downstream nodes are active at the same time. *Spreading the repair load in time* (MBMS download configuration feature) is actually intended to control the load on the network links.

One critical bottleneck link in this aggregation tree topologies is the link to the root node (here the PTP file repair server). File repair traffic of all unsatisfied receivers must pass through this link. The mechanism with the wait time window is intended to control the load on this link. In the following we discuss *how* to set the wait window size.

Lets introduce the *Sequential Delivery Time* as

$$SeqDelT\ [\text{sec}] = \frac{\sum_{i=1}^{N_{rec}} l_{miss,i}\ [\text{bit}]}{c_{fr}\ [\text{bps}]} = MDR \cdot \frac{(N_{rec} \cdot l_{file})}{c_{fr}}, \tag{6.2}$$

which is the time to delivery all missing data over the link c_{fr} between the network and the file repair server. This Sequential Delivery Time gives the lower bound of the file repair service. The file repair load is the sum of the missing data

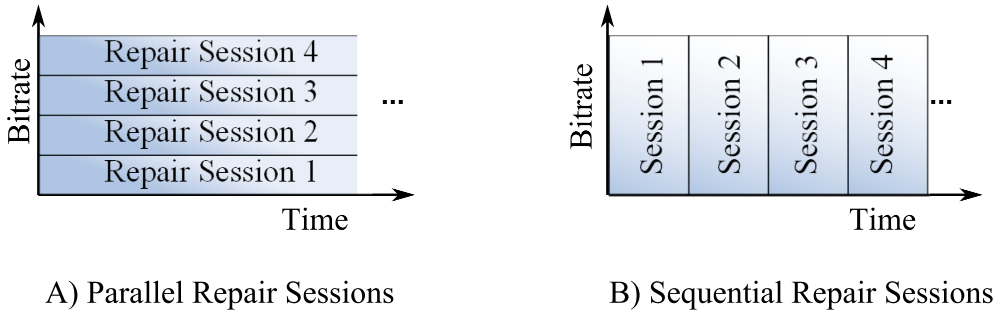


Figure 6.2: Parallel Delivery vs. Sequential Delivery.

($l_{miss,i}$) of all receivers N_{rec}. Figure 6.2b illustrates this consideration of transmitting the repair data sequentially instead of simultaneously. However, since the radio link capacity $c_{air,i}$ is typically much smaller than the connection to the file repair server c_{fr}, a number of PTP file repair sessions must be simultaneously active. Figure 6.2a illustrates this. Under optimal conditions, a new file repair session must become active as soon as one file repair finishes.

The *SeqDelT* may also be expressed as the product of the Missing Data Ration (MDR) with the target group size (N_{rec}) transmission file size (l_{file}) (see Eq. (5.9) in Section 5.2 for details). We use the Sequential Delivery Time in the following for normalization of the durations. In that way, the durations become independent from the link bitrate c_{fr} and also from the amount of missing data.

The link c_{fr} is fully utilized if the sum of the bit rates of the radio links $c_{air,i}$ of simultaneous users is equal to the bit rate of the link c_{fr}. This of course assumes, that new users start their file repair session at the moment when another user finishes the file repair. The link c_{fr} is not overloaded if the summary data bit rate of active connections is lower than the bit rate of the link c_{fr}. Eq. (6.3) expresses the condition of possible highest utilization but not overload of the Link c_{fr}.

$$c_{fr} \text{ [bps]} \cong \sum_{i=1}^{K} c_{air,i} \text{ [bps]} \tag{6.3}$$

Eq. 6.3 describes this type of dimensioning. Here, K is the number of simultaneous active sessions.

Fulfillment of the Eq. (6.3) ensures no overload of the link c_{fr}. The duration of the repair procedure is equal to the Sequential Download Time when the network is fully utilized during the whole repair time period. This corresponds to Eq. (6.3) when the bit rate of the Link c_{fr} is equal to the sum of the bit rates of the link of all active users.

6.2 Description of the Evaluation Environment

In this section, we evaluate a network topology as depicted in Figure 6.1. Only one PTP file repair server is connected to the system using a 1 Mbps link (c_{fr}). The network between file repair server and radio base station is dimensioned in such a way, that it will certainly not become the bottleneck. In order for a fair evaluation, we have secured that each radio base station serves approximately the same number of unsatisfied receivers.

Thus, there are two possible bottleneck links: Either the link c_{fr} to the file repair server or the radio link $c_{air,i}$ will become the bottleneck. We assumed an Release 99 UMTS network with a $c_{air} = 64\,kbps$ dedicated channel. Here, each receiver gets a fixed bitrate of 64 kbps independently from the other load in the cell.

We also evaluate a simplified HSPA type of shared bearer. All active users in the cell share the cell capacity. In our simplification, we have equally shared the capacity among all active users and not modeled any connectivity dependent degradations. When a receiver becomes active to start the PTP file repair, it requests capacity from its associated controller. We have selected 640 kbps ($= 10 \cdot 64$ kbps) cell capacity for the HSPA bearer. We assume for a fair comparison between the two radio types, that the same capacity is available.

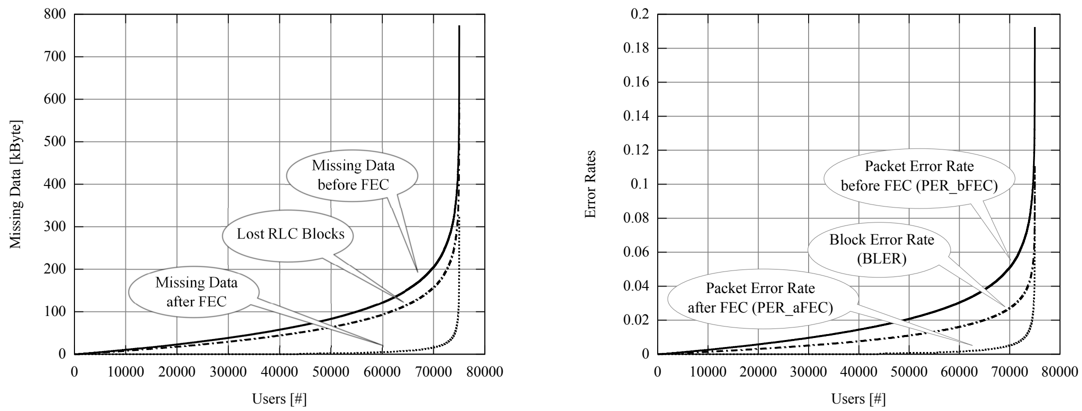
Link latency between the file repair server and the RNC nodes is in all evaluations 6 ms. Link latency between the RNC node and the receivers is in case of the Release 99 dedicated bearers 100 ms and in case of the shared HSPA bearer 10 ms.

The Georgia Tech Network Simulator (GTNetS) [102] is used for the evaluations. Most evaluation excercises are repeated several time in order to average the results.

To create the missing data distribution for the receivers, we have used a (19, 13) Reed Solomon FEC code. We have chosen a group size of 75000 receivers, which should all receive a 3 Mbyte file via MBMS download completely. The application layer FEC further protects the transmission and adds a protection overhead of 46.1% to the transmission session.

An inverse exponential distribution function with a mean of 0.01 is used to generate RLC block error rate distribution (see Figure 6.3b). Each MBMS receiver (client index) gets a radio block error rate assigned using the BLER function. This models the different positions and situations of each receiver in the cell. There are only a relative low number of receivers, which experience a high block error rate.

IP packets of 556 Byte sizes (incl. IP/UDP/FLUTE packet headers) are used for evaluation which are smaller than the RLC block size. The RLC block size is 640 Byte which corresponds to 80 ms TTI for a 64 kbps MBMS bearer. With this configuration, this may result in up to three IP packet loss events upon one link layer block loss event. The error propagation from RLC block error to IP packet error is described and evaluated in Section 4.1 in detail.



(a) Missing Data (b) Packet and Radio Block Loss Rates

Figure 6.3: Missing Data and Loss Rates.

The remaining packet loss after the FEC decoding is depicted in Figure 6.3b. The packet loss rate (PLER) is larger than the block loss rate (BLER), since a single block loss may cause one or more packet errors. This behavior is also explained in Section 4.1. The application layer FEC code reduces the packet loss rate again.

After the application layer FEC correction, 9694 receivers on average are not able to reconstruct the received file correctly. Thus 9694 receivers must use the file repair services. Around 63% of those unsatisfied receivers (i.e. 6040 receivers) require less than 10 packets or in other words less than 5 kByte of data to successfully recover the 3 MByte file. In total, 60.8 Mbyte of data need to be resent during the File Repair operation. Figure 6.3a depicts the distribution amount of missing data over all receivers.

The Sequential Delivery Time *SeqDelT* for all following evaluation exercises is 511.5 sec.

We have created a scenario where all receivers fetch the same amount of data for a comparison with the file repair data. We have shared the amount of missing data equally among the receivers. For a fair comparison, we have divided the 60.8 MByte of missing data between the 9694 receivers. Each receiver fetches 6.58 kByte of data. The Sequential Delivery Time *SeqDelT* is still 511.5 sec, since we fetch the same amount of data from the file repair server.

All the parameters of the evaluations are summarized in Table 6.1.

6.3 PTP File Repair Evaluations

In this section, we evaluate the behavior and performance of the PTP file repair procedure. It is assumed, that the MBMS data transfer phase is completed and receivers have disabled MBMS reception for this bearer. All receivers, which must run the file repair procedure to successfully reconstruct the file(s), start the

MBMS data transfer phase (Phase 1)	
Unsatisfied Receivers	9694
Total amount of missing data	60.97 MByte
Missing Data Distribution A	
Number of served users	75000
FLUTE payload length	512 Bytes
RLC block length	640 Bytes
IP/UDP/FLUTE packet header	44 Bytes
File length	3 MByte
Forward error correction	RS(19,13)
Radio BLER distribution function	Inv. Exponential Distribution with $Mean = 0.01$
Missing Data Distribution B	
Radio BLER distribution function	Constant: 6595 Bytes

PTP file repair phase (Phase 2)	
File repair server link (c_{fr})	1 Mbps
TCP segment size	1460 Bytes
SeqDelT	511 sec
Link Latency	6 ms
Dedicated Radio Link Model	$c_{air} = 64$ kbps, 100 ms dedicated for each receiver
Shared Radio Link Model	$c_{air} = 640$ kbps, 10 ms shared among active receivers

Table 6.1: Summary of simulation parameters used in the test scenario.

repair procedure by randomly picking a wait-time out of the wait-time window Δ_{wtWnd}. Lower and upper bound of the wait time window are provided by the server in-band with the MBMS transmission.

In the following, we evaluate in particular the file repair procedure duration ($d_{frp,i}$) for individual receivers, which we define as the duration from the start of the file repair until the successful completion. Further, we evaluate the duration of the repair transmission session ($d_{frt,i}$), which we define as the duration between the first HTTP request from the client and the last data section received at the receiver side. The duration $t_{ptp} = \max_i(d_{frp,i})$ is the duration between the start of the file repair for the first receiver until the end of the file repair for the last receiver. These duration definitions are graphically depicted in Figure 6.4. In the evaluation figures, we have normalized the size of the wait-time window Δ_{wtWnd} with the Sequential Delivery Time (*SeqDelT*), which describes the lowest possible file repair duration.

In the following, we first evaluate a model, where each receiver uses a dedicated

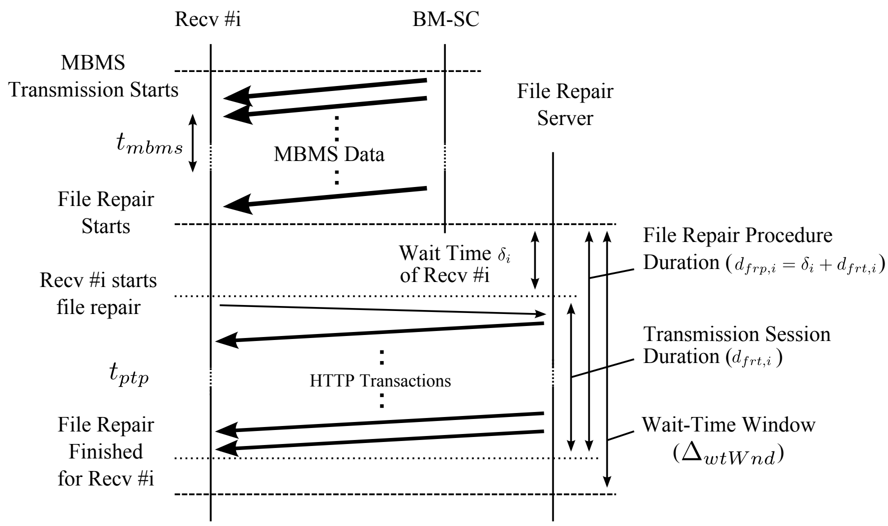


Figure 6.4: Definitions of evaluation parameters.

UMTS radio bearer with 64 kbps capacity for the file repair. After that, we use a shared HSPA bearer for the file repair.

6.3.1 File Repair with dedicated radio bearers

In order to compare the results of the different scenarios, we have first modeled a scenario similar to the model described in [100]. The packet error rates are here generated from an exponential distributed radio block error rate and the error propagation from radio block losses to IP packet losses. As explained and evaluated in chapter 4, each radio block loss may cause one or more IP packet losses.

In this section, we evaluate the file repair performance considering simplified dedicated UMTS radio bearers as defined for 3GPP Release 99. Each receiver requests a dedicated 64 kbps radio bearer (c_{air}) after the wait-time for the file repair has expired and the receiver becomes active. In our scenario, we have ensured that each radio base station serves only 10 receivers for a fair evaluation. All bearer requests are granted.

Later in this section, we compare the results with a scenario where all receivers are requesting the same amount of data, considering the same receiver group size and the same amount of missing data (in total).

The evaluation in Figure 6.5 shows the percentage of finished file repair procedures over time, parametrized with different wait-time window size as a factor of the sequential delivery time ($\Delta_{wtWnd} = x \cdot SeqDelT$, with x in range of $0.25, .., 2.0$). The x-Axes is the progress time, which is normalized with the sequential delivery time. All receivers start their file repair session within the wait-time window. The time to finish the file repair procedure is the time between the

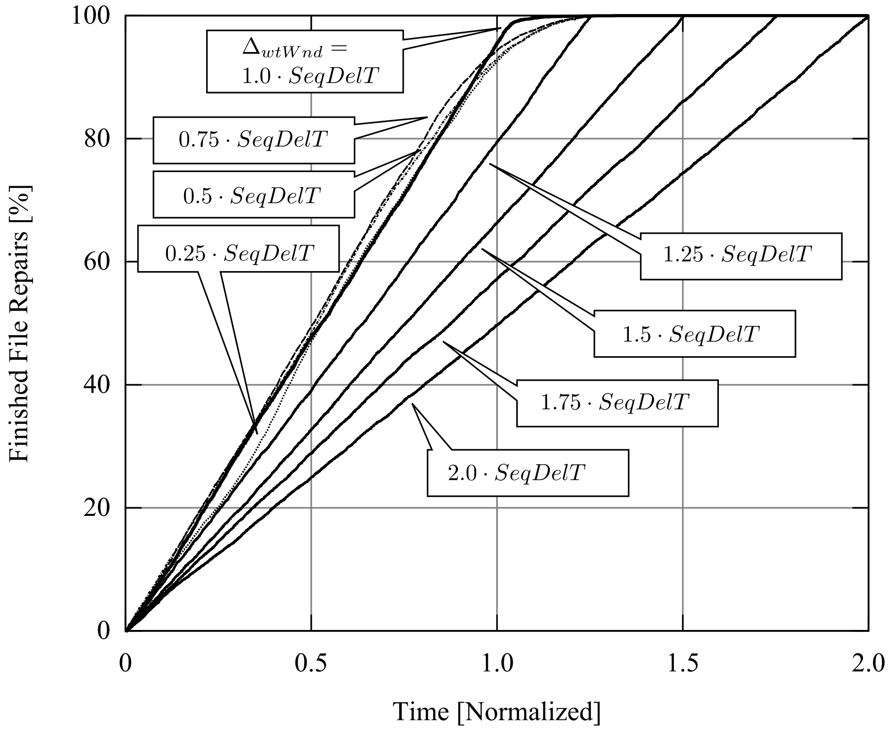


Figure 6.5: Finished file repairs over time, parametrized with different wait-time window sizes.

end of the MBMS data transfer (including the wait-time) and the completion of the file repair operation as shown in Figure 6.4.

If all clients would start the file repair operation simultaneously, each client would get a radio link capacity of about 1 Mbps/9604 user $\approx$ 0.103 kbps. The dedicated radio link would for be 99.8% unused. An optimal scenario from a dedicated radio link perspective would be, if on average only 15.6 users (1 Mbps / 64 kbps) use the file repair service simultaneously. In this case, the server link and the radio links of the clients would be optimally utilized.

In case of wait-time windows 1.25 times larger than the Sequential Delivery Time (i.e. $\Delta_{wtWnd} \geq 1.25 \cdot SeqDelT$), the duration to complete all file repair operations is only slightly longer than the actual size of the wait-time window. The curves of finished file repairs increase with a constant gradient.

In case of wait-time windows smaller than the Sequential Delivery Time (i.e. $\Delta_{wtWnd} < 1.0 \cdot SeqDelT$), the gradient of the curves is not constant. The duration to complete all file repair procedures is much longer than the size of the wait-time window. In particular when the wait- time window is $0.25 \cdot SeqDelT$, the gradient is lower for the first 50% of finishing receivers. This means that it takes longer to satisfy the same amount of receivers. For instance it takes $0.36 \cdot SeqDelT$ to satisfy 30% of receivers in case when the wait-time window size is $0.25 \cdot SeqDelT$. When the wait-time window is $0.75 \cdot SeqDelT$ it takes only $0.30 \cdot SeqDelT$ to

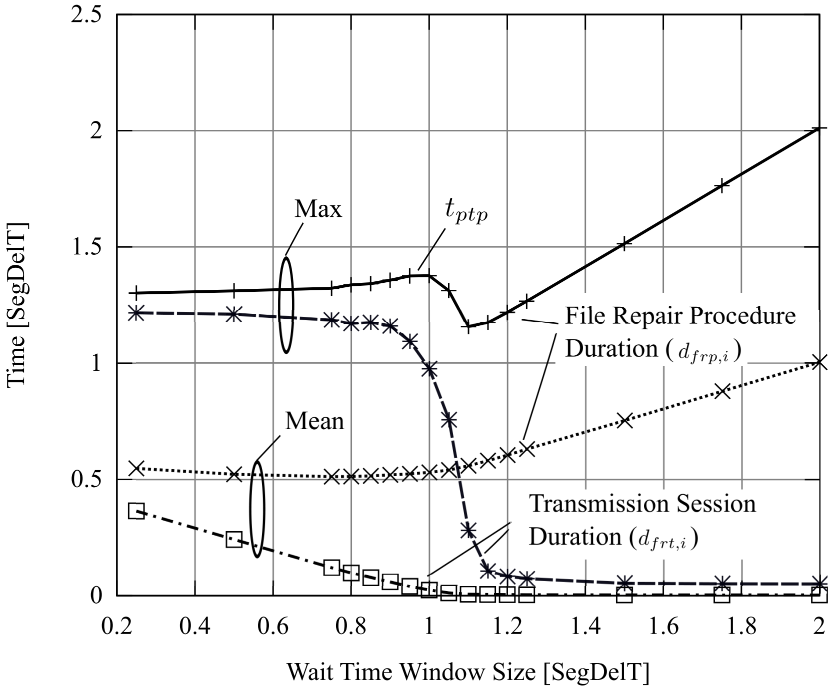


Figure 6.6: Maximal and mean file repair procedure completion duration over the wait-time window sizes (Δ_{wtWnd} normalized with $SeqDelT$).

satisfy the same share of receivers. When looking on 80% to 90% of finished receivers, then a wait-time window with size $0.75 \cdot SeqDelT$ provides the best performance. This behavior may be explained with the unequal missing data distribution. A high number of receivers only need a small amount of data, thus it is better to queue the requests a little bit in order to achieve a better link utilization. The average and the maximal file repair times are explained in detail as next.

Figure 6.5 clearly shows that the solution of spreading the repair operations in time leads to an overall better performance of the system. It allows receiving the file repair data at almost the available radio link capacity of 64 kbps. Furthermore, due to the lower link utilization of the link between the network and the file repair server, TCP experiences less often into retransmission time-outs.

$$\widehat{d_{frp}} = \max_i(d_{frp,i}) = t_{ptp} \quad (6.4)$$

$$\overline{d_{frp}} = \frac{1}{i}\sum_i (d_{frp,i}) \quad (6.5)$$

$$\widehat{d_{frt}} = \max_i(d_{frt,i}) \quad (6.6)$$

$$\overline{d_{frt}} = \frac{1}{i}\sum_i (d_{frt,i}) \quad (6.7)$$

Figure 6.6 shows the mean and the maximal durations of the repair procedure ($\widehat{d_{frp}}$, Eq. (6.4) and $\overline{d_{frp}}$, Eq. (6.5), respectively) of all receivers, measured from the end of the MBMS data transfer until the completion of the file repair procedure operation (cf. Figure 6.4 for the definitions). The maximal file repair procedure time is the same as the time, when 100 % of the receivers have finished the file repair procedure in Figure 6.5. Further, Figure 6.6 shows the mean and maximum transmission times ($\widehat{d_{frt}}$, Eq. (6.6) and $\overline{d_{frt}}$, Eq. (6.7), respectively). The transmission time is just the download time of the missing data, without the initial wait time. For larger wait-time window sizes (1.75 and $2.0 \cdot SeqDelT$), the maximal transmission duration goes down to $0.05 \cdot SeqDelT$, which is around 25 sec in our exercise here. The repair session of the receiver does not experience any congestion.

In order to achieve a sufficiently good confidence level, we have averaged the mean and the maximum values shown in Figure 6.6 over 50 repetitions.

The figure shows, that the file repair completion time increases for spreading intervals larger than the sequential delivery time (wait-time window $\Delta_{wtWnd} > 1.0 \cdot SeqDelT$). In these cases, the system is normally loaded. The increase of the maximal and mean repair procedure completion times results from the increase of the wait-time window, since the mean and maximum transmission times remain constant. In cases, where the wait-time window is larger than the sequential delivery time, the limiting factor is only the capacity of the radio link.

For wait-time windows smaller than the sequential delivery time, the bottleneck moves from the radio link up to the server link. Decreasing the wait-time window means to increase the simultaneous load on the file repair server. The file repair transmission sessions of too many receivers share simultaneously the link to the File Repair Server. The maximal transmission session length (see Figure 6.6) increases from below $0.1 \cdot SeqDelT$ to above $1.1 \cdot SeqDelT$ when the wait-time window gets smaller then the sequential delivery time. The average transmission session length is slowly increasing for decreasing wait-time window sizes.

$$t_{ptp} = \begin{cases} \sim 1.2 \cdot SeqDelT & \forall\ \Delta_{wtWnd} < (1.1 \cdot SeqDelT) \\ \sim \Delta_{wtWnd} & \forall\ \Delta_{wtWnd} > (1.1 \cdot SeqDelT) \end{cases} \quad (6.8)$$

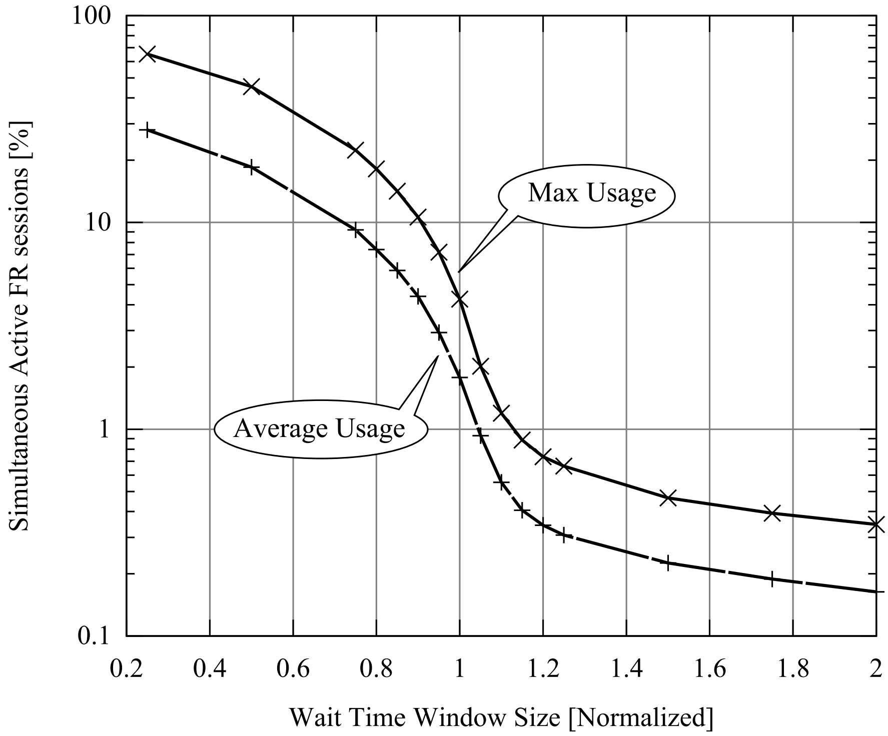
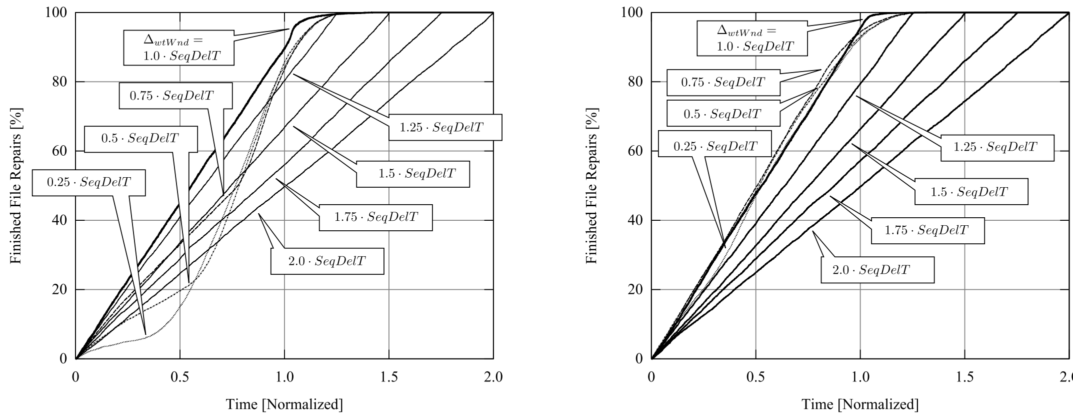


Figure 6.7: Simultaneous active sessions depending on the normalized wait time window size ($\Delta_{wtWnd}/SeqDelT$).

Eq. (6.8) summarizes the results of the maximal file repair procedure duration (t_{ptp}).

We must consider the shape of the missing data distributions when evaluating Figure 6.6: The distribution is generated from an exponentially distributed radio block error rates. Only a very small number of receivers are missing larger amounts of data. Those receivers with longer transmission sessions are in particular prone to packet losses. That is one key explanation for the different shapes of the maximal and the mean curves in Figure 6.6. That is also the reason, why the maximal transmission session duration remains almost constant with decreasing wait time windows sizes, while the average transmission session duration increases.

The load on the file repair server is depicted as simultaneous usage in Figure 6.7. The simultaneous usage is the percentage of simultaneously on-going file repair procedures. Please note the logarithmic y-axis. For wait-time windows sizes larger than $1.05 \cdot SeqDelT$, there is on average only less than 1 simultaneous file repair session on-going. This certainly means that the link between the file repair server and the system is not fully utilize. We need 15.6 simultaneously file repair sessions to fully utilize the link between the server and the system due to the limited radio link capacity of 64 kbps.



(a) Constant (b) From Exp. Distributed BLER

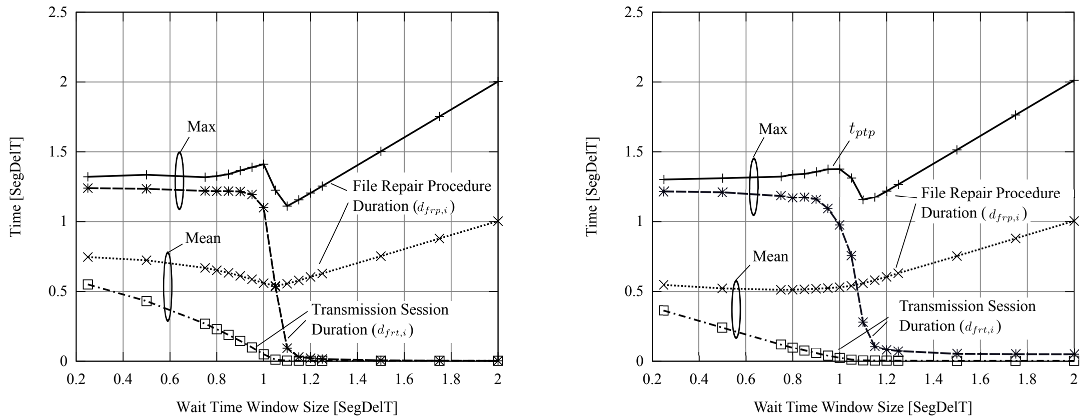
Figure 6.8: Comparison finished file repairs over time, parametrized with different wait-time window sizes.

When the wait time window size Δ_{wtWnd} becomes smaller than $0.75 \cdot SeqDelT$, then the link to the file repair server become frequently fully loaded. There is on average still less than 10 simultaneous sessions, but the maximal value is already above 20 simultaneous session. When the wait time window size becomes even smaller than $0.5 \cdot SeqDelT$, it is always fully loaded. More than 18 receivers are on average simultaneously active, leading to a fully utilization of the file repair server link, but an under utilization of each radio link.

6.3.2 File Repair equal request sizes

The missing data distribution, which we used for the previous evaluations, is generated from an inverse exponential distributed block errors in combination with a Reed Solomon RS (19,13) application layer FEC code. As depicted in Figure 6.3, the resulting curve of Missing Data after FEC is also kind of exponential distributed. The curve is characterized by a high number of receivers with low losses and a very small number of receivers with high losses. The shape of the distribution has an impact on the performance of the file repair. In reality, the shape of the distribution will depend on many effects and may be slightly different from transmission to transmission. To get a better understanding of the impact of the shape of the missing data distribution, we now compare the results against a constant missing data distribution. All receivers start request the same amount of data.

Figure 6.8 shows the comparison of the percentage of finished repair sessions over normalized time. Figure 6.8a (left) shows the percentage of finished sessions for a constant missing data distribution. Each receiver requests the same amount of data from the server. In this comparison, 9694 receivers need to request 6595 Byte (each) from the server, summing up to a total of 60.97 MByte.



(a) Constant (b) Exponential

Figure 6.9: File repair operation duration vs. time-spread configuration for different distributions of missing data.

The Sequential Delivery Time *SeqDelT* is again 511.472 sec. Figure 6.8b (right) is the same as Figure 6.5 and placed there to simplify the comparison.

The curve with the wait-time window size equal to $1.0 \cdot SeqDelT$ shows in all cases the best performance. When the wait time window size is smaller than $1.0 \cdot SeqDelT$ the percentage of finished file repair procedures increase with a much smaller gradient. There is a turn up point in the gradient, when all receivers have started their repair session (thus, when time is larger than wait time window size). For example, when the wait-time window size is $0.25 \cdot SeqDelT$, then the time progresses over 0.25.

The mean and maximal repair procedure durations are compared in Figure 6.9. The left figure (Figure 6.9a) depicts again the evaluation with constant repair request sizes over the users. There is almost no change on the maximal values. The decay at wait time window $= 1.0 \cdot SeqDelT$ is a little faster in the left figure. The average values for smaller wait-time windows sizes than *SeqDelT* are considerable different, reflecting the different shapes of the curves in Figure 6.8.

Figure 6.10 shows the simultaneous usage of the file repair service. The shapes of the curves are very similar. The left curve changes much faster from an average of above 10 simultaneous sessions at window size $0.9 \cdot SeqDelT$ to below 1 active session on average at $1.05 \cdot SeqDelT$. On the other side, the average simultaneous usage is already at 16 active sessions when the wait time window is $0.8 \cdot SeqDelT$.

6.3.3 File Repair with shared radio bearers

UMTS radio access technology has evolved over time and higher bitrate radio bearers are introduced. High Speed Packet Access (HSPA) is such a technology, which is build upon shared usage of the radio resources. The *free* radio capacity is shared among the data users.

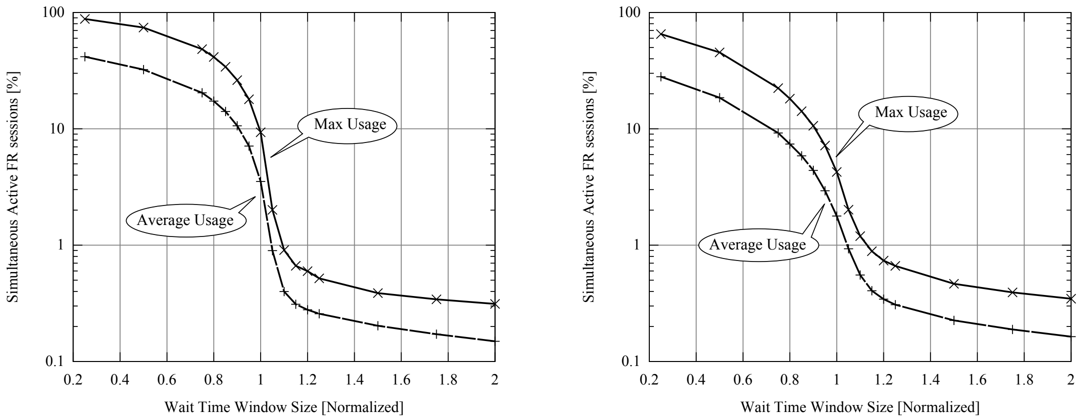


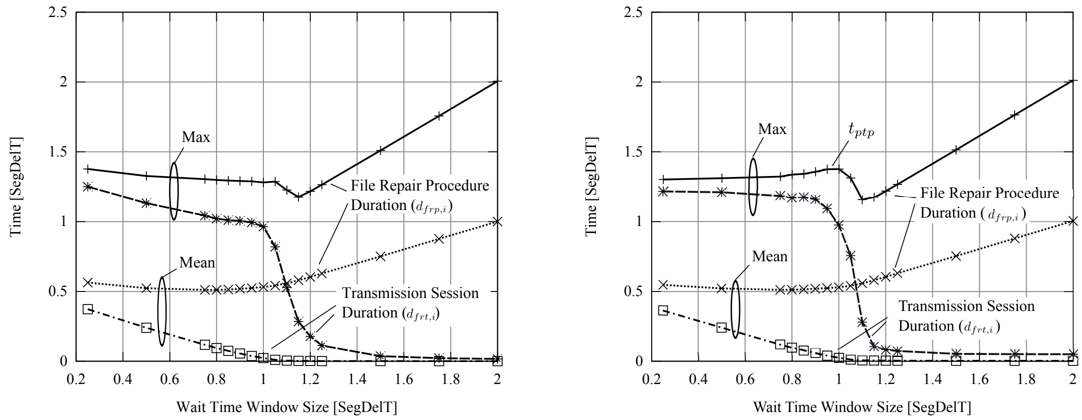
Figure 6.10: Simultaneous Usage for different distributions of missing data.

In order of a fair comparison between the system with dedicated and shared bearers, we said that the same amount of resources are available for both evaluations. In case of the model with dedicated bearers, each receiver gets its fixed shared of 64 kbps, independent from any other active receivers in the cell. We have ensured that each radio cell only serves 10 receivers. Thus, we give the HSPA bearer only 640 kbps *free* radio resources to schedule the file repair sessions.

Figure 6.11 depicts the mean and the maximal file repair procedure and transmission session durations. The system model with the shared bearers are depicted on the left. The system model with the 64 kbps dedicated bearers are depicted on the right. The missing data distributions is generated from the exponential distributed BLER in combination with the Reed-Solomon FEC code as depicted in Figure 6.3.

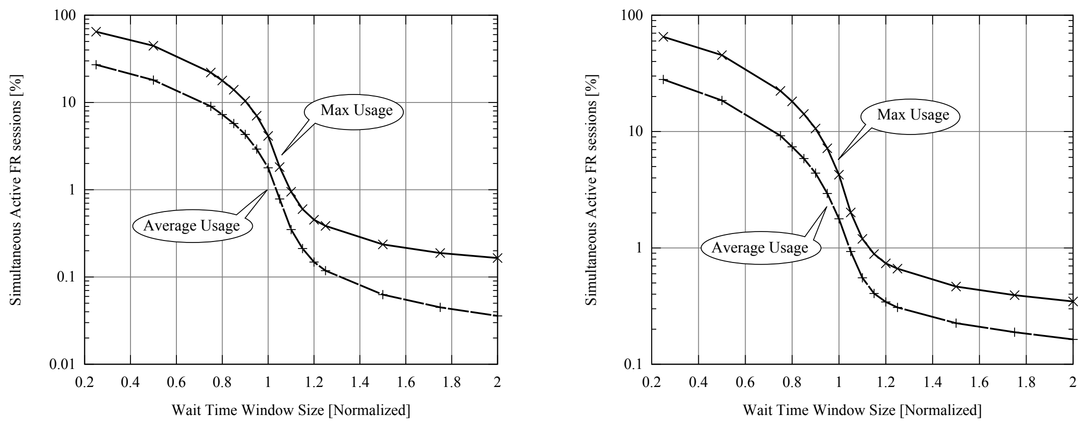
Interesting is in particular the comparison of the maximal transmission session duration ($\overline{d_{frt}}$, dashed line) in Figures 6.11a and 6.11b. It has in both figures almost the same starting point at transmission duration $\Delta_{wtWnd}(0.25 \cdot SeqDelT) = 1.2 \cdot SeqDelT$, when using a very small wait-time window size of $\Delta_{wtWnd} = 0.25 \cdot SeqDelT$. For increasing wait-time window sizes, the transmission durations decrease to a value of $1.0 \cdot SeqDelT$ in the left figure, while it remains almost constant in the right figure. Then, the transmission duration decrease similarly in both figures, when the wait-time window size gets larger than $\Delta_{wtWnd} > 1.0 \cdot SeqDelT$.

The bottleneck link is for wait-time window sizes up to $1.0 \cdot SeqDelT$ at the file repair server. For larger wait-time window sizes, the transmission duration is only limited by the available radio capacity for the receivers. Here, we also see a further decrease of the max transmission duration when the wait-time windows sizes is larger than $1.2 \cdot SeqDelT$. Receivers connected via the shared radio bearer may use a radio link bitrate of up to 640 kbps. Transmission durations for larger



(a) Shared Radio Bearers (b) Dedicated Radio Bearers

Figure 6.11: File repair operation duration vs. time-spread configuration.



(a) Shared Radio Bearers (b) Dedicated Radio Bearers

Figure 6.12: Simultaneous Usage.

wait-time window sizes remain constant in the right figure, since receivers can only use their dedicated 64 kbps link bitrate.

This behavior is also visible in the percentage of simultaneous active sessions as depicted in Figure 6.12. Here, we have less than 0.1% simultaneously active sessions when the wait-time window size Δ_{wtWnd} is larger than $1.2 \cdot SeqDelT$. Receivers fetch their data in the shared bearer scenario (left) much quicker than with dedicated bearers, leading to much lower simultaneous usage.

6.4 Conclusions of this chapter

This chapter focused on the detailed evaluation of the point to point (PTP) file repair procedure, which is defined for reliable file delivery. The PTP file repair service is offered after the MTCH transmission and allows receivers to request additional data in order to successfully reconstruct files. According to the MBMS standard, the repair load can be spread in time in order to protect the file repair server and intermediate network components against overload conditions. We have already seen in Chapter 5 that we shall use the file repair procedure to minimize the needed transmission energy for the reliable file delivery session.

In order to determine and compare the different load scenarios introduced by the file repair, we have defined the *Sequential Delivery Time* ($SeqDelT$). It is the minimal duration to transfer all file repair data over the link between the file repair server and the network. This link becomes the bottleneck link when too many simultaneous receivers request data. Using the sequential delivery time to normalized wait-time window size ($\Delta_{wtWnd}/SeqDelT$) allows us to compare different scenarios independently from the absolute requested data and the used link bitrate to connect the file repair server with the network.

Spreading the repair load within the wait-time window (Δ_{wtWnd}) is one overload prevention scheme defined by the MBMS standard. In principle, the wait-time window should have the same duration as the Sequential Delivery Time (thus $\Delta_{wtWnd} = SeqDelT$). However, when the distribution of missing data is like an exponential distributions (i.e. high number of receivers with low amounts of missing data), a slightly smaller wait-time window leads to a better performance. A higher percentage of users finish their repair service faster. When the wait-time window becomes larger than the Sequential Delivery Time, the capacity of the radio link becomes the bottleneck.

The maximal file repair procedure duration (t_{ptp}) is approximately 1.2 times the Sequential Delivery Time ($SeqDelT$) for all wait-time windows smaller than $SeqDelT$. When the wait-time windows is smaller than $SeqDelT$, then the link to the file repair server is overloaded leading to an underutilization of radio resources. The radio links cannot be fully utilized since the download bitrate is limited by the link to the file repair server. This dependency on the amount of missing data is course not nice to configure the file repair service for real-time services.

PTP file repair may use any 3GPP unicast bearer service. We have evaluated the performance of the file repair for dedicated bearers (Rel 99 bearers) and also for shared bearers (HSPA). The performance of the file repair is very similar for wait-time windows smaller than the sequential delivery time. Here, the link to the server is the prime bottleneck and a higher down link capacity does not improve the situation very much. However, radio resource utilization may be much better than with dedicated bearers, since other services may use the free transmission capacity. Thus, when receivers are active but not receiving data, the radio resources may be used by others services.

In a real system, the configuration of the file repair service (e.g. the wait-time window size) is provided in-band with the file delivery data of the first transmission phase. The server needs to estimate the repair load on the file repair servers before the MBMS data transfer, in order to provide the configuration in-band with the MBMS data. When key parameters like the MBMS transmission power and the geographical broadcast area are not changed, it is possible to estimate the configuration from earlier MBMS transmissions. However, the results show that a rough estimation of the sequential delivery time thus of the amount of necessary repair data, as an estimation error of even 25% leads only to less then 10% differences in the file repair procedure completion time.

7

Combined PTP & PTM File Repair

The MBMS specification defines three reliability increasing methods, namely PTP file repair, PTM file repair and Application-Layer FEC (AL-FEC). All reliability procedures are introduced in Section 2.4.3. PTP file repair is already evaluated in Chapter 6 and AL-FEC in Chapter 5. This chapter will discuss the PTM file repair, and how to use it. We will first discuss the pros and cons of PTM file repair against PTP file repair and then evaluate two realizations in detail.

For the PTP file repair, we have in particular seen the dependency to the amount of missing data. The more data is missing, the longer the PTP file repair takes. One major issue of the PTP file repair is, that it is configure blindly, without knowing the actual file repair load. So, it may easily happen that the link to the file repair server is fully loaded. It may also easily happen with a too conservative PTP file repair configuration that the session takes too long.

The PTM file repair allows sending file repair data to receivers. It is very likely that each receiver is missing different part of the source file. Thus, the PTM file repair provides best data, which can be used by every unsatisfied receiver with a high probability.

When using only AL-FEC, we cannot provide 100% reliability. We have seen the dependency between needed transmission resources and amount of AL-FEC in Chapter 5. It may always be possible that a receiver has not received a sufficiently high number of symbols to recover the source file. MBMS is only efficient, when many receivers are interested in the transmission.

So, what can we do with PTM file repair? We can use it to provide further FEC redundancy to receivers efficiently. When for instance the PTP file repair is too loaded because of a too optimistic configure wait-time window, the PTM file repair can be used to distribute additional FEC data and reduce the file repair load. This requires that FEC is used for the MBMS transmission in the first place.

The two phases of an MBMS Push File Delivery session are depicted in Figure 7.1. The file repair is a post delivery procedure, meaning that it is always executed after the MBMS data transfer (Phase 1 in Figure 7.1). The maximal duration of an MBMS Push File Delivery (t_{pfd}) session is $t_{pfd} = t_{mbms} + t_{fr}$. Certain receivers with good reception conditions may finish reception already after the first phase. Some services like background updates of RSS feeds put not any requirements on the Push File Delivery session duration. Other services with real

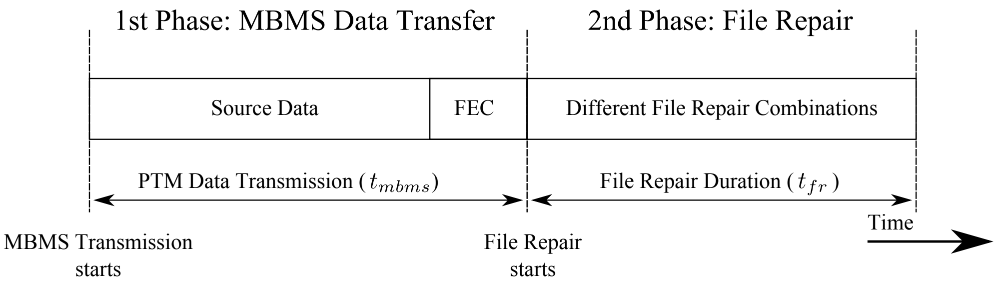


Figure 7.1: MBMS Push File Delivery Session.

time characteristics require a minimal file delivery duration to a high number of receivers.

The duration of the MBMS transmission (t_{mbms}) depends on the MBMS bearer bitrate (c_{air}) and also on the amount of additional FEC redundancy (N_{fec}). The duration of the file repair session (t_{fr}) depends on the amount of missing data, thus how much data must be provided during the file delivery. Further, the duration depends whether we use (as evaluated in this chapter) PTM or PTP file repair.

The PTM File repair procedure uses the MBMS bearers for sending additional data to receivers. The MBMS bearer may use PTP (unicast) or PTM (*true* Broadcast) radio bearers for transmission. Note that a PTM file repair session on MBMS may use PTP radio bearers when the interest in the cell is not sufficiently high. The principles of MBMS bearers and the used radio links are described in Section 2.2 and Section 2.3.1.

This chapter is structured as following: We first discuss in general the motivation and needs for combining PTP and PTM file repair. There may be several different possible combinations and we should first understand, why we should combine PTP and PTM file repair. After that, two combined file repair realizations are presented, which are then evaluated from a qualitative perspective. The conclusions of the combined PTP and PTM file repair is presented at the end of the chapter, including some usage recommendations.

7.1 Motivations for Combined PTP and PTM File Repair Procedures

The PTP file repair procedure was evaluated in the previous chapter. We have seen that the repair load on the server increases very much, when the configured wait-time window is smaller than the Sequential Delivery Time (*SeqDelT*). The radio links are not fully utilized anymore, since the link to the file repair server is overloaded. The load on the PTP file repair depends on the file size (amount of source data for MBMS transmission sessions), the target receiver group size and the individual block error rates during reception. The error rate during reception is not known before the transmission, thus the configuration of the PTP file repair procedure is based on estimates of the reception conditions.

The idea of the PTM file repair is to use again an MBMS bearer for distributing further FEC redundancy data, which is helpful for all unsatisfied receivers. Since Raptor FEC can produce an arbitrary high number of FEC symbols, it is straight forward to produce new FEC symbols. But one major issue of the MBMS system is that the system does not always know precisely the number of interested receivers per cell. Receivers do not issue a *leave indication* to the network when leaving the MBMS reception. Such a leave indication is not designed into the radio procedures, since it could overload the radio uplink (similar to the *Acknowledgment Implosion* problem). Consequently, the network does not have precise information about the actual number of receiving terminals. In particular during the end of an MBMS transmission session, terminals drop out of the reception when sufficient data are received for reconstruction.

Recall that using the MBMS PTM bearer is only more efficient than PTP bearers, when there are interested receivers in the cell (cf. Section 2.2.1. Using MBMS MTCH channels for PTM file repair without sufficient interest is a waste of radio resources and transmission energy. The radio resources could be used by other services and users.

MBMS MTCH reception is possible in *Idle Mode*, meaning that the terminal does not need to inform the network about its interest in the reception and do not have a control session established. Running the *re-counting* procedure is possible, but it does not instantaneously provide feedback. The receivers are instructed to spread their uplink messages in time in order to avoid radio uplink overload.

One major outcome of Chapter 5 is that the addition of too much FEC overhead to the MBMS transmission session is actually reducing the system efficiency. We should not reduce the load on the PTP file repair service by simply adding more and more FEC redundancy to the MBMS transmission. This result is also applicable to the PTM file repair.

Figure 7.2 depicts the schematics of a set of file repair realizations. The duration of the MBMS data transfer (t_{mbms}) can be the same for the depicted realizations, but the duration of the file repair ($t_{fr,1}$, $t_{fr,2}$, $t_{fr,3}$ or $t_{fr,4}$) depends on the actual file repair scheme and its combination. The minimal UE capability requirements do not foresee a simultaneous support of unicast and MTCH bearers (vf. Section 2.3.3). Therefore, we do not consider any realization with overlapping PTP and PTM file repair.

We know Realization 1 (*PTP Only*) already from Chapter 6. The file repair duration $t_{fr,1}$ of the PTP file repair for a given amount of missing data depend on the wait-time window size and the bitrate of the link between BM-SC and the network (c_{fr}). As shown in Chapter 6, the minimal file repair duration for all terminals is equal to ≈ 1.2 times the Sequential Delivery Time (*SeqDelT*) of the missing data over the link c_{fr} which connects the file repair server with the network.

The main challenge of the *PTP only* realization for file repair is to dimension

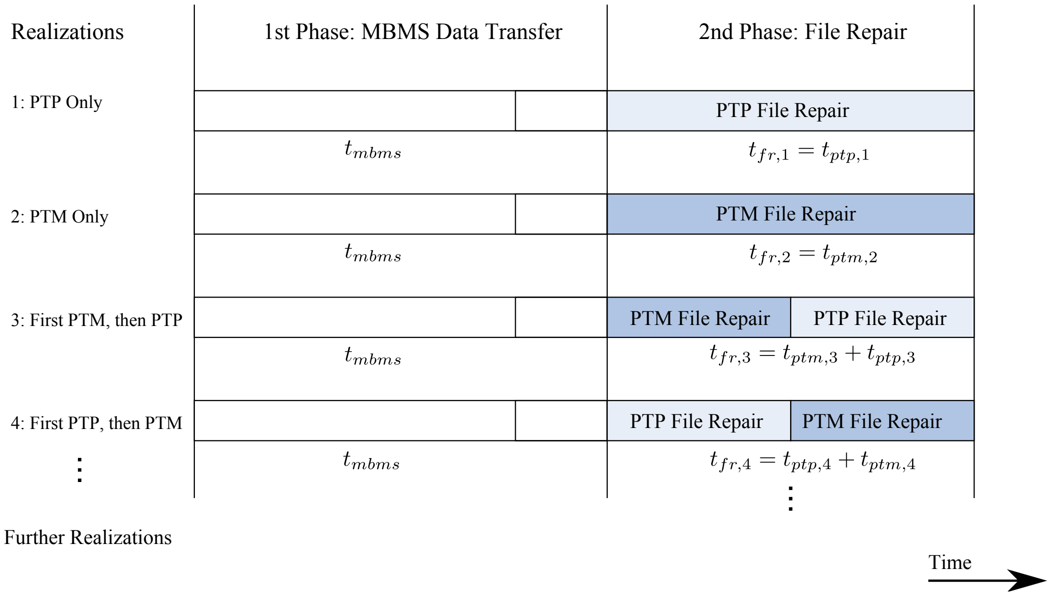


Figure 7.2: Schematic File Repair Realizations. Note that the file repair durations $t_{fr,1}, t_{fr,2}, t_{fr,3}$ and $t_{fr,4}$ are actually not equal.

the duration of the wait-time window Δ_{wtWnd}. In particular for real-time services, the session duration ($t_{mbms}+t_{fr,1}$) should be as small as possible. The value of the wait-time window size (Δ_{wtWnd}) is provided in-band with the MBMS data to all receivers, thus, it does not consider the current loss situation of the transmission.

PTM Only file repair is used in Realization 2. The BM-SC transmits additional file repair data using the MBMS bearer after the MBMS data transfer. One obvious question for this realization is, why the PTM file repair is separated from the MBMS data transfer. MBMS bearers are used in both phases. FEC redundancy is also provided at the end of the MBMS data transmission in Phase 1. In principle, one could regard such an *PTM only* file repair as providing additional FEC redundancy during Phase 1.

Realizations 3 and 4 are combined PTP and PTM file repair realizations (*First PTM, then PTP* file repair and *First PTP, then PTM* file repair). The only difference is, whether PTM or PTP file repair is executed first. One of the key issues with the PTM file repair is to determine the amount of additional FEC redundancy to be provided to all terminals. When starting with a PTM file repair (like in Realization 2 and Realization 3), the BM-SC does not have sufficient information to derive the needed amount of additional FEC data. However, when starting with a PTP file repair, the receivers are *invited* to provide feedback about missing data. The BM-SC can use this feedback for configuring the PTM file repair.

Other realizations are possible. In the following, we evaluate two example realizations is detail.

7.2 Evaluation of two Combining File Repair Realizations

In this section, we evaluate two realizations following the schematic Realization 4 (*First PTP, then PTM*). Additional FEC redundancy is provided using MBMS to all receivers, who have not received sufficient data to recover the source file (*Unsatisfied Receivers*). The main challenge is to determine the amount of additional FEC data for PTM file repair. Sending too much FEC data reduces again the system efficiency. If the MBMS bearer is used, the same data need to be sent to multiple terminals only once. The time for the file repair procedure can be largely reduced and the network resources may also be saved.

The load on the file repair server depends on the loss rate during the MBMS transfer, but the configuration for the file repair must be done before. A typical approach would assume that the load on the file repair server is approximately the same for preceding Push File Delivery transmission to the same target group. If the target receiver group is sufficiently large, then the average packet loss rate off all receivers should be approximately constant. However, it may happen that the load on the file repair server may be much higher than expected, e.g. because of faulty transmission equipment or other reasons.

When MBMS bearers are used for repair, the UEs do not need to send their repair requests to the server and the server sends the repair data only once via the bottleneck link. Thus there would be no need to spread the file repair requests load over time and the estimated configuration parameters will affect little to the file repair procedure. However, without any PTP file repair requests, the file repair server has no knowledge about the amount of missing data. If the packet loss rate during the initial MBMS transmission was much lower than anticipated, the PTM file repair is decreasing the efficiency.

Therefore, we combine the PTP and PTM file repair schemes. The PTM file repair is only used (here) in case of network overloads. We determine the amount of FEC packets sent during the PTM repair during the initial PTP file repair phase. There are different possibilities to combine the PTP repair with the PTM repair scheme. The main challenge for the PTM File Repair scheme is to get detailed understanding about the needed additional repair data for this given file repair operation. Here in this chapter, we evaluate two different schemes.

Figure 7.3 depicts different file repair realization combinations. The transmission session starts with the MBMS transmission session. Some FEC redundancy is already transmitted with the first phase to all receivers. The top most scheme is the *PTP only* file repair for comparison. *PTP only* file repair was extensively discussed in Chapter 6. We have seen in Chapter 6 that the maximal file repair duration (t_{fr}) is almost equal to the Sequential Delivery Time for wait-time windows small wait-time windows. The other two schemes are in focus for this chapter.

Combined Repair Scheme 1 (*PTP, Wait & then PTM File Repair*) starts the

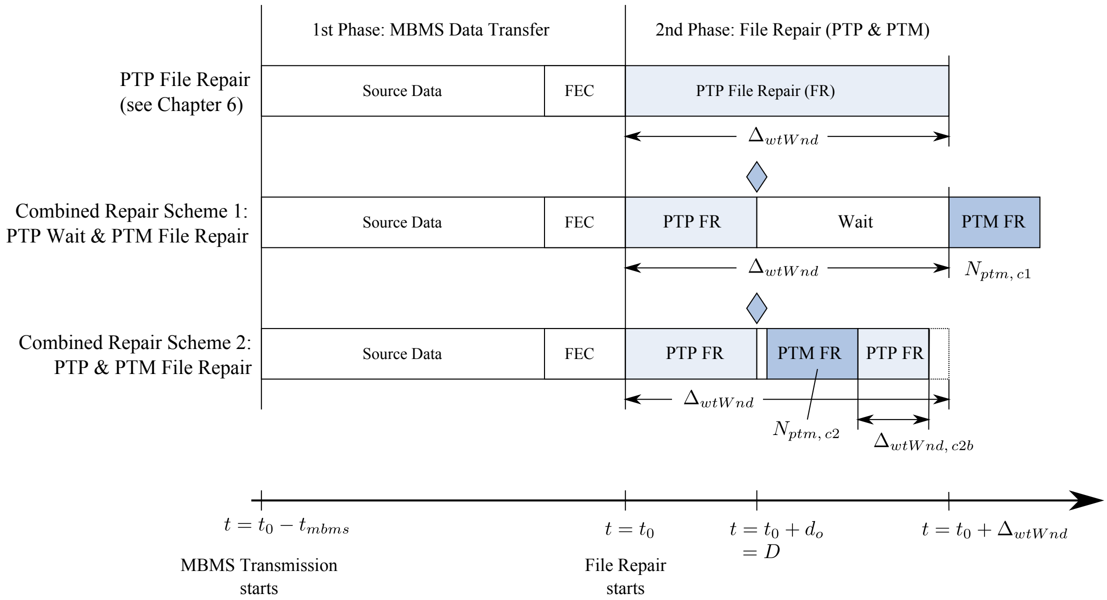


Figure 7.3: Combining PTP with PTM File Repair.

PTP repair procedure; the link to the file repair server is closely monitored. The file repair server detects at time $t = t_0 + d_o$ that the outgoing link is overloaded. The BM-SC decides at time D to start using PTM file repair. Note that the wait-time window Δ_{wtWnd} is much larger. The file repair server stops serving repair data and collects all repair requests. It may redirect all clients to the PTM file repair session, if a different MBMS bearer is used for PTM file repair. After all file repair requests are received ($t = t_0 + \Delta_{wtWnd}$), the file repair server has received exact feedback from the receivers about missing data. The BM-SC sends $N_{ptm,c1}$ additional FEC redundancy packets to all listening receivers.

Combined Repair Scheme 2 (*PTP, PTM & PTP File Repair*) also starts with the normal PTP file repair at time $t = t_0$. But the server starts the PTM file repair session immediately at point $t = t_0 + d_o$, when it has detected the overload on the link. The file repair server does not wait until the end of the wait-time window before starting the PTM file repair. The file repair server extrapolates the amount of missing data $N_{ptm,c2}$ from the received requests. The receivers suppresses the un-send repair request when the PTM file repair sessions is activated. Since the file repair server has only estimated the amount of missing data, it defines another PTP file repair sessions for the time after the PTM file repair. This second PTP file repair is configured with the wait-time window $\Delta_{wtWnd,c2b}$. Note that activating an MBMS bearer takes some time for the MBMS Notification of all receivers (in order of $d_p \approx 10\,\text{sec}$). Thus, the MBMS PTM session starts at $t = t_0 + d_o + d_p$, which is indicated by the white area between PTP and PTM file repair.

Both combined repair schemes start at time $t = t_0$ with a PTP file repair phase to collect feedback from the clients about the missing data. The amount of data for the succeeding PTM repair phase is based requested data during initial PTP file repair. In *Combined Repair Scheme 1*, the file repair server waits for all file repair requests before starting the PTM session. In case *Combined Repair Scheme 2*, the file repair server estimates the needed data based on a fraction of repair requests.

In both schemes, the load on the bottleneck link c_{fr} (see Figure 6.1 on Page 107), which connects the file repair server with the network is carefully observed. The bottleneck link is overloaded when the link queue is full and starts dropping packets. This is detected at time $t = t_0 + d_o$, where d_o depend on the actual load situation. When the link overload is detected, then all subsequent file repair requests are redirected to PTM repair. The network has an active procedure to handle *too small* wait-time windows.

In this section, we evaluate the influence of the first wait-time window size (Δ_{wtWnd} in Figure 7.3) on both combined repair schemes. Since the PTM file repair is triggered by an overload of the PTP file repair, we need to set the wait-time window smaller than the Sequential Delivery Time ($\Delta_{wtWnd} < SeqDelT$). The sequential delivery time ($SeqDelT$) is calculated as defined by Eq. (6.2) in Chapter 6.

In order to create an understanding of the influence of the wait-time window size, we set evaluate two scenarios with wait-time windows sizes $\Delta_{wtWnd} = 0.25 \cdot SeqDelT$ and $\Delta_{wtWnd} = 0.5 \cdot SeqDelT$.

The influence and dependencies of FEC redundancy on the resource usage and the file repair load is extensively evaluated in Chapter 5. The influence of the FEC code overhead is similar to the file length influence. When the transmission overhead becomes smaller, the FEC parity data becomes smaller, thus the FEC decoder is not as robust against high error rates. Naturally the amount of data, which needs to be repaired during the file repair procedure, increases. So the file repair procedure is influenced by the robustness of Raptor FEC code. We simulate multiple overheads, i.e. 5%, 7% and 9%, for different file repair procedures.

7.3 Evaluation Assumptions

In the following, we are evaluating two combined file repair schemes. The same evaluation topology as used for the PTP file repair service is also used for the evaluation of the combined PTP with PTM file repair services. The used evaluation topology is depicted in Figure 6.1 on Page 107. The PTM file repair data is sent by the *MBMS Sender* function like regular MBMS data. We assume a duration for the MBMS Notification of 10 sec, which is needed to establish the MBMS bearers. All network elements between the file repair server and the receivers are prepared for the upcoming MBMS data transfer session with the notification phase. Also the terminals are notified about the starting MBMS session (*paging*).

Parameter	Value
Number of UEs	100,000
Radio Link Bit Rate (c_{air})	64 kbps
Link to File Repair server (c_{fr})	1 Mbps
Block Error Rate Distribution	Inv. Cumulative Exponential Distribution (Mean = 1%)
MBMS Notification Duration	$d_p = 10$ seconds
IP Packet Payload Size	$l_{pp} = 512$ bytes
IP packet size	$l_{ps} = 556$ bytes
File Length	1 MByte
FEC Overhead	7% (Ideal FEC)
Unsatisfied Receivers	1492
Sequential Delivery Time	$SeqDelT = 172$ sec

Table 7.1: Summary of simulation parameters used in the test scenario.

Further details about MBMS Notification can be found in Section 2.2.3.

The link c_{fr}, which connects the file repair server and the network, is the bottleneck link. The link is overloaded on purpose using small wait-time windows sizes, so that the number of concurrent file repair sessions is large. The measurements used for the decision point D of Figure 7.3 are taken from this link.

The parameters listed in Table 7.1 are default parameters used throughout our evaluations. If any parameter is changed in a particular evaluation scenario, it will be mentioned explicitly.

Only a single simulation iteration is evaluated in the following, since we are only interested in the general behavior and dependencies of the procedures and not any quantitative conclusions. Otherwise it becomes an optimization problem, with many different and possibly even contradicting cost functions. For instance, an emergency services may have the requirement to be delivered as fast as possible. But since this emergency service is very seldom, it is allowed to have a higher resource consumption than other services. Other real-time services such *Latest News from Football* have also strict time requirements, but possibly only for e.g. 95% of the subscribers.

7.4 Scheme 1: PTP, Wait & PTM File Repair

Combined Repair Scheme 1 is introduced in Section 7.2. The sequence of PTP and PTM file repair is depicted in Figure 7.3. The file repair starts with PTP file repair for both combined repair schemes. In this section, we evaluate the influence of the first wait-time window size (Δ_{wtWnd} in Figure 7.3) on both combined repair schemes.

One main difference between *Combined Repair Scheme 1* and *Combined Repair Scheme 2* is how to determine the number of additional FEC redundancy packets for the PTM file repair session. The approach for *Combined Repair Scheme 1* is to first wait until $t = t_0 + \Delta_{wtWnd}$, so that all receivers have sent their PTP file repair request. The BM-SC serves the first PTP file repair requests as described and evaluated in Chapter 6. The receivers randomly select a start time, so that the file repair requests are uniformly distributed in the wait-time window (cf. Section 2.4.2).

But when the BM-SC detects the overload on the link between BM-SC and the network (at time $t = t_0 + d_o$), the BM-SC stops serving the PTP file repair requests. Instead the BM-SC replies with HTTP redirect messages. The BM-SC does not interrupt any on-going file repair session, it finishes the already on-going PTP file repair sessions. The BM-SC determines the configuration for the PTM file repair from the remaining file repair requests.

The parameters for the PTM file repair session are determined as following: Let N_{UEmd} be the number of unsatisfied receivers, which must run the file repair to fetch missing data. Let $l_{miss,i}$ be the missing data of the ith unsatisfied receiver. Eq. 7.1 defines $N_{PTM,Scheme1}$ as the largest amount of requested repair data of the receivers 0 to $N_{(UEmd-1)}$.

$$N_{ptm,\,c1}\ [\text{packets}] = \frac{\max(l_{miss,0}\ , \dots\ , l_{miss,N_{(UEmd-1)}})}{l_{pp}} \tag{7.1}$$

When the PTM repair phase begins, the file repair server sends $N_{ptm,\,c1}$ repair packets with FEC data using MBMS to all the UEs. The packet payload size is l_{pp} (the packet payload size should be a multiple of the FEC Symbol size to avoid padding). Some of the UEs may receive more repair packets than needed, which are locally discarded. All the UEs receive sufficient repair data as requested after the PTM repair phase. It is assumed that no further packet losses occur for instance because reliable PTP radio bearer are used (cf. Section 2.2.1. When the last UE receives enough repair packets it needs, the file repair procedure ends.

7.4.1 Influence of the wait-time window on Scheme 1

In order to create an understanding of the influence of the wait-time window size, we evaluate two scenarios with wait-time windows sizes $\Delta_{wtWnd} = 0.25 \cdot SeqDelT$ and $\Delta_{wtWnd} = 0.5 \cdot SeqDelT$.

Figure 7.4 depicts the influence of different wait-time window sizes on the duration of the file repair procedure. The figure shows two graphs with the wait-time window set to 25% of *SeqDelT* and 50% of *SeqDelT* respectively. The solid curve Graph_{25} uses the for $\Delta_{wtWnd} = 0.25 \cdot SeqDelT$ while the dotted curve Graph_{50} is used for the larged wait-time window $\Delta_{wtWnd} = 0.5 \cdot SeqDelT$. The link to the file repair server for the smaller wait-time window ($Graph_{25}$) is earlier overloaded than the larger one. The reliability of the first MBMS data transfer is

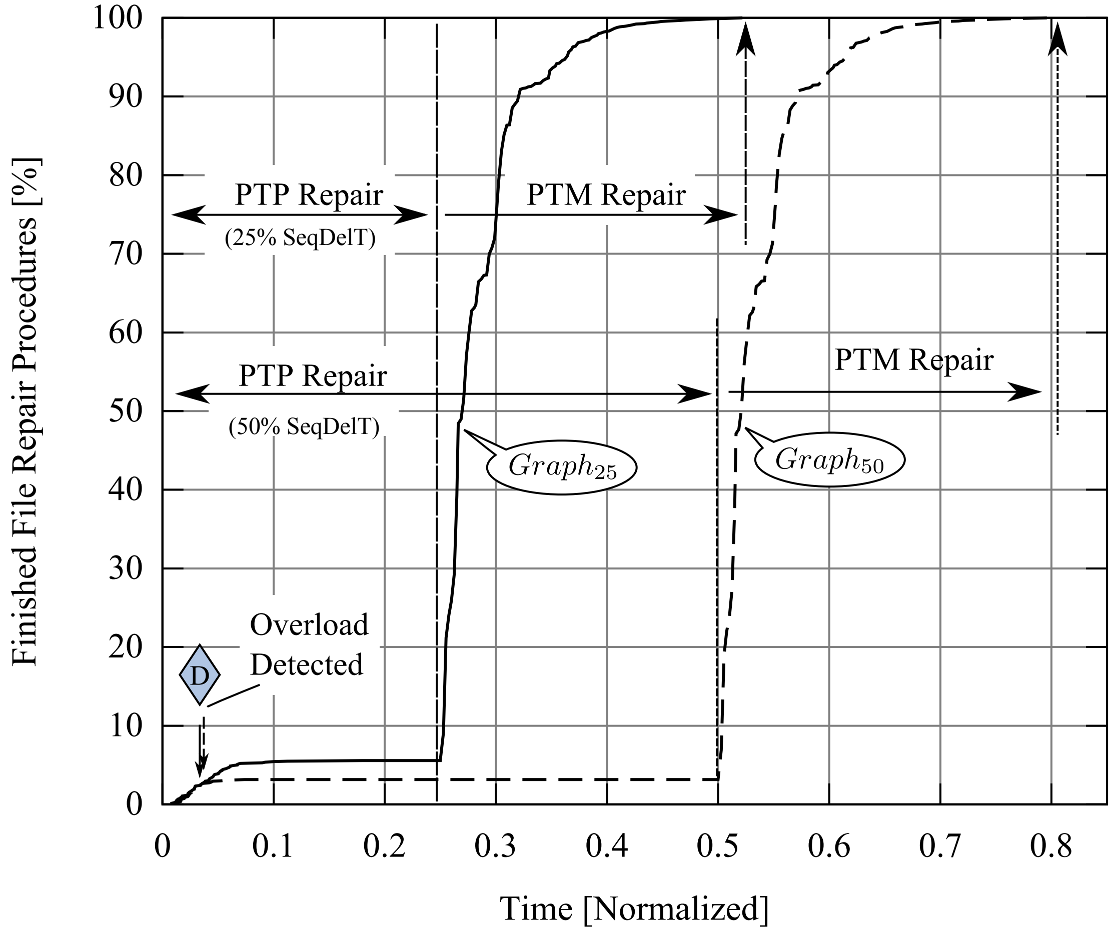


Figure 7.4: Cumulative amount of finished UEs in Scheme 1 over time. The Wait-Time Window Size (Δ_{wtWnd}) varies.

increased by adding 7% FEC redundancy to the transmission leading to a missing data ratio of $MDR \approx 0.21$.

In Figure 7.4 we see the influence of the wait-time window on *Combined Repair Scheme 1.* Both procedures provide sufficient data for all UEs. But due to the different wait-time window size, the procedure end time is quite different. Main reason is because the file repair server waits for the end-of the wait-time window before starting the PTM file repair. In the case that the wait-time window is 50% of *SeqDelT* (Graph$_{50}$), the file repair server waits twice as long as in case of the 25% *SeqDelT* wait-time window.

In Figure 7.4, Graph$_{50}$ finishes the file repair procedure almost $0.3 \cdot SeqDelT$ later than Graph$_{25}$. This time difference is almost wholly due to the wait time duration difference, i.e. the wait-time window difference, but also because receivers were satisfied already during PTP file repair. At the beginning of the file repair procedure, both curves encounter network overload at almost the same time. After the network overload is detected, the file repair procedure decides to redirect to PTM repair (Decision Point D as depicted in Figure 7.3). The repair server continues serving the already accepted file repair request. Hence both curves are still increasing after the decision to continue with PTM repair.

When using a smaller wait-time window ($\Delta_{wtWnd} = 0.25 \cdot SeqDelT$), the procedure has a little advantage compared to larger wait-time windows. A larger number of receivers have finished their file repair when the PTM repair session starts. This is because a higher number of UEs have started the file repair procedure although the BM-SC detects the overload almost at the same time $t = t_0 + d_o$. Consequently, the BM-SC continues serving them even after the BM-SC decided to switch to PTM file repair and $\approx 5.5\%$ of terminals have already finished their file repair. With the larger wait-time window of $\Delta_{wtWnd} = 0.5 \cdot SeqDelT$, only $\approx 3.1\%$ of the terminals have finished their file repair.This is also the reason, why Graph_{25} finishes earlier than Graph_{50}.

After the decision to switch to PTM repair, the file repair server *listens* to all subsequent repair requests and *memorizes* the requested data amount from each UE. After these repair requests are served, the server simply waits for the remaining repair requests to be sent to it. So the finished connections are not increased any more after the wait point and the curves remain horizontal. While the server waits for the end of the wait-time window the MBMS bearer is already allocated for the following PTM phase. After the wait-time window ends, the remaining repair requests from the UEs are all sent to the file repair server. The larger the wait-time window is, the longer the server must wait.

Once the wait-time window ends (at time $t = t_0 + \Delta_{wtWnd}$), the file repair procedure enters the PTM repair phase. In this phase, the largest repair request is used to determine the number of FEC packets $N_{ptm,c1}$as defined in Eq. (7.1). When the last UE receives its last repair packet, the file repair procedure ends.

7.4.2 Influence of FEC redundancy on Scheme 1

The addition of FEC redundancy during the MBMS data transfer influences the amount of repair data for file repair procedure, as we have seen in Section 5.3. Increasing FEC overhead means to also increase the needed transmission energy, since additional FEC overhead needs transmission energy to be transmitted as well. But with increasing FEC overhead, the amount of missing data is decreasing, thus, with increasing FEC, the load on the file repair system is decreasing and likely also the need for a PTM file repair scheme as also shown in chapter 5.

Figure 7.5 depicts the progress of the *Combined Repair Scheme 1* with 5%, 7% and 9% additional FEC redundancy during the initial MBMS transfer. The size of wait-time window Δ_{wtWnd} is for all three curves set to 25% of $SeqDelT$. The Sequential Delivery Time $SeqDelT$ is larger for lower FEC overhead, since more data is missing.

In order to compare the file repair schemes, we synchronized the start time

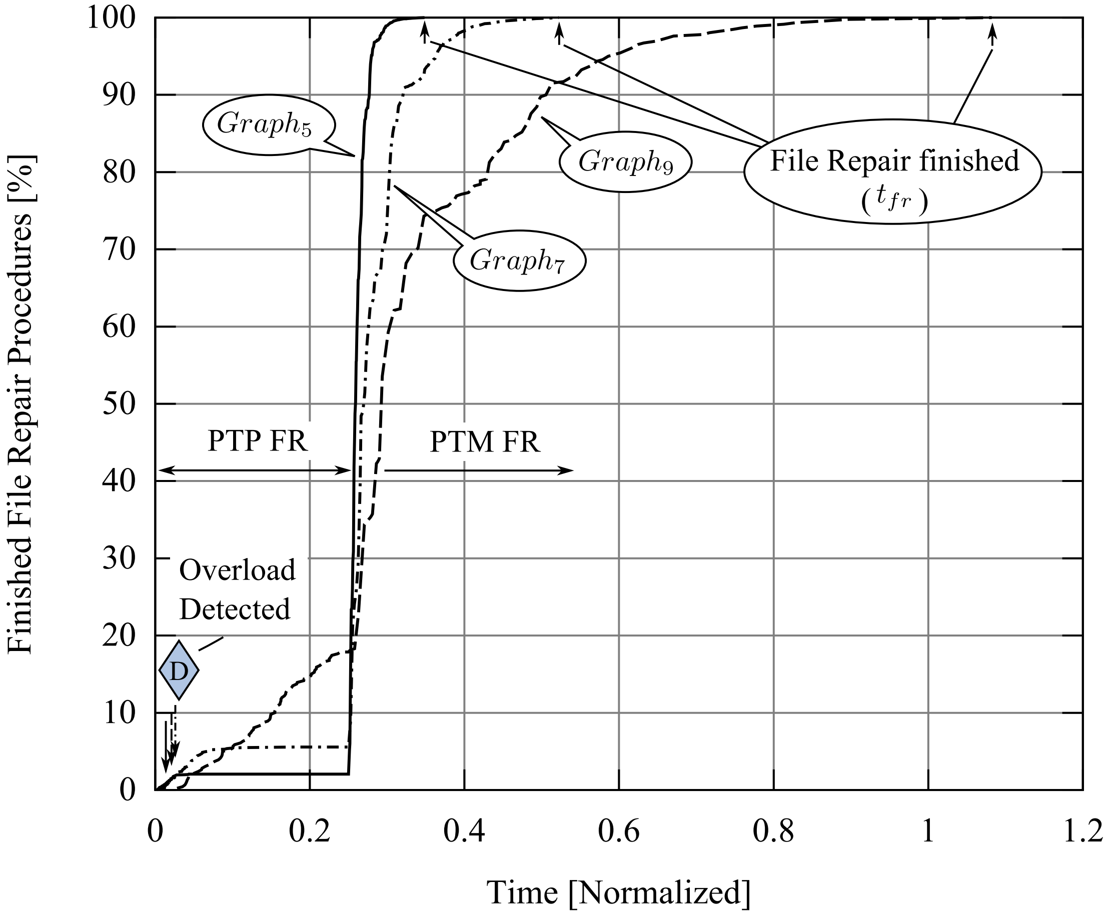


Figure 7.5: Cumulative amount of finished UEs in Scheme 1 over time. The Raptor FEC code overhead varies.

$t = t_0$ of the file repair. This is a penalty for $Graph_5$, since the MBMS data transfer t_{mbms} is 4% shorter than when adding 9% FEC overhead ($Graph_9$). All three curves use the same packet payload size $l_{pp} = 512$ Byte size.

Figure 7.5 shows that the amount of FEC redundancy is an important factor in our file repair procedure. As we have discussed in the previous chapter, the overhead has exponential relationship with the *SeqDelT*: when the FEC overhead decreases, the Sequential Delivery Time *SeqDelT* of the repair load increases proportionally.

In this *PTP, Wait plus PTM* file repair procedure (Scheme 1), the PTP repair phase uses a wait time window of size $\Delta_{wtWnd} = 0.25 \cdot SeqDelT$, so that the load on the file repair server is the same for all three procedures. The BM-SC decides in all three cases at almost the same time $t = t_0 + d_o$ to switch to PTM file repair. The BM-SC waits until $t = t_0 + \Delta_{wtWnd}$ to start sending additional FEC redundancy over MBMS. The start of the PTM file repair is well visible in

Figure 7.5. At time $t = t_0 + 0.25 \cdot SeqDelT$, all three curves change direction and the percentage of finished file repairs increase very rapidly.

It is interesting to note that the relative duration (normalized with the Sequential Delivery Time) is very different for the three curves. The amount of missing data to be repaired for $Graph_9$ is the lowest, due to the higher FEC overhead during the first MBMS transmission. Since we have normalized the durations with the Sequential Delivery Time, we can now see that $Graph_9$ performs worse than the other two graphs. Since the amount of missing data is very low, it takes almost as long to transmit the missing data once to all receivers using a 64 kbps MBMS bearer than running *PTP only* file repair with 1 Mbps link bitrate. Recall that the lower bound of *PTP Only* file repair is approximately equal to the Sequential Delivery Time. Thus, using *PTP Only* file repair would lead to a very similar performance, but without that steep increase of the finished file repair procedures.

Another interesting point is the amount of finished receivers just before the PTM file repair session starts (thus for $t = t_0 + 0.25 \cdot \Delta_{wtWnd}$): the larger the overhead is, the larger the percentage of finished receivers. The network overload always happens at the very beginning of the file repair procedure approximately at the same time. Since we use the same wait-time window size of $\Delta_{wtWnd} = 0.25 \cdot SeqDelT$, the load on the file repair server is the same. But the absolute number of unsatisfied receivers for the different configurations vary very much. The configuration which uses more AL-FEC (for instance $Graph_9$) results in a lower number of unsatisfied receivers. Thus the percentage of finished receivers of $Graph_9$ at time $t = t_0 + 0.25 \cdot \Delta_{wtWnd}$ is also larger than the one with smaller AL-FEC overhead.

7.4.3 Discussion of Scheme 1

The advantage of this *Combined Repair Scheme 1* is that the number of PTM repair packets $N_{ptm,c1}$ are determined precisely from the actual PTP repair requests. Another advantage is that the file repair is finished faster than with a *PTP only* scheme. Recall that we concluded in the last chapter that the minimal PTP file repair duration t_ptp is equal to the Sequential Delivery Time $SeqDelT$.

The obvious disadvantage is that the file repair server first needs to wait for the entire wait-time window Δ_{wtWnd}, before it can start the PTM file repair. Further, the PTM repair packets are also prone to packet losses when using MBMS MTCH channels as discussed and evaluated in Chapter 4 and Chapter 5.

But when the interest in the cells is low and if the system implements also MBMS PTP, then the PTM file repair session may use MBMS bearers with PTP radio links (as indicated in Figure 2.1 on Page 8). In this case, the PTM file repair data on the MBMS bearers use reliable radio links for the data (MBMS PTP radio links, cf. Section 2.2.1). The RLC acknowledged mode secures, that all PTM file repair packets are received correctly.

After the BM-SC decides to switch to PTM file repair (at time $t = t_0 + d_o$), the file repair procedure redirects to the PTM repair phase when the wait-time window ends. When the FEC overhead of the first phase is low, the largest repair packets number also increases, thus $N_{ptm,c1}$ is larger. Then the PTM repair phase with the larger value of $N_{ptm,c1}$ requires a longer absolute time to finish, but a shorter normalized time.

7.5 Scheme 2: PTP, PTM & PTP File Repair

The second realization of a combined file repair scheme with PTP and PTM file repair is based on the idea to shorten the time to start the actual PTM file repair session (which is one of the disadvantages of *Combined Repair Scheme 1*). But, when the PTM file repair session should start earlier, then the configuration of the PTM session needs to rely on parameter estimations. Consequently, there needs to be another PTP file repair scheduled after the PTM repair session in order to handle remaining terminals with losses.

Packet losses during PTM repair may anyhow impact the file repair scheme. When using MBMS MTCH channels for PTM repair, the transmission may be prone to packet losses as we have discussed earlier. This second repair is intended for terminals, which have, even after the PTM repair, not received sufficient data to recover the file(s).

The PTM repair starts immediately after the overload on the link to the file repair server is detected. The number of FEC packets sent during the PTM file repair is only a rough estimate, so terminals even without packet losses during PTM repair may still need additional data.

The sequence for *Combined Repair Scheme 2* is depicted in Figure 7.3. We will discuss in the following, how the BM-SC determines the number of PTM repair packets $N_{ptm,c2}$ and how the BM-SC configures the second wait-time window $\Delta_{wtWnd,\,c2b}$.

The BM-SC decides to switch to PTM file repair at time $t = t_0 + d_o$. The BM-SC stops serving new file repair requests after $t = t_0 + d_o$. Instead, terminals are redirected to the PTM file repair session. The MBMS notification phase takes here 10 seconds to inform all intermediate network elements and the terminals about the newly started MBMS bearer. The MBMS bearer is then ready at time $t = t_0 + d_o + d_p$ for the PTM repair session. The MBMS notification is described in Section 2.4.

The UEs, which have not yet sent their repair requests, cancel their PTP file repair procedure and wait for PTM repair data. The receivers which have already sent their repair requests, are continuously served with the PTP connection.

$$N_{ptm,c2}\ [\text{packets}] = \frac{\max[l_{miss,0}\ , \dots\ , l_{miss,M-1}]}{l_{pp}} \tag{7.2}$$

During the first PTP file repair procedure, the number of PTM repair packets for the MBMS bearer ($N_{ptm,c2}$) is determined very similar to the procedure in

Combined Repair Scheme 1: The BM-SC uses the feedback from the served file repair requests (between $t = t_0$ and $t = t_0 + d_o$) as a snapshot for the later file repair requests. The file repair requests are uniformly distributed in the wait-time window. In Eq. (7.2), the first M receivers have sent their file repair request. The file repair server finds the request for the largest amount of repair data $l_{miss,i}$ and derives the number of sent packets.

The second wait-time window $\Delta_{wtWnd,\,c2b}$ is set to $\Delta_{wtWnd,\,c2b} = 0.1 \cdot SeqDelT$. The configuration information for the second file repair is provided in-band with the PTM file repair data.

7.5.1 Influence of the wait-time window on Scheme 2

The evaluation of *Combined Repair Scheme 2* is started with a comparison of the wait-time window size influence. The same evaluation assumptions as for *Combined Repair Scheme 1* is selected. The performance of the *Combined Repair Scheme 2* is depicted in Figure 7.6. The curve with the solid line ($Graph_{25}$) is the progress of a file repair procedure with a wait-time window of 25% of the $SeqDelT$. The dashed line curve ($Graph_{50}$) is the progress when the wait-time window of the first PTP file repair is set to 50% of $SeqDelT$. The wait-time window of the second PTP file repair procedure is set in both cases to $\Delta_{wtWnd,\,c2} = 0.1 \cdot SeqDelT$.

Figure 7.6 depicts the progress of the percentage of finished file repair procedures over normalized time. The shapes of the two graphs are very similar. So, different wait-time windows sizes have almost no impact on the progress of Scheme 2. Both graphs in Figure 7.6 are normalized with the Sequential Delivery Time. The BM-SC detects the overload in case of $Graph_{25}$ a little earlier than in $Graph_{50}$, because the smaller wait-time window produces a higher load on the file repair server.

After the BM-SC has detected the overload at time $t = t_0 + d_o$, the BM-SC starts establishing the MBMS bearer and stops serving new file repair requests. The network elements are informed about this event using the Session Start Procedure. Terminals are informed by notification on the radio interface. The UEs, which have not yet sent the repair requests, suppress their requests. As seen in *Combined Repair Scheme 1*, the percentage of finished file repair requests increase slightly more in case of $Graph_{25}$ before the PTM repair session starts at $t = t_0 + d_o + d_p$. More PTP file repair procedure are already started in case of smaller wait-time window sizes.

In Figure 7.6, the BM-SC of $Graph_{25}$ determines a smaller number $N_{ptm,c2}$ of repair packets than the other repair server. The unsatisfied receivers start their file repair session randomly within the wait-time window and the receivers with higher losses has not scheduled the start of the file repair during this initial time window. The BM-SC derives $N_{ptm,c2}$ from the first file repair requests as defined by Eq.(7.2). So the duration PTM repair of the 25% $SeqDelT$ procedure is much shorter than the 50% $SeqDelT$ procedure. As result, the $Graph_{25}$ curve satisfies

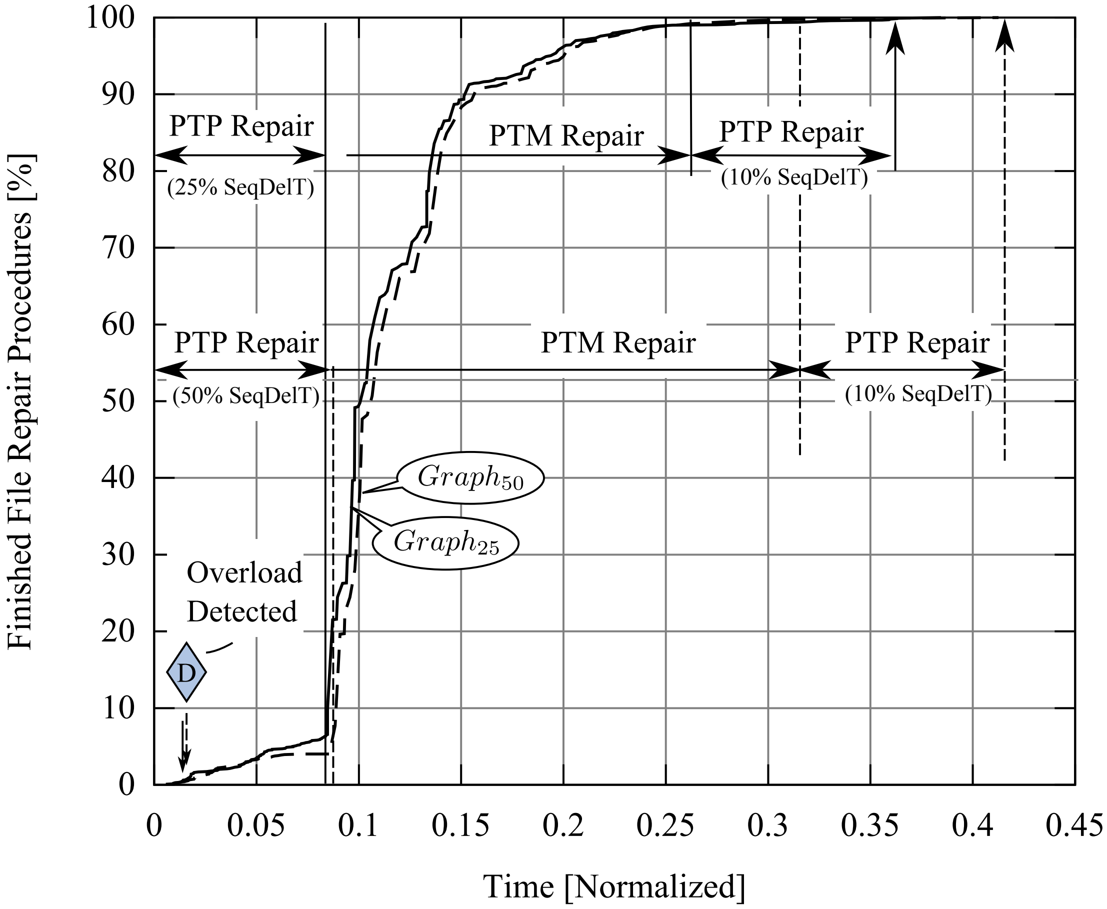


Figure 7.6: Cumulative amount of finished UEs in Scheme 2 over time. The wait time window of the 1st PTP repair varies.

$\sim 98\%$ of the receivers whereas $Graph_{50}$ satisfies even more than 99% of the UEs.

But this small superiority in satisfying more receivers with the PTM repair costs much more time and transmission resources. After the PTM phase, only few UEs have not received sufficient repair data for both curves. The second PTP file repair service for $Graph_{25}$ starts earlier than the other curve. Since we use the same wait-time window for the 2^{nd} PTP repair phase and $Graph_{25}$ starts the PTP repair much earlier than $Graph_{50}$, it ends also earlier than $Graph_{50}$. And more, the remaining repair UEs number $Graph_{50}$ is even smaller than the 25% of *SeqDelT* curve; the using of the 10% of *SeqDelT* wait-time window is very inefficient for the $Graph_{50}$. Note, that only a low number of terminals use the 2^{nd} PTP file repair service. It is likely that not the full wait-time window is used.

7.5.2 Influence of FEC redundancy on Scheme 2

The addition of FEC redundancy during the MBMS data transfer influences the amount of repair data for file repair procedure. The more the FEC redundancy, the less remaining data to be repaired. In this section, we evaluate the behavior of *Combined Repair Scheme 2* similar like for Scheme 1 in Section 7.4.2. Note, that the results in the following are normalized with the Sequential Delivery Time, which is different for the three Graphs. Therefore also the MBMS Notification Duration of $d_p = 10\,sec$ is normalized into three different values, namely 0.19 for 9% FEC; 0.061 for 7% FEC and 0.02 for 5% FEC.

In this file repair procedure scheme, the second wait-time window to 5% of $SeqDelT$, while the first wait-time window size is still set to 25% of $SeqDelT$. As we have discussed in Section 7.5, the second wait-time window can have essential influence on the total file repair duration when the $SeqDelT$ is relatively large. The simulation results are depicted in Figure 7.7.

Figure 7.7 depicts the progress of the finished repair procedures over the normalized time. The overload is detected in all three configurations almost at the same time $t = t_0 + d_o$ and the MBMS bearer for the PTM session is activated. The MBMS Notification duration is for all three graphs the same $d_p = 10\,sec$, but d_p is a different fraction of the Sequential Delivery Time and end up at different points on the normalized time axis.

The MBMS notification duration has an impact in the normalized representation. When the first MBMS transmission is protected with more FEC redundancy, then the Sequential Delivery Time to do *PTP only* is very small. Consequently, the notification duration takes already a larger share. As we have see in the evaluation of *Combined Repair Scheme 1*, PTM file repair is not efficient when the initial MBMS data transfer is already sufficiently protected with FEC redundancy. The Sequential Delivery Time ($SeqDelT$) is then already very small.

Like in Figure 7.5, the percentage of finished receivers is larger for the $Graph_9$ at time $t = t_0 + d_o + d_p$. The reason is the same as in Section 7.4.2. The PTM repair session of $Graph_9$ lasts also longer in the normalized representation as seen in Figure 7.5, although the BM-SC is sending less PTM repair data $N_{ptm,c2}$. There is almost no justification to use PTM file repair, since a *PTP Only* file repair would have finished in a similar time frame.

7.5.3 Discussion of Scheme 2

The advantage of Scheme 2 is that the PTM file repair session start immediately after the overload is detected. Thus, the file repair finishes faster than in Scheme 1.

There is only limited load on the 2^{nd} PTP file repair procedure od Scheme 2, since most terminals become satisfied with the PTM repair session. The percentage of finished repair procedures is in all three cases above 95%. This is an indication, that too much additional FEC data is provided with the PTM

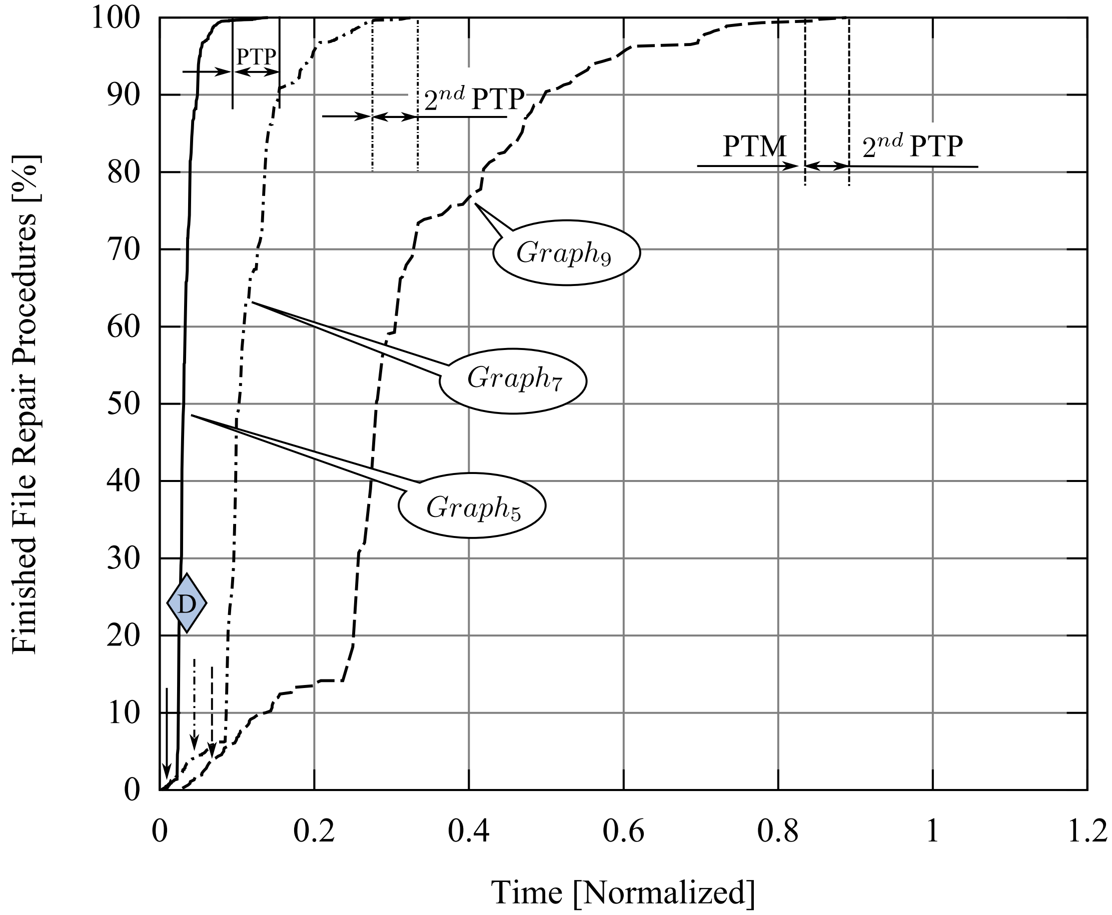


Figure 7.7: Cumulative amount of finished UEs in Scheme 2 over time. The Raptor FEC code overhead varies.

file repair session. Thus, decreasing the amount of additional FEC redundancy $N_{ptm,c2}$ would increase the load on the 2^{nd} PTP file repair and also increase the efficiency. This would be an improvement for Scheme 2. A too low load on the 2^{nd} PTP file repair is an indication for too much FEC redundancy during PTM file repair.

The main disadvantage of Scheme 2 is the complexity to determine the values for the amount of additional FEC $N_{ptm,c2}$ and also the dimensioning of the second PTP file repair. The efficiency of the MBMS Push File delivery procedure is reduced, when too much FEC is transmitted. Using an MBMS bearer becomes inefficient as soon as the interest per cell is too low (cf. Figure 2.1).

7.6 Conclusions of Combined File Repair

The PTM file repair scheme is defined for the MBMS Service Layer in [4], but no usage guidelines are given. In this chapter, we have evaluated different options for practical realizations of the PTM file repair. MBMS Service Layer in [4] does not recommend or require to provide additional FEC redundancy during the PTM file repair. However, one advantage of the Raptor FEC code is to produce additional FEC symbols *on-the-fly* from a given source block (cf. Section 3.5.3). Newly produced FEC symbols are beneficial for all unsatisfied receivers, because the probability of receiving a duplicated symbols is zero. When resending source data or earlier sent FEC redundancy, the efficiency of the PTM repair is decreased, since there is a probability that receivers discard the successfully received data.

The PTM file repair providing additional FEC redundancy may be used for different purposes. In case of real-time services, the push file delivery should be completed as quickly as possible. The acceptable amount of used resources (like transmission energy and transmission time) depends on the type of service and its value. Therefore, optimizing the file repair procedure requires a clear definition of a cost function, which can then be minimized. But such a cost function depends very much operator preferences for a given service. Without a clear definition, we can only evaluate the dependencies and behavior of the procedure.

The duration of *Combined Repair Scheme 1* depends mostly on the wait-time window size, since the file repair server waits until the end of the wait-time window before starting the PTM repair session. The file repair server *learns* the needed amount of data $N_{ptm,\,c1}$ for the PTM session during this time. Here, a shorter wait-time window is beneficial, since the terminals provide their needed amount of data $l_{miss,i}$ during a shorter period. In our evaluations, the procedure with the smaller wait-time window of 25% of *SeqDelT* showed the best performance for *Combined Repair Scheme 1*. By shrinking the wait-time window even further would speed up the duration.

In *Combined Repair Scheme 2*, the size of the 1^{st} wait-time window is not that influencing, since the file repair server immediately triggers the establishment of a PTM repair session after an overload is detected. The factor which determines mostly the total duration is the amount of sent redundancy packets $N_{ptm,\,c2}$ during the PTM repair session, thus the duration of the PTM session. This number $N_{ptm,\,c2}$ is chosen during the initial PTP file repair. We have selected the largest number of requested repair data to be $N_{ptm,\,c2}$. If this value is smaller, then the PTM repair session is shorter and the 2^{nd} PTP file repair is loaded more. An optimal value of $N_{ptm,\,c2}$ would leave just sufficient repair data to not overload the 2^{nd} PTP file repair. However, that is impossible to determine and configure before any transmission, unless the repair server waits for all file repair request as done in Scheme 1.

From the evaluations, *Combined Repair Scheme 2* is always faster in finishing all file repair request procedures than Scheme 1. Main reason for this is, that Scheme 1 first waits until the end of the first PTP file repair procedure in order to get a precise understanding of the missing data distribution. The larger the wait-time window of the first PTP file repair procedure, the longer the start of the PTM procedure is delayed. The progress of Scheme 2 is independent from the wait-time window size. The PTM file repair is started as soon as the MBMS bearer is active again.

The gains for using PTM file repair is higher for both schemes, when a larger amount of missing data needs to be repaired. We have seen in Section 7.4.2 and also in Section 7.5.2 that the performance of the procedures is decreasing with decreasing amount of missing data to be repaired. In both schemes, $Graph_9$ (using 9% FEC protection overhead during the initial MBMS transfer, leading to a lower amount of missing data after the MBMS transfer) showed a similar performance behavior than *PTP Only* file repair.

Thus, one can argue that the additional FEC redundancy, provided with the PTM file repair session using MBMS bearers, should be better directly added to the first MBMS transfer session, thus decreasing the amount of missing data and increasing the transmission time t_{mbms} (cf. 7.1). In particular for emergency and real time services the benefit of the service may justify a higher transmission cost of the first MBMS transfer session in order to reduce the amount of missing data for file repair. Sending the MBMS data with a higher transmission power reduced the block error rate, thus the amount of missing data. Adding FEC redundancy to the first MBMS transmission session increases the transmission duration and required more transmission energy, but also reduces the amount of missing data.

Combining PTP and PTM file repair schemes is clearly more complex than a *PTP Only* file repair scheme. The terminals and the BM- SC must switch between different operation modes. The BM-SC also need to properly determine the configuration parameters.

So, *where is the benefit of using PTM file repair at all?*

We have evaluated the PTM file repair under the assumption, that the first PTP file repair was configured with a too small wait-time window size, leading to an overload of the file repair server. It is impossible to consider any changing network condition when minimizing the required transmission resources for an MBMS Push File Delivery session. Thus, the combined PTM file repair schemes can be regarded as *fall-back* in order to handle unexpected conditions.

There is only limited needs for a *PTM-Only* File Repair Procedure. Here, the repair data could be combined with the FEC redundancy of the MBMS data transfer phase. One reason could be to trigger MBMS *re-counting* which is executed for MBMS Multicast and MBMS Enhanced Broadcast before the transmission session started. However, the system could also do an MBMS *re-counting* independently from the service layer.

8

Summary and Conclusions

This thesis evaluates the use of MBMS for distributing files. The MBMS service layer specification defines features to broadcast files over MBMS bearers *simultaneously* to multiple receivers. In that sense, the MBMS bearer service is a generic bearer service to distribute files or continuous streams simultaneously to multiple mobile receivers in a scalable way. MBMS Bearers may use *true* radio broadcast channels (i.e. MTCH) or unicast radio bearers (like HSPA), depending on the interest in the cell (and the implementation). But the MBMS service layer specification does not give detailed usage recommendations and configuration guidelines for MBMS. This thesis evaluates the performance and the behavior of different MBMS features.

The MBMS system can be used for *Mobile TV* like services (and was mostly discussed for Mobile TV in the past). One key issue for Mobile TV like services is that the user and the terminals are ready for receiving the streams. Mobile TV has been evaluated extensively for MBMS, DVB-H and other mobile broadcast solutions. Mobile TV services have not (yet?) taken off, although the technology has been available for quite some time already.

The advantage of *Push File Delivery* over MBMS is that only the terminal must be ready for receiving the content, but the user behavior must not necessarily change. Users may instruct their terminals to fetch all content for a given channel. Such a scheme is used for Podcasting for multimedia files, although *real* broadcast systems are today not used for Podcast distribution. The Podcast player typically fetches the different items in the background, independently from the user interactions. The user may later consume the fetched Podcast items.

Distribution of files using MBMS can also be used for other, not multimedia related services. For instance a provider may distribute new tariff information to its vending machines. Another example could be to distribute highly localized traffic information to cars. Using MBMS for such services allows the distribution of a higher amount of information in shorter time than with existing solutions.

MBMS is also discussed in context of Public Warning and Safety Systems. Mobile phones are an integral part of today's life. Most people have almost always a mobile phone at hand. A Public Warning and Safety System would enable governments to alert its population in a determined location about an imminent situation. Those services have often real-time requirements since it is necessary to reach a large share of population in short time .

8.1 Summary of Results

The thesis focuses on the system efficiency when using MBMS for delivering files to a high number of receivers. A number of different factors influence the system efficiency like the used transmission power and the added amount of application layer FEC (AL-FEC). Another, not that obvious factor, is the IP packet size.

In Chapter 4, we evaluate the *goodput* of the MBMS traffic channels. We defined the goodput as the fraction of received information bits over the error prone MBMS traffic channel and sent data bits. The IP packet sizes, the IP packet headers, the fragmentation of IP packets into the radio transmission blocks and, of course, the radio block loss probability influence the goodput of the bearer. Due to the fragmentation of, possibly variable length, IP packets into constant length radio transmission blocks, the use of shorter IP packets leads to a lower IP packet error probability. However, shorter IP packets lead to a lower goodput, since a larger share of the packet is used for the IP packet header.

We evaluated the error propagation from radio block losses to IP packet losses and finally to the goodput measure by modeling the radio link layer fragmentation mechanism. The model is configured with a constant radio block error probability in order to understand first the behavior of the system. Radio blocks are lost with a constant block loss probability and create one or more IP packet corruptions per radio block loss. We also found, that the alignment of IP packets to radio block losses lead to a preferred behavior, when the flow of MBMS download related IP packets are all of constant packet length. For download sessions, it is possible to ensure that all packets are of same length, since the files of known length are partitioned into encoding symbols of constant size. Alignment may lead to a goodput and capacity gain of more than 10% and up to 25%.

Constant block error rate distribution over a number of receivers in the same cell is not realistic. Therefore we used a radio network simulation tool in order to create more realistic block error probability traces for different locations in the cell using different transmission powers. The resulting distribution shows a low number of terminals in ugly reception conditions and a larger number receivers in moderate and good reception conditions. Then we combined the distributions of block error rates with the radio link layer fragmentation model in order to evaluate the packet error rate and the amount of missing data for an MBMS transmission in a mobile communication cell. FEC redundancy is added to the evaluate of file transmissions over MBMS. We evaluated the trade off between used transmission power and added application layer FEC for a given amount of missing data for all interested receivers.

The amount of added AL-FEC increases the transmission duration. The MBMS bearer resources (including the transmission power) is used for a longer duration. We used the measure of *transmission energy* for a fair comparison of the added amount of FEC and the used transmit power. Both, the transmit power and the time for which the resource is used at a given transmit power are considered by the transmission energy. Thus, the intention is not to use values of *absolute energy*, but rather the differences of used energy for comparison of different system configurations.

We found that a given target amount of missing data can be achieved by different combinations of IP packet sizes, transmission power settings and amount of AL-FEC. Keeping the amount of missing data constant for all receivers, we can reduce the transmission power by adding FEC redundancy at cost of increase transmission energy. The MBMS bearers must be used for a longer time to get the same amount of information across. The resulting amount of missing data must be recovered by the post delivery file repair. We also found that we can further reduce the needed transmission energy when aligning the IP packets with radio transmission block boundaries. This requires that the IP packets used for the MBMS transmission are all of same size.

The post delivery file repair is configured in-band with the transmission of MBMS data. To avoid well-known issues from reliable IP Multicast like the *Acknowledgment Implosion* problem, the point-to-point file repair allows spreading of the load in time and over different network elements. The system configures a *wait-time window* and the receivers draw randomly a start time out of the wait-time window. The system may also configure several PTP file repair server addresses and the receivers select randomly one. By doing this, the PTP file repair load is distributed uniformly in time (within the wait-time window) and over several network elements.

We found that the lower service duration of the PTP file repair service is almost equal to the *Sequential Delivery Time* of all missing data over the link between the file repair server and the system. This link actually limits the serving time and may even lead to an under-utilization of the radio resources. The PTP file repair is well dimensioned when the radio links of all active PTP file repair receivers and the link between file repair server and system is fully utilized. If the wait-time window is smaller than the Sequential Delivery Time, too many file repairs are active at the same time. Then the radio links of the individual mobile receivers is not loaded to the full extent and system resources are occupied unnecessarily long.

One way to speed up the file repair duration in case of too small wait-time windows is using PTM file repair. The PTM file repair allows the distribution of additional data over MBMS, optimally with newly produced FEC redundancy. With the distribution of new FEC redundancy the system ensures that the received data is of benefit for all receivers. However, PTM file repair increases the complexity of the MBMS system. It should be preferred to transmit more FEC data with the MBMS data during the first phase than to rely on PTM file repair.

8.2 Future Work

The Multimedias Broadcast and Multicast Services evolved with the introduction of Evolved Packet System (EPS) in the 3GPP specifications and also with the definition of the new LTE radio interface. Even with 3GPP Release 10, only the MBMS Broadcast Mode is defined for LTE. The MBMS service layer specifications and the service layer procedure are still applicable. MBMS in LTE is mainly defined as Single Frequency Network scheme, where MBMS services are broadcasted in multiple cells synchronously so that terminals may rely on radio signals from multiple transmitters. The used media rates are higher than for UTRAN MBMS as defined for 3GPP Release 6.

The general evaluation methodology, which is introduced and used in the context of this thesis, should be still applicable for evolved MBMS solutions.

A

Propagation of Radio Block losses to IP packet losses

The attached pseudo code is used to decide, whether an IP packet is lost or not. IP Packets are fragmented into one or more radio transmission blocks. If at least one of the radio transmission frames is corrupted, then entire IP packet is discarded by the RLC protocol.

The pseudo code is a corrected version of the psuedo code, submitted to 3GPP SA4 in [95].

The Algorithm takes the packet length of one IP packet as input. There is no need that the IP packet length (*pktLength*) is constant over iterations. The algorithm returns, whether or not this IP packet was successfully received as Boolean value. The algorithm keeps state about the last radio transmission blocks: The variable *lastBlockLost* indicates the state of the last radio transmission block. The value *true* means, that the last radio block was lost. The variable *spareOctets* keeps state about the free octests of the last transmission block. All radio frames are completely filled with IP packets. It is assumed that the radio layer receives the IP packets at constant rate.

The result variable *pktLost* is initialized with the state of the radio block (*lastBlockLost*) in Line 9, if there are spare octets to be filled.

Whether or not the radio transport block is lost is only determined at random every *blockLength* Byte. We used 640 Byte radio transmission blocks in our evaluations. If the IP packet spans multiple radio blocks, then there are multiple chances to loose the packet (in Lines 16 to 20). Otherwise, this is determined for the first IP packet parts in the radio block (Line 25).

If it is already determined, whether the radio block is corrupted or not and if the next IP packet also fits completely into the remaining octets of a radio frame, then this IP packet just takes the same loss state as the last IP packet (Lines 11 and 12).

The algorithm is called for each IP packet in the MBMS transmission. The number of IP packets and the packet size depend on the type of evaluation.

Algorithm A.1 IP Packet Loss depending on Radio Block Losses

```
inputs
   pktLength := IpPacketSize
outputs
   pktLost   {True or False}
globals spareOctets := 0   {Keep state }
globals lastBlockLost := false   {Keep state }

blockLength := 640  {RLC block length in Byte}
pktLost := (spareOctets ≠ 0) and lastBlockLost
remainingOctets := 0
if (pktLength ≤ spareOctets) then
   spareOctets := spareOctets − pktLength
else
   remainingOctets := pktLength − spareOctets   {Overlapping Byte to the next block}
   blocks := (int)remainingOctets/blockLength   {The packet may span several blocks}
   if (blocks > 0) then
      for i := 0 to blocks do
         pktLost := call transportBlockLost() or pktLost
      end for
   end if
   spareOctets := blockLength − (remainingOctets%blockLength)
   if (spareOctets = blockLength) then
      spareOctets := 0
   else
      lastBlockLost := call transportBlockLost()
   end if
   pktLost := pktLost or lastBlockLost
end if
return pktLost
```

B

Abbreviations and Acronyms

3GPP	3rd Generation Partnership Project
BLER	Block Error Rate
BLEP	Block Error Probability
BM-SC	Broadcast Multicast Service Center
BSF	Bootstrapping Server Functions
CDMA	Code Division Multiple Access
DRM	Digital Rights Management
DVB-H	Digital Video Broadcast - Handhelds
DVB-S	Digital Video Broadcast - Satellite
DVB-T	Digital Video Broadcast - Terrestrial
EDGE	Enhanced Data Rates for GSM Evolution
EPS	Evolved Packet System
ETSI	European Telecommunications Standards Institute
FDD	Frequency Division Duplex
FEC	Forward Error Correction
FLUTE	File Delivery over Unidirectional Transport
GBA	Generic Bootstrapping Architecture
GERAN	GSM EDGE Radio Access Network
GGSN	Gateway GPRS Support Node
GPRS	General Packet Radio Service
GSM	Global System for Mobile Communications
GTP	GPRS Tunneling Protocol
HLR	Home Location Registrar
HSPA	High Speed Packet Access
HTTP	Hypertext Transfer Protocol
IMB	Integrated Mobile Broadcast
LDPC	Low Density Parity Check Code
LTE	Long Term Evolution
MAC	Media Access Control
MBMS	Multimedia Broadcast and Multicast Services
MCCH	MBMS Control Channel
MICH	MBMS Notification Indicator Channel
MSK	MBMS Session Key
MTCH	MBMS Traffic Channel
MTK	MBMS Traffic Key
MUK	MBMS User Key

OMA	Open Mobile Alliance
PER	Packet Error Rate
PSS	Packet Switched Streaming
PTM	Point to Multipoint
PTP	Point to Point
RLC	Radio Link Control
RNC	Radio Network Controller
RTP	Real-Time Transport Protocol
UMTS	Universal Mobile Telecommunications System
UTRAN	UMTS Terrestrial Radio Access Network
S-CCPCH	Secondary Common Control Physical CHannel
SIM	Subscriber Identity Module
SDP	Session Description Protocol
SGSN	Serving GPRS Support Node
TDD	Time Division Duplex
TSG	Technical Specification Group
TTI	Transmission Time Interval
UE	User Equipment
UICC	Universal Integrated Circuit Card
XML	Extensible Markup Language

Bibliography

[1] H. Holma and A. Toskala, Eds., *WCDMA for UMTS*. New York, NY, USA: John Wiley & Sons, Inc., 2002.

[2] H. Holma and A. Toskala, *HSDPA/HSUPA for UMTS: High Speed Radio Access for Mobile Communications*. John Wiley & Sons, 2006.

[3] 3GPP TS 23.246, "Multimedia Broadcast/Multicast Service (MBMS); Architecture and functional description," 3rd Generation Partnership Project (3GPP)," Tech. Spec., 2006. [Online]. Available: http://www.3gpp.org/ftp/Specs/archive/23_series/22.246/

[4] 3GPP TS 26.346, "Multimedia Broadcast/Multicast Service (MBMS); Protocols and codecs," 3rd Generation Partnership Project (3GPP)," Tech. Spec., 2006. [Online]. Available: http://www.3gpp.org/ftp/Specs/archive/26_series/26.346/

[5] 3GPP TS 25.346, "Introduction of the Multimedia Broadcast/Multicast Service (MBMS) in the Radio Access Network (RAN); Stage 2," 3rd Generation Partnership Project (3GPP)," Tech. Spec., 2008. [Online]. Available: http://www.3gpp.org/ftp/Specs/archive/25_series/25.346/

[6] 3GPP TS 43.246 v6.10.0, "Multimedia Broadcast/Multicast Service (MBMS) in the GERAN; Stage 2," 3rd Generation Partnership Project (3GPP)," Tech. Spec., 2006. [Online]. Available: http://www.3gpp.org/ftp/Specs/archive/43_series/43.246/

[7] 3GPP TS 33.246, "3G Security; Security of Multimedia Broadcast/Multicast Service (MBMS)," 3rd Generation Partnership Project (3GPP)," Tech. Spec., 2006. [Online]. Available: http://www.3gpp.org/ftp/Specs/archive/33_series/33.246/

[8] 3GPP TS 23.060, "General Packet Radio Service (GPRS) Service description; Stage 2," 3rd Generation Partnership Project (3GPP)," Tech. Spec., 2006. [Online]. Available: http://www.3gpp.org/ftp/Specs/archive/23_series/23.060/

[9] 3GPP TS 29.060, "General Packet Radio Service (GPRS); GPRS Tunnelling Protocol (GTP) across the Gn and Gp interface," 3rd Generation Partnership Project (3GPP)," Tech. Spec., 2006. [Online]. Available: http://www.3gpp.org/ftp/Specs/archive/29_series/29.060/

[10] H. S. Thompson, D. Beech, M. Maloney, and N. Mendelsohn, “XML schema part 1: Structures,” World Wide Web Consortium (W3C), W3C Recommendation REC-xmlschema-1-20041028, Oct. 2004, available at http://www.w3.org/TR/xmlschema-1/. [Online]. Available: http://www.w3.org/TR/xmlschema-1/

[11] P. V. Biron and A. Malhotra, “XML schema part 2: Datatypes,” World Wide Web Consortium (W3C), W3C Recommendation REC-xmlschema-2-20041028, Oct. 2004, available at http://www.w3.org/TR/xmlschema-2/. [Online]. Available: http://www.w3.org/TR/xmlschema-2/

[12] M. Handley and V. Jacobson, “SDP: session description protocol,” Internet Engineering Task Force, RFC 2327, Apr. 1998. [Online]. Available: http://www.rfc-editor.org/rfc/rfc2327.txt

[13] Open Mobile Alliance, “OMA Push OTA Protocol (25-April-2001): WAP-235-PushOTA-20010425-a,” Open Mobile Alliance,” Tech. Spec., 2004.

[14] 3GPP TS 33.220, “Generic Authentication Architecture (GAA); Generic bootstrapping architecture,” 3rd Generation Partnership Project (3GPP),” Tech. Spec., 2006. [Online]. Available: http://www.3gpp.org/ftp/Specs/archive/33_series/33.220/

[15] J. Arkko, E. Carrara, F. Lindholm, M. Naslund, and K. Norrman, “MIKEY: multimedia internet keying,” Internet Engineering Task Force, RFC 3830, Aug. 2004. [Online]. Available: http://www.rfc-editor.org/rfc/rfc3830.txt

[16] 3GPP TR 25.803, “S-CCPCH performance for MBMS,” 3rd Generation Partnership Project (3GPP),” Tech. Rep., 2005. [Online]. Available: http://www.3gpp.org/ftp/Specs/archive/25_series/25.803/

[17] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, “RTP: a transport protocol for real-time applications,” Internet Engineering Task Force, RFC 3550, July 2003. [Online]. Available: http://www.rfc-editor.org/rfc/rfc3550.txt

[18] 3GPP TS 26.234, “Transparent end-to-end transparent Packet-switched Streaming Service (PSS); Protocols and codecs,” 3rd Generation Partnership Project (3GPP),” Tech. Spec., 2009. [Online]. Available: http://www.3gpp.org/ftp/Specs/archive/26_series/26.234/

[19] T. Paila, M. Luby, R. Lehtonen, V. Roca, and R. Walsh, “FLUTE - File Delivery over Unidirectional Transport,” Internet Engineering Task Force, RFC 3926, Oct. 2004. [Online]. Available: http://www.rfc-editor.org/rfc/rfc3926.txt

[20] ITU-T Recommendation H.264, "Advanced video coding for generic audiovisual services," Telecommunication Standardization Sector of ITU, Geneva, Switzerland, (ISO/IEC 14496-10), 2003.

[21] 3GPP TS 26.401, "General audio codec audio processing functions; Enhanced aacPlus general audio codec; General description," 3rd Generation Partnership Project (3GPP)," Tech. Spec., 2006. [Online]. Available: http://www.3gpp.org/ftp/Specs/archive/26_series/26.401/

[22] 3GPP TS 26.410, "General audio codec audio processing functions; Enhanced aacPlus general audio codec; Floating-point ANSI-C code," 3rd Generation Partnership Project (3GPP)," Tech. Spec., 2006. [Online]. Available: http://www.3gpp.org/ftp/Specs/archive/26_series/26.410/

[23] 3GPP TS 26.411, "General audio codec audio processing functions; Enhanced aacPlus general audio codec; Fixed-point ANSI-C code," 3rd Generation Partnership Project (3GPP)," Tech. Spec., 2006. [Online]. Available: http://www.3gpp.org/ftp/Specs/archive/26_series/26.411/

[24] 3GPP TS 26.290, "Audio codec processing functions; Extended Adaptive Multi-Rate - Wideband (AMR-WB+) codec; Transcoding functions," 3rd Generation Partnership Project (3GPP)," Tech. Spec., 2006. [Online]. Available: http://www.3gpp.org/ftp/Specs/archive/26_series/26.290/

[25] 3GPP TS 26.304, "Floating-point ANSI-C code for the Extended Adaptive Multi-Rate - Wideband (AMR-WB+) codec," 3rd Generation Partnership Project (3GPP)," Tech. Spec., 2006. [Online]. Available: http://www.3gpp.org/ftp/Specs/archive/26_series/26.304/

[26] 3GPP TS 26.273, "Speech codec speech processing functions; Extended Adaptive Multi-Rate - Wideband (AMR-WB+) speech codec; Fixed-point ANSI-C code," 3rd Generation Partnership Project (3GPP)," Tech. Spec., 2006. [Online]. Available: http://www.3gpp.org/ftp/Specs/archive/26_series/26.273/

[27] M. Watson, M. Luby, and L. Vicisano, "Forward error correction (FEC) building block," Internet Engineering Task Force, RFC 5052, Aug. 2007. [Online]. Available: http://www.rfc-editor.org/rfc/rfc5052.txt

[28] C. Bormann, C. Burmeister, M. Degermark, H. Fukushima, H. Hannu, L. e. Jonsson, R. Hakenberg, and T. Koren, "Robust header compression (ROHC): framework and four profiles: RTP, UDP, ESP, and uncompressed," Internet Engineering Task Force, RFC 3095, July 2001. [Online]. Available: http://www.rfc-editor.org/rfc/rfc3095.txt

[29] M. Ghaderi and S. S. Keshav, "Multimedia messaging service: system description and performance analysis," in *First International*

Conference on Wireless Internet, 2005, July 2005. [Online]. Available: blizzard.cs.uwaterloo.ca/keshav/home/Papers/data/05/mms.pdf

[30] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee, "Hypertext transfer protocol – HTTP/1.1," Internet Engineering Task Force, RFC 2616, June 1999. [Online]. Available: http://www.rfc-editor.org/rfc/rfc2616.txt

[31] J. Postel, "Transmission control protocol," Internet Engineering Task Force, RFC 793, Sept. 1981. [Online]. Available: http://www.rfc-editor.org/rfc/rfc793.txt

[32] R. Keller, T. Lohmar, R. Tönjes, and J. Thielecke, "Convergence of cellular and broadcast networks from a multi-radio perspective," *IEEE Personal Communications Magazine*, vol. 8, no. 2, pp. 51–56, April 2001.

[33] R. Walsh, L. Xu, and T. Paila, "Hybrid Networks - A Step Beyond 3G," in *3rd International Symposium on Wireless Personal Multimedia Communication*, November 2000.

[34] T. Lohmar, F. Hundscheidt, R. Keller, and R. Tönjes, "Multi-User Services in IMT-2000," in *10th International Telecommunication Network Strategy and Planning Symposium, Networks 2002*, June 2002.

[35] S. Acharya, M. Franklin, and S. Zdonik, "Balancing push and pull for data broadcast," in *ACM SIGMOD*, May 1997. [Online]. Available: citeseer.ist.psu.edu/19637.html

[36] T. Lohmar and U. Horn, "Hybrid Broadcast-Unicast distribution of Mobile TV over 3G Networks," in *IEEE LCN 2006*, November 2006.

[37] F. Hartung, J. H. Uwe Horn, M. Kampmann, T. Lohmar, and M. Lundevall, "Delivery of Mobile Broadcast Services in 3G Networks," *IEEE Transactions on Broadcasting*, vol. 53, no. 1-2, pp. 188–199, March 2007.

[38] H. Schulzrinne, A. Rao, and R. Lanphier, "Real time streaming protocol (RTSP)," Internet Engineering Task Force, RFC 2326, Apr. 1998. [Online]. Available: http://www.rfc-editor.org/rfc/rfc2326.txt

[39] –, "Integrated Mobile Broadcast (IMB) Service Scenarios and System Requirements," GSMA, London, UK, Tech. Rep., 2009. [Online]. Available: http://www.gsmworld.com/technology/4334.htm

[40] M. Sagfors, R. Ludwig, M. Meyer, and J. Peisa, "Buffer management for rate-varying 3G wireless links supporting TCP traffic," in *Vehicular Technology Conference (VTC) Spring*, April 2003.

[41] K. Savetz, N. Randall, and Y. Lepage, *MBONE: Multicasting Tomorrow's Internet*. John Wiley and Sons Inc (Computers), March 1996. [Online]. Available: http://www.savetz.com/mbone/

[42] Z. Albanna, K. Almeroth, D. Meyer, and M. Schipper, "IANA guidelines for ipv4 multicast address assignments," Internet Engineering Task Force, RFC 3171, Aug. 2001. [Online]. Available: http://www.rfc-editor.org/rfc/rfc3171.txt

[43] S. Deering, "Host extensions for IP multicasting," Internet Engineering Task Force, RFC 1112, Aug. 1989. [Online]. Available: http://www.rfc-editor.org/rfc/rfc1112.txt

[44] W. Fenner, "Internet group management protocol, version 2," Internet Engineering Task Force, RFC 2236, Nov. 1997. [Online]. Available: http://www.rfc-editor.org/rfc/rfc2236.txt

[45] B. Cain, S. Deering, I. Kouvelas, B. Fenner, and A. Thyagarajan, "Internet group management protocol, version 3," Internet Engineering Task Force, RFC 3376, Oct. 2002. [Online]. Available: http://www.rfc-editor.org/rfc/rfc3376.txt

[46] S. Deering, W. Fenner, and B. Haberman, "Multicast listener discovery (MLD) for ipv6," Internet Engineering Task Force, RFC 2710, Oct. 1999. [Online]. Available: http://www.rfc-editor.org/rfc/rfc2710.txt

[47] E. Vida and E. Costa, "Multicast Listener Discovery Version 2 (MLDv2) for IPv6," Internet Engineering Task Force, RFC 3810, June 2004. [Online]. Available: http://www.rfc-editor.org/rfc/rfc3810.txt

[48] D. Estrin, D. Farinacci, A. Helmy, D. Thaler, S. Deering, M. Handley, V. Jacobson, and C. Liu, "Protocol independent multicast-sparse mode (PIM-SM): protocol specification," Internet Engineering Task Force, RFC 2362, June 1998. [Online]. Available: http://www.rfc-editor.org/rfc/rfc2362.txt

[49] T. Ballardie, P. Francis, and J. Crowcroft, "Core based trees," in *ACM SIGCOMM*, Sept 1993.

[50] A. Ballardie, "Core based trees (CBT version 2) multicast routing – protocol specification –," Internet Engineering Task Force, RFC 2189, Sept. 1997. [Online]. Available: http://www.rfc-editor.org/rfc/rfc2189.txt

[51] ——, "Core Based Trees (CBT) Multicast Routing Architecture," Internet Engineering Task Force, RFC 2201, Sept. 1997. [Online]. Available: http://www.rfc-editor.org/rfc/rfc2201.txt

[52] J. Moy, "Multicast extensions to OSPF," Internet Engineering Task Force, RFC 1584, Mar. 1994. [Online]. Available: http://www.rfc-editor.org/rfc/rfc1584.txt

[53] V. Roca, L. Costa, R. Vida, A. Dracinschi, and S. Fdida, "A survey of multicast technologies," Sept. 2000. [Online]. Available: citeseer.ist.psu.edu/roca00survey.html

[54] A. Striegel and G. Manimaran, "A Survey of QoS Multicasting Issues," *IEEE Communications*, vol. 40, no. 6, pp. 82–87, June 2002. [Online]. Available: citeseer.ist.psu.edu/article/striegel02survey.html

[55] T. Lohmar, P. Casagranda, C. Giné-Sado, C. Janneteau, P. Seite, and Y. M. Tian, "Architectural issues for multicast group partitioning in mobile, hybrid networks," in *IST Mobile Summit 2003*, June 2003.

[56] S. Paul, K. K. Sabnani, J. C.-H. Lin, and S. Bhattacharyya, "Reliable Multicast Transport Protocol (RMTP)," *IEEE Journal of Selected Areas in Communications*, vol. 15, no. 3, pp. 407–421, 1997. [Online]. Available: citeseer.ist.psu.edu/paul97reliable.html

[57] K. Rothermel and C. Maihöfer, "A Robust and Efficient Mechanism for Constructing Multicast Acknowledgement Trees," in *IEEE International Conference on Computer Communications and Networks, ICCCN '99)*, 1999. [Online]. Available: citeseer.ist.psu.edu/rothermel99robust.html

[58] S. Floyd, V. Jacobson, C.-G. Liu, S. McCanne, and L. Zhang, "A reliable multicast framework for light-weight sessions and application level framing," *IEEE/ACM Transactions on Networking*, vol. 5, no. 6, pp. 784–803, 1997. [Online]. Available: citeseer.ist.psu.edu/floyd95reliable.html

[59] W. T. Strayer, B. J. Dempsey, and A. C. Weaver, *Xtp: The Xpress Transfer Protocol*. Addison-Wesley Pub (Sd), August 1992.

[60] K. Robertson, K. Miller, M. White, and A. Tweedly, "Starburst multicast file transfer protocol (MFTP) specification," draft-miller-mftp-spec-03, Internet Engineering Task Force, Internet Draft, Apr. 1998, work in progress. [Online]. Available: http://www.ietf.org/internet-drafts/draft-miller-mftp-spec-03.txt

[61] C. Huitema, "The case for packet level FEC," in *Protocols for High-Speed Networks*, 1996, pp. 109–120. [Online]. Available: citeseer.ist.psu.edu/huitema96case.html

[62] J. Nonnenmacher, "Reliable multicast transport to large groups," 1998. [Online]. Available: citeseer.ist.psu.edu/nonnenmacher98reliable.html

[63] J. Nonnenmacher and E. Biersack, "Reliable multicast: where to use FEC," in *Protocols for High-Speed Networks*, 1996, pp. 134–148. [Online]. Available: citeseer.ist.psu.edu/nonnenmacher96reliable.html

[64] L. Rizzo, "Effective erasure codes for reliable computer communication protocols," *ACM Computer Communication Review*, vol. 27, no. 2, pp. 24–36, Apr. 1997. [Online]. Available: citeseer.ist.psu.edu/rizzo97effective.html

[65] L. Rizzo and L. Vicisano, "RMDP: An FEC-based Reliable Multicast Protocol for Wireless Environments," *Mobile Computing and Communications Review*, vol. 2, no. 2, April 1998. [Online]. Available: citeseer.ist.psu.edu/rizzo98rmdp.html

[66] J. W. Byers, M. Luby, M. Mitzenmacher, and A. Rege, "A Digital Fountain Approach to Reliable Distribution of Bulk Data," in *SIGCOMM*, 1998. [Online]. Available: citeseer.ist.psu.edu/byers98digital.html

[67] E. Schooler and J. Gemmell, "Using multicast fec to solve the midnight madness problem," 1997. [Online]. Available: http://citeseer.ist.psu.edu/schooler97using.html

[68] M. Handley, S. Floyd, B. Whetten, R. Kermode, L. Vicisano, and M. Luby, "The reliable multicast design space for bulk data transfer," Internet Engineering Task Force, RFC 2887, Aug. 2000. [Online]. Available: http://www.rfc-editor.org/rfc/rfc2887.txt

[69] B. Whetten, L. Vicisano, R. Kermode, M. Handley, S. Floyd, and M. Luby, "Reliable multicast transport building blocks for one-to-many bulk-data transfer," Internet Engineering Task Force, RFC 3048, Jan. 2001. [Online]. Available: http://www.rfc-editor.org/rfc/rfc3048.txt

[70] M. Luby, J. Gemmell, L. Vicisano, L. Rizzo, and J. Crowcroft, "Asynchronous layered coding (ALC) protocol instantiation," Internet Engineering Task Force, RFC 3450, Dec. 2002. [Online]. Available: http://www.rfc-editor.org/rfc/rfc3450.txt

[71] M. Luby, J. Gemmell, L. Vicisano, L. Rizzo, M. Handley, and J. Crowcroft, "Layered coding transport (LCT) building block," Internet Engineering Task Force, RFC 3451, Dec. 2002. [Online]. Available: http://www.rfc-editor.org/rfc/rfc3451.txt

[72] M. Luby, L. Vicisano, J. Gemmell, L. Rizzo, M. Handley, and J. Crowcroft, "Forward error correction (FEC) building block," Internet Engineering Task Force, RFC 3452, Dec. 2002. [Online]. Available: http://www.rfc-editor.org/rfc/rfc3452.txt

[73] S. McCanne, V. Jacobson, and M. Vetterli, "Receiver-driven layered multicast," in *ACM SIGCOMM*, August 1996. [Online]. Available: citeseer.ist.psu.edu/steven96receiverdriven.html

[74] M. Luby and L. Vicisano, "Compact Forward Error Correction (FEC) Schemes," Internet Engineering Task Force, RFC 3695, Feb. 2004. [Online]. Available: http://www.rfc-editor.org/rfc/rfc3695.txt

[75] ETSI TS 102 472 V1.1.1 (2006-06), "IP Datacast over DVB-H: Content Delivery Protocols (CDP)," European Telecommunications Standards Institute (ETSI)," Tech. Spec., 2006. [Online]. Available: http://www.dvb-h.org/PDF/ts_102472v010101p.pdf

[76] OMA BCAST, "File and Stream Distribution for Mobile Broadcast Services: OMA-TS-BCAST_Distribution-V1_0-20070712-C," Open Mobile Alliance," Tech. Spec., 2007.

[77] N. Borenstein and N. Freed, "MIME (multipurpose internet mail extensions) part one: Mechanisms for specifying and describing the format of internet message bodies," Internet Engineering Task Force, RFC 1521, Sept. 1993. [Online]. Available: http://www.rfc-editor.org/rfc/rfc1521.txt

[78] K. Moore, "MIME (multipurpose internet mail extensions) part two: Message header extensions for non-ascii text," Internet Engineering Task Force, RFC 1522, Sept. 1993. [Online]. Available: http://www.rfc-editor.org/rfc/rfc1522.txt

[79] M. Watson, "Basic Forward Error Correction (FEC) Schemes," Internet Engineering Task Force, Tech. Rep. 5445, Mar. 2009. [Online]. Available: http://www.ietf.org/rfc/rfc5445.txt

[80] J. Lacan, V. Roca, J. Peltotalo, and S. Peltotalo, "Reed-Solomon Forward Error Correction (FEC) Schemes," Internet Engineering Task Force, Tech. Rep. 5510, Apr. 2009. [Online]. Available: http://www.ietf.org/rfc/rfc5510.txt

[81] V. Roca, C. Neumann, and D. Furodet, "Low Density Parity Check (LDPC) Staircase and Triangle Forward Error Correction (FEC) Schemes," Internet Engineering Task Force, Tech. Rep. 5170, June 2008. [Online]. Available: http://www.ietf.org/rfc/rfc5170.txt

[82] M. Luby, A. Shokrollahi, M. Watson, and T. Stockhammer, "Raptor Forward Error Correction Scheme for Object Delivery," Internet Engineering Task Force, Tech. Rep. 5053, Oct. 2007. [Online]. Available: http://www.ietf.org/rfc/rfc5053.txt

[83] ETSI EN 302 304 V1.1.1 (2004-11), "DVB-H - Transmission System for Handheld Terminals," European Telecommunications Standards Institute (ETSI)," EN, 2004. [Online]. Available: http://www.dvb-h.org/PDF/DVB-H%20Specification%20-%20En302304.V1.1.1.pdf

[84] ETSI TS 102 034 V1.4.1 (2009-08), "Transport of MPEG-2 TS Based DVB Services over IP Based Networks (and associated XML)," European Telecommunications Standards Institute (ETSI)," Tech. Spec., 2009. [Online]. Available: http://www.etsi.org/deliver/etsi_ts/102000_102099/102034/

[85] R. G. Gallager, *Low-Density Parity-Check Codes.* MIT Press Classic Series, 1963.

[86] A. Shokrollahi, "LDPC Codes: An Introduction," Digital Fountain, Tech. Rep., 2003. [Online]. Available: http://www.ics.uci.edu/~welling/teaching/ICS279/LPCD.pdf

[87] M. Luby, "LT Codes," in *IEEE Symposium on the Foundations of Computer Science (FOCS)*, November 2002.

[88] A. Shokrollahi, "Raptor Codes," Digital Fountain, Digital Fountain Technical Report DF2003-06-001 DF2003-06-001, June 2003.

[89] ——, "Raptor Codes," *IEEE Transactions on Information Theory*, vol. 6, no. 52, pp. 2551–2567, 2006.

[90] D. J. MacKay, *Information theory, inference, and learning algorithms.* Cambridge Univ Press, 2003.

[91] M. Luby, M. Watson, T. Gasiba, T. Stockhammer, and W. Xu, "Raptor Codes for Reliable Download Delivery in Wireless Broadcast Systems," in *CCNC 2006*, January 2006.

[92] S4-040230, "Raptor code specification for MBMS file download, Digital Fountain," 3GPP TSG SA WG4, SA4#31, Montreal, Canada, TDoc, 2004.

[93] OMA BCAST, "Mobile Broadcast Services: OMA-TS-BCAST_Services-V1_0-20070710-C," Open Mobile Alliance," Tech. Spec., 2007.

[94] P. Elias, "Coding for two noisy channels," in *Information Theory, 3rd London Symp.*, 1955, pp. 61–76.

[95] S4-040120, "Mapping of SDUs to Radio Blocks for FEC simulations, Nortel Networks," 3GPP TSG SA WG4, PSM SWG Ad-hoc, Lund, Sweden, TDoc, 2004.

[96] T. Lohmar and J. Huschke, "Radio Resource Optimization for MBMS File Transmissions," in *IEEE BMSB 2009*, May 2009.

[97] T. Lohmar, Z. Peng, and P. Mähönen, "Performance Evaluation of a File Repair Procedure Based on a Combination of MBMS and Unicast Bearers," in *IEEE WOWMOM 2006*, June 26–29 2006.

[98] 3GPP TS 3GPP 30.03U, "Selection procedures for the choice of radio transmission technologies of the UMTS," 3rd Generation Partnership Project (3GPP)," Tech. Spec., 1998. [Online]. Available: http://www.3gpp.org/ftp/Specs/archive/30_series/30.03U/

[99] P. Walters, *An introduction to Ergodic Theory.* Springer, New York, 1982.

[100] T. Lohmar and M. Elisova, "Evaluation of the File Repair Operations for Multicast/Broadcast Download Deliveries," in *European Wireless 2005*, April 2005.

[101] D. Gomez-Barquero, A. Aguilella, and N. Cardona, "Multicast Delivery of File Download Services in 3G Mobile Networks with MBMS," in *IEEE International Symposium on Broadband Multimedia Systems and Broadcasting, 2008*, March 2008.

[102] G. F. Riley, "The Georgia Tech Network Simulator," Software online: http://www.ece.gatech.edu/ research/ labs/ MANIACS/ gtnets.htm, 2003.

Acknowledgments

The research presented in this thesis has been carried out during the years 2003 to 2009, while working as Research Engineer at Ericsson Eurolab, Ericsson GmbH in Herzogenrath. I would like to thanks in particular Dr. Norbert Niebert and Dr. Michael Meyer for making this possible.

I would like to thank Prof. Petri Mähönen for his advice and support.

During the last years, I had the pleasure of working with great colleagues at Ericsson Research. We had lots of inspiring discussions. This work greatly benefited from discussions with Jörg Huschke in the area of the 3G radio network aspects. I would also like to thank my colleagues Daniel, Frank, Jörg, Joachim, Johannes, Mai-Anh, Markus and Tobbe for their support with reviewing the work.

Foremost, I want to thank my wife Gabi, for her patience, faith, and support over the past eight years.

I also would like to thank all the master students, who have contributed to the success of this work. In particular I would like to thank Zhaoyi Peng and Magda Elisova.

List of Publications

Conference contributions relevant to the thesis at hand

C1 **T. Lohmar**, Z. Peng, and P. Mähönen, “Performance Evaluation of a File Repair Procedure Based on a Combination of MBMS and Unicast Bearers”, in *Proc. of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WOWMOM) 2006*, Niagara Falls, NY, USA, June 2006.

C2 **T. Lohmar**, J. Huschke, “Radio Resource Optimization for MBMS File Transmissions”, in *Proc.of IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB) 2009*, Bilbao, Spain, May 2009.

C3 **T. Lohmar** and M. Elisova, “Evaluation of the File Repair Operations for Multicast/Broadcast Download Deliveries”, in *Proc. of European Wireless 2005*, Nicosia, Cyprus, April 2005.

C4 **T. Lohmar** and U. Horn, “Hybrid Broadcast-Unicast distribution of Mobile TV over 3G Networks”, in *Proc. of The 31st Annual IEEE Conference on Local Computer Networks (LCN 2006)*, Tampa, FL, USA, November 2006.

Journal papers relevant to the thesis at hand

C5 F. Hartung, U.Horn, J. Huschke, M. Kampmann, **T. Lohmar**, M. Lundevall, “Delivery of Mobile Broadcast Services in 3G Networks”, *IEEE Transactions on Broadcasting*, vol 53, p. 188-199, March 2007.

C6 **T. Lohmar**, J.-A. Ibanez, A. Zanin, M. Blockstrand, “Scalable push file delivery with MBMS”, *Ericsson Review*, Issue no 01, 2009.

C7 F. Hartung, U.Horn, J. Huschke, M. Kampmann, **T. Lohmar**, “MBMS – IP Multicast/Broadcast in 3G Networks”, International Journal of Digital Multimedia Broadcasting Volume 2009, *Hindawi Publishing Corporation*, doi:10.1155/2009/597848, `http://dx.doi.org/10.1155/2009/597848`.

Conference contributions and Journal papers in addition to the research presented in this thesis

A1 F. Gabin, M. Kampmann, **T. Lohmar**, C. Priddle, "3GPP Mobile Multimedia Streaming Standards", in IEEE Signal Processing Magazine, vol. 24, issue 6, pp. 134-138, November 2010.

A2 J. Dölfel, **T. Lohmar**, "A Generic Framework for File Delivery Services via MBMS", Technologien und Anwendungen - Vorträge der 13. ITG-Fachtagung, Aprl. 2008, Osanbrück.

A3 J.-A. Ibanez, **T. Lohmar**, D. Turina, A. Zanin, "Mobile TV over 3G networks - Enablers and service evolution", *Ericsson Review*, Issue no 01, 2008.

A4 F. Hartung, U. Horn, J. Huschke, M. Kampmann, **T. Lohmar**, "Mobile TV Services in 3G Networks", in Proceedings Kommunikation in Verteilten Systemen (KiVS) 2007, Berne, Switzerland, February 2007.

A5 F. Hartung, U. Horn, J. Huschke, M. Kampmann, **T. Lohmar**, "Delivery of Mobile TV services in 3G cellular networks", in Proceedings 12. Dortmunder Fernsehseminar, Dortmund, Germany, March 2007.

A6 R. Tönjes, C. Simon, **T. Lohmar**, C. Janneteau, P. Casagranda: "Broadcast/Multicast Services in Multi-Access Architectures", 12th WWRF - Wireless World Research Forum, Toronto, 4-5 November, 2004.

A7 R. Tönjes, K. Mößner, **T. Lohmar**, M. Wolf: "OverDRiVE Conclusions - Resource Sharing in Hybrid Radio Networks", IST Mobile Summit, Lyon, 27-30 June 2004.

A8 **T. Lohmar**, H. Wiemann, F. Hundscheidt, M. Meyer, R Keller, "Support of Multicast Services in 3GPP", Praxis der Informationsverarbeitung und Kommunikation (PIK), Jul.-Aug. 2004.

A9 **T. Lohmar**, R. Keller, R. Tönjes, F. Hundscheidt: "Large User Group Support in Mobile Networks", 9th WWRF - Wireless World Research Forum, Zurich, July, 2003.

A10 **T. Lohmar**, P. Casagranda, C. Gine-Sado, C. Janneteau, P. Seite, and Y. M. Tian, "Architectural issues for multicast group partitioning in mobile, hybrid networks", in IST Mobile Summit 2003, Aveiro, Portugal, June 2003.

A11 C. Janneteau, Y. Tian, S. Csaba, **T. Lohmar**, Y.-H. Lach, R. Tafazolli, "Comparison of Three Approaches Towards Mobile Multicast", IST Mobile Summit 2003, Aveiro, Portugal, June 2003.

A12 R. Tönjes, K. Mößner, **T. Lohmar**, M. Wolf, "OverDRiVE - Spectrum Efficient Multicast Services to Vehicles", IST Mobile Summit 2002, Thessaloniki, June, 2002.

A13 **T. Lohmar**, F. Hundscheidt, R. Keller, and R. Tönjes, "Multi-User Services in IMT-2000", in 10th International Telecommunication Network Strategy and Planning Symposium, Networks 2002, Munich, June 2002.

A14 R. Tönjes, **T. Lohmar**, M. Vorwerk, R. Keller: "Flow Control for Multi-Access Systems", IEEE International Symposium on Personal, Indoor and Mobile Radio Communication (PIMRC 2002), Lisbon, 15-18. Sep.2002.

A15 **T. Lohmar**, F. Hundscheidt, R. Keller, R. Tönjes, "Interactive Broadcast Services within IMT-2000 and beyond Systems", 7th WWRF, Wireless World Research Forum. 3rd-4th December, 2002.

A16 M. Frank, T. Göransson, W. Hansmann, O. Johansson, **T. Lohmar**, T. Paila, R. Tönjes, L. Xu: "Mobility management in a hybrid radio system", RSRCP Journal - Réseaux et systèmes répartis, calculateurs parallèles, Special Issue on Mobility and Internet, Hermes Science Publications, Volume 13, No. XXX, 2001.

A17 T. Paila, S. Alladin, M. Frank, T. Göransson, W. Hansmann, **T. Lohmar**, R. Tönjes, L. Xu: "Flexible Network Architecture for Future Hybrid Wireless Systems", Mobile Summit 2001; Barcelona; 10.-12. September 2001.

A18 R. Keller, **T. Lohmar**, R. Tönjes, and J. Thielecke, "Convergence of cellular and broadcast networks from a multi-radio perspective", IEEE Personal Communications Magazine, vol. 8, no. 2, pp. 51-56, April 2001.

A19 R. Keller, **T. Lohmar**, R. Tönjes, J. Thielecke: "Convergence of Broadcast and New Telecom Networks", Wireless Personal Communications Magazine, 17, pp. 269-282, Kluwer Academic Publishers, Netherlands, 2001.

A20 R. Keller, **T. Lohmar**, R. Tönjes, J. Thielecke: "Convergence of Cellular and Broadcast Networks from a Multi-Radio Perspective", IEEE Personal Communications, 8(2): 51-56, 2001.

A21 T. Paila, S. Alladin, M. Frank, T. Göransson, W. Hansmann, **T. Lohmar**, R. Tönjes, L. Xu: "Flexible Network Architecture for Future Hybrid Wireless Systems", Mobile Summit 2001; Barcelona; 10.-12. September 2001.

A22 J. Diederich, **T. Lohmar**, M. Zitterbart and R. Keller, "A QoS Model for Differentiated Services in Mobile Wireless Networks", in Paper Digest of the 11th IEEE Workshop on Local and Metropolitan Area Networks (IEEE LANMAN), Boulder, Colorado, USA, March 2001

A23 E. Kovacs, R. Keller, **T. Lohmar**, R. Kroh, A. Held, "Adaptive Mobile Applications over Cellular Advanced Radio", IEEE 11th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2000), London, UK, September 2000.

A24 J. Diederich, **T. Lohmar**, M. Zitterbart, R. Keller, "On Integrating Differentiated Services and UMTS Networks", in Proceedings of 25th Conference on Local Computer Networks (LCN 2000), Tampa, Florida, USA, Seite 196-197, November 2000 (DLZK-lcn00)

A25 A. Krämling, M. Scheibenbogen, **T. Lohmar**, "Dynamic Channel Allocation in Wireless ATM Networks", In International Conference on Telecommunications (ICT'98), Porto Carras, Greece, June 1998.