ETSI TS 102 527-4 V1.2.1 (2014-12)

Digital Enhanced Cordless Telecommunications (DECT);
New Generation DECT;
Part 4: Light Data Services;
Software Update Over The Air (SUOTA),
content downloading and HTTP based applications

Reference
RTS/DECT-NG268-4

Keywords
access, data, DECT, DPRS, IMT-2000, internet, interoperability, interworking, mobility, packet mode, profile, radio, synchronization, TDD, TDMA

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Contents

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://ipr.etsi.org).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This Technical Specification (TS) has been produced by ETSI Technical Committee Digital Enhanced Cordless Telecommunications (DECT).

The present document is based on ETSI EN 300 175, parts 1 [1] to 8 [8], ETSI EN 300 444 [14] and ETSI EN 301 649 [15]. General attachment requirements and speech attachment requirements are based on ETSI EN 301 406 [11] (replacing ETSI TBR 006 [i.2]) and ETSI EN 300 176-2 [10] (previously covered by ETSI TBR 010 [i.3]). Further details of the DECT system may be found in ETSI TR 101 178 [i.1].

The present document has been developed in accordance to the rules of documenting a profile specification as described in ISO/IEC 9646-6 [i.14].

The information in the present document is believed to be correct at the time of publication. However, DECT standardization is a rapidly changing area, and it is possible that some of the information contained in the present document may become outdated or incomplete within relatively short time-scales.

The present document is part 4 of a multi-part deliverable covering the New Generation DECT as identified below:

Part 1: "Wideband speech";

Part 2: "Support of transparent IP packet data";

Part 3: "Extended wideband speech services";

Part 4: "Light Data Services; Software Update Over The Air (SUOTA), content downloading and HTTP based applications";

Part 5: "Additional feature set nr. 1 for extended wideband speech services".

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**may not**", "**need**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

1 Scope

The present document specifies a set of functionalities of the New Generation DECT.

The functionalities defined in this profile are based on DECT base standard, ETSI EN 300 175, parts 1 [1] to 8 [8], DECT Generic Access Profile (GAP), ETSI EN 300 444 [14], and DECT Packet Radio Service (DPRS), ETSI EN 301 649 [15].

The New Generation DECT provides the following basic new functionalities:

- wideband voice service;
- packet-mode data service supporting Internet Protocol with efficient spectrum usage and high data rates.

All DECT devices claiming to be compliant with this Application Profile will offer at least the basic services defined as mandatory. In addition to that, optional features can be implemented to offer additional DECT services.

The aim of the present document is to guarantee a sufficient level of interoperability and to provide an easy route for development of DECT data applications, with the features of the present document being a common fall-back option available in all compliant to this profile equipment.

2 References

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the reference document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at http://docbox.etsi.org/Reference.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

2.1 Normative references

The following referenced documents are necessary for the application of the present document.

[1] ETSI EN 300 175-1: "Digital Enhanced Cordless Telecommunications (DECT); Common Interface (CI); Part 1: Overview".

[2] ETSI EN 300 175-2: "Digital Enhanced Cordless Telecommunications (DECT); Common Interface (CI); Part 2: Physical layer (PHL)".

[3] ETSI EN 300 175-3: "Digital Enhanced Cordless Telecommunications (DECT); Common Interface (CI); Part 3: Medium Access Control (MAC) layer".

[4] ETSI EN 300 175-4: "Digital Enhanced Cordless Telecommunications (DECT); Common Interface (CI); Part 4: Data Link Control (DLC) layer".

[5] ETSI EN 300 175-5: "Digital Enhanced Cordless Telecommunications (DECT); Common Interface (CI); Part 5: Network (NWK) layer".

[6] ETSI EN 300 175-6: "Digital Enhanced Cordless Telecommunications (DECT); Common Interface (CI); Part 6: Identities and addressing".

[7] ETSI EN 300 175-7: "Digital Enhanced Cordless Telecommunications (DECT); Common Interface (CI); Part 7: Security features".

[8] ETSI EN 300 175-8: "Digital Enhanced Cordless Telecommunications (DECT); Common Interface (CI); Part 8: Speech and audio coding and transmission".

[9] ETSI EN 300 176-1: "Digital Enhanced Cordless Telecommunications (DECT); Test specification; Part 1: Radio".

[10] ETSI EN 300 176-2: "Digital Enhanced Cordless Telecommunications (DECT); Test specification; Part 2: Audio and speech".

[11] ETSI EN 301 406: "Digital Enhanced Cordless Telecommunications (DECT); Harmonized EN for Digital Enhanced Cordless Telecommunications (DECT) covering the essential requirements under article 3.2 of the R&TTE Directive; Generic radio".

[12] Void.

[13] Void.

[14] ETSI EN 300 444: "Digital Enhanced Cordless Telecommunications (DECT); Generic Access Profile (GAP)".

[15] ETSI EN 301 649: "Digital Enhanced Cordless Telecommunications (DECT); DECT Packet Radio Service (DPRS)".

[16] ETSI TS 102 527-1: "Digital Enhanced Cordless Telecommunications (DECT); New Generation DECT; Part 1: Wideband Speech ".

[17] ETSI TS 102 527-3: "Digital Enhanced Cordless Telecommunications (DECT); New Generation DECT; Part 3: Extended Wideband Speech Services".

[18] Void.

[19] IETF RFC 2616: "Hypertext Transfer Protocol -- HTTP/1.1".

[20] IETF RFC 1034: "Domain names - concepts and facilities" (STD 13).

[21] IETF RFC 1035: "Domain names - implementation and specification" (STD 13).

[22] W3C Recommendation 23 November 2010: "XHTML™ 1.1 - Module-based XHTML - Second Edition".

NOTE: http://www.w3.org/TR/2010/REC-xhtml11-20101123/.

[23] IETF RFC 3629 (2003): "UTF-8, a transformation format of ISO 10646".

2.2 Informative references

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] ETSI TR 101 178: "Digital Enhanced Cordless Telecommunications (DECT); A high Level Guide to the DECT Standardization".

[i.2] ETSI TBR 006: "Digital Enhanced Cordless Telecommunications (DECT); General terminal attachment requirements".

[i.3] ETSI TBR 010: "Digital Enhanced Cordless Telecommunications (DECT); General Terminal Attachment Requirements; Telephony Applications".

[i.4] Void.

[i.5] The Broadband Forum's (formerly DSL-Forum) Technical Report 069 (TR-069): "Technical Reports for a Customer Premises Equipment (CPE) WAN Management Protocol".

[i.6] Web pages of the Unicode Consortium.

NOTE: http://www.unicode.org/.

[i.7] IEEE 802.3™: "IEEE Standard for Information technology - Specific requirements - Part 3: Carrier Sense Multiple Access with Collision Detection (CMSA/CD) Access Method and Physical Layer Specifications" (also known as ISO/IEC 8802-3).

[i.8] IEEE 802.5™: "IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements - Part 5: Token Ring Access Method and Physical Layer Specification" (also known as ISO/IEC 8802-5).

[i.9] IETF RFC 1661: "The Point-to-Point Protocol (PPP)".

[i.10] ISO/IEC 8859-1: "Information technology -- 8-bit single-byte coded graphic character sets -- Part 1: Latin alphabet No. 1".

[i.11] ISO/IEC 8859-2: "Information technology -- 8-bit single-byte coded graphic character sets -- Part 2: Latin alphabet No. 2".

[i.12] ISO/IEC 8859-15: "Information technology -- 8-bit single-byte coded graphic character sets -- Part 15: Latin alphabet No. 9".

[i.13] ISO/IEC 9646-7: "Information technology - Open Systems Interconnection - Conformance testing methodology and framework - Part 7: Implementation Conformance Statements".

[i.14] ISO/IEC 9646-6: "Information technology - Open Systems Interconnection - Conformance testing methodology and framework - Part 6: Protocol profile test specification".

[i.15] ETSI TS 102 527-5: "Digital Enhanced Cordless Telecommunications (DECT); New Generation DECT; Part 5: Additional feature set nr. 1 for extended wideband speech services".

[i.16] IETF RFC 791 (1981): "Internet Protocol" (STD 5).

[i.17] IETF RFC 768 (1980): "User Datagram Protocol" (STD 6).

[i.18] IETF RFC 793 (1981): "Transmission Control Protocol" (STD 7).

[i.19] IETF RFC 3986: "Uniform Resource Identifier (URI): Generic Syntax" (STD 66).

NOTE: IETF RFC 3986 obsoletes IETF RFC 2396.

[i.20] IETF RFC 2817: "Upgrading to TLS within HTTP/1.1".

3 Definitions, symbols and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in ETSI EN 301 649 [15] and the following apply:

distributed application: application available to the user on a DECT handset, for which part of the code (behaviour) and/or data is located on the handset (local tier) and part of it is located in the network (remote tier), and more specifically on one or more HTTP servers hosted by-or on behalf of-the FP's vendor

Light Data Services (LDS): basic DECT data services with limited data rate and simplified implementation

software package: set of files sharing the same version identifier, and needed by the PP for installing or upgrading an application or a firmware

NOTE: The software package is often simply referred to as the "software".

software upgrade Downloading Server (DS): site of a PP vendor, or operated on behalf of a PP vendor, from where the software image releases can be downloaded

software upgrade Management Server (MS): site of a PP vendor, or operated on behalf of a PP vendor, where information about new software image releases for handsets, and their locations (on the downloading server) can be found

Software Upgrade Over The Air (SUOTA): capability to upgrade the Software or the Firmware in the PP by means of downloading it from the FP via the DECT air interface

software version identifier: parameter that identifies a software package, including the software package version

NOTE: From PP to FP, this parameter identifies the currently installed software package. From FP to PP it identifies the software package to be installed as a result of the upgrade (and is shared by all the files needed for the upgrade). Details and examples are provided in clause 7.5.5.2.1.

3.2 Symbols

For the purposes of the present document, the symbols given in ETSI EN 301 649 [15] and the following apply:

C	For conditional to support (process mandatory)
I	For irrelevant or out-of-scope (provision optional, process optional), not subject for testing
M	For mandatory to support (provision mandatory, process mandatory)
N/A	For not-applicable (in the given context the specification makes it impossible to use this capability)
O	For optional to support (provision optional, process mandatory)
O.x	Option comprising number of items
X	Excluded, not allowed

The symbols defined in this clause are applied for procedures, features, and services in the present document if not explicitly otherwise stated. The interpretation of status columns in all tables is as follows:

- Provision mandatory, process mandatory means that the indicated feature service or procedure are implemented as described in the present document, and may be subject to testing.
- Provision optional, process mandatory means that the indicated feature, service or procedure may be implemented, and if implemented, the feature, service or procedure are implemented as described in the present document, and may be subject to testing.

NOTE: The used notation is based on the notation proposed in ISO/IEC 9646-7 [i.13].

3.3 Abbreviations

For the purposes of the present document, the abbreviations given in ETSI EN 301 649 [15] and the following apply:

ACS	Auto-Configuration Server
ASCII	American Standard Code for Information Interchange
BER	Bit Error Rate
CISS	Call Independent Supplementary Services
CLIP	Calling Line Identification Presentation
CLSS	ConnectionLess Supplementary Service
CNIP	Calling Name Identification Presentation
D-GMEP	DPRS Generic Media Encapsulation Protocol
DBPSK	Differential Binary Phase Shift Keying
DPRS	DECT Packet Radio Service
DNS	Domain Name Server
DQPSK	Differential Quadrature Phase Shift Keying
DS	Download(ing) Server
EMC	Equipment Manufacturer Code
GAP	Generic Access Profile
GF	higher layer information control channel - a logical channel to the MAC layer
GFSK	Gaussian Frequency-Shift Keying
GMCI	Generic Media Context Identifier
GMEP	See D-GMEP

HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
HTTPS	HTTP Secure
HW	HardWare
IE	Information Element
I_{PF}	higher layer Information channel protected, transported multiplexed with signalling in the E+U type slots
I_{PM}	higher layer Information channel (protected) with multi subfield format
I_{PMR}	higher layer Information channel (protected) with multi subfield format and error correction using MOD-2 retransmission mechanism
I_{PQ}	higher layer Information channel (protected) with single B-field
I_{PQR}	higher layer Information channel (protected) with single subfield format, with error correction using MOD-2 retransmission mechanism
LAPC	Link Access Protocol for the Control plane (a DLC layer C-plane protocol entity)
LDS	LightData Services
L_H	Length of HW Version identifier
L_S	Length of SW Version identifier
L_U	Length of URL content in this IE
MOD-2	Modulo 2
MS	Management Server
MT	MAC control channel on A-tail field, or one message on such channel
n	Current requested file number (value of the "fileNumber" parameter in the current "Handset Version indication" command)
N_f	Number of files to be downloaded for a given Software upgrade ($1 \leq N_f \leq 15$)
PD	Protocol Discriminator
QAM	Quadrature Amplitude Modulation
SIPF	higher layer connectionless channel (protected) transported multiplexed with signalling in the E+U type slots
SOAP	Simple Object Access Protocol
SSL	Secure Sockets Layer
SUOTA	Software Upgrade Over The Air
SW	SoftWare
TBR	Technical Basis for Regulation
TCP/IP	Transmission Control Protocol/Internet Protocol
TI	Transaction Identifier
TLS	Transport Layer Security
UIS	User Initiated SUOTA
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
UTF-8	UCS Transformation Format 8 bits
WAN	Wide Area Network
XHTML	eXtensible HyperText Markup Language
XML	eXtensible Markup Language

4 Description of services

4.1 Services covered by the present document

4.1.1 Introduction

The present document defines a set of what has been named "Light Data Services". Light Data Services are packet mode data services for specific applications, based on DPRS [15], and designed to be implementable using simplified operation modes.

The following Light Data Services are defined by the present document:

- Software Upgrade Over The Air (SUOTA); SUOTA may be automatic, user initiated, or possibly pushed from the management server.
- Binary content download; this is used in SUOTA for the actual software download, and can also be used to define proprietary distributed applications to be used on handsets.
- HTTP based applications; this is intended to allow the design of DECT specific applications based on a limited browsing functionality.

Further Light Data Services may be created by further releases of the present document, or by other DECT specifications.

4.1.2 Application scenario

The Light Data Services defined by the present document have been designed as a complement to voice service terminals. Therefore, the expected scenario is that PPs and FPs implementing the present specification, are also implementing one of the DECT voice services. The services defined by the present document have been optimized to be a natural complement of New Generation DECT; part 1: wideband speech [16], and New Generation DECT: part 3: extended wideband speech services [17]. However, it is also possible the use the Light Data Services in combination with plain GAP [14] terminals.

The application scenario assumes that there is a data connection at the DECT FP that allows the connectivity to external application servers that participate in the service from application point of view. Such data connectivity is typically via the Internet. However, other scenarios of connectivity may exist, including the case when the FP incorporates locally the network side application server. The network side implementation of the scenario is out of the scope of DECT standardization. However, the descriptions given in the present document will assume the most expected case of data connectivity via the Internet, and remote application servers located at any internet location. Other cases, as exotic connectivity or local implementation of the network side server are adaptation of the general scenario, without impact on the DECT air interface protocols.

4.2 Light Data Services Protocol architecture

The common characteristic of all Light Data Services defined by the present document is the use of the DPRS [15], generic media encapsulation interworking mode (DPRS [15], clause B.8). The generic media encapsulation is a DPRS facility that allows the direct transportation of multiple application protocols. In the case of the Light Data Services defined in the present document, the application protocol is HTTP (as defined by RFC 2616 [19]) supporting only the modes defined by DPRS [15], clause B.8.

Another characteristic of the protocol architecture defined in the present document is that the application protocol is transported without the use of any transport layer protocol (TCP [i.18] or UDP [i.17]). This approach, that has been chosen in order to simplify implementations, relies on the request/response nature of the application protocol and requires some collaboration from it in order to perform the tasks normally done by the transport layer (TCP).

The result of the approach is that PPs do not need to implement the TCP protocol.

4.2.1 Data protocol reference configuration

Figures 1 and 2 define the U-plane and C-plane protocol stacks used in the Light Data Services defined by the present document.

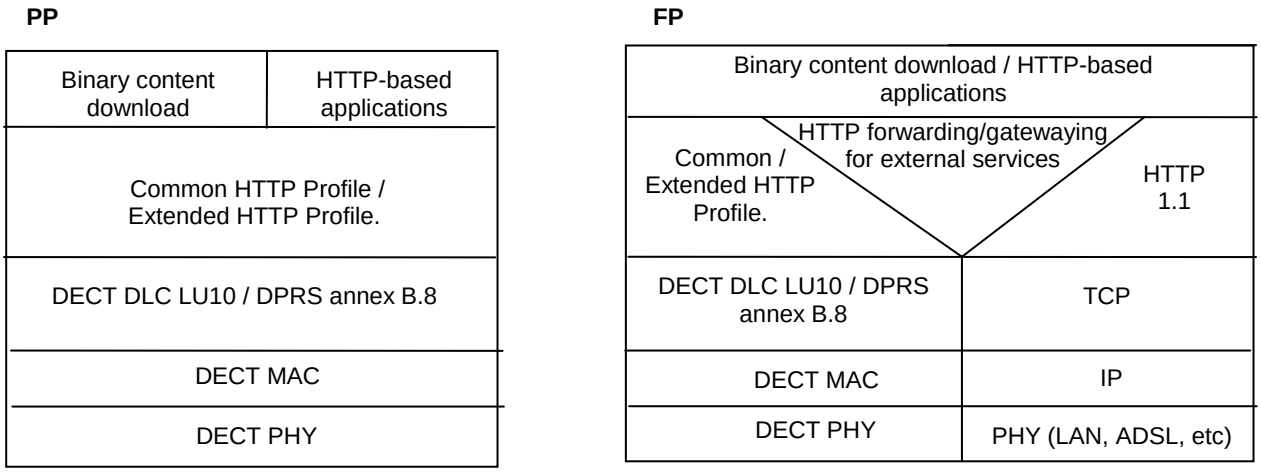


Figure 1: Reference model of the U-plane protocol stack

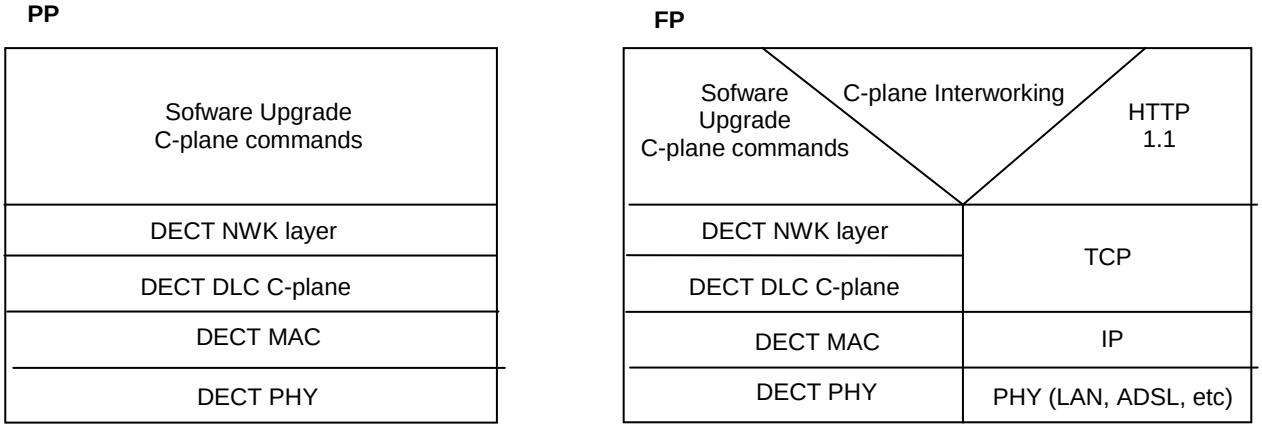


Figure 2: Reference model of the C-plane protocol stack

4.3 Description of functionality and functional split between PT and FT

The top level service provided to the Portable Part implementing the present document is a layer 7 application based on the IETF protocol HTTP [19]. Assuming the most usual scenario, as described in clause 4.1.2, the application level interaction happens between the DECT PP-that is therefore the end-point of the application protocol-and any host located at any Internet location, with TCP/IP connectivity towards the DECT system.

The server located on the Internet-also called the network side server-is a regular "host" from Internet point of view, i.e. it is a host for IP [i.16], TCP [i.18] and the application layer. However, in the DECT system, the IP and TCP layers are terminated at the FP, while only the application layer travels to the PP. Therefore the DECT FP is the end-point for IP and TCP connections.

4.3.1 Functionality of the DECT FP

In addition to being the end-point of IP [i.16] and TCP [i.18] layers, the FP may be involved in a PP data session in different ways:

- Simple forwarding of PP requests to the WAN (to DECT specific service), with possible adjustment of requests (Range header).

- Forwarding of PP requests to the WAN (to a non DECT specific service), with adjustment of responses to the limited profiles used by the PP ("HTTP Common profile", "HTTP Browsing profile", "Light Browsing profile").
- Addition of a security service: creation of an SSL/TLS tunnel from FP to HTTP server. Refer to [i.20] for details.
- Hosting of an HTTP server for local services.

4.3.2 Functionality of the DECT PP

The DECT PP acts as the end-point for the application protocol (HTTP). The PP is usually the "client" in a client/server relationship with the network host that is usually the "server". However this asymmetric relation is irrelevant from the DECT point of view, that would as well support the opposite client/server relationship if required.

4.3.3 U-plane interworking and protocol architecture

The U-plane protocol architecture (see figure 1) reflects the principle of TCP and IP termination at FP side. The application protocol (HTTP in all cases covered by the present document) is transported over DPRS LU10 service using the generic encapsulation described in clause B.8 of ETSI EN 301 649 [15].

The fact that the air interface transport is done without a transport layer, requires some cooperation from the application layer. In the case of HTTP, the message oriented structure of the protocol with a request/response operation mode in near all cases, makes this direct operation possible. The design of the application at both places should, however, take into account that it will be transported without a transport layer (TCP) over part of the data path.

The DPRS U-plane transport (LU10) is able to deliver the application packets in a reliable way and is able to carry information about the position of the application packet boundaries. In the event of impossibility of reliable transmission (for instance due to successive radio errors), the DPRS U-plane transport is able to indicate to application layer that the received application packet is not complete.

4.3.4 Dynamic considerations on U-plane

Limiting ratio "Application packet size / PDU size". Due to the operation without transport layer, one of the factors that should be considered by the HTTP application design at both sides is using a correct value for the maximum size of the HTTP packets (messages). While the operation with TCP allows the transportation of variable size and virtually very large application packets, in the scenario covered by the present document, it is convenient to restrict the size of the packets to proper values, that makes the ratio "size of the application packet/size of the DECT DLC segment" not too large. The DECT DLC segment is the PDU of the LU10, with a size of 64 bytes for 2-level modulation and long slot.

NOTE 1: The current version of the present document does not use the optional application packet segmentation into several SDUs (a.k.a 'chopping' facility); so using the 'application packet size' here is roughly equivalent to using the 'SDU size' for computing the ratio. If chopping were used (see ETSI EN 301 649 [15], clauses B.8.2 and 12.8.4 for indicator in the <<SETUP CAPABILITY>> IE), using the 'SDU size' instead of the 'Application packet size' might be more relevant in some situations and not in other, and this is therefore left for further study.

Increasing ratio "Application packet size / PDU size" upon good radio link quality. For optimal operation, the value of this ratio should be dynamically adjusted depending on the radio conditions: for optimal radio link quality the size of application packets (and the ratio) may be increased. If the radio link quality decreases, and there are transmission errors and DLC retransmissions, the ratio should be reduced, by reducing the maximum size of the application packets.

NOTE 2: However, this capability of dynamically adjusting the application packet depending on radio conditions requires communication between the DECT Management Entity and the application layer, and is seen as an option for advanced implementations.

Minimum application packet size. From a radio link quality perspective, a value of 12 kBytes (equivalent to 200 times the DECT PDU size) is considered a convenient value for the maximum application packet size in FT => PT direction, and this is the mandatory figure to be supported by all devices implementing the present document. For the PT => FT direction, the mandatory value to be supported by all implementations is 1,5 kBytes.

The implementations may optionally support the use of larger application packet sizes. The supported maximum size at both sides (when different of the default values) shall be announced by means of the <<Setup Capability>> IE (see clause 12.22 of ETSI EN 301 649 [15]). Application packet sizes larger than the default values may only be used when supported by both peers.

Maximum SDU size. Although a minimum application packet size is defined from a radio link perspective, the actual application packet size may be limited by memory constraints on real devices on sending and receiving sides. This constraint is expressed in terms of maximum size of SDUs. The Maximum value for the SDU size is negotiated on a per connection basis with the <<IWU-Attributes>> IE (see clause 7.6.1.2.1.3 and ETSI EN 301 649 [15], clause B.2.1).

Because chopping is not used in the present document, the size of an application packet shall therefore not exceed the maximum SDU size (minus SDU header size = 2) used for the connection.

NOTE 3: The maximum value for the "Max SDU size" fields is 131 040 bytes (16 380 × 8) (see ETSI EN 301 649 [15], clause B.2.1).

The "Range" header of HTTP/1.1 should be used for controlling the application packet size at application level. The value set in the Range header should be consistent with the values announced in the <<SETUP CAPABILITY>> IE.

Application level packet size control. When application packet sizes larger than the default values are supported, it is assumed that the implementation has the capability to adjust the real size of the packets according to radio conditions.

NOTE 4: It means that the applications should not use always, by default, the maximum supported size.

PPs should control the size of the uplink application packets by deliberately limiting the size of application packets it issues (e.g. avoiding the use of unnecessary HTTP headers), and should control the size of downlink application packets by means of the "Range" header in HTTP/1.1 requests.

Application level packet size versus radio conditions. The exact value of the application packet size, and how it is related to the radio conditions is up to the implementer.

NOTE 5: A value of 32 000 bytes is equivalent to approximately 5 seconds of transmission over a single bearer, long slot channel and optimal radio conditions.

4.3.5 C-plane interworking

The DECT C-plane is used for different functions in the Light Data Services architecture. One of the most obvious is the DPRS virtual call setup and release. In most application cases, this is usually done by the PP side after request of the data application (see for instance the figure 3). On the FP side, the sending of HTTP requests through the set-up virtual call triggers the opening of one or several TCP connections with the distant Internet host, or hosts (since in some scenarios several hosts participate in the service). In some cases, the DECT C-plane is used to transport the routing information (URL) of this network side server (see clause 7.5.5.2.1).

It is also possible to use the opposite scenario, where the DECT FP receives a TCP connection from a distant host, and this action triggers a FP initiated virtual call setup towards the PP. This scenario is, however, less usual and, because of it, the FP initiated virtual call is optional.

4.4 Service and Performance Objectives

In order to allow wide use of the present specification, and taking into account that in most cases it will be a complement of DECT voice services, only single bearer operation over long slot and DPRS Class 4 are defined as mandatory. There is however the option of using DPRS Class 3 or Class 2 management, and, if Class 2 is used, the option of offering multibearer and asymmetric bearers support.

The service and performance objectives are shown in tables 1 and 2.

Table 1: Summary of service capabilities

Service	DPRS Class mode		
	Class 4	**Class 3**	**Class 2**
Point-to-point protected data transfer PP-FP with ARQ	YES	YES	YES
Point-to-point protected data transfer FP-PP with ARQ	YES	YES	YES
Point-to-multi-point data transfer FP-PP	-	-	-
Point-to-point data transfer PP-PP (distributed communication)	-	-	-
Authentication	YES	YES	YES
Encryption	YES	YES	YES
Connection oriented operation	YES	YES	YES
Virtual Call (VC) operation	YES	YES	YES
Intra-cell bearer handover	YES	YES	YES
Inter-cell bearer handover	YES	YES	YES
Inter-cell connection handover (for multicell systems)	OPTIONAL	OPTIONAL	OPTIONAL
Inter-cell external handover	OPTIONAL	OPTIONAL	OPTIONAL
Suspend /resume	-	YES	YES
Multibearer connections			YES
Asymmetric connections			YES
Fast setup			OPTIONAL

Table 2: Summary of Performance objectives

Performance	DPRS Class	
	Class 3 or 4	**Class 2**
Maximum supported SDU size (FT = > PT)	≥ 752 octets	≥ 752 octets
Maximum supported SDU size (PT = > FT)	≥ 752 octets	≥ 752 octets
Mandatory sustainable unidirectional or bidirectional throughput.	64 kbit/s	64 kbit/s
Optional maximum sustainable unidirectional throughput (Class 2 system), GFSK 2-level modulation.	-	844,8 kbit/s net
Optional sustainable full bi-directional throughput (DPRS Class 2 system), GFSK 2-level modulation	-	460,8 kbit/s net
Maximum one-way delay	Down to 50 ms configurable	Down to 50 ms configurable
Establishment of PT to FT physical connection (average)	< 50 ms	< 50 ms
Establishment of FT to PT physical connection (average)	< 50 ms	< 50 ms
Undetected bit error ratio	$< 10^{-10}$	$< 10^{-10}$
Uncorrected bit error ratio (for air interface BER 10^{-3} and delay = 100 ms)	$< 10^{-7}$	$< 10^{-7}$

4.5 General application environments

Figures 3 and 4 describe the general application scenario of the Light Data Services described in the present document.

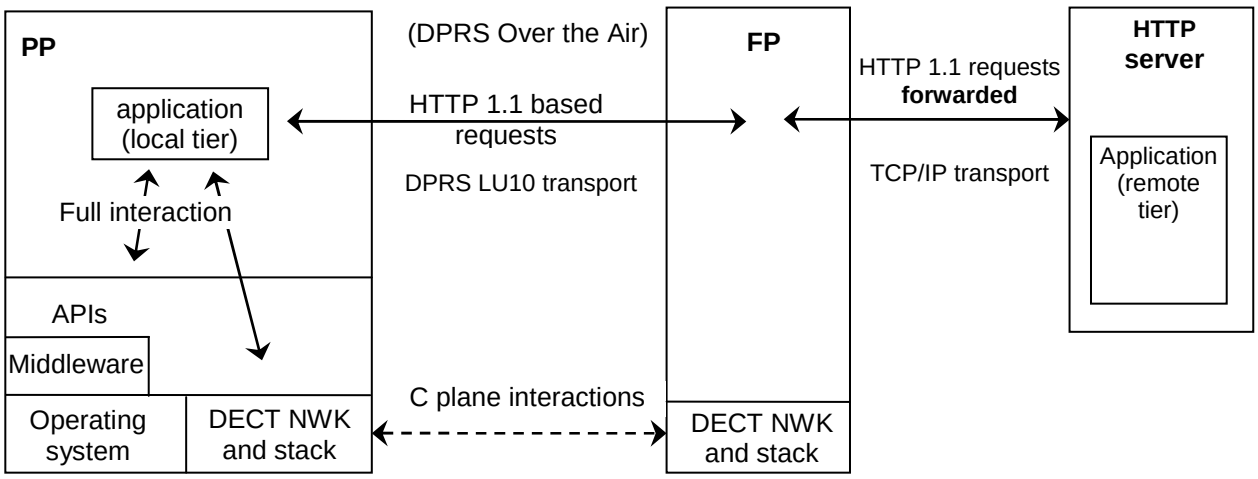


Figure 3: General application scenario (applicable to all services)

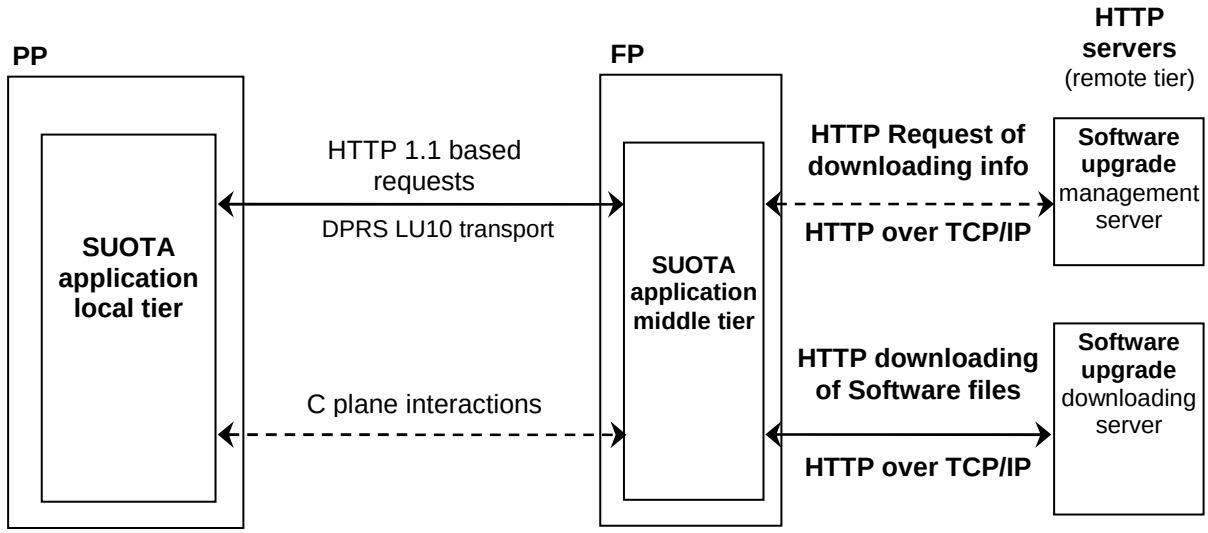


Figure 4: Application scenario for Software upgrade over the air (SUOTA)

5 Relevant requirements

The requirements of ETSI EN 301 649 [15] relevant for Class 2, Class 3 or Class 4 equipment shall apply with the modifications stated in clauses 5 and 6 of the present document.

The encapsulation of external data protocol shall be done as stated in ETSI EN 301 649 [15], clause B.8.

In any case, the requirements of ETSI EN 300 176-1 [9] shall apply as well.

5.1 Service and feature definitions

5.1.1 PHL service definitions

For the purpose of the present document, the definitions of ETSI EN 301 649 [15], clause 4.3.1 shall apply.

5.1.2 MAC service definitions

For the purposes of the present document, the definitions of ETSI EN 301 649 [15], clause 4.3.2 shall apply.

5.1.3 DLC service definitions

For the purposes of the present document, the definitions of ETSI EN 301 649 [15], clause 4.3.3 shall apply.

5.1.4 NWK feature definitions

For the purposes of the present document, the definitions of ETSI EN 301 649 [15], clause 4.3.4 plus the following shall apply:

General Light Data Service Procedures [NGLDS-N.1]: NWK layer procedures needed for the operation of this profile.

Software upgrade over the air, C-plane [NGLDS-N.2]: NWK layer procedures needed for upgrading the software of the PP.

5.1.5 Application service definitions

For the purposes of the present document, the definitions of ETSI EN 301 649 [15], clause 4.3.5 plus the following shall apply:

Binary content download [NGLDS-A.1]: Ability to download binary files (or other) from a content server using HTTP protocol.

Software upgrade over the air [NGLDS-A.2]: Ability to upgrade the software of the PP.

HTTP-based applications [NGLDS-A.3]: Ability to browse HTML pages from the PP.

5.1.6 Management Entity (ME) definitions

For the purposes of the present document, the definitions of ETSI EN 301 649 [15], clause 4.3.7 shall apply.

5.1.7 Call Control (CC) and mobility management service definitions

For the purposes of the present document, the definitions of ETSI EN 301 649 [15], clause 4.3.8 shall apply.

5.1.8 U-plane service and interworking definitions

For the purposes of the present document, the definitions of ETSI EN 301 649 [15], clause 4.3.9 shall apply.

6 Profile specific requirements

6.1 General

The tables listed in this clause define the status of all protocol elements (i.e. features, services, and procedures), which can be: mandatory, optional, conditional under the provision of another protocol element, outside the scope of the present document, or not applicable. The status is identified by the status column designations defined in clause 3.2, and is described separately for FT and PT.

All optional elements shall be process mandatory according to the procedures described in the present document.

Protocol elements defined as mandatory, optional or conditional in this clause are further defined in the referenced DECT specification, or, if needed, in clause 7 of the present document.

New Generation DECT; part 4 is defined as an application specific access profile of DPRS [15]. All procedures not specific to the New Generation DECT, part 4, are referenced to their original description in ETSI EN 301 649 (DPRS) [15].

The requirements of ETSI EN 301 649 [15] relevant for Class 2, Class 3 or Class 4 (depending on the supported Class(es)) equipment shall apply with the modifications stated, if needed, in clause 7 of the present document.

The encapsulation of external data protocol shall be done as stated in ETSI EN 301 649 [15], clause B.8 (Generic media encapsulation) and clause B.8.3.4 (HTTP limited set nr. 2).

NOTE: The HTTP limited set nr. 2 (ETSI EN 301 649 [15], clause B.8.3.4) implements the mandatory requirements of the present document (see clauses 7.6, A.1 and A.2). The HTTP limited set nr. 2, contrarily to the HTTP limited set nr. 1 (ETSI EN 301 649 [15], clause B.8.3.3), does not require that all FP implementations include a full HTTP implementation. Some FPs could even be designed to be as transparent as possible from the application-layer point of view (i.e. HTTP), relying on the HTTP subset implemented by the PP to ensure the dialog with the server. One of the reasons for this is that full-featured browsing is not among the first targeted applications of the present document (although some PP implementations could be rich enough to allow it). Also, some of the targeted applications could rely on a NG-DECT part 4 specific server
(a server implementing the requirements of the present document), thus purposely restricting the set of HTTP features used on server side.

In any case, the requirements of ETSI EN 300 176-1 [9] and ETSI EN 300 176-2 [10] shall be met by all equipment conforming to the present document.

The requirements tables in the following clauses are derived from the ETSI EN 301 649 [15]. In the service to procedure and feature to procedure mapping tables, the status of each particular item is explicitly stated only when it constitutes a change to the status indicated in ETSI EN 301 649 [15].

6.2 General class/service/interworking support

6.2.1 Class/service support

The following service classes and end-user services shall be supported by New Generation DECT, part 4 equipment.

Table 3: General class and service support

Item	Name of service	Reference	Support status	
			PT	FT
DPRS-G.1	DPRS Class 1	4.3.8 [15]	I	I
DPRS-G.2	DPRS Class 2	4.3.8 [15]	O	O
DPRS-G.3	Frame Relay (FREL)	4.3.9 [15] and annex B [15]	M	M
DPRS-G.4	Character stream	4.3.9 [15] and annex C [15]	I	I
DPRS-G.5	DPRS Class 3	4.3.9 [15] and annex C [15]	O	O
DPRS-G.6	DPRS Class 4	4.3.9 [15] and annex C [15]	M	M
NOTE: The reference column refers to the relevant clause in the referenced document.				

6.2.2 Protocol interworking support

The following protocol interworking modes shall be supported by New Generation DECT, part 4 equipment.

Table 4: General service/interworking support

			Status	
Service	**Interworking**	**Reference**	**PT**	**FT**
DPRS-G.3, Frame Relay (FREL)		4.3.9 [15] and annex B [15]	M	M
	DPRS-I.1, Ethernet	4.3.9 [15] and B.4 [15]	I	I
	DPRS-I.2, Token Ring	4.3.9 [15] and B.5 [15]	I	I
	DPRS-I.3, IP	4.3.9 [15] and B.6 [15]	I	I
	DPRS-I.4, PPP	4.3.9 [15] and B.7 [15]	I	I
	DPRS-I.5, Generic media encapsulation	4.3.9 [15], B.8 [15] and 7.6.1.2 (see note 2)	M	M
DPRS-G.4, Character stream		4.3.9 [15] and annex C [15]	I	I
	DPRS-I.6, V.24	4.3.9 [15] and C.4 [15]	I	I
NOTE 1: The reference column refers to the relevant clause in the present or in the referenced document. NOTE 2: The applicable clauses in 7.6.1.2 according to table 13 shall apply.				

On regard to the Interworking conventions, the specific interworking requirements described in clause 7.7 shall also apply.

6.3 Void

6.4 Physical layer (PHL) requirements

6.4.1 Physical layer (PHL) services

New Generation DECT, part 4 devices shall support the following Physical layer (PHL) services.

Table 5: Physical layer service support [15]

Item	**Name of service**	**Reference**	**Support status**	
			PT	**FT**
DPRS-P.1	GFSK modulation	4.3.1 [15]	M	M
DPRS-P.2	π/2 DBPSK modulation	4.3.1 [15]	O	O
DPRS-P.3	π/4 DQPSK modulation	4.3.1 [15]	O	O
DPRS-P.4	π/8 D8PSK modulation	4.3.1 [15]	O	O
DPRS-P.5	16 QAM modulation	4.3.1 [15]	O	O
DPRS-P.6	64 QAM modulation	4.3.1 [15]	O	O
DPRS-P.7	Physical Packet P32	4.3.1 [15]	O	O
DPRS-P.8	Physical Packet P64	4.3.1 [15]	M	M
DPRS-P.9	Physical Packet P67	4.3.1 [15]	O	O
DPRS-P.10	Physical Packet P80	4.3.1 [15]	O	O
DPRS-P.11	General PHL	4.3.1 [15]	M	M
DPRS-P.12	Fast hopping radio	4.3.1 [15]	O	O
NOTE: The reference column refers to the relevant clause in the referenced document.				

6.4.2 Modulation schemes

The following modulation schemes defined by ETSI EN 300 175-2 [2], annex D shall be supported.

Table 6: Allowed combinations of modulation schemes

Modulation scheme	S-field	A-field	B + Z-field	Support status
1a	GFSK	GFSK	GFSK	M
1b	π/2-DBPSK	π/2-DBPSK	π/2-DBPSK	O
2	π/2-DBPSK	π/2-DBPSK	π/4-DQPSK	O
3	π/2-DBPSK	π/2-DBPSK	π/8-D8PSK	O
5	π/2-DBPSK	π/2-DBPSK	16 QAM	O [15]
6	π/2-DBPSK	π/2-DBPSK	64 QAM	O

6.4.3 PHL service to procedure mapping

The PHL service to procedure mapping of ETSI EN 301 649 [15], clause 5.3 shall apply.

6.5 MAC layer requirements

6.5.1 MAC layer services

New Generation DECT data devices shall support the following MAC layer services.

Table 7: MAC service support

Item	Name of service	Reference (note 1)	Support status	
			PT	FT
DPRS-M.1	General	4.3.2 [15]	M	M
DPRS-M.2	Non continuous broadcast	4.3.2 [15]	O	O
DPRS-M.3	Continuous broadcast	4.3.2 [15]	M	M
DPRS-M.4	Paging broadcast	4.3.2 [15]	M	M
DPRS-M.5	B-field advanced connection control	4.3.2 [15]	O	O
DPRS-M.6	I_{PM}_error_detection	4.3.2 [15]	M	M
DPRS-M.7	I_{PM}_error_correction	4.3.2 [15]	O	O
DPRS-M.8	U-plane point-to-multipoint service	4.3.2 [15]	I	I
DPRS-M.9	C_S higher layer signalling	4.3.2 [15]	M	M
DPRS-M.10	C_F higher layer signalling	4.3.2 [15]	O	O
DPRS-M.11	Encryption activation(GAP-M.7)	4.3.2 [15]	M	M
DPRS-M.12	Encryption deactivation(GAP-M.14)	4.3.2 [15]	C73	C73
DPRS-M.13	Quality control	4.3.2 [15]	M	M
DPRS-M.14	Physical channel selection	4.3.2 [15]	M	M
DPRS-M.15	SARI support	4.3.2 [15]	M	O
DPRS-M.16	DPRS Bearer handover	4.3.2 [15]	M	M
DPRS-M.17	Fast setup	4.3.2 [15]	C74	C74
DPRS-M.18	Connection handover	4.3.2 [15]	O	O
DPRS-M.19	G_F channel	4.3.2 [15]	C76	C76
DPRS-M.20	I_{PQ}_error_detection	4.3.2 [15]	O	O
DPRS-M.21	I_{PQ}_error_correction	4.3.2 [15]	O	O
DPRS-M.22	I_{Px}_encoded protected	4.3.2 [15]	C75	C75
DPRS-M.23	I_{PF} channel	4.3.2 [15]	C76	C76
DPRS-M.24	Full slot	4.3.2 [15]	O	O
DPRS-M.25	Long slot 640	4.3.2 [15]	M	M

Item	Name of service	Reference (note 1)	Support status	
			PT	FT
DPRS-M.26	Long slot 672	4.3.2 [15]	O	O
DPRS-M.27	Double slot	4.3.2 [15]	O	O
DPRS-M.28	Multibearer connections	4.3.2 [15]	C74	C74
DPRS-M.29	Asymmetric connections	4.3.2 [15]	C74	C74
DPRS-M.30	Simplified A-field advanced connection control	4.3.2 [15]	M	M
DPRS-M.31	Re-keying (GAP.M.15)	4.3.2 [15]	C77	C77
DPRS-M.32	Early encryption (GAP.M.16)	4.3.2 [15]	C78	C78
DPRS-M.33	AES/DSC2 encryption (GAP.M.17) (see note 2)	4.3.2 [15]	O	O
C73: IF DPRS-N.28 or DPRS-N.29 then M else I. C74: IF DPRS-M.5 THEN O ELSE I. C75: IF DPRS-P.5 (16 QAM) OR DPRS-P.6 (64 QAM) THEN M ELSE O. C76: IF DPRS-M.29 THEN M ELSE O. C77: IF NWK layer procedure "Re-keying during a call" of DPRS-N.43 (GAP.N.35) is implemented THEN M ELSE O. C78: IF NWK layer procedure "Early encryption" of DPRS-N.43 (GAP.N.35) is implemented THEN M ELSE O.				
NOTE 1: The reference column refers to the relevant clause in the referenced document. NOTE 2: IF implemented THEN DPRS-N.44 (GAP.N.36) shall be implemented.				

6.5.2 MAC service to procedure mapping

The MAC layer service to procedure mapping specified in ETSI EN 301 649 [15], clause 6.2, with the following changes and additional features shall apply [15].

Table 8: MAC service to procedure mapping

			Status	
Service	Procedure	Reference (clause)	PT	FT
DPRS-M.1 General		4.3.2 [15]	M	M
	Frame and Multiframe structure	10.1.1 [15]	M	M
	Bit mappings	10.1.2 [15]	M	M
	Scrambling	10.1.4 [15]	M	M
	Error control	10.1.5 [15]	M	M
	RFP idle receiver scan sequence	10.1.8 [15]	M	M
	PT states and state transitions for PTs not supporting fast setup	10.1.10.1 [15]	C802	C802
	Identities	10.1.11 [15]	M	M
	A-field Multiplexer (T-MUX)	10.21.1 [15]	M	M
	B-field control Multiplexer (E/U-MUX), basic modes	10.21.2.1 [15]	C804	C804
DPRS-M.2 Non continuous broadcast		4.3.2 [15]	O	O
	Request for specific Q channel information	10.2.1 [15]	O	O
	Request for a new dummy	10.2.2 [15]	O	O
DPRS-M.3 Continuous broadcast		4.3.2 [15]	M	M
	Downlink broadcast	10.3 [15]	M	M
DPRS-M.4 Paging broadcast		4.3.2 [15]	M	M
	Paging messages	10.4.1 [15]	M	M
	MAC layer information messages procedures	10.4.2 [15]	M	M
	LCE paging procedure	10.4.3.1 [15]	M	M
	MAC paging procedure	10.4.3.2 [15]	C801	C801
	Paging detection: High duty cycle (when there is an active virtual connection in suspend state)	10.4.4.2 [15]	C809	C809

			Status	
Service	**Procedure**	**Reference (clause)**	**PT**	**FT**
	Paging detection: High duty cycle (when there is no active virtual connection)	10.4.4.2 [15]	O	O
	Paging detection: Normal duty cycle (when there is an active virtual connection in suspend state)	10.4.4.1 [15]	C810	C810
	Paging detection: Normal duty cycle (when there is no active virtual connection)	10.4.4.1 [15]	C811	C811
	Paging detection: Low duty cycle (when there is no active virtual connection)	10.4.4.3 [15]	O	O
DPRS-M.5 B-field advanced connection control		4.3.2 [15]	O	O
	Fast setup	10.10.1.2 [15]	O	O
	idle-locked state with set-up detection	11.1.3.2 [3]	O	I
	Logical connection setup	10.5 [15]	M	M
	Logical connection release	10.6 [15]	M	M
	Connection modification to change bandwidth (including suspend)	10.7.1 [15]	M	M
	Connection modification to change MAC service type	10.7.2.1 [15]	O	O
	Connection modification to change slot type	10.7.2.2 [15]	O	O
	Connection modification to change modulation scheme or adaptive codec rate	10.7.2.3 [15]	O	O
	B-field Single bearer Physical connection setup	10.8.1 [15]	M	M
	B-field Physical Connection release	10.9 [15]	M	M
	B-field Single duplex bearer setup	10.10.1 [15]	M	M
	Usage of channel list messages	10.10.1.3 [15]	M	M
	B-field Crossed bearer release	10.11.2 [15]	O	O
	B-field Unacknowledged bearer release	10.11.1 [15]	M	M
	B-field Acknowledged bearer release	10.11.3 [15]	O	O
DPRS-M.8 U-plane point-to-multipoint service		4.3.2 [15]	I	I
	Connectionless SI_P mode	10.13.3 [15]	M	M
DPRS-M.9 C_S higher layer signalling		4.3.2 [15]	M	M
	C_S channel data	10.14.1 [15]	M	M
DPRS-M.10 C_F higher layer signalling		4.3.2 [15]	O	O
	C_F channel data	10.14.2 [15]	M	M
	B-field control Multiplexer (E/U-MUX), C_F modes	10.21.2.2 [15]	M	M
DPRS-M.11 Encryption activation (GAP-M.7)		4.3.2 [15]	M	M
	Encryption process - initialization and synchronization	10.15.1 [15]	M	M
	Encryption mode control	10.15.2 [15]	M	M
	Encryption handover control	10.15.3 [15]	M	M
DPRS-M.12 Encryption deactivation (GAP-M.14)		4.3.2 [15]	C73	C73
	Encryption mode control	10.15.2 [15]	M	M
DPRS-M.13 Quality control		4.3.2 [15]	M	M
	RFPI handshake	10.16.1 [15]	M	M
	PT frequency correction procedure	10.16.2 [15]	O	O
	Bearer quality report	10.16.3 [15]	M	M
	Bearer quality report for asymmetric bearers (MAC-mod2-ACK)	10.16.3.1 [15]	C803	C803
	Bearer and connection control	10.16.4 [15]	O	O

			Status	
Service	**Procedure**	**Reference (clause)**	**PT**	**FT**
	A-CRC handshake	10.16.5 [15]	M	M
DPRS-M.14 Physical channel selection		4.3.2 [15]	M	M
	Physical channel selection	10.17 [15]	M	M
DPRS-M.15 SARI support		4.3.2 [15]	M	O
	Downlink broadcast	10.3.2.3 [15]	M	M
DPRS-M.16 DPRS Bearer handover		4.3.2 [15]	M	M
	B-field MAC Bearer replacement procedure	10.18 [15]	C804	C804
	B-field MAC Bearer handover procedure	10.19 [15]	C805	C805
	A-field Bearer handover request (M_T)	10.23.4 [15]	C806	C806
DPRS-M.17 fast setup		4.3.2 [15]	C74	C74
	FT initiated initial duplex bearer setup	10.10.1.3 [15]	M	M
	PT receiver scan sequence	10.1.9 [15]	M	M
	Fast setup control in MAC resume and control page message	10.4.3.2.2 [15]	M	O
	PT states and state transitions for PTs supporting fast setup	10.1.10.2 [15]	M	M
	Listen for setup control codes in Release message	10.11.6 [15]	M	M
DPRS-M.18 Connection handover		4.3.2 [15]	O	O
	B-field Advanced connection handover	10.12 [15]	C804	C804
	A-field connection handover request (MT)	10.23.5 [15]	C806	C806
DPRS-M.19 G_F channel		4.3.2 [15]	C76	C76
	G_F channel transmission	10.20.1.1 [15]	O	O
	G_F channel data reception	10.20.1.2 [15]	M	M
DPRS-M.6 $I_{PM_}$error_detection service		4.3.2 [15]	M	M
	Type 3: $I_{P\_}$ error_detection symmetric MAC service	5.6.2.1 [3]	M	M
	Type 7: $I_{P\_}$ error_detection asymmetric MAC service	5.6.2.2 [3]	C803	C803
	Multi-subfield protected B-field	6.2.1.3.3 [3]	M	M
	Q1/Q2 bit setting for: $I_{P\_}$ error_detection	10.8.1.3.2 [3]	M	M
	Protected I channel error_detect procedure	10.13.1 [15]	M	M
DPRS-M.7 $I_{PM_}$error_correction service		4.3.2 [15]	O	O
	Type 4: $I_{P\_}$ error_correction symmetric MAC service	5.6.2.1 [3]	M	M
	Type 8: $I_{P\_}$ error_correction asymmetric MAC service	5.6.2.2 [3]	C803	C803
	Multi-subfield protected B-field	6.2.1.3.3 [3]	M	M
	MOD-2 protected channel operation	10.8.2 [3]	M	M
	Protected I channel error_correct mode	10.13.2 [15]	M	M
DPRS-M.20 $I_{PQ_}$error_detection service		4.3.2 [15]	O	O
	Type 3: $I_{P\_}$ error_detection symmetric MAC service	5.6.2.1 [3]	M	M
	Type 7: $I_{P\_}$ error_detection asymmetric MAC service	5.6.2.2 [3]	C803	C803
	Single-subfield protected B-field	6.2.1.3.4 [3]	M	M
	Q1/Q2 bit setting for: $I_{P\_}$ error_detection	10.8.1.3.2 [3]	M	M

			Status	
Service	**Procedure**	**Reference (clause)**	**PT**	**FT**
	Protected I channel error_detect procedure	10.13.1 [15]	M	M
DPRS-M.21 I_{PQ}_error_correction service		4.3.2 [15]	O	O
	Type 4: I_P_ error_correction symmetric MAC service	5.6.2.1 [3]	M	M
	Type 8: I_P_ error_correction asymmetric MAC service	5.6.2.2 [3]	C803	C803
	Single-subfield protected B-field	6.2.1.3.4 [3]	M	M
	MOD-2 protected channel operation	10.8.2 [3]	M	M
	Protected I channel error_correct mode	10.13.2 [15]	M	M
DPRS-M.22 I_{PX}_encoded protected		4.3.2 [15]	C75	C75
	Type 5: I_P_ encodec protected symmetric MAC service	5.6.2.1 [3]	M	M
	Type 9: I_P_ encodec protected asymmetric MAC service	5.6.2.2 [3]	C803	C803
	Channel coding	I.1 [3]	M	M
DPRS-M.23 I_{PF} channel		4.3.2 [15]	C76	C76
	B-field control Multiplexer (E/U mux), E+U mode	10.22.2.3 [15]	M	M
	I_{PF} channel general	10.22.1 [15]	M	M
	I_{PF} channel advanced procedures	10.22.2 [15]	O	O
	I_{PF} channel error correct procedures	10.22.3 [15]	C807	C807
	SI_{PF} channel	10.22.4 [15]	C808	C808
DPRS-M.24 Full slot		4.3.2 [15]	O	O
	D-field mapping for the full slot structure (physical packet P32)	6.2.1.1.2 [3]	M	M
	B-field mapping for the full slot structure (physical packet P32)	6.2.1.3.1.2 [3]	M	M
DPRS-M.25 Long slot 640		4.3.2 [15]	M	M
	D-field mapping for the variable slot structure (physical packet P00j) with j = 640	6.2.1.1.4 [3]	M	M
	B-field mapping for the half and long slot structure (physical packet P00j) with j = 640	6.2.1.3.1.3 [3]	M	M
	Additional procedures for Long and double slots	D.2 [15]	M	M
DPRS-M.26 Long slot 672		4.3.2 [15]	O	O
	D-field mapping for the variable slot structure (physical packet P00j) with j = 672	6.2.1.1.4 [3]	M	M
	B-field mapping for the half and long slot structure (physical packet P00j) with j = 672	6.2.1.3.1.3 [3]	M	M
	Additional procedures for Long and double slots	D.2 [15]	M	M
DPRS-M.27 Double slot		4.3.2 [15]	O	O
	D-field mapping for the double slot structure (physical packet P80)	6.2.1.1.1 [3]	M	M
	B-field mapping for the double slot structure (physical packet P80)	6.2.1.3.1.1 [3]	M	M

			Status	
Service	**Procedure**	**Reference (clause)**	**PT**	**FT**
	Additional procedures for Long and double slots	D.2 [15]	M	M
DPRS-M.28 Multibearer connections		4.3.2 [15]	C74	C74
	Multi bearer Physical connection setup	10.8.2 [15]	M	M
	MBC Multibearer control	10.8.2.1 [15]	M	M
DPRS-M.29 Asymmetric connections		4.3.2 [15]	C74	C74
	Double simplex bearers	10.10.2 [15]	M	M
	Upstream Double simplex bearer setup (B-field)	10.10.2.2 [15]	M	M
	Downstream Double simplex bearer setup (B-field)	10.10.2.2 [15]	M	M
	Fast bearer release	10.11.3 [15]	M	M
	Unacknowledged double simplex bearer release	10.11.1 [15]	M	M
	Acknowledged double simplex bearer release	10.11.2 [15]	O	O
DPRS-M.30 simplified A-field connection control		4.3.2 [15]	M	M
	PT initiated A-field advanced bearer setup (MT)	10.23.2 [15]	M	M
	Connection/bearer release (MT)	10.23.3 [15]	M	M
	Connection modification to change MAC service type	10.7.2.1 [15]	O	O
	Connection modification to change slot type	10.7.2.2 [15]	O	O

C73: IF DPRS-N.28 or DPRS-N.29 THEN M ELSE I.
C74: IF DPRS-M.5 THEN O ELSE I.
C75: IF DPRS-P.5 (16 QAM) OR DPRS-P.6 (64 QAM) THEN M ELSE O
C76: IF DPRS-M.29 THEN M ELSE O.
C801: IF DPRS-M.5 THEN M ELSE O.
C802: IF DPRS-M.17 THEN I ELSE M.
C803: IF DPRS-M.29 THEM M ELSE I.
C804: IF DPRS-M.5 THEM M ELSE I.
C805: IF DPRS-M.5 THEM O ELSE I.
C806: IF DPRS-M.30 THEM M ELSE I.
C807: IF DPRS-M.7 OR DPRS-M.21 THEM M ELSE I.
C808: IF DPRS-M.8 THEM O ELSE I.
C809: IF (DPRS-ME.2 OR DPRS-ME.3) THEM O ELSE I.
C810: IF (DPRS-ME.2 OR DPRS-ME.3) THEM M ELSE I.
C811: IF DPRS-N.8 THEM M ELSE O.

NOTE: The reference column refers to the relevant clause in ETSI EN 301 649 [15] or in the referenced document.

6.6 DLC layer

6.6.1 DLC layer services

New Generation DECT, part 4 devices shall support the following DLC layer services.

Table 9: DLC service status

			Status	
Item no.	**Name of service**	**Reference**	**PT**	**FT**
DPRS-D.1	LU10 Enhanced Frame RELay service (EFREL)	4.3.3 [15]	M	M
DPRS-D.2	FU10a	4.3.3 [15]	M	M
DPRS-D.3	FU10b	4.3.3 [15]	I	I
DPRS-D.4	FU10c	4.3.3 [15]	M	M
DPRS-D.5	Data Link Service (LAPC + Lc) class A service	4.3.3 [15]	M	M
DPRS-D.6	Data Link Service (LAPC + Lc) class U service	4.3.3 [15]	O	O
DPRS-D.7	Lc Frame delimiting and sequencing service	4.3.3 [15]	M	M
DPRS-D.8	Broadcast Lb service	4.3.3 [15]	M	M
DPRS-D.9	Inter-cell voluntary connection handover	4.3.3 [15]	O	O
DPRS-D.10	Connection modification	4.3.3 [15]	C92	C92
DPRS-D.11	Encryption activation (GAP-D.6)	4.3.3 [15]	M	M (note 2)
DPRS-D.12	Encryption deactivation (GAP-D.9)	4.3.3 [15]	C91	C91
DPRS-D.13	Connectionless U-plane	4.3.3 [15]	I	I
C91: If DPRS-N.28 or DPRS-N.29 then M else I. C92: (If DPRS-M.5 THEN M ELSE O).				
NOTE 1: The reference column refers to the relevant clause in the referenced document. NOTE 2: The status in DRPS (and GAP) is conditional: "IF feature GAP.N.17 OR GAP.N.27" THEN M ELSE I.", but GAP.N.17 becomes M when DPRS-N.43 (a.k.a GAP.N.35, "Enhanced security") is implemented, which is the case for the present document.				

6.6.2 DLC service to procedure mapping

The DLC layer service to procedure mapping specified in ETSI EN 301 649 [15], clause 7.2 shall apply.

6.7 NWK layer

6.7.1 General

The NWK layer provisions shall include the following entities:

- Call Control (CC).
- Mobility Management (MM).
- Link Control Entity (LCE).
- ConnectionLess Message Service (CLMS).

New Generation DECT data equipment is based on DPRS Class 2, 3 or 4 management (see [15], clause 4.3.8), and therefore requires a NWK layer.

6.7.2 NWK features

New Generation DECT data devices shall support the following NWK layer features.

Table 10: NWK features status

Feature supported				
Features			Status	
Item no.	Name of feature	Reference (note 1)	PT	FT
DPRS-N.1	PT initiated virtual call	4.3.4 [15]	M	M
DPRS-N.2	Off hook	4.3.4 [15]	M	M
DPRS-N.3	On hook (full release)	4.3.4 [15]	M	M
DPRS-N.4	Dialled digits (basic)	4.3.4 [15]	O	O
DPRS-N.5	Register recall	4.3.4 [15]	O	O
DPRS-N.6	Go to DTMF signalling (defined tone length)	4.3.4 [15]	O	O
DPRS-N.7	Pause (dialling pause)	4.3.4 [15]	O	O
DPRS-N.8	FT initiated virtual call	4.3.4 [15]	O	O
DPRS-N.9	Authentication of PP (GAP N.9)	4.3.4 [15]	M	M
DPRS-N.10	Authentication of user	4.3.4 [15]	O	O
DPRS-N.11	Location registration	4.3.4 [15]	M	O
DPRS-N.12	On air key allocation (GAP N.12) (see note 2)	4.3.4 [15]	M	M
DPRS-N.13	Identification of PP	4.3.4 [15]	O	O
DPRS-N.14	Service class indication/assignment	4.3.4 [15]	O	O
DPRS-N.15	Alerting	4.3.4 [15]	O	O
DPRS-N.16	ZAP	4.3.4 [15]	O	O
DPRS-N.17	Encryption activation FT initiated (GAP N.17)	4.3.4 [15]	M	M
DPRS-N.18	Subscription registration procedure on-air	4.3.4 [15]	M	M
DPRS-N.19	Link control	4.3.4 [15]	M	M
DPRS-N.20	Terminate access rights FT initiated	4.3.4 [15]	M	O
DPRS-N.21	Partial release	4.3.4 [15]	O	O
DPRS-N.22	Go to DTMF (infinite tone length)	4.3.4 [15]	O	O
DPRS-N.23	Go to Pulse	4.3.4 [15]	O	O
DPRS-N.24	Signalling of display characters	4.3.4 [15]	O	O
DPRS-N.25	Display control characters	4.3.4 [15]	O	O
DPRS-N.26	Authentication of FT (GAP N.26)	4.3.4 [15]	O	O
DPRS-N.27	Encryption activation PT initiated (GAP N.27)	4.3.4 [15]	O	O
DPRS-N.28	Encryption deactivation FT initiated (GAP N.28)	4.3.4 [15]	O	O
DPRS-N.29	Encryption deactivation PT initiated (GAP N.29)	4.3.4 [15]	O	O
DPRS-N.30	Calling Line Identification Presentation (CLIP)	4.3.4 [15]	O	O
DPRS-N.31	Internal call	4.3.4 [15]	O	O
DPRS-N.32	Service call	4.3.4 [15]	O	O
DPRS-N.33	Dynamic parameters allocation	4.3.4 [15]	C1001	C1001
DPRS-N.34	Service Negotiation at virtual call setup	4.3.4 [15]	C1002	C1002
DPRS-N.35	In call service change	4.3.4 [15]	C1105	C1105
DPRS-N.36	NWK layer management	4.3.4 [15]	M	M
DPRS-N.37	Identity assignment	4.3.4 [15]	O	O
DPRS-N.38	DECT External handover	4.3.4 [15]	O	O
DPRS-N.39	Message Waiting Indication	4.3.4 [15]	O	O
DPRS-N.40	Detach	4.3.4 [15]	O	O
DPRS-N.41	Periodic location registration	4.3.4 [15]	O	O
DPRS-N.42	On-air modification of user parameters	4.3.4 [15]	O	O
DPRS-N.43	Enhanced security (GAP.N.35)	4.3.4 [15]	M	M
DPRS-N.44	AES/DSAA2 authentication (GAP.N.36)	4.3.4 [15]	C1003	C1003
NGLDS-N.1	General Light Data Service Procedures	5.1.4	M	M
NGLDS-N.2	Software upgrade over the air, C-plane	5.1.4	M	M

C1001: IF (DPRS-ME.2 OR multi-context supported (7.6.1.2.2) OR Generic multiprotocol supported (7.6.1.2.3) OR application packet size different from basic service settings; see ETSI EN 301 649 [15] clauses 12.22 and A.2) THEN M ELSE O.

C1002: IF DPRS-ME.2 THEN M ELSE (IF (LU10 Interworking conventions and HTTP profile for enhanced binary content download (7.6.1.2.2) OR LU10 Interworking conventions and HTTP profile for Generic multiprotocol binary content download (7.6.1.2.3) THEN O ELSE I) (see clause 7.5.3).

C1003: IF MAC service DPRS-M.33 (GAP.M.17) THEN M ELSE O.

C1105: IF DPRS-ME.2 OR multi-context supported (7.6.1.2.2) OR Generic multiprotocol supported (7.6.1.2.3) THEN M ELSE O.

NOTE 1: The reference column refers to the relevant clause in this or in the referenced document.

NOTE 2: This feature is required to be supported in the PT to guarantee the same level of security among all the handsets that operates in a system. The invocation of the feature is however optional to the operator.

6.7.3 NWK features to procedures mapping

The NWK layer feature to procedure mapping specified in ETSI EN 301 649 [15], clause 8.2 with the following changes and additional features shall apply.

Table 11: NWK feature to procedure mapping

Feature/Procedure mapping			Status	
Feature	**Procedure**	**Ref.**	**PT**	**FT**
DPRS-N.1, PT initiated virtual call		4.3.4 [15]	M	M
	PT initiated virtual call request (outgoing call)	7.5.1	M	M
	Overlap sending	8.3 [14]	M	O
	Outgoing call proceeding	8.4 [14]	M	O
	Outgoing call confirmation	8.5 [14]	M	O
	Outgoing call connection	8.6 [14]	M	M
	Sending keypad information	8.10 [14]	O	O
DPRS-N.2, Off Hook		4.3.4	M	M
	PT initiated virtual call request (outgoing call)	7.5.1	M	M
	Incoming call connection	8.15 [14]	M	M
DPRS-N.8, FT initiated virtual call		4.3.4 [15]	O	O
	FT initiated virtual call request (incoming call)	7.5.2	M	M
	Incoming call confirmation	8.13 [14]	M	M
	PT alerting	8.14 [14]	M	M
	Incoming call connection	8.15 [14]	M	M
DPRS-N.11, Location registration		4.3.4 [15]	M	O
	Location registration	8.28 [15]	M	M
	Location update	8.29 [15]	M	O
	Terminal capability indication	7.5.7	M	M
DPRS-N.17, Encryption activation FT initiated (GAP N.17)		4.3.4 [15]	M	M
	Cipher-switching initiated by FT using DSC	8.33 [14]	M	M
	Cipher-switching initiated by FT using DSC2	8.45.10 [14]	M	M
	Storing the Derived Cipher Key (DCK)	8.27 [14]	M	M
	Enforcement of encryption	7.5.4.3	M	M
DPRS-N.18, Subscription registration user procedure on-air		4.3.4 [15]	M	M
	Obtaining access rights	8.30 [15]	M	M
	Terminal capability indication	7.5.7	M	M
DPRS-N.19, Link control		4.3.4 [15]	M	M
	Indirect FT initiated link establishment, for devices supporting complete MAC procedures. Initial setup paging.	12.11.1.1 [15]	C1101	C1101
	Indirect FT initiated link establishment, for devices supporting simplified (A-field) MAC procedures. Initial setup paging.	12.11.2.1 [15]	C1102	C1102
	Fast Paging	12.12 [15]	O	O
	Collective and group ringing	12.13 [15]	O	O
	Direct FT initiated link establishment	12.14 [15]	O	O
	Direct PT initiated link establishment	8.36 [14]	M	M
	Link release "normal"	8.37 [14]	M	M
	Link release "abnormal"	8.38 [14]	M	M
	Link release "maintain"	8.39 [14]	I	I
	Indirect FT initiated link establishment, for devices supporting complete MAC procedures. LCE Resume Paging	12.11.1.2 [15]	C1101	C1103
	Indirect FT initiated link establishment, for devices supporting simplified (A-field) MAC procedures. LCE Resume Paging	12.11.2.2 [15]	C1104	C1104
DPRS-N.24, Signalling of display characters		4.3.4	O	O
	Display	8.16 [15]	M	M

Feature/Procedure mapping			Status	
Feature	**Procedure**	**Ref.**	**PT**	**FT**
	Terminal capability indication	7.5.7	M	M
DPRS-N.25, Display control characters		4.3.4 [15]	O	O
	Display	8.16 [15]	M	M
	Terminal capability indication	7.5.7	M	M
DPRS-N.33, Dynamic parameters allocation		4.3.4 [15]	C1001	C1001
	Dynamic parameters allocation	12.8 [15]	M	M
DPRS-N.34, Service Negotiation at virtual call setup		4.3.4 [15]	C1002	C1002
	Call Resources/Parameters negotiation	7.5.8	M	M
	Service Negotiation specific rules	7.5.3	M	M
DPRS-N.35, In call service change		4.3.4 [15]	C1105	C1105
	Service change - Bandwidth Change	12.6.1 [15]	O	O
	Slot type change	12.6.2 [15]	O	O
	MAC Service change	12.6.3 [15]	O	O
	Modulation type or adaptive codec rate change	12.6.4 [15]	O	O
	DPRS Management Entity Class and other Call-attributes change	12.6.5 [15]	O	O
	MAC Packet lifetime, DLC Window size, DLC Transit delay and C_F channel attributes change	12.6.6 [15]	O	O
	IWU-attributes change - General	12.7.1 [15], 7.5.9	M	M
	Interworking type change	12.7.2 [15]	I	I
	IP address change (IP IWU)	12.7.3 [15]	I	I
	Maximum SDU size change	12.7.4 [15]	O	O
DPRS-N.36, NWK layer management		4.3.4 [15]	M	M
	Management of MM procedures	12.18 [15]	M	M
	Management - Location registration initiation	13.2 [14]	M	C1107
	Management - Assigned individual TPUI	13.3 [14]	M	C1107
	Management - PMID	12.19 [15]	M	M
	Management - DCK	13.6 [14]	M	M
	Management - Broadcast attributes	7.5.10, 12.17 [14]	M	M
	Management - Storage of subscription related data	13.7 [14]	M	M
	U-plane handling	12.17 [15]	M	M
	Length of NWK layer messages	12.20 [15]	C1108	C1108
	Identities	12.21 [15]	M	M
			[15]	
NGLDS-N.1 General Light Data Service Procedures		5.1.4	M	M
	Service change rejection	7.5.4.1	M	M
	Interactions with telephony service	7.5.4.2	M	M
NGLDS-N.2 Software upgrade over the air, C-plane		5.1.4	M	M
	Information exchange in the C-Plane	7.5.5	M	M
	SUOTA push mode	7.5.6	O	O
	Enforcement of encryption	7.5.4.3	M	M

Feature/Procedure mapping			Status	
Feature	Procedure	Ref.	PT	FT
C1001: IF (DPRS-ME.2 OR multi-context supported (7.6.1.2.2) OR Generic multiprotocol supported (7.6.1.2.3) OR application packet size different from basic service settings; see ETSI EN 301 649 [15] clauses 12.22 and A.2) THEN M ELSE O.				
C1002: IF DPRS-ME.2 THEN M ELSE (IF (LU10 Interworking conventions and HTTP profile for enhanced binary content download (7.6.1.2.2) OR LU10 Interworking conventions and HTTP profile for Generic multiprotocol binary content download (7.6.1.2.3) THEN O ELSE I) (see clause 7.5.3).				
C1101: IF DPRS-M.5 THEN M ELSE I.				
C1102: IF DPRS-M.30 THEN (IF DPRS-N.8 OR DPRS-ME.1 THEN M ELSE O) ELSE I.				
C1103: IF DPRS-M.5 THEN (IF single cluster system THEN O ELSE M) ELSE I.				
C1104: IF DPRS-M.30 AND DPRS ME.3 THEN M ELSE I.				
C1105: IF DPRS-ME.2 OR multi-context supported (7.6.1.2.2) OR Generic multiprotocol supported (7.6.1.2.3) THEN M ELSE O.				
C1107: IF DPRS-N.11 THEN M ELSE I.				
C1108: IF (DPRS-N.34 OR Generic multiprotocol supported (7.6.1.2.3)) THEN M ELSE O.				
NOTE: The reference column refers to the relevant clause in the present document, except where stated otherwise.				

6.8 Application features

6.8.1 Application features

New Generation DECT, part 4 devices shall support the following application features.

Table 12: Application features status

Feature supported			Status	
Item no.	Name of feature	Reference	PT	FT
DPRS-A.1	AC_bitstring_mapping	4.3.5 [15]	M	M
DPRS-A.2	Multiple subscription registration	4.3.5 [15]	O	N/A
DPRS-A.3	Manual entry of the PARK	4.3.5 [15]	O	N/A
NGLDS-A.1	Binary content download	5.1.5	M	M
NGLDS-A.2	Software upgrade over the air	5.1.5	M	M
NGLDS-A.3	HTTP based applications	5.1.5	O	M
NOTE: The reference column refers to the relevant clause in the present or in the referenced document.				

6.8.2 Application features to procedures mapping

The Application feature to procedure mapping specified in ETSI EN 301 649 [15], clause 8.4 with the following additional features shall apply.

Table 13: Application feature to procedure mapping

Feature/Procedure mapping			Status	
Feature	Procedure	Ref. (note 1)	PT	FT
NGLDS-A.1 Binary content download		5.1.5	M	M
	General Light Data Services [NGLDS-N.1]	5.1.4	M	M
	Binary content download general requirements	7.6.1.1	M	M
	LU10 Interworking conventions and HTTP profile for simple binary content download	7.6.1.2.1	M	M
	LU10 Interworking conventions and HTTP profile for multi-context binary content download	7.6.1.2.2	O	O
	LU10 Interworking conventions and HTTP profile for Generic multiprotocol binary content download	7.6.1.2.3	O	O
	Binary content download media type	7.6.1.3	M	M
	Binary content download sequence	7.6.1.4	M	M

Feature/Procedure mapping			Status	
Feature	**Procedure**	**Ref. (note 1)**	**PT**	**FT**
	URI-based PP to FP confidentiality requirement	7.6.1.5.1	O	O
	URI-based PP to FP authentication requirement	7.6.1.5.2	O	O
	PP to FP enhanced interactivity	7.6.1.6	O	M
	Common HTTP profile	A.1	M	M
NGLDS-A.2 Software upgrade over the air		5.1.5	M	M
	Binary content download [NGLDS-A.1]	5.1.5	M (note 2)	M
	Software upgrade over the air, C-plane [NGLDS-N.2]	5.1.4	M	M
	Software upgrade over the air general requirements	7.6.2.1	M	M
	Basic SUOTA protocol steps	7.6.2.2	M	M
	Enhanced SUOTA protocol steps	7.6.2.3	N/A	O
	PP security requirements in URL1 and URL2	7.6.2.4	O	O
	Final notification of success and multiple upgrade SUOTA	7.6.2.5	M	M
	Notification of failure	7.6.2.6	M	M
	User initiated SUOTA flag	7.6.2.7	M (note 5)	M
	SUOTA interface to the management server	B	N/A	M
NGLDS-A.3 HTTP based applications		5.1.5	O	M
	Binary content download [NGLDS-A.1]	5.1.5	M	M
	HTTP based applications general requirements	7.6.3.1	M	M
	Support of additional HTTP header fields	7.6.3.2	M	M
	Support of additional media-types	7.6.3.3	M	M
	Support of character encodings	7.6.3.4	M	M
	Simple XHTML profile	7.6.3.5	M	M
	Baseline XHTML profile	7.6.3.6	O	M
	Extended HTTP profile (notes 3 and 4)	A.2	C1301	M

C1301: IF Baseline XHTML profile supported THEN M ELSE O.

NOTE 1: The reference column refers to the relevant clause in the present document.

NOTE 2: When used for NGLDS-A.2, Binary content download shall only use the "Simple" profile of clause 7.6.1.2.1 with Class 4 DPRS management; switching over to the "Multi-context" profile of clause 7.6.1.2.2 shall not be used (although allowed in general by 7.6.1.2.1 when DPRS-N.35 is implemented). Switching over to Class 3 DPRS management through the use of DPRS-N.35 is also not allowed.

NOTE 3: Support of this profile implies support of the "Common HTTP profile" (clause A.1). Support of the "Common HTTP profile" is also implied for PT and FT by the support of Binary content download.

NOTE 4: As described in DPRS [15]/B.8.3.2, the "Extended HTTP profile" (alias protocol DPRS [15]/B.8.3.5 with protocol indicator 1077) may be used by both parties even if protocol indicator 1079 (referring to DPRS [15]/B.8.3.4, Common HTTP profile) is indicated at connection setup. This is the case at least when basic services are used. When this occurs, the FT is informed of PT support for B.8.3.5 through the HTTP User-Agent Header (see clauses 7.6.3.2, A.1.4 and A.2.3).

NOTE 5: The status refers to the setting of the flag in the 'Handset version indication' command. The status does not refer to the support by the PT of 'User initiated SUOTA' (which is handled in clause 7.6.2.2.1).

6.9 Distributed communications

The distributed communication mode (PP-PP communication) is not part of the present document.

Table 14: Distributed communication requirements

Feature supported			Status		
Feature	**Name of feature**	**Ref.**	**PT**	**FT**	HyP
DPRS-DC.1	Distributed Communication	4.3.6 [15]	I	I	I
NOTE: The reference column refers to the relevant clause in the referenced document.					

6.10 Management Entity (ME)

6.10.1 Management Entity (ME) operation modes

In regard to the New Generation DECT, part 4equipment, the following ME operation modes from ETSI EN 301 649 [15], clause 9.1 shall apply.

Table 15: Management Entity Requirements

Feature supported			Status	
Feature	**Name of feature**	**Ref.**	**PT**	**FT**
DPRS-ME.1	Class 1 management	4.3.7 [15]	I	I
DPRS-ME.2	Class 2 management	4.3.7 [15]	O	O
DPRS-ME.3	Class 3 management	4.3.7 [15]	O	O
DPRS-ME.4	Class 4 management	4.3.7 [15]	M	M
NOTE: The reference column refers to the relevant clause in the referenced document.				

6.10.2 Management Entity (ME) mode to procedures mapping

In regard to the New Generation DECT, part 4 equipment, the operation mode to procedure mapping specified in ETSI EN 301 649 [15], clause 9.1.2 shall apply.

7 Profile specific procedures description

7.1 General

This clause identifies differences and additions to the feature/service/procedure definitions and descriptions as specified in ETSI EN 301 649 [15], DPRS.

7.2 Management Entity (ME) procedures

No differences/additions - the procedures as specified in ETSI EN 301 649 [15], clauses 9 and A.1 shall apply.

7.3 MAC layer procedures

No differences/additions - the procedures as specified in ETSI EN 301 649 [15], clause 10 shall apply.

7.4 DLC layer procedures

No differences/additions - the procedures as specified in ETSI EN 301 649 [15], clause 11 shall apply.

7.5 NWK layer procedures

The procedures as specified in ETSI EN 301 649 [15], clause 12 shall apply with the modifications and additional procedures listed in the present clause.

7.5.1 PT initiated virtual call request (outgoing call)

The following text together with the associated clauses defines the mandatory requirements with regard to the present document:

The procedure shall be performed as defined in ETSI EN 300 444 [14], clause 8.2 with the following specific provisions:

At least a limited support of DPRS-N.34 (Service negotiation) is required. More specifically:

- All information elements described in the procedures associated to this feature (i.e. << IWU ATTRIBUTES >>, << CALL ATTRIBUTES >>, << CONNECTION ATTRIBUTES >>, << TRANSIT DELAY >> and << WINDOW SIZE >>) may be used.
- The << IWU ATTRIBUTES >> IE shall always be transmitted, but for some of the profiles defined in the present document, some restrictions shall apply to the allowed field values (see clause 7.6.1.2).
- The << BASIC SERVICE >> IE shall always be transmitted (as required by EN 300 175-5 [5]), with the following restrictions:
  - the IE <<Basic Service (basic service = other)>> may be used, indicating that the service negotiation only relies on the IEs << IWU ATTRIBUTES >>, << CALL ATTRIBUTES >>, << CONNECTION ATTRIBUTES >>, << TRANSIT DELAY >> and << WINDOW SIZE >>.
    - This applies in particular to the "Generic multiprotocol binary content download" profile defined in clause 7.6.1.2.3, when DPRS Class 2 management is used.
    - Use of basic service 'other' with Class 3 or Class 4 implies presence of all 5 IEs, as no default IE is defined in this case.
  - Alternatively, the IE <<Basic Service (basic service = light data service with ME Class 4)>> or the IE <<Basic Service (basic service = light data service with ME Class 3)>> may be used, defining a default setting for all parameters not transmitted in the IEs.
    - This applies in particular to the "Simple binary content download" profile (clause 7.6.1.2.1) and the "Multi-context binary content download" profile (clause 7.6.1.2.2). For these two profiles however, use of IE <<IWU-ATTRIBUTES>> is mandatory, but the transmission of the other 4 IEs is forbidden (only default values apply for them)

NOTE 1: Any parameter transmitted in any of the IEs supersedes the default setting done by the "Basic service".

In any case (full or limited DPRS-N.34 support), the parameters of the call may be changed after setup using the feature DPRS-N.35, if supported.

NOTE 2: If neither feature DPRS-N.34 nor DPRS-N.35 are supported, then only calls with parameter setting equal to the basic service may be set up.

Table 16: Values used within the {CC-SETUP} message

Information element	Field within the information element	Standard values within the field/information element	Normative action/comment
<< Basic service >>			
	< Basic service >	'1001'B	Light data service default setup attributes with ME Class 4 (see note 2)
	< Basic service >	'1010'B	Light data service default setup attributes with ME Class 3 (see note 2)
	< Basic service >	'1111'B	Other
NOTE 1: For the additional information elements needed to identify the required service/parameters see DPRS-N.34 Service Negotiation feature. NOTE 2: Default light data service setup attributes are described in clause A.2 of DPRS (ETSI EN 301 649 [15]).			

7.5.2 FT initiated virtual call request (incoming call)

The following text together with the associated clauses defines the mandatory requirements with regard to the present document.

The procedure shall be performed as defined in ETSI EN 300 444 [14], clause 8.12 with the following specific provisions.

At least a limited support of DPRS-N.34 (Service negotiation) is required. More specifically:

- all information elements described in the procedures associated to this feature (i.e. << IWU ATTRIBUTES >>, << CALL ATTRIBUTES >>, << CONNECTION ATTRIBUTES >>, << TRANSIT DELAY >> and << WINDOW SIZE >>) may be used.
- The << IWU ATTRIBUTES >> IE shall always be transmitted, but for some of the profiles defined in the present document, some restrictions shall apply to the allowed field values (see clause 7.6.1.2).
- The << BASIC SERVICE >> IE shall always be transmitted (as required by ETSI EN 300 175-5 [5]), with the following restrictions:
  - the IE <<Basic Service (basic service = other)>> may be used, indicating that the service negotiation only relies on the IEs << IWU ATTRIBUTES >>, << CALL ATTRIBUTES >>, << CONNECTION ATTRIBUTES >>, << TRANSIT DELAY >> and << WINDOW SIZE >>.
    - This applies in particular to the "Generic multiprotocol binary content download" profile defined in clause 7.6.1.2.3, when DPRS Class 2 management is used.
    - Use of basic service 'other' with Class 3 or Class 4 implies presence of all 5 IEs, as no default IE is defined in this case.
  - Alternatively, the IE <<Basic Service (basic service = light data service with ME Class 4)>> or the IE <<Basic Service (basic service = light data service with ME Class 3)>> may be used, defining a default setting for all parameters not transmitted in the IEs.
    - This applies in particular to the "Simple binary content download" profile (clause 7.6.1.2.1) and the "Multi-context binary content download" profile (clause 7.6.1.2.2). For these two profiles however, use of IE <<IWU-ATTRIBUTES>> is mandatory, but the transmission of the other 4 IEs is forbidden (only default values apply for them)

NOTE 1: Any parameter transmitted in any of the IEs supersedes the default setting done by the "Basic service".

In any case (full or limited DPRS-N.34 support), the parameters of the call may be changed after setup using the feature DPRS-N.35, if supported.

NOTE 2: If neither feature DPRS-N.34 nor DPRS-N.35 are supported, then only calls with parameter setting equal to the basic service may be set up.

Table 17: Values used within the {CC-SETUP} message

Information element	Field within the information element	Standard values within the field/information element	Normative action/comment
<< Basic service >>			
	< Basic service >	'1001'B	Light data service default setup attributes with ME Class 4 (see note 2)
	< Basic service >	'1010'B	Light data service default setup attributes with ME Class 3 (see note 2)
	< Basic service >	'1111'B	Other
NOTE 1: For the additional information elements needed to identify the required service/parameters see DPRS-N.34 Service Negotiation feature. NOTE 2: Default light data service setup attributes are described in clause A.2 of DPRS (ETSI EN 301 649 [15]).			

7.5.3 Service Negotiation specific rules

The additional Information elements described as part of feature DPRS-N.34:

- shall be used in calls using Management Class 2 (DPRS-ME.2). In that case:
  - the <<IWU ATTRIBUTES>> IE shall always be transmitted over the air at connection setup;
  - the other 4 IEs may or may not be present (if absent, the connection relies on the corresponding default IE for Class 2).

NOTE 1: One should not confuse the IE *use status* and the IE *transmission status* (relevant when IE is used). For instance the <<CALL ATTRIBUTE>> IE is mandatory in DPRS (use status=M) but may not be transmitted for Class 2 (transmission status=O). If it is not transmitted, the default IE value for the DPRS management class used for the connection applies.

- and shall also be used in Calls using DPRS-ME.4 orDPRS-ME.3. However in that case:
  - the <<IWU-ATTRIBUTES>> IE shall always be transmitted;
  - the other 4 IEs:
    - may or may not be transmitted in general; this especially applies if profile "Generic multiprotocol binary content download" profile defined in clause 7.6.1.2.3 is used;
    - shall never be transmitted if either the "Simple binary content download" profile (clause 7.6.1.2.1) or the "Multi-context binary content download" profile (clause 7.6.1.2.2) is used.

Service change. The parameters of the call may be changed after initial set up using Service Change (DPRS-N.35) if supported.

NOTE 2: If feature DPRS-N.35 is not supported and only a limited support of DPRS-N-34 is implemented (as requested by the simple or multi-context profiles), only calls with parameter settings equal to the basic service may exist.

NOTE 3: The above provision is intended for implementations supporting DPRS-N.34 due to the support of DPRS-ME.2, or any of the optional binary content download procedures, when initiating a call with a peer that does not support DPRS-N.34, in order to avoid the rejection of the setup due to the lack of support of DPRS-N.34 by the other side.

7.5.4 General procedures

This clause lists specific general procedures applicable to all services provided by this profile.

7.5.4.1 Service change rejection

Any implementation not supporting the feature DPRS-N.35 (Service change) shall at least be able to reject an incoming {CC-SERVICE-CHANGE} message with a {CC-SERVICE-REJECT}, even if the implementation does not understand completely the message because it does not support the handling of long NWK layer messages (ETSI EN 301 649 [15], clause 12.20).

7.5.4.2 Interactions with telephony service

This clause describes the applicable procedure to solve the possible interactions in devices (PP or FP) implementing the present document and also a voice telephony service as described in ETSI EN 300 444 (GAP) [14], ETSI TS 102 527-1 [16] or ETSI TS 102 527-3 [17].

The procedures described in the present clause determine the basic rules for handling the interaction. The following subprocedures are provided and shall be supported by all devices implementing the present document plus any voice telephony service (either ETSI EN 300 444 (GAP) [14], ETSI TS 102 527-1 [16], ETSI TS 102 527-3 [17] or ETSI TS 102 527-5 [i.15]):

- Switching procedure when a light data service call is already established and there is an incoming voice call.
- Using a light data service when a voice call is already established.
- Handling of other interactions.

The procedures for simultaneous handling of light data service and voice calls require further study and will be added in future revisions of the present document.

7.5.4.2.1 Switching procedure when a light data service call is already established and there is an incoming voice call

This clause describes the behaviour of systems which do not support simultaneously data and voice calls but that may however switch from data to voice calls, when an incoming voice call happens during a light data service call. It provides the mechanisms for notification of a possible incoming voice call when a data call is in progress, and for possible switching between the calls.

Setting the "Support of simultaneous DPRS and voice calls" flag in <<SETUP CAPABILITY>> IE

The PP or FP sending the << SETUP-CAPABILITY >> IE, shall set bits 4, 5 of octet 4 ("Support of simultaneous DPRS and voice calls") correctly.

NOTE 1: A PP or FP may use a default << SETUP-CAPABILITY>> IE instead of sending one, i.e. using the default IE mechanism of the DPRS-N.33 feature ("Dynamic parameters allocation"). Such a default IE is defined when a basic service is used for the connection (see ETSI EN 301 649 (DPRS) [15], clause A.2).

All devices implementing the present document plus any voice telephony service (either ETSI EN 300 444 (GAP) [14], ETSI TS 102 527-1 [16], ETSI TS 102 527-3 [17] or ETSI TS 102 527-5 [i.15]) shall at least support "Simultaneous DPRS and voice calls not supported, however switching procedure supported".

If PP or FP supports more enhanced behaviours (simultaneous voice and data calls active or not):

- This entity shall align its behaviour to the remote entity capability if it is lower.
- Those more enhanced behaviours are not defined in detail in the current version of the present document.

Incoming voice call during preliminary exchanges of the SUOTA process

If an incoming internal or external call occurs while the PP has started a SUOTA process:

- either PP and FP are exchanging C-plane <<IWU-WU>> commands;
- or PP is upgrading its memory due to the firmware upgrade.

The FP shall present the incoming voice call as a standard incoming call to the PP. The PP may process or ignore this incoming call.

NOTE 2: Call might not even be presented to the user on PP side. However this should be carefully handled by the PP as this is considered as a temporary interruption of the telephony service for the user.

Incoming voice call while light data service call is already established

This clause describes how to process an incoming internal or external voice call when a data connection is already established. The data call may be related to any of the light data services.

When receiving an incoming internal or external voice call during an ongoing light data service call, the following procedures of ETSI TS 102 527-3 [17] shall be used with the modifications stated in the present document:

- "Call waiting indication (external or internal)", see ETSI TS 102 527-3 [17], clause 7.4.3.5.2.
- "Call waiting rejection", see ETSI TS 102 527-3 [17], clause 7.4.3.5.7.
- "CLIP on call waiting" see ETSI TS 102 527-3 [17], clause 7.4.3.5.10.
- "CNIP on call waiting", see ETSI TS 102 527-3 [17], clause 7.4.3.5.11.

For the PP involved in the light data service, the FP shall indicate the incoming voice call using the "Call waiting indication (external or internal)" procedure of ETSI TS 102 527-3 [17], additionally the FP shall use the "CLIP on call waiting" as well as "CNIP on call waiting" procedures of ETSI TS 102 527-3 [17].

When receiving this call waiting indication, the PP shall decide between any of the following options:

- Either continue with the light data service call (and ignore the call waiting).

EXAMPLE: This may be the case when the PP is involved in a critical step of a Software update procedure.

- Or reject the incoming call using the "Call waiting rejection (from PP to FP)" procedure of ETSI TS 102 527-3 [17].
- Or switch from the data call to the voice call. For this, the PP shall release it using a {CC-RELEASE} message. The FP shall answer with a {CC-RELEASE-COM} message. If the voice call is still waiting, the FP shall re-present the incoming call by sending a {CC-SETUP} message. See figure 5 below for details of data service to voice call switching procedure.
- The decision taken by the PP shall be based on the type of light data service, the state of the process, its own implementation capabilities, and may involve, or not, interaction with the end-user.

NOTE 3: If the PP is performing a Software Upgrade over the air, the PP is responsible for ensuring that the decision made does not compromise its vital functions.

In the meantime, if the remote party desists from the call, or if another PP accepts the call, the FP shall use the "call release and call release rejection" procedure of ETSI TS 102 527-3 [17] to inform the PP that the incoming voice call is no longer relevant (use of call status CS idle). The PP shall, in this case, not continue with the switching procedure. See figure 6 for details.

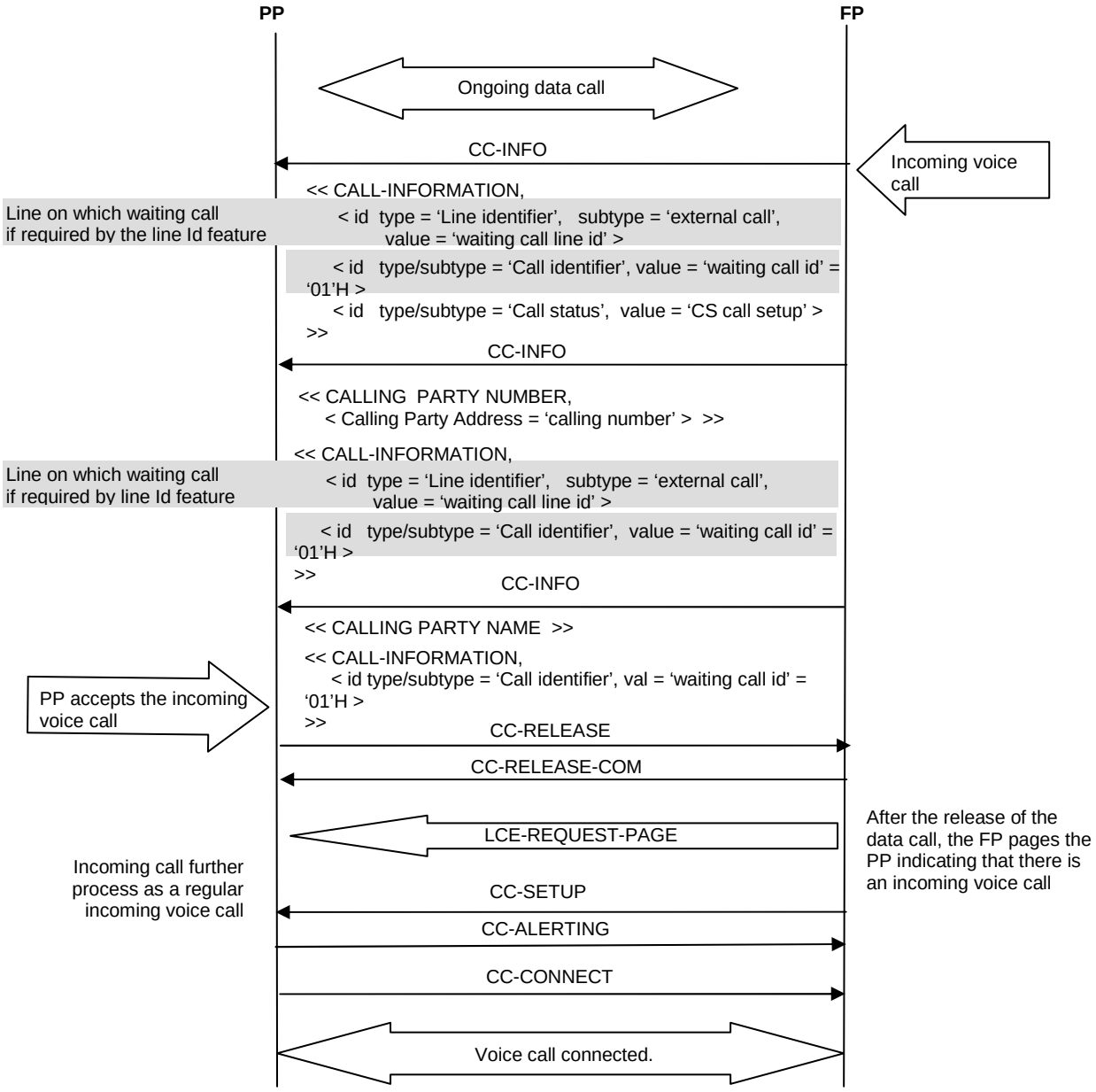


Figure 5: Incoming voice call accepted by the user during established data call

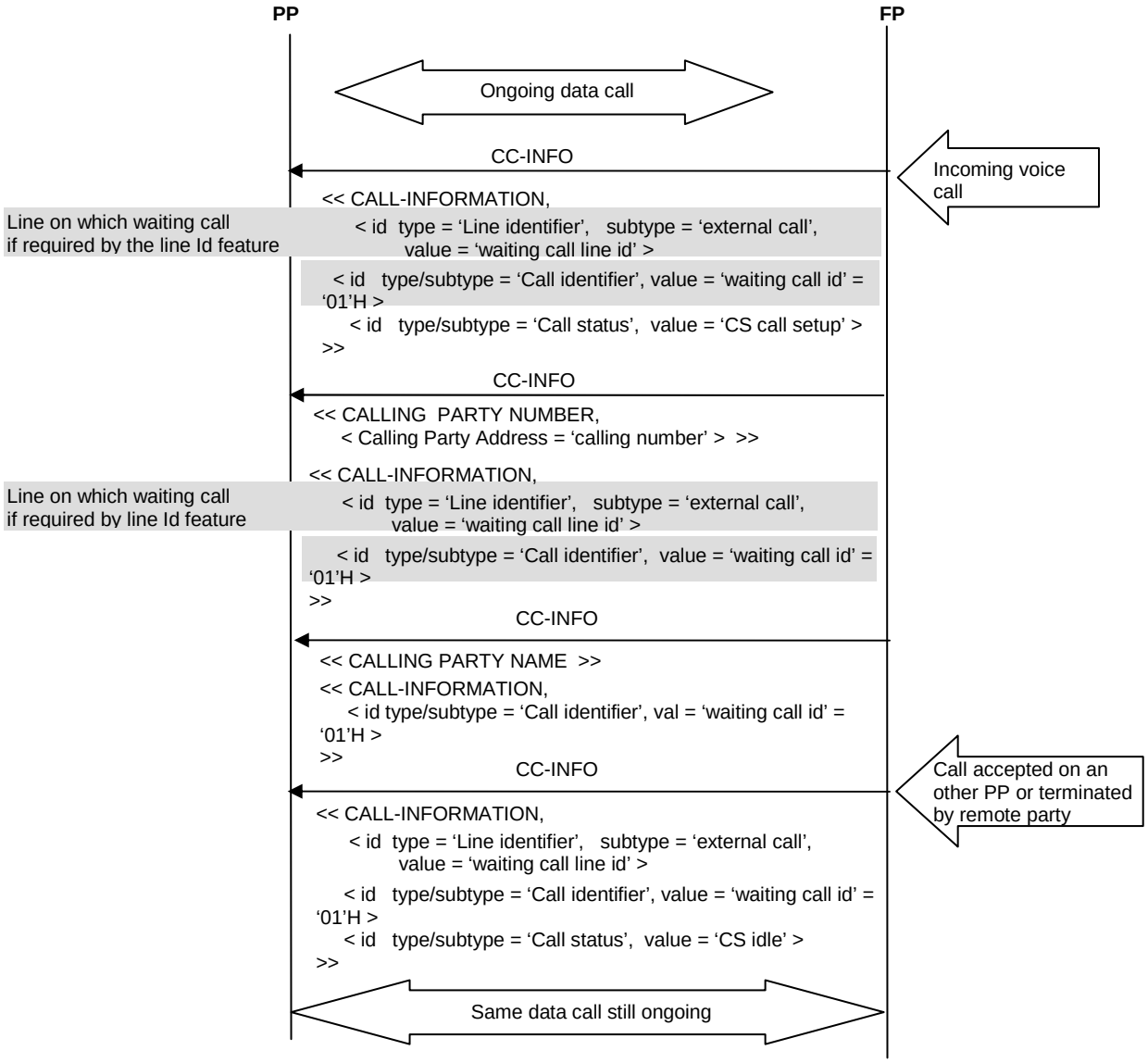


Figure 6: Incoming voice call during a data call but accepted on another PP or terminated by remote party

Behaviour when there are several PPs in the cell

If there are further PPs in the FP cell, not involved in a data call as described in the present document (either because they do not implement it, or because there is no active call), the regular procedures described in ETSI EN 300 444 [14] (GAP), ETSI TS 102 527-1 [16] or ETSI TS 102 527-3 [17] (as supported by the PP) shall be used with these PPs with no modifications.

If the FP is configured to page all, or several, PPs, when an incoming call arrives, the FP shall use the procedure described in the present clause for notifying the call to the PP (or PPs) that is(are) involved in a data call, and *simultaneously* shall use the standard LCE paging procedure (according to the voice profile supported) to notify all other PPs of the incoming call. The paging procedure for other PPs shall not be delayed waiting for the completion or response of the PP(s) which is(are) involved in the data call.

7.5.4.2.2 Simultaneous handling of light data service and voice calls

More advanced terminals (PP or FP) may implement the following procedures:

- Simultaneous handling of data and voice calls, but with only one active call at a time (the data call is suspended when the voice call is in progress).
- Systems supporting full simultaneous voice and data active calls (with simultaneous active bearers over the air interface).

The procedures for simultaneous handling of voice and data calls are for further study and will be added in future revisions of the present document.

7.5.4.2.3 Using a light data service when a voice call is already established

Initiating a light data service during established voice calls

When a voice call (internal or external) is already established, the PP and FP may exchange C-plane messages related to the SUOTA service.

If the FP does not support simultaneous voice and data calls (indicated in the << SETUP-CAPABILITY >> information element), the PP shall not start processing any data call, as there is no guarantee that the FP will support the voice call in parallel with the virtual call. In other words, the PP should always wait until the end of the voice call before establishing a virtual call intended for any light data service.

The procedure when the FP supports more enhanced behaviours (simultaneous voice and data calls), is left for future revisions of the present document.

SUOTA push mode notifications during a voice call

A FP implementing the SUOTA push mode procedure (see clause 7.5.6) may use an established voice call to send the generic event notification relating to the software upgrade indication.

7.5.4.2.4 Handling of other interactions

Generic event notifications for telephony features while data service call is already established

For devices supporting NG-DECT part 3 (ETSI TS 102 527-3 [17]), Generic event notifications may be sent by using any established call or using the CLSS procedure. The FP may use the established data call for sending the notifications.

NOTE: If the data call is used for SUOTA and the PP is in upgrade process, the notification on PP side might be ignored.

Interactions with the "Call identification" feature ([NG1.N.13] of ETSI TS 102 527-3 [17])

If the "Call identification" feature ([NG1.N.13] of ETSI TS 102 527-3 [17]) is implemented on FP side, call identification is intended for voice calls only. More specifically no call identification shall be assigned by the FP at light data service setup.

7.5.4.3 Enforcement of encryption

7.5.4.3.1 Encryption of NG-DECT part 4 data calls

Use of encryption is mandatory in NG-DECT part 4 data transfer. Therefore all data calls shall be encrypted. The FT shall initiate the encryption as described in DPRS-N.17 feature.

7.5.4.3.2 Encryption of NG-DECT part 4 information exchange over C-plane

Additionally the transfer of any C-plane commands related to SUOTA should be encrypted, even if they are transported over an existing voice call, data call, service call or over CLSS.

- If the sending of the C-plane command is done by re-using an active link (data, voice or service call), the FT should initiate the encryption of the link for this call before sending the command.
- If the sending of the C-plane command is done using the CLSS procedure, the FT should start the encryption before the sending of the first {FACILITY} message.

NOTE: In terms of implementation, this may imply that any call has to be systematically encrypted or that the application waits until the end of a call and uses the CLSS procedure if the call was not originally encrypted. This may also imply that the FP starts encryption each time the CLSS procedure is used.

7.5.5 Information exchange in the C-Plane

Software upgrade uses C-plane message sequences and message formats to allow FT and PT to exchange information about software version and availability.

Information exchange in the C-Plane is based on the Information Element <<IWU to IWU>>. This is used in particular for the exchange of version information.

7.5.5.1 C-Plane commands general format

The C-plane commands are based on CISS {FACILITY} messages, which contain the Information Element <<IWU to IWU>>, using the dedicated protocol discriminator '06'H.

For the purpose of transmitting the {FACILITY} message containing the << IWU to IWU >> information element to the peer entity, the DECT entity shall either:

- use an already established link used by any Connection Oriented service (such as a voice, data or service call), if existing; or
- if there is no existing call at the time of sending the {FACILITY} message, use the CLSS procedure as defined in clause 10.4.2.3 of ETSI EN 300 175-5 [5]:
  - If the FT is the initiator, the FP shall initiate indirect link establishment as defined in clause 7.3.8 of ETSI TS 102 527-1 [16] "Indirect FT initiated link establishment" procedure. The short format and the full format with IPUI are allowed in the paging messages. The LCE header shall be set to either; the '000'B value (indicating "no U-plane") or to the '100'B value (indicating "General code for voice service").
  - If the PT is the initiator, the PP shall initiate direct link establishment as defined in clause 8.36 of ETSI EN 300 444 [14] "Direct PT initiated link establishment" procedure.
  - In both cases, full slot and long slot (j = 640) are allowed as slot type. The chosen slot type is decided by the initiating party.

Whatever {FACILITY} transport mode is used (re-use of an already established call, or use of the CLSS procedure), the {FACILITY} message shall be coded with the dummy transaction identifier (TI) value 6 and the protocol discriminator (PD) '0100'B for CISS.

NOTE: When the CLSS procedure is used, the <<Portable Identity>> IE is mandatory from PT to FT. When an already established link is used, this IE is optional.

The content of the IWU-to_IWU Information element carried in the {FACILITY} message shall be as described in table 18.

Table 18: General <<IWU to IWU>>-based software upgrade C-plane commands format

Information element	Field within the information element	Standard values within the field/IE	Normative action/comment
<<IWU to IWU>>			
	<Length of content>	L	Length of content (1 octet)
	<S/R bit>	1	Transmission of message
	<Protocol Discriminator>	06H	Software upgrade over the air
	<Command >	0…127	Software upgrade command
	<Command specific byte 0>		
	…		
	<Command specific byte L-2>		

7.5.5.2 Software upgrade commands

The following software upgrade commands are defined:

Bits 8 7 6 5 4 3 2 1	Meaning	PP => FP	FP => PP
0 0 0 0 0 0 0 0	handset version indication	YES	
0 0 0 0 0 0 0 1	handset version available		YES
0 0 0 0 0 0 1 0	URL indication	YES	YES
0 0 0 0 0 0 1 1	negative acknowledgement	YES	YES
All other values reserved			

7.5.5.2.1 "Handset version indication" command

The "Handset version indication" command allows a PP to check availability of a new software image (defined as a set of files, numbered from 1 to N_f, $N_f \leq 15$), and to request the urls targeting all of these files (one for each use of the command).

It is sent to the FP which sends the requested file url as a response, in a "Handset version available" command (see clause 7.5.5.2.2).

Decision to send a "Handset version indication" command shall be taken by the PP according to the provisions of clause 7.6.2.2.1, "Step 1-PP sends a "Handset version indication" command to the FP. The PP shall not queue several "Handset version indication" commands.

The URL following this command (see octet 7), if present, shall contain a valid URL1 value to be used in step 2 (see clause 7.6.2.2.2, "Step 2-FP retrieves url of the next file to be downloaded (FP_URL2)").

Bit:	8	7	6	5	4	3	2	1	Octet:
0	<< IWU-TO-IWU >>								1
Length of Contents (L = LS + LH + 11)									2
1	S/R = 1	**Protocol Discriminator = "Software upgrade over the air" = '06'H**							3
0/1 ext	**Command = "Handset version indication" = '0'H**								4
EMC value high byte									5
EMC value low byte									6
URL1 to follow									7
0	reserved					**fileNumber**			8
0/1 ext	**flags**					**reason**			8a
<SW Version identifier> = 1									9
Length of SW Version identifier (1 ≤ LS ≤ 20)									10
SW Version identifier first octet (IA5 character)									11
…									…
SW Version identifier last octet (IA5 character)									LS + 10
<HW Version identifier> = 2									LS + 11
Length of HW Version identifier (1 ≤ LH ≤ 20)									LS + 12
HW Version identifier first octet (IA5 character)									LS + 13
…									…
HW Version identifier last octet (IA5 character)									LS + LH + 12

NOTE: The extension bits in octets 4 and 8a (bit 8) shall be set to '1'B as they are respectively the last octet of octet group 4 and the last octet of octet group 8 in the current version of the present document.

Figure 7: <<IWU-TO-IWU>> information element for the "Handset version indication" command

Equipment Manufacturer's Code (EMC)

Equipment Manufacturer's Code 16-bit value, as defined in ETSI EN 300 175-6 [6].

URL1 to follow (URL1_to_follow) (octet 7)

Bits	**8 7 6 5 4 3 2 1**	**Meaning**
	0 0 0 0 0 0 0 0	URL does not follow
	0 0 0 0 0 0 0 1	
	to	Number of URL messages following in a "URL indication" command
	1 1 1 1 1 1 1 1	

If present, URL1 value shall follow in one or several "URL indication" commands (see clause 7.5.5.2.3).

File number field (fileNumber = "n") (octet 8)

Interpretation of the fileNumber field depends on the reason field value.

- If the reason field is "0", fileNumber is the ordinal number of the requested file. This implies that files with number between 1 and fileNumber-1 where applied successfully.
- If the reason field is not "0", the "fileNumber" field refers to the file number of the file whose application failed.

Reason field (reason) (octet 8a)

The interpretation of the "Handset version indication" command depends on the reason field value:

- when the "reason" field is "0", the command is a request for a new file, as described in clause 7.6.2.2.1.
- when the "reason" field is non zero, the command is a notification of failure, as described in clause 7.6.2.6.

NOTE 1: The meaning of the "fileNumber" field also depends on the "reason" field value. See "File number field" description.

Bits 4 3 2 1	Meaning
0 0 0 0	Success of previous files application. Request for a new file (indicated by "fileNumber" field)
0 0 0 1	Download of file with indicated fileNumber failed (file not found)
0 0 1 0	Application of file with indicated fileNumber failed (see note 2)
0 0 1 1	Unable to download in time-New DelayMinutes requested (see clause 7.5.5.2.2, DelayMinutes field for details)
0 1 0 0	Download of file with indicated fileNumber failed (server not found; see note 3)

All other values reserved.

NOTE 2: Application of a file refers to the intended use of the file during installation of the new software, and until the next file is requested.

NOTE 3: Value '0001'B (file not found) can be used instead, if the FP is not aware that the server was not found.

Flags (octet 8a)

Bits 7 6 5	Meaning
x x 0	Software upgrade not initiated by the user (see note 4)
x x 1	User initiated software upgrade (see note 4)

Bits 6 and 7 are reserved for further standardization. They shall be set to "0".

NOTE 4: This flag is transmitted to the MS via the FT (at least in Basic SUOTA mode) and may influence (or not) the MS upgrade policy (planning, etc). See 'external triggering event' in clause 7.6.2.2.1 and clause 7.6.2.7.

SW Version identifier (octet 9)

A field identifier value of "1" shall be used in octet 9. PP *current* software version identifier. The length of this parameter shall be of 20 octets maximum. Only IA5 characters shall be used.

The SW version identifier value is software provider specific, and shall be defined by the PP vendor. It shall not be empty. The value used shall represent the currently installed version of the software, not the targeted SW version identifier (although it is known from the first response of the FP on).

The field value shall be set to a value allowing the MS to determine the software identity and version to be downloaded (not only the version, unless the software identity is implicit). Examples of field values:

- SW Version identifier ="FIRMWARE-1.2.0".
- SW Version identifier ="WEATHER-WIDGET-1.2.0".
- SW Version identifier ="1.2.0".

NOTE 5: The FP should not interpret this value and should always rely on the management server response to get the new software version identifier documented in the "Handset version available" command.

NOTE 6: In case the SUOTA feature is used for a software **first** installation (i.e. not for an "upgrade"), a special value of the field should be defined to indicate this (i.e. the initial software version identifier cannot be used as it is not the currently installed version).

HW Version identifier (octet L_H + 11)

A field identifier value of "2" shall be used in octet L_H + 11. PP hardware version identifier. The length of this parameter shall be of 20 octets maximum. Only IA5 characters shall be used.

This value does not change during the whole lifecycle of the PP.

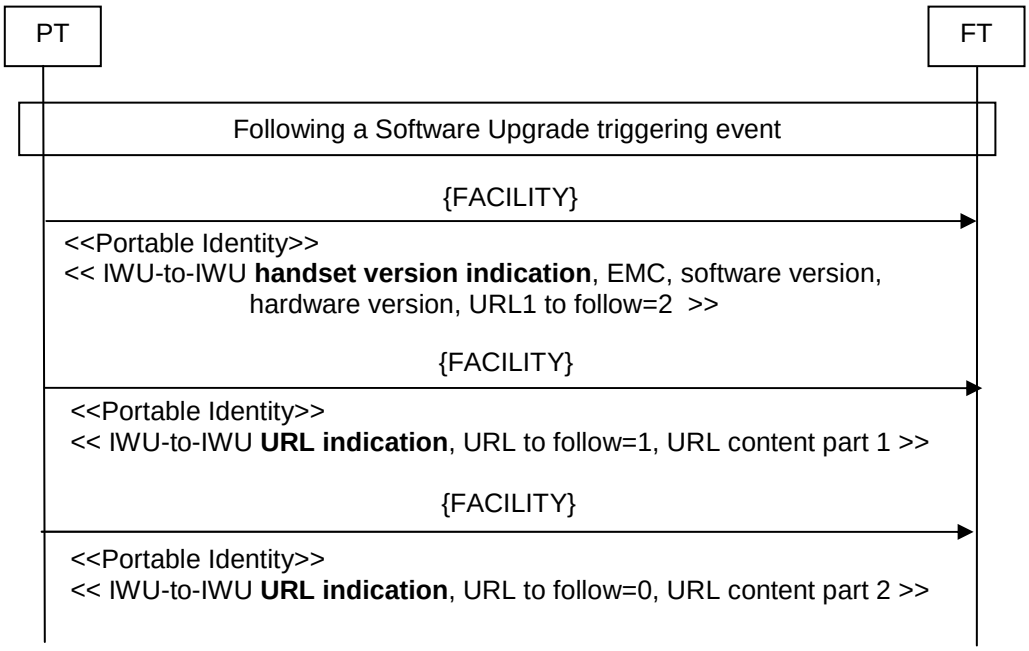


Figure 8: "Handset Version indication" procedure

NOTE 7: "URL indication" command is described in clause 7.5.5.2.3.

NOTE 8: The <<Portable Identity>> IE is mandatory from PT to FT when the {FACILITY} message is sent with the CLSS procedure and optional otherwise.

A handset software version is clearly identified by the parameters EMC, SW Version identifier, and HW Version identifier, which can be used by the management server to assign a new software image.

7.5.5.2.2 "Handset version available" command

Bit:	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**Octet:**
0	<< IWU-TO-IWU >>								**1**
Length of Contents (L = Ls + 8)									**2**
1	S/R = 1	**Protocol Discriminator = "Software upgrade over the air" = '06'H**							**3**
0/1 ext	**Command = "Handset version available" = '1'H**								**4**
DelayMinutes value high byte									**5**
DelayMinutes value low byte									**6**
URL2 to follow									**7**
0/1 ext	**User interaction**				reserved				**8**
<**SW Version identifier**> = 1									**9**
Length of SW Version identifier (0 ≤ Ls ≤ 20)									**10**
SW Version identifier first octet (IA5 character)									**11**
…									**…**
SW Version identifier last octet (IA5 character)									**Ls + 10**

NOTE: The extension bits in octets 4 and 8 (bit 8) shall be set to '1'B as they are respectively the last octet of octet group 4 and the last octet of octet group 8 in the current version of the present document.

Figure 9: <<IWU-TO-IWU>> information element for the "Handset version available" command

Delay for download in minutes (DelayMinutes) (octets 5 and 6)

Delay in minutes, starting from current time (i.e. from reception of the "Handset version available" command), to be respected by the PP until the actual software image download can start.

This delay allows the MS:

- to avoid server overload by distributing downloads over time;
- to minimize user disturbance, by differing downloads until the next night;
- to schedule distinct software upgrades to the same PP (each with its own set of files).

A value of "0" indicates that an immediate download is required by the MS.

A value of "FFFF"H indicates that the delay is undefined (the PP may attempt a download at any time).

First file download: The delay is especially significant for the first file (when fileNumber = "1" in the PP request). The PP shall respect the indicated delay as a lower bound and shall try to immediately download the software after the delay has timed out.

More specifically, the download of the first file shall start as soon as possible after CurrentTimeMinutes + DelayMinutes, where CurrentTimeMinutes refers to the reception time in minutes of the request (from any time origin).

Next files download: For the next files (fileNumber $\geq$ 2), and provided the PP does not ask for a new delay (see below), the FP shall set the DelayMinutes parameter to "0", so that the overall upgrade time will be as short as possible.

Possible request for a new delay: If the PP cannot respect the delay for a file (the first file, or any file), e.g. if the PP was switched off or the server was unreachable when the delay timed out, the behaviour of the PP is left up to the PP vendor. For example:

- the PP may try to download the software anytime afterwards; or
- the PP may respect a build-in download window. More specifically in that case:
  - the PP will not try to download the current file after CurrentTimeMinutes + DelayMinutes + DownloadWindowMinutes, where DownloadWindowMinutes represents the built-in download window length for downloading;
  - additionally, if the download window cannot be respected, the PP asks for a new DelayMinutes value (hence a new download window), using the "reason" field value of "Unable to download in time-New DelayMinutes requested" (see clause 7.5.5.2.1), in order to be able to download the remaining (or all) files.

NOTE 1: In case of firmware upgrade, a PP may request a new download window while keeping some already downloaded files in the following cases:

- because it has sufficient memory to keep them while still being functional;
- or because it can no longer revert to a working version (but only a rescue version only allowing firmware upgrade).

URL2 to follow (URL2_to_follow) (octet 7)

Bits	**8 7 6 5 4 3 2 1**	**Meaning**
	0 0 0 0 0 0 0 0	URL does not follow
	0 0 0 0 0 0 0 1	
	to	Number of URL messages following in a "URL indication" command
	1 1 1 1 1 1 1 1	

If present, URL2 value shall follow in one or several "URL indication" commands (see clause 7.5.5.2.3).

EXAMPLE: For an example with an 'URL2 to follow' value of 0, see clause 7.6.2.2.3, option 1 ("No url").

User interaction (octet 8)

Bits	7 6 5	Meaning
	0 0 0	No user interaction required
	0 0 1	User interaction required
	0 1 0	User interaction undefined

All other values reserved.

This field allows the MS to indicate whether a user interaction is required or not before the actual download of the software can take place. Value '010'B indicates that there is no MS requirement concerning user interaction.

SW Version identifier (octet 9)

A field identifier value of '1' shall be used. Software version identifier of the software to be installed as a result of the upgrade (not the currently installed version). The length of this parameter shall be of 20 octets maximum. Only IA5 characters shall be used.

If a new software version is available, this field shall not be empty (the length shall not be '0') in order to allow future upgrades. An empty "software version identifier" (length = 0) shall be used if there is no new version available (see clause 7.6.2.2.3, option 1).

The SW version identifier value is software provider specific, and shall be agreed with the MS operator.

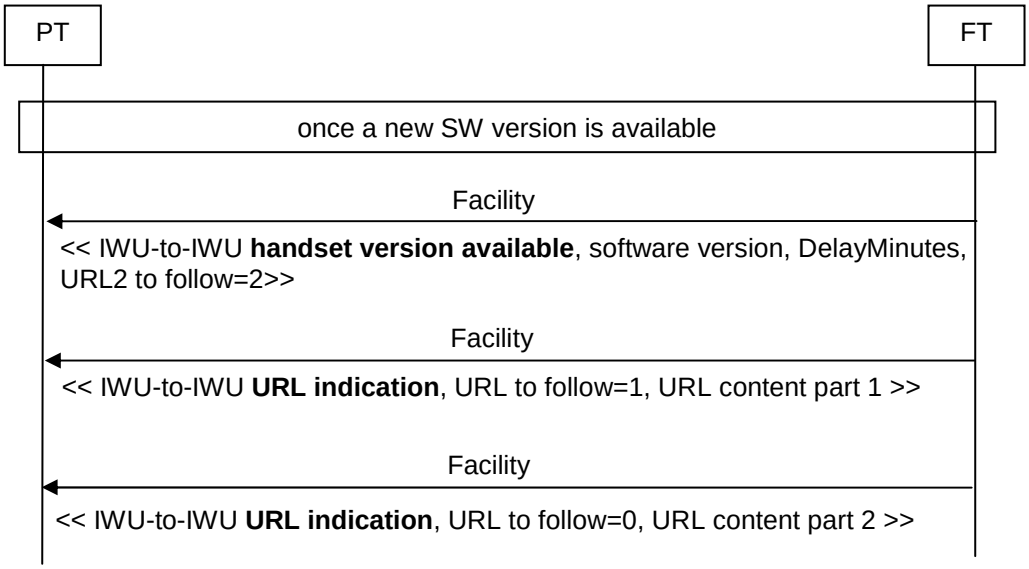


Figure 10: "Handset Version available" procedure

NOTE 2: "URL indication" command is described in clause 7.6.2.4.2.3.

7.5.5.2.3 "URL indication" command

Bit:	8	7	6	5	4	3	2	1	Octet:
0	<< IWU-TO-IWU >>								1
Length of Contents (L = Lu + 8)									2
1	S/R = 1	**Protocol Discriminator = "Software upgrade over the air" = '06'H**							3
0/1 ext	**Command = "URL indication" = '2'H**								4
URL to follow									5
Length of URL content in this IE (Lu)									6
URL content first octet (IA5 character)									7
…									…
URL content last octet (IA5 character)									Lu + 6

Figure 11: <<IWU-TO-IWU>> information element for the "URL indication" command

URL to follow (URL_to_follow) (octet 5)

Bits	**8 7 6 5 4 3 2 1**	**Meaning**
	0 0 0 0 0 0 0 0	This message is the last message and contains the last part of the URL
	0 0 0 0 0 0 0 1	
	to	Number of remaining "URL indication" messages following this message
	1 1 1 1 1 1 1 1	

NOTE: The extension bit of octet 4 is set to 1 as it is the (first and) last octet of octet group 4 (reduced to one octet in the current version of the present document).

URL content (octets 7 to Lu + 6)

The maximum information length of a LAPC UI frame is 63 octets (see ETSI EN 300 175-4 [4]). As a consequence, each "URL indication" command may contain up to 55 octets of URL content, following a 'header' of 8 octets, as shown in table 18a.

Table 18a: Message structure of the "URL indication" command

		Octet:
Transaction Identifier	Protocol Discriminator	1
Message Type = {FACILITY}		2
Information element = <<IWU to IWU>>		3
Length of contents = L (octets)		4
Protocol Discriminator = "Software upgrade over the air" = '06'H		5
Command = "URL indication" = '02'H		6
URL to follow		7
Length of URL content in this IE (Lu)		8
URL content first octet (IA5 character)		9
URL content last octet (IA5 character)		Lu+6

When the CLSS procedure is used, the <<Portable Identity>> IE is mandatory from PT to FT. When an already established link is used, this IE is optional. This IE is 9 octets long.

To summarize, the URL content shall be segmented into several messages:

- From FP to PP, if the URL indication content length exceeds 55 octets
- From PP to FP:
  - If the <<Portable Identity>> IE is used, and the URL indication content length exceeds 46 octets (55-9).

- Or, if the <<Portable Identity>> IE is NOT used, and the URL indication content length exceeds 55 octets.

7.5.5.2.4 "Negative acknowledgement" command

The "negative acknowledgment" command is used by the FP or the PP for notifying errors arising during the use of C-plane software upgrade commands.

NOTE 1: The allowed directions of use depend on the "Reject reason" field value.

It shall not be used to notify errors arising from the use of "Binary content download" feature over the U-plane, when downloading the software image.

Table 19: Values used within <<IWU to IWU>> information element for the "negative acknowledgement" command

Information element	Field within the information element	Standard values within the field/IE	Normative action/comment
<<IWU to IWU>>			
	<**Length of content**>	LH	Length of content
	<**Protocol Discriminator**>	06H	Software upgrade
	<**Command**=negative acknowledgement >	03H	Software upgrade command
	<**Reject reason**>	0...255	Reject Reason

Reject reason

Bits 8 7 6 5 4 3 2 1	Meaning	Direction
0 0 0 0 0 0 0 0	Reserved	both
0 0 0 0 0 0 0 1	Retry later - Connection refused (see note 2)	FP to PP
0 0 0 0 0 0 1 0	Retry later - FP resources overflow (see note 2)	FP to PP
0 0 0 0 0 0 1 1	File does not exist (see clause 7.6.2.2.3, option 3)	FP to PP
0 0 0 0 0 1 0 0	Invalid URL1 format	FP to PP
0 0 0 0 0 1 0 1	Unreachable URL1 (server error)	FP to PP
0 0 0 0 0 1 1 0	Command format error	both

all other values reserved.

NOTE 2: The "Retry later" error code may be used in case the FP is faced with several simultaneous upgrade requests (using the "handset version indication" command) from several PPs, which it cannot handle simultaneously.

Retry later - Connection refused. The "Retry later - Connection refused" value shall be used when the FP is currently involved in a data call with another PP and cannot support the setup of an additional connection with this PP.

NOTE 3: This requirement does not apply when the FP only sent a "Handset version available" command to another PP. It only applies as soon as that other PP did setup a data call as a result of that command for downloading the software.

- The PP receiving a negative acknowledgement with this reject reason shall wait for at least 5 minutes before sending a new "Handset version indication" command.

If the FP does not support two simultaneous data calls, but sent a "Handset version available" command to several PPs (because none of them engaged in a data call yet), then if one of them sets up a data call with the FP, any other PP attempting to setup an additional data call shall immediately receive a {CC-RELEASE-COM} with reject reason "Insufficient resources".

- The PP receiving the {CC-RELEASE-COM} shall initiate a new C-plane exchange before any new attempt to setup a data call for downloading the software.

Retry later - FP resources overflow. The "Retry later - FP resources overflow" value shall be used in case the PP connection was accepted but the remaining hardware resources are not sufficient to handle the upgrade.

7.5.6 SUOTA push mode

A FP implementing the present procedure shall use the "generic event notification" procedure of ETSI TS 102 527-3 [17], clause 7.4.1, to forward software upgrade indications to the PP.

The <<CALL-INFORMATION>> information element shall not be sent as part of this notification.

For indication of values used in <<Events notification>> information element, see table 20.

Table 20: Values used within {FACILITY} message for software upgrade indication

Information element	Field within the information element	Standard values within the field/IE	Normative action/comment
<<Events notification>>	<Event type>	4	Software upgrade indication
	<Event sub type>	0 or 1	0 Unknown 1 Firmware upgrade
	<Event multiplicity >	0	Not used. Shall have a value of "0"

A PP implementing the present procedure should attempt a Software upgrade Over The Air (see clause 7.6.2) when receiving the notification. However, the PP exact behaviour is left up to the PP vendor. More specifically, a "Handset Version Indication" command could be sent anytime afterwards, or not be sent at all.

EXAMPLE 1: A PP could respect a built-in timeout, and send a "Handset Version Indication" command as described in clause 7.6.2.2.1 only if it has the opportunity to do so within this timeout, and ignore the notification otherwise.

EXAMPLE 2: A PP could respect a delay between the sending of two consecutive "Handset Version Indication" commands for security reasons, forcing it not to respect the built-in timeout.

EXAMPLE 3: A PP could ignore notifications transferred from a FP from a different vendor, from a blacklisted FP, etc.

7.5.7 Terminal capability indication

The contents of the <Terminal Capability> information elements shall be based on the requirements of ETSI EN 301 649 [15], clause 12.3.

For the purpose of this ASAP only the status of the fields and specific values implementation that **has changed** is indicated in this clause. For the rest whatever specified in ETSI EN 301 649 [15] shall apply.

Table 21: Values used within the <<TERMINAL CAPABILITY>> information element

Information element	Field within the information element	Standard values within the field/information element	Normative action/comment
<<Terminal capability>>			
	<ext4>	0	
	<Profile indicator_1>	'x 1 x x x x x'B	OUT OF SCOPE (DPRS Stream support)
		'1 x x x x x x'B	OPTIONAL (Asymmetric bearer)
	<ext4a>	0	
	<Profile indicator_2>	'x x x x x x 1'B	OPTIONAL(DPRS Class 2 management and B-field procedures (DPRS-M.5) supported (ETSI EN 301 649 [15])
	<ext4b>	0	
	<Profile indicator_3>	'x 1 x x x x x'B	OUT OF SCOPE (Ethernet support)
		'1 x x x x x x'B	OUT OF SCOPE (Token Ring support)
	<ext4c>	0	
	<Profile indicator_4>	'x x x x x x 1'B	OUT OF SCOPE (IP support)
		'x x x x x 1 x'B	OUT OF SCOPE (PPP support)
		'x x x x 1 x x'B	OUT OF SCOPE (V.24 support)
		'x x x 1 x x x'B	OPTIONAL (C_F supported)
		'x x 1 x x x x'B	OPTIONAL (I_{PQ} services supported)

Information element	Field within the information element	Standard values within the field/information element	Normative action/comment
		'1 x x x x x x'B	MANDATORY (Generic Media Encapsulation supported)
	< ext4d >	0	
	< ext4e >	0	
		'x x 1 x x x x'B	OPTIONAL (Channel G_F supported, see note 2)
		'x 1 x x x x x'B	OPTIONAL (PT with fast hopping radio)
		'1 x x x x x x'B	OPTIONAL (Capability to support "no emission" mode) (see ETSI EN 300 175-3 [3], clause 9.4)
	< ext4f >	0	
	< ext4g >	0	
		'x x x x x x 1'B	OPTIONAL (E+U-type mux and channel I_{PF} basic procedures supported, see note 1)
		'x x x x x 1 x'B	OPTIONAL (Channel I_{PF} advanced procedures supported)
		'x x x x 1 x x'B	OUT OF SCOPE (Channel SI_{PF} supported)
	<Packet data category>	any	OUT OF SCOPE of the present document. System may be categorized due to other data capabilities. If not categorized, code shall be set as 0.
	< ext4h >	1	
		'x x x x x x 1'B	OPTIONAL (DPRS Class 3 management and A-field procedures (DPRS-M.30) supported (see ETSI EN 301 649 [15])
		'x x x x x 1 x'B	MANDATORY (DPRS Class 4 management and A-field procedures (DPRS-M.30) supported (see ETSI EN 301 649 [15])
		'x 1 x x x x x'B	MANDATORY (Support of Light data services (the present document)
NOTE 1: IF DPRS-M.23 THEN MANDATORY ELSE OPTIONAL. NOTE 2: IF DPRS-M.19 THEN MANDATORY ELSE OPTIONAL.			

7.5.8 Call resources/parameters negotiation

The contents of the messages applicable to this procedure shall be based on the requirements of the ETSI EN 301 649 [15], clause 12.5, with the fields and values listed in table 22.

Table 22: Values used within the {CC-SETUP} message

Information element	Field within the information element	Standard values within the field/information element	Normative action/comment
<< IWU attributes >>			
	< Length of Contents >	any	
	< Coding standard >	01	Profile defined coding.
	< Profile >	00000	DPRS Frame Relay support (MANDATORY)
	< Negotiation Indicator >	000, 010	- Negotiation not possible (see note 1) - Peer attribute negotiation
	< Profile Subtype > (octet 4)	1 000	DECT Generic media encapsulation (MANDATORY, Other options Out of scope)
	<Maximum SDU size> PT => FT (octets 5 and 5a)	> 191 (equivalent to 1 528 octets)	At least 1 528 octets (codec as 191) shall be supported (see note 7)

Information element	Field within the information element	Standard values within the field/information element	Normative action/comment
	<Maximum SDU size> FT => PT (octets 5b and 5c)	> 1 528 (equivalent to 12 224 octets)	At least 12 224 octets (codec as 1 528) shall be supported (see note 7)
	< Profile Subtype attributes> (octet 6)	several values and number of octets possible	See DPRS [15], clauses B.2 and B.8
<< Call attributes >>			
	< Coding standard >	00	
	< NWK layer attributes >	00010, 00011, 00110	DPRS Class 2, DPRS Class 3, DPRS Class 4. Only values for implemented ME Classes need to be supported
	< C-plane class >	010	Class A shared is only mandatory, rest are optional and need not be supported by the peer side
	< C-plane routing >	0000, 0100	C_S only; C_F only; Support of C_F is optional
	< ext5 >	1	
	< U-plane symmetry >	00	Symmetric (see note 6)
	< LU identification >	01010	LU10.
	< ext6 >	1	
	< U-plane class >	101	Class 2; SELective
	< U-plane frame type >	1010	FU10a/c mandatory for support
<< Connection attributes >>			Signifies the maximum capabilities of the sender for the requested call. If not included the default value shall be assumed (see clause 12.5.1 of [15])
	< Symmetry >	001 010 101	Symmetric only connection Asymmetric reversible Asymmetric one-way-only Asymmetric types are applicable only when operating in ME Class 2 and B-field signalling (DPRS-M.5)
	< Connection identity >	0000	Not yet numbered
	ext4	0, 1	If 1 is indicated, the octets 4a, 4b and 4c shall not be included and their values shall be understood to be equal to the value set in < Target bearers (P = > F direction) >
	< Maximum bearers (P = > F direction) >	00nnnnn nnnnn = 1 to 23	If "Symmetric" has been indicated max. value that needs to be supported is 12
	ext4a	0, 1	If 1 is indicated, the octets 4b and 4c shall not be included and their values shall be understood to be equal to the values set in octets 4 and 4a respectively
	< Minimum bearers (P = > F direction) >	01nnnnn nnnnn = 0 to 23	
	ext4b	0, 1	If 1 is indicated, the octets 4c shall not be included and its value shall be understood to be equal to the value set in octet 4b
	< Maximum bearers (F = > P direction) >	10nnnnn nnnnn = 1 to 23	
	ext4c	1	
	< Minimum bearers (F = > P direction) >	11nnnnn nnnnn = 0 to 23	
	< ext5 >	0, 1	If 1 is indicated, octet 5a shall not be included and its value shall be understood to be equal to the value set in octet 5

Information element	Field within the information element	Standard values within the field/information element	Normative action/comment
	< MAC slot size >	001 100 101	long slot 640 MANDATORY to support full slot Optional to support double slot Optional to support
	< MAC service P = > F >	0010 0011 0110 0111	I_{PM}; detect: MANDATORY to support I_{PMR}; Mod-2 correct: Optional I_{PQ}; detect: optional I_{PQR}; Mod-2 correct: Optional Support of "I_P; Mod-2 correct" and I_{PQ} is optional
	< ext5a >	1	
	< spare >	000	
	< MAC service F = > P >	0010 0011 0110 0111	I_{PM}; detect: MANDATORY to support I_{PMR}; Mod-2 correct: Optional I_{PQ}; detect: optional I_{PQR}; Mod-2 correct: Optional Support of "I_P; Mod-2 correct" and I_{PQ} is optional
	< Ext6 >	1, 0	If 1 is indicated, octet 6a shall not be included and its value shall be understood to be equal to the value set in octet 6
	< C_F channel attributes P = > F >	000, 010, 011, 100, 101	C_F never (C_S only) C_F priorities A, B, C or D Support of C_F is optional
	< MAC packet life time P = > F >	0 to 7	Values > 0 only for I_P_error_correct
	< Ext6a >	1	
	< C_F channel attributes F = > P >	000, 010, 011, 100, 101	C_F never (C_S only) C_F priorities A, B, C or D Support of C_F is optional
	< MAC packet life time F = > P >	0 to 7	Values > 0 only for I_P_error_correct or I_{PQ}_error_correct
	< Ext7 >	1	See note 5
	< A-attributes >	000	2-level modulation scheme
	< B-attributes >	000, 001, 010	2-level modulation scheme 4-level modulation scheme 8-level modulation scheme The support of 4 and 8 level modulation scheme is optional
<< Transit delay >>			For the default value in case it is not included see clause 12.5.1 of [15]
	< Forward Delay >	0 All	Infinite - Mandatory for support Rest - optional
	< Backward Delay >	0 All	It is not required to support different values in Backwards direction
<< Window size >>			(See note 2) For the default values if not included see clause 12.5.1 of [15]
	ext3	0	

Information element	Field within the information element	Standard values within the field/information element	Normative action/comment
	< Window size value (forward) >	All	The value shall be placed in both 3 and 3a octets as defined in ETSI EN 300 175-5 [5], clause 7.7.43 (for the range of allowed values see clause 11.1.1 of [15]) Maximum allowed for this profile value = 256 (see note 3)
	ext3a	1	
	< Window size value (forward) continue >	All	
	ext4	0	
	< Window size value (backward) >	All	The value shall be placed in both 3 and 3a octets as defined in ETSI EN 300 175-5 [5], clause 7.7.43 (for the range of allowed values see clause 11.1.1 of [15]) Maximum allowed for this profile value = 256
	ext4a	1	
	< Window size value (backward) continue >	All	

NOTE 1: This value may only be used if all other parameters have values equal to the default values (see clause 12.5.1 of [15]).
NOTE 2: If octet group 4 (i.e. 4, 4a, 4b) is omitted the values defined in Octet group 3 apply for both directions.
NOTE 3: The values introduced in clause 11.1.1 of [15] need to be respected in all window-size fields.
NOTE 4: The direction of the connection downlink (FT-to-PT) or up-link (PT-to-FT) will be dynamically negotiated at MAC layer.
NOTE 5: For backwards compatibility, if octet 7 is not included support of 2-level modulation scheme for both A- and B-field shall be assumed.
NOTE 6: If "Symmetric" is indicated octets 4b, 4c, 5a and 6a need not to be included.
NOTE 7: Value 191 (equivalent to 1 528 octets) is codec as '0000001'B in octet 5 and '0111111'B in octet 5a.

7.5.9 IWU-attributes change

The contents of the messages applicable to this procedure shall be based on the requirements of the ETSI EN 301 649 [15], clause 12.7.

For the purpose of this ASAP only the status of the fields and specific values implementation that **has changed** is indicated in this clause. For the rest whatever specified in ETSI EN 301 649 [15] shall apply.

Table 23: Values used within the {CC-SERVICE-CHANGE} message

Information element	Field within the information element	Standard values within the field/information element	Normative action/comment
<<IWU attributes>>			
	<Profile>	00001	OUT OF SCOPE (Stream support)
		00000	MANDATORY (FREL support)
	<Profile Subtype>	0000	OUT OF SCOPE (IEEE 802.3 [i.7]/ Ethernet (WLAN))
		1000	OUT OF SCOPE (Interworking to V.24 circuits (RS232)
		0001	OUT OF SCOPE (IEEE 802.5 [i.8], (clause B.5))
		0010	OUT OF SCOPE (Internet Protocol (IP) (clause B.6 (RFC 791 [i.16]))
		0100	OUT OF SCOPE (Point-to-Point Protocol (clause B.7 (RFC 1661 [i.9]))
		1000	MANDATORY (Generic Media Encapsulation Protocol (clause B.8))

7.5.10 Broadcast attributes management

The contents of the messages applicable to this procedure shall be based on the requirements of the ETSI EN 301 649 [15], clause 12.16.

For the purpose of this ASAP only the status of the fields and specific values implementation that **has changed** is indicated in this clause. For the rest whatever specified in ETSI EN 301 649 [15], shall apply.

Table 24: Extended higher layer capabilities interpretation by the PP

BIT Number	Attribute	Value	Note
a27	Generic Media Encapsulation	1	MANDATORY
a29	Ethernet	x	OUT OF SCOPE
a30	Token Ring	x	OUT OF SCOPE
a31	IP	x	OUT OF SCOPE
a32	PPP	x	OUT OF SCOPE
a33	V.24	x	OUT OF SCOPE
a45	DPRS Class 3 or Class 4 management and A-field procedures (or both) supported (DPRS-M.30),	1	MANDATORY (see note)
a46	DPRS Class 2 management and B-field procedures supported (DPRS-M.5)	0,1	OPTIONAL (IF DPRS-ME.2 THEN "1" ELSE "0").
NOTE: The supported management Classes are inferred from the state of bit a21 (MAC suspend/resume supported). If bit a21= 1 then Class 3 and Class 4 are supported, if a21 is = 0, then only Class 4 is supported.			

Table 25: Extended higher layer capabilities part 2 interpretation by the PP

BIT Number	Attribute	Value	Note
< a25 - a28 >	NG-DECT Packet Data Category	any	Irrelevant for the present specification. System may be categorized due to other data capabilities
a35	no-emission mode support	0,1	OPTIONAL
a45	Light data services (ETSI TS 102 527-4) supported	1	MANDATORY

7.6 Application layer procedures

The procedures as specified in ETSI EN 301 649 [15], clause 12 shall apply with the modifications and additional procedures listed in the present clause.

7.6.1 Binary content download

7.6.1.1 Binary content download general requirements

The "Binary content download" feature allows the design of "PP to server" distributed applications (see the definition below), based on HTTP, and usable with any Part 4 FP.

NOTE 1: "Binary" refers to DECT layers transparency toward the type of files being transmitted to the PP.

Distributed application. A distributed application is an application available to the user on a DECT handset, for which part of the code (behaviour) and/or data is located on the handset (local tier) and part of it is located in the network (remote tier). The remote tier may consist in one or more HTTP servers.

There is a strong relationship between the local and remote tiers of the application, so that the PP and servers are assumed to be designed in a compatible manner.

NOTE 2: This feature may be used with several possible design structures: HTTP servers hosted by the PP manufacturer, or PP vendor, or other parties.

For interoperability reasons, the remote tier of a pure binary content download application should not be hosted on the FP.

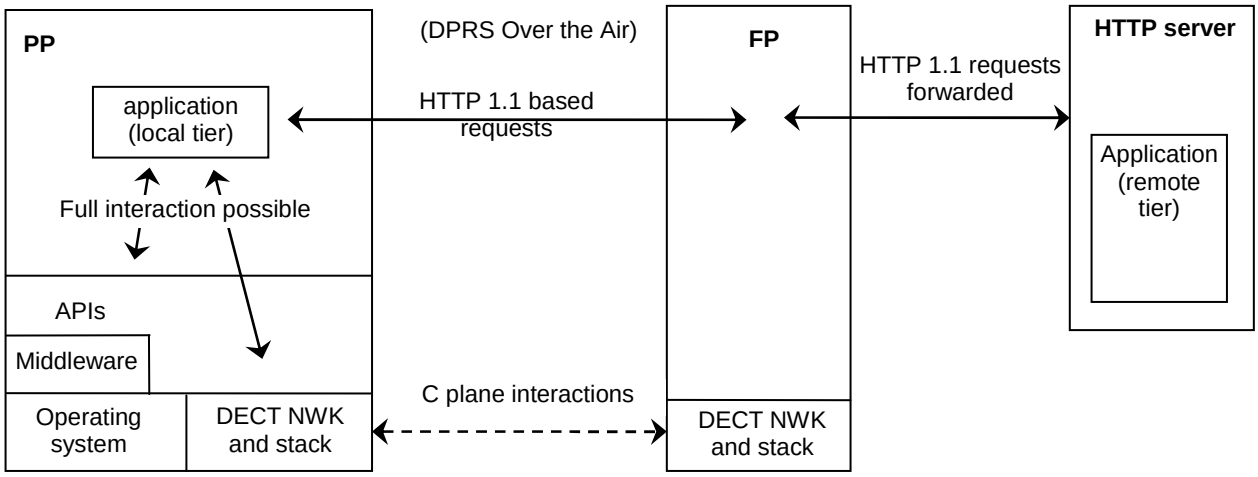


Figure 12: Distributed application using "Binary content download"

7.6.1.2 LU10 interworking conventions and HTTP profile

7.6.1.2.1 LU10 interworking conventions and HTTP profile for "Simple binary content download"

7.6.1.2.1.1 Concerned devices and connections

For the "Simple binary content download" procedure, a file transfer shall occur in the U-Plane based on a single context of HTTP protocol, transported over DPRS Generic Interworking Encapsulation Protocol (ETSI EN 301 649 [15], clause B.8).

Devices allowed to use this profile. NG DECT Part 4 PTs may setup a connection with this profile if FT is also Part 4. Any Part 4 device shall at least support this profile, as described below.

Characterizing connections using this profile. Connections setup by a device allowed to use the profile (see above) using a basic service (LDS with Class 4, or LDS with Class 3), and that are created with a single context.

Table 25a: Relevant capabilities for use of this profile

Relevant features for use of this profile	PT	FT	Comments
DPRS Generic media encapsulation procedure			
ETSI EN 301 649 [15], clause B.8.4.3 (see note 1)	M	M	As amended by the present procedure.
DPRS management			
Class 2 (DPRS-ME.2)	I	I	
Class 3 (DPRS-ME.3)	O	O	If supported, shall not be used for NGLDS-A.2 feature (SUOTA)
Class 4 (DPRS-ME.4)	M	M	
Basic services	M	M	Use of one of the basic services listed below is mandatory on the connection initiating side. It is always used together with limited support of DPRS-N.34 (see below in this table)
Light data service, with Class 4 DPRS management (A.2.1)	M	M	
Light data service, with Class 3 DPRS management (A.2.2)	C2511	C2511	Can only be used by PT if both parties declare support for Class 3. If supported, shall not be used for NGLDS-A.2 feature (SUOTA)

Relevant features for use of this profile	PT	FT	Comments
Features			
Dynamic Parameter Allocation (DPRS-N.33)	M	M	Only the default IE based allocation shall be used (note 2)
Service Negotiation at virtual call setup (DPRS-N.34)	M	M	<<IWU-ATTRIBUTE>> IE shall be present (and only this N.34 related IE) (see clause 7.6.1.2.1.3)
In call service change (DPRS-N.35)	O	O	notes 3, 4
C2511: IF DPRS Class 3 management (DPRS-ME.3) supported THEN M ELSE I.			
NOTE 1: Simplified single-context Interworking to External Networks (or to an application proxy).			
NOTE 2: Use of the present profile implies support of protocol indicator 1079 only (i.e. of ETSI EN 301 649 [15], clause B.8.3.4, alias Common HTTP profile), as described in ETSI EN 301 649 [15], clause A.2.1, table A.11 and ETSI EN 301 649 [15], clause A.2.2, table A.17 relating to the (mandatorily used) basic services. However, as described in ETSI EN 301 649 [15], clause B.8.3.2, the "Extended HTTP profile" (alias protocol ETSI EN 301 649 [15], clause B.8.3.5 with protocol indicator 1077) may however be used by both parties even though protocol indicator 1079 is indicated at connection setup. When this occurs, the FT is anyway informed of PT support for ETSI EN 301 649 [15], clause B.8.3.5 through the HTTP User-Agent Header (see clauses 7.6.3.2, A.1.4 and A.2.3).			
NOTE 3: This feature may be used by PT or FT in order to switch over to the multi-context profile (see clause 7.6.2.1.2) by creating a new context, if both parties comply with table 25d of clause 7.6.1.2.2. This implies that both parties support DPRS-N.35. For the sake of simplicity, it is not allowed to go back to the present profile after switching over to multi-context. This shall not be used if the connection was created for the NGLDS-A.2 feature (SUOTA).			
NOTE 4: If supported, this feature may also be used in order to change the DPRS management class during the connection (see ETSI EN 301 649 [15], clause 12.6.5).			

7.6.1.2.1.2 Allowed PT behaviors

Table 25b: Allowed PT behaviors and application features when using this profile

Condition for using the behavior	Prerequisite FT	PT behavior	Restrictions
PT supports Class 4 only (i.e. not Class 3)	FT supports Class 4 at least	Start a simple BCD connection with LDS CLass 4 basic service (note 1)	To be used for NGLDS-A.2 feature. May be used for NGLDS-A.3 feature.
PT supports Class 3 and Class 4		Start a simple BCD connection with LDS Class 4 basic service (note 1)	To be used for NGLDS-A.2 feature (note 3). May be used for NGLDS-A.3 feature (note 4).
	FT supports Class 4 and Class 3	Start a simple BCD connection with LDS Class 3 basic service (note 1)	Shall not be used for NGLDS-A.2 feature. Should be used for NGLDS-A.3 feature if FT also declares support for class 3. N.35 is not needed.
NOTE 1: When using this basic service, use of B.8.3.4 (note 5) is assumed, but use of B.8.3.5 (note 7) compliant application packets is however allowed (see ETSI EN 301 649 [15], clause B.8.3.2) see also note 2.			
NOTE 2: A FP implementing "NGLDS-A.3 HTTP based applications" necessarily supports ETSI EN 301 649 [15], clause B.8.3.5 (alias "Extended HTTP profile" of clause A.2).			
NOTE 3: Switching over to Class 3 with N.35 (even if both parties support Class 3 and N.35) is not allowed.			
NOTE 4: This should however be avoided if FT also supports Class 3 but N.35 is not supported by both parties, because this implies that the connection will never take advantage of suspend resume although both parties support it.			
NOTE 5: ETSI EN 301 649 [15], clause B.8.3.4 (protocol indicator 1079) is HTTP limited set nr.2, also known as Common HTTP profile (clause A.1 of the present document). ETSI EN 301 649 [15], clause B.8.3.4 includes: - support by both the PP and FP of the HTTP partial GET method (see note 6); - support by the PP of the media type "application/octet-stream" at least; - support of a limited set of HTTP headers.			
NOTE 6: A partial GET is characterized by the presence of a "Range" header in an HTTP GET request. Presence of this header is mandatory for all GET requests between the PP and the FP (as described in the "Common HTTP profile"); even if a whole resource is requested.			
NOTE 7: ETSI EN 301 649 [15], clause B.8.3.5 (protocol indicator 1077) is HTTP limited set nr.3 also known as Extended HTTP profile (clause A.2 of the present document).			

7.6.1.2.1.3 Negotiable elements

The present profile provides limited support for 'Service Negotiation at virtual call setup' (DPRS-N.34), as indicated in table 25a above.

Negotiation indicator. The negotiation indicator shall be '010' indicating that peer attribute negotiation is enabled.

Negotiable subfields. The << IWU-ATTRIBUTES >> IE shall only be used in order to negotiate the "Max SDU size" value that will be used for the connection in each direction (PT => FT and FT => PT).

Negotiated value. For a given direction, the << IWU-ATTRIBUTES >> IE:

- shall be used by the initiating side, in order to systematically specify the "Max SDU size" it supports;
- shall be used by the receiving side, in order to send back the negotiated value for the "Max SDU size" in this direction, that is:
  - if the value received from the initiating side is acceptable (i.e. lower or equal to the "Max SDU size" value that the receiving side supports), the receiving side shall send back the received value as negotiated value;
  - if the value received from the initiating side is NOT acceptable (i.e. greater than the "Max SDU size" value that the receiving side supports), the receiving side shall send back the "Max SDU size" it supports as negotiated value.

Other subfields shall be fixed (see 'Values of non-negotiable subfields' below).

NOTE: The negotiated "Max SDU size" (i.e. that will be used for the connection) in a given direction is the minimum of the two values supported for this direction by the PT and FT.

Allowed range. The "Max SDU size" (in octets) supported by each side for a given direction shall be between the minimum and maximum values indicated in table 25c for this direction.

Table 25c: Allowed range for "Max SDU size" value supported (and negotiated)

Max SDU Size	Minimum value	Maximum value
direction PT => FT	752 (coded '0000000'B + '1011110'B = '5E'H = 94 = 752/8)	1528 (note 3) (coded '0000001'B + '0111111'B = 'BF'H = 191 = 1528/8)
direction FT => PT	752 (coded '0000000'B + '1011110'B = '5E'H = 94 = 752/8)	12224 (note 3) (coded '0001011'B + '1111000'B = '5F8'H = 1528 = 12224/8)
NOTE 1: The 4 values shall be considered by both the PT and FT. NOTE 2: The 4 values are multiples of 8 (as the value coding is 'value/8'). The value proposed by the initiating side and the negotiated value shall therefore necessarily be also multiples of eight. NOTE 3: The maximum values are compatible with the maximum application packet sizes specified for the present document (12 kBytes downstream and 1,5 kBytes upstream) when chopping (application packet size segmentation) is not used. If a smaller value is negotiated, then the maximum application packet size used for the connection shall also be reduced accordingly.		

EXAMPLE: In particular, the << IWU-ATTRIBUTES >> IE:
- shall not be used in order to create more than one context.
- shall not be used in order to change the application protocol identifier associated with the first (and only) context.

Values of non-negotiable subfields. Figure 12a shows negotiable (grey background) and non-negotiable subfields. Both the initiating and receiving sides shall use the fixed value shown in figure 12a for non-negotiable subfields:

Bit:	8	7	6	5	4	3	2	1	Octet:
		<< IWU-ATTRIBUTES >> '12'H							**1**
		Length of Contents (L) '0A'H							**2**
	1	Code std '01'B		Profile '00000'B (DPRS: Frame Relay services)					**3**
	ext4 1	Negotiation indicator '010'B (Peer attribute negotiation)			Profile subtype '1000'B (DECT Generic Media Encapsulation)				**4**
	0	Maximum SDU size (PT ⇒ FT) MSB (7 bits)							**5**
	0	Maximum SDU size (PT ⇒ FT) LSB (7 bits)							**5a**
	0	Maximum SDU size (FT ⇒ PT) MSB (7 bits)							**5b**
	1	Maximum SDU size (FT ⇒ PT) LSB (7 bits)							**5c**
	ext6 1	Operation code '01'B		Optional groups '00'B		CH=0 '0'B	Spare '00'B		6
	seq '0'B	Generic Media context indicator (GMCI) '0000000'B (PT) '0000001'B (FT)							6a
	Application protocol identifier (MSB) '04'H								**6b**
	'37'H								**6c**

NOTE 1: The negotiation indicator shall be '010' indicating that peer attribute negotiation is enabled.
NOTE 2: The present profile limits negotiation to octets 5, 5a, 5b and 5c. Octets 5b and 5c shall always be present (even though they may in principle be absent when both sizes happen to be the same). All other fields shall always contain the fixed indicated value.
NOTE 3: The 'seq' flag (octet 6a, bit 8) shall be unset on both sides, indicating that SDU numbering is not used.
NOTE 4: The GMCI value used for the connection shall be 1. As the GMCI value is always defined by the FP:
- the PT shall use value "0" (indicating a not yet defined GMCI) in the <<IWU-ATTRIBUTES>> IE of the {CC-SETUP} (first message of the connection); and
- the FT shall use value 1 in the <<IWU-ATTRIBUTES>> IE of the {CC-CONNECT}.
Then in all subsequent messages (e.g. in SDU headers), the defined value 1 shall be used.

Figure 12a: << IWU-ATTRIBUTES >> IE with negotiable (in grey background) and non-negotiable data

7.6.1.2.2 LU10 interworking conventions and HTTP profile for "Enhanced binary content download"

For the "Multi-context binary content download" procedure, a file transfer shall occur in the U-Plane based on one or several contexts of HTTP protocol, transported over DPRS Generic Interworking Encapsulation Protocol (ETSI EN 301 649 [15], clause B.8).

Devices allowed to use this profile. NG DECT Part 4 PTs may setup a connection with this profile (or switch to this profile from 7.6.1.2.1 through use of DPRS-N.35) if FT is also Part 4, and both parties (PT and FT) indicate support of ETSI EN 301 649 [15], clause B.8.4.2 in a sent <<SETUP CAPABILITY>> IE. If a Part 4 PT or FT indicates support for ETSI EN 301 649 [15], clause B.8.4.2, it shall support the "Multi-context binary content download" as described below.

Characterizing connections using this profile. Connections setup by a device allowed to use the profile (see above) with a basic service (LDS with Class 4, or LDS with Class 3), as soon as more that one context is created.

- This may occur at connection setup, if DPRS-N.34 is supported by both parties.
- This may also occur if a connection uses profile 7.6.1.2.1, if new contexts are created at some time (implying support of DPRS-N.35 by both parties).

NOTE: As a consequence, support of this profile (ETSI EN 301 649 [15], clause B.8.4.2 capability indicated on both sides through DPRS-N.33 with either default IE or sent IE) does not imply use of this profile at connection setup.

Table 25d: Relevant capabilities for use of this profile

Relevant features for use of this profile	PT	FT	Comments
DPRS Generic media encapsulation procedure			
ETSI EN 301 649 [15], clause B.8.4.2 (Multi-context Interworking; see note 1)	M	M	As amended by the present procedure
DPRS management			
Class 2 (DPRS-ME.2)	I	I	
Class 3 (DPRS-ME.3)	O	O	
Class 4 (DPRS-ME.4)	M	M	
Basic services	M	M	Use of one of the basic services listed below is mandatory on the connection initiating side. It is always used together with limited support of DPRS-N.34 (see below in this table)
Light data service, with Class 4 DPRS management (A.2.1)	M	M	
Light data service, with Class 3 DPRS management (A.2.2)	C2541	C2541	Can only be used if both parties declare support for it
Features			
Dynamic Parameter Allocation (DPRS-N.33)	M	M	On PT and FT sides, use of the sent IE is required in order to indicate support for "B.8.4.2" as it is not supported in any default IE (notes 2, 3)
Non-GMEP capabilities: not listed here	M	M	Shall be listed in the sent IE (not implied) see note 6
Minimum (or optional) GMEP capabilities as listed below:			(note 5)
46H (Multicontext Interworking, B.8.4.2; see note 1)	M	M	(note 4)
01H (HTTP limited set nr.1, B.8.3.3, p.i.c 1078)	I	I	(note 4)
02H (HTTP limited set nr.2, B.8.3.4, p.i.c 1079)	M	M	Common HTTP profile (clause A.1) (note 4)
03H (HTTP limited set nr.3, B.8.3.5, p.i.c 1077)	O	M	Extended HTTP profile (clause A.2) (note 4)
Service Negotiation at virtual call setup (DPRS-N.34)	M	M	<<IWU-ATTRIBUTE>> IE shall be present (and only this N.34 related IE) (note 7)
In call service change (DPRS-N.35)	M	M	This allows to create new contexts and to release contexts (note 8)

C2541: IF DPRS Class 3 management (DPRS-ME.3) supported THEN M ELSE I.

NOTE 1: ETSI EN 301 649 [15], clause B.8.4.2, Multi-context Interworking to an application proxy.

NOTE°2: This feature uses either a default IE or a non default <<SETUP CAPABILITY>> IE sent to the opposite side (PT first, FT answer) at location registration and/or location subscription. The sent IE can only be taken into account if both side send it, otherwise a default IE is used.

NOTE 3: Protocol identifier codes (p.i.c) are defined in ETSI EN 301 649 [15], clauses B.2.1.1.4.3 and B.8.3.2.

NOTE 4: This is a capability octet value to be indicated (or not) within the <<SETUP CAPABILITY>> IE. The device may indicate several values. See ETSI EN 301 649 [15], clauses 12.8 and 12.22.

NOTE 5: All supported protocols shall be present in the IE even if the implementation only targets the use of basic services (that include an assumed default IE in clause A.2).

NOTE 6: See ETSI EN 301 649 [15], clause A.2.1, table A.11 if DPRS-ME.4 only is supported, or clause A.2.2, table A.17 if both DPRS-ME.4 and DPRS-ME.3 are supported.

NOTE 7: The IE shall be used in order to systematically specify the "Max SDU size" PT => FT and FT => PT (The indicated values shall be independent of the application (if more than one is supported) and context (if more than one is created). Octets 5b and 5c shall always be present, even if both sizes happen to be the same. The IE may also be used in order to create more than one context, or in order to specify the application protocol identifier associated with each context (among protocols indicated with N.33). Other uses not listed here are not allowed.

NOTE 8: DPRS-N.35 is the only way to use more than 4 contexts, or to release an existing context. N.35 may be used instead of N.34 in order to create more than one context (see note 7). This feature may also be used in order to change the DPRS management class during the connection (see ETSI EN 301 649 [15], clause 12.6.5).

Table 25e: Allowed PT behaviors and application features when using this profile

Condition for using the behavior	Prerequisite FT	PT behavior	Restrictions
As described in table 25b for the considered behavior	As described in table 25b for the considered behavior	Any behavior as described in table 25b for profile 7.6.1.2.1	Allowed with the restrictions described in table 25b for the considered behavior
None	None	Use N.34 in order to start the connection with more than one context	Shall not be used for NGLDS-A.2 feature. Shall not be used for NGLDS-A.3 feature. (note 2)
None	None	Use N.35 in order to change the number of contexts	N.35 allows to create new contexts and to release contexts. NGLDS-A.2 shall not be used in this context A new context could be created in order to support NGLDS-A.3.
PT supports Class 3 and Class 4	FT supports Class 3 and Class 4	Use N.35 in order to change the DPRS management class	Both parties support Class 4 + Class 3 See ETSI EN 301 649 [15], clause 12.6.5.
N/A	N/A	Go back to profile 7.6.1.2.1 by deleting all but one contexts	Not allowed
NOTE 1: No conditions except those defined in table 25d above for use of the profile. NOTE 2: This possibility is for a not yet defined feature.			

7.6.1.2.3 LU10 interworking conventions and HTTP profile for "Generic multiprotocol binary content download"

For the "Generic multiprotocol binary content download" procedure, a file transfer shall occur in the U-Plane based on one or several contexts of HTTP protocol, transported over DPRS Generic Interworking Encapsulation Protocol (ETSI EN 301 649 [15], clause B.8).

Devices allowed to use this profile. NG DECT Part 4 PTs may setup a connection with this profile if FT is also Part 4, and both parties (PT and FT) indicate support of ETSI EN 301 649 [15], clause B.8.4.1 in a sent <<SETUP CAPABILITY>> IE. If a Part 4 PT or FT indicates support for ETSI EN 301 649 [15], clause B.8.4.1, it shall support the "Generic Multiprotocol binary content download" as described below.

Characterizing connections using this profile. Connection setup by a device allowed to use the present profile (see above) and either not supporting the "Simple" or "Multi-context" profiles or using DPRS-ME.2 or any of the protocols not allowed with the "Simple" or "Multi-context" profiles (Full HTTP, DNS, ETSI EN 301 649 [15], clause B.8.3.3).

Table 25f: Relevant capabilities for use of this profile

Relevant features for use of this profile	PT	FT	Comments
DPRS Generic media encapsulation procedure			
ETSI EN 301 649 [15], clause B.8.4.1 (see note 1)	M	M	As amended by the present procedure.
DPRS management			
Class 2 (DPRS-ME.2)	M	M	
Class 3 (DPRS-ME.3)	O	O	
Class 4 (DPRS-ME.4)	O	O	
Basic services			
Light data service, with Class 4 DPRS management (A.2.1)	C2561	C2561	Can only be used by PT if both parties declare support for Class 4.
Light data service, with Class 3 DPRS management (A.2.2)	C2562	C2562	Can only be used by PT if both parties declare support for Class 3
Features			
Dynamic Parameter Allocation (DPRS-N.33)	M	M	On PT and FT sides, use of the sent IE is required in order to indicate support for "B.8.4.1" as it is not supported in any default IE (notes 2, 3)
Non-GMEP capabilities: not listed here	M	M	Shall be listed in the sent IE (not implied) See note 6
Minimum (or optional) GMEP capabilities as listed below:			(note 5)
45H (Generic Multiprotocol Interworking, B.8.4.1; see note 1)	M	M	(note 4)

Relevant features for use of this profile	PT	FT	Comments
01H (HTTP limited set nr.1, B.8.3.3, p.i.c 1078)	O	O	(note 4)
02H (HTTP limited set nr.2, B.8.3.4, p.i.c 1079)	M	M	Common HTTP profile (clause A.1) (note 4)
03H (HTTP limited set nr.3, B.8.3.5, p.i.c 1077)	O	O	Extended HTTP profile (clause A.2) (note 4)
04H (Full HTTP, as in RFC 2616 [19], p.i.c 80)	M	M	
0AH (DNS, as in RFC 1034 [20] and RFC 1035 [21], p.i.c 53)	M	M	
Service Negotiation at virtual call setup (DPRS-N.34)	C2563	C2563	
In call service change (DPRS-N.35)	M	M	
C2561: IF DPRS Class 4 management (DPRS-ME.4) supported THEN M ELSE I. C2562: IF DPRS Class 3 management (DPRS-ME.3) supported THEN M ELSE I. C2563: IF DPRS Class 2 management (DPRS-ME.2) supported THEN M ELSE O.			
NOTE 1: ETSI EN 301 649 [15], clause B.8.4.1, Generic Multiprotocol Interworking to External IP Networks. NOTE°2: This feature uses either a default IE or a non default <<SETUP CAPABILITY>> IE sent to the opposite side (PT first, FT answer) at location registration and/or location subscription. For class 3 or 4, the sent IE can only be taken into account if both side send it, otherwise a default IE is used. For class 2, a sent IE from one side may be used with a default IE implied from the other side. NOTE 3: Protocol identifier codes (p.i.c) are defined in ETSI EN 301 649 [15], clauses B.2.1.1.4.3 and B.8.3.2. NOTE 4: This is a capability octet value to be indicated (or not) within the <<SETUP CAPABILITY>> IE. The device may indicate several values. See ETSI EN 301 649 [15], clauses 12.8 and 12.22. NOTE 5: All supported protocols shall be present in the IE even if the implementation only targets the use of basic services (that include an assumed default IE in clause A.2). NOTE 6: See ETSI EN 301 649 [15], clause A.2.1, table A.11 if DPRS-ME.4 only is supported, or clause A.2.2, table A.17 if both DPRS-ME.4 and DPRS-ME.3 are supported.			

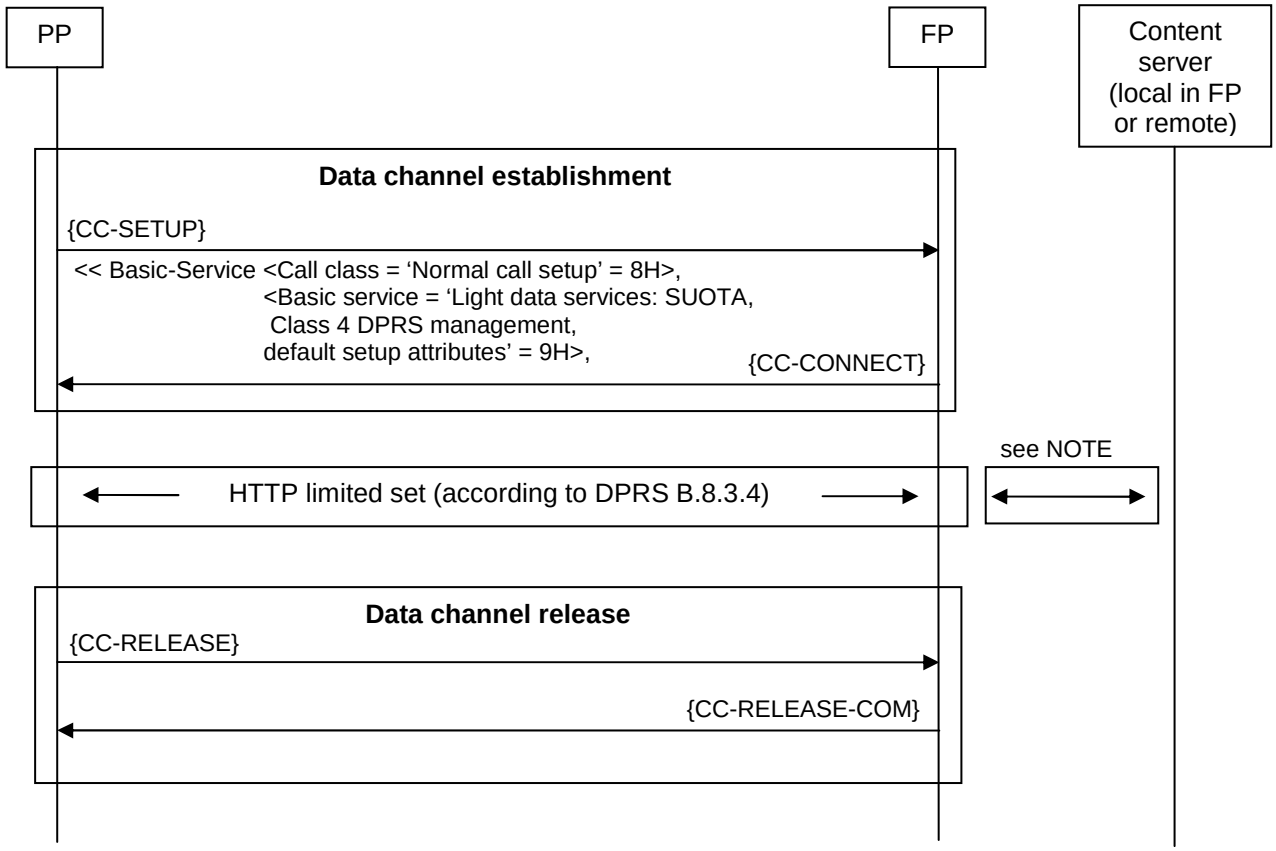
7.6.1.3 Binary content download media type

For the "Binary content download" feature, the client shall support at least media type "application/octet-stream", as specified in clause A.1.5.

NOTE: In the context of the HTTP protocol, media types are used in the "Accept" (from client) and "Content-Type" (from server) header fields. Media types allow content type negotiation and content type notification.

7.6.1.4 Binary content download sequence

The binary content download feature shall use the following sequence of messages.



NOTE 1: FP to content server data exchange is out of scope of the "Binary content download" feature description. This data exchange may or may not involve HTTP.
NOTE 2: Basic service value is defined in ETSI EN 300 175-5 [5]. Default setup attributes are defined in ETSI EN 301 649 (DPRS) [15], clause A.2.

Figure 13: DECT connection for "Binary content download"

File downloading resumption. Thanks to the use of partial GETs ("Range" header mechanism), the PP can download a file in several steps, either by using several GET messages within the same connection, and/or by using several connections (e.g. following a connection break down).

7.6.1.5 URI-based PP to FP security requirements

The following two procedures allow the PP to require the FP to use one or several security features with the server when retrieving an HTTP resource over the network.

NOTE 1: These procedures use the Request URI and therefore extend clause A.1.3, "Request URI and Host header field" of the "Common HTTP profile".

NOTE 2: Both types of security requirements (respectively defined in clauses 7.6.1.5.1 and 7.6.1.5.2) may be combined in a single URI.

NOTE 3: Use of the following two procedures, together with an extension of their use with URLs sent via the C-plane, is illustrated in clause B.3 in the case of SUOTA. See also clause 7.6.2.4.

In case the FP does not implement the required security feature, the FP shall send the status code "700 Unimplemented security feature".

7.6.1.5.1 URI-based PP to FP confidentiality requirement

The "URI-based PP confidentiality requirement" procedure allows the PP to request the FP to use confidentiality when retrieving an HTTP resource over the network.

NOTE 1: A security requirement for confidentiality is not intended to protect PP to FP exchange of data, which are protected by DECT specific means.

A PP implementing the present procedure:

- shall be able to send to the FP a URI with "https" scheme value, as in:
  GET https://suota.example.com/path/file.bin **HTTP/1.1;**

- shall be prepared to receive a status code "700 Unimplemented security feature" as a response to the request.

A FP implementing the present procedure:

- shall implement HTTPS;

- shall interpret the presence of the "https" scheme value as a requirement for using HTTPS with the corresponding server for ensuring the confidentiality of the exchange, and shall fulfil this requirement.

NOTE 2: HTTPS relies on a SSL/TLS layer between HTTP and TCP.

NOTE 3: Encryption only relies on the server encryption private and public keys, and corresponding certificate (the FP does not need to own any encryption key). The FP should implement a TLS/SSL stack and embed/trust the public key of the server certificate authority, or of one of its ancestors.

7.6.1.5.2 URI-based PP to FP authentication requirement

The "URI-based PP authentication requirement" procedure allows the PP to request the FP to use a client authentication scheme on behalf of the PP when retrieving an HTTP resource over the network.

A PP implementing the present procedure:

- shall be able to send to the FP a URI with *authority* part containing a *userinfo* element of the form "username:password@", as in the following example:

EXAMPLE: **GET** http://pp1:mypassword@suota.example.com/path/file.bin **HTTP/1.1**.

- shall be prepared to receive a status code "700 Unimplemented security feature" as a response to the request.

A FP implementing the present procedure:

- shall implement HTTP-based "basic" **and** "digest" authentication;

- shall interpret the presence of a "userinfo" element within the "authority" part of the URI as a requirement for using HTTP-based "basic" or "digest" authentication (as requested by the server) with the corresponding server for ensuring the authentication of the PP, and shall fulfil this requirement.

NOTE: For PP authentication to work this way, the PP should entrust the FP with its own authentication password, and therefore should trust the FP.

7.6.1.6 PP to FP enhanced interactivity

The "PP to FP enhanced interactivity" procedure allows the PP to send HTTP requests with data (most notably "completed forms") to the FP (and beyond). It allows to create enhanced "Binary content download" based applications.

NOTE 1: This procedure is not used by the "Software upgrade over the air" feature, which relies on the C-plane for the same purpose.

A PP compliant with the present procedure shall implement the "Extended HTTP profile" of clause A.2.

NOTE 2: Distributed applications with a PP embedded local tier may exist that do not make use of the enhanced interactivity provided by the present procedure-only relying on URLs for PP to FP (limited) interactivity. However, URL-based interactivity (such as parameters sending) is inefficient for large amounts of data and does not provide a standard way of handling character encodings.

A PP compliant with the present procedure shall be able to send POST requests using the "multipart/form-data" format as described in clause A.2 (and especially upon request of the server, e.g. using "encType" attribute if XHTML <form> tags are supported).

A FP compliant with the present procedure shall be prepared to receive POST requests as described in clause A.2 and to forward them to the content download application remote tier.

7.6.2 Software upgrade over the air (SUOTA)

7.6.2.1 SUOTA general requirements

7.6.2.1.1 Definitions

Software package: A set of files sharing the same version identifier, and needed by the PP for installing or upgrading an application or a firmware. The software package is often simply referred to as the "software".

Software version identifier: This parameter identifies a software package, including the software package version. From PP to FP, this parameter identifies the currently installed software package. From FP to PP it identifies the software package to be installed as a result of the upgrade (and is shared by all the files needed for the upgrade). Details and examples are provided in clause 7.5.5.2.1.

Software upgrade management server (MS): The site of a PP vendor, or operated on behalf of a PP vendor, where information about new software image releases for handsets, and their locations (on the downloading server) can be found.

Software upgrade downloading server (DS): The site of a PP vendor, or operated on behalf of a PP vendor, from where the software image releases can be downloaded.

7.6.2.1.2 SUOTA general description

The "Software upgrade over the air" feature allows the software (or firmware) upgrade of a New Generation DECT, Part 4 PP in front of a New Generation DECT, Part 4 FP. This software may be made of one or several files sharing the same version. The SUOTA feature includes procedures for:

- information exchange in the C-Plane; and
- actual downloading of the software (or firmware) by the PP.
    - For this download, the "Binary content download" feature shall be used (i.e. an HTTP connection is used between the PP and the FP). Moreover, the binary content download shall only use the "Simple" profile described in clause 7.6.1.2.1, with Class 4 DPRS management (see also note 2 in table 13 for more details).
    - Implementation of the "Binary content download" feature is therefore a prerequisite for the SUOTA feature.

As an addition to the "Binary content download" model, the SUOTA feature uses an extra "middle tier" on the FP which exchanges software upgrade specific commands with the PP and issues its own requests toward the management server.

The SUOTA feature defines the following two types of SUOTA. Both SUOTA types use a common PP to FP protocol.

1) **Basic SUOTA** is described end-to-end from PP to MS and from PP to DS, and uses plain HTTP up to the servers. FP to MS protocol is the object of annex B.
2) **Enhanced SUOTA** enables the use of a non limited list of optional additional features for the FP to MS exchanges (authentication, confidentiality, device identification, etc.).

NOTE 1: Enhanced SUOTA is expected to be used mostly in the case of telecom operator managed devices.

Only the PP or MS can decide to use enhanced SUOTA; otherwise, basic SUOTA shall be used by the FP. In enhanced SUOTA, although the FP to MS protocol is enhanced, URL1 received from the PP shall still be used by the FP as an entry point to the MS.

NOTE 2: This means that with enhanced SUOTA, the FP to MS protocol should still be HTTP based.

NOTE 3: Use of enhanced SUOTA by the PP/MS can only be successful if the FP implements the needed enhanced features.

Multiple handset systems

In multiple handset systems, each handset handles SUOTA independently. In case of simultaneous attempts from several PPs, the FP could reject all but one attempt from them (using a "Retry later" negative acknowledgement for the other, see clause 7.5.5.2.4).

FP upgrade

Although the SUOTA feature only deals with the upgrade of PPs, a FP implementing the feature shall be capable of firmware upgrade of itself (with a software upgrade method outside the scope of the present document).

The SUOTA process for a PP shall not be interrupted by the upgrade of the FP.

Upgrade of the FP shall always guarantee that the FP is still compatible with the implemented features of the present document as a result of the upgrade. In particular, this means that PPs shall still be able to use SUOTA.

7.6.2.1.3 Protocol overview

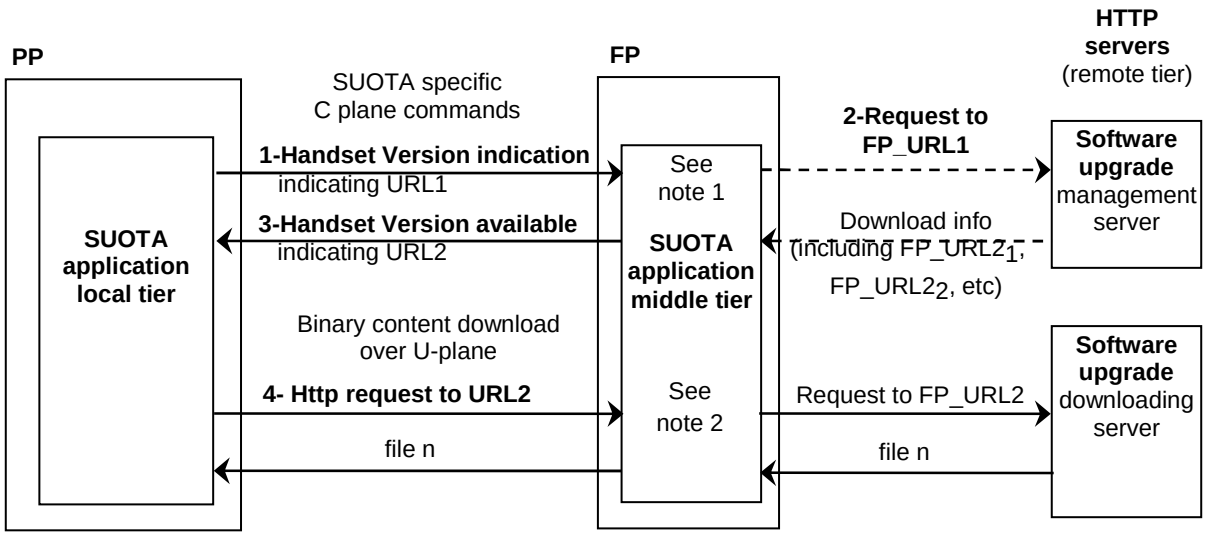


Figure 14: Software upgrade overview

The SUOTA feature comprises 4 steps summarized in figure 14 above and further detailed in clause 7.6.2.2:

- Steps 1, 3 and 4 are repeated as many times as there are files to be downloaded for a given software upgrade: the PP carries out step 1 again after step 4, until URL2 in the response (step 3) is absent.
- Step 2 is the step in which the FP retrieves the next file to be downloaded. First use of step 2 by the FP includes a request to MS (FP_URL1), in order to get all file urls needed for the upgrade (FP_URL2_n, $1 \leq n \leq N_f$). In the next uses of step 2, the FP retrieves the next file url locally.

NOTE 1: A PP may hold several URL1 values. See clause 7.6.2.2.1.

For the description of the SUOTA feature, the following definitions apply:

URL1: URL1 is defined as a PP-embedded URL entry point to the PP vendor management server.

NOTE 2: A PP could hold several URL1 values. See clause 7.6.2.2.1.

FP_URL1: Request url to the MS, used to retrieve the set of file urls needed for the upgrade.

NOTE 3: If the FP to MS interface of annex B is used, FP_URL1 is equal to URL1 with appended parameters.

FP_URL2: Url variable representing the current file url to be downloaded, from FP point of view (retrieved in step 2).

URL2: Url variable representing the current file url to be downloaded, from PP point of view. URL2 is used by the PP in the "Binary content download" feature from PP to FP. URL2 may be equal or not to FP_URL2 depending on the used scenario (see clauses 7.6.2.2.3 and 7.6.2.2.3 for details).

7.6.2.1.4 SUOTA protocol steps: overview

The "Software upgrade over the air" protocol comprises the 4 steps defined in the following sub-clauses. These steps are used several times. First use of step 2 is remote, subsequent uses of step 2 are local. Figure 15 summarizes the use of these steps.

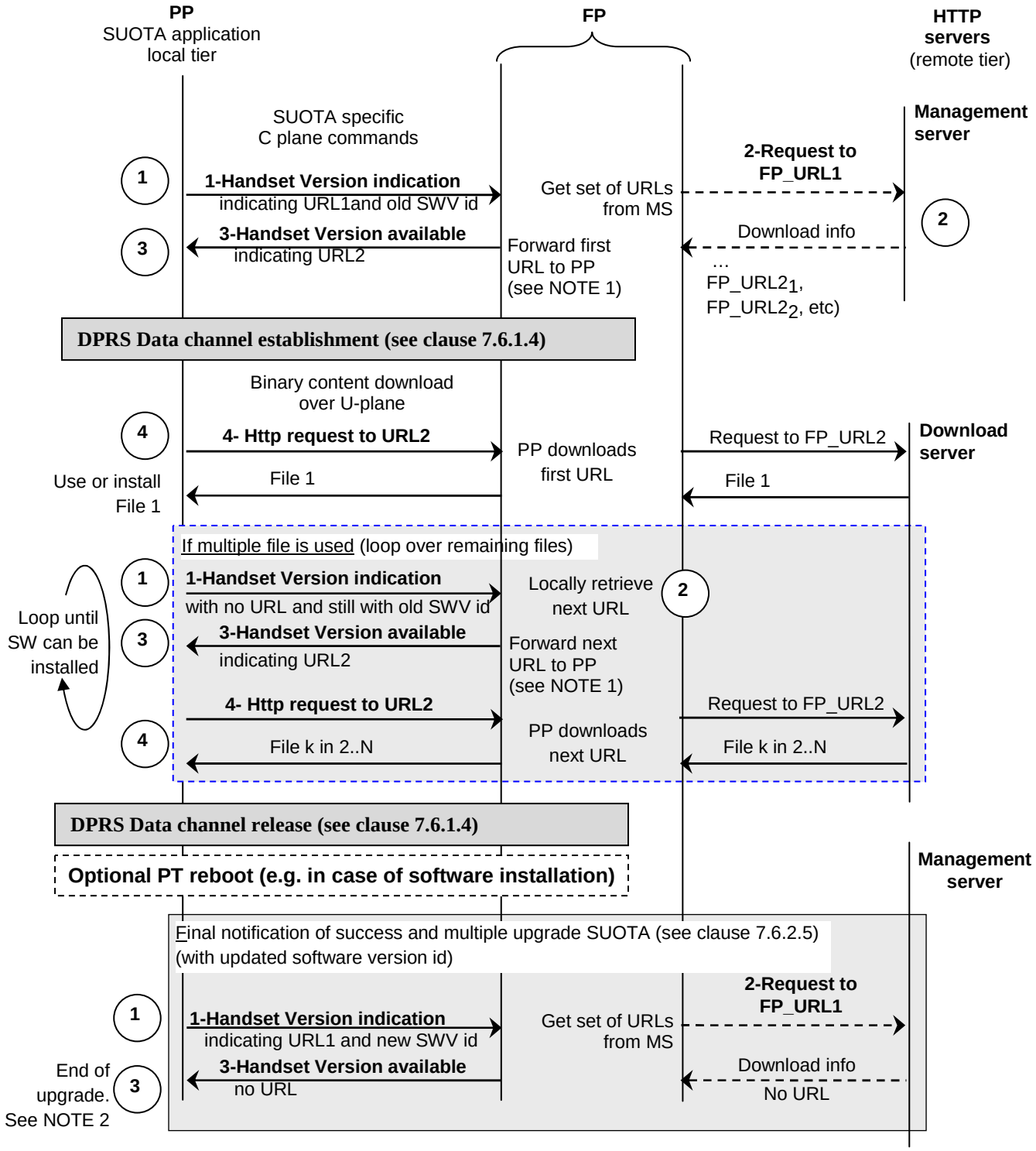


Figure 15: Software upgrade overview sequence chart

NOTE 1: FP_URL2$_1$, FP_URL2$_2$, … may be forwarded as is to the PP (URL2 = FP_URL2$_1$, etc.) or a transient URL2 value pointing to the FP itself may be used instead.

NOTE 2: Retrying software upgrade allows to systematically notify the management server of successful software upgrade. In case of so-called multiple-upgrade SUOTA, the whole process is restarted (in that case, the MS sends a new set of URL instead of "no url"). See clause 7.6.2.5.

NOTE 3: A single PP upgrade may imply the download of several files (cf. the "fileNumber" parameter in the handset version indication command of clause 7.5.5.2.1). Hence the optional "multiple file" block for downloading files 2 to N_f ($N_f \geq 2$).

Use of data connection. The data channel established for downloading the first file shall be used for all subsequent exchanges relating to the upgrade (except for the final notification of success or failure, or in the case of a connection error). This applies:

- in the case of a multiple file upgrade,
  - to all handset version indication/available exchanges (steps 1 and 3) needed for additional files (from the 2nd one on);
  - to all file downloads, including the first one (step 4).

At the end of the last file download, the DPRS data channel shall be released. The final notification of success (clause 7.6.2.5) or failure (clause 7.6.2.6), steps 1 and 3, shall therefore never use this data channel.

Optimization of connection duration. The PP should make the duration of the data connection as small as possible. This will in particular optimize the behavior of a multi-handset DECT system when several PPs need an upgrade.

EXAMPLE: If the PP internal memory used for storing the firmware has to be completely erased prior to a firmware upgrade, this should preferably be done before setting up the data connection.

7.6.2.2 Basic SUOTA protocol steps

Use of Basic SUOTA is illustrated in clause B.3.

7.6.2.2.1 Step 1-PP sends a "Handset version indication" command to the FP

In step 1, the PP sends to the FP a "Handset Version indication" command as defined in clause 7.5.5.2.1. This command shall be sent:

- either following an "external triggering event" as defined below. In that case the "fileNumber" parameter of this command is set to "1", indicating that the first file for the software upgrade to come (if any) is requested.
  - In that case, the <URL to follow> field shall be set to the number of "URL indication" commands following this command that are needed for transmitting URL1.
- or following step 4 (clause 7.6.2.2.4) in order to loop over the steps to retrieve subsequent files (in case more than one file is needed, i.e. in case of multiple file software upgrade).
  - In that case, the <URL to follow> field shall be set to "0".

External triggering event. The external events triggering Basic SUOTA Step 1 and their status on PT side are given by table 25g.

Table 25g: External events triggering Step 1

Event type	Comment	PT side status
Periodic attempt	Coded in the PP for regularly checking availability of a new version (note 1)	C7601
Successful software upgrade	See clause 7.6.2.6, "Final notification of success and multiple step SUOTA" for details	M
Push	When software upgrade is triggered by an MS originating message. See clause 7.5.6 for details	O (note 2)
User initiated SUOTA	See also clause 7.6.2.7, "User initiated SUOTA flag" for the transmission of this event to the MS	C7601 (note 3)
C7601: At least one of the two event types ('periodic attempt' and 'user initiated SUOTA') shall be supported on PT side.		
NOTE 1: The PP could for example send a "Handset version indication" command every 5th day. NOTE 2: This corresponds to the status of procedure 7.5.6, 'SUOTA push mode', on PT side within feature NGLDS-N.2 (see clause 6.7.3). NOTE 3: The status of this event refers to the existence of a menu in the PT for initiating an upgrade manually.		

Command parameters. The "Handset version indication" command includes, among other parameters:

- The <URL1 to follow> parameter, indicating presence of URL1 in one or more subsequent "URL indication" command(s). URL1 is the entry point to the management server (only present for the first use of step 1).
- The current "Software version identifier" of the software installed on the PP and to be upgraded. Goal of the "Software upgrade over the air" feature is the download of a newer version. In case installation of a new version requires uninstallation of the previous version, the previous version identifier shall still be used.
- The value of a file counter ("fileNumber"), indicating the number of the requested file. To request the first file, the fileNumber parameter shall be set to "1".

NOTE 1: A PP may hold several URL1 values, especially to provide for the case the FP would not implement a security requirement included in one of them (see clause 7.6.2.4). Additionally, a different URL1 value could be tried if a "negative acknowledgement" (see clause 7.5.5.2.4) of type "unreachable URL1 (server error)" is received from the FP in response to a "Handset Version indication" command.

Between two consecutive uses of the "Handset version indication" command, the "Software version identifier" shall not change, unless the new software image has been completely downloaded and installed in between (i.e. all files for the new version were received).

NOTE 2: In contrast to this, the "Handset version available" command sent in step 3 (see clause 7.6.2.2.3, option 2) as a response from FP to PP contains the **new** "Software version identifier".

Between two uses of the "Handset version indication" command with the same "Software version identifier", the PP shall increment the value of the "fileNumber" parameter by 1, in order to get the next file url in the subsequent "Handset version available" command. Incrementation of the parameter shall also be interpreted by the FP as a notification of correct download and handling of the previous file by the PP.

Data transport. See 'Use of data connection' in clause 7.6.2.1.4.

Error handling: The PP shall not use the same "fileNumber" value twice for the same "Software version identifier", unless:

- it has not received the corresponding response;
- it uses it to notify a failure as specified in clause 7.6.2.6, "Notification of failure", with a non-zero reason field. This includes possible requests for a new delay if the PP built-in download window cannot be respected. Use of a non-zero value shall imply contacting the MS in step 2.

7.6.2.2.2 Step 2-FP retrieves url of the next file to be downloaded (FP_URL2)

In step 2, the FP retrieves the url of the next file to be downloaded (called $\mathbf{FP\_URL2_n}$) either locally or remotely from the management server.

The first time step 2 is used (n = 1), the FP shall contact the MS in order to retrieve the set of all FP_URL2 values (if a new software version is available). The FP determines the request to be used toward the MS for this purpose (called **FP_URL1**). More specifically:

- the FP shall construct FP_URL1 using URL1 received from the PP as a base URL;
- FP_URL1 shall be constructed by appending device specific information to URL1, as defined in clause B.1, "Basic SUOTA FP to management server interface";
- the FP shall send an HTTP GET request to FP_URL1 on the MS.

NOTE: The FP should be prepared to receive an HTTP redirection status code (3xx) as a result of using URL1.

Based on the received parameters, the MS answers the FP, as described in clause B.2, "Basic SUOTA management server to FP interface":

- if there is a new software version available, the MS returns the download information for the next software version, including the set of all needed **FP_URL2** values (N_f values, with $1 \leq N_f \leq 15$).
- or, if there is NO new software version available, the MS returns an empty download information (see clause B.2, 'Case of an up to date PP').

The download information is received as text information with "Content-Type" value of "application/xml".

In the next uses of step 2 ($n \geq 1$), **FP_URL2$_n$** is retrieved locally in the FP.

Data transport. See 'Use of data connection' in clause 7.6.2.1.4.

7.6.2.2.3 Step 3-The PP receives the "Handset version available" command from the FP

In step 3, the FP sends to the PP as a response to the "Handset version indication" command:

- either a "Handset version available" command defined in clause 7.5.5.2.2. See options 1, 2 and 4 below; or
- if an error occurs, a "Negative acknowledgement" command defined in clause 7.5.5.2.4. See options 3 and 4 below.

Both commands can only be sent to the PP upon reception by the FP of a "Handset version indication" command.

Data transport. See 'Use of data connection' in clause 7.6.2.1.4.

Options for use of step 3. Upon reception of a "Handset version indication" command with a given "Software version identifier", and with a "fileNumber" field equal to "n" (i.e. for the "nth use" of step 3), the FP shall follow one of the following options:

*Option 1 ("No url")***:** If there is no new software image available for the given "Software version identifier", the FP shall reply with a "Handset version available" command with:

- a <URL2 to follow> field set to "0" ("No url" response); and
- an empty <SW version identifier> field.

Option 1 should only be used once, that is, when n = "1".

Use of option 1 shall end the Software upgrade process on PP side (and until a future triggering event occurs).

Note that two main situations lead to option 1:

- an external event triggered the sending of this command (reboot, periodic attempt, etc), but the PP had an up-to-date version installed;
- the PP just completed the installation of a new software version (having received all files pertaining to that version) and sends this command in order to allow for a possible multiple upgrade SUOTA (see step 1, clause 7.6.2.2.1)-but there is no additional step foreseen on MS side. This can be considered a subcase of the previous situation.

*Option 2 ("OK")***:** If a new software image is available for the given "Software version identifier" and the requested file number belongs to [1.. Nf], the FP shall reply with a "Handset version available" command, which:

- shall contain the "Software version identifier" of the new software image (i.e. necessarily different from the value sent in the "Handset version indication" command);
- shall contain a <URL2 to follow> field set to a non-zero value and shall be followed with one or more "URL indication" command(s) containing URL2$_n$, to be used by the PP in step 4 for downloading the nth file (see "URL2$_n$ value determination" below).

*URL2$_n$ value determination***:** URL2$_n$ shall be equal to FP_URL2$_n$. In other words, the FP forwards FP_URL2$_n$ received from the MS "as is" to the PP.

NOTE: This is not necessarily the case when Enhanced SUOTA is used, see clause 7.6.2.3.3.

*DelayMinutes value***:** The provisions of clause 7.5.5.2.2 shall be respected.

*Option 3 ("Negative acknowledgement")***:** If an error occurs, either due to the received 'Handset version indication' command, or due to the FP to MS interaction, the FP shall notify the PP by using a "Negative acknowledgement" command, with a suitable reason field (see clause 7.5.5.2.4).

EXAMPLE 1: (Error due to the received 'Handset version indication' command) If a new software image is available for the given "Software version identifier" but the requested file number exceeds the maximum value ($n > N_f$), the FP shall reply with a "Negative acknowledgement" command, with reason field "File does not exist". This should however never happen, as the PP having received the needed N_f files should have installed the new software version already.

EXAMPLE 2: (Error due to the FP to MS interaction) If the management server could not be reached, the FP shall reply with a "Negative acknowledgement" command, with reason field " Unreachable URL1 (server error)".

If an error occurs, the PP shall nevertheless revert to a stable version.

*Option 4 ("Answer to failure report to MS")***:** If the reason field in the "Handset version indication" was non-zero indicating a PP failure and implying contacting the MS in step 2, the FP shall answer to the PP according to the provisions of clause 7.6.2.6, "Notification of failure".

7.6.2.2.4 Step 4-PP and FP gets the current file from the downloading server

Step 4 is used in the case option 2 is used in step 3. The PP initiates an HTTP connection to $URL2_n$, using the "Binary content download" feature.

The entry point used on the network side by the FP shall always be $FP\_URL2_n = URL2_n$. The FP is almost transparent and also uses HTTP toward the network. More specifically, the FP:

- may use HTTP GET partial request toward the network (as the PP does toward the FP when using the "Binary content download" feature);
- may adjust the "Range" and "Content-Range" headers to its own memory and data rate constraints.

After downloading the file, the PP shall use the downloaded file as appropriate (e.g. install it, and possibly install the whole new software version if this is the last file), and go back to step 1 again (until the process ends in step 3, option 1, when the FP sends a "Handset version available" command with no URL2 value)

Media type: The PP shall use "Accept: application/octet-stream" header in the request. The only media type used by the server in the "Content-Type" header shall be "application/octet-stream".

NOTE: In the context of the HTTP protocol, media types are used in the "Accept" (from client) and "Content-Type" (from server) header fields. Media types allow content type negotiation and content type notification.

Data transport. See 'Use of data connection' in clause 7.6.2.1.4.

Error handling: In case of failure of a GET request during the software image download in step 4:

- If an HTTP error is received from the server, the FP shall forward it as is to the PP.
- If the download was interrupted for any reason, the PP may use additional partial GET requests to retrieve the whole file (see clause 7.6.1.4, "File downloading resumption" and clause A.1.7.3).
- If the download cannot occur and be completed immediately after the "DelayMinutes" value timed out, the PP shall follow the provisions of clause 7.5.5.2.2 about the "DelayMinutes" parameter.

7.6.2.3 Enhanced SUOTA protocol steps

Enhanced SUOTA is introduced in clause 7.6.2.1.2, "SUOTA general description" and defined in the present clause. Enhanced SUOTA re-uses 7.6.2.2, "Basic SUOTA protocol steps" as a prerequisite, but allows some variations. Although the set of protocol enhancements usable with Enhanced SUOTA is not limited, the principles described in the present 7.6.2.3 clause shall be respected.

In particular, use of Enhanced SUOTA for the PP upgrade shall always be decided by the PP in step 1, or equivalently by the MS in step 2, but never by the FP itself (although it may be **initiated** by the FP in step 1 in some cases, see clause 7.6.2.3.1).

NOTE 1: Use of enhanced SUOTA by the PP/MS can only be successful if the FP implements the needed enhanced features.

NOTE 2: Use of Enhanced SUOTA is illustrated with 4 examples in annex C.

7.6.2.3.1 Step 1- Enhanced SUOTA possible variants

The provisions of clause 7.6.2.2.1 (Step 1) apply with the following modifications.

When using Enhanced SUOTA, URL1 received from the PP in step 1 shall be used in any case as an entry point to the MS in step 2.

Enhanced SUOTA initiation in step 1. Enhanced SUOTA may be initiated in step 1 using one of the following methods:

- Use of an enhanced feature may be initiated by the PP using security requirements for URL1 in step 1 (see clause 7.6.2.4, "PP security requirements in URL1 and URL2" and clause C.1 for an example).
- Use of an enhanced feature may be **initiated** by the FP in step 1, using URL1 as a known identifier triggering the use of some enhanced features (hence Enhanced SUOTA), although URL1 as such does not necessarily reflect this (i.e. does not contain any security requirement). This method can only be used if the FP knows the PP and MS, and therefore knows that the PP and MS require these features to be used.

NOTE: For instance, the FP could use HTTP POST method to send data to URL1, instead of the HTTP GET method, or initiate an HTTPS session to URL1, etc. See clauses C.3 and C.4 for examples.

7.6.2.3.2 Step 2-Enhanced SUOTA possible variants

The provisions of clause 7.6.2.2.2 (Step 2) apply with the following modifications.

Enhanced SUOTA initiation in step 2: Enhanced SUOTA may be initiated in step 2 using the following method:

- Use of an enhanced feature may be initiated by the MS in step 2 upon reception of a request to URL1 or to a URL1-based request, if this is a dynamically discoverable feature (e.g. RFC 2817 [i.20] describes a way for a server to initiate a TLS session upon receiving a plain HTTP request, see clause C.2).

As stated above, with Enhanced SUOTA, URL1 received from the PP in step 1 shall be used in any case as an entry point to the MS by the FP in step 2. However, if Enhanced SUOTA is used, and in contrast to when Basic SUOTA is used:

- The protocol used to retrieve the download information need not be HTTP.
- The FP to MS interface need **not** be the one described in annex B (whereas Basic SUOTA uses annex B). The way needed input parameters are provided to the MS is out of scope of the present document.
- The download information could be retrieved by the FP from the MS in one or several steps, whereas Basic SUOTA retrieves all information in one step (in the first use of step 2).
- Whether download information is retrieved locally or remotely in the n^{th} use of step 2 is out of scope of the present document (whereas the FP using Basic SUOTA retrieves all information remotely in the 1^{st} use of step 2, and locally afterwards).

NOTE: In any case, the file url for the n^{th} file (FP_URL2_n) should be retrieved by the FP for the n^{th} use of step 2 at the latest.

7.6.2.3.3 Step 3-Enhanced SUOTA possible variants

The provisions of clause 7.6.2.2.3 (Step 3) apply with the following modifications.

$URL2_n$ value determination (option 2). If Step 3 option 2 is used (see clause 7.6.2.2.3), and in contrast to Basic SUOTA, $URL2_n$ sent to the PP may be different from FP_ $URL2_n$ received from the MS.

In any case, the FP shall still keep FP_URL2_n and use it in step 4.

More specifically, $URL2_n$ may be chosen by the FP as a transient "self-pointing URL".

NOTE: A self-pointing URL points to the FP itself (acting as a local server) and uses "localhost" as its host name (see clause A.1.3.1).

7.6.2.3.4 Step 4-Enhanced SUOTA possible variants

The provisions of clause 7.6.2.2.4 (Step 4) apply with the following modifications.

When using Enhanced SUOTA:

- The "Binary content download" feature is still used between the PP and the FP for transferring the next file to the PP. The PP initiates an HTTP connection to $URL2_n$, using the "Binary content download" feature.

- However, $URL2_n$ sent by the FP to the PP could point to an entry point located on the FP itself, the FP acting as a local server (the FP hides FP_URL2_n from the PP).

- The FP still uses FP_URL2_n received from the MS in step 2 to retrieve the file from the DS.

NOTE 1: FP_URL2_n in scenario 2 when Enhanced SUOTA is used could in fact be any address from any protocol-dependent addressing scheme (it could even not be an URL).

NOTE 2: The PP may be aware of the use of Enhanced SUOTA, if URL2 received from the FP is local.

Error handling: In case of failure of a GET request during the software image download in step 4, and as an addition to the error cases already handled in clause 7.6.2.2.4:

- If an error is received from the server (potentially non-HTTP based), the FP shall use the most appropriate HTTP error toward the PP.

7.6.2.4 PP security requirements in URL1 and URL2

The present procedure describes an enhancement of clause 7.6.2.2, "SUOTA protocol steps", allowing the PP to require security functions to be used by the FP when retrieving software upgrade information, and when actually downloading the software image.

NOTE 1: Step 2 and step 4 of clause 7.6.2.2 already allow the FP to use alternative mechanisms (possibly involving security) even if the PP does not require anything.

NOTE 2: The present procedure is illustrated in clause C.1 (Enhanced SUOTA example initiated by the PP).

A PP or FP **implementing** the present procedure shall implement at least one of the procedures described in "URI-based PP to FP security requirements" of clause 7.6.1.5. Furthermore, the provisions of clause 7.6.1.5 shall also apply to URL1 sent via the C-plane.

A PP **using** the present procedure shall, as an enhancement to step 1 (clause 7.6.2.2.1), add one or more security requirement(s) to URL1 sent in the "URL indication" commands following the "Handset version indication" command.

The PP shall check whether the FP supports all of these requirements before using them. If the FP does not implement all of the security functions required in a given URL1 value, the PP shall not use this value and shall use another URL1 value it holds. A PP shall always hold at least one URL1 value with no attached security requirement.

A FP implementing the present procedure shall, when receiving URL1 with security requirements, perform the following actions if applicable:

- as an enhancement to step 2 (clause 7.6.2.2.2), make use of all of these security functions when collecting software upgrade information from the management server. This only applies if the FP does not ignore URL1 as specified in step 2 (clause 7.6.2.2.2);

- as an enhancement to step 3 (clause 7.6.2.2.3), transmit FP_URL2 = URL2 with included security requirements. This only applies if the FP choose scenario 1 (clause 7.6.2.2.4.1).

NOTE 3: The management server, inferring that the FP implements theses functions may send FP_URL2 with the same security functions.

7.6.2.5 Final notification of success and multiple upgrade SUOTA

As specified in clause 7.6.2.2.1, "Step 1-PP sends a "Handset version indication command to the FP", a successful software upgrade shall be considered as an external event triggering a "SUOTA attempt" on PP side.

The corresponding "Handset version indication" command shall contain the new "Software version identifier" (of the just installed software version), and a "fileNumber" parameter reset to "1". This allows:

- the PP to systematically notify the MS of a successful software upgrade;
- the MS to possibly initiate a new upgrade in the case of multiple upgrade SUOTA.

NOTE: A multiple upgrade SUOTA could be needed if a direct upgrade to the target software version is not possible or has not been tested.

7.6.2.6 Notification of failure

In case the PP is unable to download or install file "n", it shall send a "Handset version indication" command for the purpose of notifying the failure to the FP and MS.

- The "reason" field of the "Handset version indication" command shall be set to a non-zero value, as specified in clause 7.5.5.2.1, "*Handset version indication* command".
- The "fileNumber" field (= "n") shall be interpreted as the number of the file, for which "download" or "application" (i.e. intended use, for example: installation, reading of contained parameters, etc.) failed.
- The "Software version identifier" field shall still refer to the currently installed version of the software, not to the version for which a failure occurred (which is not installed). The MS can identify the failing version from the download information it sends in response to this value.

When using **Basic SUOTA**, the FP shall in turn notify the MS a specified in clause B.1, setting the reason parameter in FP_URL1 to the selected non-zero value (default value when absent is "0").

When using **Enhanced SUOTA**, the way the FP notifies the MS, if it does, is out of scope of the present document.

MS to FP answer to the notification of failure. Whatever kind of SUOTA is used, in response to a failure notification, the MS:

- may re-send download information (with a possibly updated "DelayMinutes" value);
- should re-send download information with updated "DelayMinutes" value if the "reason" field value was "Unable to download in time-New DelayMinutes requested";
- may indicate "No url" (see **"No url" (option 1)** in clause 7.6.2.2.3, for details), thus taking into account the failure.

FP to PP answer to the notification of failure. The FP shall reply to the PP according to the MS answer with either option 1, 2 or 3 of clause 7.6.2.2.3.

- Special case of option 2. In case the MS re-sends download information, the FP shall answer the PP as usual in a "Handset version indication" command:
  - with the url for the n^{th} file, where "n" was the "fileNumber" field value in the "Handset version indication" command notifying failure; and
  - with the new "DelayMinutes" value received from the MS.

Data transport. See 'Use of data connection' in clause 7.6.2.1.4.

7.6.2.7 User initiated SUOTA flag

User initiated SUOTA flag. The 'User initiated SUOTA' flag of the "Handset version indication" command shall indicate (when set) that a real user originates the SUOTA upgrade attempt (e.g. triggering SUOTA through a dedicated menu).

NOTE 1: Access to this service need not be granted to the end user (e.g. could be reserved for the after-sales service).

Transmission of the information to the MS. When using **Basic SUOTA**, the FP shall in turn notify the MS of the information, as specified in clause B.1, setting the "UIS" parameter in FP_URL1 to the suitable value (i.e. either '0'H or '1'H; default value when the UIS parameter is absent is "0").

When receiving this information, the MS uses it depending on its upgrade policy, which is out of scope of the present document.

When using **Enhanced SUOTA**, the way the FP notifies the MS, if it does, is out of scope of the present document.

NOTE 2: Notifying the MS enables a better user experience: for a User initiated SUOTA, the MS could in return grant a "0" value for the "DelayMinutes" parameter in the "Handset version available" response.

7.6.3 HTTP-based applications

7.6.3.1 HTTP-based applications general requirements

The "HTTP-based applications" feature allows easy implementation of user interface based applications, thanks to a rendering engine located in the PP. The application code and data are remotely hosted, either in the FP, or in an HTTP server, or both (see figure 16).

NOTE 1: The "HTTP-based applications" feature does not allow local interactions between the application and the PP local APIs.

There are two different XHTML profiles; the Simple XHTML profile (see clause 7.6.3.5) and the Baseline XHTML profile (see clause 7.6.3.6).

The user interface is XHTML based. DECT handset shall implement the "Simple XHTML profile" of clause 7.6.3.5 and may implement the "Baseline XHTML profile" of clause 7.6.3.6. Remote applications designed for use on DECT handsets implementing one of these profiles shall be DECT specific and only use the subset of XHTML defined there. The Baseline profile is defined for ensuring interoperability but remains optional for the PP.

Implementation of the "Extended HTTP profile" of clause A.2 (including the "Common HTTP Profile" of clause A.1) is a pre-requisite for a PP implementing the "Baseline XHTML profile".

NOTE 2: A FP may also implement server-side "HTTP-based applications" compliant applications. In that case requirements applying to a DECT specific server apply to the FP.

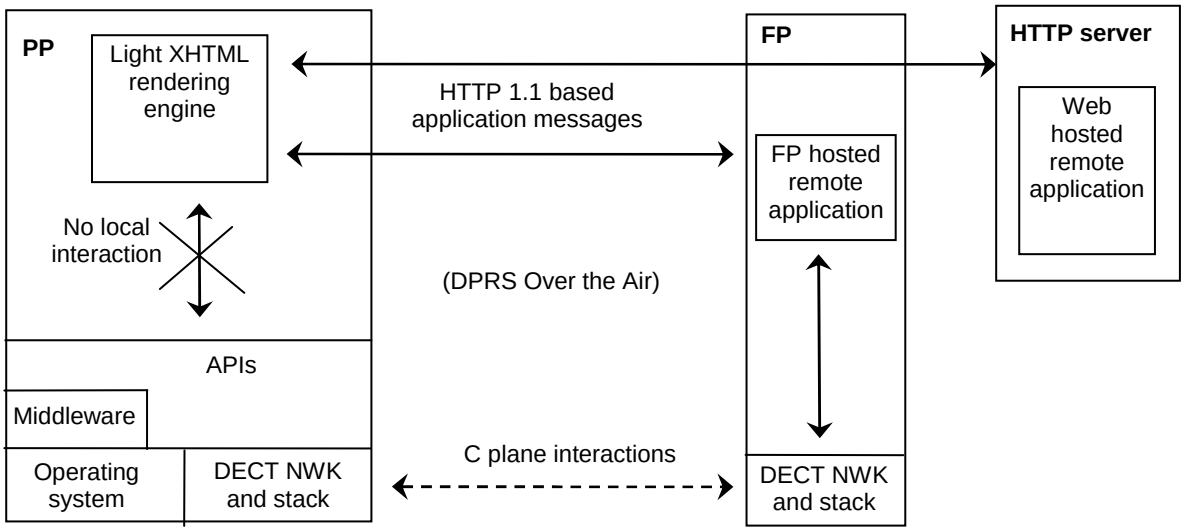


Figure 16: Remote application using the "HTTP-based application" feature

NOTE 3: An "HTTP based application" based application remote tier (unlike that of a "Binary content download" based application) works on any PP implementing the used XHTML subset (i.e. does not need to be hosted by the PP vendor).

NOTE 4: The remote tier of an "HTTP based application" based application (unlike that of a "Binary content download" based application) may be located on the FP.

7.6.3.2 Support of additional HTTP header fields

In addition to the uses of HTTP header fields defined in clause A.1.4, the PP shall support the following header fields, or new uses of these fields.

Table 26: Additional HTTP header fields

Name (note 1)	Direction	PP status	FP status (note 2)	Normative action/comment
Accept	PP to FP	O (note 3)	C3601	Additional use for describing the type of audio, image files supported. See clause 7.6.3.3.
Accept-Charset	PP to FP	O (note 4)	C3601	To indicate support of UTF-8, iso-8859-1 and possibly additional charsets (see note 5).
User-Agent	PP to FP	M	C3601	Possible values for version Part4 version 1.2.1: - NG-DECT-Part4/1.2.1 (A.1; NGLDS-A.3) - NG-DECT-Part4/1.2.1 (A.1; A.2; NGLDS-A.3) - NG-DECT-Part4/1.2.1 (A.1; A.2; NGLDS-A.3; 7.6.3.6) See also clause A.1.4.
C3601: IF the FP plays the server role THEN M ELSE N/A				
NOTE 1: Because NGLDS-A.3 (HTTP based applications) implies at least support of the 'Common HTTP profile', the headers of table A.1 (clause A.1.4) shall be supported *in addition* to the headers listed here. If a header is listed in both tables, the information of the present table supersedes that of table A.1.				
NOTE 2: As indicated by C3601 conditional, a FP not being the targeted server of the HTTP request shall only forward PP requests to the server, and shall in particular transmit header fields received from the PP to the server without removing nor altering them.				
NOTE 3: If present, the 'Accept' header field shall at least contain the value 'utf-8'. If absent, support of 'utf-8' by the PT is anyway required (see clause 7.6.3.3).				
NOTE 4: If present, the 'Accept-Charset' header field shall at least contain the value 'application/xhtml+xml'. If absent, support of 'application/xhtml+xml' by the PT is anyway required (see clause 7.6.3.4).				
NOTE 5: HTTP uses the phrase "character set" (abbreviated as "charset"), whereas "character encoding" would be more appropriate.				

7.6.3.3 Support of additional media-types

When the PP requests a media file of a given media type (e.g. "image", "audio", etc) with the GET method, the PP should use the "Accept" header in order to indicate to the server its preferred subtype(s) for the given media type. The server then indicates the actual subtype used in the "Content-Type" header of the response.

NOTE 1: In the context of a binary content download (see clause 7.6.1), an image or audio file may be sent with the "application/octet-stream" media type.

NOTE 2: Absence of the "Accept" header means that the PP accepts *all* media subtypes for the given media subtype.

7.6.3.4 Support of character encodings

For text based resources that need to be displayed on the screen (text or html file), the media type itself is not enough to fully determine the interpretation of the received media.

The PP shall at least support UTF-8 [23] character encoding formats.

NOTE 1: Support of UTF-8 only means support of the UTF-8 encoding structure at a minimum. For more information see annex E.

The PP may support ISO/IEC 8859-1 [i.10], which is equivalent to support Unicode [i.6] code points from U+0000 to U+00FF.

NOTE 2: However, for code points between U+0080 and U+00FF, UTF-8 encoding differs from ISO/IEC 8859-1 [i.10] encoding.

When requesting a text based resource, the PP should notify the server of the accepted character encodings, using the "Accept-Charset" header. The PP may support other encoding formats (ISO/IEC 8859-2 [i.11], ISO/IEC 8859-15 [i.12], UTF-16, etc), that may be easier to use or more concise for some languages. In that case, it should also use the "Accept-Charset" header to indicate it.

The server should indicate-when sending a document to the PP-the used character encoding, using the "charset" parameter of the "Content-Type" header field. The PP shall use this information in order to correctly display the content to the user. In case the charset parameter is absent, the PP shall assume that the document is UTF-8 encoded.

7.6.3.5 Simple XHTML profile

The PP shall at least support the following XHTML1.1 [22] modules. Text nodes shall use UTF-8 [23] encoding.

NOTE: The "Simple XHTML profile" only requires partial implementation of some modules, as indicated in the "Comment" column.

Table 27: Supported XHTML modules in Simple XHTML profile

Module	Element	Comment
Structure	body, head, html, title	Complete
Text	br, p	Partial
Hypertext	a	Partial (href attribute only)
Style Attribute Module	blabla	Used in
Presentation	b	Partial

The PP shall understand all listed tags. However, the present document does not enforce the way the information is presented to the user. The PP should follow the tag definition whenever possible, and as much as possible, but may use alternative ways to render the tag in case of display or other constraints. For example, the PP shall understand (but not necessarily follow) the "<b>" tag for bold face presentation: if the PP does not hold the corresponding bold face font, it could render it with quotes, or not render it at all.

7.6.3.6 Baseline XHTML profile

The PP shall at least support the following XHTML1.1 [22] modules:

NOTE: The "Baseline XHTML profile" only requires partial implementation of some modules, as indicated in the "Comment" column.

7.6.3.6.1 Basic elements support

Table 28: Supported XHTML modules in Baseline XHTML profile

Module	Element	Comment
Structure	body, head, html, title	Complete
Text	br, h1, h2, h3, h4, h5, h6, p	Partial
Hypertext	a	Complete
Style Attribute Module	blabla	Used in
List	ol, ul, li	Partial. The "value" attribute on "li" element restarts numbering
Presentation	b, i	Partial
Base	base	Complete

7.6.3.6.2 Image element support

The "img" tag shall be recognized even if the PP is not able to display images. In that case it may use the "alt" (alternate) attribute of this tag in order to display a text replacement for the image.

Table 29: Supported XHTML modules, image elements

Module	Element	Comment
Image	img	Complete

7.6.3.6.3 Tables support

Table 30: Supported XHTML modules, basic tables

Module	Element	Comment
Basic Tables	table, td, tr	Partial

7.6.3.6.4 Forms support

Table 31: Supported XHTML modules, forms

Module	Element	Comment
Forms	form, input, select, option, textarea	Partial

Consistently with clause A.2 ("Extended HTTP profile"), a <form> element shall have an enctype field with value "multipart/form-data"

7.7 Interworking requirements

NOTE: See also clause 7.6.1.2 on LU10 interworking conventions and HTTP profile.

7.7.1 IWU-attributes information element

The procedures as specified in ETSI EN 301 649, [15], clauses B.1 to B.4 and B.6 shall apply with the following differences:

- Only Frame relay (FREL) service and Interworking type Generic media encapsulation ([15], clause B.8) shall be used.
- The coding of the IWU-ATTRIBUTES information element ([15], clause B.2) shall be as follows.

Table 32: IWU-ATTRIBUTES information element support status

Supported parameters					
Field no.	**Name of fields**	**Ref.**	**Support**	**Values**	
				Allowed	**Supported**
1	ID of IWU attributes of variable length		M	18	
2	Length of Contents (L)		M	0 to 255	any
3	Coding standard		M	1	1
3	Profile		M	0	0
4	Negotiation indicator		M	0,2,4,6	0,2
4	Profile subtype	B.2.1	M	0 to 15	8
5, 5a	Maximum SDU size (PT => FT or both ways)	B.2.1	M	0 to 16 383 (equivalent to 0 to 131 064 octets)	191 to 16 383 (equivalent to 1 528 to 131 064 octets)
5b, 5c	Maximum SDU size (FT => PT, optional)	B.2.1	O	0 to 16 383 (equivalent to 0 to 131 064 octets)	1 528 to 16 383 (equivalent to 12 224 to 131 064 octets)
6	Application protocol control set	B.2.1.1.3	M	All values for Generic Encapsulation Interworking	All values for Generic Encapsulation Interworking
NOTE: See ETSI EN 301 649 [15], clauses B.2 and B.8.					

7.7.2 SDU sizes and setting of SDU boundaries

The chopping facility defined in DPRS Generic Encapsulation Interworking (see DPRS [15], clause B.8.2) may be optionally implemented to split large application packets into smaller SDUs. This chopping facility, when supported shall be announced by means of the IE <SETUP CAPABILITY> (see DPRS [15], clause 12.22), and shall be specifically invoked at context creation using the flag described in DPRS [15], clause B.2.1.1.4.1. This facility shall only be invoked when supported by both sides.

When the chopping facility is used, the size of the application packet segments (including the D-GMEP header), shall be set as the maximum supported SDU size announced at <<IWU ATTRIBUTES>>, except the last (or the only one) segment that may be of smaller size.

In order to use the chopping facility, the features DPRS-N.33 (Dynamic Parameters Allocation) and DPRS-N.35 (Service Change) shall be supported.

Unless the chopping facility is in use, the SDUs shall be equal to an application packet plus the D-GMEP header (see DPRS [15], clause B.8.2).

7.8 Physical layer procedures

No differences/additions - the procedures as specified in ETSI EN 301 649 [15], clause 5 shall apply.

Annex A (normative): HTTP Profiles

A.1 Common HTTP profile (HTTP limited set nr.2)

NOTE: The "Common HTTP profile" is the HTTP limited set nr. 2 listed in DPRS (ETSI EN 301 649 [15], clause B.8.3.4). The name "Common HTTP profile" is used locally in the present document.

A.1.1 General requirements

The "Common HTTP profile" defined in clause A.1 represents the minimum HTTP profile that a New Generation DECT, part 4 PPs implementing HTTP based features shall implement. This profile is based on HTTP version 1.1 [19].

NOTE: The "Common HTTP profile" is a pre-requisite for:

- The **"Binary content download"** feature (clause 7.6.1).
- The **"Software upgrade over the air (SUOTA)"** feature (clause 7.6.2).
- The **"HTTP-based applications"** feature (clause 7.6.3).
- The **"Extended HTTP profile"** (clause A.2).

Use of the present annex relies on the interworking conventions defined in ETSI EN 301 649 [15] (DPRS), clause B.8.3.1.

A.1.2 Supported HTTP methods

A.1.2.1 GET method

The PP and FP shall only use HTTP unconditional partial GET requests.

NOTE: HTTP 1.1 [19] describes conditional and unconditional GET requests. Partial GET requests may be conditional or unconditional.

A GET request in the context of the present "Common HTTP profile", being always "partial", shall systematically include a "Range" header field. See clause A.1.7 for the use of a "Range" header field.

As requested by version 1.1 of HTTP, such a GET request shall always contain a "Host" header field.

A.1.2.2 HEAD method

The PP and FP should support the HTTP HEAD method, which may be useful for debugging purposes.

A.1.2.3 POST method

When both sides only support the present "Common HTTP profile", the POST method shall not be used by the PP.

NOTE: DPRS [15], HTTP limited set nr. 1 (clause B.8.3.3) and HTTP limited set nr. 3 (clause B.8.3.5) both require support of the POST method. This is however not supported within HTTP limited set nr. 2 (Common HTTP profile) described in clause A.1.

A.1.2.4 Pipelining of requests

The PP shall not use pipelining of idempotent requests (i.e. GET, HEAD, etc, but not POST) as described in HTTP 1.1 [19], section 8.1.2.2. As a result, non transparent FPs need not support the handling of pipelined requests.

NOTE: This restriction does not apply to requests from distinct flows or applications, i.e. handled with separate contexts as described in DPRS clause B.8 (unless the PP or FP only supports a single context). Such flows would use separate TCP connections on the network side anyway.

A.1.3 Request URI and Host header field

Request URI: The request URI is the URI reference included in the request line of an HTTP request.

EXAMPLE 1: The request line "**`GET`** `/path/file.bin` **`HTTP/1.1`**" uses the URI reference "`/path/file.bin`" as "request URI". This URI reference is a relative URI reference (i.e. does not start with "http") but uses an absolute path (i.e. it starts with "/").

EXAMPLE 2: The request line "**`GET`** `http://suota.example.com/path/file.bin` **`HTTP/1.1`**" uses the URI reference "`http://suota.example.com/path/file.bin`" as "request URI". This URI reference is a full URI. Such a request URI is allowed by HTTP 1.1 but not by the "Common HTTP profile" in general.

NOTE 1: RFC 3986 [i.19] (section 4.2) defines a URI reference as either a URI or a URI relative reference. This RFC obsoletes RFC 2396 however referred to by HTTP 1.1. The "request URI" defined by HTTP 1.1 should therefore be called a "request URI reference".

Mandatory 'Host' header. A "Common HTTP profile" compliant HTTP request shall always contain a "Host" header field, as required by HTTP 1.1 (see examples below).

Mandatory 'request URI' form. A "Common HTTP profile" compliant request URI sent over the DECT air interface shall be either:

1. (Short form) an absolute path URI relative reference, as in the following example:

EXAMPLE 3:
```
GET /path/file.bin HTTP/1.1
Host: content-download.example.com
```

An absolute path URI relative reference shall always start with the slash character ("/"). In particular, it cannot be empty and shall at least contain the slash character.

2. (Long form) a full URI. A full URI shall only be used when the short form cannot be used, i.e. in the following cases:

   - when the URI contains a confidentiality requirement (see clause 7.6.1.5.1), as in the following example:

EXAMPLE 4: **GET** https://suota.example.com/path/file.bin **HTTP/1.1**.
```
Host: suota.example.com
```

   - when the URI contains an authentication requirement (see clause 7.6.1.5.2), as in the following example:

EXAMPLE 5:
```
GET http://pp1:mypassword@suota.example.com/path/file.bin HTTP/1.1
Host: suota.example.com
```

When a full URI is used as Request URI (as shown in examples 4 and 5 above), the host part of that URI and the "Host" header field value shall be identical.

NOTE 2: An HTTP request sent to a proxy should contain a full http URI. If the FP is behind a proxy, the FP may have to modify the request URI before forwarding the request to the proxy. A full http URI can be constructed from an absolute path URI relative reference using the mandatory "Host" header field value.

A.1.3.1 Use of the 'localhost' Host value

When the "Host" header field value "localhost" is sent over the DECT air interface, it shall refer to the FP.

Example of Request URIs targeting the FP:

```
GET /FPservice1/config.txt HTTP/1.1
Host: localhost
```

A.1.4 Supported HTTP header fields

The HTTP header fields in table A.1 shall be supported by the FP and/or PP.

Table A.1: Supported HTTP header fields

Name	Direction	PP status	FP status (notes 1, 2)	Normative action/comment
Host	PP to FP	M	M	Presence M in all (partial) GET requests
Accept	PP to FP	O	CA101	Absence means support of all media subtypes
Content-Type	FP to PP	M	CA101	
Content-Length	FP to PP	M	CA101	
Location	FP to PP	M	CA101	See A.1.6.
User-Agent	PP to FP	M	CA101	This header shall include the following substring: "NG-DECT-Part-4/<part 4 version> (comment)" The comment shall include a semi-colon separated list of optional clauses or features that are implemented.
Byte-range operations related headers (see also clause A.1.7)				
Range	PP to FP	M	CA101	Presence M in all (partial) GET requests
Content-Range	FP to PP	M	CA101	Presence M in successful GET responses (206 Partial Content)
CA101: IF the FP plays the server role (see clause A.1.3.1) THEN M ELSE N/A.				
NOTE 1: As indicated by CA101 conditional, a FP not being the targeted server of the HTTP request shall only forward PP requests to the server. Such a FP does not need to support header fields, except for the 'Host' header field from the PP in order to locate the server, and shall in particular transmit header fields received from the PP to the server without removing nor altering them. NOTE 2: Before forwarding the response to the PP, a FP not being the targeted server of the HTTP request should perform a broad parsing of the response header: - in order to filter out header fields NOTlisted above (if sent by the server), as recommended in the following section ('Use of HTTP header fields not listed above'); - but without removing nor altering header fields listed above (as indicated by value 'N/A' of CA101 conditional above).				

Use of HTTP header fields not listed above. The sending of http header fields not listed above (especially from FP to PP) may significantly affect the overall transfer rate over the DECT air interface (i.e. the ratio of HTTP header size to HTTP payload size could easily go beyond 100 % or more). As a consequence, it is recommended to avoid sending HTTP header fields not listed in table A.1 in order to guarantee an optimized transfer.

EXAMPLE 1: The sending from FP to PP of HTTP header fields such as 'Server', 'Date', 'Last-Modified', 'ETag', etc., should be avoided.

The PP shall ignore a header that it does not understand. It shall not reject the response as a result.

NOTE: Clause 7.6.3.2 defines an amended value for PPs implementing the "HTTP-based applications" feature.

EXAMPLE 2: **Examples of product token and related comment**: a PP implementing New Generation DECT, part 4 version 1.2.1 shall include one of the following strings in the User-Agent header:
- NG-DECT-Part4/1.2.1 (A.1)
- NG-DECT-Part4/1.2.1 (A.1; A.2)
- NG-DECT-Part4/1.2.1 (A.1; NGLDS-A.3)
- NG-DECT-Part4/1.2.1 (A.1; A.2; NGLDS-A.3)
- NG-DECT-Part4/1.2.1 (A.1; A.2; NGLDS-A.3; 7.6.3.6).

A.1.5 Supported media types

At least media type = "application/octet-stream" shall be supported (i.e. requested/accepted) as content type by both PP and FP.

NOTE: In the context of the HTTP protocol, media types are used in the "Accept" (from client) and "Content-Type" (from server) header fields. Media types allow content type negotiation and content type notification.

A.1.6 Redirections of GET (or HEAD) requests

In the context of clause A.1, a redirection received from the server in an HTTP response indicates that the client shall issue a new HTTP request in order to get the requested resource. A redirection is characterized by an HTTP status code of the 3xx class in the HTTP response. The url for the new request is indicated in the "Location" header of the response.

NOTE 1: In general, the exact behaviour of the PP upon redirection depends on the status code received. See HTTP 1.1 [19], section 10.3 for details.

NOTE 2: Redirections of POST requests are possible. However, clause A.1 restricts itself to GET (and HEAD) requests. Redirections of GET (or HEAD) requests may be handled by the PP without user interaction.

NOTE 3: For the redirections of POST requests, see clause A.2.

The PP shall be able to handle redirections of GET requests with the following status codes:

- 301, "Moved Permanently". If subsequent requests to the same initial url are planned, the PP may use the redirection target url ("Location" header value) instead of the initial url.

- 302, "Found". If subsequent requests to the same initial url are planned, the PP shall however still use the initial url (e.g. for the case the redirection target would change in between).

- 307, "Temporary Redirect". In the context of a GET (or HEAD) request, the PP behaviour shall be identical as when 302 is used.

NOTE 4: The behaviour of the PP upon 301, 302, or 307 redirections is similar, except in case subsequent requests to the same initial url are planned.

The PP shall therefore also support the "Location" header of the corresponding HTTP response.

NOTE 5: For the redirections of POST requests, the PP should also be able to handle the 303 status code.

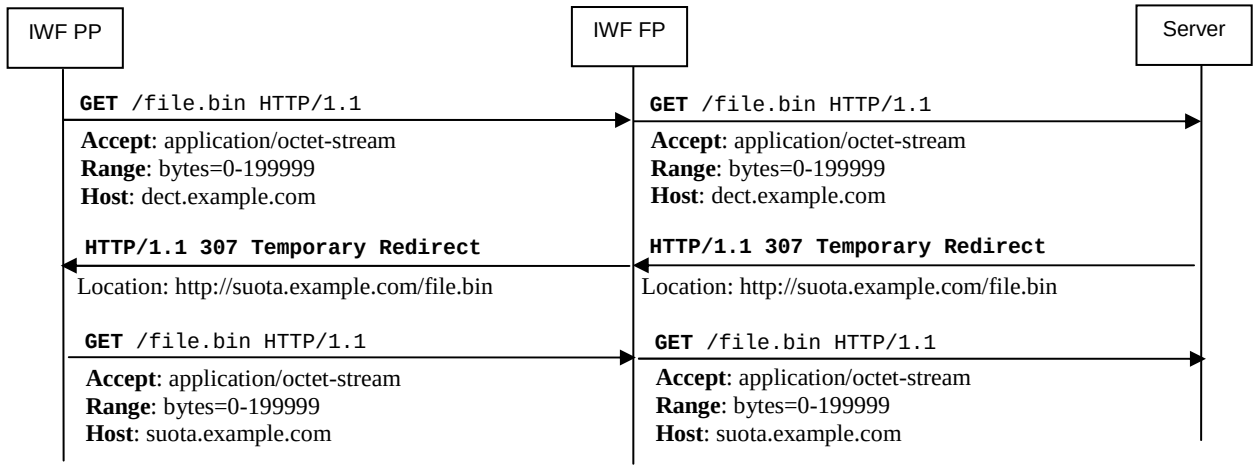


Figure A.1: Example of redirection handled by the PP

NOTE 6: A redirection in general may also change the name of the file to be retrieved. This is however not recommended for the "Binary Content download" feature (file oriented feature).

The FP shall not handle redirections on behalf of the PP. See also figure A.1.

A.1.7 Byte-range operations

Resource: A file available over the Internet at a location specified by a URI.

Byte-range: A connex subset (interval) of the requested resource considered as a series of bytes. More specifically, a byte range is defined as a sub-interval of [0, (L-1)], where "L" is the total length of the resource.

HTTP 1.1 byte-range mechanism shall be used systematically by the PP in order to:

- limit the number of bytes received as a response to a single HTTP request;
- resume the downloading of a resource following a broken connection.

As a restriction to HTTP 1.1 a PP shall only request a single range of bytes in any byte range request.

NOTE: For low-memory FPs, the HTTP server hosting DECT specific content to be downloaded should also implement range requests.

The PP and the FP shall implement the following subset of HTTP 1.1 byte range operations.

A.1.7.1 Byte-range operations related responses

206 Partial Content: (RFC 2616 [19], section 10.2.7). This is the response for a successful byte range request (instead of "200 OK").

NOTE 1: A response with status code 206 should not include a Content-Range header with "*" as the response range specifier, as in: `Content-Range: byte */200000.`

NOTE 2: The `Content-Range` header contains in principle the resource file size. As a consequence, the PP may evaluate and check the needed memory space for downloading the resource before performing the actual download and upgrade.

416 Requested Range Not Satisfiable: (RFC 2616 [19], section 10.4.17). This is to be used when none of the requested ranges has a non-empty intersection with the targeted resource.

NOTE 3: A 416 response should include a Content-Range header one of the following forms:
`Content-Range: byte */L`, where L is the total length of the resource.
`Content-Range: byte */*`, if the length of the resource is unknown.

A.1.7.2 Byte-range operations related header fields

HTTP 1.1 defines the following range request related headers.

Range (from PP to FP): This is the header used by a PP to make a byte-range request. As a restriction to HTTP 1.1 "Range" header format, the PP shall always request a single range at most.

NOTE 1: The use of this header by the FP to retrieve resources from the network is out of the scope of the present specification and is not affected by this restriction. The FP should also use this mechanism toward the network in order to cope with its own limited byte rate or memory resources.

Content-Range (from FP to PP): This header shall be used by the FP to specify the actual range of bytes sent as a response to a byte-range request.

NOTE 2: The actual content range may be smaller than the requested one.

NOTE 3: HTTP 1.1 requires that a Content-Length header be also sent along with the Content-Range header (redundant information).

Accept-Ranges (from FP to PP): If present from FP to PP, this header shall always have the value "bytes" (default value), indicating to the PP that range requests are accepted. It shall never have the value "none".

NOTE 4: For low-memory FPs, the HTTP server hosting DECT specific content to be downloaded should implement range requests for all available resources and never use the "Accept-Range: none" value.

If-Range: Not supported.

A.1.7.3 Byte range operation use cases

This clause describes the HTTP message flows for various use cases.

NOTE: For the sake of simplicity:

- DECT messages are left out. Each following "GET" message can be part of the same or of a newly established DECT connection.
- FP to servers exchanges conform to the scenario described in clause B.3 (almost transparent FP).

A.1.7.3.1 Use case 1: standard downloading with default application packet size of 12 kBytes

In the following use case, an image file of 120 kBytes is downloaded using ten HTTP GET commands with HTTP packet size set at the default value of 12 kBytes by means of the RANGE parameter. See table A.2 and figure A.2.

Table A.2: Use case 1 sizes

Size of file to be downloaded	120 000 bytes
Maximum HTTP packet size over air i/f	12 000 bytes
Maximum HTTP packet size over network i/f	12 000 bytes

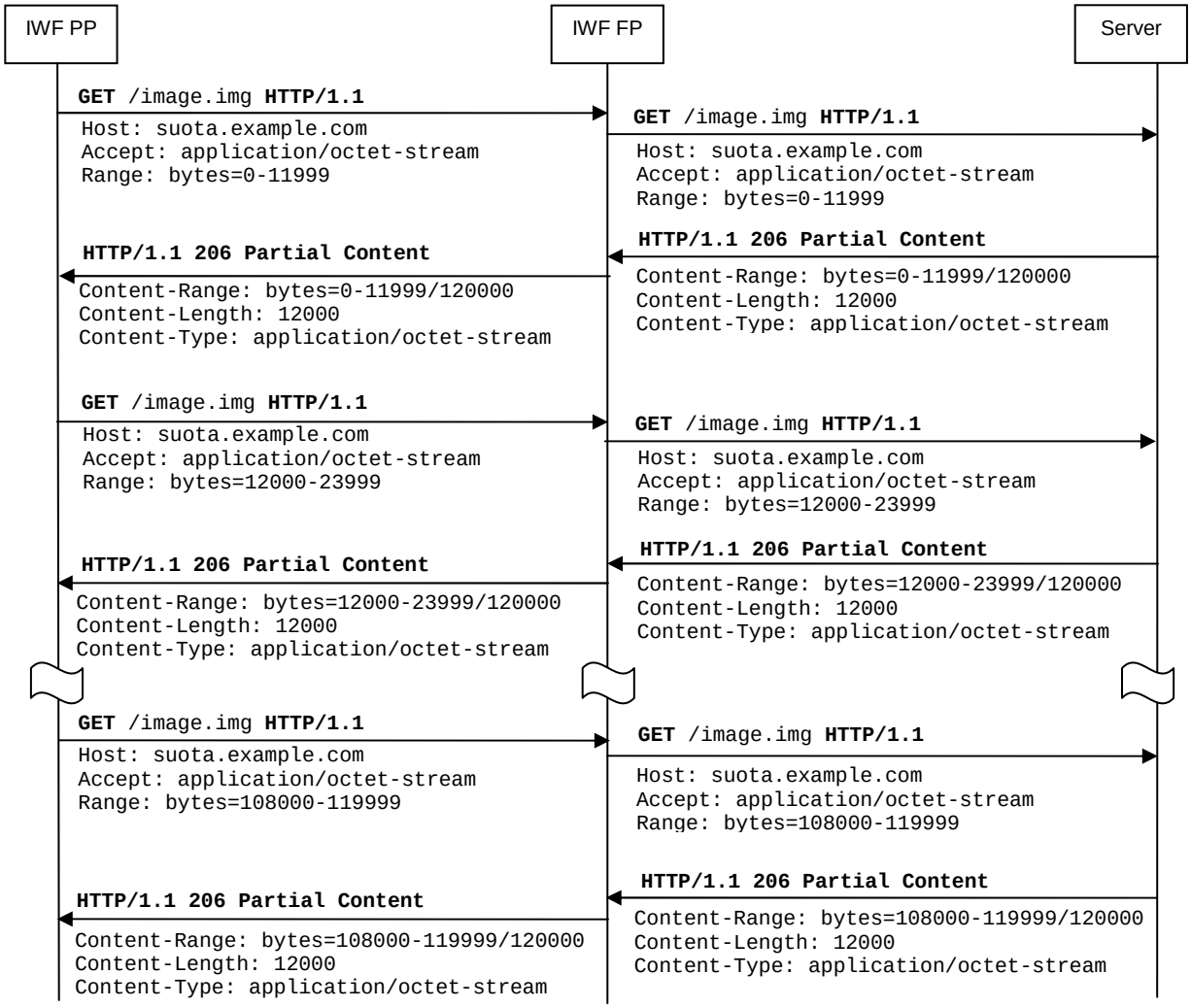


Figure A.2: Use case 1: standard downloading with default application packet size of 12 kBytes

A.1.7.3.2 Use case 2: standard downloading with application packet size of 48 kBytes

In this use case, both peers have announced the support of a maximum application packet size of 48 kBytes (or possibly larger) in the <<SETUP CAPABILITY>> IE, by means of DPRS-N.33 "Dynamic parameters allocation".

The same image file of 120 kBytes is downloaded using three HTTP GET requests with a range width of 48 kBytes. See table A.3 and figure A.3.

Table A.3: Use case 2 sizes

Size of file to be downloaded	120 000 bytes
Maximum HTTP packet size over air i/f	48 000 bytes
Maximum HTTP packet size over network i/f	48 000 bytes

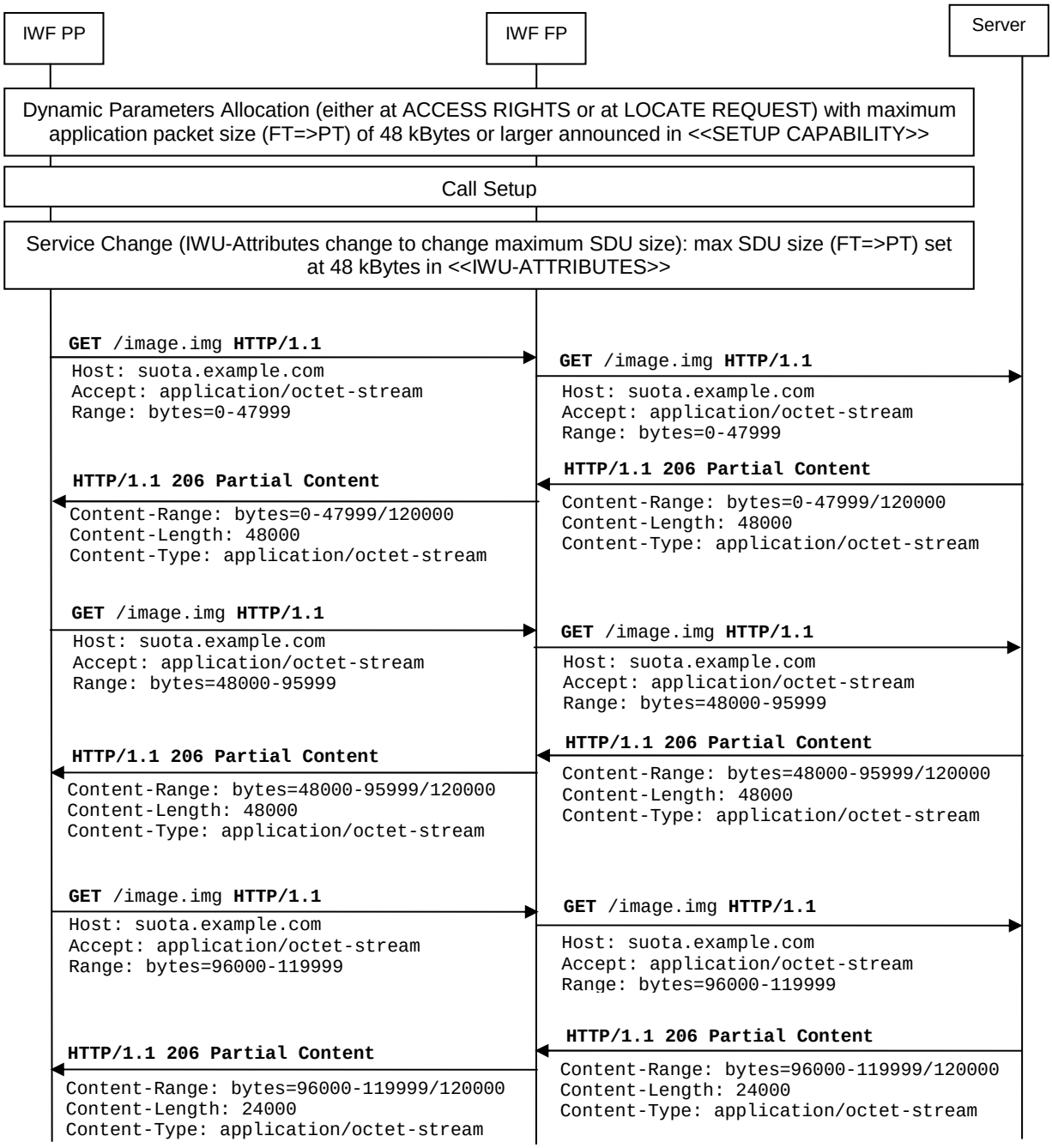


Figure A.3: Use case 2: standard downloading with application packet size of 48 kBytes

A.1.7.3.3 Use case 3: Download with interruption in-between

In this use case, a connection break down makes it necessary for the PP to issue the interrupted GET request again (see also "file downloading resumption" in clause 7.6.1.4). See table A.4 and figure A.4.

Table A.4: Use case 3 sizes

Size of file to be downloaded	120 000 bytes
Maximum HTTP packet size air i/f	12 000 bytes
Maximum HTTP packet size over network i/f	12 000 bytes

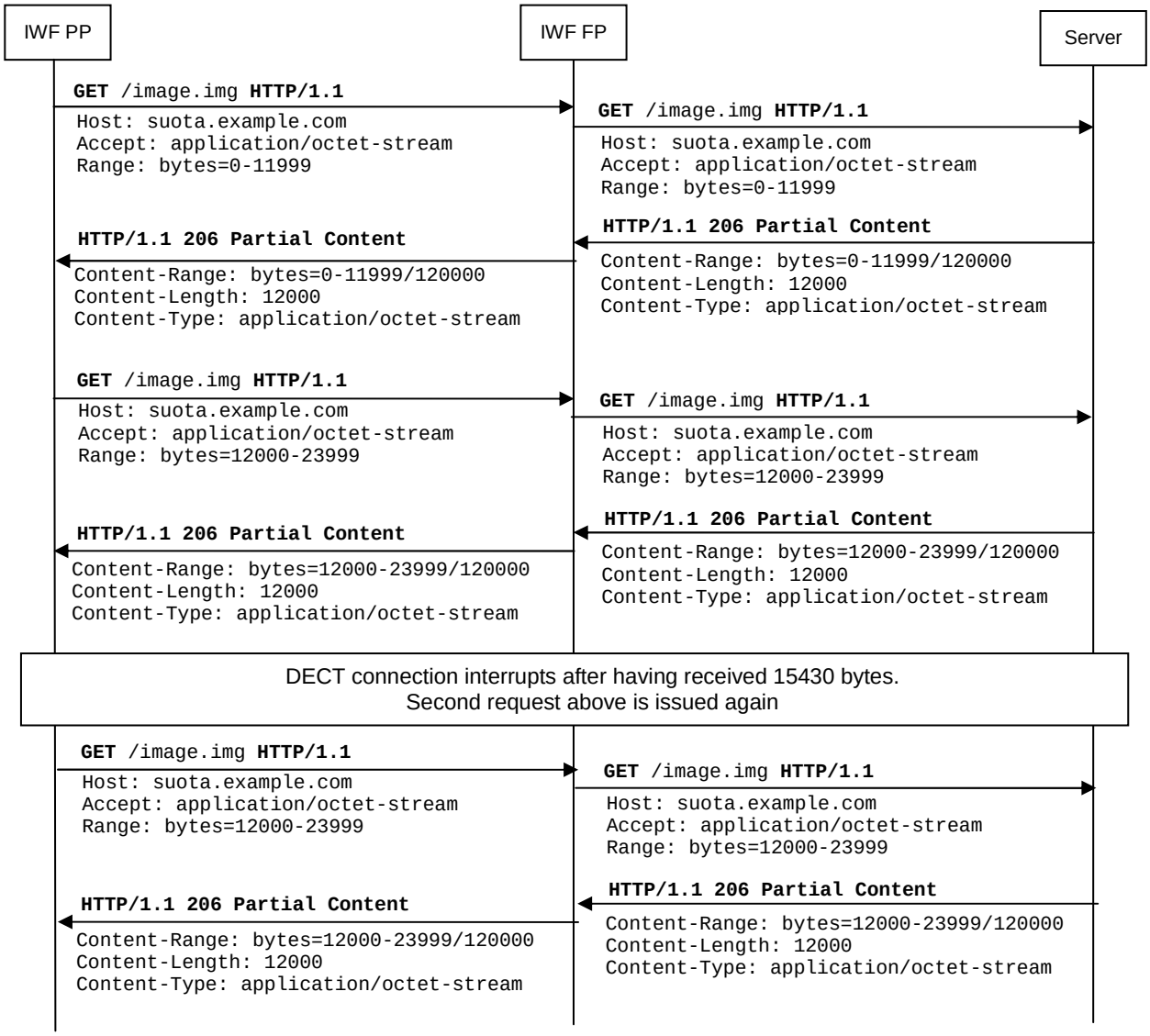


Figure A.4: Use case 3: Download with interruption in-between

A.1.8 Supported HTTP errors

The PP shall be prepared to receive HTTP status codes related to client errors (of the form 4XX) or server errors (of the form 5XX) and shall not crash as a result. See table A.5 below.

NOTE 1: Both client or server error codes are originating from the server as a response to a request. The server is supposed to know which side the error comes from.

NOTE 2: Redirection (3XX) status codes to be supported are documented in clauses A.1.6 and A.2.2.2 (for PPs implementing clause A.2).

NOTE 3: See RFC 2616 [19], section 10, "Status Code Definitions" for a list of status codes. Note however that this RFC does not list all existing codes. See for example RFC 2817 [i.20], section 7.1.

Table A.5: Most frequently used HTTP error codes

Status line	Comment (or RFC 2616 [19] extracts)
HTTP/1.1 500 Internal Server Error	General purpose server internal error
HTTP/1.1 404 Not found	Domain not found by the DNS Resource or file not found is the server (but could be available later) May be used instead of 403 Forbidden
HTTP/1.1 403 Forbidden	The server understood the request, but is refusing to fulfill it. Authorization will not help and the request SHOULD NOT be repeated
HTTP/1.1 400 Bad Request	The request could not be understood by the server due to malformed syntax. The client SHOULD NOT repeat the request without modifications
HTTP/1.1 401 Unauthorized	User has to provide credentials OR user provided credentials were wrong

A.2 Extended HTTP profile (HTTP limited set nr.3)

NOTE 1: The "Extended HTTP profile" is the HTTP limited set nr. 3 listed in DPRS (ETSI EN 301 649 [15], clause B.8.3.5). The name "Common HTTP profile" is used locally in the present document.

The "Extended HTTP profile" defined in the present annex defines a strict superset of the "Common HTTP profile" defined in clause A.1 and is designed for applications requiring higher interactivity between the user and the application.

NOTE 2: The "Common HTTP profile" limits user inputs to the server to URLs values. The "Extended HTTP profile" allows to send parameters or other data to the server.

NOTE 3: The "Extended HTTP profile" is referenced in:

- The **"Binary content download"** feature (see clause 7.6.1.6).
- The **"HTTP-based applications"** feature (7.6.3).

A.2.1 General requirements

A PP and FP implementing the "Extended HTTP profile" shall implement the "Common HTTP profile" as a pre-requisite.

The "Extended HTTP profile" implies implementation of the POST method with the "Post Redirect Get" pattern" (clause A.2.2.2.2).

A FP implementing the "Extended profile" shall be able:

- to support the reception of a POST request of at least 1 500 bytes from the PP;
- to forward such a POST request (received from the PP) to the server;
- to forward the server response to the PP.

A PP or FP implementing the "Extended HTTP profile" shall be able to support a POST response of at least 1 024 bytes.

NOTE: The server should either respect this limit for the response, or redirect the POST request as defined in clause A.2.2.2.2.

A.2.2 POST method

This clause describes the use of HTTP 1.1 POST method within the "Extended HTTP profile". This method is provided in addition to the partial GET method (see clause A.1) for allowing PP to FP sending of data (i.e. not limited to the sending GET requests).

A PP implementing the POST method feature shall support the POST method with media type "multipart/form-data".

NOTE 1: The POST method is most notably useful for sending HTML completed forms (see also clause 7.6.3.6.3) to the server. However, the user interface for sending user entered data may be any type of user interface and need not be HTML forms.

NOTE 2: The "multipart/form-data" format is known to properly handle character encodings, contrary to GET "query string" method(s) or related "x-www-form-urlencoded" POST method, for sending completed form data.

A.2.2.1 Example of POST method

An example of POST method is shown in figure A.5.

Example of form sending	**Comments**
POST /dect/service1 HTTP/1.1‘0d0a’H **Accept**: text/html‘0d0a’H **Content-Type**: multipart/form-data; boundary=----7d81‘0d0a’H **Host**: example.dect.com‘0d0a’H **Content-Length**: xxx‘0d0a’H ‘0d0a’H ----7d81‘0d0a’H **Content-Disposition**: **form-data**; name="Firstname"‘0d0a’H **Content-Type**: text/plain;charset=UTF-8‘0d0a’H ‘0d0a’H ‘c389’Hric‘0d0a’H ----7d81‘0d0a’H **Content-Disposition**: **form-data**; name="Lastname"‘0d0a’H **Content-Type**: text/plain;charset=UTF-8‘0d0a’H ‘0d0a’H Leb‘c593’Huf‘0d0a’H ----7d81--‘0d0a’H	‘0d0a’H = carriage return + line feed (always) The used boundary between form entries is defined here A ‘blank line’ shall be present here A ‘blank line’ shall be present here ‘Éric’ was entered on a UTF-8 compliant device A ‘blank line’ shall be present here ‘Lebœuf’ was entered on a UTF-8 compliant device The last boundary line shall end with two hyphens (‘--’)

Figure A.5: Example of POST method

NOTE 1: Considering the POST request as a series of bytes, the following notation has been used in the above example:

- bytes corresponding to printable US-ASCII characters are represented by the corresponding character;
- hexadecimal notation has been used for non-ASCII characters, and for ASCII but non-printable characters (i.e. carriage return character '0D'H and line feed character '0A'H);
- white spaces within request and header lines always stand for the single byte '20'H (no other encoding should be used for white spaces at these places).

NOTE 2: A POST request can also send binary data-or data considered as binary data-to the server (e.g. a file), although this is not shown in the above example.

A.2.2.2 Redirection of POST requests

A.2.2.2.1 General requirements

As an addition to clause A.1.6, the present clause handles the redirection of POST methods.

The PP shall be able to handle redirections of POST requests with the following status codes:

- 302, "Found". In the context of a POST request, the PP behaviour shall be identical to as when 303 is used.

- 303, "See Other". The redirection target url ("Location" header value) shall be requested with a **GET** request (and not POST). If subsequent requests to the same initial url are planned, the PP shall however still use the initial url.

- 307, "Temporary Redirect". If subsequent requests to the same initial url are planned, the PP shall however still use the initial url (e.g. for the case the redirection target would change in between).

A.2.2.2.2 Post-Redirect-Get pattern

The Post-Redirect-Get pattern is known to allow adequate user experience when sending form data to a server.

In the case of a DECT PP, it also allows the use of a partial GET request by the PP (as described in clause A.1.7) to retrieve the POST response.

Use of this pattern is initiated by the server which redirects the client instead of sending the POST response directly. The client retrieves the response by issuing a subsequent GET (partial) request as the result of the redirection.

The server should either use this pattern to handle POST requests, or respect the limit defined in clause A.2.1 for the size of the response. See also figure A.6.

NOTE: The most adequate http redirection code to be used in this case is "303 See Other".

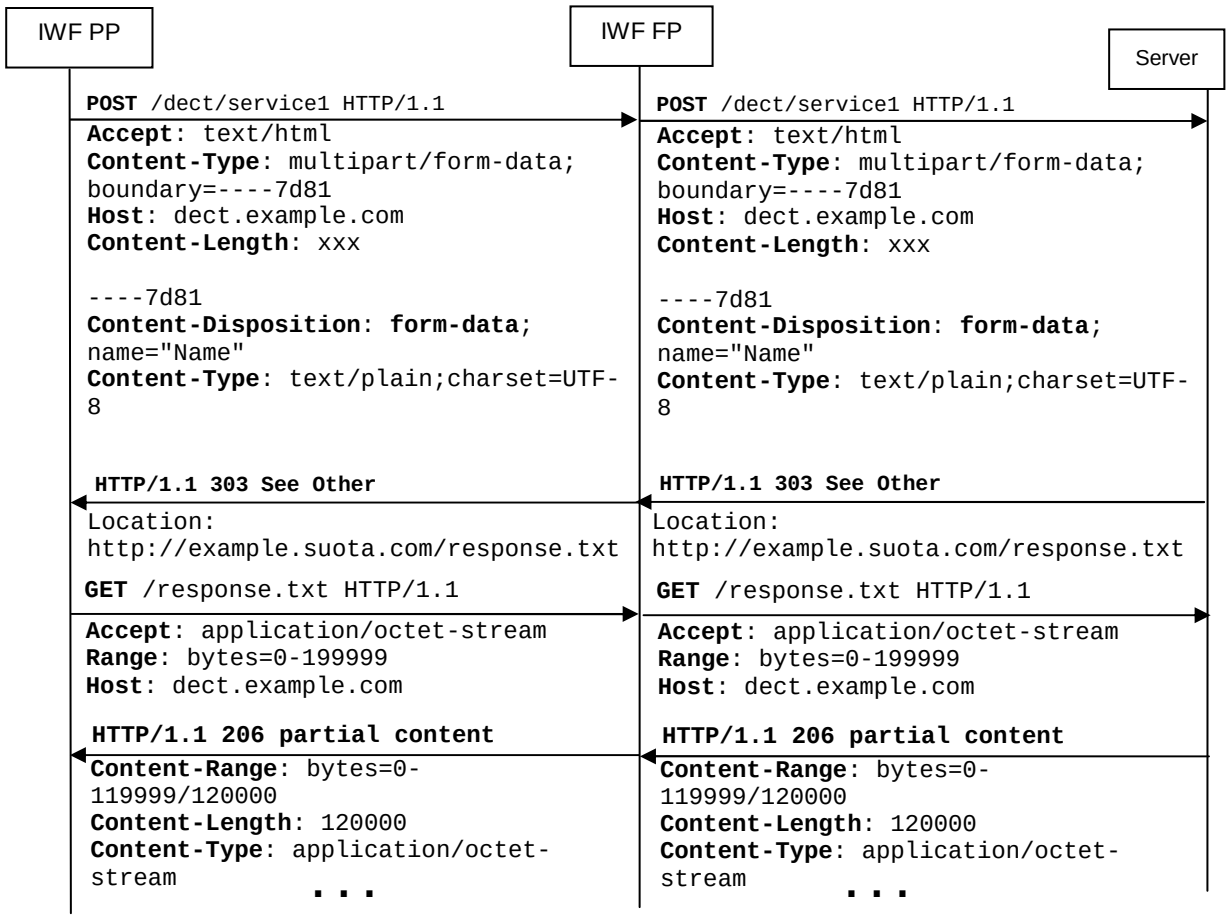


Figure A.6: Post-Redirect-Get pattern

A.2.3 Supported HTTP header fields

The supported HTTP headed fields are shown in table A.6.

Table A.6: Supported HTTP header fields

Name (note 1)	Direction	PP status	FP status	Normative action/comment
User-Agent	PT to FT	M	CA201 (note 2)	The value for Part4 version 1.2.1 shall at least indicate support of A.1, A.2, as in: - NG-DECT-Part4/1.2.1 (A.1; A.2) See also clause A.1.4 for the list of possible User-Agent values containing A1, A2.
CA201: IF the FT plays the server role THEN M ELSE N/A.				
NOTE 1: Because the "Extended HTTP profile" defines a strict superset of the "Common HTTP profile", the headers of table A.1 (clause A.1.4) shall be supported *in addition* to the headers listed above. If a header is listed in both tables, the information of the present table supersedes that of table A.1.				
NOTE 2: As indicated by CA201 conditional, a FP not being the targeted server of the HTTP request shall only forward PP requests to the server and shall in particular transmit the User-Agent header field to the server without removing nor altering it.				

Annex B (normative): Basic SUOTA

Clauses B.1 and B.2 describe the Basic SUOTA interface to the MS, used by the FP. This interface shall be implemented by the FP and shall be implemented by NG-DECT Part4 compliant management servers.

NOTE 1: Basic SUOTA is defined end-to-end up to the MS and DS. This allows to handle (at least) the SUOTA "pure" cross-vendor case, involving a PP and FP from different vendors, and for which the PP is unknown to the FP.

NOTE 2: The use of this interface is described in clause 7.6.2.2.2, "Step 2- FP retrieves downloading information from management server".

NOTE 3: This interface could also be used for software/firmware upgrade of the FP itself.

Clause B.3 describes a software upgrade example using Basic SUOTA.

NOTE 4: Although annex B is normative, clause B.3 only describes an example implementation and is therefore informative.

B.1 Basic SUOTA FP to management server interface

The purpose of the "Basic SUOTA FP to management server interface" described in the present clause is to allow the FP to request the files urls needed for a given software upgrade (see step 2, clause 7.6.2.2.2).

FP_URL1. Name of the FP request to the management server. For the "Basic SUOTA FP to management server interface", this request takes the form of a URL with parameters in the "query string".

The FP requests all needed file urls at once (i.e. in a single FP_URL1 request). File urls are supposed to be delivered by the management server in the order they are needed by the PP for the actual software upgrade, from 1 to N_f.

B.1.1 FP request (FP_URL1) construction

In order to construct FP_URL1, the FP shall append the following strings to URL1:

- **<linking character>**: "?" or "&" or nothing; more specifically, a question mark ("?") shall be used if the base url does not contain this character; otherwise, an ampersand "&" shall be used if "?" is not the terminating character of the base url; otherwise, the linking character shall be absent.
- "**EMC=**"<coded Equipment Manufacturer's Code value, 4 hexadecimal characters (see note)>.
- "**&SWVid=**"<coded Software Version id value, less than 40 hexadecimal characters (see note)>.
- "**&HWVid=**"<coded Hardware Version id value, less than 40 hexadecimal characters (see note)>.

NOTE: These values are copied from the "Handset version indication" command received from the PP and coded as printable **hexadecimal** strings. Leading zeros are optional.

In order to construct FP_URL1, the FP may in addition append the following strings to URL1, and **shall** do it if it is not using the default value for the corresponding parameters:

- "**&reason=**"<reason value from "Handset version indication", 1 hexadecimal character, default value '0'H>.
- "**&UIS=**"<User initiated SUOTA flag value = bit 5 of octet 8a in "Handset version indication", default value '0'H>.

If the reason field is not zero (and therefore necessarily present in FP_URL1), the FP shall in addition add the following string to URL1:

- "**&fileNumber=**"<fileNumber field value from "Handset version indication", 1 hexadecimal character>.

EXAMPLE: if URL1= "http://suota.example.com/info", FP_URL1 computed as above could be:

- "http://suota.example.com/info?EMC=01ab&SWVid=
4649524d574152452d312e322e30&HWVid=322e32,
in order to ask for download information concerning the version **next to** "FIRMWARE-1.2.0", and provided that the <EMC> and <HW Version identifier> values received in the "Handset version indication" command were respectively '01ab'H, '322E32'H.

- "http://suota.example.com/info?EMC=01ab&SWVid=4649524d574152452d312e322e30&HWVid=322e32&reason=2&fileNumber=2, to report a failure when applying file number 2 under the same conditions.

B.1.2 Private parameters

FP_URL1 may include one or more private parameter(s), if the following conditions are met:

- the FP including these parameters knows the MS and knows the MS requires or accepts these parameters;
- each parameter name is different from all of the parameters defined in clause B.1.1.

NOTE: For example, an MS specific PP and/or FP device id may be inserted for device management purposes.

B.2 Basic SUOTA management server to FP interface

A management server complying with the SUOTA DECT specific interface shall answer the FP request with an "application/xml" value for the **Content-Type** header (figure B.1).

```
<?xml version="1.0" encoding="utf-8"?>
<SUOTA>
   <!-- total size in bytes of all files (present in the list below) -->
    <SoftwareTotalSize>201927</SoftwareTotalSize>

    <!-- software version id of the file set (printable hexadecimal coded) -->
    <SoftwareVersionId>312e3131</SoftwareVersionId>

    <!-- yes or no; if yes, download should only start upon user authorisation -->
    <UserInteraction>no</UserInteraction>

    <!-- Delay suggestion (in min) before download should start
         between 0 and 65535 minutes -->
    <DelayMinutes>300</DelayMinutes>

    <!-- list of all files needed for the upgrade
         between 1 and 15 files, in the order needed for the upgrade -->
    <FileList>
       <File>http://suota.example.com/path/to/file/file1.bin</File>
       <File>http://suota.example.com/path/to/file/file1.md5sum</File>
       <File>http://suota.example.com/path/to/file/file2.bin</File>
       <File>http://suota.example.com/path/to/file/file2.md5sum</File>
    </FileList>
</SUOTA>
```

Figure B.1: Example of management server returned information following a FP request when a new software version is available

The document element shall be named "SUOTA" and shall contain the following subelements:

- **SoftwareTotalSize:** total size of the new software (added sizes of all files); decimal coded.

NOTE: If a new software version is available for the PP, there is at least one file (with non-zero size); if the PP is up to date and no new version is available, then see 'Case of an up to date PP' below.

- **SoftwareVersionId:** software version identifier of the new software; As the present document makes no provision as to the format of the "Software Version id" parameter, it shall be hexadecimal coded.

- **UserInteraction:** parameter indicating that the user authorization should be received before the download starts (value: "yes" or "no", without the quotes).

- **DelayMinutes:** (in minutes, from 0 to 65 535). This value is to be used for the first file of the list (As the PP downloads files sequentially, the next files will have most of the time a DelayMinutes parameter of "0". See clause 7.5.5.2.2).

- **FileList:** List of file urls, each of which being included in a sub "File" element.

  - File urls shall be listed in the order they are needed by the PP for the actual software upgrade. In case several orders are acceptable for the PP, the MS shall nevertheless always use the same order.

The MS shall respect the order of the elements specified above, and shall not add extra information, such as attributes or elements not specified above.

The FP shall respect the order of the urls in the list, i.e. shall send them in "Handset Version available" commands in the same order.

Case of an up to date PP. If there is NO new software version available (i.e. PP is up to date) the same XML elements shall be used, but with the following contents (figure B.2):

SoftwareTotalSize: The total size value shall be '0'.

SoftwareVersionId: the XML element shall be present and empty.

UserInteraction: the XML element shall be present and empty.

DelayMinutes: the XML element shall be present and empty.

FileList: the XML element shall be present and empty (i.e. the <File> subelement shall not be present.

```
<?xml version="1.0" encoding="utf-8"?>
<SUOTA>
   <!–XML answer in the case of NO new version available -->
    <SoftwareTotalSize>0</SoftwareTotalSize>
    <SoftwareVersionId/>
    <UserInteraction/>
    <DelayMinutes/>
    <FileList/>
</SUOTA>
```

Figure B.2: Management server returned information when No new version is available

B.3 Basic SUOTA possible implementation (example)

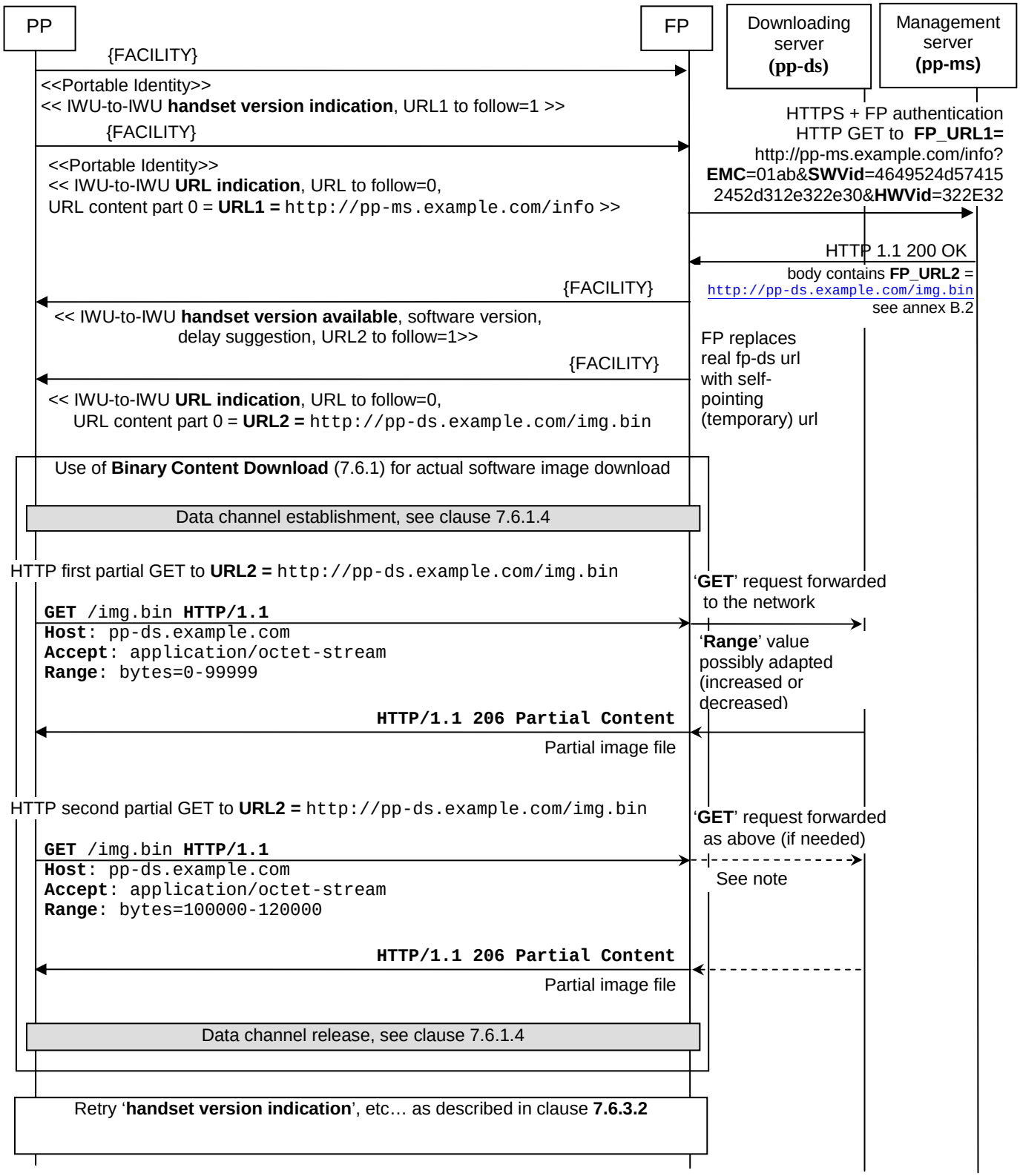
This clause details a possible implementation of Basic SUOTA as described in clause 7.6.2.2 (see also figure B.3).

Thanks to the information received from the PP in the "Handset version indication" and "URL indication" commands (especially URL1):

- The FP creates FP_URL1, by appending PP specific parameters (received in "Handset version indication") to URL1 (received in "URL indication"). See clause B.1.
- The FP contacts the PP-vendor management server (pp-ms) on behalf of the PP and retrieves the software upgrade information there, in the format described in clause B.2.
- The FP forwards to the PP the retrieved FP_URL2 unchanged (FP_URL2 = URL2) in the "Handset version available" command. In this example, only one file is needed for the upgrade.

The PP establishes a "Binary content download" connection with the FP and sends HTTP partial GET requests to the FP:

- The FP forwards almost transparently the PP partial GET requests to the DS.
- For a given partial GET request from the PP, the FP may retrieve a smaller or larger range of the target resource than requested by the PT, thus adjusting it to its own memory or rate constraints.



NOTE: For a given partial GET request from the PP, the FP may retrieve a smaller or larger range of the target resource than requested by the PT, thus adjusting it to its own memory capacity. In some cases this second retrieval step towards the network may therefore not be necessary.

Figure B.3: Example of Basic SUOTA possible implementation

Annex C (informative): Enhanced SUOTA

This annex illustrates 4 possible uses of Enhanced SUOTA, in clauses C.1, C.2, C.3, and C.4 (see also figures C.1, C.2, C.3 and C.4).

C.1 Enhanced SUOTA example-use of Basic/Digest authentication and HTTPS from FP to MS, initiated by the PP

Clause C.1 illustrates one Enhanced SUOTA (clause 7.6.2.3) use case: the use of two enhanced features (https + basic or digest http-based authentication from FP to MS) requested by the PP using security requirements as described in clause 7.6.3.4, "PP security requirements in URL1 and URL2".

In this example, use of security features from FP to MS implies the use of the same security features for the download of the file itself from the DS (i.e. within use of the "Binary content download" feature). This is because the MS- inferring from step 2 that the FP is capable of using security features-answers with FP_URL2 also including the same kind of security requirements.

More specifically, exchanges between the FP and the MS, and between the FP and the DS are protected in the following way:

- Confidentiality is ensured through TLS/SSL encryption between the FP and the server:
  - For software upgrade information, encryption is requested by the PP by using an https request. HTTPS itself is however only used between the FP and the management server.
  - For software image downloading, encryption is requested by the MS by including an https url in the software upgrade information. HTTPS itself is again only used between the FP and the management server.

NOTE: Encryption only relies on the server encryption private and public keys, and corresponding certificate (the FP does not need to own any encryption key). The FP should implement a TLS/SSL stack and embed/trust the public key of the server certificate authority, or of one of its ancestors.

- **PP authentication** is ensured through HTTP basic or digest authentication, operated by the FP on behalf of the PP.

The PP should entrust the FP with its own authentication password, and therefore should trust the FP.

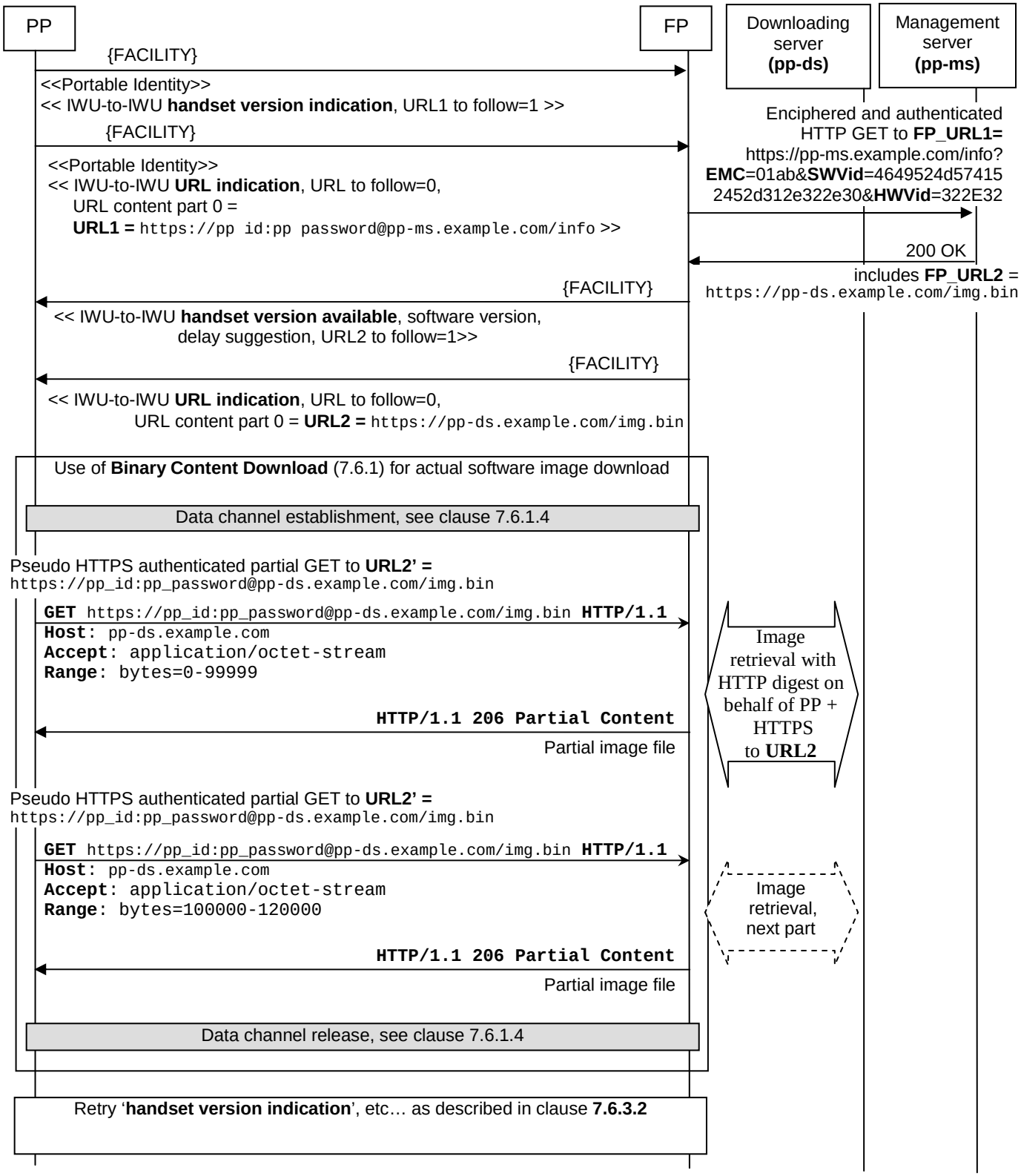


Figure C.1: Example of Enhanced SUOTA-use of Basic/Digest authentication and HTTPS from FP to MS, initiated by the PP

C.2 Enhanced SUOTA example-use of HTTPS from FP to MS, initiated by the MS

Clause C.2 illustrates one Enhanced SUOTA (see clause 7.6.2.3) use case: use of an enhanced feature (https) initiated by the MS.

Enhanced SUOTA is initiated by the MS in step 2, requiring the use of HTTPS toward the server, after receiving a plain http request (FP_URL1 = http://pp-ms.example.com/info?EMC=01ab&SWVid=312E3130&HWVid=322E3237).

Receiving a request to FP_URL1, the MS requires an upgrade to TLS. As a result, HTTPS is used for the FP to MS session, and also for the downloading of files (as in clause C.1).

NOTE: In this example, although Enhanced SUOTA is used, the MS used is still a Basic SUOTA complying MS (using clauses B.1 and B.2 but with security features).

In this example, the URL2 value sent by the FP to the PP is a temporary local URL pointing to the FP itself as described in clause 7.6.2.3.3, "Step 3-Enhanced SUOTA possible variants", the FP acting as local server. Alternatively, if the PP supports security requirements and the FP knows this, the FP could directly send a "distant" URL2 value with security requirements, as in clause C.1).

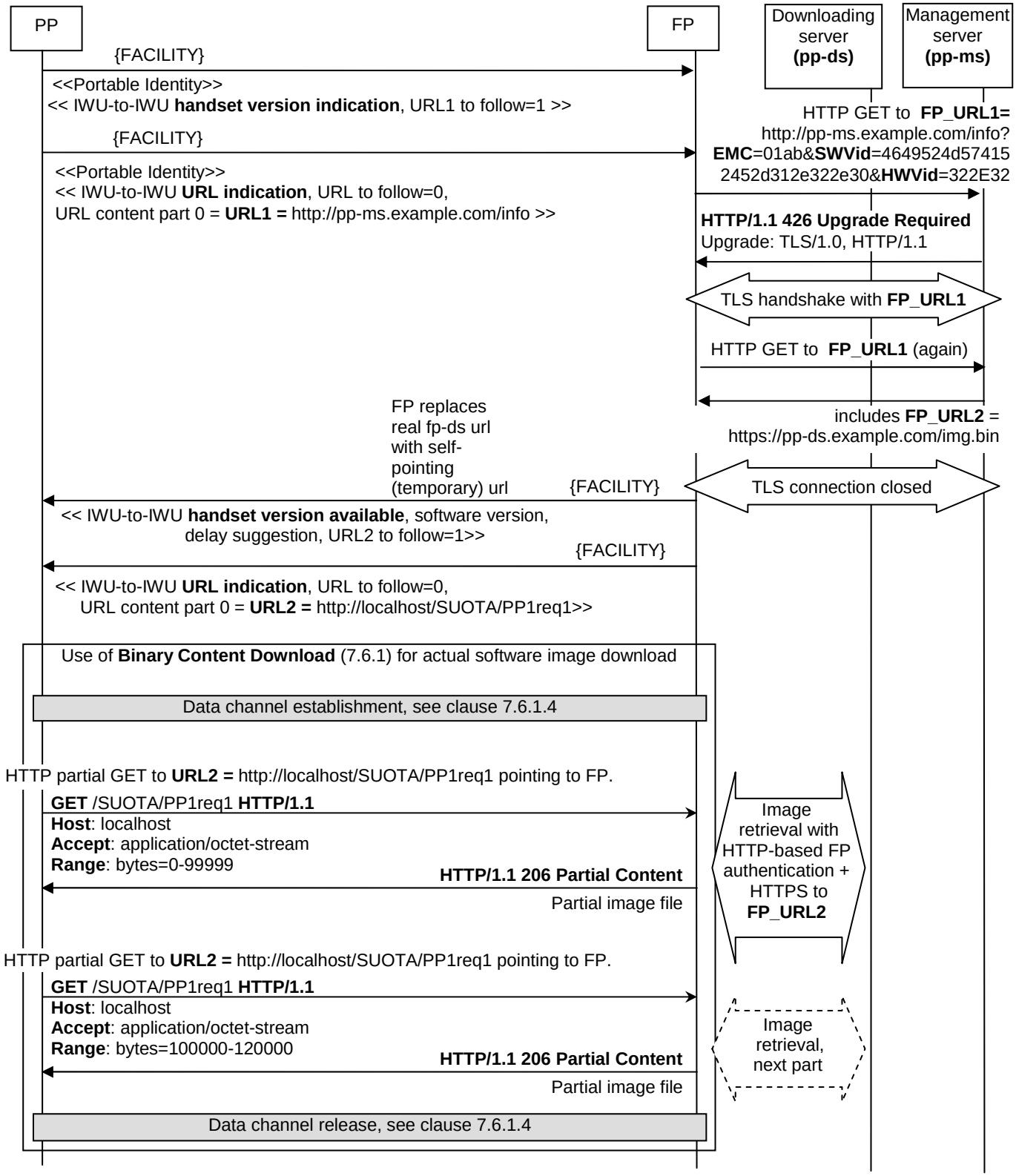


Figure C.2: Example of Enhanced SUOTA-use of HTTPS from FP to MS, initiated by the MS

C.3 Enhanced SUOTA example-use of HTTPS from FP to MS, initiated by the FP

Clause C.3 illustrates one Enhanced SUOTA (see clause 7.6.2.3) use cases: use of an enhanced feature (https) initiated by the FP.

This use case is similar to the one described in clause C.2, because the result is the same: URL1 is used, except that HTTPS is used instead of HTTP. However the way HTTPS use is initiated is different.

In the present example, Enhanced SUOTA is initiated by the FP in step 1, directly setting an HTTPS connection with the MS to pp-ms.example.com because **URL1 =** http://pp-ms.example.com/info is recognized as a known URL1 value (e.g. because pp-ms = fp-ms).

NOTE: In this example again, as in clause C.2, and although Enhanced SUOTA is used, the MS used is still a Basic SUOTA complying MS (using annexes B.1 and B.2 but with security features).

In this example again, the URL2 value sent by the FP to the PP is a temporary local URL pointing to the FP itself as described in clause 7.6.2.3.3, "Step 3-Enhanced SUOTA possible variants", the FP acting as local server. Alternatively, if the PP supports security requirements and the FP knows this, the FP could directly send a "distant" URL2 value with security requirements, as in clause C.1).

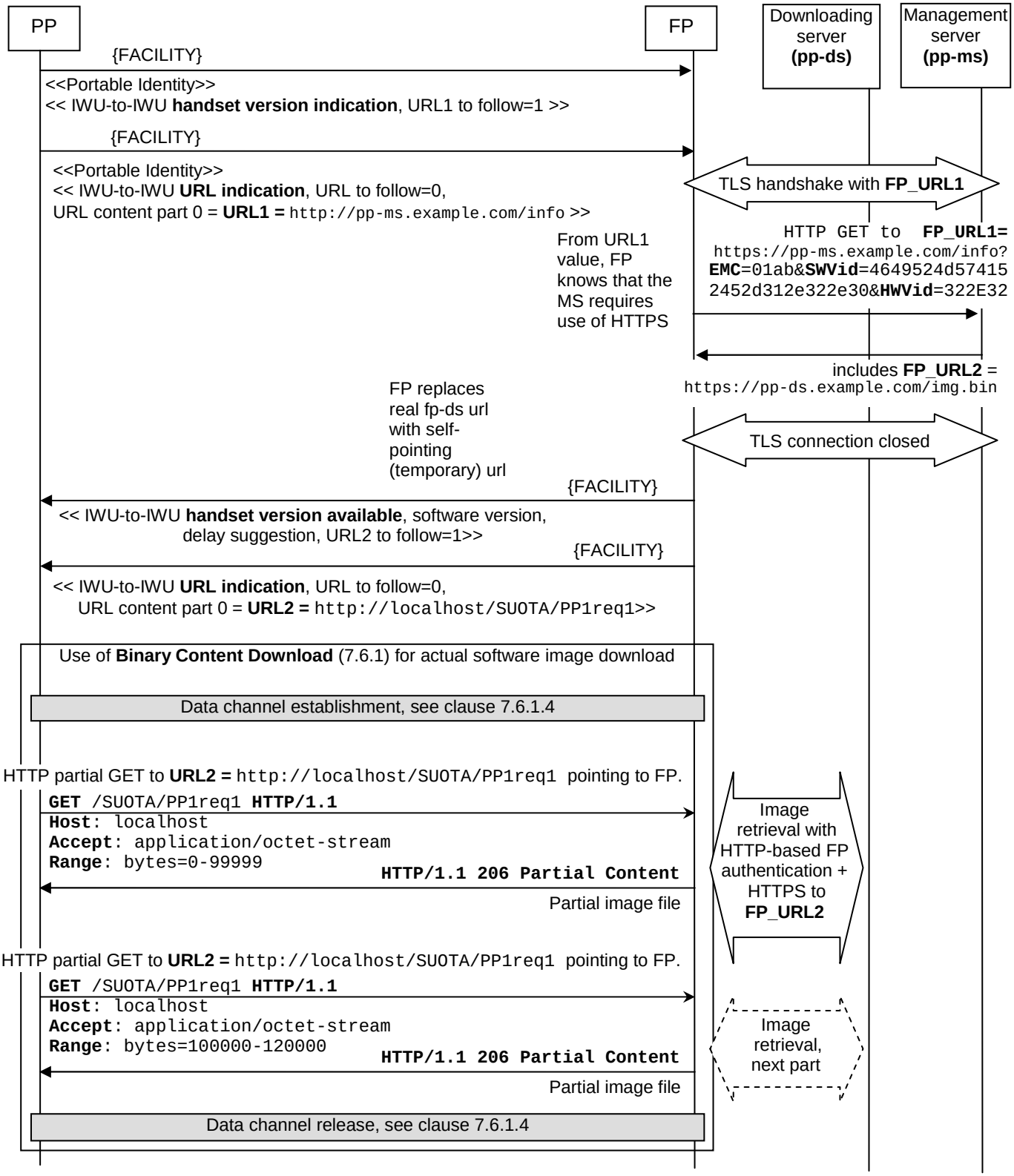


Figure C.3: Example of Enhanced SUOTA-use of HTTPS from FP to MS, initiated by the FP

C.4 Enhanced SUOTA example-use of TR-069

C.4.1 Introduction and diagram

Clause C.4 illustrates one Enhanced SUOTA (see clause 7.6.2.3) use case: the FP triggers the use of TR-069 [i.5], because it knows the MS requires it.

In the present example, the used MS is TR-069 compliant. The FP is the TR-069 client and knows the MS. It uses this MS to manage its own firmware updates, as well as the PPs firmware updates.

NOTE 1: In the context of TR-069 [i.5], the management server is called the "ACS" (Auto-Configuration Server).

The FP to MS connection is enciphered using TLS/SSL. The FP (i.e. the TR-069 client) should authenticate to the management server using either HTTP basic or digest authentication, or a SSL/TLS based strong authentication.

The FP still uses the parameters received in the "Handset version indication" message, but posts them in a TR-069 compliant message to the management server ("Inform" message).

The TR-069 compliant scenario requires the following adjustments to the four "SUOTA protocol steps" described in clauses 7.6.2.2 and 7.6.2.3:

- TR-069 requires the FP (TR-069 client) to notify the MS, for each file, of the correct download and application (e.g. installation). The present example interprets the sending of a "Handset version indication" for fileNumber="n" by the PP as an acknowledgement of correct download and application of file "n-1". This allows the FP to first send a TransferComplete message for file "n-1" in step 2.

- For the sake of clarity, message exchanges between the FP and MS in step 2 are split into two TR-069 sessions. In a real life case, "TransferComplete" message from FP to MS for file "n-1" and "Download" message from MS to FP for file "n" could be exchanged in the same session.

- This example assumes that file url number "n" is received from the MS in the n^{th} use of step 2 (not in the first use of step 2, as it would be the case with Basic SUOTA). This is however not necessarily the case: step 2 could include several download requests from the MS. For example, the MS could send all download requests in the first use of step 2, allowing the FP to get all file urls in one step as for Basic SUOTA.

NOTE 2: TR-069 allows a device to refuse some download requests, using "9004 Resources exceeded" fault message toward the server as described in clause C.4.2.3.3. In that case the MS will repeat these requests during the next connections.

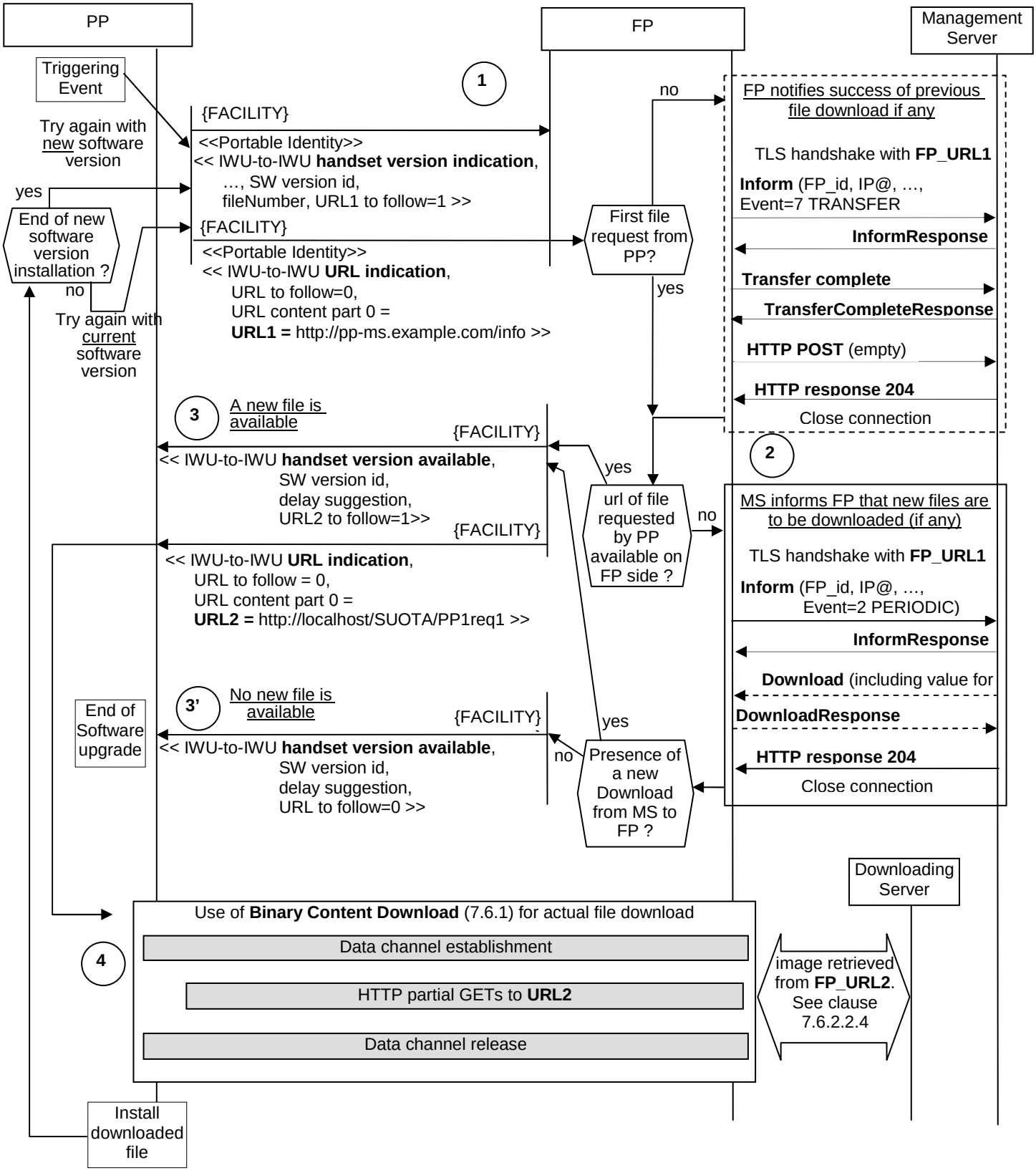


Figure C.4: Example of Enhanced SUOTA- FP as TR-069 client

C.4.2 Detailed messages

C.4.2.1 General message format

TR-069 uses a SOAP envelope encompassing each of the exchanged messages. The message itself is inserted as the soapenv:Body element content. For conciseness, the following envelope is not repeated in the message descriptions of the following clauses.

```
<soapenv:Envelope soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
                  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
                  xmlns:soap="http://schemas.xmlsoap.org/soap/encoding/"
                  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                  xmlns="urn:dslforum-org:cwmp-1-0">

    <soapenv:Header>
       <ID soapenv:mustUnderstand="1"> <!-- Unique session ID defined by the session initiator--> </ID>
       <NoMoreRequests soapenv:mustUnderstand="0">1</NoMoreRequests>
    </soapenv:Header>

    <soapenv:Body>

       <!-Message content as described in the following clauses goes here-->

    </soapenv:Body>
</soapenv:Envelope>
```

The <ID> element content is a unique ID for the TR-069 session, chosen by the session initiator, and inserted by both sides (FP and management server) in every message pertaining to this session. Thanks to this ID the other side is able to uniquely identify the session to which a received message belongs.

C.4.2.2 Preliminary "Inform" exchange

C.4.2.2.1 Inform

Assumption is made here that the same session (hence the same "Inform" message) is used for both:

- The FP acknowledging a previous successful or failed "Download", using a "TransferComplete message. This accounts for the presence of an event of type "7 TRANSFER COMPLETE" in the Inform message <Event> element. The command key value for this event should be copied from the corresponding **Download** message.
- The MS requesting a download by the device (represented by the FP), using a "Download" message. This accounts for the presence of an event of type "2 PERIODIC" in the Inform message. The command key value for this event should remain empty.

```
<Inform>
   <DeviceId> <!--FP identification-->
      <Manufacturer>SUPPLIERXY</Manufacturer>
      <OUI>XXXXXX</OUI>   <!--Organizationally Unique Identifier-->
      <ProductClass>ProductZ</ProductClass>
      <SerialNumber>SN-XYZT</SerialNumber>
   </DeviceId>
   <Event soap:arrayType="cwmp:EventStruct[2]">
      <EventStruct>
         <EventCode>2 PERIODIC</EventCode>
         <CommandKey> <!-- empty --> </CommandKey>
      </EventStruct>
      <EventStruct>   <!-- This event indicates that the transferComplete method will be tried later in the session -->
         <EventCode>7 TRANSFER COMPLETE</EventCode>
         <CommandKey> <!-- command key copied from corresponding Download message --> </CommandKey>
      </EventStruct>
   </Event>
   <MaxEnvelopes>1</MaxEnvelopes>
   <CurrentTime>2008-09-15T15:40:00Z</CurrentTime>
   <RetryCount>0</RetryCount> <!-- starts at 0-->
   <ParameterList soap:arrayType="cwmp:ParameterValueStruct[6]">
      <ParameterValueStruct>
         <Name>Device.DeviceSummary</Name>
         <Value xsi:type="xsd:string"></Value>
      </ParameterValueStruct>
      <ParameterValueStruct>
         <Name>Device.DeviceInfo.HardwareVersion</Name>
         <Value xsi:type="xsd:string"></Value>
      </ParameterValueStruct>
      <ParameterValueStruct>
         <Name>Device.DeviceInfo.SoftwareVersion</Name>
         <Value xsi:type="xsd:string"></Value>
      </ParameterValueStruct>
      <ParameterValueStruct>
         <Name>Device.ManagementServer.ConnectionRequestURL</Name>
         <Value xsi:type="xsd:string"></Value>
      </ParameterValueStruct>
      <ParameterValueStruct>
         <Name>Device.ManagementServer.ParameterKey</Name>
         <Value xsi:type="xsd:string"></Value>
      </ParameterValueStruct>
      <ParameterValueStruct>
         <Name>Device.LAN.IPAddress</Name>
         <Value xsi:type="xsd:string"></Value>
      </ParameterValueStruct>
   </ParameterList>
</Inform>
```

NOTE: In this example, the value "2 PERIODIC" is used by the FP, indicating that the "**Inform**" request belongs to a set of periodically sent "Inform" messages. This is because the Inform message is supposed to be sent following a periodically sent "Handset Version Indication" command.

C.4.2.2.2 InformResponse

InformResponse from the management server acknowledges receipt of the Inform message.

```
<InformResponse>
   <MaxEnvelopes>1</MaxEnvelopes>
</InformResponse>
```

C.4.2.3 Download exchange

C.4.2.3.1 Download (from server to FP)

The **Download** message is the request from the management server that will cause the FP to download one file.

The time at which the download takes place is described in the message with a delay in seconds from the time when the message was received. In the corresponding "Handset version indication" command, the FP should therefore use a value of <DelayMinutes> = (DelaySeconds/60) minutes.

NOTE: The sequencing of actual file downloads with TR-069 [i.5] is therefore time-based.

```
<Download>
   <CommandKey><!--server defined download instance ref.--></CommandKey>
   <FileType>1 Firmware Upgrade Image</Event>
   <URL> <!-- download target url = FP_URL2 --> </URL>
   <Username> <!-- if authentication needed --> </Username>
   <Password> <!-- if authentication needed --> </Password>
   <FileSize> <!-- file size in bytes --> </FileSize>
   <TargetFileName> <!-- in FP or PP file system --> </TargetFileName>
   <DelaySeconds> <!-- DelayMinutes = DelaySeconds/60 --> </DelaySeconds>
   <SuccessURL> <!-- optional, for browser --> </SuccessURL>
   <FailureURL> <!-- optional, for browser --> </FailureURL>
</Download>
```

C.4.2.3.2 DownloadResponse (from FP to server)

DownloadResponse is used in case of successful "**Download**" request. The FP should always set the status to "1", indicating that the download result will be notified in a subsequent TransferComplete message (i.e. the download will not take place immediately).

```
 <DownloadResponse>
   <Status> 1 </Status>
</DownloadResponse>
```

Error handling. In the following cases, the FP should answer the "Download" request with a fault message as defined in clause B.4.2.3:

- **9004 Resources exceeded:** if an attempt is made to queue an additional "Download" request when the FP's file transfer queue is already full.
- **9003 Invalid arguments:** should be used to reject the Download request if the FP detects the presence of the "userinfo" component in the file source URL.
- **9010 Download Failure:** if the FP rejects the Download request because the FileSize argument exceeds the available space on the device.
- **9012 File transfer server authentication failure** (associated with Upload, Download, TransferComplete or AutonomousTransferComplete methods).
- **9013 Unsupported protocol for file transfer** (associated with Upload and Download methods).

NOTE: The above list does not mention the all-purpose error codes described in clause B.4.2.3.

C.4.2.3.3 Preventing too many Download messages

The management server could attempt to send several "Download" requests in a row (i.e. not sending HTTP response 204 immediately). In case the FP wishes to handle a limited number of "Download" requests at a time, it may reply with a "9004 Resources exceeded" SOAP error message (see clause B.4.2.3). See also figure C.5.

NOTE 1: However, a FP using TR-069 should be able to queue at least three file transfers (download or upload).

NOTE 2: Whatever strategy is used on FP side, each file download will only start following a PP originating "Handset Version indication" message to the FP.

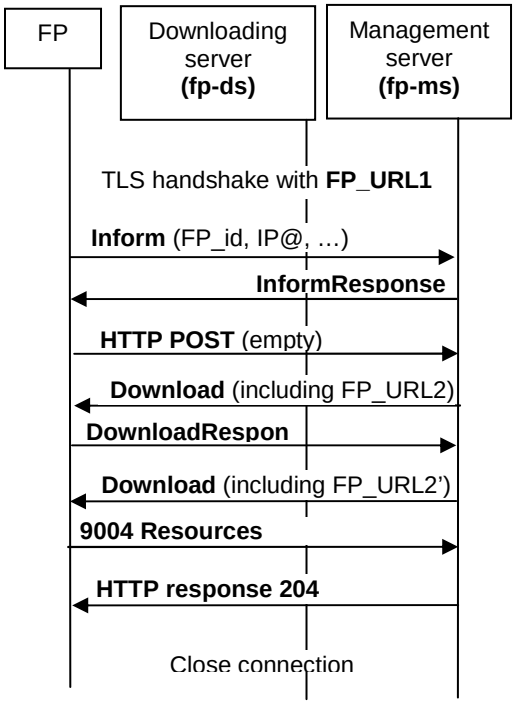


Figure C.5: Prevent multiple Download requests from the management server

C.4.2.4 Transfer complete exchange

For each download performed, the FP will send a distinct **TransferComplete** message to the management server. This message should only be sent upon successful download and use (e.g. installation, test) of the file. If the download and/or use of the file fail, the device should not retry the download but will use the **TransferComplete** message to report a failure to the management server.

C.4.2.4.1 TransferComplete (from FP to server)

The command key value should again be copied from the corresponding **Download** message (as for the Inform message).

```
<TransferComplete>
   <CommandKey> <!--server defined download instance ref.--> </CommandKey>
   <FaultStruct>
      <FaultCode></FaultCode>
      <FaultString></FaultString>
   </FaultStruct>
   <StartTime> </StartTime>
   <CompleteTime>  </CompleteTime>
</TransferComplete>
```

C.4.2.4.2 TransferCompleteResponse (from server to FP)

TransferCompleteResponse is an empty response message acknowledging receipt of the TransferComplete message.

```
<TransferCompleteResponse/>
```

C.4.2.5 Error handling-"Fault" message

In order to notify an error to the other party, the FP or management server should send a "Fault" message of the following form.

NOTE 1: A Fault message is sent on the SOAP layer.

```
<soap:Envelope
      xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
      xmlns:cwmp="urn:dslforum-org:cwmp-1-1">

   <soap:Header>
        <cwmp:ID soap:mustUnderstand="1">1234</cwmp:ID>
   </soap:Header>

   <soap:Body>
      <soap:Fault>
         <faultcode>Server</faultcode>
         <faultstring>CWMP fault</faultstring>
         <detail>
            <cwmp:Fault>
               <FaultCode>9000</FaultCode>
               <FaultString>Download method not supported</FaultString>
            </cwmp:Fault>
         </detail>
      </soap:Fault>
   </soap:Body>

</soap:Envelope>
```

NOTE 2: FP to management server error codes are of the form 9xxx. Management server to FP errors are of the form 8xxx.

All purpose error codes are defined in table C.1 below.

Table C.1: Error codes

Management server to FP response (SOAP faultcode value = "Server")	FP to management server response (SOAP faultcode value = "Server")	Description
8000	9000	Method not supported
8001	9001	Request denied (no reason specified)
8002	9002	Internal error

C.4.2.6 Alternative exchanges

C.4.2.6.1 RequestDownload (from server to FP)

In some cases, the FP could use the RequestDownload message, in order to force a "Download" (figure C.6).

NOTE 1: A management server could not implement this method.

NOTE 2: The FP cannot require a specific "Download" message instance.

```
<RequestDownload>
  <FileType>1 Firmware Upgrade Image</Event>
</RequestDownload>
```

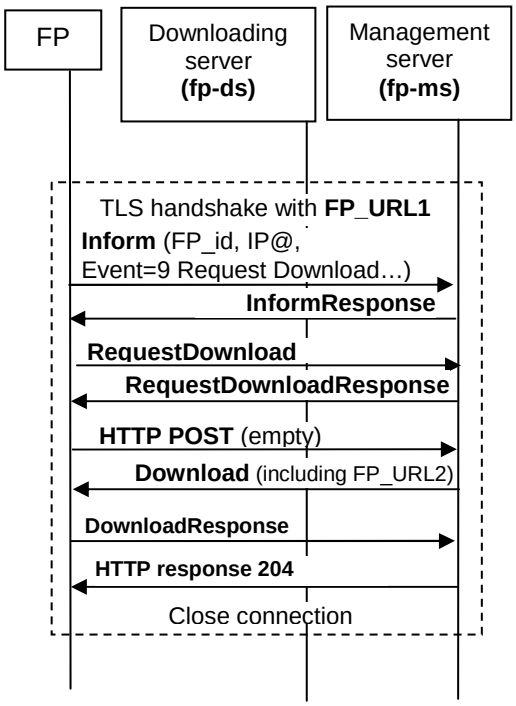


Figure C.6: Use of the RequestDownload method

C.4.2.6.2 RequestDownloadResponse (from server to FP)

RequestDownloadResponse is an empty response message acknowledging receipt of the RequestDownload message.

```
<RequestDownloadResponse/>
```

Error handling: In the following cases, the management server will answer the **RequestDownload** request with a fault message as defined in clause B.4.2.3, instead of a **RequestDownloadResponse** message. The applicable codes are the following:

	Error code	**Error string**	**SOAP faultcode**
-	8003	Invalid arguments	Client
-	8005	Retry request	Server

NOTE: The above list of applicable codes does not mention the all-purpose error codes described in clause B.4.2.3.

Annex D (informative): Recommended implementation of procedures

This annex illustrates some possible implementations of procedures and coding of messages.

D.1 Example of coding of "Handset version indication" followed by one "URL indication" C-plane message (PP to FP)

The following coding depicts a use case with user Initiated SUOTA, request for a new file, URL to follow on 1 <<IWU-TO-IWU>> message.

In the current example, the PP sends to the PP the following parameters:

- EMC= "4660" (1234H)
- FileNumber = "1": ordinal number for the requested file (as reason field is "0")
- Reason = "0": request for a new file
- Flag= "001": User initiated software upgrade
- SW Version identifier = "1.2.0"
- HW Version identifier = "v2.1.0"
- URL1 = "http://pp-ms.example.com/info"

Table D.1: Example of a Handset version indication command sent over one <<IWU to IWU>> information element

Bit:	8	7	6	5	4	3	2	1	Octet	Standard values within the field/IE	Normative action/comment
	0	<< IWU-TO-IWU >>							1		
	Length of Contents (L = L_S + L_H + 11)								2	16H	L = L_S + L_H + 11 = 5+6+11=22
	1	S/R = 1	**Protocol Discriminator = "Software upgrade over the air" = '06'H**						3	C6H	
	0/1 ext	**Command = "Handset version indication" = '0'H**							4	80H	Extension bit8=1 See note
		EMC value high byte							5	12H	
		EMC value low byte							6	34H	
		URL1 to follow							7	01H	Number of URL messages following in a "URL indication" command =1
	0	reserved			**fileNumber**				8	01H	reserved = 0H requested fileNumber = 1H
	0/1 ext	**flags**			**Reason**				8a	90H	Extension bit 8=1 flags = "001" Reason = "0" Success of previous files application. Request for a new file (indicated by "fileNumber" field)
	<SW Version identifier> = 1								9	01H	
	Length of SW Version identifier (1 ≤ L_S ≤ 20)								10	05H	L_S = 5
	SW Version identifier first octet (IA5 character)								11	31H	Character (1) = '1'
	…								12	2EH	Character (2) = '.'
	…								13	32H	Character (3) = '2'
	…								14	2EH	Character (4) = '.'
	SW Version identifier last octet (IA5 character)								L_S +10 (15)	30H	Character (5) = '0'
	<HW Version identifier> = 2								L_S +11 (16)	02H	
	Length of HW Version identifier (1 ≤ L_H ≤ 20)								17	06H	L_H = 6
	HW Version identifier first octet (IA5 character)								18	76H	Character (1) = 'v'
	…								19	32H	Character (2) = '2'
	…								20	2EH	Character (3) = '.'
	…								21	31H	Character (4) = '1'
	…								22	2EH	Character (5) = '.'
	HW Version identifier last octet (IA5 character)								L_S + L_H +12 = 5+6+12 = 23	30H	Character (6) = '0'

NOTE: One should take care the extension bit be set to 1.

Table D.2: Example of a PP to FP URL indication command sent over one <<IWU to IWU>> IE

Bit:	8	7	6	5	4	3	2	1	Octet	Standard values within the field/IE	Normative action/comment
	0	<< IWU-TO-IWU >>							1		
	Length of Contents (L = Lu + 8)								2	25H	Lu + 8 = 29 +8 = 37
	1	S/R = 1	**Protocol Discriminator = "Software upgrade over the air" = '06'H**						3	C6H	
	0/1 ext	**Command = "URL indication" = '2'H**							4	82H	Extension bit 8=1, see note
	URL to follow								5	00H	This message is the last message and contains the last part of the URL
	Length of URL content in this IE (Lu)								6	1DH	Lu = 29
	URL content first octet (IA5 character)								7	68H	Character (1) = 'h'
	URL content second octet (IA5 character)								8	74H	Character (2) = 't'
	…								…		
	URL content last octet (IA5 character)								Lu+6 = 35	6FH	Character (29) = 'o'
NOTE: One should take care the extension bit be set to 1.											

D.2 Example of coding of "Handset version available" followed by three "URL indication" C-planes messages (FP to PP)

The following coding depicts a use case where FP answers to PP to indicate the downloading URL.

The parameters to be sent by the PP are the following:

- DelayMinutes = "256 minutes" (100H)
- User Interaction = "User interaction required"
- Software version identifier of the software to be installed = "1.2.1"
- URL2 = "http://pp-ds.example.com/hdphone/manufacturer-model/model-hardware/ v2.1.0/1.2.1/URLSegmentedOverThreeMessages-img.bin"

As URL2 is more than 55 octets long (here 118), it is split over (here 3) segments:

- Segment 1 (55 octets) = "http://pp-ds.example.com/hdphone/manufacturer-model/mod"
- Segment 2 (55 octets) = "el-hardware/ v2.1.0/1.2.1/URLSegmentedOverThreeMessages"
- Segment 3 (8 octets) = "-img.bin"

Table D.3: Example of a Handset version available command sent over one <<IWU to IWU>> IE

Bit: 8	7	6	5	4	3	2	1	Octet	Standard values within the field/IE	Normative action/comment
0	<< IWU-TO-IWU >>							1		
Length of Contents (L = Ls + 8)								2	0DH	L = Ls + 8 = 13
1	S/R = 1	**Protocol Discriminator = "Software upgrade over the air" = '06'H**						3	C6H	
0/1 ext	**Command = "Handset version available" = '1'H**							4	91H	Extension bit 8 = 1 See note
	DelayMinutes value high byte							5	01H	
	DelayMinutes value low byte							6	00H	
	URL2 to follow							7	03H	Number of URL messages following in a "URL indication" command = 3
0/1 ext	**User Interaction**			reserved				8	90H	Extension bit 8 = 1 User Interaction =001B Reserved =0000B
	<SW Version identifier> = 1							9	01H	
	Length of SW Version identifier (1 ≤ Ls ≤ 20)							10	05H	Ls = 5
	SW Version identifier first octet (IA5 character)							11	31H	Character (1) = '1'
	…							12	2EH	Character (2) = '.'
	…							13	32H	Character (3) = '2'
	…							14	2EH	Character (4) = '.'
	SW Version identifier last octet (IA5 character)							Ls +10 = 15	31H	Character (5) = '1'

NOTE: One should take care the extension bit be set to 1.

Table D.4: Example of a FP to PP URL indication command sent over 3 <<IWU to IWU>> IEs - First message

Bit: 8	7	6	5	4	3	2	1	Octet	Standard values within the field/IE	Normative action/comment
0	<< IWU-TO-IWU >>							1		
Length of Contents (L = Lu + 8)								2	3FH	55+8= 63
1	S/R = 1	**Protocol Discriminator = "Software upgrade over the air" = '06'H**						3	C6H	
0/1 ext	**Command = "URL indication" = '2'H**							4	82H	
URL to follow								5	02H	Number of URL messages following in a "URL indication" command = 2
Length of URL content in this IE (Lu)								6	37H	Lu = 55
URL content first octet (IA5 character)								7	68H	Character (1) = 'h'
URL content second octet (IA5 character)								8	74H	Character (2) = 't'
…										
URL content last octet (IA5 character)								Lu+6 = 61	64H	Character (55) = 'd'

Table D.5: Example of one FP to PP URL indication command sent over 3 <<IWU to IWU>> IEs - Second message

Bit:	8	7	6	5	4	3	2	1	Octet	Standard values within the field/IE	Normative action/comment
	0	<< IWU-TO-IWU >>							1		
	Length of Contents (L = Lu + 8)								2	3FH	55+8= 63
	1	S/R = 1	**Protocol Discriminator = "Software upgrade over the air" = '06'H**						3	C6H	
	0/1 ext	**Command = "URL indication" = '2'H**							4	82H	
	URL to follow								5	01H	Number of URL messages following in a "URL indication" command = 1
	Length of URL content in this IE (Lu)								6	37H	Lu = 55
	URL content first octet (IA5 character)								7	65H	Character (1) = 'e'
	URL content second octet (IA5 character)								8	6CH	Character (6) = 'l'
	URL content third octet (IA5 character)								9	2DH	Character (3) = '-'
	…										
	URL content last octet (IA5 character)								LU+6 = 61	73H	Character (55) = 's'

Table D.6: Example of one FP to PP URL indication command sent over 3 <<IWU to IWU>> IEs - Third message

Bit:	8	7	6	5	4	3	2	1	Octet	Standard values within the field/IE	Normative action/comment
	0	<< IWU-TO-IWU >>							1		
	Length of Contents (L = Lu + 8)								2	0DH	5+8= 13
	1	S/R = 1	**Protocol Discriminator = "Software upgrade over the air" = '06'H**						3	C6H	
	0/1 ext	**Command = "URL indication" = '2'H**							4	82H	
	URL to follow								5	00H	This message is the last message and contains the last part of the URL
	Length of URL content in this IE (Lu)								6	08H	Lu = 8
	URL content first octet (IA5 character)								7	2DH	Character (1) = '-'
	URL content second octet (IA5 character)								8	69H	Character (6) = 'i'
	URL content third octet (IA5 character)								9	6DH	Character (3) = 'm'
	…										
	URL content last octet (IA5 character)								Lu+6 = 8+6 = 14	6EH	Character (8) = 'n'

D.3 Example of IWU-ATTRIBUTES coding (PP to FP direction)

The following coding depicts a use case where the SDU max values declared by the PP are the following.

SDU Size	Value	Coded value
direction PT => FT	1024	1024/8 = 128 = 0080H Adding extension bits, coding is: '0000 0001'B+ '00000000'B = **0100H**
direction FT => PT	2048	2048/8 = 256 = 0100H Adding extension bits, coding is: '0000 0010'B+ '10000000'B = **0280H**

NOTE: In the table above, extension bits are underlined, final coded value is in bold.

Table D.7: Example of one IWU-ATTRIBUTES sent over 1 <<IWU to IWU>> information element

Bit:	8	7	6	5	4	3	2	1	Octet:	Standard values within the field/IE
		<< IWU-ATTRIBUTES >> '12'H							1	
		Length of Contents (L) '0A'H							2	
	1	'01'B		'00000'B					3	A0H
	1	Negotiation indicator '010'B		Profile subtype '1000'B					4	A8H
	0	Maximum SDU size (PT ⇒ FT) MSB (7 bits)							5	01H
	0	Maximum SDU size (PT ⇒ FT) LSB (7 bits)							5a	00H
	0	Maximum SDU size (FT ⇒ PT) MSB (7 bits)							5b	02H
	1	Maximum SDU size (FT ⇒ PT) LSB (7 bits)							5c	80H
	1	Operation code '01'B		Optional groups '00'B		CH=0 '0'B	Spare '00'B		6	A0H
	seq '0'B	Generic Media context indicator (GMCI) '0000000'B							6a	00H (PP to FP direction)
	Application protocol identifier (MSB) '04'H								6b	04H
	'37'H								6c	37H

Annex E (informative): Implementation guidelines

E.1 PP implementation guidelines

E.1.1 Capability bits

The handset declares the following capability bits:

- Please make sure to respect: ETSI EN 300 175-5 [5], clause 7.7.41.
- Please make sure to respect: ETSI EN 300 444 [14], clause 8.17.
- Please make sure to follow: ETSI EN 301 649 [15], clause 12.3.
- Please make sure to follow: clause 7.5.7, table 21 (all profile indicator bits).

E.1.2 Security mechanisms

Data calls are considered similar to ETSI TS 102 527-3 [17] (DECT-NG PART3) voice calls from a security point of view. See clause 6.7.2, feature DPRS-N.43 (alias for GAP N.35) and DPRS clause 8.2 for the per procedure status:

- All Enhanced security GAP.N.35 mandatory procedures apply.
- All authentication mechanisms from ETSI TS 102 527-3 [17] (DECT-NG PART3) are re-used (registration + locate request).
- All encryption mechanisms from ETSI TS 102 527-3 [17] (DECT-NG PART3) are re-used (all calls are encrypted).

E.1.3 At locate request

- No <<SETUP CAPABILITY>> IE sent at locate request.

E.1.4 C-plane exchanges

Please check annex D for example of C-Plane messages.

All C-Plane commands in section 7.5.5.1:

- Extension bit coding is set to 1. Please strictly follow annex D and figures 7, 9 and 11.

Handset version indication message:

- Please make sure the EMC is correctly coded (do not invert low and high bytes).

URL indication command coding:

- Be careful about IWU segmentation, especially if the URL string is bigger than 46 octets (46 limit is specific to the handset).
- Be careful about URL to follow coding (n,n-1….0). Please strictly follow annex D, figures 7, 9, 11 and example in figure 10.

E.1.5 DPRS call setup

- Long slot radio format is used. (Make sure no full slot is re-used from any previous voice call).
- <<basic service>> / <call class> = 'Normal call setup'.
- <<basic service>>/ <basic service> = Light data service class 4 (see clause 7.6.1.2.1).
- <<IWU-ATTRIBUTES>> information element.
  - Sent systematically in the {CC-SETUP} (see clause 7.6.1.2.1.1, 'concerned devices and connections').
  - <<IWU-ATTRIBUTES>> coding (see annex D):
    - IWU/DLC level: No sequence numbering used at SDU level (see clause 7.6.1.2.1.3).
    - The handset supports at least Maximum SDU size of 752 octets (in both directions).
    - The handset sends [Maximum SDU size PT=> FT] and [Maximum SDU size FT => PT] supported by the handset, depending on its memory capabilities (see clause 7.6.1.2.1.3).
    - The coding of the GMCI bit is 0 in the {CC-SETUP} (handset value).

E.1.6 During established call

MAC/DLC/IWU layers:

- If the handset supports **only** the SUOTA feature (and no other data service): The PT behaviour follows the 'Simple binary content download' profile described in clause 7.6.1.2.1.
- DLC/MAC level: Sequence numbering is used at PDU level (MAC level).

NOTE: Besides, sequence numbering is NOT used at SDU level as already indicated above in clause E.2.5.

- The PT uses the "maximum SDU size" **which was sent back by the FT in the {CC-CONNECT}** (for each direction PT=> FT and FT=>PT). See clause 7.6.1.2.1.3.
- In the DPRS Frame Relay SDU (DLC level):
  - The PP always uses the GMCI bit value received at {CC-CONNECT}.
  - Extension bit for sequence number is set to 1 (no optional octet for sequence number).

HTTP requests:

- GET HTTP request, short form (see clause A.1.3).
- HTTP error handling: FT supports all errors defined in clauses A.1 and A.1.8.
- Some HTTP fields are mandatory to send and to support (see clause A.1.4).
- **Recommended**: The HTTP range requests from the handset should be consistent with the maximum SDU size PT->FT.

E.2 FP implementation guidelines

E.2.1 Capability bits

The FP declares the following capability bits:

- Please make sure to respect: ETSI EN 300 175-5 [5], clause F.3 (especially bit a45).

- Please make sure to respect: ETSI EN 300 444 [14], clause 13.6.
- Please make sure to follow: ETSI EN 301 649 [15], clause 12.16.
- Please make sure to follow clause 7.5.10.

E.2.2 Security mechanisms

Data calls are considered similar to ETSI TS 102 527-3 [17] (DECT-NG PART3) voice calls from a security point of view. See clause 6.7.2, feature DPRS-N.43 (alias for GAP N.35) and DPRS clause 8.2 for the per procedure status:

- All Enhanced security GAP.N.35 mandatory procedures apply.
- All authentication mechanisms from ETSI TS 102 527-3 [17] (DECT-NG PART3) are re-used (registration + locate request).
- All encryption mechanisms from ETSI TS 102 527-3 [17] (DECT-NG PART3) are re-used (all calls are encrypted).

E.2.3 At locate accept

- No <<SETUP CAPABILITY>> IE sent at locate request.

E.2.4 C-plane exchanges

Please check annex D for example of C-Plane messages.

All C-Plane commands in section 7.5.5.1:

- Extension bit coding is set to 1. Please strictly follow annex D and figures 7, 9 and 11.

URL indication command coding:

- Be careful about IWU segmentation, especially if the URL string is bigger than 55 octet (55 limit is specific to the FT).
- Be careful about URL to follow coding (n, n-1….0). Please strictly follow annex D, figures 7, 9 and 11 and example figure 10.

Handset version available message:

- Please make sure the DelayMinutes is correctly coded (do not invert low and high bytes).

E.2.5 {CC-CONNECT} format for a DPRS call

<<IWU-ATTRIBUTES>> information element.

- Sent systematically in {CC-CONNECT} (see clause 7.6.1.2.1.1, 'concerned devices and connections').
- <<IWU-ATTRIBUTES>> IE coding (see clause 7.6.1.2.1.3):
  - IWU/DLC level: No sequence numbering is used at SDU level.
  - The FP at least supports a Maximum SDU size of 752 octets (in both directions).
  - The FP sends [Maximum SDU size PT=> FT] and [Maximum SDU size FT => PT] **which is the final value negotiated between PT and FT** (see clause 7.6.1.2.1.3).
  - The FP chooses the GMCI value (and always chooses 1).

E.2.6 During established call

MAC/DLC/IWU layers:

- The FP behaviour follows the "Simple binary content download" profile described in clause 7.6.1.2.1.
- DLC/MAC level: Sequence numbering is used at PDU level (MAC level).

NOTE: Besides, sequence numbering is NOT used at SDU level as already indicated above in clause E.3.5.

- The FP uses the "maximum SDU size" **which was sent back by the FP in the {CC-CONNECT}** (for each direction PT => FT and FT => PT). See clause 7.6.1.2.1.3.
- In the DPRS Frame Relay SDU (DLC level):
  - The FP always uses the GMCI bit value that it sent at {CC-CONNECT}.
  - Extension bit for sequence number is set to 1 (no optional octet for sequence number).

HTTP requests format:

- GET HTTP request short form (see clause A.1.3).
- HTTP error handling: FT supports all errors defined in clauses A.1 and A.1.8.
- Optimized HTTP in downstream direction, recommendation to avoid sending superfluous http fields (see clause A.1.4).

HTTP headers support:

- a FP not being the targeted server of an HTTP request:
  - forwards the request to the server without analysing the header fields, except for the 'Host' header allowing the location of the server;
  - should however perform a broad parsing of the response header received from the server, in order to filter out header fields not relevant for the DECT PP (see clause A.1.4). A detailed parsing of these header fields is not needed.

Annex F (informative): DLC layer Guidelines and Use cases

F.1 Detailed requirements

F.1.1 Transmission class

The present document uses of LU10 (see ETSI EN 300 175-4 [4], clause 11.12), with 'transmission class 2' (variable throughput LUx retransmission), defined in clause 14.2.3 of ETSI EN 300 175-4 [4].

The frame type (=PDU type) is FU10a for data PDUs and FU10c for acknowledgement PDUs.

F.1.2 Retransmission protocol

In all DPRS and in its derived standards (such as the present document), DLC retransmission uses the SELective (SEL) retransmission protocol.

NOTE: ETSI EN 300 175-4 [4] (DLC) allows both Go back N and SELective algorithms (see clause 14.3.4 there), while DPRS and the present document only use SELective.

F.1.3 PDU structure and payload

F.1.3.1 FU10a frames alias PDUs

PDUs and FU10a frames are the same thing. The PDUs are created by the DLC layer and handed over to the MAC layer for transmission.

In our case, PDUs are 64 octets long (long slots), as they are transported in an I_{PM} (protected multi-subfield) channel with 64 octet payload (the long slot is 640 bits = 8 x 80 = 8 x (64+16) = 8x64 (payload) + 8x16 (CRC). The payload is 8x64 bits = 8x8 octets=64 octets)

F.1.3.2 FU10a frames numbering (PDU sequence numbers)

FU10a frames are sequence numbered irrespective of their content.

In particular, FU10a frames containing FU10c frames will be numbered as any other FU10a frames (i.e. as those containing data) by the receiving side.

The modulo used for SN numbering will be 256 (as for any window size below 128; see DLC 11.2.3: "The modulo operation will be taken into account, e.g. SN = 0 will be after SN = 511 and for window sizes ≤ 128 (SN = [0.255]), SN = 0 will be after SN = 255").

F.1.3.3 PDU structure

A PDU structure is as follows: SN_1 ($Length\_of\_info_1$ $info_{n1}$) … ($Length\_of\_info_k$ $info_{nk}$), with indices used for indicating a field length in octets.

- SN_1 is the PDU sequence number octet (1 octet long).
- for i in [1..k], '$Length\_of\_info_i$' is the ith length of info field. It contains:
  - a LI field (called here LI_i);
  - a More bit M (called here M_i).

F.1.3.4 PDU length

With the above notation, PDU length = 1+k+n1+…+nk.

For long slots, PDU length has to be 64, therefore 1+k+n1+…+nk = 64.

F.1.3.5 PDU payload

This is the room for transporting content, i.e. either SDU segments or FU10c frames (for ACKs and NACKs in the reverse direction). It is the reunion of all $info_{ni}$ fields, for i in [1..k].

In general, a PDU may combine:

- one or several FU10c frames;
- optionally followed by a single SDU segment;
- followed by a padding (if needed) of at least one octet.

NOTE 1: FU10c frames are always first in the PDU.

Maximum payload size. When long slots are used, a PDU with no padding and with only one length of info field has a 62 octet long payload. In that case, the PDU structure is as follows:

- SN_1_(1 octet).
- Length_of_$info_1$ (1 octet) containing $LI_1 = 62$ and $M_1 = 0$ or 1.
- $info_{n1}$ (62 octets) with $n1 = LI_1 = 62$.

NOTE 2: $M_1 = 0$ is rarely used: only if the last SDU segment happens to be exactly 62 octets long (see SDU boundary).

F.1.3.6 PDU padding

If the last octets at the end of a PDU are not used:

- the first unused octet contains a length of info field with (LI,M) = (0,0);
- if more than one octet are not used, then subsequent unused octets are filled with 'F0'H. This padding value is not mandatory but highly recommended.

NOTE: If only one octet is unused, the length of info field with (LI,M) = (0,0) terminates the PDU and is not followed with any 'info' value (i.e. no 'F0'H octet follows).

F.1.4 SDUs

F.1.4.1 Definition

SDU are the external protocol data units (i.e. external to the DLC layer). It is made of a header (one or two octets) and a body, as described in ETSI EN 301 649 [15], clause B.8.2.1.

The DLC layer does not care about the SDU content (in particular, the DLC layer does not distinguish between the SDU header and SDU data).

F.1.4.2 SDU Segments

A piece of SDU embedded in a PDU is called a segment.

In most cases, the PDU contains a single SDU segment. However, in principle (but not in SUOTA), depending on the relative sizes of PDUs and SDUs, a single PDU may contain several segments (each from a different SDU).

In the present document, a PDU always contains either 0 or 1 SDU segment (not more): a party (PT or FT) always sends a single SDU at a time, waiting for an SDU in the reverse direction before sending another one. The reasons are the following:

- a PDU may contain 0 SDU segment if it is used to convey only FU10c frames;
- a PDU may not contain more than 1 SDU segment because:
  - No two SDU segments for the same SDU are transported in the same PDU, as SDU segments are always of the maximum possible size.
  - No two SDU segments for different SDUs are transported in the same PDU, because, in the present document:
    - chopping is not used (i.e. application packet size are never split into several SDUs); as a result, an SDU is never followed by another SDU for sending the same application packet;
    - HTTP pipelining is not used: the PT always waits for the response before sending a new application packet (therefore a single SDU is sent at a time);
    - the FT does not send requests, but only answers to PT requests. The answer always fits in a single SDU (because chopping is not used) and answers cannot be consecutive (also because pipelining is forbidden).

Therefore, data PDUs in the present document will always contain one segment which may be a starting, ending or inner segment, or a full SDU.

F.1.4.3 SDU boundaries and more bit (M)

In LU10, the PDUs also transport SDU boundaries (where a SDU terminates and (if any) the next one starts).

The SDU boundary resides in the length of info field preceding the last SDU segment. This segment is found in the last PDU used for that SDU. The length of info field containing the SDU boundary is:

- $1 \leq LI \leq 62$;
- M = 0 ('More' bit is '0').

ETSI EN 300 175-4 [4], clause 11.12.2: In all cases, the original SDU boundaries will be preserved (i.e. service integrity will be maintained) by use of a length indicator and extended More bit as defined in ETSI EN 300 175-4 [4], clause 13.3.

F.1.5 Transit delay and SDU lifetime

The value in ETSI EN 301 649 [15], clauses A.2.1 (table A.10) and A.2.2 (table A.16) is 0; which according to ETSI EN 300 175-5 [5], clause 7.7.42 means infinite (no limit for the transit delay).

DLC 14.3.4.1 indicates that there is the following equality: 'SDU lifetime' = T(R) = transit delay.

So the SDU lifetime is Infinite.

F.1.6 (transmitter) Window size

Window size information element. ETSI EN 300 175-5 [5], clause 7.7.43 (<<WINDOW-SIZE>> IE) defines an IE allowing both parties to negotiate the window size.

NOTE 1: In this IE, The following values are not used in the present document: "Maximum PDU length (PT => FT)", "SDU LAPU timer (PT => FT)" as well as their counterparts in the reverse direction.

ETSI EN 301 649 [15], clause 11.1.1 Window size: 32 is the minimum M supported window size for data rate up to 96 kbit/s. This value is also used in ETSI EN 301 649 [15], clauses A.2.1 and A.2.2, tables A.9 (class 4) and A.15 (class 3).

NOTE 2: On transmitter side, the transmitter is however allowed to use a smaller value unilaterally. So the size of 32 is does not introduce an important limitation.

F.1.7 FU10c frames (=one ACK + several NACKs)

F.1.7.1 FU10c in FU10a frames (i.e. in a PDU)

Use of ACKs and NACKs. LU10 with SELective retransmission protocol requires the transmission of ACKs and NACKs.

In the present document, FU10c frames are exclusively transported within FU10a frames (PDUs) of the reverse direction DLC link. Transmission of FU10c frames using the G_F channel (as described in ETSI EN 300 175-4 [4], clause 12.11.2.2) is NOT supported.

NOTE 1: For the mandatory support of this mechanism, see ETSI EN 301 649 [15], clause 11.2.3.1 that uses the procedure described in ETSI EN 300 175-4 [4], clause 12.11.2.1.

NOTE 2: DLC links are monodirectional. This is however possible because the network connection implies the use of two mono-directional links in opposite directions.

Acknowledgement of FU10c frames (i.e. of ACKs and NACKs). As a consequence, these FU10c frames (or more precisely the PDUs containing them) will be also acknowledged (as any other PDUs).

F.1.7.2 FU10c frames + data in the same PDU

Although it is allowed to send FU10c frames and data together in the same PDU, this possibility can only be used with the last SDU segment.

Equal SDU segments (except last one). The reason is that in the context of LU10 (hence in the present document), SDUs can only be segmented into segments of the same length (see ETSI EN 301 649 [15], clause 11.1).

As described in clause F.1.4.3 and in the use cases of clause F.2.1.3, these segments are of length 62 (because long slots are used). This is the maximum possible length and therefore excludes cohabitation of data with FU10c frames.

Last SDU segment and data. This rule obviously does not apply to the last SDU segment which can be of any length in [1..62]. When the last SDU segment length is smaller than 62 - 8 = 54, it is possible to sent at least one FU10c together with it.

NOTE: When data are sent together with one or more FU10c frames, the latter are always sent first.

F.1.7.3 Use of consecutive FU10c frames

When more than 5 NACKs need to be transmitted, it is allowed to use several consecutive FU10c frames. Consecutive FU10c frames may be either consecutive in the same PDU or consecutive in consecutive PDUs.

A FU10c frame cannot be segmented (i.e. the whole FU10c frame is contained in a single PDU).

NOTE 1: There is currently no standard defined limit to the number of consecutive NACKs that can be sent by the receiver. The maximum number of NACKs sent consecutively should however depend on the receiver window size (last NACK has to be in the receiver window).

NOTE 2: See clause F.1.7.4 for details about how such consecutive frames can be used.

Maximum number of FU10c frames within a single PDU. There can be up to 7 FU10c frames within a given PDU. Let f be the number of included FU10c frames, and p the remaining number of octets. We have 1 + 8f + p = 64 (see clause F.1.3). Therefore, f=7 and p=7.

NOTE 3: The remaining 7 octets can be either filled with padding (see clause F.1.3.6) or (rarely) with the last segment of a PDU (of size 6 max) or both.

NOTE 4: The note 3 of ETSI EN 300 175-4 [4] clause 12.11.2.1, indicates a maximum of 8 FU10c frames. This is a rough maximum but the real maximum value is 7, as shown above.

F.1.7.4 Detailed process

A PDU can contain one or more FU10c frame(s) (in that case, FU10c frames are included first).

FU10c frames are 7 octets long. One or more FU10c frames may be used for the sending of acknowledge information. This acknowledge information comprises one ACK and 0 or more NACKs.

- the (only) ACK octet contains a PDU sequence number (called 'n') and is therefore noted ACK(n). It is said to be "cumulative" because it acknowledges that, all PDU(m) with sequence number $m \leq n-1$ were received);
- the NACK octets (if any) are selective (i.e. NACK(m) only indicates non-receipt of PDU(m)).

The contiguous FU10c frames used for sending acknowledge information respect the following rules:

- The first (and possibly only) FU10c frame sent uses either (NA1, NA2) = (0,1) or (1,0), for sending one ACK and 0 or more NACKs (see table below).
  - (NA1, NA2) = (1,0) is only used if at least one NACK has to be sent.
  - If less than 5 NACKs have to be sent (but at least one), then the remaining octets contain a copy of the last NACK.
- The subsequent FU10c frames if needed all use (NA1, NA2) = (1,1). This implies that the first FU10c frame sent (see above) used (NA1, NA2) = (1,0) and contained exactly 5 (different) NACKs.
  - all of these subsequent FU10c frames except the last one contain exactly 6 different NACKs;
  - the last subsequent FU10c frame contains between 1 and 6 different NACKs. The unused NACK octets (if any) contain a copy of the last NACK.

NOTE: A PDU outside (i.e. below) the receiver window (i.e. already received and acked at least once) has to be acked again by the receiver until it is no longer sent by the transmitter, otherwise it could be continuously re-sent. The reason the transmitter re-sends a PDU could be that it did not receive the ACK.

The interpretation of the RSNs as either ACKs or NACKs is indicated by bits NA1 and NA2 [of octet 7 of FU10c], as indicated in the following table (copied from ETSI EN 300 175-4 [4], clause 14.3.4.2.5).

Bit	NA1	NA2	Meaning
	0	0	This frame contains an Ack of a synchronization message in RSN#1, no NACKs
	0	1	This frame contains only one ACK message in RSN#1, no NACKs
	1	0	This frame contains one ACK message in RSN#1plus five NACK messages in RSN#2-RSN#6
	1	1	This frame contains six NACK messages

F.1.8 Synchronization frames

Synchronization frames are not used in the present document. They are however illustrated in use case 6 for information.

F.1.9 End of activity rules

End of activity rules as described in ETSI EN 300 175-4 [4] clauses 14.3.4.1.2 (Tx side end-of-activity rule) and 14.3.4.2.2 (Rx side end-of-activity rule) are used in order to avoid:

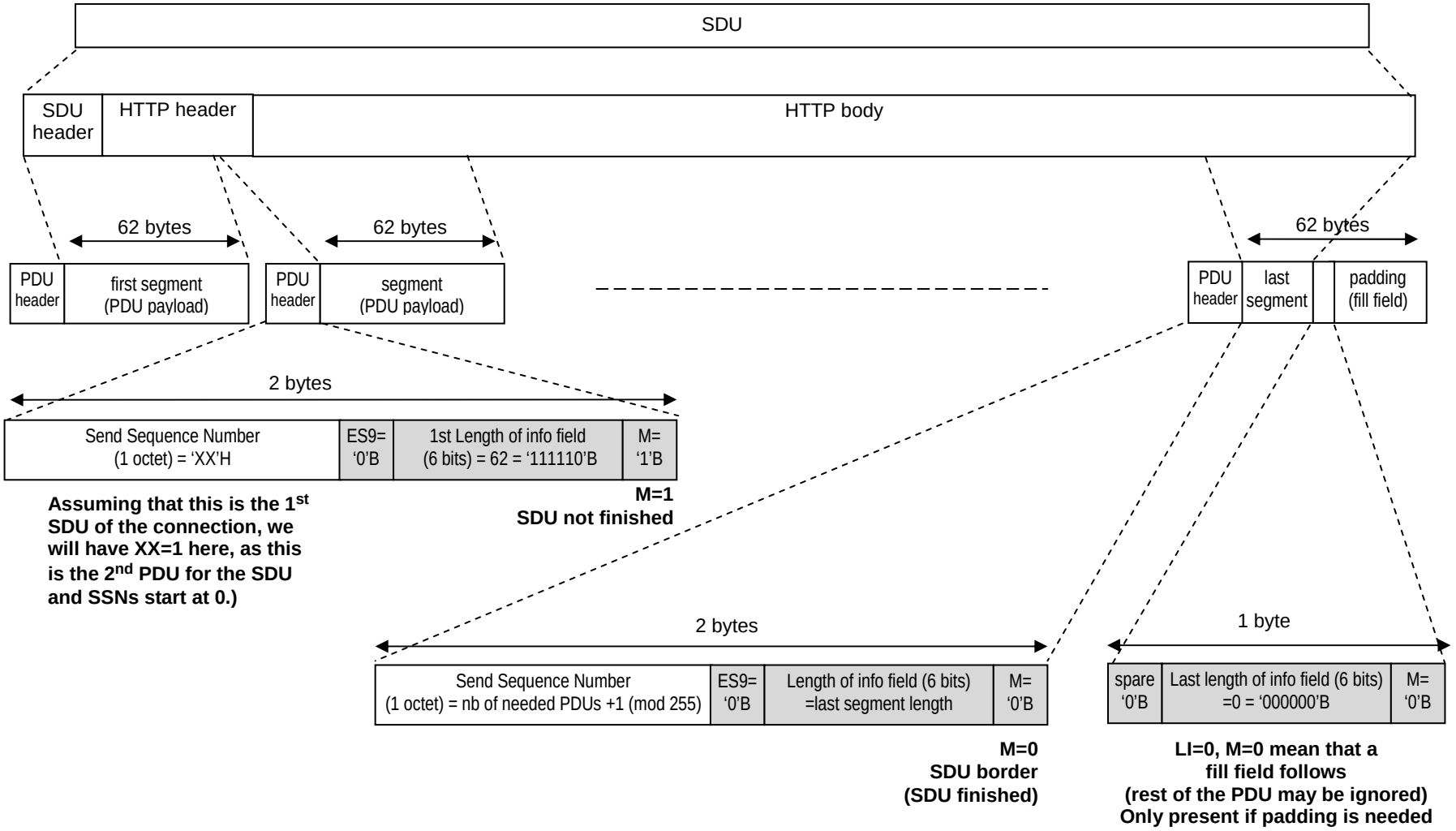
- Infinite request answer loops (ACK, ACK of ACK, etc.).
- Infinite retransmission of ACKs.

F.2 PDU encoding examples

F.2.1 Encoding of PDUs (for SDU data)

F.2.1.1 Figure

Figure F.1 shows the encoding of a PDU for SDU data.



NOTE 1: We ignore the case of a PDU holding segments of different SDUs because it cannot happen in SUOTA (no HTTP pipelining allowed).
NOTE 2: Octets in grey are called 'Length indicator octets' and contain a 'Length of info' field; there may be several of them in the same PDU (as shown here for the last PDU).

Figure F.1: Encoding of a PDU for SDU data

F.2.1.2 Table

kth PDU header value (2 bytes) (note 1)	**All PDUs except last one (k=1..K-1) (more bit M=1)**	**last PDU (k=K) (M=0)**	**Additional length octet in last PDU (before padding) (note 6)**
First octet (SSN)	hexa[s+k-1 mod 255] (note 2)		
Second octet (length octet) (note 3)			
Padding needed (r > 0)	'7D'H (note 4)	hexa[2*r] (notes 1 and 3)	**YES** (value= '00'H]) (note 5)
No padding needed (r=0)	'7D'H (note 4)	'7C'H (note 7)	**NO**

NOTE 1: Assumption is that K PDUs are needed for transferring the SDU: for k=1..K, PDU(k) is transmitted. SDU length is broken down as follows:
- SDU_length=62q+r, with q= SDU_length div 62, r=SDU_length mod 62;
- if r>0, K=q+1, else if r=0, K=q.

NOTE 2: s is the first available SSN number for transmitting the SDU. SSNs are incremented modulo 255. The SSNs used for transmitting the SDU are: s, (s+1) mod 255, …, (s+K-1) mod 255.

NOTE 3: The length octet includes ES9 (always 0 because SSNs here are modulo 255), the length indicator LI, and the more bit M. It is equal to (ES9='0'B, LI=segment length, M) and therefore coded as 'hexa[2*segment_length+M]'.

NOTE 4: hexa[2*62+1] = '7D'H.

NOTE 5: Length octet is here equal to (ES9='0'B, LI=0, M= 0)='00'H. (LI,M)=(0,0) is a special value indicating that the rest of the PDU may be ignored.

NOTE 6: If r=61 (i.e. if SDU_length=62q+61, K=q+1), this additional length octet is not followed by any padding (the additional length octet itself is the only padding octet).

NOTE 7: hexa[2*62+0] = '7C'H.

F.2.1.3 Use cases

F.2.1.3.1 Conventions

NOTE 1: On the left hand side, the number in parentheses is the SSN of the PDU. On the right hand side, the numbers in parentheses are the length of the element in octets.

NOTE 2: For the value of the fill field, see DLC 13.5.

F.2.1.3.2 Use case 1

Use case 1: SDU of minimum length, initial SDU of the connection (s=0)

SDU_length=752=12*62+8

r=8, K=13

Window size (16) is not reached…

Used SSNs = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12.

```
PDU(0)  = 007D <data(62)>
PDU(1)  = 017D <data(62)>
PDU(2)  = 027D <data(62)>
PDU(3)  = 037D <data(62)>
PDU(4)  = 047D <data(62)>
PDU(5)  = 057D <data(62)>
PDU(6)  = 067D <data(62)>
PDU(7)  = 077D <data(62)>
PDU(8)  = 087D <data(62)>
PDU(9)  = 097D <data(62)>
PDU(10) = 0A7D <data(62)>
PDU(11) = 0B7D <data(62)>
```

PDU(12) = 0C10 <data(8)> <last Length indicator octet(1)='00'H> <fill field(53)='F0…F0'H >

NOTE: The SDU border is the least significant bit of 0C10 in PDU(12). An even value (here 10) indicates an SDU border.

F.2.1.3.3 Use case 2

Use case 2: SDU of minimum length for reaching the window, initial SDU of the connection (s=0)

SDU_length=931=15*62+1

r=0, K=16

Window size (16) is reached…

Used SSNs = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12,13,14,15.

PDU(0) = 007D <data(62)>
PDU(1) = 017D <data(62)>
PDU(2) = 027D <data(62)>
PDU(3) = 037D <data(62)>
PDU(4) = 047D <data(62)>
PDU(5) = 057D <data(62)>
PDU(6) = 067D <data(62)>
PDU(7) = 077D <data(62)>
PDU(8) = 087D <data(62)>
PDU(9) = 097D <data(62)>
PDU(10) = 0A7D <data(62)>
PDU(11) = 0B7D <data(62)>
PDU(12) = 0C7D <data(62)>
PDU(13) = 0D7D <data(62)>
PDU(14) = 0E7D <data(62)>
PDU(15) = 0F02 <data(1)> <last Length indicator octet(1)='00'H> <fill field(60)='F0…F0'H >

NOTE 1: On the left hand side, the number in parentheses is the SSN of the PDU. On the right hand side, the numbers in parentheses are the length of the element in octets.

NOTE 2: The SDU border is the least significant bit of 0F02 in PDU(15). 02 even =>border.

NOTE 3: For the value of the fill field, see DLC 13.5.

F.2.1.3.4 Use case 3

Use case 3: SDU of minimum length for reaching the window but with NO padding, initial SDU of the connection (s=0)

SDU_length=992=16*62+0

r=0, K=16

Window size (16) is reached…

Used SSNs = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12,13,14,15.

PDU(0) = 007D <data(62)>
PDU(1) = 017D <data(62)>
PDU(2) = 027D <data(62)>
PDU(3) = 037D <data(62)>
PDU(4) = 047D <data(62)>
PDU(5) = 057D <data(62)>
PDU(6) = 067D <data(62)>
PDU(7) = 077D <data(62)>

```
PDU(8)  = 087D <data(62)>
PDU(9)  = 097D <data(62)>
PDU(10) = 0A7D <data(62)>
PDU(11) = 0B7D <data(62)>
PDU(12) = 0C7D <data(62)>
PDU(13) = 0D7D <data(62)>
PDU(14) = 0E7D <data(62)>
PDU(15) = 0F7C <data(62)>
```

NOTE 1: On the left hand side, the number in parentheses is the SSN of the PDU. On the right hand side, the numbers in parentheses are the length of the element in octets.

NOTE 2: The SDU border is the least significant bit of `0F7C` in PDU(15). 7C even =>border.

F.2.1.3.5 Use case 4

Use case 4: very small SDU. The SDU border is in the first PDU.

SDU_length=50=0*62+50

r=50, K=1

Used SSNs = 0.

`PDU(0) = 0064 <data(50)>` <last Length indicator octet`(1)='00'H> <fill field(11)='F0…F0'H >`

NOTE 1: On the left hand side, the number in parentheses is the SSN of the PDU. On the right hand side, the numbers in parentheses are the length of the element in octets.

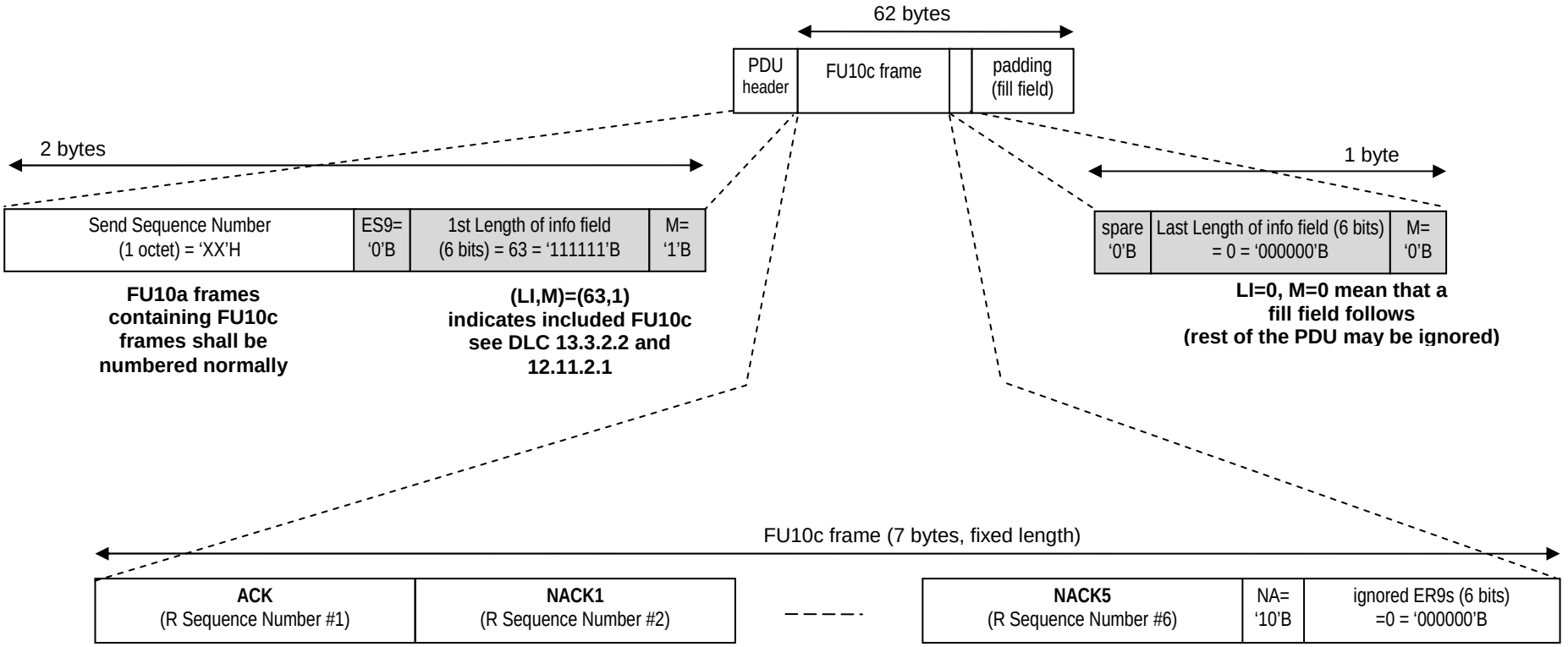
NOTE 2: The SDU border is the least significant bit of `0F7C` in PDU(15). 7C even =>border.

F.2.2 Encoding of PDUs (for ACK and NACKs)

F.2.2.1 Figure

Figure F.2 shows the encoding of a PDU for ACKs and NACKs.

[When one FU10c frame only is used], the ACK/NACKs PDU has a useful length of 10 octets (2+7+1) and a padding of 54 octets (if we do not send any data).



NOTE 1: Octets in grey are called 'Length indicator octets' and contain a 'Length of info' field; there may be several of them in the same PDU (as shown here).
NOTE 2: The ACK and NACK octets do not contain an ACK/NACK bit. Bits NA1 and NA2 (indicated above collectively as 'NA') are used instead.

Figure F.2: Encoding of a PDU for ACK/NACK

F.2.2.2 Use cases

Case 1: EXAMPLE 1 of DPRS 11.2.3 (completed with encompassing FU10a)

EXAMPLE 1: Example of completion of a FU10c frame in case the receiver has successfully received all PDUs with numbers up to SN = 254 and PDU with SN = 2, missing SN = 255, 0 and 1. The window size has been defined as ≤ 128, allowing to ignore the bit 9:

Octet_1 = 255 (ACK 255);
Octet_2 = 255 (NACK 255);
Octet_3 = 0 (NACK 0);
Octet_4 = 1 (NACK 1);
Octet_5 = 1 (NACK 1);
Octet_6 = 1 (NACK 1);
Octet_7 - "10000000" (1 ACK plus 5 NACKs). See clause 14.3.4.2.1.

FU10c_length=7=0*62+7

r=7, K=1

Used SSN = 2 (this is an example; NOTE: it is not related to the acked/nacked RSNs).

PDU(2)=027F <fu10c(7)=FFFF0001010180> <last Length indicator octet(1)='00'H> <fill field(54)='F0...F0'H>

Case 2: EXAMPLE 3 of DPRS 11.2.3 (completed with encompassing FU10a)

EXAMPLE 3: Example of completion of a FU10c frame in case the receiver has successfully received all PDUs with numbers up to SN = 254 and there are no missing PDUs (window size is irrelevant in this case, since bit 9 is set to 0):

Octet_1 = 255 (ACK 255);
Octet_2 = 0 (NULL);
Octet_3 = 0 (NULL);
Octet_4 = 0 (NULL);
Octet_5 = 0 (NULL);
Octet_6 = 0 (NULL);
Octet_7 - "01000000" (This frame contains only one ACK message in RSN#1, no NACKs). See clause 14.3.4.2.1.

FU10c_length=7=0*62+7

r=7, K=1

Used SSN = 2 (this is an example; NOTE: it is not related to the acked/nacked RSNs).

PDU(2)=027F <fu10c(7)=FF000000000040> <last Length indicator octet(1)='00'H> <fill field(54)='F0...F0'H>

F.3 DLC layer PDU transmission use cases

F.3.1 Use case 1 - transmission OK

Figure F.3 shows the use case of a successful transmission of a DLC layer PDU.

Send 1 SDU of size 'max SDU size' (SDU_length=752=12 × 62 + 8).

Number of PDUs needed: 13 (window size 16 not reached)

One ACK only (no NACKs)

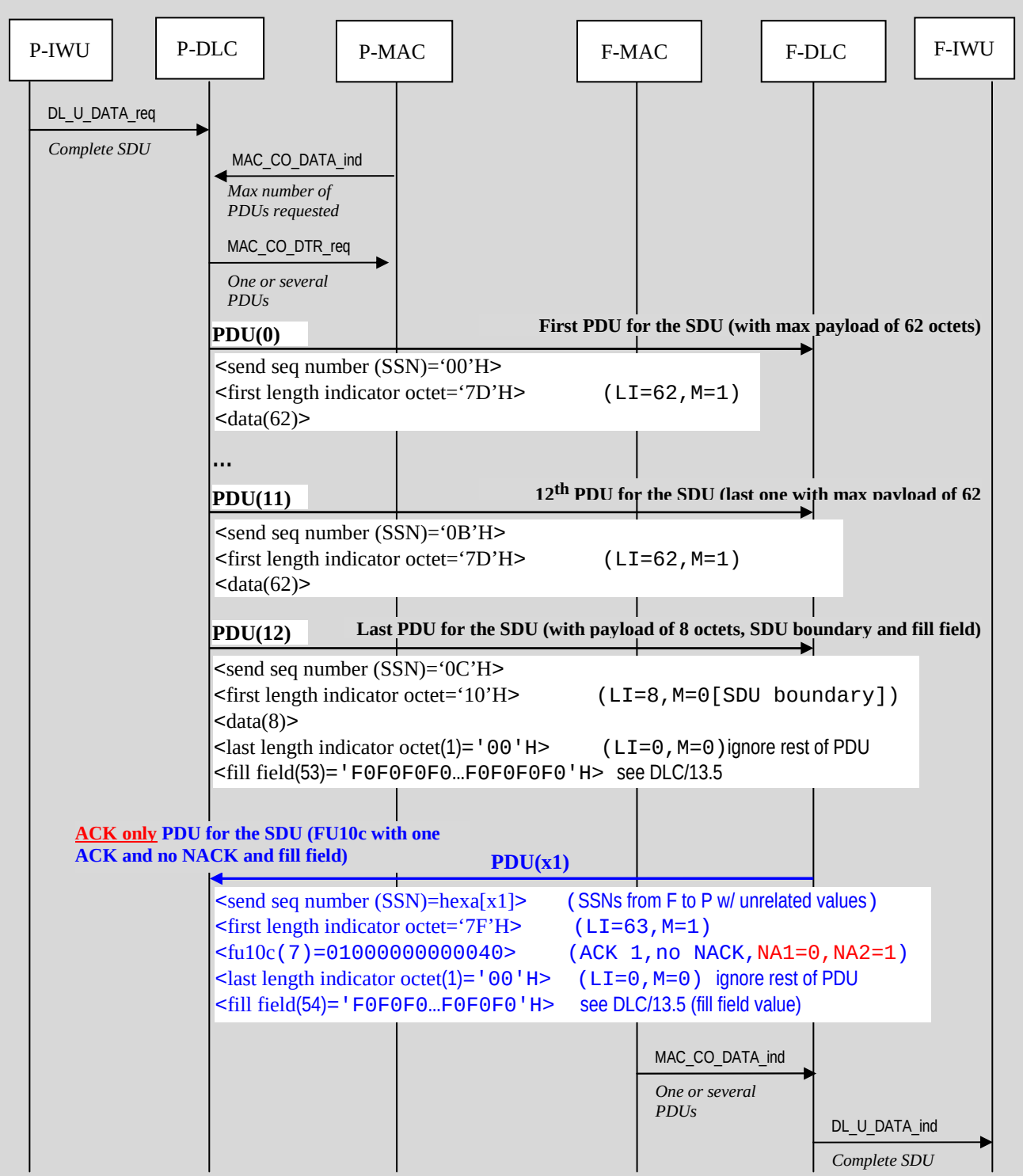


Figure F.3: Successful PDU transmission

F.3.2 Use case 2 - very small PDU - transmission OK

Figure F.4 shows the use case of a successful transmission of DLC layer very small PDU.

Send 1 SDU of length x ≤ 62.

NOTE: The three cases shown below (payload length ≤ 60, =61, =62) remain relevant when sending the SDU last segment, with x='last segment size'.

Number of PDUs needed: only 1.

One ACK only (no NACKs)

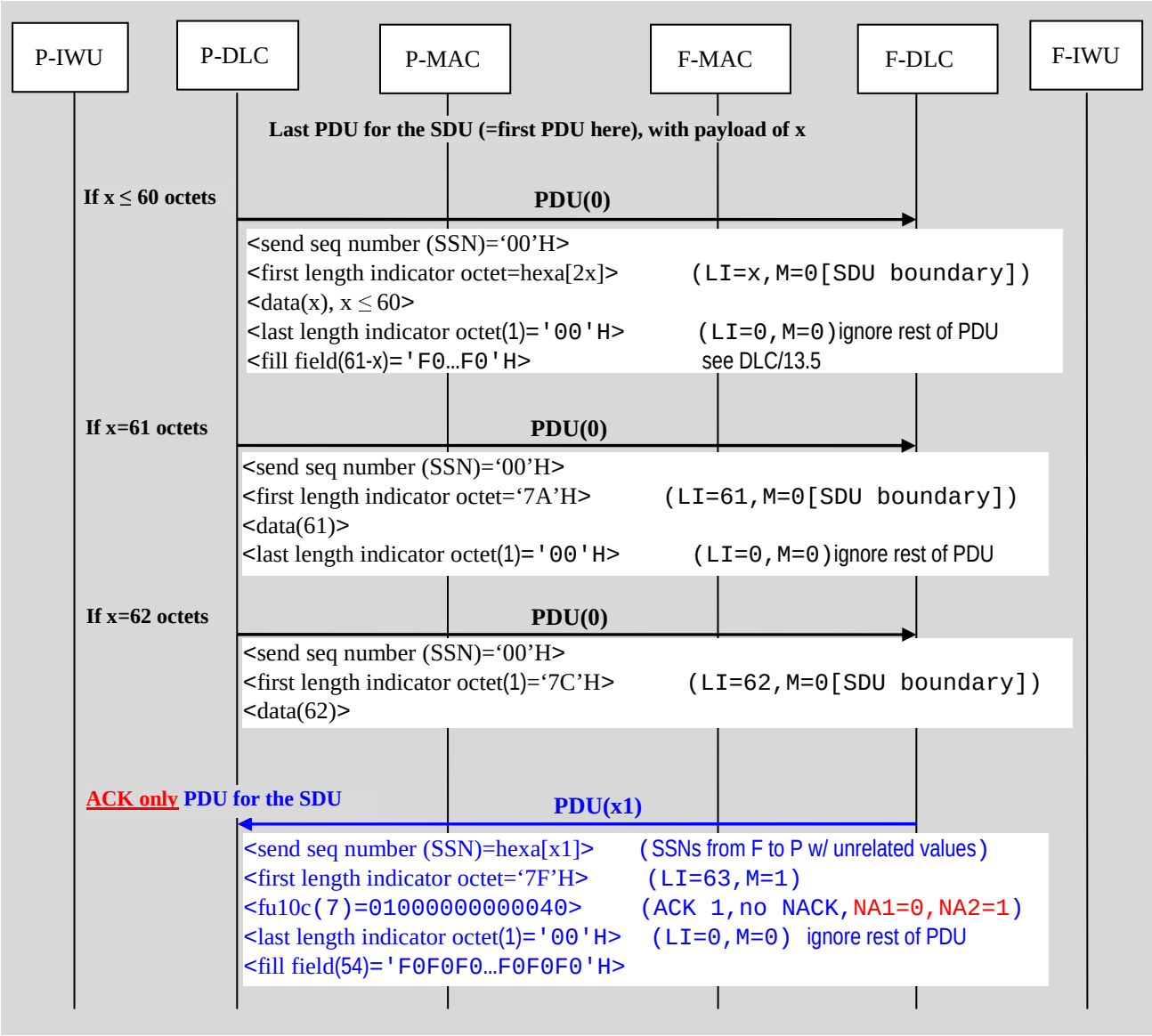


Figure F.4: Successful very small PDU transmission

F.3.3 Use case 3 - one PDU lost

Figure F.5 shows the use case of a lost PDU.

Send 1 SDU of size 'max SDU size' (SDU_length=752=12 × 62 + 8).

Number of PDUs needed: 13 (window size 16 not reached).

PDU(5) not received or not well received.

One ACK + one NACK first, then one ACK

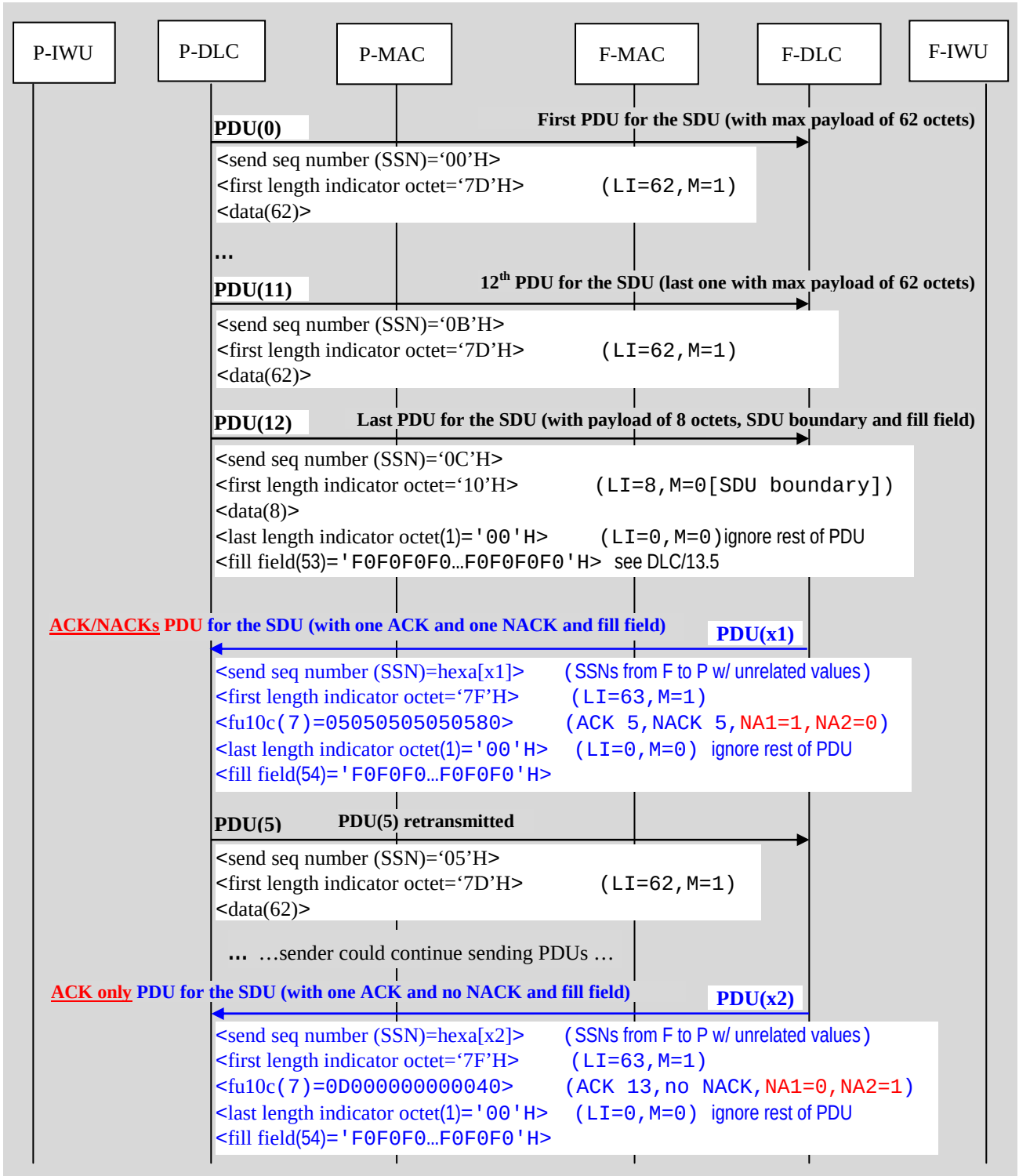


Figure F.5: Lost PDU

F.3.4 Use case 4 - many PDUs lost - two FU10c frames in a row

Figure F.6 shows the use case of may lost PDUs (two FU10c frames in a row).

Send 1 SDU of size 14 × 62 + 8 = 876.

Number of PDUs needed: 15 (window size 16 not reached).

PDU(4, 5, 7, 9, 10, 11,13) not received or not well received.

One ACK + 7 NACKs (2 FU10c frames) first, then one ACK.

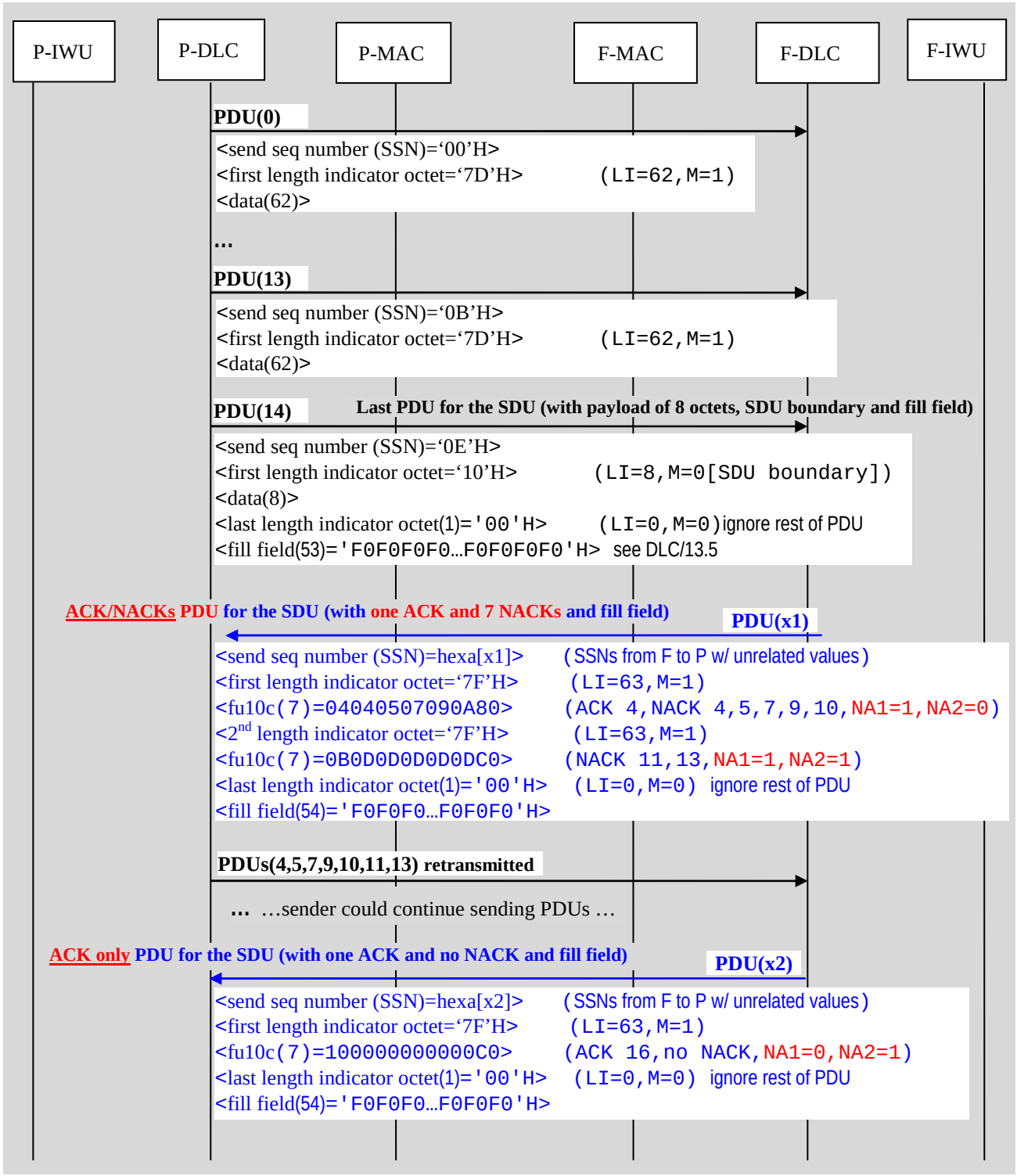


Figure F.6: Many lost PDUs

F.3.5 Use case 5 - FU10c frame + data in the same PDU

Figure F.7 shows the use case of a FU10c frame + data in the same PDU.

Send 1 SDU of size 'max SDU size' (SDU_length=752=12 × 62 + 8).

Number of PDUs needed: 13 (window size 16 not reached). FU10c frame used with the last SDU segment (because there is room left for this), and acknowledging PDU(0) from receiving side (a previous ACK 8 from transmitter side).

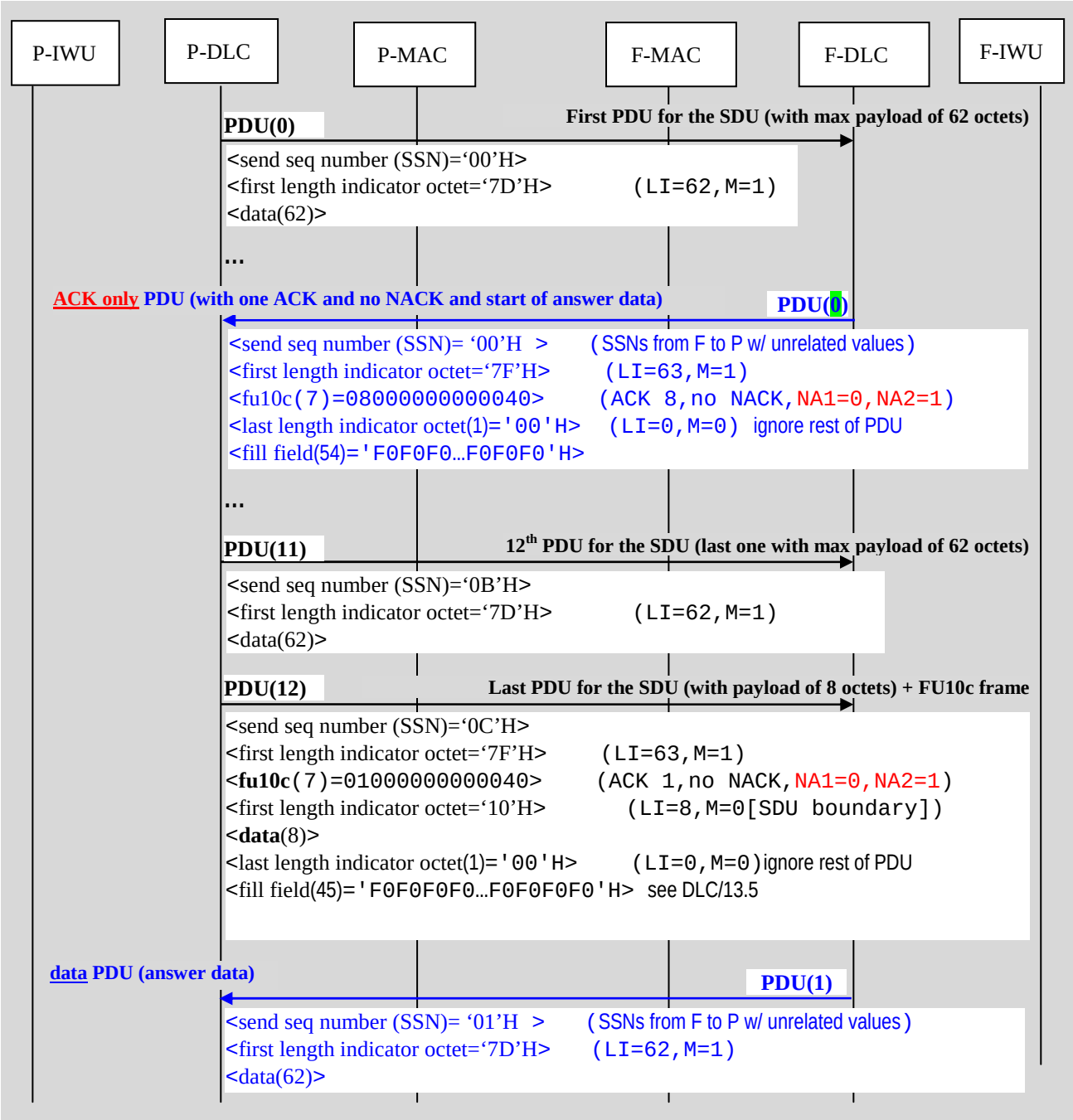


Figure F.7: FU10c frame + data in the same PDU

F.3.6 Use case 6 - Sync frame

Figure F.8 shows the use case of a synch frame transmission.

NOTE: Synchronization frames are not used in SUOTA.

Potential use case: First SDU with 13 PDUs becomes obsolete (How does DLC layer know ?) so that the answer on PT side is no longer needed.

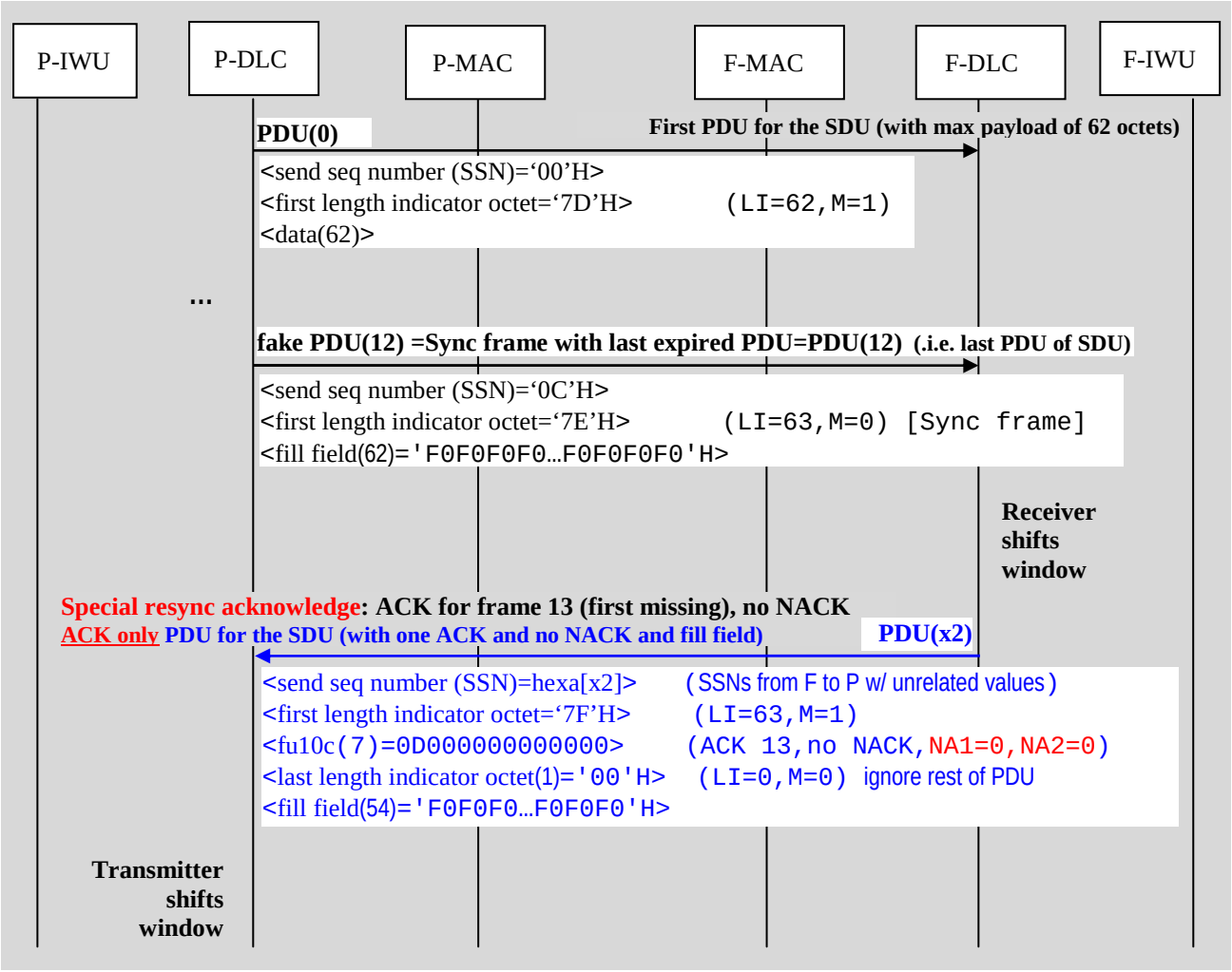


Figure F.8: Sync frame transmission

F.3.7 Use case 7 - Repeating the PDU number in the first NACK

This use case clarifies the difference between the following two FU10c frames:

- ACK/NACK (ACK 255, NACK 255, NACK 0, NACK 1, NACK 1, NACK 1)
- ACK/NACK (ACK 255, NACK 0, NACK 1, NACK 1, NACK 1, NACK 1)

At first glance both frames seem to contain the same information, while the second one saves one octet for the case more NACKs need to be sent. The seeming equivalence relies on the fact that ACK 255 implicitly contains NACK 255, because if PDU(255) is received, then ACK 0 could be used instead).

Example of use of the above first FU10c frame, assuming ACK is sent once for every 4 PDUs received:

```
Tx                                  Rx

253    ------------------------->    253
254    ------------------------->    254
255    --------------X lost
0      --------------X lost
1      --------------X lost
2      ------------------------->    2
3      ------------------------->    3
      <-------------------------     ACK/NACK (ACK 255, NACK 255, NACK 0, NACK 1, NACK 1, NACK 1)
```

It is highly recommended to repeat PDU number 255 as a NACK in order to speed up the retransmission of PDU(255), in the same way as for PDU(0) and PDU(1).

ACK 255 indeed indicates that Rx has received all PDUs until PDU(254) and that the next expected PDU is PDU(255). However, with ACK 255 only, the transmitter does not know yet if PDU(255) was received but not yet acknowledged, or not received at all. As a result, retransmission of ACK 255 will only occur after *ACK reception timeout* or after transmission of the complete current window (assuming by hypothesis that it cannot be advanced) when window retransmission starts.

NOTE: This is to some extend a convention, because Rx having received PDU(255) could be forced by standard to use ACK 0 instead of ACK 255 (so that ACK 255 would always contain an implicit NACK 255). This convention allows Rx to request retransmission of the first PDU of the window without requiring the advance of the window yet (thus decoupling window advance requests from retransmission requests).

NACK 255, 0, 1 indicate that Rx has not received PDUs 255, 0, 1 and hence that Tx will retransmit them as soon as possible.

In other words, if NACK 255 is not included as in "ACK 255, NACK 0, NACK 1, NACK 1, NACK 1, NACK 1", then Tx would (possibly immediately) retransmit only PDU(0) and PDU(1) as it has already transmitted PDU(255), but would wait some time before retransmitting PDU(255) as indicated above, thus temporarily blocking the advance of the window.

F.4 DLC and DPRS relevant and irrelevant clauses

F.4.1 Relevant clauses

F.4.1.1 ETSI EN 300 175-4 [4] (DLC)

clause 11 U-plane service characteristics
clause 11.1 General
clause 11.12 LU10
clause 12.11 FU10 frame structure

NOTE 1: Clause 13.3 is explicitly referenced in clause 11.12.3.1 (Send side procedures) of the same document.

clause 13.3.1.2 Length indicator field format for service LU10
clause 13.3.2.2 Length indicator field parameters for LU10 service
(LI,M) combinations + max payload size + meaning of More bit

NOTE 2: Clause 13.4 is explicitly referenced in clause 11.12.3.1 (Send side procedures) of the same document.

clause 13.4 Sequence number elements
clause 13.4.1 Send sequence number format
clause 13.4.3 Receive sequence number format
clause 14.3.4 [transmission] class 2 procedures

F.4.1.2 ETSI EN 301 649 [15] (DPRS)

clause 11, DLC layer procedures
clause 11.1.1 Window size
clause A.2.1, basic service for LDS class 4, table A.7
clause A.2.2, basic service for LDS class 3, table A.13

F.4.2 Irrelevant clauses

F.4.2.1 ETSI EN 300 175-4 [4] (DLC)

clause 7 (for C-plane)

Annex G (informative): Bibliography

ETSI TR 102 570: "Digital Enhanced Cordless Telecommunications (DECT); Requirements for New Generation DECT".

ETSI EN 300 824: "Digital Enhanced Cordless Telecommunications (DECT); Cordless Terminal Mobility (CTM); CTM Access Profile (CAP)".

ISO/IEC 8073 (1997): "Information technology - Open Systems Interconnection - Protocol for providing the connection-mode transport service".

IETF RFC 2460 (1998): "Internet Protocol, Version 6 (IPv6)".

IETF RFC 1541 (1993): "Dynamic Host Configuration Protocol".

IETF RFC 826 (1982): "An Ethernet Address Resolution Protocol" (STD 37).

IETF RFC 903 (1984): "A Reverse Address Resolution Protocol" (STD 38).

IETF RFC 894 (1984): "A Standard for the Transmission of IP datagrams over Ethernet Networks" (STD 41).

IETF RFC 948 (1985): "Two methods for the transmission of IP datagrams over IEEE 802.3 networks" (STD 43).

IETF RFC 2388: "Returning Values from Forms: multipart/form-data".

History

Document history		
V1.1.1	October 2009	Publication
V1.2.1	December 2014	Publication