MIME media types for DPNSS Objects
<draft-mukundan-sipping-dpnss-00.txt>

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026 [1].

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html

Abstract

This document describes MIME type for application/DPNSS objects for use in SIP [3] applications, according to the rules defined in RFC 2048 [4]. These types can be used to identify Digital Private Network Signaling System 1 (DPNSS 1) [2] objects within a SIP message such as INVITE or INFO, as might be implemented when using SIP in an environment where part of the call involves interworking to DPNSS based Private Networks. RFC 3204 [9] addresses a similar need for ISUP and QSIG.
1. Introduction

DPNSS is an industry standard interface (reference BTNR 188) defined between a Private Branch Exchange (PBX) and an Access Network (AN). DPNSS extends facilities normally only available between extensions on a single PBX to all extensions on PBXs that are connected together in a private network.

There are configurations in which DPNSS encoded signaling information needs to be transported between SIP entities as part of the payload of SIP messages, where the preservation of DPNSS data is necessary for the proper operation of DPNSS based, private network features.

There exists equivalent, currently deployed, mechanisms that allow carriage of DPNSS data transparently over an ISUP network in a SS7 environment where part of the call involves interworking to DPNSS based Private Networks.

The Figure 1 below depicts a typical setup where application/DPNSS media type could be used.

```
+-----+   +-----+   +-----+
|     |   /   |     |
+-----+         +-----+         +-----+
```

Figure 1: Typical Network Configuration for application/DPNSS in SIP

There could also be ‘mixed’ configurations, where DPNSS data may have to be transparently passed between DPNSS based private networks over intervening SIP *and* ISUP networks - where the DPNSS data encapsulated in the ISUP message could be transferred over SIP network using the application/DPNSS media type defined in this document.

This document is specific to the transport of DPNSS data between SIP entities and would not apply to the transportation of DPNSS messages in other applications. This media type is intended for DPNSS application information that is used within the context of a SIP session, and not as a general purpose transport of Switched Circuit Network (SCN) signaling.

The definition of media type for DPNSS application information does not address fully how the non-SIP and SIP entities exchanging messages determine or negotiate compatibility. It is assumed that this is addressed by alternative means such as the configuration of the interworking functions. This document also does not address aspects of interworking messages and parameters between DPNSS and SIP networks.

This is intended to be an IETF approved MIME type, and to be defined through an RFC.
2. The application/DPNSS media type

DPNSS messages are composed of arbitrarily occurring (in terms of the octets’ position in the message), heterogenously encoded International Reference Alphabet (IRA; previously known as International Alphabet No.5 or IA5) [10] binary octets. Further, the IRA binary encoded octets could either be 1B2I-IRA or 3B4I-IRA encoded (refer to BTNR 188 [2] issue 6, Section 4, Annex 2, Appendix I, for the definition of 1B2I/3B4I-IA5 encoding) – all formatted as per the syntax defined in BTNR 188 [2], that is transparent to SIP processing. Hence, the best way to encode application/DPNSS as a MIME object is to use binary encoding. This is in conformance with the restrictions imposed on the use of binary data for MIME (RFC 2045 [5]). It should be noted that the rules mentioned in the RFC 2045 apply to Internet mail messages and not to SIP messages.

DPNSS layer 3 messages, as defined by BTNR 188, do not have a message length field. The application/DPNSS mime encoded data shall be prepended (before DPNSS Message Group/Type message header) with a single binary coded octet message length field that represents the decimal value of the message length. Hence, though a DPNSS layer 3 message as defined by BTNR 188 can only be a maximum of 45 octets, the binary encoded MIME representation of application/DPNSS can be a maximum of 46 octets. The message length field value shall indicate the actual length of DPNSS layer 3 message being MIME encoded – hence, it’s decimal value can range from 1 to 45 based on the DPNSS layer 3 message length.

Though the number of bytes of DPNSS data received can be computed (for example, by doing a byte count of the MIME body content) by the DPNSS layer 3 application at the receiver end, the (apparently redundant) message length field will aid easy implementation of processing the DPNSS data at the receiver end.

A typical implementation would remove the message length octet before passing on the DPNSS data to the DPNSS layer 3 application; the message length can be passed on to the DPNSS layer 3 application using a primitive of local significance.

Refer section 4 – ‘Illustrative example’ for additional details.

2.1 Message Buffering Option

BTNR 188 specifies the DPNSS layer 3 message length to be a maximum of 45 octets. This max limit on the layer 3 message length is arrived at based on the max limit set on the Information field of the UI(C) (Unnumbered Information-Complete), DPNSS layer 2 frame. However, for communication between two peer DPNSS layer 3 applications over an intervening SIP network, it would be desirable to buffer multiple DPNSS ‘incomplete’ (‘incomplete’ as defined in BTNR 188) messages (e.g., buffer multiple EEM-Is; EEM-Is are DPNSS layer 3 End-to-End incomplete messages) at the sender’s end and transfer it as a single
DPNSS ‘complete’ (‘complete’ as defined in BTNR 188) or ‘incomplete’ message (e.g., as a single EEM-C or EEM-I; an EEM-C is a DPNSS layer 3 End-to-End complete message) in the MIME body.

It is up to the sending and receiving DPNSS layer 3 application to handle the ‘oversized’ DPNSS layer 3 messages – for e.g., the receiving DPNSS layer 3 application could fragment the ‘oversized’ DPNSS ‘complete’ message to its constituent ‘incomplete’ messages at the start of message processing.

It is assumed that the availability of support for this buffering option would be known beforehand by way of configuration (e.g., BUFFERING = YES/NO provisioned for a DPNSS network or interface) or other alternative means.

This simple buffering method would improve the real-time performance for DPNSS layer 3 message processing and reduces the number of SIP messages required (and consequently the signaling traffic and associated overheads). For example, it would be more efficient to send one DPNSS layer 3 EEM-C of 225 octets as a MIME object in one SIP message instead of sending five DPNSS layer 3 EEM-Is of 45 octets each as MIME object in 5 SIP messages.

Hence, with the Message Buffering option, the binary encoded MIME representation of application/DPNSS can be a maximum of 255 octets. The message length field value shall indicate the actual length of buffered DPNSS layer 3 message being MIME encoded – hence, it’s decimal value can range from 1 to 255 based on the buffered DPNSS layer 3 message length.

The buffering shall be applicable only to ‘incomplete’ DPNSS layer 3 messages. For example, multiple EEM-Is can be buffered and sent as one EEM-C or as one EEM-I (if the length of the buffered EEM-Is were to exceed 255 octets); buffering shall not be applied to multiple EEM-Cs or fragments of EEM-Is and/or EEM-Cs.

3. The DPNSS Media Type

This media type is defined by the following information:

- Media type name: application
- Media subtype name: DPNSS
- Required parameters: none
- Optional parameters: version
- Encoding scheme: binary
- Security considerations: See section 5.

The use of the ‘version’ parameter allows identification of different versions (indicative of the issue of BTNR 188 publication) of DPNSS protocol specification. This allows for correct interoperation amongst DPNSS PBXs supporting different issues of the DPNSS protocol.

Table 1 below is a list of protocol issues supported by the ‘application/DPNSS’ media type.
Table 1: DPNSS Protocol issues

<table>
<thead>
<tr>
<th>version</th>
<th>protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>DPNSS protocol issue 1 (BTNR 188, May 1983)</td>
</tr>
<tr>
<td>2.00</td>
<td>DPNSS protocol issue 2 (BTNR 188, February 1984)</td>
</tr>
<tr>
<td>3.00</td>
<td>DPNSS protocol issue 3 (BTNR 188, September 1984)</td>
</tr>
<tr>
<td>4.00</td>
<td>DPNSS protocol issue 4 (BTNR 188, March 1986)</td>
</tr>
<tr>
<td>5.00</td>
<td>DPNSS protocol issue 5 (BTNR 188, December 1989)</td>
</tr>
<tr>
<td>6.00</td>
<td>DPNSS protocol issue 6 (BTNR 188, January 1995)</td>
</tr>
</tbody>
</table>

To support new issues of DPNSS protocol specification, that may be published in the future, the ‘version’ field shall be contiguously increasing floating point values and shall have one to one correspondence with the BTNR DPNSS protocol specification issue number. For example, if a new DPNSS protocol specification issue 7 were to be released in June 2003, then the ‘version’ value will be 7.00; if the one released next were to have the issue number 7.1, then the ‘version’ value will be 7.10; the one with sub-issue number 7.11 (in case there were to be sub-issues as well in the future) will have ‘version’ value of 7.11 and so on.

The Content-Disposition header [7] may be included to describe how the encapsulated DPNSS is to be processed, and in particular what the handling should be if the received Content-Type is not recognized. The default disposition-type for a DPNSS message body is "signal". This type indicates that the body part contains signaling information associated with the session, but does not describe the session.

Supplementing the description of the Content-Disposition header in [7], as well as any characterization of the Content-Disposition header in the SIP standard, is the following BNF describing disposition-types and disposition-params that may be used in the header of DPNSS MIME bodies.

```
Content-Disposition   =  "Content-Disposition" ":"
                      disposition-type *( ;" disposition-param )
disposition-type      =  "signal" |  disp-extension-token
disposition-param     =  "handling" "="
                      ( "optional" | "required" | other-handling )
                      |   generic-param
other-handling        =  token
disp-extension-token  =  token
```

The following is how a typical header would look:

```
Content-Type: application/DPNSS; version=6.00
Content-Disposition: signal; handling=optional
```

The handling parameter, handling-parm, describes how the SIP User Agent Server (UAS) process should react if it receives a message body whose content type or disposition type it does not understand. If the parameter has the value "optional", the UAS MUST ignore the message body; if it has the value "required", the UAS MUST return 415 (Unsupported Media Type). If the handling parameter is missing, the
4. Illustrative example

SIP message format requires a Request line followed by Header lines followed by a CRLF separator followed by the message body. To illustrate the use of the ‘application/DPNSS’ media type, below is an INVITE message which has the originating SDP information and an encapsulated DPNSS Initial Service Request Message (ISRM).

Note that the two payloads are demarcated by the boundary parameter (specified in RFC 2046 [6]) which in the example has the value "unique-mime-boundary".

```
INVITE sip:91805538301@k1.wipro.com SIP/2.0
Via: SIP/2.0/UDP ec.wipro.com
From: sip:91808520408@ec.wipro.com
To: sip:91805538301@k1.wipro.com
Call-ID: EC4323458918276550455555@ec.wipro.com
CSeq: 4567 INVITE
Contact: <sip:dummy@wipro.com>
Content-Length: 384
Content-Type: multipart/mixed; boundary=unique-mime-boundary
MIME-Version: 1.0

--unique-mime-boundary
Content-Type: application/SDP; charset=ISO-10646

v=0
o=dummy 2890844526 2890842807 IN IP4 166.218.71.81
s=SDP seminar
c=IN IP4 MG121.wipro.com
t= 2873397496 2873404696
m=audio 9092 RTP/AVP 0 3 4
--unique-boundary-1
Content-Type: application/DPNSS; version=1;
Content-Disposition: signal; handling=optional

24 00 10 2A 31 23 2A 35 30 2A 38 37 32 38 38 35
38 23 35 39 30 32 33 36 30
--unique-mime-boundary
```

In this example:

- **Message length = 24**
- **Message type = 00 (ISRM)**
- **Service Indication Codre (SIC) = 10 (64 kbs A-law Speech)**
- **Calling Line Category (CLC) = 1 (Ordinary)**
- **Originating Line Identity (OLI) = 872 8858**
- **Destination Address = 590 2360**

Typically, DPNSS layer 3 messages are represented using formal IA5 notation; hence, the DPNSS layer 3 message used in the example can be represented (leaving out the message type and SIC field) as:

```
*1#*50*8728858#5902360
```
Note: Since binary encoding is used in the MIME body for the DPNSS payload, each byte is encoded as a byte, and not as a two-character hex or decimal representation. In the example above, decimal representation of integer values are used to represent the message length, message type and the SIC fields; rest of the fields are represented as 1B2A-IA5 because a literal (binary) encoding of those bytes would have been confusing and unreadable.

5. Security considerations

Information contained in the DPNSS MIME body may include sensitive customer information, potentially requiring use of encryption.

Security mechanisms are provided in RFC 3261 (SIP - Session Initiation Protocol) and should be used as appropriate for both the SIP message and the encapsulated DPNSS body.

6. IANA considerations

This document registers the "application/DPNSS" MIME media type.

Registrations for the ‘version’ symbols used within the DPNSS MIME type must specify a definitive specification reference, identifying a particular issue of the specification, to which the new symbol shall refer.

Note that where a specification is fully peer-to-peer backwards compatible with a previous issue (i.e., the compatibility mechanism is supported by both), then there is no need for separate symbols to be registered. The symbol for the original specification should be used to identify backwards-compatible upgrades of that specification as well.

7. References

7.1 Normative References


7.2 Informative References


8. Authors Addresses

Ranjith Mukundan Phone: +91-80-8520408 x2108
Wipro Technologies Email: ranjith.mukundan@wipro.com
72, Electronics City,
Hosur Main Road,
Bangalore 561229
India

Venkatesh Seshasayee Phone: +91-80-8520408 x2109
Wipro Technologies Email: venkatesh.seshasayee@wipro.com
72, Electronics City,
Hosur Main Road,
Bangalore 561229
India

Ken Morneault Phone: +1-703-484-3323
Cisco Systems Inc. Email: kmorneau@cisco.com
13615 Dulles Technology Drive
Herndon, VA. 20171
USA

Ravi Shiroor Phone: +91-120-2585040 x427
Aplion Networks Email: rshiroor@aplion.stpn.soft.net
B-21 Sector 58
Noida 201301
India

Narsimloo Mangalpally Phone: +1-613-967-5034
Nortel Networks Email: narsim@nortelnetworks.com
250 Sidney Street
Belleville, Ontario K8P 3Z3
Canada