Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 17, 2019

B. Haberman
JHU APL
J. Levine
Taughannock Networks

July 16, 2018

Using a DNS SRV Record to Locate an X.509 Certificate Store
draft-bhjl-x509-srv-04

Abstract

This document describes a method to allow parties to locate X.509 certificate stores with Domain Name System Service records in order to retrieve certificates and certificate revocation lists. The primary purpose of such retrievals is to facilitate the association of X.509 and PGP public keys with e-mail addresses to allow for encrypted e-mail exchanges.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on January 17, 2019.

Copyright Notice

Table of Contents

1. Introduction

X.509 and PGP public keys can be used to encrypt or sign e-mail messages. In order to verify a sender's signature or encrypt an e-mail, the e-mail client needs to locate the appropriate public key. The X.509-based Public Key Infrastructure (PKI) [RFC5280] provides the necessary services to allow for the retrieval of certificates and certificate revocation lists, but lacks the discovery mechanism needed to associate e-mail domains with specific PKI servers.

This document specifies an approach that uses a Domain Name System (DNS) Service Record (SRV) that allows mail service providers to advertise the X.509 or PGP certificate store [RFC4387] that contains certificates and certificate revocation lists for their e-mail users. Additionally, this document specifies the appropriate query strings to use when accessing the certificate store.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. Service Record Format

The general format of a DNS SRV record is documented in [RFC2782] as:

```
_Service._Proto.Name TTL Class SRV Priority Weight Port Target
```

To support the advertisement of an X.509 certificate store, service providers publish an SRV record for the certificates service with the appropriate parameters, as described in [RFC4387], section 3.2. An example of such an SRV record is:

```
_certificates._tcp 86400 IN SRV 0 0 443 certs.example.com
```

The parameters of the DNS SRV record are set based on the operational needs of the service provider. The DNS SRV record SHOULD be signed via DNSSEC [RFC4033][RFC4034]. The server MUST be an https server and will typically use port 443. The certificate of the https server SHOULD be validated by a DNSSEC signed TLSA record, and MAY also be validated by a certificate authority.

3. Certificate Store Queries

To retrieve an X.509 S/MIME certificate, the attribute type is "uri", and the URI is constructed using the path described in [RFC4387], Section 3.3, specifically "/certificates/search.cgi". Using the SRV record above to look up a certificate for bob@example.com, the URI would be:

```
https://certs.example.com/certificates/search.cgi?uri=bob%40example.com
```

X.509 certificate stores MUST support the uri attribute and MAY support other attributes.

To retrieve a PGP certificate, the attribute type is "email", and the URI is constructed using the path described in [RFC4387], Section 3.3, specifically "/pgpkeys/search.cgi". Using the SRV record above to look up a certificate for bob@example.com, the URI would be:

```
https://certs.example.com/pgpkeys/search.cgi?email=bob%40example.com
```

PGP certificate stores MUST support the email attribute and MAY support other attributes.

4. Name Matching

SMTP [RFC5321] specifies that the local part of a mailbox is interpreted only by the mailbox domain itself. This document does not update or modify that document.

If a certificate store has no certificate with an e-mail address that matches the uri or email attribute in a retrieval request, but it does have a certificate with an e-mail address that the mailbox domain treats similarly to the requested address, the server MAY return that certificate. The definition of what is sufficiently similar is a matter of local policy, but the intention is that a human correspondent would consider the the two addresses to deliver mail to the same person or entity.

5. Certificate Validation

The certificate is returned as a blob of binary data. If multiple certificates are returned, the response is encoded as multipart/mixed as described in [RFC4387] section 2.

X509 S/MIME certificates are validated by checking for a signature by a Certificate Authority (CA) that is acceptable to the validating party. This specification defines an additional validation technique. The domain MAY publish validation certificates using TLSA records at the name _smimeca._tcp. The TLSA records MUST have PKIX-TA or DANE-TA usage[RFC7218]. A validation certificate published by a domain MUST NOT be used to validate certificates other than those with e-mail addresses in that domain.

Since the relationship between a domain and its mailbox users is in general unknown to correspondents, a client applies a local policy to decide whether to use a S/MIME certificate validated only by a signing certificate published by the domain.

PGP certificates are validated by the PGP web of trust. A domain can endorse the certificates it publishes by signing them with a signature of postmaster@<domain>. Since the relationship between a domain and its mailbox users is in general unknown to correspondents. a client applies a local policy to decide whether to use a PGP certificate retrieved from a certificate server. This policy would typically be the same one used to decide whether to use a certificate retrieved from a traditional PGP key server.

6. Certificate use and cacheing

Clients SHOULD cache responses to queries as advised by http cache headers. This includes both returned certificates, and 404 failures saying that an address (or other search key) has no certificate.

S/MIME keys retrieved from the certificate store SHOULD NOT be used for validation of signatures on incoming mail without further validation of the certificate. S/MIME signed mail includes a copy of the signing certificate which, if it can be validated, typically would be used instead.

7. Security Considerations

Certificate queries could be used to try to validate lists of e-mail addresses. This is essentially the same problem that mail servers face with VRFY, EXPN, and RCPT TO probes, and the same countermeasures would apply, such as rate limiting, blacklisting abusive clients, and returning fake results for non-existent addresses.

DNSSEC signatures on the SRV record and the https server certificate ensure that any keys retrieved by the technique described in this document are the ones published by the domain’s management. But since correspondents often do not know the relationship between a domain and its mailbox users, it would be imprudent to assume that such certificates are in fact ones issued to or used by mailbox recipients or to assume that mail encrypted using the certificates will be readable only by the intended recipient without further information about the certificates.

A domain could publish man-in-the-middle certificates that allowed it to decode and read mail, and perhaps re-encrypt it using different certificates used by the recipients. In some cases this would be entirely legitimate, e.g., a financial institution that is required to log all of its employees’ correspondence. In other cases, it could be intrusive or improper surveillance of the contents of users’ mail. Identifying or describing the relationship between a domain and its mail users is beyond the scope of this document.

8. IANA Considerations

IANA is requested to update two entries in the Service Name and Transport Protocol Port Number Registry.

8.1. Certificates service

Service Name: certificates

Transport Protocol(s): tcp

Assignee: IESG

Contact: <chair@ietf.org>

Description: Server for S/MIME and PGP certificates

Reference: [this document]

Port Number: none

Service Code: none

Known Unauthorized Uses: none

8.2. Smimeca service

Service Name: simeca

Transport Protocol(s): tcp

Assignee: IESG

Contact: <chair@ietf.org>

Description: Per-domain authority certificate for S/MIME certificates

Reference: [this document]

Port Number: none

Service Code: none

Known Unauthorized Uses: none

9. Acknowledgements

We thank Wei Chuang, Nicolas Lidzborski, and Andreas Schulze for comments and suggestions.

10. Normative References

[RFC2782] Gulbrandsen, A., Vixie, P., and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)", RFC 2782, DOI 10.17487/RFC2782, February 2000, <https://www.rfc-editor.org/info/rfc2782>.

[RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, DOI 10.17487/RFC4033, March 2005, <https://www.rfc-editor.org/info/rfc4033>.

[RFC4034] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Resource Records for the DNS Security Extensions", RFC 4034, DOI 10.17487/RFC4034, March 2005, <https://www.rfc-editor.org/info/rfc4034>.

[RFC4387] Gutmann, P., Ed., "Internet X.509 Public Key Infrastructure Operational Protocols: Certificate Store Access via HTTP", RFC 4387, DOI 10.17487/RFC4387, February 2006, <https://www.rfc-editor.org/info/rfc4387>.

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008, <https://www.rfc-editor.org/info/rfc5280>.

[RFC5321] Klensin, J., "Simple Mail Transfer Protocol", RFC 5321, DOI 10.17487/RFC5321, October 2008, <https://www.rfc-editor.org/info/rfc5321>.

[RFC7218] Gudmundsson, O., "Adding Acronyms to Simplify Conversations about DNS-Based Authentication of Named Entities (DANE)", RFC 7218, DOI 10.17487/RFC7218, April 2014, <https://www.rfc-editor.org/info/rfc7218>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

Authors' Addresses

Brian Haberman
Johns Hopkins University Applied Physics Lab

Email: brian@innovationslab.net

John Levine
Taughannock Networks
PO Box 727
Trumansburg, NY 14886

Phone: +1 831 480 2300
Email: standards@taugh.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: 4 January 2021

C. Bormann
Universität Bremen TZI
S. Leonard
Penango, Inc.

3 July 2020

Concise Binary Object Representation (CBOR) Tags for Object Identifiers
draft-bormann-cbor-tags-oid-07

Abstract

The Concise Binary Object Representation (CBOR, RFC 7049) is a data format whose design goals include the possibility of extremely small code size, fairly small message size, and extensibility without the need for version negotiation.

The present document defines CBOR tags for object identifiers (OIDs). It is intended as the reference document for the IANA registration of the CBOR tags so defined.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on 4 January 2021.

Copyright Notice

Table of Contents

1. Introduction

The Concise Binary Object Representation (CBOR, [RFC7049]) provides for the interchange of structured data without a requirement for a pre-agreed schema. RFC 7049 defines a basic set of data types, as well as a tagging mechanism that enables extending the set of data types supported via an IANA registry.

The present document defines CBOR tags for object identifiers (OIDs, [X.660]), which many IETF protocols carry. The ASN.1 Basic Encoding Rules (BER, [X.690]) specify binary encodings of both (absolute) object identifiers and relative object identifiers. The contents of these encodings can be carried in a CBOR byte string. This document defines two CBOR tags that cover the two kinds of ASN.1 object identifiers encoded in this way. The tags can also be applied to arrays and maps for more articulated identification purposes. It is intended as the reference document for the IANA registration of the tags so defined.

1.1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

The terminology of RFC 7049 applies; in particular the term "byte" is used in its now customary sense as a synonym for "octet".

2. Object Identifiers

The International Object Identifier tree [X.660] is a hierarchically managed space of identifiers, each of which is uniquely represented as a sequence of primary integer values [X.680]. While these sequences can easily be represented in CBOR arrays of unsigned integers, a more compact representation can often be achieved by adopting the widely used representation of object identifiers defined in BER; this representation may also be more amenable to processing by other software making use of object identifiers.

BER represents the sequence of unsigned integers by concatenating self-delimiting [RFC6256] representations of each of the primary integer values in sequence.

ASN.1 distinguishes absolute object identifiers (ASN.1 Type "OBJECT IDENTIFIER"), which begin at a root arc ([X.660] Clause 3.5.21), from relative object identifiers (ASN.1 Type "RELATIVE-OID"), which begin relative to some object identifier known from context ([X.680] Clause 3.8.63). As a special optimization, BER combines the first two integers in an absolute object identifier into one numeric identifier by making use of the property of the hierarchy that the first arc has only three integer values (0, 1, and 2), and the second arcs under 0 and 1 are limited to the integer values between 0 and 39. (The root arc "joint-iso-itu-t(2)" has no such limitations on its second arc.) If X and Y are the first two integers, the single integer actually encoded is computed as:

```
X * 40 + Y
```

The inverse transformation (again making use of the known ranges of X and Y) is applied when decoding the object identifier.

Since the semantics of absolute and relative object identifiers differ, this specification defines two tags:

Tag TBD111: tags a byte string as the [X.690] encoding of an absolute object identifier (simply "object identifier" or "OID").

Tag TBD110: tags a byte string as the [X.690] encoding of a relative object identifier (also "relative OID"). Since the encoding of each number is the same as for [RFC6256] Self-Delimiting Numeric Values (SDNVs), this tag can also be used for tagging a byte string that contains a sequence of zero or more SDNVs.

2.1. Requirements on the byte string being tagged

A byte string tagged by TBD111 or TBD110 MUST be a syntactically valid BER representation of an object identifier: A concatenation of zero or more SDNV values, where each SDNV value is a sequence of one or more bytes that all have their most significant bit set, except for the last byte, where it must be unset; the first byte of each SDNV cannot be 0x80 (which would be a leading zero in SDNV's base-128 arithmetic).

In other words:

* its first byte, and any byte that follows a byte that has the most significant bit unset, MUST NOT be 0x80 (this requirement excludes expressing the primary integer values with anything but the shortest form)

* its last byte MUST NOT have the most significant bit set (this requirement excludes an incomplete final primary integer value)

If either of these invalid conditions are encountered, the tag is invalid.

[X.680] restricts RELATIVE-OID values to have at least one arc, i.e., their encoding would have at least one SDNV. This specification permits empty relative object identifiers; they may still be excluded by application semantics.

To enable the search for specific object ID values, it is RECOMMENDED that definite length encoding (see Section 2.2.2 of [RFC7049]) is used for the byte strings used as tag content for these tags.

The valid set of byte strings can also be expressed using regular expressions on bytes, using no specific notation but resembling [PCRE]. Unlike typical regular expressions that operate on character sequences, the following regular expressions take bytes as their domain, so they can be applied directly to CBOR byte strings.

For byte strings with tag TBD111:

```
"/^(([\x81-\xFF][\x80-\xFF]*)?[\x00-\x7F])+$/"
```

For byte strings with tag TBD110:

```
"/^(([\x81-\xFF][\x80-\xFF]*)?[\x00-\x7F])*$/"
```

A tag with tagged content that does not conform to the applicable regexp is invalid.

3. Examples

3.1. Encoding of the SHA-256 OID

ASN.1 Value Notation: { joint-iso-itu-t(2) country(16) us(840) organization(1) gov(101) csor(3) nistalgorithm(4) hashalgs(2) sha256(1) }

Dotted Decimal Notation: 2.16.840.1.101.3.4.2.1

```
06                                  # UNIVERSAL TAG 6
   09                               # 9 bytes, primitive
      60 86 48 01 65 03 04 02 01    # X.690 Clause 8.19
#      |   840  1  |  3  4  2  1     show component encoding
#   2.16         101
```

Figure 1: SHA-256 OID in BER

```
D8 6F                               # tag(111)
   49                               # 0b010_01001: mt 2, 9 bytes
      60 86 48 01 65 03 04 02 01    # X.690 Clause 8.19
```

Figure 2: SHA-256 OID in CBOR

3.2. Encoding of a MIB Relative OID

Given some OID (e.g., "lowpanMib", assumed to be "1.3.6.1.2.1.226" [RFC7388]), to which the following is added:

ASN.1 Value Notation: { lowpanObjects(1) lowpanStats(1) lowpanOutTransmits(29) }

Dotted Decimal Notation: .1.1.29

```
0D                                  # UNIVERSAL TAG 13
   03                               # 3 bytes, primitive
      01 01 1D                      # X.690 Clause 8.20
#      1  1 29                        show component encoding
```

Figure 3: MIB relative object identifier, in BER

```
D8 6E                               # tag(110)
   43                               # 0b010_01001: mt 2 (bstr), 3 bytes
      01 01 1D                      # X.690 Clause 8.20
```

Figure 4: MIB relative object identifier, in CBOR

This relative OID saves seven bytes compared to the full OID encoding.

4. Discussion

Staying close to the way object identifiers are encoded in ASN.1 BER makes back-and-forth translation easy; otherwise we would choose a more efficient encoding. Object identifiers in IETF protocols are serialized in dotted decimal form or BER form, so there is an advantage in not inventing a third form. Also, expectations of the cost of encoding object identifiers are based on BER; using a different encoding might not be aligned with these expectations. If additional information about an OID is desired, lookup services such as the OID Resolution Service (ORS) [X.672] and the OID Repository [OID-INFO] are available.

5. Tag Factoring with OID Arrays and Maps

TBD111 and TBD110 can tag CBOR arrays and maps. The idea is that the tag is factored out from each individual byte string; the tag is placed in front of the array or map instead. The tags TBD111 and TBD110 are left-distributive.

When the TBD111 or TBD110 tag is applied to an array, it means that the respective tag is imputed to all items in the array that are byte strings. For example, when the array is tagged with TBD111, every array item that is a binary string is an OID.

When the TBD111 or TBD110 tag is applied to a map, it means that the respective tag is imputed to all keys in the map that are byte strings. The values in the map are not considered specially tagged.

Array and map nesting is permitted. For example, a 3-dimensional array of OIDs can be composed by using a single TBD111 tag, followed by an array of arrays of arrays of binary strings. All such binary strings are considered OIDs.
// That was part of the original proposal. I find it hard to imagine
// how to stop the influence of the tag deep into a nested structure.
// That's why I would rather limit this to one level (no nesting).
// But see the Figure below, which needs a nesting of two. Please
// discuss.

6. Applications and Examples of OIDs

6.1. X.500 Distinguished Name

Consider the X.500 distinguished name:

Attribute Types	Attribute Values
c (2.5.4.6)	US
l (2.5.4.7) s (2.5.4.8) postalCode (2.5.4.17)	Los Angeles CA 90013
street (2.5.4.9)	532 S Olive St
businessCategory (2.5.4.15) buildingName (0.9.2342.19200300.100.1.48)	Public Park Pershing Square

Table 1: Example X.500 Distinguished Name

Table 1 has four "relative distinguished names" (RDNs). The country and street RDNs are single-valued. The second and fourth RDNs are multi-valued.

The equivalent representations in CBOR diagnostic notation and CBOR are:

```
111([{ h'550406': "US" },
     { h'550407': "Los Angeles", h'550408': "CA",
       h'550411': "90013" },
     { h'550409': "532 S Olive St" },
     { h'55040f': "Public Park",
       h'0992268993f22c640130': "Pershing Square" }])
```

Figure 5: Distinguished Name, in CBOR Diagnostic Notation

```
d8 6f                                   # tag(111)
   84                                   # array(4)
      a1                                # map(1)
         43 550406                      # 2.5.4.6 (4)
         62                             # text(2)
            5553                        # "US"
      a3                                # map(3)
         43 550407                      # 2.5.4.7 (4)
         6b                             # text(11)
            4c6f7320416e67656c6573      # "Los Angeles"
         43 550408                      # 2.5.4.8 (4)
         62                             # text(2)
            4341                        # "CA"
         43 550411                      # 2.5.4.17 (4)
         65                             # text(5)
            3930303133                  # "90013"
      a1                                # map(1)
         43 550409                      # 2.5.4.9 (4)
         6e                             # text(14)
            3533322053204f6c697665205374   # "532 S Olive St"
      a2                                # map(2)
         43 55040f                      # 2.5.4.15 (4)
         6b                             # text(11)
            5075626c6963205061726b      # "Public Park"
         4a 0992268993f22c640130    # 0.9.2342.19200300.100.1.48 (11)
         6f                             # text(15)
            5065727368696e6720537175617265 # "Pershing Square"
```

Figure 6: Distinguished Name, in CBOR (109 bytes)

(This example encoding assumes that all attribute values are UTF-8 strings, or can be represented as UTF-8 strings with no loss of information.)

7. CDDL Control Operators

CDDL specifications may want to specify the use of SDNVs or SDNV sequences (as defined for the tag content for TBD110). This document introduces two new control operators that can be applied to a target value that is a byte string:

* ".sdnv", with a control type that contains unsigned integers. The byte string is specified to be encoded as an [RFC6256] SDNV (BER encoding) for the matching values of the control type.

* ".sdnvseq", with a control type that contains arrays of unsigned integers. The byte string is specified to be encoded as a sequence of [RFC6256] SDNVs (BER encoding) that decodes to an array of unsigned integers matching the control type.

Figure 7 shows an example for the use of ".sdnvseq" for a part of a structure using OIDs that could be used in Figure 6.

```
// We could define another control operator that includes the X*40+Y
// magic, so the example can actually use "[2, 5, 4, 6]". We could
// also add an operator that parses dotted decimal integer sequences,
// so we can use "2.5.4.6". I don't see a strong reason for that.

country-rdn = {country-oid => country-value}
country-oid = bytes .sdnvseq [85, 4, 6]
country-value = text .size 2
```

Figure 7: Using .sdnvseq

8. IANA Considerations

8.1. CBOR Tags

IANA is requested to assign the CBOR tags in Table 2, with the present document as the specification reference.

Tag	Data Item	Semantics
TBD111	multiple	object identifier (BER encoding)
TBD110	multiple	relative object identifier (BER encoding); SDNV [RFC6256] sequence

Table 2: Values for New Tags

8.2. CDDL Control Operators

IANA is requested to assign the CDDL Control Operators in Table 3, with the present document as the specification reference.

Name	Reference
.sdnv	[this document, Section 7]
.sdnvseq	[this document, Section 7]

Table 3: New CDDL Operators

9. Security Considerations

The security considerations of RFC 7049 apply.

The encodings in Clauses 8.19 and 8.20 of [X.690] are quite compact and unambiguous, but MUST be followed precisely to avoid security pitfalls. In particular, the requirements set out in Section 2.1 of this document need to be followed; otherwise, an attacker may be able to subvert a checking process by submitting alternative representations that are later taken as the original (or even something else entirely) by another decoder supposed to be protected by the checking process.

OIDs and relative OIDs can always be treated as opaque byte strings. Actually understanding the structure that was used for generating them is not necessary, and, except for checking the structure requirements, it is strongly NOT RECOMMENDED to perform any processing of this kind (e.g., converting into dotted notation and back) unless absolutely necessary. If the OIDs are translated into other representations, the usual security considerations for non-trivial representation conversions apply; the primary integer values are unlimited in range.

9.1. Conversions Between BER and Dotted Decimal Notation

[PKILCAKE] uncovers exploit vectors for the illegal values above, as well as for cases in which conversion to or from the dotted decimal notation goes awry. Neither [X.660] nor [X.680] place an upper bound on the range of unsigned integer values for an arc; the integers are arbitrarily valued. An implementation SHOULD NOT attempt to convert each component using a fixed-size accumulator, as an attacker will certainly be able to cause the accumulator to overflow. Compact and efficient techniques for such conversions, such as the double dabble algorithm [DOUBLEDABBLE] are well-known in the art; their application to this field is left as an exercise to the reader.

10. References

10.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC6256] Eddy, W. and E. Davies, "Using Self-Delimiting Numeric Values in Protocols", RFC 6256, DOI 10.17487/RFC6256, May 2011, <https://www.rfc-editor.org/info/rfc6256>.

[RFC7049] Bormann, C. and P. Hoffman, "Concise Binary Object Representation (CBOR)", RFC 7049, DOI 10.17487/RFC7049, October 2013, <https://www.rfc-editor.org/info/rfc7049>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[X.660] International Telecommunications Union, "Information technology -- Procedures for the operation of object identifier registration authorities: General procedures and top arcs of the international object identifier tree", ITU-T Recommendation X.660, July 2011.

[X.680] International Telecommunications Union, "Information technology -- Abstract Syntax Notation One (ASN.1): Specification of basic notation", ITU-T Recommendation X.680, August 2015.

[X.690] International Telecommunications Union, "Information technology -- ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)", ITU-T Recommendation X.690, August 2015.

10.2. Informative References

[DOUBLEDABBLE] Gao, S., Al-Khalili, D., and N. Chabini, "An improved BCD adder using 6-LUT FPGAs", DOI 10.1109/newcas.2012.6328944, 10th IEEE International NEWCAS Conference, June 2012, <https://doi.org/10.1109/newcas.2012.6328944>.

[OID-INFO] Orange SA, "OID Repository", 2016, <http://www.oid-info.com/>.

[PCRE] Ho, A., "PCRE - Perl Compatible Regular Expressions", 2018, <http://www.pcre.org/>.

[PKILCAKE] Kaminsky, D., Patterson, M., and L. Sassaman, "PKI Layer Cake: New Collision Attacks against the Global X.509 Infrastructure", DOI 10.1007/978-3-642-14577-3_22, Financial Cryptography and Data Security pp. 289-303, 2010, <https://doi.org/10.1007/978-3-642-14577-3_22>.

[RFC7388] Schoenwaelder, J., Sehgal, A., Tsou, T., and C. Zhou, "Definition of Managed Objects for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)", RFC 7388, DOI 10.17487/RFC7388, October 2014, <https://www.rfc-editor.org/info/rfc7388>.

[X.672] International Telecommunications Union, "Information technology -- Open systems interconnection -- Object identifier resolution system", ITU-T Recommendation X.672, August 2010.

Appendix A. Change Log

This section is to be removed before publishing as an RFC.

A.1. Changes from -06 to -07

Reduce the draft back to its basic mandate: Describe CBOR tags for what is colloquially know as ASN.1 Object IDs.

A.2. Changes from -05 to -06

Refreshed the draft to the current date ("keep-alive").

A.3. Changes from -04 to -05

Discussed UUID usage in CBOR, and incorporated fixes proposed by Olivier Dubuisson, including fixes regarding OID nomenclature.

A.4. Changes from -03 to -04

Changes occurred based on limited feedback, mainly centered around the abstract and introduction, rather than substantive technical changes. These changes include:

* Changed the title so that it is about tags and techniques.

* Rewrote the abstract to describe the content more accurately, and to point out that no changes to the wire protocol are being proposed.

* Removed "ASN.1" from "object identifiers", as OIDs are independent of ASN.1.

* Rewrote the introduction to be more about the present text.

* Proposed a concise OID arc.

* Provided binary regular expression forms for OID validation.

* Updated IANA registration tables.

A.5. Changes from -02 to -03

Many significant changes occurred in this version. These changes include:

* Expanded the draft scope to be a comprehensive CBOR update.

* Added OID-related sections: OID Enumerations, OID Maps and Arrays, and Applications and Examples of OIDs.

* Added Tag 36 update (binary MIME, better definitions).

* Added stub/experimental sections for X.690 Series Tags (tag <<X>>) and Regular Expressions (tag 35).

* Added technique for representing sets and multisets.

* Added references and fixed typos.

Authors' Addresses

Carsten Bormann
Universität Bremen TZI
Postfach 330440
D-28359 Bremen
Germany

Phone: +49-421-218-63921
Email: cabo@tzi.org

Sean Leonard
Penango, Inc.
5900 Wilshire Boulevard
21st Floor
Los Angeles, CA, 90036
United States of America

Email: dev+ietf@seantek.com
URI: http://www.penango.com/

Network Working Group
Internet-Draft
Intended status: Informational
Expires: April 27, 2017

G. Deen
NBCUniversal
L. Daigle
Thinking Cat Enterprises LLC

October 24, 2016

Glass to Glass Internet Ecosystem Introduction
draft-deen-daigle-ggie-02

Abstract

This document introduces the Glass to Glass Internet Ecosystem (GGIE). GGIE's purpose is to improve how the Internet is used create and consume video, both amateur and professional, reflecting that the line between amateur and professional video technology is increasingly blurred. Glass to Glass refers to the entire video ecosystem, from the camera lens to the viewing screen. As the name implies, GGIE's scope is the entire video ecosystem from capture, through the steps of editing, packaging, distributed and searching, and finally viewing. GGIE is not a complete end to end architecture or solution, it provides foundational elements that can serve as building blocks for new Internet video innovation.

This is a companion effort to the GGIE W3C Taskforce in the W3C Web and TV Interest Group.

This document is being discussed on the ggie@ietf.org mailing list.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on April 27, 2017.

Table of Contents

1. Introduction

In terms of shear bandwidth, the Internet's largest use, without any close second competitor, is video. This is thanks to the proliferation of Internet connected devices capable of capturing and/or watching streamed video. As of 2015 there are reports that YouTube users upload over 500 hours of video every minute, and that during evening hours NetFlix accounts for a staggering 50+% of Internet traffic. The number of users using the Internet for both ends of the video create-view lifecycle grows daily worldwide, and this is creating an enormous strain on the underlying Internet infrastructure at nearly every point from the core to the edge.

While video is one of the most conceptually simple uses of the Internet, it is perhaps one of the most complex technically, built from standards created by a large number of organizations and groups some dating from before the modern Internet even existed. Many critical parts of this complex ecosystem were not created with either video's particular characteristics or vast scale of popularity in mind. This has lead to both the degradation of the viewer experience and many Internet policy issues around access to bandwidth for video and the needed infrastructure to support the continued explosion in video transport on the Internet.

The pace of video growth has been faster than new bandwidth for the past many years, and all indicators are that, instead of abating, it is actually accelerating as new users, new ways of sharing video, and new types of video continue to be added. The Cisco Visual Networking Index an excellent source of detail on this subject.

The combined current high levels of bandwidth consumed by video, plus the accelerating pace of video's growth mean that to meet users' demand for video, we must do more than simply rely on adding more bandwidth. While other traditional improvements such as more efficient codecs with better compression ratios are expected to contribute to keep video flowing on the Internet, many in the Internet video technology world have explored options to see if any new approaches could be added to the mix to help the problem. That was the motivation behind the creation of the GGIE Taskforce within the W3C in 2014 with the charter to examine the end to end video ecosystem and identify new areas of opportunity to improve video's use of the Internet.

The W3C GGIE taskforce explored ways that video uses the Internet and developed a series of use cases detailing specific scenarios ranging from video capture, the editing and production cycle, through to delivery to viewers. Out of these use cases there emerged a recognition that there might be a new opportunity to improve Internet

video by enabling edge devices, and the underlying network to more actively participate in making delivery optimization choices beyond the simple ways the do currently.

The GGIE approach is to apply and evolve existing technologies to the task of optimizing Internet video transport to permit applications, video devices, and the network to more actively participate in making smart access and transport decisions. This approach recognizes that there are already extensively-deployed video infrastructure elements that need to continue to work and be part of the optimized video ecosystem. These deployed devices, applications, players, and tools are responsible for the already high levels of video bandwidth consumption, and to only address new devices would not be solving the larger, most important problem. This is why GGIE is an evolution of how video uses the Internet, and not a revolution involving wholesale replacement of existing architecture or protocols.

GGIE is not a complete solution to the video problem. It provides foundational building blocks that are intended to be used by innovators in their work to create new optimizations, and novel techniques to help address the video problem in the long term.

GGIE initially proposes a simple framework of three components that will permit improved playback device identification of viewing sources and enable network level awareness of video transport and new cache selection chocies. GGIE proposes: Using existing content identifiers as a means to identify a work, or title; Data level identifiers to identify the encoded video data for a particular manifestation of the title; A mapping service that permits bi-directional resolution of these identifiers.

This document outlines the basic proposal for these three base GGIE components and introduces the overall GGIE approach to evolving the current video ecosystem by introducing basic standardized building blocks for innovators to build upon the Glass to Glass Internet Ecosystem.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. Motivation: Video is filling up the pipes

The growth in video bandwidth need is exceeding the growth in the bandwidth provisioning. This trend is in fact accelerating, meaning the growth rate of video is growing faster than the growth rate of

provisioning. Traditional techniques of caching, higher efficiency codecs, etc, are all being used to help address the probiem and have helped the Internet to continue to support the growth of video thus far.

Video has been the top use of Internet bandwidth for several years and is larger than the bandwidth used by all other applications combined. This trend is unlikely to ease or reverse itself as users of the Internet continue to make Internet transported video one of their top uses of the Internet, either for uploading and sharing video they creator, or as a primary sources for viewing video to a wide variety of viewing devices: computers, tablets, phones, connected televisions, game consoles, and AV receivers.

Adding to user demand, video itself is continually experiencing innovation introducing ever higher resolutions (SD, HD, 4K, 8K...), higher video quality, new distribution services (live one to many streaming), and new user uses. The Cisco Visual Networking Index projects that by 2019 there will be nearly a million minutes of video per second transported by the Internet, a making up 80-90 percent of all IP traffic.

The movitation behind GGIE is to help find new methods that can be brought to bear, in addition to all the exiting ones, to help manage the explosion in Internet video.

4. Video is different

Video is different than other uses of the network due to its combimed high bandwidth demands and high sensitivity to latency and dropped packets. Streaming of basic high-definition 1080p requires bandwidth in the low Mbps translating into Gigabytes for each hour of video, all transported with consistent low latency and very little packet loss in order to deliver a suitable watching experience the viewer. This differentiates video from other Internet applications as some have low latency and packet loss requirements but don’t need high bandwidth, while others may demand high bandwidth, they will tolerate high latency and dropped packets. An email user can tolerate an extra moment to retransmit dropped packets, and a web page user can tolerate a slow DNS lookup, but a video viewer sees latency and dropped packets as jittery playback and low bandwidth as a fundamental barrier to streaming at all. From the user’s perspective the network has faield to meet their need. (Audio has similar challenges in terms of intolerance of delay and jitter, but the data sizes are significantly smaller).

Video data sizes continue to grow at roughly 4x per format iteration as cameras and playback devices are able to capture and display

higher quality images. Early digital video was often captured at either 320x240 pixel resolution or 640x480 standard definition resolution. High definition or HD video at 1920x1080 became possible on some parts of the Internet after 2011, although even in 2016 it remains unavailable or unreliable through many connections such as DSL and many mobile networks. Camera and player technologies are currently expanding again to permit 4K or 3840x2160 pixel resolution reflecting a 4x data increase over HD.

Streaming is very demanding, requiring consistent frame to frame playback in consistent constant time. Advanced features such as pause, fast forward, rewind, slow motion, and fine scrubbing are considered by users as standard features in players that the network must support and serve to further the challenge facing the Internet.

New video abilities such as live streaming by users (both one to one and one to many) bring what has traditionally been done by professional broadcasters with dedicated broadcast infrastructure into the realm of every day users with connected smartphones using the Internet as a real-time global broadcast infrastructure.

5. Historical Approaches to supporting Video on the Internet

5.1. Video as an application

Internet video engineering began by adapting preexisting standards used for over the air broadcast (OTA) and physical media. Video encodings, such as AVI and MPEG2, originally designed for playback from local storage connected to the player where added to the data types carried by existing protocols like HTTP, and new protocols such as RTSP and HLS. Early use of the Internet for video was a copy-and-play model replacing the use of OTA broadcast and physical media to copy video between systems.

As Internet bandwidth became sufficient to allow delivery of video data at the same rate it was being decoded, it became possible to stream video originally at very low resolutions such as 160x120 pixels (19.2 kilopixels), eventually permitting standard definition (SD) 640x480 pixels (0.3 megapixels), and later high definition of 1920x1080 pixels (2 megapixels). This trend continues with some providers beginning to offer 4K or 3840x2160 pixels (8.3 megapixels) requiring very reliable and generous Internet bandwidth end to end connection between the viewer and source.

Unlike the Web, email, and network file sharing which have been engineered and standardized in Internet focused organizations such as the W3C and IETF, video is dependent on standards developed by a very large number of groups, companies, and organizations which include

the IETF, W3C but also MPEG, SMPTE, CTA, IEEE, ANSI, ISO, networking and technology companies, many others. In contrast to the extensive end to end expert knowledge and engineering done to create the Web and email, Internet video has largely been an evolved cobbling and adaption exercise done by engineers with their focus on a few, or one, particular aspect or problem at a time, and little interaction between other parts of the Internet video ecosystem. While it is very much possible to deliver video over the Internet, this uncoordinated cobbling has resulted in many areas of inefficiency where engineering done from an end to end perspective could provide the opportunity to vastly improve how video uses the Internet, which offers the hope of improving the quality of video and increasing the amount of video which can be delivered.

5.2. Video as a network problem

Network, video, and application engineers have constructed elaborate solutions for dealing with bandwidth and processing limitations, network congestion, lossy transport protocols, and the ever growing size of video data. These solutions commonly fall into one of several solution types:

1. Reducing data sizes through resolution changes, compression, and more efficient encodings

2. Downloading before playing instead of real-time streaming

3. Positioning the data close to the viewer via caches, typically on the network edge

4. Fetching of video data at a rate faster than playback

5. Transport protocols that attempt to deliver video data such that the data arrives as if it were done on a congestion free/lossless network

6. Dynamic reselection of sources and transport routes on either a real-time or frequent intervals, 10-15 seconds, using player feedback mechanisms or network telemetry

5.3. Video Ecosystem Encapsulation

The current delivery ecosystem for video has been primarily developed at the higher application layers of the stack. While there has been some video work done at lower levels such as general-purpose transport improvements, caching protocols in CDNi, various multicasting approaches, and other efforts, the majority of video-specific work has previously been done by groups such as ISO’s Moving

Pictures Expert Group (MPEG) which have focused on codecs and codec transport optimized for use on the Internet. These efforts have made video possible on the Internet, but they have done so largely while treating the underlying network as a basic transporter of data. This has resulted in little information being exposed to the network, information that could be used to optimize delivery of the video, and in an architecture that pushes more and more of the intelligence into an ever more complex and isolated core.

The current video model benefits from a significant amount of operational, feature, and protocol encapsulation that has come about due to different groups working independently on the components that make it up. Like any system in which distinct pieces are well encapsulated from one another, this means it is possible to engage in improvements at the networking layer without the need to coordinate with higher levels of the video architecture.

6. Problem Statement and Solution Criteria

At its most basic the problem to be solved for video delivery is how to simultaneous maximize all of the following conditions: The number of viewing devices simultaneously supported by the network; The quality of video as measured by bit-rate and resolution; The number of distinct unique streams that can be delivered.

Solution Constraints

1. Bandwidth growth alone is not a solution

2. Codec efficiency improvements alone are not a solution

3. Existing devices, infrastructure, video delivery techniques must as much as possible continue to be supported and benefit from new solutions.

7. The Glass to Glass Internet Ecosystem: GGIE

GGIE is an effort to improve video’s use of the Internet by examining the end to end video ecosystem from the glass lens of the camera through to the glass the screen, and to identify areas of simplifications, standardization, and reengineering to make better use of bandwidth enabling smarter network use by video creators, distributors, and viewers. GGIE is focused on how video uses the Internet, and not on how it is encoded or compressed. Likewise GGIE does not deal with content protection. GGIE’s scope however does include creator and viewer privacy, content identification and recognition as a means to enable smarter network usage, edge caching, and discoverability.

GGIE benefits from the encapsulation of the video ecosystem elements enabling it to introduce evolutional features to elements without disrupting other distinct encapsulated parts.

GGIE is intended to work with a wide variety of video encoding codecs, and video distribution and transport protocols. While examples using MPEG-DASH are used due to is pervasive use, GGIE is not limited to MPEG-DASH or any other video distribution system or codec.

Beyond improving the simple experience of a viewer using the Internet to watch linear video, it is hoped that a set of improved Internet video infrastructure standards will provide a foundation that permits innovators to create the next generation of Internet video content (such as multisource personalized composite experiences, interactive stories, and live personal broadcasting, to name a few).

Due to the very diverse and large deployment of existing video playback devices and infrastructure, it is viewed as essential that any evolved ecosystem continues to work with the majority of the legacy deployment without the need for updates or changes to the existing ecosystem.

7.1. Related work: W3C GGIE Taskforce

A companion effort ran through 2015 in the W3C Web and TV Interest Group’s GGIE Taskforce. The W3C GGIE group developed a series of use-cases on discovery, search, delivery, identity, and metadata which can be found at https://www.w3.org/2011/webtv/wiki/GGIE_TF

8. GGIE work of relevance to the IETF

This section assumes a working familiarity with video creation and consumption "life cycle". For reference, an overview has been provided in the Appendix.

8.1. Affected IETF work areas

It is expected that significant improvement is possible in the video transport ecosystem by modest evolution and adaption of existing standards for addressing, transporting, and routing of video data flows between sources and display.

8.2. Example use cases

The following example use case help illustrate the use of the GGIE core elements

8.2.1. Alternate Source Discovery

Description: A video player is streaming a movie from a CDN cache in the core of the network. This use case illustrates the use of a media identifier to query a media address resolution service to locate additional alternate sources that offer the same movie.

1. The video player user selects a movie to watch from a list using the player application UI.

2. The video player application has the media identifier of the movie in the metadata description of the movie. This identifier is passed to the playback device when the movie selected.

3. The playback device send a search query to the Media Address Resolution Service (MARS) which includes the media identifier, and additional query parameters use to filter the results returned.

4. The MARS server searches its database and returns all the Media Encoding Networks matching the media identifier and filters the results using the additional parameters submitted in the query. Each Media Encoding Network represents a different encoding of the video.

5. The player then examines the returned list of media encoding networks and selects, from its perspecitve, the optimal source for the title.

6. The player then directs its streaming requests to the selected Media Encoding Network addresses to obtain the video data for the movie.

7. The video data is decoded and displayed on the screen

8.2.2. Alternate Format Discovery

Description: A video player is streaming a movie, and wants to send the audio to another device for playback. However, the current video data being streamed does not contain any audio that matches the codecs the audio device can play. The audio device uses the core GGIE services to locate an alternate encoding of the movie that contains audio it can decode.

1. The user directs the video player to send the audio portion of the playing video to an external audio device.

2. The video player application passes the media idenfitier for the video to the audio device as well as the media encoding network address the video player is using.

3. The audio device begins streaming from the media encoding network is was given, but discovers the data does not include audio that is able to decode.

4. The audio device sends a search query to the Media Address Resolution Service (MARS) which includes the media identifier, and additional query parameters including the list of audio codecs and language choice it is able to decode.

5. The MARS server searches its database and returns all the Media Encoding Networks matching the media identifier and filters the results to only those matching the language and audio codec supplied in the search.

6. The audio player examines the returned list of media encoding networks, selects a media encoding network and begins streaming data from it.

7. The external audio player decodes the returned movie data and plays it for the user.

8.3. Core GGIE elements

GGIE proposes three initial fundamental pieces:

1. Media Identifiers which identify the video at the title, or work level;

2. Media Encoded Networks which are subnets used to reference the encoded video data;

3. Media Address Resolution Service which maps Media Identifiers for a title to the Media Encoded Networks containing the encoded video versions of the title.

These three foundational elements help by exposing information that can be used in selection in a way that is independent of the video encoding and video data storage choice. It also enables more sophisticated video use cases beyond the basic single device playing a video stream from an origin server over a flow controlled protocol.

8.3.1. Media Identifiers

A Media Identifier is a URI that carries a content identifier system declaration, and a content identifier from the system that refers unambiguously to a work, or title. This can be any content identification system, GGIE does not specify the system used.

For example, a media identifier for a title identified by an EIDR value would include a declaration that the identifier is from EIDR, and would additionally contain the EIDR value.

At the application level, such as UI program guide applications, search engines, and metadata databases, it is the identification of the work or identity of the video that is typically of interest and not the encoding, bit-rate, or the location of CDN caches etc. For example, a UI would indicate that "the Minions movie" as opposed to "a 15 megabit per second, HEVC encode with high dynamic range and Dolby encoded 7.1 English audio of the Minions movie". Those additional technical details are important when choosing a particular encoded manifestation of the movie for delivery, decode, and playback, but they are not generally needed as information to be presented to the user or used to make viewing choices. Such technical information is used after the user has chosen the title to watch, but is used by the playback device not the user in selecting the video. Media Identifiers in GGIE contain only title information, and not encoding information.

There are many media identifiers in use for both personal and professional content, with new ones being introduced seemingly weekly. To try to create a single identifier to either harmonize or replace the others, repeatedly been proven in practice to be an impossible task. Recognizing this, the GGIE instead proposes to standardize a URI which would contain at least two fields: 1) A scheme identifier; 2) An unambiguous title identifier (note: this is unambiguous only within domain of the identified scheme).

For professional content, titles are increasingly identified with a scheme called EIDR that can identify both master versions of works, and edit level versions. Likewise, advertisments use a scheme called AD-ID.

8.3.2. Media Address Resolution Service (MARS)

The media address resolution service (MARS) provides bidirectional mapping of Media Identifiers to Media Encoding Networks. It is queryable using a query protocol which returns any results matching the terms of the query parameters.

A Media Identifier alone isn't sufficient to connect a device to a video data source. The media identifier distinguishes the work, but not the technical details of an instance of the work such as codec, bit-rate, resolution, high dynamic range video, audio encoding, nor does it include information about available streaming sources etc. The Media Address Resolution Service (MARS) provides this association. It can be queried with the Media Identifier, and optional filtering parameters, and will return Media Encoding Network addresses for instances of matching encodings of the work.

This translation is used commonly in video streaming services today. The link provided in the program guide UI will include a unique identifier for the work which is then mapped by the streaming service backend into a URI containing a network identifier and other info which point to a caching server and the media data files in the cache. MARS generalizes this and make it available via query over the network.

8.3.3. Media Encoding Networks (MEN)

Media Encoding Encoding Networks are arrangements of encoded video data that are assigned addresses under a shared prefix and subnet following a scheme appropriate for the encoding used by the video data. Each Media Encoding Network instance represents a distinct instance of a set of associated encodings for a work. Different Media Encoded Network address assignment schemes would be defined under GGIE to handle different encode data such as MPEG-DASH and HLS.

For example, a single MEN instance would hold each of the different variable bit-rate encodes for a single encoding of a video If a new encoding instance of the video was prepared, it would have separate and distinct MEN assigned to it.

8.3.3.1. Example: Using Media Encoding Networks with MPEG-DASH

A very basic form a video delivery uses persistent connection from a player to a video file source which then streams the video by transmitting the video file data, byte by byte in sequence, from the first byte of the file until the last. This trivial approach requires the device to know the server IP address and port number to connect to. Essentially this involved simply transporting the file from the source to the playback device in byte order.

In practice simple file streaming is not used beyond local device to device playing in home networks as it doesn't permit dynamic bit rate selection, source or session fail over, or trick play (pause, skip forward, skip backward) etc. Instead manifest files contain lists of available servers holding MPEG-DASH encodings of the larger video

file divided into fragments containing short portions (e.g. 2-15 seconds) of the video called chunks by MPEG-DASH. (GGIE generalizes the MPEG-DASH chunk term into the more general shards). Each shard is a distinct file typically named to reflects the video encode it belongs to, and it's sequence position.

For example the shards for MY-VIDEO might be names MY-VIDEO-001, MY-VIDEO-002, ... MY-VIDEO-nnn. The player then requests the shards in the order it wants them over a data transport protocol such as http, with the translation of the actual data sent in response to requests for the named shards being handled by the data server.

So under MPEG-DASH the player is sent a manifest file containing the address of the data server and the shard name to request. The player then iterates over the available shards in the order desired by the user. The manifest then contains URI's with the SERVER-ADDRESS and the CHUNK name. This file can be sent once per video play, or more commonly is sent at an interval of ~15 seconds to permit the sending CDN to customize for each player, and to respond quickly to changes in the network delivery performance and availability.

Each shard request by the device involves a network level server IP address and port number, and an application level shard name. The network is thus able to manage the routing of request to the server, and the routing of the response, but it lacks the information needed to do anything else to help optimize the video data transport.

GGIE proposes using Media Encoding Networks an evolution of this that has the benefit of being backward compatible with manifest files, while enabling the transport network and video ecosystem to have more information to the network about the video transport flowing over it.

Using Media Encoding Networks for MPEG-DASH will be described in another Internet-Draft, but the basic proposal is to assign the shards into a sequence of IP addresses organized to reflect the same ordering association that the chunk names followed in the MPEG-DASH scheme. These shard addresses form a Media Encoding Network, and they expose to the network layer knowledge of the specific video data being transported between requesting device and the file server holding the data.

This in practice means that Media Encoding Network addresses refer to the shard and not the server holding the shard. This then permits the network to be involved in the routing of the request for the shard, as opposed to the CDN preparing the manifest file. Among other benefits, this permits the network to provide path failover functionality beyond the CDN manifest.

This enables the network to be involved in shard source selection. Consider the use case wherein the network becomes aware of a local cache that holds the requested shard, and is closer to the device than another cache deeper in the network. The network can direct the request to the local cache and save the transit cost and bandwidth of sending the request and response exchange with the deeper cache. This can reduce network congestion as well as deliver faster transport for the shard to the playback device.

8.3.4. Media Encoding Network Gateways

In this new approach, the server providing the shard data is possibly better viewed as acting as a gateway to the shard addresses versus being just a file server. In practical terms, existing CDN caches can perform this role by mapping the requested shard address to the on disk file containing the shard. However, new CDN caches can be developed work directly with the Media Encoding Network scheme, and can act as smart caches proactively provisioning data within the Media Encoding Network address space.

9. Conclusion and Next Steps

GGIE seeks to help address this problem by establish standards based foundational building blocks that innovators can build upon creating smarter delivery and transport architectures instead of relying on raw bandwidth growth to satisfy video’s growth.

Next steps will include describing the working prototypes of the GGIE core elements and more exended use cases addressed by GGIE many of which were defined in the W3C GGIE Taskforce.

10. Acknowledgements

Contributions to this document came from Bill Rose, Gaurav Naik, John Brzozowski.

11. IANA Considerations

None (yet).

12. Security Considerations

12.1. Privacy Concerns

The assignment of persistent IPv6 Prefixes to MEN permits the video being streamed to be identified at the network level by observing the destrination addreses sent from the player to the media gateway. In situations where it is desired by the user to prevent this level of

observation is necessary to obscure the true MEN prefix of the video being streamed.

12.1.1. Privacy via VPN

One remediation is the use of a VPN that will encapsulate and hide the traffic between the player and the streaming cache, or at least between the trusted network the player resides on and the streaming cache network. This will make identification of the actual video title from the open Internet during transit.

12.1.2. Session Prefix Renumbering

Another technique is to have the player and streaming cache remap the IPv6 prefix for the streaming session to a new prefix. Under such a renumbering the cache will advertise to the routing layer and respond to requests sent from the player to the session prefix just as it would to the original video MEN prefix.

13. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

Appendix A. Overview of the details of the video lifecycle

This section outlines the details of the video lifecycle -- from creation to consumption -- including the key handholds for building applications and services around this complex data. The section also provides more detail about the scope and requirements of video (scale of data, real-time requirements).

Note: this document only deals with streaming video as used by movies, TV shows, news broadcasts, sports events, music concert broadcasts, product videos, personal videos, etc. It does not deal with video conferencing or WebRTC style video transport.

A.1. Media Lifecycle

The complex workflow of creating media and consuming it is decomposable into a series of distinct common phases.

A.1.1. Capture

The capture phase involves the original recording of the elements which will be edited together to make the final work. Captured media elements can be static images, images with audio, audio only, video only, or video with audio. In sophisticated capture scenarios more than one device maybe simultaneously recording.

A.1.1.1. Capture Metadata

The creation of metadata for the element, and for the final video begins at capture. Typical basic capture metadata includes Camera ID, exposure, encoder, capture time, and capture format. Some systems record GPS location data, assigned asset ids, assigned camera name, camera spatial location and orientation.

A.1.2. Store

The storage phase involves the transport and storage of captured elements data. During the capture phase, an element is typically captured into memory in the capture device and is then stored onto persistent storage such as disc, SD or memory card. Storage can involve network transport from the recording device to an external storage system using either storage over IP protocols such as iSCSI, a data transport such as FTP, or encapsulated data transport over a protocol such as HTTP.

Storage systems can range from basic disk block storage, to sophisticated media asset libraries

A.1.2.1. Storage Metadata

Storage systems add to the metadata associated with media elements. For basic block storage, a file name and file size is typical, as are a hierarchical grouping, creation date, and last-access date. For library system an identifier unique to the library is typical, as well as grouping by one or more attributes, a time stamp recording the addition to the library and a last access time.

A.1.3. Edit

Editing is the phase where one or more elements are combined and modified to create the final video work. In the case of live streaming, the edit phase maybe bypassed.

A.1.4. Package

Packaging is the phase in which the work is encoded in one or more video and audio codecs. These maybe produce multiple data files, or they may be combined into a single file container. Typically, creation or registration of a unique work identifier, for example an Entertainment Identifier from EIDR, is assigned in the packaging phase.

A.1.4.1. Package Metadata

A.1.5. Distribute

The distribute phase is publishing or sharing the packaged work to viewers. Often it involved uploading to a site such as YouTube, or Facebook for social media, or sending the packaged media to streaming sites such as Hulu.

It is common for the distribution site to repackage the video often transcoding it to codecs and bitrates chosen by the distributor as more efficient for their needs. Distribution of content expected to be widely viewed often includes prepositioning of the content on a CDN (Content Distribution Network).

Distribution involves delivery of the video data to the viewer.

A.1.5.1. Distribution Metadata

Distribution often adds or changes considerable amounts of metadata. The distributor typically assigns a Content Identifier to the work, that is unique to the distributor and their content management system (CMS). Additional actions by the distributor such as repacking and transcoding to new codecs or bitrates can require significant changes to the media metadata.

A secondary use of distribution metadata is enabling easy discovery of the content either through a library catalog, EPG (electronic program guide), or search engine. This phase often includes significant new metadata generation involving tagging the work by genre (sci-fi, drama, comedy), sub-genre (space opera, horror, fantasy), actors, director, release date, similar works, rating level (PG, PG-13), language level, etc.

A.1.6. Discovery

The discovery phase is the precursor to viewing the work. It is where the viewer locates the work either through a library catalog, a

playlist, an EPG, or a search. The discover phase connects interested viewers with distribution sources.

A.1.6.1. Discovery Metadata

It is typical for discovery systems to parse media metadata to use the information as part of the discovery process. Discovery systems may parse the content to extract imagery and audio as additional new metadata for the work to ease the viewers navigation of the discovery process perhaps as UI elements. The system may import new externally generated metadata about the work and associate it in its search system, such as viewer reviews, metadata cross reference indices.

A.1.7. Viewing

The viewing phase encompasses the consumption of the work from the distributor. For Internet delivered video it is typical for delivery to involve a CDN to perform the actual delivery.

A.2. Video is not like other Internet data

Video is distinctly different from other Internet data. There are many characteristics that contribute to video's unique Internet needs. The most significant characteristics are:

1. large size of video data (Gigabytes per hour of video)

2. high bandwith demands (Mbps to Gbps)

3. low latency demands of streamed video

4. responsiveness to trick play requests by the user (stop, fast forward, fast reverse, jump ahead, jump back)

5. multiplicity of formats and encodings/bit rates that are acceptable substitutes for one another

A.2.1. Data Sizes

Simply put compared to all other common Internet data sizes, video is huge. A still image often ranges from 100KB to 10MB. A video file can commonly range from 100MB to 50GB. Encoding and compression options permit streaming videos using bandwidth ranging from 700Kbps for extremely compressed SD video, to 1.5-3.0 Mbps for SD video, to 2.5-6.0 Mbps for HD video, and 11-30Mbps for 4K video.

Still images have 4 dimensional properties that affect their data size:

1. number of horizontal X pixels

2. number of vertical Y pixels

3. bytes per pixel

4. compression factor for the image encoding.

Video adds to this:

1. frames per second playback rate

2. visual continuity between frames (meaning users notice when frames are skipped or played out of order)

3. discontiguous jumps between frames such as skipping forward or backwards to inserting frames from other sources between contiguous frames (advertisement placement)

Each video format roughly increases by x4 the data needs of the previously resolution: (1) SD is 640x480 pixels; (2) HD is 1920x1080 pixels; (3) 4K is 3840x2160 pixels.

Video, like still images, assigns a number of pixels to store color and luminance information. This currently evolving alongside resolutions after being stagnant for many years. The introduction of high dynamic range videos or HDR has changed the color gamut for video and increased the number of bits needed to carry luminance from 8 to 10 and in some formats more.

Compression is often misunderstood by viewers. Compression does not change the video resolution, SD is still 640x480 pixels, HD is still 1980x1080 pixels. What changes is the quality of the detail in each frame, and between frames.

Video is in its simplest form a series of still images shown sequentially over time, adding an additional attribute to manage.

A.2.2. Low Latency Transport

Viewers demand that video plays back without any stutter, skips, or pauses, which translates into low latency, high reliability transport of the video data.

A.2.3. Multiplicity of Acceptable Formats

One of the unique aspects of video viewing is that there can exist multiple different encodings/versions of the same video, many of which are acceptable substitutes for one another. This is a unique aspect of video viewing and differentiates video delivery from other data transports.

Other application data types don’t have or leverage the concept of semantic equivalences to the same extent as video. Even email, which supports multiple encodings in a multipart MIME message, has a finite number of representations of "the message", shipped as one unit, whereas video often has many distinct different encodings each as separate file or container of files managed as a distinct entity from the others.

A.3. Video Transport

A.3.1. File vs Stream

There are two common ways of transporting video on the Internet: 1) File based; 2) Streaming. File based transport can use any file transport protocol with FTP and BitTorrent being two popular choices.

File based playback involves copying a file and then playing it. There are schemes which permit playing portions of the file while it progressively is copied, but these schemes involve moving the file from A->B then playing on B. FTP and BitTorrent are examples of file copy protocols.

Streaming playback is most similar to a traditional Cable or OTA viewing of a video. The video is delivered from the streaming service to the playback device in real time enabling the playback device to receive, decode, and display the video data in real time. Communication between the player and the source enable pausing, fast forward and rewind by managing the data blocks which are sent to the player device.

Authors’ Addresses

Glenn Deen
NBCUniversal

Email: rgd.ietf@gmail.com

Leslie Daigle
Thinking Cat Enterprises LLC

Email: ldaigle@thinkingcat.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: January 8, 2017

G. Deen
Comcast-NBCUniversal
G. Naik
Drexel University
J. Brzozowski
Comcast
L. Daigle
Thinking Cat Enterprises LLC
W. Rose
WJR Consulting
M. Townsley
Cisco

July 7, 2016

Using Media Encoding Networks to address MPEG-DASH video
draft-deen-naik-ggie-men-mpeg-dash-00

Abstract

This document describes an approach to using a Media Encoding Network of IPv6 Prefixes and Addresses as identifiers for MPEG-DASH encoded video. This is part of the GGIE Glass to Glass Internet Ecosystem effort for Internet Video.

This document is being discussed on the ggie@ietf.org mailing list.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on January 8, 2017.

Table of Contents

1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

2. Introduction

GGIE, the Glass to Glass Internet Ecosystem, described in [I-D.deen-daigle-ggie], is an effort to improve video's use of the Internet though evolving and applying modern Internet networking technology to Interet video.

This document is a proposed Media Encoding Network organizational definition for MPEG-DASH enoded video. In the following sections, we describe a Media Encoding Network structure for MPEG-DASH content using IPv6 addresses as the address for MPEG-DASH video chunks, and organizing these addresses into a IPv6 subnet under a prefix.

A MPEG-DASH encoded video organizaed following this Media Encoding Network scheme is in turn referrable to using the assigned prefix, with each distinct encoding of the video being assigned a distinct prefix. Hence two copies of the same video encode would share the same prefix, while a different encode would have a different prefix.

Other Media Encoding Networks organizational definitions are possible for MPEG-DASH video. The simple organizational structure defined in this document is designed to work, in a backwards compatible manner, with existing MPEG-DASH video players.

2.1. Media Encoding Networks

One of the concepts being discussed in GGIE is that of a Media Encoding Network. As introduced in the GGIE Introduction [I-D.deen-daigle-ggie] document, a Media Encoding Network consists of the data elements of a audio-video encoding of a work organized following a distinct logical structure appropriate for efficiently transporting and accessing the data elements for the video asset. Network level identifiers are assigned to each of these elements under a shared prefix and following an address assigment plan appropropriate for the type of encoding used for the AV data.

Media Encoding Networks is a generalized abstraction intented to be used with many different enoding and transport schemes.

GGIE recognizes that there is currently a great diversity of encoding and transports such as MPEG-DASH [DASH] and HTTP Live Streaming (HLS) [I-D.pantos-http-live-streaming] to name but two, with more continuing to be developed and introduced. Recognizing this diversity and innovative environment, GGIE proposes the Media Encoding Network as a resuable abstraction that can be trailored and defined with different logical organizations to support different environments, applications, and media encodings.

A Media Encoding Network is a logical entity that can be assigned a network level identifier enabling it to be referred to at a network device level and permitting devices and the network to worked cooperatively to optimize data transport and access choices.

3. MPEG-DASH Internet Video Concepts

A common technique used in the delivery of a media or video on the Internet via streaming services and CDNs is to break up an encoding of a video into chunks or media segments containing a fixed duration of video. MPEG-DASH [DASH] is an example of such an approach. The segments typically represent small portions of the video with 6-10 seconds of video playback being common. In most implementations, the segments of videos are identified by file names and served to clients using conventional web servers using HTTP GET requests.

Systems such as MPEG-DASH enable client players to switch between encodings of different quality levels of the video with higher quality encodings requiring large amounts of data, and conversely lower quality encodings requiring smaller amounts of data. The system coordinates each encoding to produce points of alignment called intra-coded frames or iFrames where a player can switch between different encodings without missing frames of the video playback. Thus, a player can adapt to changing network conditions without re-buffering or freezing of the playback.

When the encodings are broken into segments, the segments are organized such that the playback system can switch to a different encoding level from the version it has been playing by requesting the next segment of data holding the iFrame matching the next iFrame of the current encoding. In practice each segment of an encoding is an individual file stored on video or CDN server and playback consists of the player repeatedly requesting the next file in sequence from the server, with the file names following a consistent incremental naming scheme indicating an encoding identifier and a segment sequence identifier.

Typically, a video file is processed by an encoder to produce two or more different quality encodings with each encoded version being passed through a process to break into segment files with aligned iFrames and each file named with a name identifying the encoding and sequence number. This process requires coordination to create iFrame alignments and a consistent naming convention to allow players to transition between encodings and to iteratively access the next correct segment.

3.1. Internet Video playback as a network

Transitioning between segments is an example of a simple directed graph (or digraph). Each segment is a vertex or node and the naming convention defines an ordered directed traversal of the graph, and the iFrame aligned segments forming the edges of the graph. It is also possible to recognize that the directed graph behavior of a

player switching between segments can more generally be viewed as a network such as it is used on the Internet.

The network of segments can be identified using the IP addressing scheme from the Internet, in particular IPv6 is well suited for this due to the large number of addresses available in it's 128-bit address space. IPv4 could also be used, but with only 32 bits of address space the available addresses would be quickly exhausted in practical use.

This is really a simple evolution of the way MPEG-DASH chunks are organized today as files with names such as MOVIE-SEGMENT-00, MOVIE-SEGMENT-01,... and so on. In practical terms, this scheme simply replaces the ASCII filename, with a 128-bit number represented as HEX digits. In this way, this scheme remains compatible with existing CDN serving of MPEG-DASH video.

4. MPEG-DASH Video Chunk Addressing

Staying consistent with Media Encoding Networks being a generic abstraction, the more generic term Shard is used in place of the MPEG-DASH specific Chunk for individual units of encoded video data.

IPv6 addresses [RFC4291] are specified in and are broken into two parts that split the available 128 bits of address space as follows:

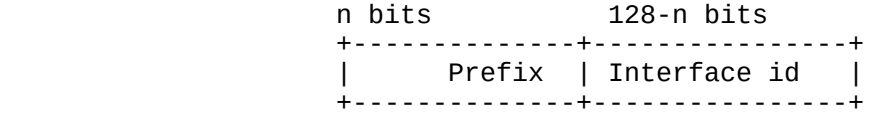

Figure 1: IPv6 Address

One addressing approach to naming segments can be as follows:

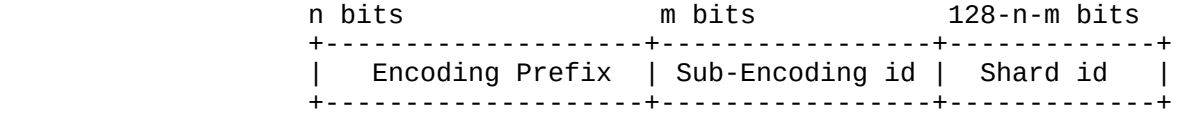

Figure 2: Proposed

Which consists of an Encoding Prefix that is uniquely assigned to a set of aligned MPEG-DASH encodings of the video, a sub-encoding id which identifies a particular encoding, and the id of the individual shard of encoded video data.

The encoding prefix permits a set of encodings to be associated with one another. Grouping a set of encodings of a video under a shared Encoding Prefix permits referencing all the segments of a group of encodings as a single entity under the Encoding Prefix.

The sub-encoding id groups the shards of a single sub-encoding together under an identifier to permit managing the collection of segments as a single entity.

Shards that share MPEG iFrame aligment share the same Shard id. This then defines a network layout with shards for each different bit-rate organized sequentially and contiguously under a shared sub-encoding subnet and shards with aligned iFrames being organized with the same shard id across sub-encoding subnets.

5. Video Playback

This approach permits the Prefix to identify a particular group of encodings of a video. Each encoding has an assigned series of addresses consisting of the prefix, followed by the series of address bits that uniquely identify the shard. All the playback pathways are preserved in this addressing scheme of the edges of the graph.

The above approach works well for a video that is encoded by one party that can coordinate the encoding process, to produce aligned iFrames, and assign the common encoding prefix and segment assignments for the network.

A playback device can be provided the Prefix for the network, and can iterate through the segments to play the video. It can jump between sub-encode subnets to select different quality or vary the bit rate of the playback.

6. Implementation

For the evaluation of this scheme, a prototype video streaming service implementing this approach was developed. In particular, it provides an Electronic Program Guide (EPG) and uses an open-source HTML5 video player with MPEG-DASH. Instead of providing the player with HTTP URIs for each segment of video, our this prototype uses global IPv6 addresses. This change is transparent to the host operating system, the HTML5 video player, and the network. The service backend is implemented in Python and utilizes other open source components. A demonstration at IETF96 is planned to be shown during Bits-n-Bytes.

7. Conclusion and Next Steps

This draft proposes a Media Encoding Network addressing scheme for MPEG-DASH Internet video using IPv6 addresses. It is an example that can built upon to define other more complex Media Encoding Network schemes for MPEG-DASH and other encoding/transports.

8. Acknowledgements

9. IANA Considerations

None (yet).

10. Security Considerations

None (yet).

11. References

11.1. Normative References

[I-D.deen-daigle-ggie]
Deen, G. and L. Daigle, "Glass to Glass Internet Ecosysten Introduction", draft-deen-daigle-ggie-01 (work in progress), June 2016.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, DOI 10.17487/RFC4291, February 2006, <http://www.rfc-editor.org/info/rfc4291>.

11.2. Informative References

[DASH] ISO, "Dynamic adaptive streaming over HTTP (DASH) -- Part 1: Media presentation description and segment formats", <http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?csnumber=65274>.

[I-D.pantos-http-live-streaming]
Pantos, R. and W. May, "HTTP Live Streaming", draft-pantos-http-live-streaming-19 (work in progress), April 2016.

Authors' Addresses

Glenn Deen
Comcast-NBCUniversal

Email: rgd.ietf@gmail.com

Gaurav Naik
Drexel University

Email: gn@drexel.edu

John Jason Brzozowski
Comcast

Email: John_Brzozowski@Cable.Comcast.com

Leslie Daigle
Thinking Cat Enterprises LLC

Email: ldaigle@thinkingcat.com

Bill Rose
WJR Consulting

Email: brose@wjrconsulting.com

Mark Townsley
Cisco
Paris

Email: townsley@cisco.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: July 24, 2017

C. Holmberg
J. Axell
Ericsson
January 20, 2017

IANA Registration of New Session Initiation Protocol (SIP) Resource-Priority Namespace for Mission Critical Push To Talk service
draft-holmberg-dispatch-mcptt-rp-namespace-05

Abstract

This document creates an additional Session Initiation Protocol (SIP) Resource-Priority namespace to meet the requirements of the 3GPP defined Mission Critical Push To Talk, and places this namespace in the IANA registry.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on July 24, 2017.

Table of Contents

1. Introduction

The Third Generation Partnership Project (3GPP) has defined a Mission Critical Push To Talk (MCPTT) over LTE service [TS.3GPP.22.179] . The MCPTT service supports an enhanced PTT service, suitable for mission critical scenarios, based upon 3GPP Evolved Packet System (EPS) services. The requirements for the MCPTT service defined within 3GPP can also form the basis for a non-mission critical Push To Talk (PTT) service.

The MCPTT service is intended to support communication between several users (a group call), where each user can gain permission to talk in an arbitrated manner. However, the MCPTT service also supports private calls between pairs of users.

MCPTT is primarily targeted to provide a professional Push To Talk service to e.g., public safety, transport companies, utilities or industrial and nuclear plants. In addition to this, a commercial PTT service for non-professional use (e.g., groups of people on holiday) may be delivered through an MCPTT system. Based on their operational model, the performance and MCPTT features in use vary per user organization, where functionality which is more mission critical specific (e.g., Imminent Peril Call) might not be available to commercial customers.

The MCPTT service provides its users with different priorities for the access to network resources in order to provide means to prioritize between calls when resources are scarce. These priorities take into account among other things the priority and role of the caller, the priority and type of the group, and the situation in which the call is made.

The SIP level call control procedures using these namespaces are specified in [TS.3GPP.24.379]. The namespaces defined here will

support a wide range of queuing options. The namespaces correspond to what can be supported over the 3GPP Rx interface, defined in [TS.3GPP.29.214]. The usage of the namespaces can be tailored to the needs of the operator. The mechanism to do this is to configure which values a specific user is allowed to use. This configuration is specified in [TS.3GPP.24.384].

High priority calls when there is danger of life, for either the public safety worker or any other human, need to be set up immediately and thus require preemption. Other calls may be less sensitive in call set-up time but have a high priority once established. For these calls a queueing mechanism is more appropriate. The MCPTT data transfer service currently under development can benefit from a queueing mechanism. Another example is video only calls that are not critical in call set-up time, but where keeping the call is important.

This document creates additional Session Initiation Protocol (SIP) Resource-Priority namespaces to meet the requirements of the 3GPP defined Mission Critical Push To Talk, and places these namespaces in the IANA registry.

2. Applicability

This document defines namespaces applicable for MCPTT services defined by 3GPP that use the network services of a 3GPP defined LTE network. The use of this namespace outside such networks is undefined.

3. New SIP Resource-Priority Namespaces Created

3.1. Introduction

This document introduces the MCPTT namespaces mcpttp and mcpttq, the name coming from the 3GPP defined Mission Critical Push To Talk service.

3.2. The MCPTT namespaces

The mcpttp namespace uses the priority levels listed below from lowest to highest priority.

mcpttp.0 (lowest priority)
mcpttp.1
mcpttp.2
mcpttp.3
mcpttp.4
mcpttp.5

mcpttp.6
mcpttp.7
mcpttp.8
mcpttp.9
mcpttp.10
mcpttp.11
mcpttp.12
mcpttp.13
mcpttp.14
mcpttp.15 (highest priority)

Intended algorithm for mcpttp is preemption.

New Warning code: No.

New SIP response code: No.

The mcpttq namespace uses the priority levels listed below from lowest to highest priority.

mcpttq.0 (lowest priority)
mcpttq.1
mcpttq.2
mcpttq.3
mcpttq.4
mcpttq.5
mcpttq.6
mcpttq.7
mcpttq.8
mcpttq.9
mcpttq.10
mcpttq.11
mcpttq.12
mcpttq.13
mcpttq.14
mcpttq.15 (highest priority)

Intended algorithm for mcpttq is queuing.

New Warning code: No.

New SIP response code: No.

4. Security Considerations

This document does not have any impact on the security of the SIP MCPTT protocol. Its purpose is purely administrative in nature.

5. IANA Considerations

Abiding by the rules established within [RFC4412] and [RFC7134] , this is an Informative RFC creating two new namespaces, their associated priority-values, and intended algorithms.

6. Acknowledgments

The authors would like to thank Bob Fredericks, Baruh Hason, Mary Barnes and Keith Drage for comments and discussions.

7. Change Log

[RFC EDITOR NOTE: Please remove this section when publishing]

Changes from draft-holmberg-dispatch-mcptt-rp-namespace-04.

- - Editorial changes based on gen-art review. Renderin of authors name and address fixed.

Changes from draft-holmberg-dispatch-mcptt-rp-namespace-03.

- - Editorial changes based on sec- and opt- directorate reviews.

Changes from draft-holmberg-dispatch-mcptt-rp-namespace-01.

- - Removal of Conventions section.
- - Editorial changes.

Changes from draft-holmberg-dispatch-mcptt-rp-namespace-00.

- - The two namespaces have been spelt out explicitly.
- - The numbering of priority levels is changed from 1-16 to 0-15.
- - Address of one author has changed.

8. Normative References

[RFC4412] Schulzrinne, H. and J. Polk, "Communications Resource Priority for the Session Initiation Protocol (SIP)", RFC 4412, DOI 10.17487/RFC4412, February 2006, <http://www.rfc-editor.org/info/rfc4412>.

[RFC7134] Rosen, B., "The Management Policy of the Resource Priority Header (RPH) Registry Changed to "IETF Review"", RFC 7134, DOI 10.17487/RFC7134, March 2014, <http://www.rfc-editor.org/info/rfc7134>.

[TS.3GPP.22.179]
3GPP, "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Mission Critical Push To Talk (MCPTT) over LTE; Stage 1", 3GPP TS 22.179 13.3.0, December 2015.

[TS.3GPP.29.214]
3GPP, "3rd Generation Partnership Project; Technical Specification Group Core Network and Terminals; Policy and Charging Control over Rx reference point;", 3GPP TS 29.314 13.7.0, September 2016.

[TS.3GPP.24.379]
3GPP, "3rd Generation Partnership Project; Technical Specification Group Core Network and Terminals; Mission Critical Push To Talk (MCPTT) call control; Protocol specification;", 3GPP TS 24.379 13.2.0, September 2016.

[TS.3GPP.24.384]
3GPP, "3rd Generation Partnership Project; Technical Specification Group Core Network and Terminals; Mission Critical Push To Talk (MCPTT) configuration management; Protocol specification", 3GPP TS 24.384 13.2.0, September 2016.

Authors' Addresses

Christer Holmberg
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

Email: christer.holmberg@ericsson.com

Joergen Axell
Ericsson
Groenlandsgatan 31
Stockholm 16480
Sweden

Email: jorgen.axell@ericsson.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: June 4, 2017

J. Levine
Taughannock Networks
T. Herkula
optivo GmbH

December 1, 2016

Signalling one-click functionality for list email headers
draft-levine-herkula-oneclick-10

Abstract

This document describes a method for signaling a one-click function for the List-Unsubscribe email header field. The need for this arises out of the actuality that mail software sometimes fetches URLs in mail header fields, and thereby accidentally triggers unsubscriptions in the case of the List-Unsubscribe header field.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on June 4, 2017.

Table of Contents

1. Introduction and Motivation

An [RFC2369] List-Unsubscribe email header field can contain HTTPS [RFC7230] URIs. In that header field the HTTPS URI is intended to unsubscribe the recipient of the message from the list. But anti-spam software often fetches all resources in mail header fields automatically, without any action by the user, and there is no mechanical way for a sender to tell whether a request was made automatically by anti-spam software or manually requested by a user. To prevent accidental unsubscriptions, senders return landing pages with a confirmation step to finish the unsubscribe request. A live user would recognize and act on the confirmation step, but an automated system would not. That makes the unsubscription process more complex than a single click.

Operators of broadcast marketing lists tend to be primarily concerned about deliverability of their mail: whether the mail is delivered to the recipients and how the messages are presented, e.g., whether in

the primary inbox or in a junk folder. Many mail systems allow recipients to report mail as spam or junk, and mail streams from senders whose mail is often reported as junk tend to have poor deliverability. Hence the mailers want to make it as easy as possible for recipients to unsubscribe; if an unsubscription process is too difficult, the recipient's alternative is to report mail from the sender as junk until the mail no longer appears in the recipient's inbox.

Operators of recipient mail systems are aware that their users do not make a clear distinction between unsubscription and junk. In some cases they allow trustworthy mailers to request notification when their mail is reported as junk, so they can unsubscribe the recipient, but the process of identifying trustworthy mailers and notifying them does not scale well to large numbers of small mailers. This specification provides a way for recipient systems to notify the mailer automatically, using only information within the mail message, and without prearrangement. Some recipient systems might wish to send an unsubscription notice to mailers whenever a user reports a message as junk, or they might offer the user the option to report and unsubscribe.

If a mail recipient is unsubscribing manually and the unsubscription process requires confirmation, the resulting web page is presented to the recipient who can then click the appropriate button. But when the unusubscribe action is combined with a user junk report, there is no direct user interaction with the mailer's web site. Similarly, if a mail system automatically unsubscribes recipient mailboxes that have been closed or abandoned, there can be no interaction with a user who is not present. In those cases, the unsubscription process has to work without manual intervention, and in particular without requiring that software attempt to interpret the contents of a confirmation page.

This document addresses this part of the problem, with an HTTPS POST action for mail receivers. Mail senders can distinguish this action from other unsubscribe requests and handle it as a one-click unsubscription without manual intervention by the mail recipient.

This document has several goals.

- Allow email senders to signal that a [RFC2369] List-Unsubscribe header field has One-Click functionality.

- Allow MUA (Mail User Agent) users to unsubscribe from mailing lists in a familiar environment and without leaving the MUA context. A receiving system can process an unsubscription request

in the background without further interaction, and know that it can be fully processed by the mail sender's system.

2. Definitions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119] when written in all capital letters.

3. Implementation

3.1. Mail senders

A mail sender that wishes to enable one-click unsubscriptions places one List-Unsubscribe header field and one List-Unsubscribe-Post header field in the message. The List-Unsubscribe header field MUST contain one HTTPS URI. It MAY contain other non-HTTP/S URIs such as MAILTO:. The List-Unsubscribe-Post header MUST contain the single key/value pair "List-Unsubscribe=One-Click". As described below, the message MUST have a valid DKIM signature that covers at least the List-Unsubscribe and List-Unsubscribe-Post headers.

The URI in the List-Unsubscribe header MUST contain enough information to identify the mail recipient and the list from which the recipient is to be removed, so that the unsubscription process can complete automatically. Since there is no provision for extra POST arguments, any information about the message or recipient is encoded in the URI. In particular, One-click has no way to ask the user what address or from what list the user wishes to unsubscribe.

The POST request MUST NOT include cookies, http authorization, or any other context information. The unsubscribe operation is logically unrelated to any previous Web activity and context information could inappropriately link the unsubscribe to previous activity.

The URI SHOULD include an opaque identifier or other hard to forge component in addition to or instead of the plain-text names of the list and the subscriber. The server handling the unsubscription SHOULD verify that the opaque or hard to forge component is valid. This will deter attacks in which a malicious party sends spam with List-Unsubscribe links for a victim list, with the intention of causing list unsubscriptions from the victim list as a side effect of users reporting the spam, or where the attacker does POSTs directly to the mail sender's unsubscription server.

The mail sender needs to provide the infrastructure to handle POST requests to the specified URI in the List-Unsubscribe header, and to handle the unsubscribe requests that its mail will provoke.

The mail sender MUST NOT return an HTTPS redirect, since redirected POST actions have historically not worked reliably, and many browsers have turned redirected http POSTs into GETs.

This document does not update [RFC2369] so the usage of List-Unsubscribe URIs other than for one-click remains unchanged.

3.2. Mail receivers

A mail receiver can do a one-click unsubscription by performing an HTTPS POST to the HTTPS URI in the List-Unsubscribe header. It sends the key/value pair in the List-Unsubscribe-Post header as the request body.

The POST content SHOULD be sent as "multipart/form-data" [RFC7578] or MAY be sent as "application/x-www-form-urlencoded". These encodings are the ones used by web browsers when sending forms. The target of the POST action is the same as the one in the GET action for a manual unsubscription, so this is intended to allow the same server code to handle both.

The mail receiver MUST NOT perform a POST on the the HTTPS URI without user consent. When and how the user consent is obtained is not part of this specification.

4. Additional Requirements

The message needs at least one valid authentication identifier. In this version of the specification the only supported identifier type is DKIM [RFC6376]. Hence senders MUST apply at least one valid DKIM signature to the message.

The List-Unsubscribe and List-Unsubscribe-Post headers MUST be covered by the signature and included in the "h=" tag of a valid DKIM-Signature header field.

If the message does not have the required DKIM signature, the mail receiver SHOULD NOT offer a one-click unsubscribe for that message.

5. Header Syntax

The following ABNF imports fields, WSP, and CRLF from [RFC5322]. It imports ALPHA and DIGIT from [RFC5234].

```
fields /= list-unsubscribe-post

ldh = ALPHA 0*(ALPHA | DIGIT | "-")

list-unsubscribe-post = "List-Unsubscribe-Post:" 0*1WSP postarg CRLF

postarg = "List-Unsubscribe=One-Click"
```

6. Security Considerations

The List-Unsubscribe header can contain a plaintext or encoded version of the recipient address, but that address is usually also in the To: header. This specification allows anyone with access to a message to unsubscribe the recipient of the message, but that's typically the case with existing List-Unsubscribe, just with more steps.

A malicious mailer could send spam with content intended to provoke large numbers of unsubscriptions, with suitably crafted headers to send POST requests to servers that perhaps don't want them. But it's been possible to provoke GET requests in a similar way for a long time (and much easier, due to spam filter auto-fetches) so the chances of significantly increased annoyance seem low. The contents of the List-Unsubscribe-Post header is limited to a single known key/value pair to prevent an attacker from creating malicious messages where the POST operation could simulate a user filling in an arbitrary form on a victim web site.

The unsubscribe operation provides a strong hint to the mailer that the address to which the message was sent was valid, and could in principle be used as a way to test whether an email address is valid. In practice, though, there are simpler ways such as embedding image links into the HTML of a message and seeing whether the recipient fetches the images.

Since the mailer's server that receives the POST request cannot in general tell where the request is coming from, the URI SHOULD contain an opaque identifier or other hard to forge component to identify the list and recipient address. That can ensure that the request originated from List-Unsubscribe and List-Unsubscribe-Post headers in a message the mailer sent. Also, the request MUST NOT include cookies or other context information to prevent the server from associating the request with previous web requests.

7. IANA Considerations

IANA is requested to add a new entry to the Permanent Message Header Field Names registry.

Header field name: List-Unsubscribe-Post

Applicable protocol: mail

Status: standard

Author/Change controller: IETF

Specification document: this document

8. Examples

8.1. Simple

Header in Email

```
List-Unsubscribe: <https://example.com/unsubscribe/opaquepart>
List-Unsubscribe-Post: List-Unsubscribe=One-Click
```

Resulting POST request

```
POST /unsubscribe/opaquepart HTTP/1.1
Host: example.com
Content-Type: application/x-www-form-urlencoded
Content-Length: 26

List-Unsubscribe=One-Click
```

8.2. Complex

Header in Email

```
List-Unsubscribe: <mailto:listrequest@example.com?subject=unsubscribe>,
    <https://example.com/unsubscribe.html?opaque=123456789>
List-Unsubscribe-Post: List-Unsubscribe=One-Click
```

Resulting POST request

```
POST /unsubscribe.html?opaque=123456789 HTTP/1.1
Host: example.com
Content-Type: application/x-www-form-urlencoded
Content-Length: 26

List-Unsubscribe=One-Click
```

8.3. Complex with multipart/form-data

Header in Email

```
 List-Unsubscribe: <mailto:listrequest@example.com?subject=unsubscribe>,
     <https://example.com/unsubscribe.html/opaque123456789>
 List-Unsubscribe-Post: List-Unsubscribe=One-Click
```

Resulting POST request

```
POST /unsubscribe.html/opaque123456789 HTTP/1.1
Host: example.com
Content-Type: multipart/form-data; boundary=------FormBoundaryjWmhtjORrn
Content-Length: 218

------FormBoundaryjWmhtjORrn
Content-Disposition: form-data; name="List-Unsubscribe"

One-Click
------FormBoundaryjWmhtjORrn--
```

9. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC2369] Neufeld, G. and J. Baer, "The Use of URLs as Meta-Syntax for Core Mail List Commands and their Transport through Message Header Fields", RFC 2369, DOI 10.17487/RFC2369, July 1998, <http://www.rfc-editor.org/info/rfc2369>.

[RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <http://www.rfc-editor.org/info/rfc5234>.

[RFC5322] Resnick, P., Ed., "Internet Message Format", RFC 5322, DOI 10.17487/RFC5322, October 2008, <http://www.rfc-editor.org/info/rfc5322>.

[RFC6376] Crocker, D., Ed., Hansen, T., Ed., and M. Kucherawy, Ed., "DomainKeys Identified Mail (DKIM) Signatures", STD 76, RFC 6376, DOI 10.17487/RFC6376, September 2011, <http://www.rfc-editor.org/info/rfc6376>.

[RFC7230] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", RFC 7230, DOI 10.17487/RFC7230, June 2014, <http://www.rfc-editor.org/info/rfc7230>.

[RFC7578] Masinter, L., "Returning Values from Forms: multipart/form-data", RFC 7578, DOI 10.17487/RFC7578, July 2015, <http://www.rfc-editor.org/info/rfc7578>.

Appendix A. Change Log

Remove this section before publication, please.

A.1. Changes from -09 to -10

Bad cookies. Explain MUA.

A.2. Changes from -08 to -09

Editorial clarifications, strip out obsolete mentions of variable POST arguments.

A.3. Changes from -07 to -08

Editorial changes per Ops directorate and security review.

Simplify POST argument to one field.

Send no context info.

A.4. Changes from -06 to -07

Added example with multipart/form-data encoding

A.5. Changes from -05 to -06

Add opaque parts to the security discussion. Editing changes, entities are now senders and receivers, MUSTage clarified.

A.6. Changes from -04 to -05

Reorganize first sections and add more background. Add ABNF. Add more security advice.

A.7. Changes from -03 to -04

Require HTTPS. More motivation.

A.8. Changes from -02 to -03

Describe motivation in intro. Clarify required DKIM. More paranoid scenarios.

Authors' Addresses

John Levine
Taughannock Networks
PO Box 727
Trumansburg, NY 14886

Phone: +1 831 480 2300
Email: standards@taugh.com
URI: http://jl.ly

Tobias Herkula
optivo GmbH
Wallstrasse 16
Berlin 10179
DE

Phone: +49 30 768078 129
Email: t.herkula@optivo.com
URI: https://www.optivo.com

Dispatch Working Group
Internet Draft
Intended status: Informational
Expires: August 2016

N. Weinronk
Gamma Communications

February 18, 2016

Last Diverting Line Identity
draft-weinronk-dispatch-last-diverting-line-id-00.txt

Status of this Memo

Abstract

This document proposes an extension to the Session Initiation Protocol (SIP).

In cases where applications/services (for example verification / billing) are provided by a network that is not the originating network the Network Asserted Identity is needed to provide these services.

This extension provides the ability for a 'diversion service' to provide a Network Asserted Identity of the last diverting user to these applications/services.

This extension defines a new general header, Last Diverting Line Identity which conveys the Network Asserted Identity of the diverting party to these applications/services.

Table of Contents

1. Introduction

In cases where applications/services (for example verification / billing) are provided by a network that is not the originating network the Network Asserted Identity is needed to provide these services.

This extension provides the ability for a 'diversion service' to provide a Network Asserted Identity of the last diverting user to these applications/services.

This extension defines a new general header, Last Diverting Line Identity which conveys the Network Asserted Identity of the diverting party to these applications/services.

In the legacy telephony network in the UK this information is provided by the Last Diverting Line Identity parameter. Note: This ISUP parameter is defined in the UK under the 'Nationally defined for National User' parameter code range of values.

2. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC-2119 [RFC2119].

In this document, these words will appear with that interpretation only when in ALL CAPS. Lower case uses of these words are not to be interpreted as carrying RFC-2119 significance.

3. Definitions

Diversion:

The 'diversion service' could be defined as in [RFC7044] or as in [TS.24604].

NICC:

The UK Interoperability Standards Organisation.

4. Abbreviations

3GPP - 3rd Generation Partnership Project

ETSI - European Telecommunication Standard Institute

ISDN - Integrated Services Digital Network

ISUP - ISDN User Part

ITU - International Telecommunication Union

SIP - Session Initiation Protocol

TS - Technical Specification

UA - User Agent

UK - United Kingdom

5. Overview

In cases where applications/services (for example verification / billing) are provided by a network that is not the originating network the Network Asserted Identity is needed to provide these services.

This extension provides the ability for a ’diversion service’ to provide a Network Asserted Identity of the last diverting user to these applications/services.

This extension defines a new general header, Last Diverting Line Identity which conveys the Network Asserted Identity of the diverting party to these applications/services.

It could be added by SIP UAs, SIP Redirect Servers or SIP Proxy Servers.

In the legacy telephony network in the UK this information is provided by the Last Diverting Line Identity parameter. Note: This ISUP parameter is defined in the UK under the ’Nationally defined for National User’ parameter code range of values.

Example headers are:

Last-Diverting-Line-Identity: <sip:+441632123456@example.com;user=phone>

Last-Diverting-Line-Identity: <tel:+441632123456>

6. Formal Syntax

The following syntax specification uses the augmented Backus-Naur Form (BNF) as described in RFC-2234 [RFC2234].

Definition of new Last Diverting Line Identity header field:

The Last Diverting Line Identity header field is used among trusted SIP entities (typically intermediaries) to carry the verified identity of the diverting user.

Last-Diverting-Line-Identity = "Last-Diverting-Line-Identity" HCOLON LDLI-value

LDLI-value = name-addr

A Last-Diverting-Line-Identity header field value MUST consist of exactly one name-addr. It MUST be a sip, sips or tel URI.

6.1 The "ldli" Privacy Type

This specification adds a new priv-value to the Privacy header [RFC3323]. The presence of this privacy type in a Privacy header field indicates that the user would like the Last Diverting Line Identity to be kept private with respect to untrusted SIP entities.

priv-value = "ldli"

If the "ldli" priv-value is not present the LDLI-value presentation is allowed.

If the "ldli" priv-value is present then the LDLI-value presentation is restricted.

This document adds the following entry to Table 2 of [RFC3261]:

Header field	where	proxy	ACK	BYE	CAN	INV	OPT	REG
Last-Diverting-Line-Identity		amdr	-	-	-	o	-	-

Header field	where	proxy	SUB	NOT	REF	INF	UPD	PRA
Last-Diverting-Line-Identity		amdr	-	-	-	-	-	-

The Last-Diverting-Line-identity header carries the following information, with the mandatory parameters required when the header is included in a request:

LDLI-value a mandatory parameter for capturing the Last Diverting Line Identity.

7. Why not use existing headers

Use of the last History-Info header entry [RFC7044] was considered however this is mapped to/from the ISUP Redirecting Number and there are cases where the ISUP Redirecting Number is not the Network Asserted Identity of the last diverting user - for example the ETSI ISDN Partial Re-routing service as implemented in the UK.

Note: In the UK the mapping would be to/from the new SIP header and the UK ISUP Last Diverting Line Identity parameter which provides the same functionality in UK ISUP leaving the ISUP Redirecting Number mapping to/from History-Info header as in the existing IETF / 3GPP / ITU / NICC specifications.

8. Security Considerations

This document defines a header field for SIP. The use of the Transport Layer Security (TLS) protocol [RFC5246] as a mechanism to ensure the overall confidentiality of the Last-Diverting-Line-Identity header fields is strongly RECOMMENDED. If TLS is NOT used, the intermediary MUST ensure that the messages are only sent within an environment that is secured by other means or that the messages don’t leave the intermediary’s domain. This results in Last-Diverting-Line-Identity’s having at least the same level of security as other headers in SIP that are inserted by intermediaries. With TLS, Last-Diverting-Line-Identity header fields are no less, nor no more, secure than other SIP header fields, which generally have even more impact on the subsequent processing of SIP sessions than the Last-Diverting-Line-Identity header field.

Note that while using the SIPS scheme (as per [RFC5630]) protects Last-Diverting-Line-Identity from tampering by arbitrary parties outside the SIP message path, all the intermediaries on the path are trusted implicitly. A malicious intermediary could arbitrarily delete, rewrite, or modify Last-Diverting-Line-Identity. This specification does not attempt to prevent or detect attacks by malicious intermediaries.

In terms of ensuring the privacy of LDLI-value, the same security considerations as those described in [RFC3323] apply. The Privacy

Service that's defined in [RFC3323] MUST also support the new Privacy header field priv-value of "ldli".

9. IANA Considerations

9.1. Registration of SIP Last-Diverting-Line-Identity Header

This document defines a new SIP header field name:

Last-Diverting-Line-Identity

The following changes should be made to the header sub-registry under:

http:///www.iana.org/assignments/sip-parameters

The following row has been added to the header field section:

Header Name	Compact Form	Reference
Last-Diverting-Line-Identity	none	[????]

9.2. Registration of "ldli" for SIP Privacy Headers

This document defines a new priv-value for the SIP Privacy header:

ldli

The following changes should be made to

http://www.iana.org/assignments/sip-priv-values

The following has been added to the registration for the SIP Privacy header:

Name	Description	Registrant	Reference
ldli	Privacy requested for Last-Diverting-Line-Identity header	[????]	[????]

10. References

10.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2234] Crocker, D. and Overell, P.(Editors), "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, Internet Mail Consortium and Demon Internet Ltd., November 1997.

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[RFC3323] Peterson, J., "A Privacy Mechanism for the Session Initiation Protocol (SIP)", RFC 3323, November 2002.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

[RFC5630] Audet, F., "The Use of the SIPS URI Scheme in the Session Initiation Protocol (SIP)", RFC 5630, October 2009.

[RFC7044] Barnes, M., Audet, F., Schubert, S., van Elburg, J., and C. Holmberg, "An Extension to the Session Initiation Protocol (SIP) for Request History Information", RFC 7044,February 2014.

[TS.24604] 3GPP, "Communication Diversion (CDIV) using IP Multimedia (IM) Core Network (CN) subsystem; Protocol specification"

11. Acknowledgments

NICC SIP Task Group

This document was prepared using 2-Word-v2.0.template.dot.

Authors' Address

Nigel Weinronk
Gamma Communications

Phone: +443332403421
Email: nigel.weinronk@gamma.co.uk

Contributors' Addresses

Nick Ireland
NICC

Phone: +447889861066

Email: nick.ireland@niccstandards.org.uk

Perry Wilks
BT

Phone: +442087262646
Email: perry.wilks@bt.com