University of Glasgow

School of Computing Science

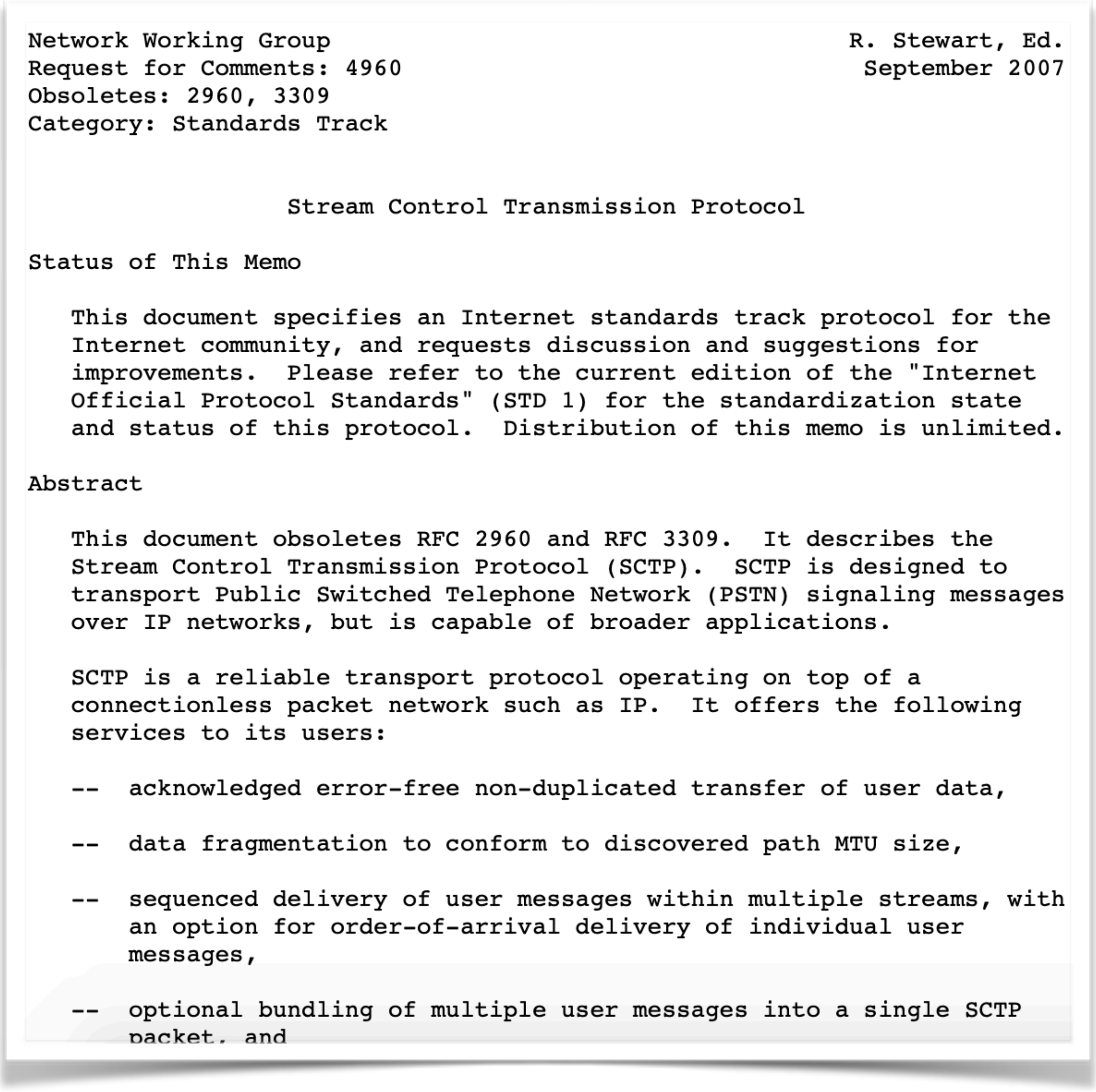
Parsing Protocol Standards

Stephen McQuistin
Colin Perkins

Multi-Service Networks workshop
5th July 2019

IETF protocol standards

- Developed by large groups of people, often remotely
- Process is iterative and incremental
- Output is a document that is mostly English prose
- No good way to automatically verify or validate a standards document
- Inconsistencies & ambiguities in specs → buggy implementations

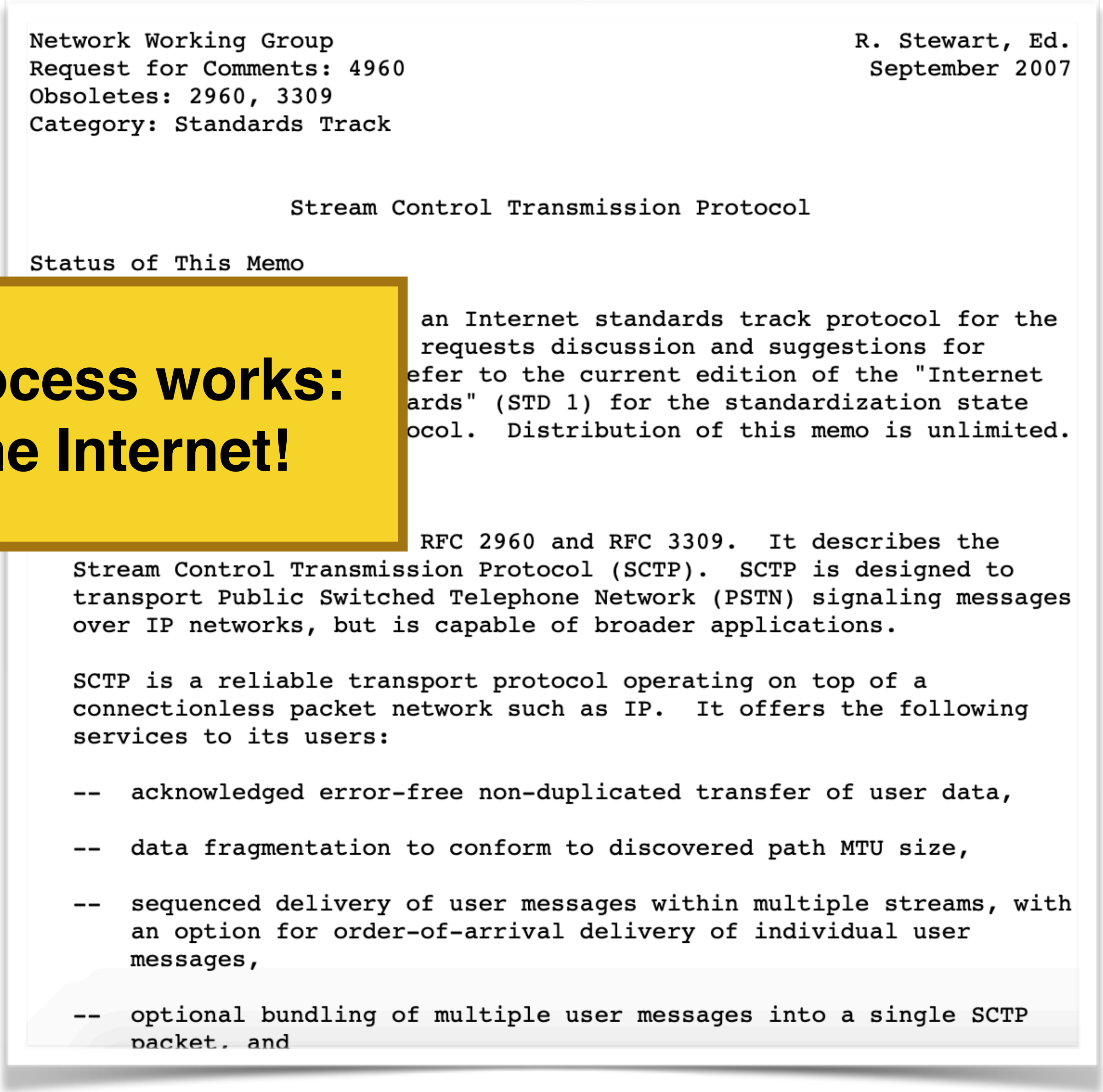


```
Network Working Group                                    R. Stewart, Ed.
Request for Comments: 4960                                September 2007
Obsoletes: 2960, 3309
Category: Standards Track


                  Stream Control Transmission Protocol

Status of This Memo

   This document specifies an Internet standards track protocol for the
   Internet community, and requests discussion and suggestions for
   improvements.  Please refer to the current edition of the "Internet
   Official Protocol Standards" (STD 1) for the standardization state
   and status of this protocol.  Distribution of this memo is unlimited.

Abstract

   This document obsoletes RFC 2960 and RFC 3309.  It describes the
   Stream Control Transmission Protocol (SCTP).  SCTP is designed to
   transport Public Switched Telephone Network (PSTN) signaling messages
   over IP networks, but is capable of broader applications.

   SCTP is a reliable transport protocol operating on top of a
   connectionless packet network such as IP.  It offers the following
   services to its users:

   --  acknowledged error-free non-duplicated transfer of user data,

   --  data fragmentation to conform to discovered path MTU size,

   --  sequenced delivery of user messages within multiple streams, with
       an option for order-of-arrival delivery of individual user
       messages,

   --  optional bundling of multiple user messages into a single SCTP
       packet, and
```

IETF protocol standards

- Developed by large groups of people, often remotely
- Process is iterative and incremental
- Output is a document
  English prose
- No good way to automatically verify or validate a standards document
- Inconsistencies & ambiguities in spec → buggy implementations

.. but the process works: we have the Internet!



```
Network Working Group                                    R. Stewart, Ed.
Request for Comments: 4960                                September 2007
Obsoletes: 2960, 3309
Category: Standards Track


                  Stream Control Transmission Protocol

Status of This Memo

                             an Internet standards track protocol for the
                             requests discussion and suggestions for
                            efer to the current edition of the "Internet
                            ards" (STD 1) for the standardization state
                            ocol.  Distribution of this memo is unlimited.



                             RFC 2960 and RFC 3309.  It describes the
   Stream Control Transmission Protocol (SCTP).  SCTP is designed to
   transport Public Switched Telephone Network (PSTN) signaling messages
   over IP networks, but is capable of broader applications.

   SCTP is a reliable transport protocol operating on top of a
   connectionless packet network such as IP.  It offers the following
   services to its users:

   --  acknowledged error-free non-duplicated transfer of user data,

   --  data fragmentation to conform to discovered path MTU size,

   --  sequenced delivery of user messages within multiple streams, with
       an option for order-of-arrival delivery of individual user
       messages,

   --  optional bundling of multiple user messages into a single SCTP
       packet, and
```

Improving protocol standards

- **Goal: shift towards a test-driven development style approach, where running a suite of validation and verification tools over a standards document becomes commonplace**

- Don’t want to replace the process, but to augment it

Describing protocol parsing

- First aim: build a tool that allows for a parser for the specified protocol to be generated automatically

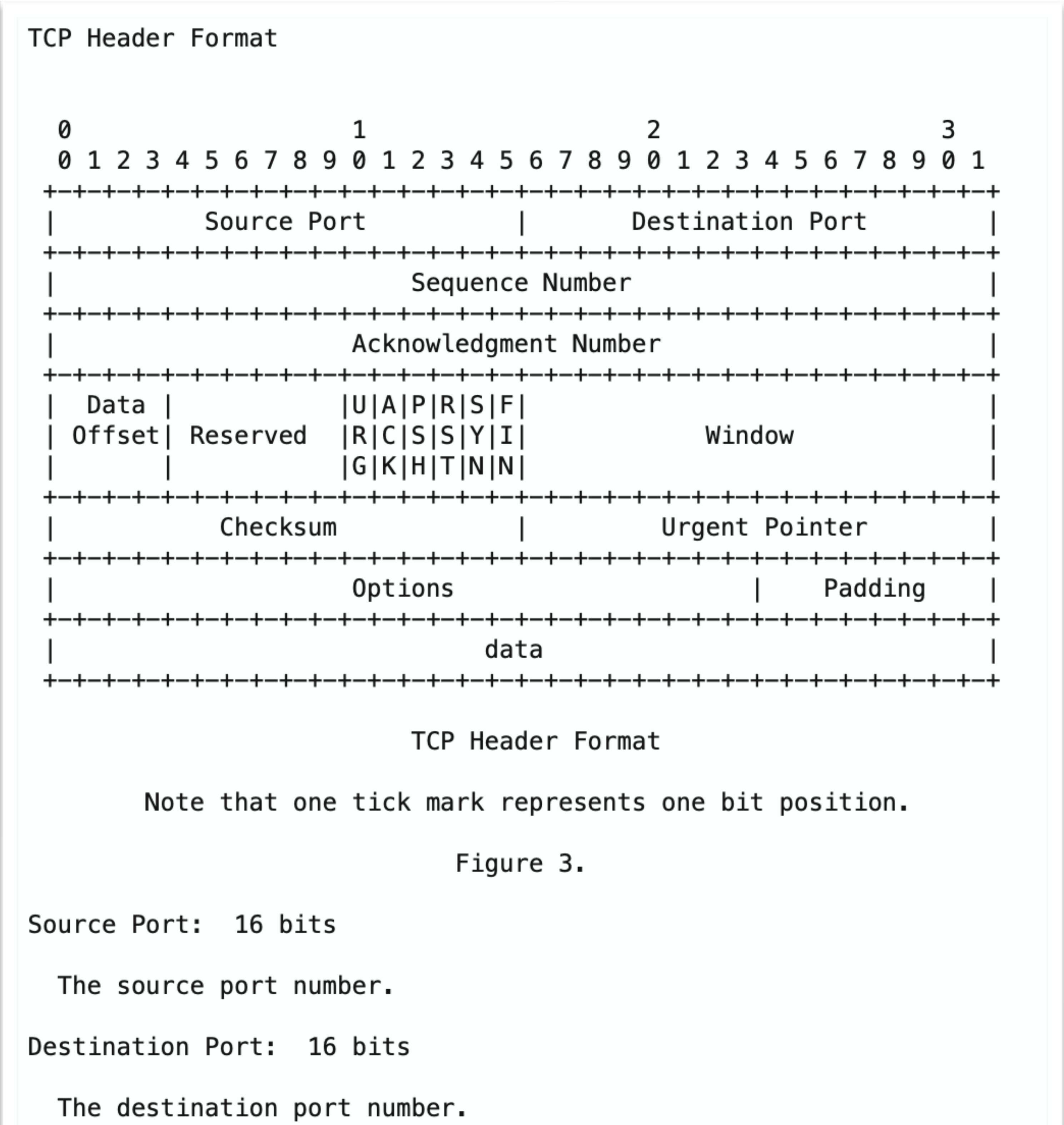
- Need a machine-readable description of the protocol’s data units, and all the metadata needed to parse them

- Good place to start: knowing what the protocol looks like forms the basis of more complex tools

Design principles

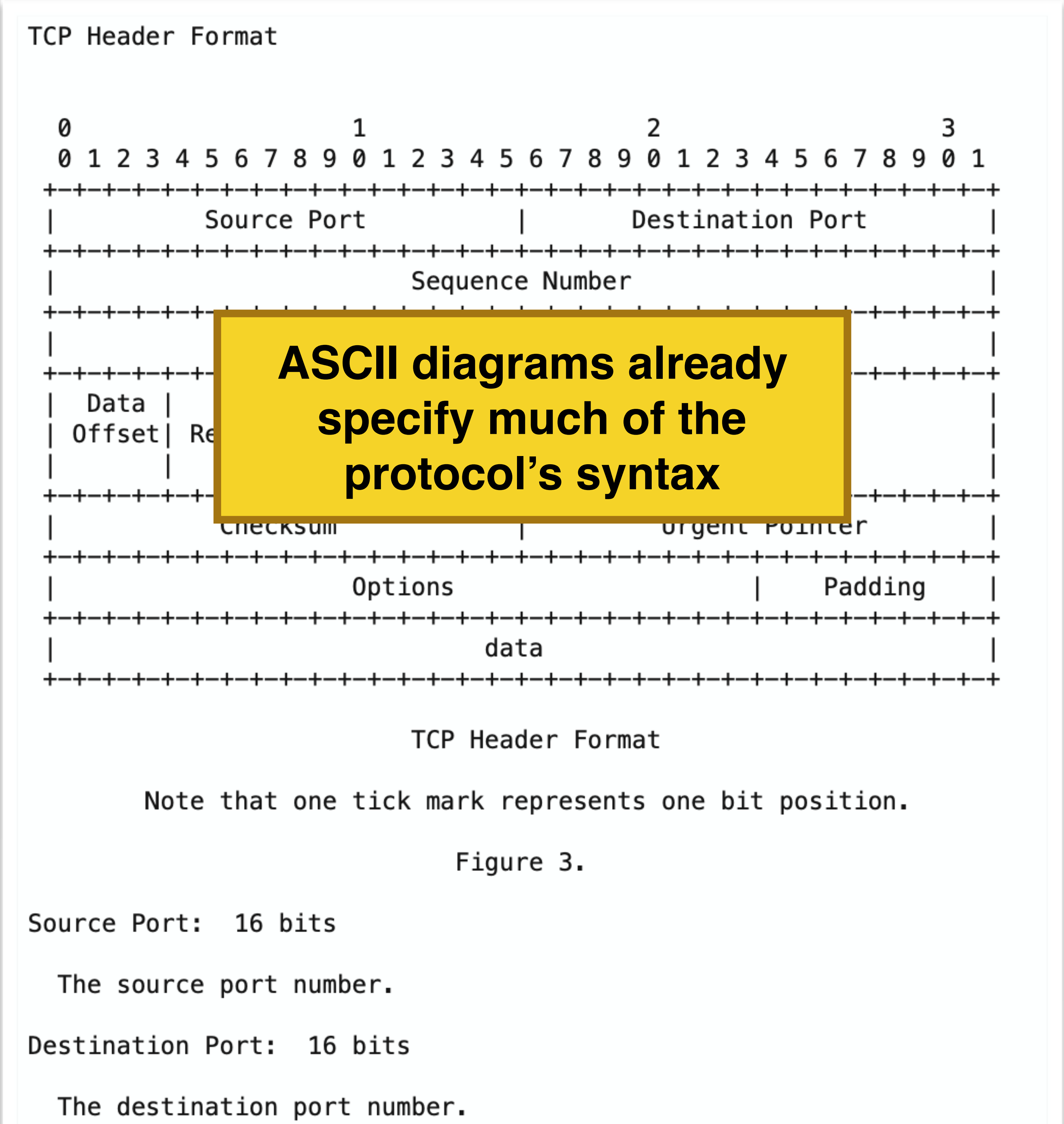
- Most readers are human
- Authorship tools are diverse
- Canonical specifications
- Expressiveness
- Minimise required change

ASCII packet diagrams



```
TCP Header Format


  0                   1                   2                   3
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |          Source Port          |       Destination Port        |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                        Sequence Number                        |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                    Acknowledgment Number                      |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |  Data |           |U|A|P|R|S|F|                               |
 | Offset| Reserved  |R|C|S|S|Y|I|            Window             |
 |       |           |G|K|H|T|N|N|                               |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |           Checksum            |         Urgent Pointer        |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                    Options                    |    Padding    |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                             data                              |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

                          TCP Header Format

        Note that one tick mark represents one bit position.

                             Figure 3.

Source Port:  16 bits

  The source port number.

Destination Port:  16 bits

  The destination port number.
```

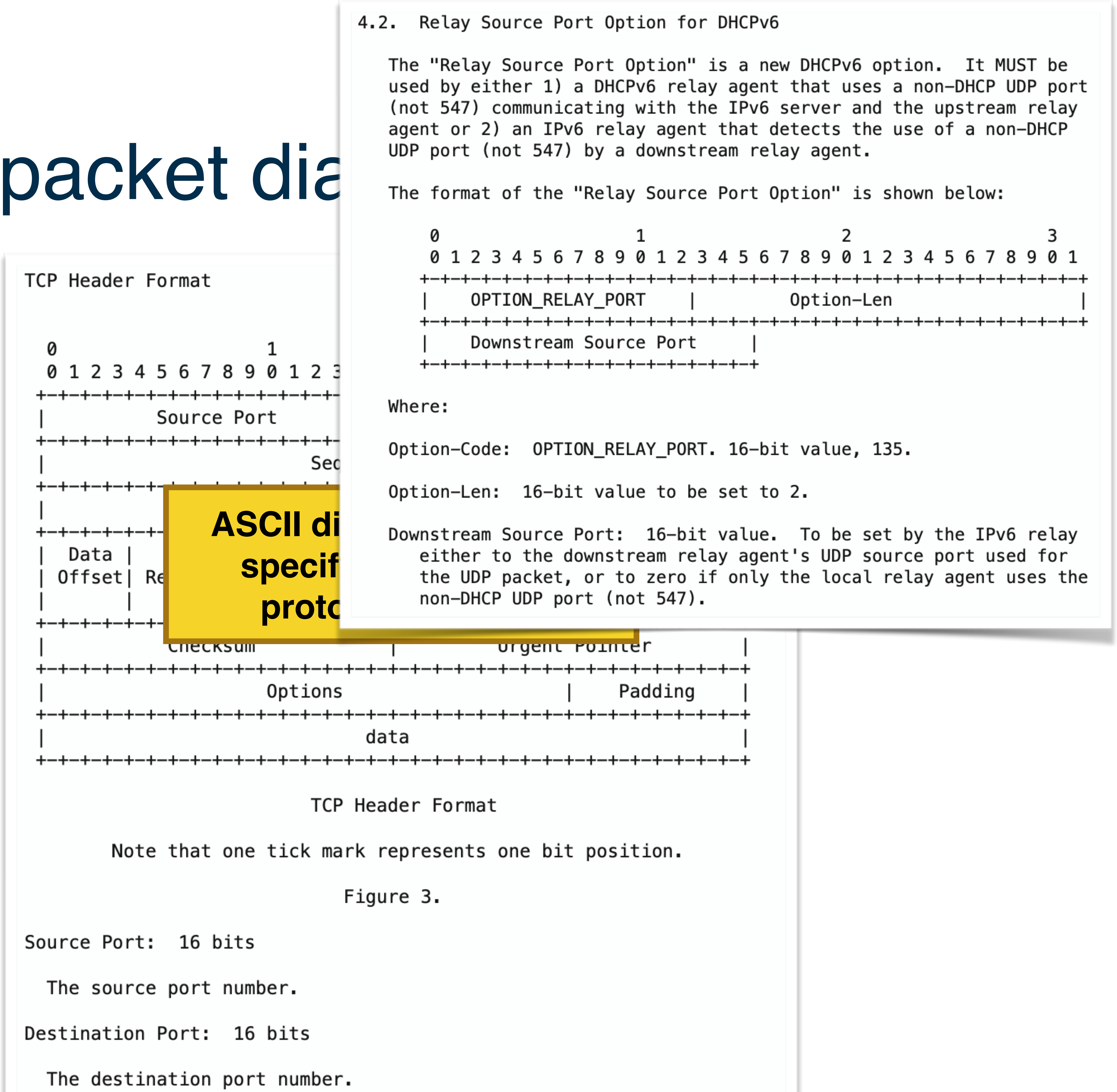
ASCII packet diagrams



```
TCP Header Format


    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |          Source Port          |       Destination Port        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                        Sequence Number                        |
   +-+-+-+-+-+-                                          -+-+-+-+-+
   |                                                               |
   +-+-+-+-+-+-                                          -+-+-+-+-+
   |  Data |                                                       |
   | Offset| Re                                                    |
   |       |                                                       |
   +-+-+-+-+-+-                                          -+-+-+-+-+
   |           Checksum            |         Urgent Pointer        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                    Options                    |    Padding    |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                             data                              |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

                            TCP Header Format

          Note that one tick mark represents one bit position.

                               Figure 3.

  Source Port:  16 bits

    The source port number.

  Destination Port:  16 bits

    The destination port number.
```

ASCII diagrams already specify much of the protocol's syntax

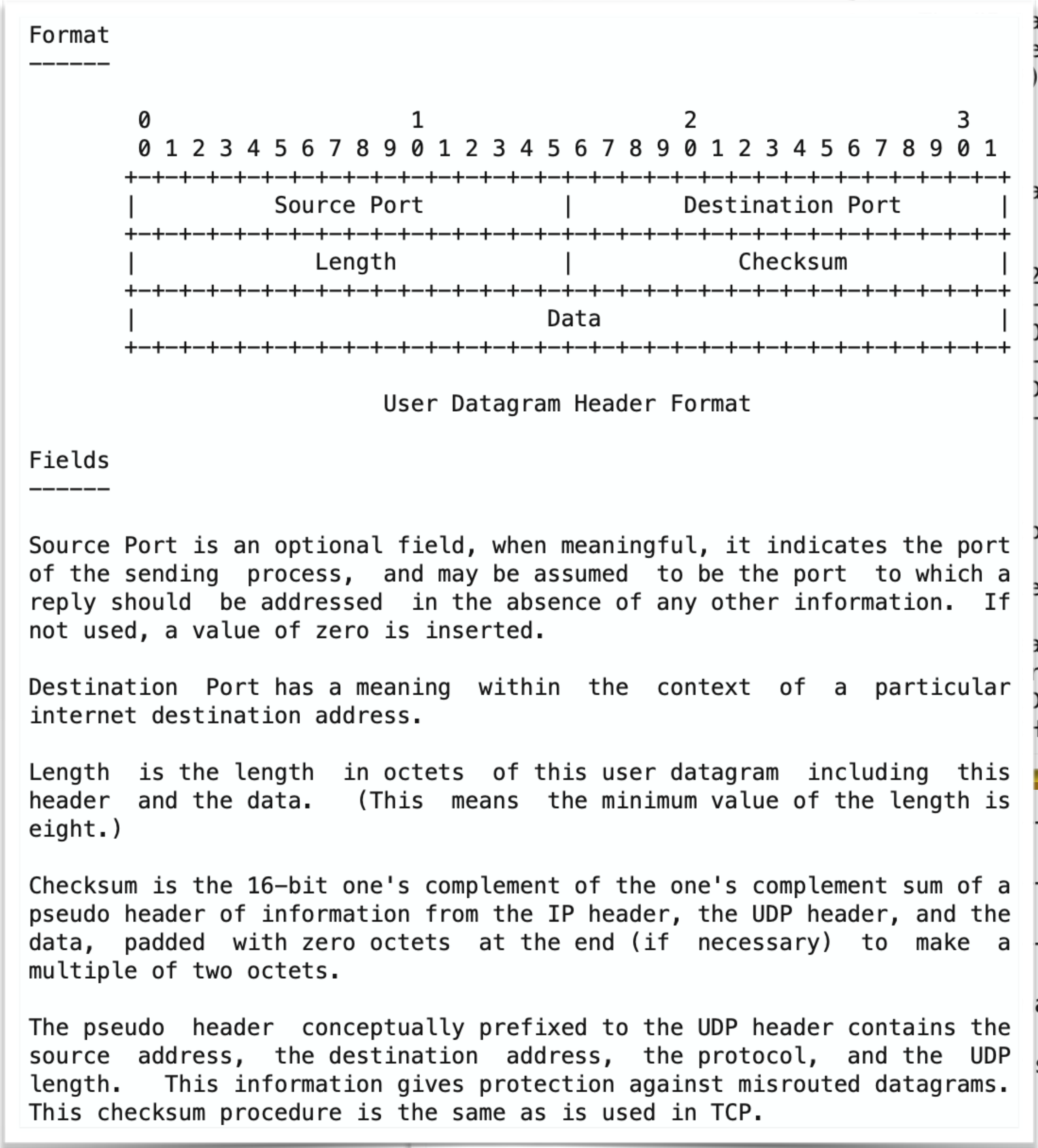
ASCII packet dia



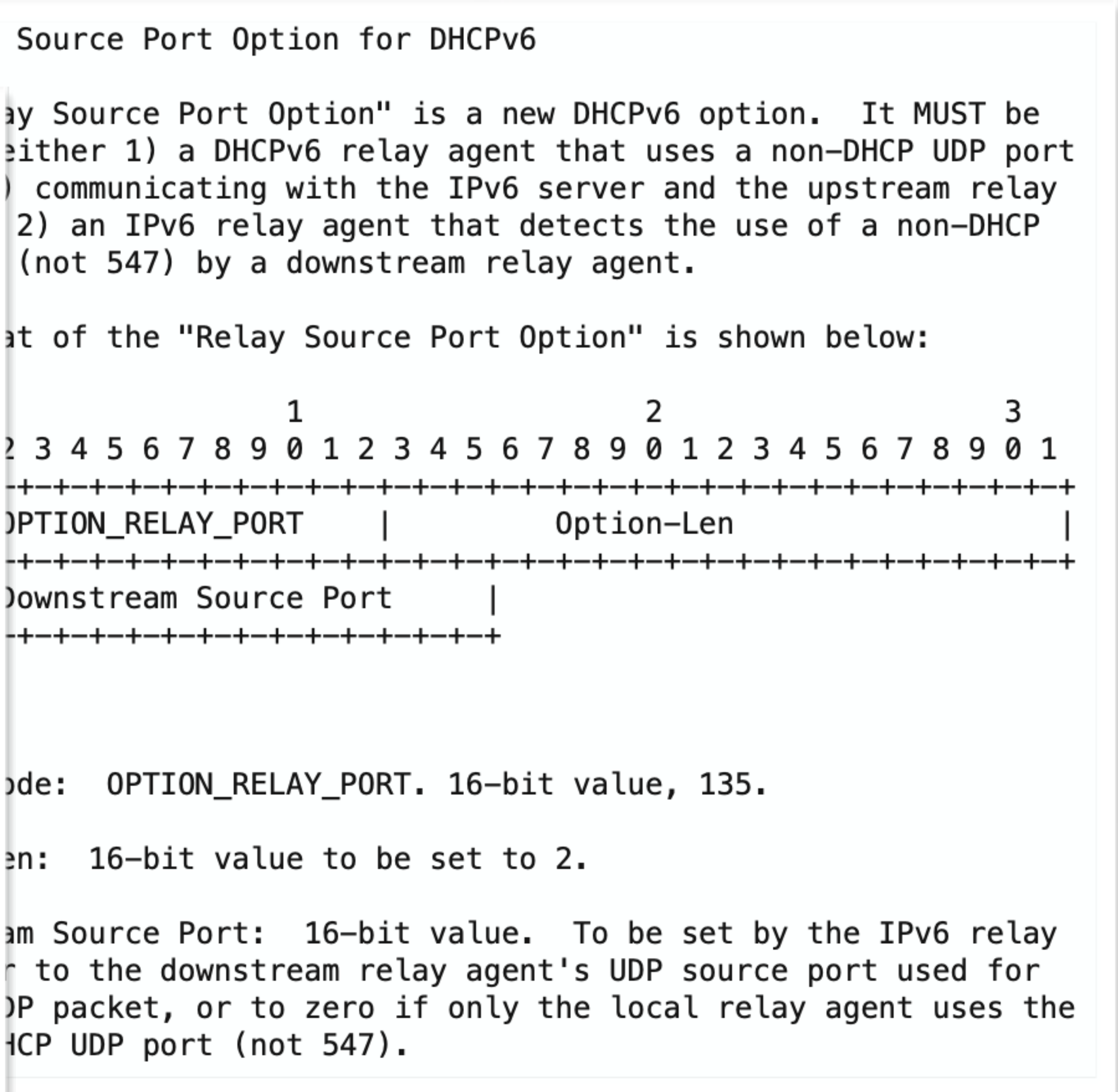
```
4.2.  Relay Source Port Option for DHCPv6

   The "Relay Source Port Option" is a new DHCPv6 option.  It MUST be
   used by either 1) a DHCPv6 relay agent that uses a non-DHCP UDP port
   (not 547) communicating with the IPv6 server and the upstream relay
   agent or 2) an IPv6 relay agent that detects the use of a non-DHCP
   UDP port (not 547) by a downstream relay agent.

   The format of the "Relay Source Port Option" is shown below:

       0                   1                   2                   3
       0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |    OPTION_RELAY_PORT      |         Option-Len                |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |    Downstream Source Port     |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

   Where:

   Option-Code:  OPTION_RELAY_PORT. 16-bit value, 135.

   Option-Len:  16-bit value to be set to 2.

   Downstream Source Port:  16-bit value.  To be set by the IPv6 relay
      either to the downstream relay agent's UDP source port used for
      the UDP packet, or to zero if only the local relay agent uses the
      non-DHCP UDP port (not 547).
```

ASCII di
specif
proto

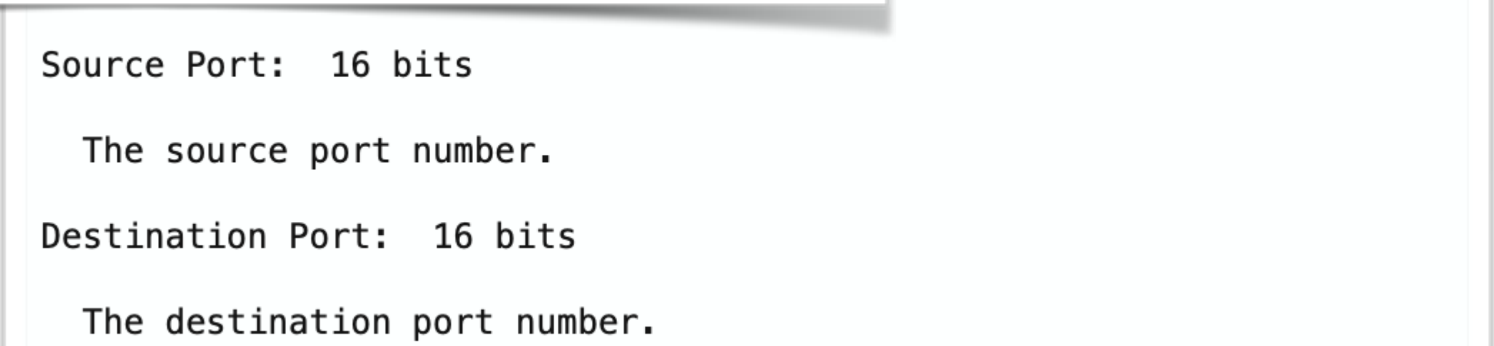
```
TCP Header Format


  0                   1
  0 1 2 3 4 5 6 7 8 9 0 1 2 3
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
 |          Source Port
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
 |                            Seq
 +-+-+-+-+-+-
 |
 +-+-+-+-+-+-
 |  Data |
 | Offset| Re
 |       |
 +-+-+-+-+-+-
 |           Checksum            |         Urgent Pointer        |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                    Options                    |    Padding    |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                             data                              |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

                          TCP Header Format

          Note that one tick mark represents one bit position.

                               Figure 3.

Source Port:  16 bits

  The source port number.

Destination Port:  16 bits

  The destination port number.
```



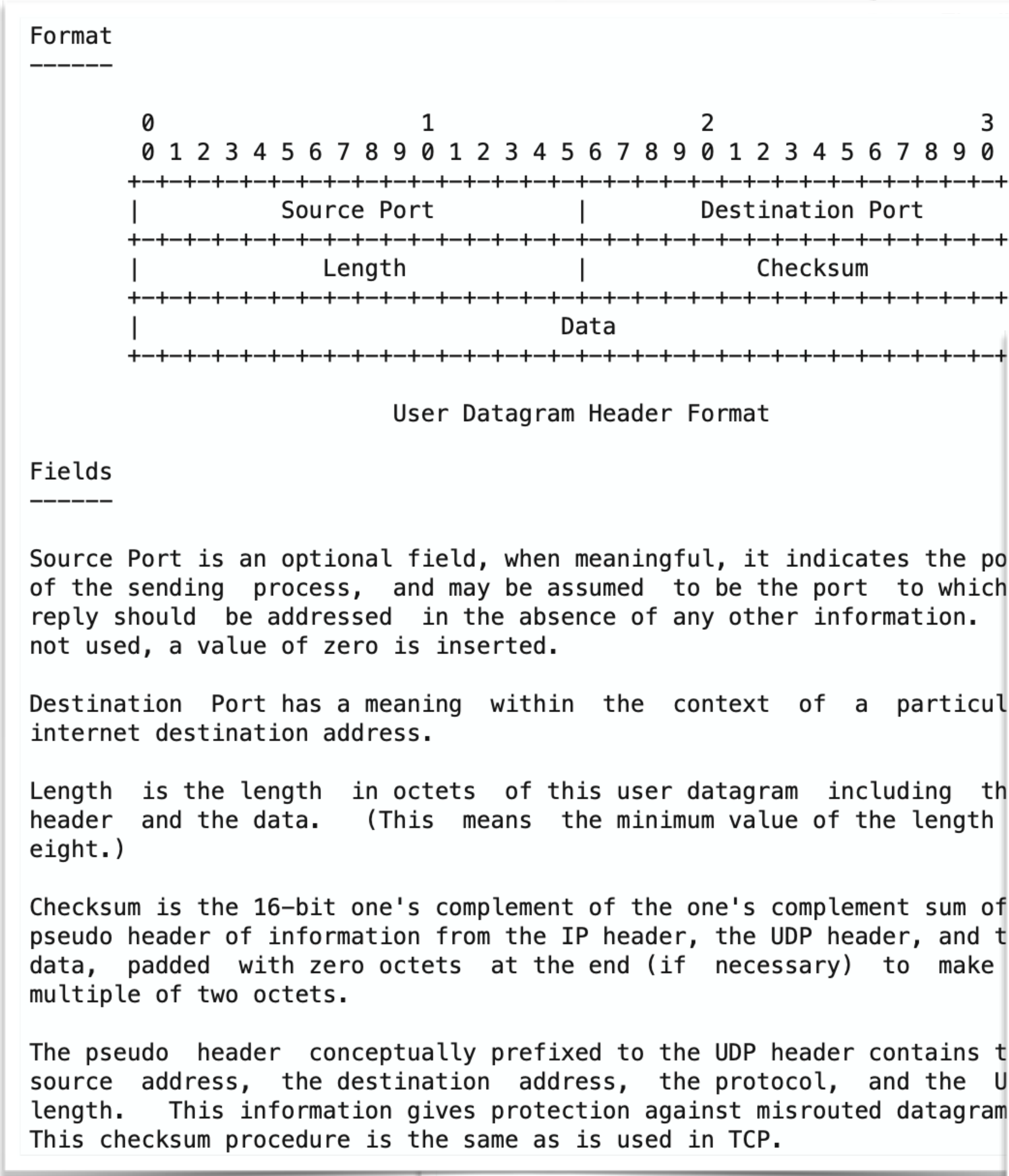
```
Format
------

         0                   1                   2                   3
         0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
        |          Source Port          |       Destination Port        |
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
        |            Length             |           Checksum            |
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
        |                             Data                              |
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

                          User Datagram Header Format

Fields
------

Source Port is an optional field, when meaningful, it indicates the port
of the sending  process,  and may be assumed  to be the port  to which a
reply should  be addressed  in the absence of any other information.  If
not used, a value of zero is inserted.

Destination  Port has a meaning  within  the  context  of  a  particular
internet destination address.

Length  is the length  in octets  of this user datagram  including  this
header  and the data.   (This  means  the minimum value of the length is
eight.)

Checksum is the 16-bit one's complement of the one's complement sum of a
pseudo header of information from the IP header, the UDP header, and the
data,  padded  with zero octets  at the end (if  necessary)  to  make  a
multiple of two octets.

The pseudo  header  conceptually prefixed to the UDP header contains the
source  address,  the destination  address,  the protocol,  and the  UDP
length.   This information gives protection against misrouted datagrams.
This checksum procedure is the same as is used in TCP.
```



```
4.2.  Relay Source Port Option for DHCPv6

ay Source Port Option" is a new DHCPv6 option.  It MUST be
either 1) a DHCPv6 relay agent that uses a non-DHCP UDP port
) communicating with the IPv6 server and the upstream relay
 2) an IPv6 relay agent that detects the use of a non-DHCP
 (not 547) by a downstream relay agent.

at of the "Relay Source Port Option" is shown below:

                1                   2                   3
2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
PTION_RELAY_PORT     |          Option-Len                  |
-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
ownstream Source Port      |
-+-+-+-+-+-+-+-+-+-+-+-+-+-+



ode:  OPTION_RELAY_PORT. 16-bit value, 135.

en:  16-bit value to be set to 2.

am Source Port:  16-bit value.  To be set by the IPv6 relay
 to the downstream relay agent's UDP source port used for
P packet, or to zero if only the local relay agent uses the
HCP UDP port (not 547).
```



```
 Urgent Pointer             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
          |    Padding      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

at

s one bit position.

Source Port:  16 bits

  The source port number.

Destination Port:  16 bits

  The destination port number.
```



Format

```
         0                   1                   2                   3
         0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
        |          Source Port          |       Destination Port        |
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
        |            Length             |           Checksum            |
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
        |                             Data                              |
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

                        User Datagram Header Format
```

Fields

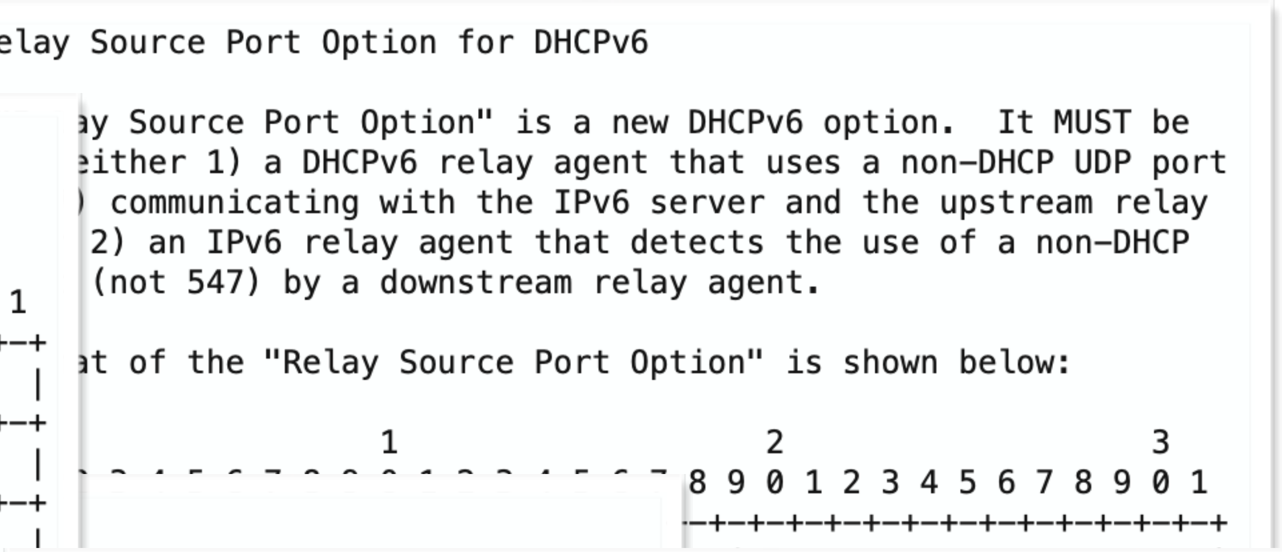
Source Port is an optional field, when meaningful, it indicates the po
of the sending process, and may be assumed to be the port to which
reply should be addressed in the absence of any other information.
not used, a value of zero is inserted.

Destination Port has a meaning within the context of a particul
internet destination address.

Length is the length in octets of this user datagram including th
header and the data. (This means the minimum value of the length
eight.)

Checksum is the 16-bit one's complement of the one's complement sum of
pseudo header of information from the IP header, the UDP header, and t
data, padded with zero octets at the end (if necessary) to make
multiple of two octets.

The pseudo header conceptually prefixed to the UDP header contains t
source address, the destination address, the protocol, and the U
length. This information gives protection against misrouted datagram
This checksum procedure is the same as is used in TCP.

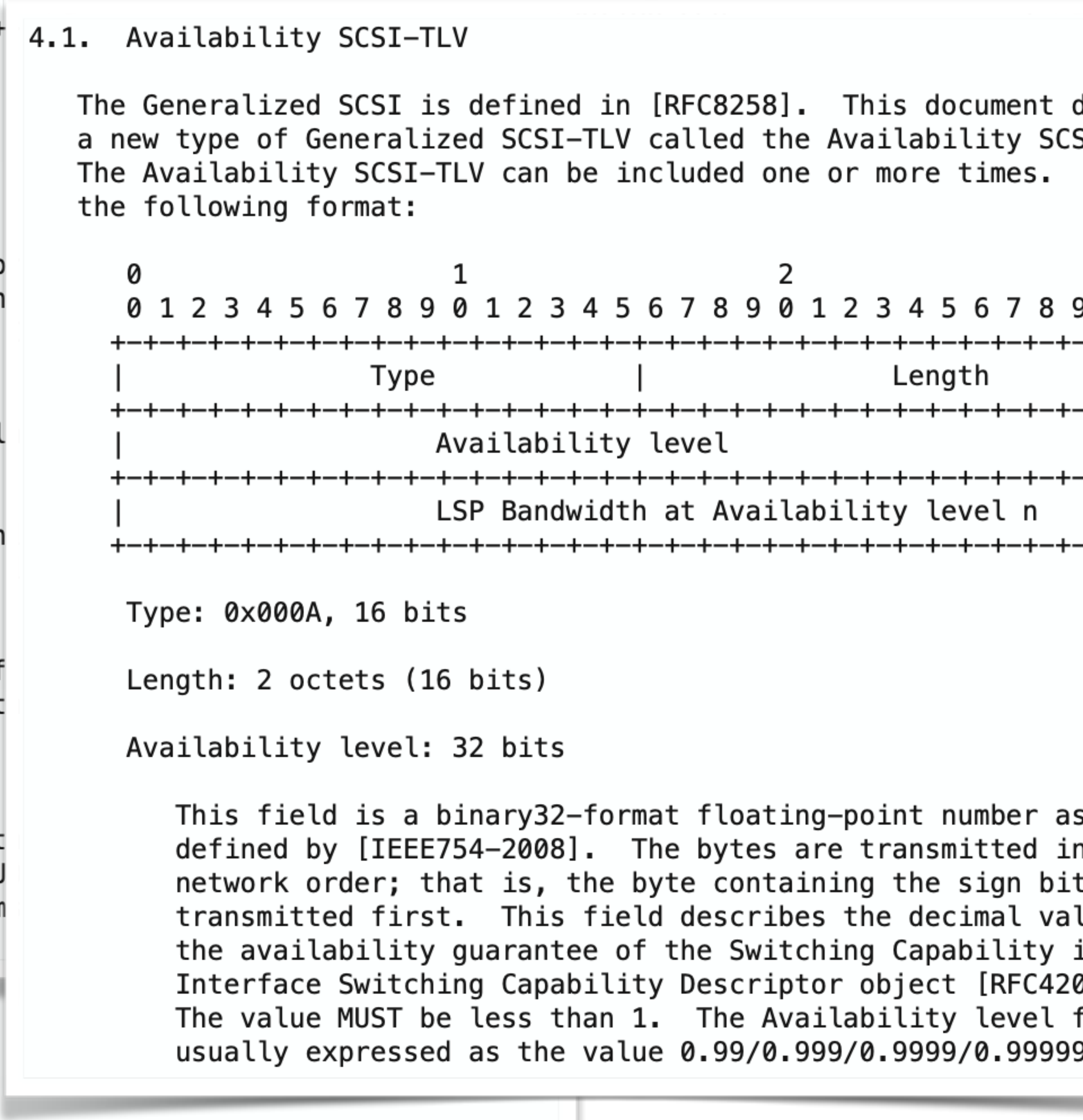


4.2. Relay Source Port Option for DHCPv6

ay Source Port Option" is a new DHCPv6 option. It MUST be
either 1) a DHCPv6 relay agent that uses a non-DHCP UDP port
) communicating with the IPv6 server and the upstream relay
2) an IPv6 relay agent that detects the use of a non-DHCP
(not 547) by a downstream relay agent.

at of the "Relay Source Port Option" is shown below:

```
                 1                   2                   3
                       8 9 0 1 2 3 4 5 6 7 8 9 0 1
                       -+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```



4.1. Availability SCSI-TLV

The Generalized SCSI is defined in [RFC8258]. This document d
a new type of Generalized SCSI-TLV called the Availability SCS
The Availability SCSI-TLV can be included one or more times.
the following format:

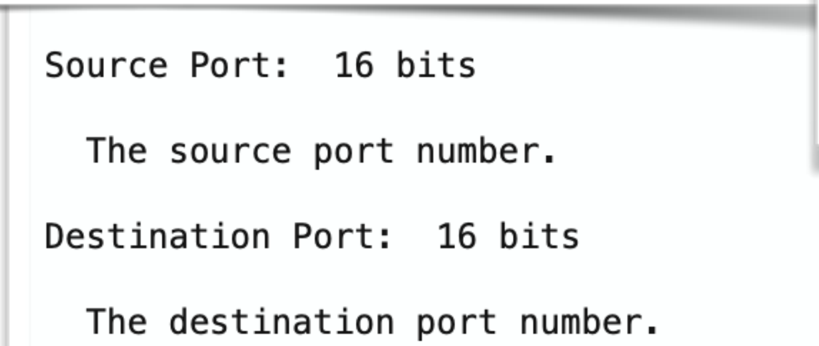
```
 0                   1                   2
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Type             |              Length
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Availability level
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    LSP Bandwidth at Availability level n
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type: 0x000A, 16 bits

Length: 2 octets (16 bits)

Availability level: 32 bits

This field is a binary32-format floating-point number as
defined by [IEEE754-2008]. The bytes are transmitted in
network order; that is, the byte containing the sign bit
transmitted first. This field describes the decimal val
the availability guarantee of the Switching Capability i
Interface Switching Capability Descriptor object [RFC420
The value MUST be less than 1. The Availability level f
usually expressed as the value 0.99/0.999/0.9999/0.99999



Source Port: 16 bits

The source port number.

Destination Port: 16 bits

The destination port number.

4.2. Relay Source Port Option for DHCPv6

ay Source Port Option" is a new DHCPv6 option. It MUST be
either 1) a DHCPv6 relay agent that uses a non-DHCP UDP port
) communicating with the IPv6 server and the upstream relay
2) an IPv6 relay agent that detects the use of a non-DHCP
(not 547) by a downstream relay agent.

at of the "Relay Source Port Option" is shown below:

```
                    1                   2                   3
                                   8 9 0 1 2 3 4 5 6 7 8 9 0 1
                                   -+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Format

```
         0                   1                   2                   3
         0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
        |          Source Port          |       Destination Port        |
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
        |            Length             |           Checksum            |
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
        |                              Data                             |
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

User Datagram Header Format

Fields

Source Port is an optional field, when meaningful, it indicates the po
of the sending process, and may be assumed to be the port to which
reply should be addressed in the absence of any other information.
not used, a value of zero is inserted.

Destination Port has a meaning within the context of a particul
internet destination address.

uding th
e length

nt sum of
er, and t
to make

ontains t
nd the U
datagram

e following is the format of the MRT Capability Parameter.

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
U|F| MRT Capability (0x050E)   |      Length (= 1)             |
-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
S| Reserved    |
-+-+-+-+-+-+-+-+
```

MRT Capability TLV Format

ere:

bit: The unknown TLV bit MUST be 1. A router that does not
recognize the MRT Capability TLV will silently ignore the TLV and
process the rest of the message as if the unknown TLV did not
exist.

bit: The forward unknown TLV bit MUST be 0 as required by
Section 3 of [RFC5561].

4.1. Availability SCSI-TLV

The Generalized SCSI is defined in [RFC8258]. This document d
a new type of Generalized SCSI-TLV called the Availability SCS
The Availability SCSI-TLV can be included one or more times.
the following format:

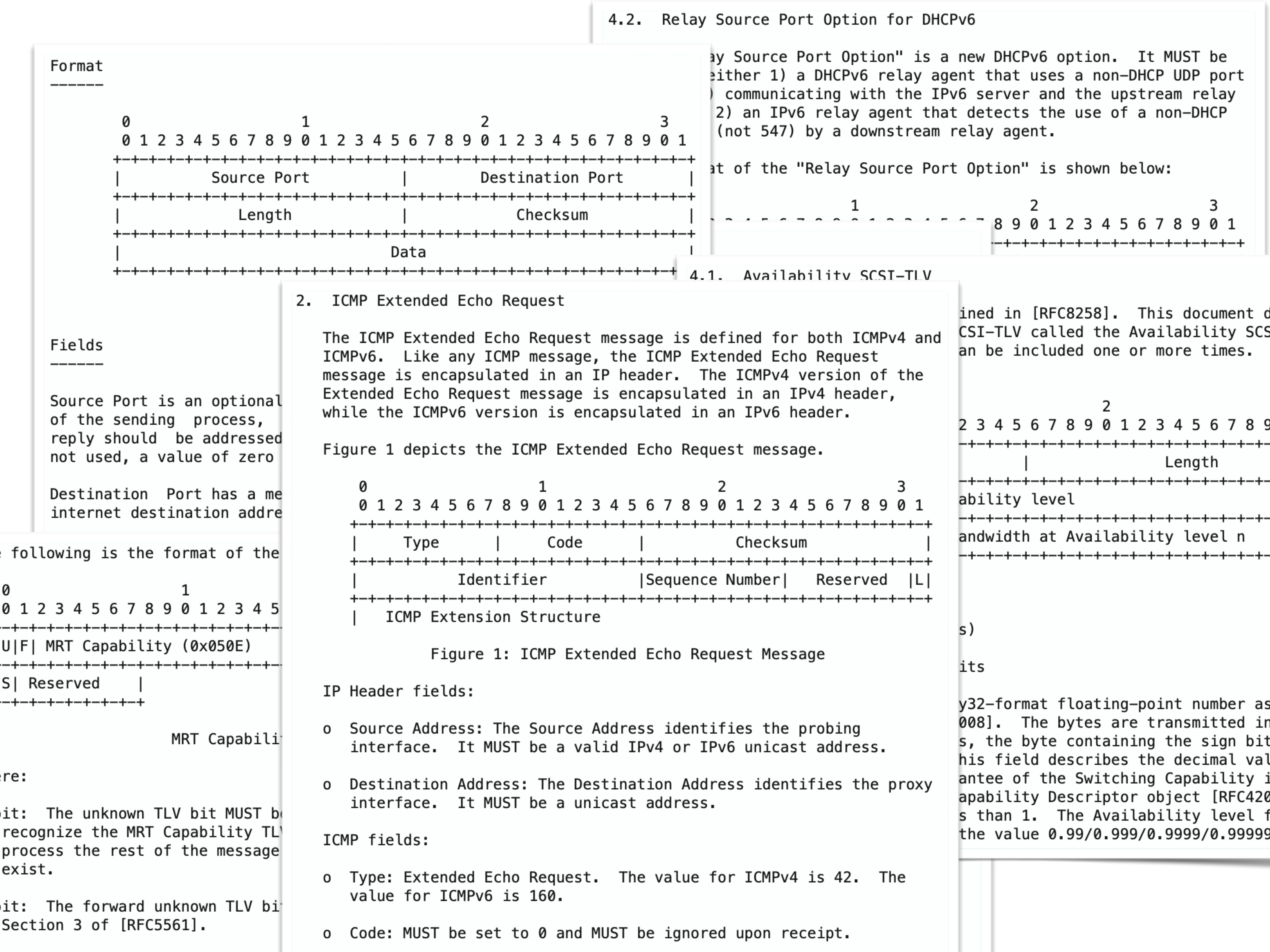
```
 0                   1                   2
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               Type            |               Length
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Availability level
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     LSP Bandwidth at Availability level n
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type: 0x000A, 16 bits

Length: 2 octets (16 bits)

Availability level: 32 bits

This field is a binary32-format floating-point number as
defined by [IEEE754-2008]. The bytes are transmitted in
network order; that is, the byte containing the sign bit
transmitted first. This field describes the decimal val
the availability guarantee of the Switching Capability i
Interface Switching Capability Descriptor object [RFC420
The value MUST be less than 1. The Availability level f
usually expressed as the value 0.99/0.999/0.9999/0.99999



Format

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Source Port          |       Destination Port        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Length             |           Checksum            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             Data                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Fields

Source Port is an optional
of the sending process,
reply should be addressed
not used, a value of zero

Destination Port has a me
internet destination addre

4.2. Relay Source Port Option for DHCPv6

ay Source Port Option" is a new DHCPv6 option. It MUST be
either 1) a DHCPv6 relay agent that uses a non-DHCP UDP port
) communicating with the IPv6 server and the upstream relay
 2) an IPv6 relay agent that detects the use of a non-DHCP
 (not 547) by a downstream relay agent.

at of the "Relay Source Port Option" is shown below:

```
                    1                   2                   3
                       8 9 0 1 2 3 4 5 6 7 8 9 0 1
                      -+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

2. ICMP Extended Echo Request

The ICMP Extended Echo Request message is defined for both ICMPv4 and ICMPv6. Like any ICMP message, the ICMP Extended Echo Request message is encapsulated in an IP header. The ICMPv4 version of the Extended Echo Request message is encapsulated in an IPv4 header, while the ICMPv6 version is encapsulated in an IPv6 header.

Figure 1 depicts the ICMP Extended Echo Request message.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |     Code      |          Checksum             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Identifier          |Sequence Number|   Reserved  |L|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  ICMP Extension Structure
```

Figure 1: ICMP Extended Echo Request Message

IP Header fields:

o Source Address: The Source Address identifies the probing interface. It MUST be a valid IPv4 or IPv6 unicast address.

o Destination Address: The Destination Address identifies the proxy interface. It MUST be a unicast address.

ICMP fields:

o Type: Extended Echo Request. The value for ICMPv4 is 42. The value for ICMPv6 is 160.

o Code: MUST be set to 0 and MUST be ignored upon receipt.

4.1. Availability SCSI-TLV

ined in [RFC8258]. This document d
CSI-TLV called the Availability SCS
an be included one or more times.

```
                2
2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9
-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
    |                Length
-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
ability level
-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
andwidth at Availability level n
-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

s)

its

y32-format floating-point number as
008]. The bytes are transmitted in
s, the byte containing the sign bit
his field describes the decimal val
antee of the Switching Capability i
apability Descriptor object [RFC420
s than 1. The Availability level f
the value 0.99/0.999/0.9999/0.99999

 following is the format of the

```
0                   1
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
U|F| MRT Capability (0x050E)
-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
S| Reserved     |
-+-+-+-+-+-+-+-+
```

MRT Capabili

ere:

it: The unknown TLV bit MUST b
recognize the MRT Capability TL
process the rest of the message
exist.

it: The forward unknown TLV bi
Section 3 of [RFC5561].

The FEC type for the P2MP PW Upstream FEC Element is encoded as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|P2MP PW Up=0x82|C|             PW Type         | PW Info Length|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    AGI Type   | AGI Length    |           AGI Value           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                       AGI Value (contd.)                      ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   AII Type    | SAII Length   |           SAII Value          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                       SAII Value (contd.)                     ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|PMSI Tunnel Typ|PMSI TT Length |                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               +
+                                                               +
~                    Transport LSP ID                           ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                       Optional Parameters                     |
~                                                               ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: P2MP PW Upstream FEC Element

* P2MP PW Up:

  8-bit representation for the P2MP PW Upstream FEC type.

* C bit:

  A value of 1 or 0 indicating whether a control word is present or absent for the P2MP PW.

Relay Source Port Option for DHCPv6

ay Source Port Option" is a new DHCPv6 option. It MUST be
either 1) a DHCPv6 relay agent that uses a non-DHCP UDP port
) communicating with the IPv6 server and the upstream relay
2) an IPv6 relay agent that detects the use of a non-DHCP
(not 547) by a downstream relay agent.

at of the "Relay Source Port Option" is shown below:

```
              1                   2                   3
               8 9 0 1 2 3 4 5 6 7 8 9 0 1
              -+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
3
9 0 1
-+-+-+
     |
-+-+-+
     |
-+-+-+
     |
-+-+
```

4.1. Availability SCSI-TLV

ined in [RFC8258]. This document d
CSI-TLV called the Availability SCS
an be included one or more times.

```
       2
2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9
-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |          Length
-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
ability level
-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
andwidth at Availability level n
-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

s)

its

y32-format floating-point number as
008]. The bytes are transmitted in
s, the byte containing the sign bit
his field describes the decimal val
antee of the Switching Capability i
apability Descriptor object [RFC420
s than 1. The Availability level f
the value 0.99/0.999/0.9999/0.99999

age is defined for both ICMPv4 and
ICMP Extended Echo Request
ader. The ICMPv4 version of the
capsulated in an IPv4 header,
lated in an IPv6 header.

Echo Request message.

```
        2                   3
6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Checksum            |
-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Sequence Number|   Reserved |L|
-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Echo Request Message

ss identifies the probing
interface. It MUST be a valid IPv4 or IPv6 unicast address.

o Destination Address: The Destination Address identifies the proxy interface. It MUST be a unicast address.

ICMP fields:

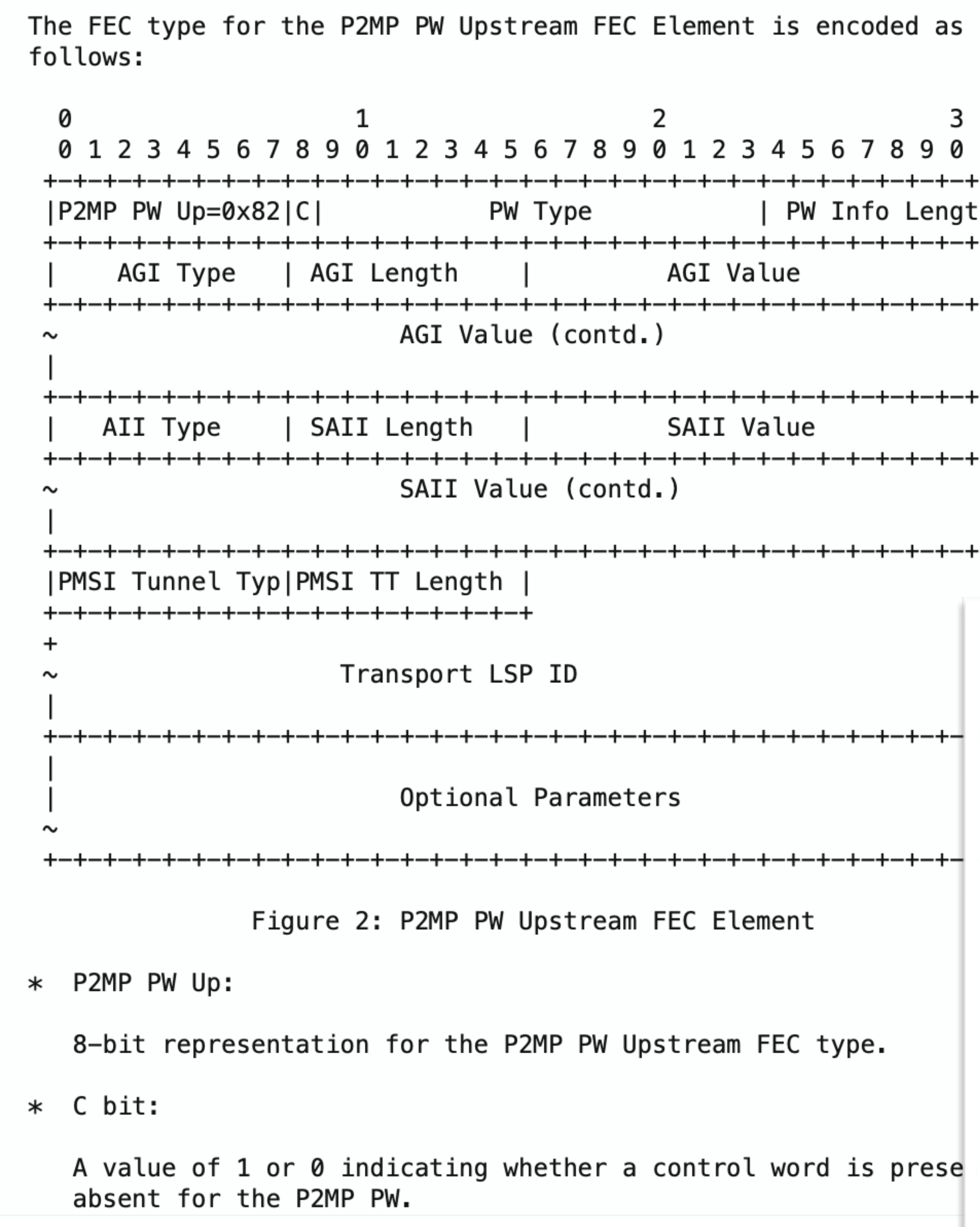
o Type: Extended Echo Request. The value for ICMPv4 is 42. The value for ICMPv6 is 160.

o Code: MUST be set to 0 and MUST be ignored upon receipt.

ere:

bit: The unknown TLV bit MUST b
recognize the MRT Capability TL
process the rest of the message
exist.

bit: The forward unknown TLV bi
Section 3 of [RFC5561].



The FEC type for the P2MP PW Upstream FEC Element is encoded as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|P2MP PW Up=0x82|C|             PW Type         | PW Info Length|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    AGI Type   | AGI Length    |          AGI Value            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                    AGI Value (contd.)                         ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   AII Type    | SAII Length   |          SAII Value           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                    SAII Value (contd.)                        ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|PMSI Tunnel Typ|PMSI TT Length |                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
+
~                Transport LSP ID
|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|
|                    Optional Parameters
~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

Figure 2: P2MP PW Upstream FEC Element

* P2MP PW Up:

  8-bit representation for the P2MP PW Upstream FEC type.

* C bit:

  A value of 1 or 0 indicating whether a control word is prese
  absent for the P2MP PW.

Relay Source Port Option for DHCPv6

ay Source Port Option" is a new DHCPv6 option. It MUST be
either 1) a DHCPv6 relay agent that uses a non-DHCP UDP port
) communicating with the IPv6 server and the upstream relay
2) an IPv6 relay agent that detects the use of a non-DHCP
(not 547) by a downstream relay agent.

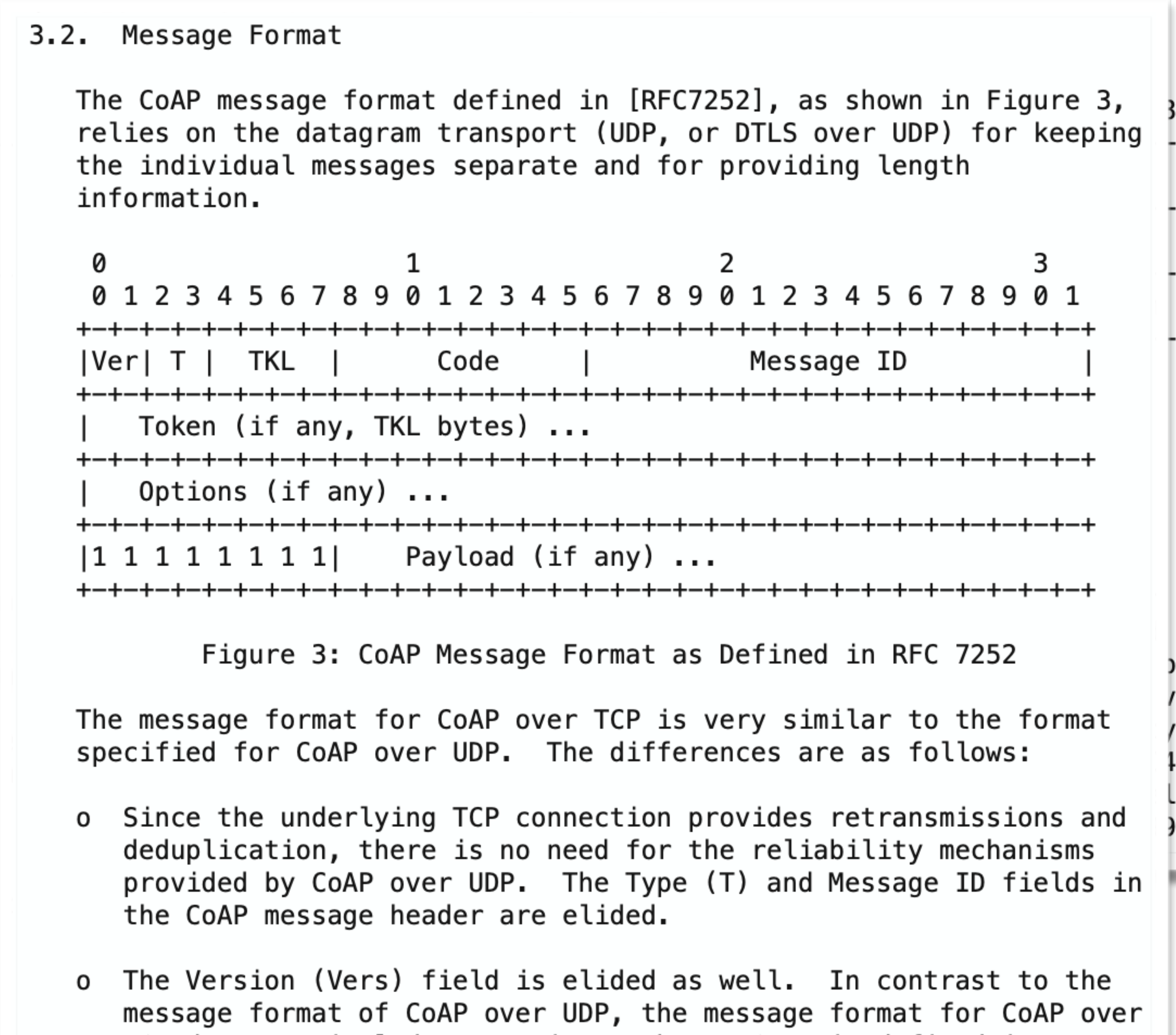
at of the "Relay Source Port Option" is shown below:

```
                  1                   2                   3
                        8 9 0 1 2 3 4 5 6 7 8 9 0 1
                        -+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

4.1. Availability SCSI-TLV

ined in [RFC8258]. This document d
CSI-TLV called the Availability SCS
an be included one or more times.

age is defined for both ICMPv4 and
ICMP Extended Echo Request



3.2. Message Format

The CoAP message format defined in [RFC7252], as shown in Figure 3, relies on the datagram transport (UDP, or DTLS over UDP) for keeping the individual messages separate and for providing length information.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Ver| T |  TKL  |      Code     |          Message ID           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Token (if any, TKL bytes) ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Options (if any) ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|1 1 1 1 1 1 1 1|    Payload (if any) ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3: CoAP Message Format as Defined in RFC 7252

The message format for CoAP over TCP is very similar to the format specified for CoAP over UDP. The differences are as follows:

o Since the underlying TCP connection provides retransmissions and deduplication, there is no need for the reliability mechanisms provided by CoAP over UDP. The Type (T) and Message ID fields in the CoAP message header are elided.

o The Version (Vers) field is elided as well. In contrast to the message format of CoAP over UDP, the message format for CoAP over

ere:

bit: The unknown TLV bit MUST b
recognize the MRT Capability TL
process the rest of the message
exist.

bit: The forward unknown TLV bi
Section 3 of [RFC5561].

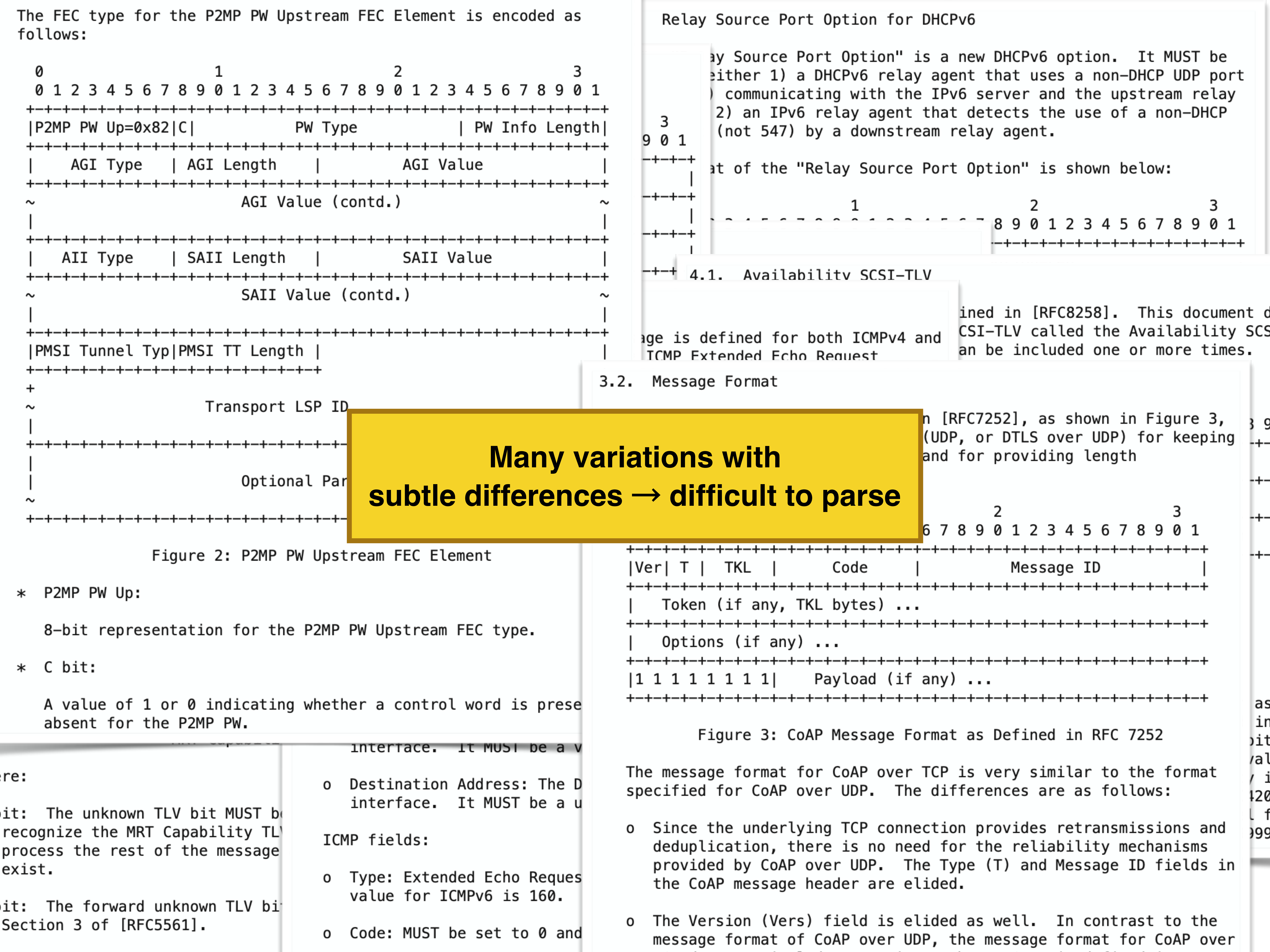
interface. It MUST be a v

o Destination Address: The D
  interface. It MUST be a u

ICMP fields:

o Type: Extended Echo Reques
  value for ICMPv6 is 160.

o Code: MUST be set to 0 and



The FEC type for the P2MP PW Upstream FEC Element is encoded as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|P2MP PW Up=0x82|C|             PW Type         | PW Info Length|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    AGI Type   | AGI Length    |          AGI Value            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                       AGI Value (contd.)                      ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   AII Type    | SAII Length   |          SAII Value           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                       SAII Value (contd.)                     ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|PMSI Tunnel Typ|PMSI TT Length |                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
+
~                   Transport LSP ID
|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|
|                       Optional Par
~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

Figure 2: P2MP PW Upstream FEC Element

* P2MP PW Up:

  8-bit representation for the P2MP PW Upstream FEC type.

* C bit:

  A value of 1 or 0 indicating whether a control word is prese
  absent for the P2MP PW.

ere:

it: The unknown TLV bit MUST b
recognize the MRT Capability TL
process the rest of the message
exist.

it: The forward unknown TLV bi
Section 3 of [RFC5561].

interface. It MUST be a v

o Destination Address: The D
   interface. It MUST be a u

ICMP fields:

o Type: Extended Echo Reques
   value for ICMPv6 is 160.

o Code: MUST be set to 0 and

Relay Source Port Option for DHCPv6

ay Source Port Option" is a new DHCPv6 option. It MUST be
either 1) a DHCPv6 relay agent that uses a non-DHCP UDP port
) communicating with the IPv6 server and the upstream relay
2) an IPv6 relay agent that detects the use of a non-DHCP
(not 547) by a downstream relay agent.

at of the "Relay Source Port Option" is shown below:

```
              1                   2                   3
               8 9 0 1 2 3 4 5 6 7 8 9 0 1
            -+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
 3
9 0 1
-+-+-+
    |
-+-+-+
    |
-+-+-+
    |
-+-+
```

4.1. Availability SCSI-TLV

ined in [RFC8258]. This document d
CSI-TLV called the Availability SCS
an be included one or more times.

age is defined for both ICMPv4 and
ICMP Extended Echo Request

3.2. Message Format

n [RFC7252], as shown in Figure 3,
(UDP, or DTLS over UDP) for keeping
and for providing length

```
                    2                   3
 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Ver| T |  TKL  |      Code     |          Message ID           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Token (if any, TKL bytes) ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Options (if any) ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|1 1 1 1 1 1 1 1|    Payload (if any) ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3: CoAP Message Format as Defined in RFC 7252

The message format for CoAP over TCP is very similar to the format specified for CoAP over UDP. The differences are as follows:

o Since the underlying TCP connection provides retransmissions and deduplication, there is no need for the reliability mechanisms provided by CoAP over UDP. The Type (T) and Message ID fields in the CoAP message header are elided.

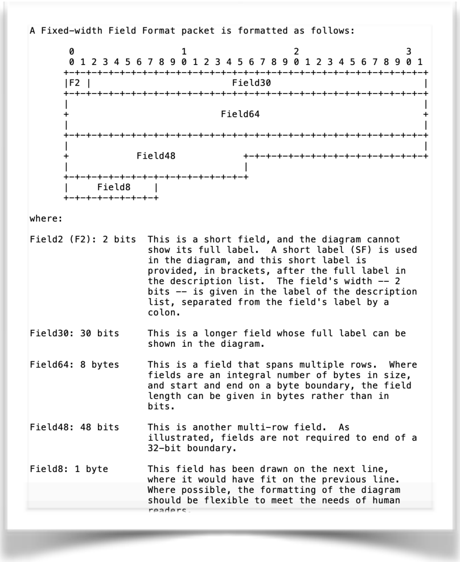
o The Version (Vers) field is elided as well. In contrast to the message format of CoAP over UDP, the message format for CoAP over

Many variations with subtle differences → difficult to parse

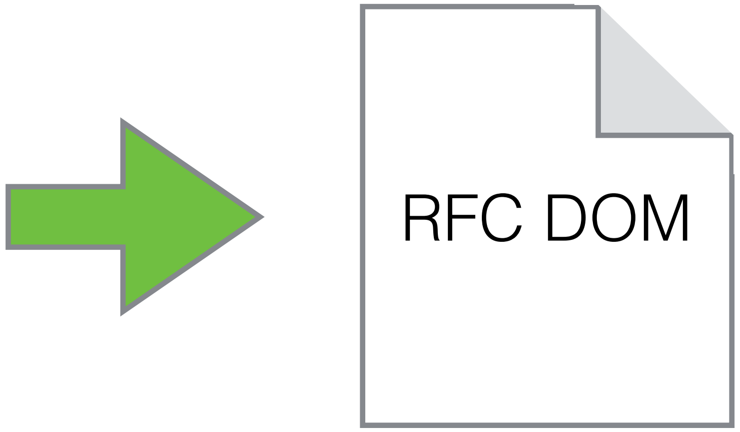
Augmented ASCII diagrams

- Much can be achieved just by being consistent
- Need other elements: constraints on field values, optional fields, links between PDUs, …
- Adheres to the design principles given earlier

Parsing protocol standards

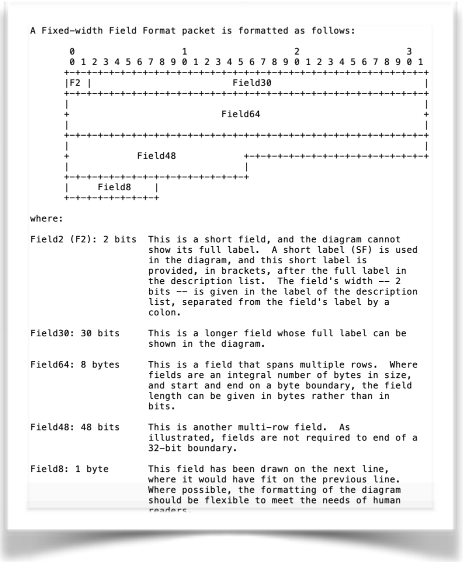


```
A Fixed-width Field Format packet is formatted as follows:

       0                   1                   2                   3
       0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |F2 |                        Field30                            |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                                                               |
      +                          Field64                              +
      |                                                               |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                                                               |
      +            Field48            +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                               |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |     Field8    |
      +-+-+-+-+-+-+-+-+

where:

Field2 (F2): 2 bits  This is a short field, and the diagram cannot
                     show its full label.  A short label (SF) is used
                     in the diagram, and this short label is
                     provided, in brackets, after the full label in
                     the description list.  The field's width -- 2
                     bits -- is given in the label of the description
                     list, separated from the field's label by a
                     colon.

Field30: 30 bits     This is a longer field whose full label can be
                     shown in the diagram.

Field64: 8 bytes     This is a field that spans multiple rows.  Where
                     fields are an integral number of bytes in size,
                     and start and end on a byte boundary, the field
                     length can be given in bytes rather than in
                     bits.

Field48: 48 bits     This is another multi-row field.  As
                     illustrated, fields are not required to end of a
                     32-bit boundary.

Field8: 1 byte       This field has been drawn on the next line,
                     where it would have fit on the previous line.
                     Where possible, the formatting of the diagram
                     should be flexible to meet the needs of human
                     readers.
```

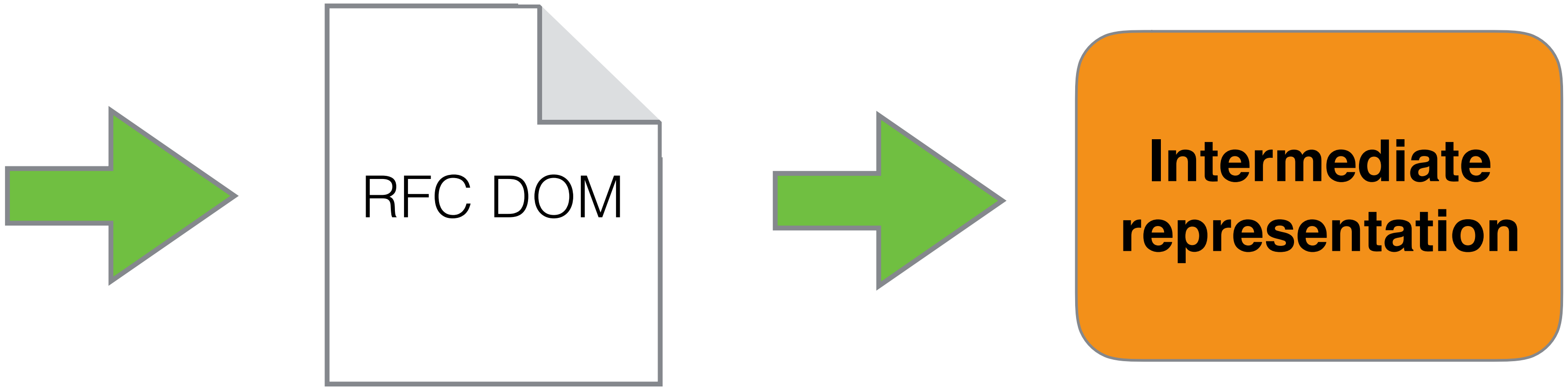


- Parse input into an RFC document object model
- RFC DOM is already well specified
- Allows for different input formats

Parsing protocol standards

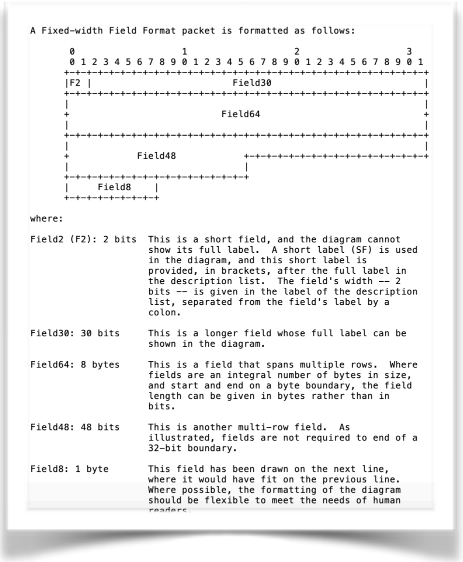


```
A Fixed-width Field Format packet is formatted as follows:

   0                   1                   2                   3
   0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |F2 |                          Field30                          |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |                                                               |
  +                            Field64                            +
  |                                                               |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |                                                               |
  +             Field48           +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |                               |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |     Field8    |
  +-+-+-+-+-+-+-+-+

where:

Field2 (F2): 2 bits  This is a short field, and the diagram cannot
                     show its full label.  A short label (SF) is used
                     in the diagram, and this short label is
                     provided, in brackets, after the full label in
                     the description list.  The field's width -- 2
                     bits -- is given in the label of the description
                     list, separated from the field's label by a
                     colon.

Field30: 30 bits     This is a longer field whose full label can be
                     shown in the diagram.

Field64: 8 bytes     This is a field that spans multiple rows.  Where
                     fields are an integral number of bytes in size,
                     and start and end on a byte boundary, the field
                     length can be given in bytes rather than in
                     bits.

Field48: 48 bits     This is another multi-row field.  As
                     illustrated, fields are not required to end of a
                     32-bit boundary.

Field8: 1 byte       This field has been drawn on the next line,
                     where it would have fit on the previous line.
                     Where possible, the formatting of the diagram
                     should be flexible to meet the needs of human
                     readers.
```

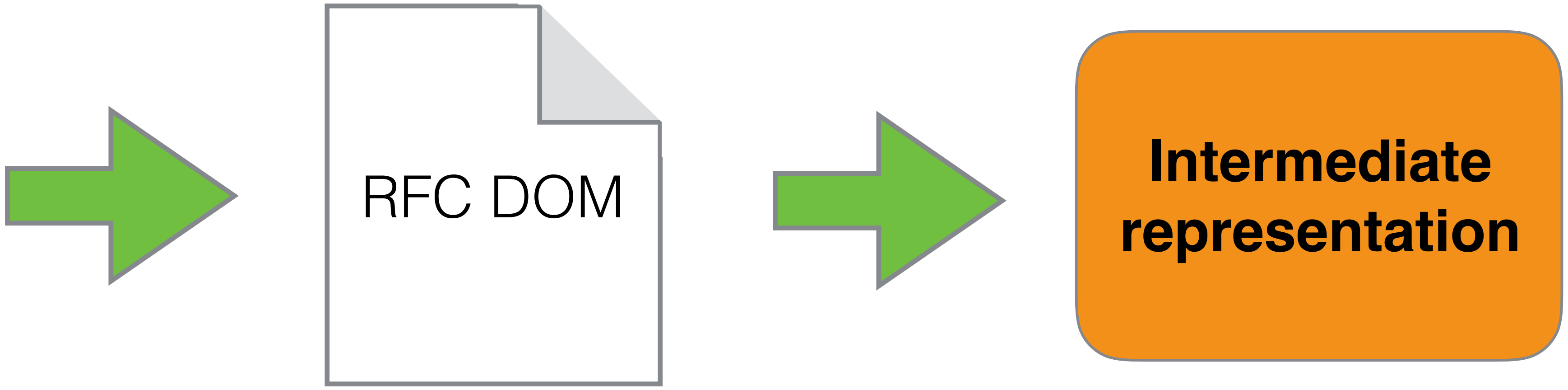


- Extract a protocol definition from the RFC DOM, and capture it in an intermediate representation
- Captures the syntax of the protocol and how to parse it
- Allows for different input languages, whose expressivity might vary

Parsing protocol standards



```
A Fixed-width Field Format packet is formatted as follows:

    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |F2 |                         Field30                           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   +                            Field64                            +
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   +              Field48          +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |    Field8     |
   +-+-+-+-+-+-+-+-+

where:

Field2 (F2): 2 bits  This is a short field, and the diagram cannot
                     show its full label.  A short label (SF) is used
                     in the diagram, and this short label is
                     provided, in brackets, after the full label in
                     the description list.  The field's width -- 2
                     bits -- is given in the label of the description
                     list, separated from the field's label by a
                     colon.

Field30: 30 bits     This is a longer field whose full label can be
                     shown in the diagram.

Field64: 8 bytes     This is a field that spans multiple rows.  Where
                     fields are an integral number of bytes in size,
                     and start and end on a byte boundary, the field
                     length can be given in bytes rather than in
                     bits.

Field48: 48 bits     This is another multi-row field.  As
                     illustrated, fields are not required to end of a
                     32-bit boundary.

Field8: 1 byte       This field has been drawn on the next line,
                     where it would have fit on the previous line.
                     Where possible, the formatting of the diagram
                     should be flexible to meet the needs of human
                     readers.
```



RFC DOM

Intermediate representation

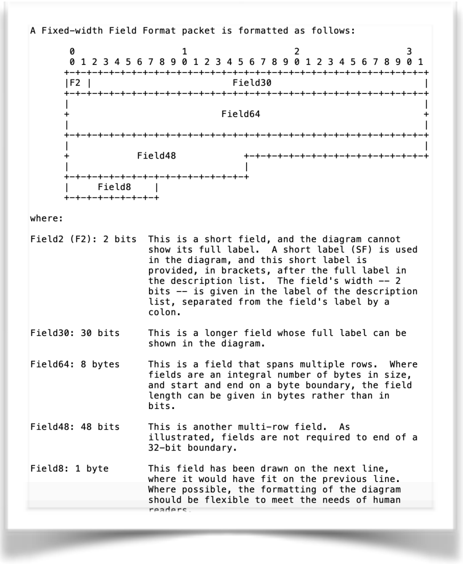
- Intermediate representation captures all of metadata required to parse the protocol

- The layout of each PDU

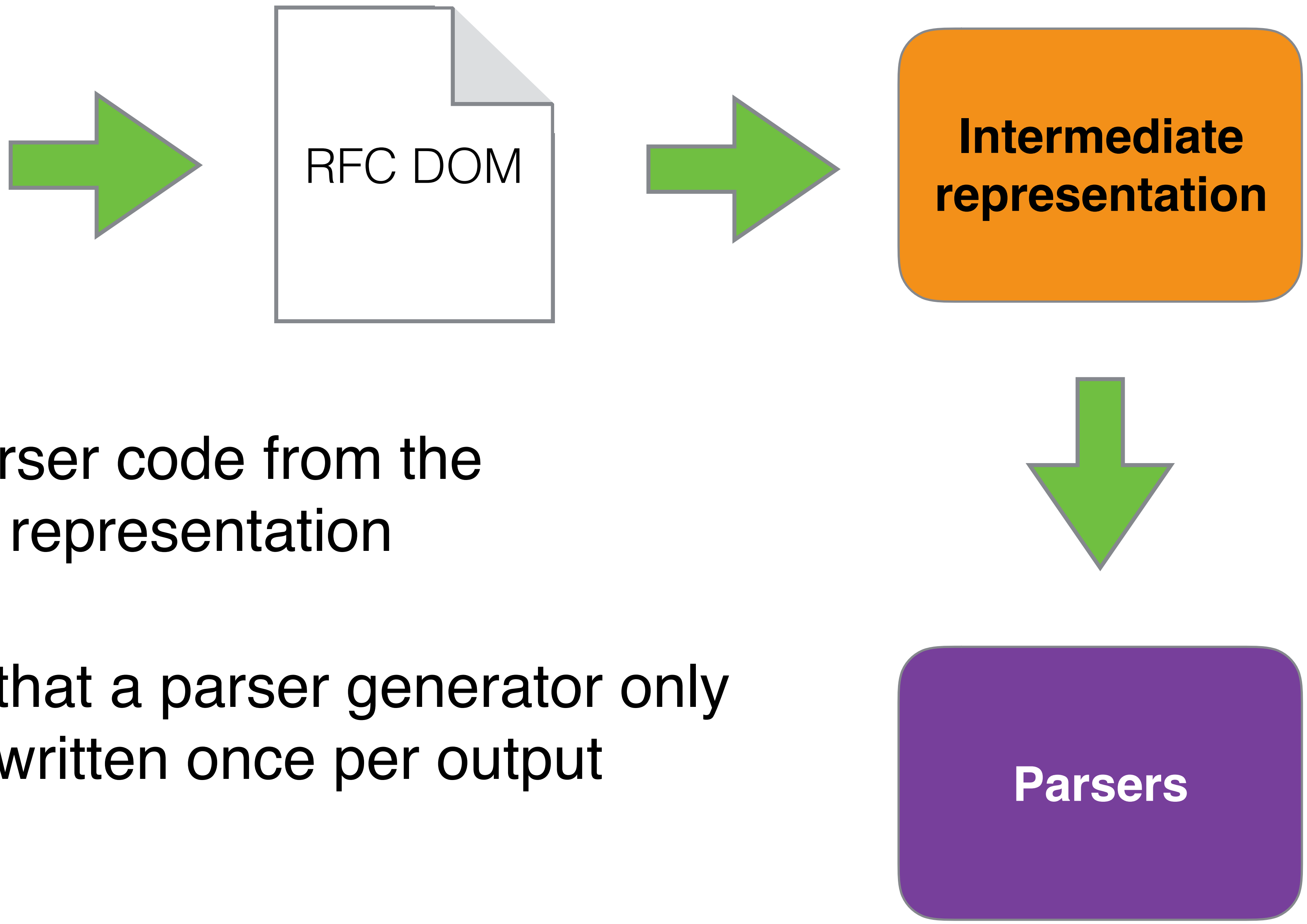
- Parsing context for out-of-band data

- Helper methods for encrypted fields

Parsing protocol standards



```
A Fixed-width Field Format packet is formatted as follows:

       0                   1                   2                   3
       0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |F2 |                         Field30                           |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                                                               |
      +                           Field64                             +
      |                                                               |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                                                               |
      +            Field48            +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                               |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |     Field8    |
      +-+-+-+-+-+-+-+-+

where:

Field2 (F2): 2 bits  This is a short field, and the diagram cannot
                     show its full label.  A short label (SF) is used
                     in the diagram, and this short label is
                     provided, in brackets, after the full label in
                     the description list.  The field's width -- 2
                     bits -- is given in the label of the description
                     list, separated from the field's label by a
                     colon.

Field30: 30 bits     This is a longer field whose full label can be
                     shown in the diagram.

Field64: 8 bytes     This is a field that spans multiple rows.  Where
                     fields are an integral number of bytes in size,
                     and start and end on a byte boundary, the field
                     length can be given in bytes rather than in
                     bits.

Field48: 48 bits     This is another multi-row field.  As
                     illustrated, fields are not required to end of a
                     32-bit boundary.

Field8: 1 byte       This field has been drawn on the next line,
                     where it would have fit on the previous line.
                     Where possible, the formatting of the diagram
                     should be flexible to meet the needs of human
                     readers.
```



RFC DOM

Intermediate representation

Parsers

- Generate parser code from the intermediate representation
- Split means that a parser generator only needs to be written once per output language

Summary

- IETF standardisation process can create ambiguous standards: want to introduce tooling without harming the parts of the process that work well
- ASCII diagrams already capture much of a protocol's syntax
- Augmenting ASCII diagrams and using them consistently allows tooling to extract protocol syntax
- Capturing protocol parsing in a common intermediary format allows for flexibility
- Automated parser generation from the intermediary format enables test-driven development → better standards