ETSI TS 102 599 V1.1.1 (2007-09)

Technical Specification

Methods for Testing and Specification (MTS); Internet Protocol Testing (IPT): IPv4 to IPV6 Transitioning; Requirements Catalogue

Reference

DTS/MTS-IPT-018-IPv6-TrsReqCat

Keywords

IP, IPv6, testing

ETSI

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

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

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

Important notice

Individual copies of the present document can be downloaded from:
http://www.etsi.org

The present document may be made available in more than one electronic version or in print. In any case of existing or perceived difference in contents between such versions, the reference version is the Portable Document Format (PDF). In case of dispute, the reference shall be the printing on ETSI printers of the PDF version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at
http://portal.etsi.org/tb/status/status.asp

If you find errors in the present document, please send your comment to one of the following services:
http://portal.etsi.org/chaircor/ETSI_support.asp

Copyright Notification

Contents

Intellectual Property Rights

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

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

Foreword

This Technical Specification (TS) has been produced by ETSI Technical Committee Methods for Testing and Specification (MTS).

1 Scope

The present document is a catalogue of all of the IPv4 to IPv6 transitioning-related requirements extracted from the following IETF specifications:

RFC 2529 [1]: "Transmission of IPv6 over IPv4 Domains without Explicit Tunnels".

RFC 2765 [2]: "Stateless IP/ICMP Translation Algorithm (SIIT)".

RFC 2766 [3]: "Network Address Translation - Protocol Translation (NAT-PT)".

RFC 2893 [4]: "Transition Mechanisms for IPv6 Hosts and Routers".

RFC 3056 [5]: "Connection of IPv6 Domains via IPv4 Clouds".

RFC 3596 [6]: "DNS Extensions to Support IP Version 6".

RFC 4213 [7]: "Basic Transition Mechanisms for IPv6 Hosts and Routers".

RFC 4214 [8]: "Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)".

2 References

References are either specific (identified by date of publication and/or edition number or version number) or non-specific.

- For a specific reference, subsequent revisions do not apply.
- Non-specific reference may be made only to a complete document or a part thereof and only in the following cases:
  - if it is accepted that it will be possible to use all future changes of the referenced document for the purposes of the referring document;
  - for informative references.

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

For online referenced documents, information sufficient to identify and locate the source shall be provided. Preferably, the primary source of the referenced document should be cited, in order to ensure traceability. Furthermore, the reference should, as far as possible, remain valid for the expected life of the document. The reference shall include the method of access to the referenced document and the full network address, with the same punctuation and use of upper case and lower case letters.

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

2.1 Normative references

The following referenced documents are indispensable for the application of the present document. For dated references, only the edition cited applies. For non-specific references, the latest edition of the referenced document (including any amendments) applies.

[1] IETF RFC 2529: "Transmission of IPv6 over IPv4 Domains without Explicit Tunnels".

[2] IETF RFC 2765: "Stateless IP/ICMP Translation Algorithm (SIIT)".

[3] IETF RFC 2766: "Network Address Translation - Protocol Translation (NAT-PT)".

[4] IETF RFC 2893: "Transition Mechanisms for IPv6 Hosts and Routers".

[5] IETF RFC 3056: "Connection of IPv6 Domains via IPv4 Clouds".

[6] IETF RFC 3596: "DNS Extensions to Support IP Version 6".

[7] IETF RFC 4213: "Basic Transition Mechanisms for IPv6 Hosts and Routers".

[8] IETF RFC 4214: "Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)".

3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

DNS	Domain Name System
IANA	Internet Assigned Number Association
ICMP	Internet Control Message Protocol
IETF	Internet Engineering Task Force
IP	Internet Protocol
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
NAT	Network Address Translation
PMTU	Path Maximum Transmission Unit
RFC	Request For Comments (IETF terminology for a draft standard)
SIIT	Stateless IP/ICMP Translation algorithm
TCP	Transport Control Protocol
UDP	User Datagram Protocol

4 Requirements catalogue

The requirements related to the transitioning from the Internet Protocol version 4 (IPv4) to the Internet Protocol version 6 (IPv6) are specified in a number of IETF documents. These documents include the basic transition mechanisms for IPv6 hosts and routers [7] as well as requirements for transitioning without the use of explicit tunnels [1], the use of the Stateless IP/ICMP Translation algorithm (SIIT) [2], protocol translation as part of NAT [3], transitioning mechanisms for IPv6 Hosts and Routers [4], connecting IPv6 domains through IPv4 networks [5], extension to DNS to support IPv6 [6] and the use of Intra-Site Automatic Tunnel Addressing Protocol (ISATAP) [8]. The present document is a catalogue of all of the normative requirements from these security specifications. Each requirement is given a unique identifier (e.g. RQ_003_1234) and the following information is included with each:

- the clause number in the RFC from which the requirement has been extracted;
- the type of requirement (Mandatory, Optional or Recommended);
- the type of device to which the requirement applies (for example, Host or Router);
- the actual text from which the requirement was extracted.

Requirements extracted from RFC 2529

Identifier: RQ_003_1001
RFC Clause: 2
Type: Optional
Applies to: Node

Requirement:
The default MTU size for IPv6 packets on an IPv4 domain of 1480 octets may be varied by a Router Advertisement containing an MTU option which specifies a different MTU.

Specification Text:
The default MTU size for IPv6 packets on an IPv4 domain is 1480 octets. This size may be varied by a Router Advertisement [DISC] containing an MTU option which specifies a different MTU, or by manual configuration of each node.

Note that if by chance the IPv6 MTU size proves to be too large for some intermediate IPv4 subnet, IPv4 fragmentation will ensue. While undesirable, this is not disastrous. However, the IPv4 "do not fragment" bit MUST NOT be set in the encapsulating IPv4 header.

Identifier: RQ_003_1002
RFC Clause: 2
Type: Optional
Applies to: Node

Requirement:
The default MTU size for IPv6 packets on an IPv4 domain of 1480 octets may be varied by manual configuration of each node.

Specification Text:
The default MTU size for IPv6 packets on an IPv4 domain is 1480 octets. This size may be varied by a Router Advertisement [DISC] containing an MTU option which specifies a different MTU, **or by manual configuration of each node.**

Note that if by chance the IPv6 MTU size proves to be too large for some intermediate IPv4 subnet, IPv4 fragmentation will ensue. While undesirable, this is not disastrous. However, the IPv4 "do not fragment" bit MUST NOT be set in the encapsulating IPv4 header.

Identifier: RQ_003_1003
RFC Clause: 2
Type: Mandatory
Applies to: Node

Requirement:
The IPv4 "do not fragment" bit MUST NOT be set in the encapsulating IPv4 header.

Specification Text:
The default MTU size for IPv6 packets on an IPv4 domain is 1480 octets. This size may be varied by a Router Advertisement [DISC] containing an MTU option which specifies a different MTU, or by manual configuration of each node.

Note that if by chance the IPv6 MTU size proves to be too large for some intermediate IPv4 subnet, IPv4 fragmentation will ensue. While undesirable, this is not disastrous. **However, the IPv4 "do not fragment" bit MUST NOT be set in the encapsulating IPv4 header.**

Identifier: RQ_003_1004
RFC Clause: 3
Type: Mandatory
Applies to: Node

Requirement:
IPv6 packets SHALL BE transmitted in IPv4 packets with an IPv4 protocol type of 41,

Specification Text:
IPv6 packets are transmitted in IPv4 packets [RFC 791] with an IPv4 protocol type of 41, the same as has been assigned in [RFC 1933] for IPv6 packets that are tunneled inside of IPv4 frames. The IPv4 header contains the Destination and Source IPv4 addresses. The IPv4 packet body contains the IPv6 header followed immediately by the payload.

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |Version|  IHL  |Type of Service|          Total Length         |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |         Identification        |Flags|      Fragment Offset    |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |  Time to Live | Protocol 41   |         Header Checksum       |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                       Source Address                          |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                    Destination Address                        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                    Options                    |    Padding    |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |            IPv6 header and payload ...              /
   +-------+-------+-------+-------+-------+------+------+
```

Identifier: RQ_003_1005
RFC Clause: 3
Type: Recommendation
Applies to: Node

Requirement:
If there are IPv4 options in the IPv4 header, then padding SHOULD be added to the IPv4 header such that the IPv6 header starts on a boundary that is a 32- bit offset from the end of the datalink header.

Specification Text:
If there are IPv4 options, then padding SHOULD be added to the IPv4 header such that the IPv6 header starts on a boundary that is a 32- bit offset from the end of the datalink header.

Identifier: RQ_003_1006
RFC Clause: 3
Type: Recommendation
Applies to: Node

Requirement:
The Time to Live field of the IPv4 header SHOULD be set to a low value, to prevent such packets accidentally leaking from the IPv4 domain.

Specification Text:
The Time to Live field SHOULD be set to a low value, to prevent such packets accidentally leaking from the IPv4 domain. This MUST be a configurable parameter, with a recommended default of 8.

Identifier: RQ_003_1007
RFC Clause: 3
Type: Mandatory
Applies to: Node

Requirement:
The Time to Live field of the IPv4 header MUST be a configurable parameter.

Specification Text:
The Time to Live field SHOULD be set to a low value, to prevent such packets accidentally leaking from the IPv4 domain. This **MUST be a configurable parameter**, with a recommended default of 8.

Identifier: RQ_003_1008
RFC Clause: 3
Type: Recommendation
Applies to: Node

Requirement:
The Time to Live field of the IPv4 header has a recommended default of 8.

Specification Text:
The Time to Live field SHOULD be set to a low value, to prevent such packets accidentally leaking from the IPv4 domain. This MUST be a configurable parameter, **with a recommended default of 8.**

Identifier: RQ_003_1009
RFC Clause: 4
Type: Mandatory
Applies to: Router

Requirement:
The Interface Identifier of an IPv4 interface is the 32-bit IPv4 address of that interface, with the octets in the same order in which they would appear in the header of an IPv4 packet, padded at the left with zeros to a total of 64 bits.

Specification Text:
The Interface Identifier [AARCH] of an IPv4 interface is the 32-bit IPv4 address of that interface, with the octets in the same order in which they would appear in the header of an IPv4 packet, padded at the left with zeros to a total of 64 bits. Note that the "Universal/ Local" bit is zero, indicating that the Interface Identifer is not globally unique. When the host has more than one IPv4 address in use on the physical interface concerned, an administrative choice of one of these IPv4 addresses is made.

Identifier: RQ_003_1010
RFC Clause: 4
Type: Mandatory
Applies to: Router

Requirement:
The "Universal/ Local" bit in an Interface Identifier of an IPv4 interface is zero.

Specification Text:
The Interface Identifier [AARCH] of an IPv4 interface is the 32-bit IPv4 address of that interface, with the octets in the same order in which they would appear in the header of an IPv4 packet, padded at the left with zeros to a total of 64 bits. **Note that the "Universal/ Local" bit is zero, indicating that the Interface Identifer is not globally unique.** When the host has more than one IPv4 address in use on the physical interface concerned, an administrative choice of one of these IPv4 addresses is made.

Identifier: RQ_003_1011
RFC Clause: 4
Type: Mandatory
Applies to: Router

Requirement:
Unless a router is handling both native LAN and "6over4" on the same physical interface, an IPv6 address prefix used for stateless autoconfiguration of an IPv4 interface MUST have a length of 64 bits

Specification Text:
An IPv6 address prefix used for stateless autoconfiguration [CONF] of an IPv4 interface MUST have a length of 64 bits except for a special case mentioned in Section 7.

Identifier: RQ_003_1012
RFC Clause: 4
Type: Mandatory
Applies to: Router

Requirement:
The IPv6 Link-local address [AARCH] for an IPv4 virtual interface is formed by appending the Interface Identifier to the prefix FE80::/64.

Specification Text:
The IPv6 Link-local address [AARCH] for an IPv4 virtual interface is formed by appending the Interface Identifier, as defined above, to the prefix FE80::/64.

```
    +-------+-------+-------+-------+-------+-------+------+------+
    |  FE      80      00      00      00      00      00     00  |
    +-------+-------+-------+-------+-------+-------+------+------+
    |  00      00   |  00   |  00   |    IPv4 Address             |
    +-------+-------+-------+-------+-------+-------+------+------+
```

Identifier: RQ_003_1013
RFC Clause: 5
Type: Mandatory
Applies to: Node

Requirement:
The Source/Target Link-layer Address option MUST have the following form when the link layer is IPv4:

Bit	Field
1 - 8	Type
9 - 16	Length
17 - 32	(must be zero)
33 - 64	IPv4 Address

Specification Text:
The procedure for mapping IPv6 addresses into IPv4 virtual link-layer addresses is described in [DISC]. The Source/Target Link-layer Address option has the following form when the link layer is IPv4. Since the length field is in units of 8 bytes, the value below is 1.

```
    +-------+-------+-------+-------+-------+-------+-------+-------+
    | Type  |Length | MUST be zero  |         IPv4 Address          |
    +-------+-------+-------+-------+-------+-------+-------+-------+
```

Type:
1 for Source Link-layer address.
2 for Target Link-layer address.

Length:
1 (in units of 8 octets).

IPv4 Address:

The 32 bit IPv4 address, in network byte order. This is the address

the interface currently responds to, and may be different from the
Interface Identifier for stateless autoconfiguration.

Identifier: RQ_003_1014
RFC Clause: 5
Type: Mandatory
Applies to: Node

Requirement:
In the Source/Target Link-layer Address option, the Type field shall have one of the following values:

1 for Source Link-layer address.
2 for Target Link-layer address.

Specification Text:
The procedure for mapping IPv6 addresses into IPv4 virtual link-layer addresses is described in [DISC]. **The Source/Target Link-layer Address option** has the following form when the link layer is IPv4. Since the length field is in units of 8 bytes, the value below is 1.

```
 +-------+-------+-------+-------+-------+-------+-------+-------+
 | Type  |Length | MUST be zero  |          IPv4 Address         |
 +-------+-------+-------+-------+-------+-------+-------+-------+
```

Type:
1 for Source Link-layer address.
2 for Target Link-layer address.

Length:
1 (in units of 8 octets).

IPv4 Address:

The 32 bit IPv4 address, in network byte order. This is the address
the interface currently responds to, and may be different from the
Interface Identifier for stateless autoconfiguration.

Identifier: RQ_003_1015
RFC Clause: 5
Type: Mandatory
Applies to: Node

Requirement:
In the Source/Target Link-layer Address option, the Length field shall have the value 1.

Specification Text:
The procedure for mapping IPv6 addresses into IPv4 virtual link-layer addresses is described in [DISC]. **The Source/Target Link-layer Address option** has the following form when the link layer is IPv4. Since the length field is in units of 8 bytes, the value below is 1.

```
 +-------+-------+-------+-------+-------+-------+-------+-------+
 | Type  |Length | MUST be zero  |          IPv4 Address         |
 +-------+-------+-------+-------+-------+-------+-------+-------+
```

Type:
1 for Source Link-layer address.
2 for Target Link-layer address.

Length:
1 (in units of 8 octets).

IPv4 Address:

The 32 bit IPv4 address, in network byte order. This is the address
the interface currently responds to, and may be different from the
Interface Identifier for stateless autoconfiguration.

Identifier: RQ_003_1016
RFC Clause: 5
Type: Mandatory
Applies to: Node

Requirement:
In the Source/Target Link-layer Address option, the IPv4 Address field shall be the 32 bit IPv4 address, in network byte order.

Specification Text:
The procedure for mapping IPv6 addresses into IPv4 virtual link-layer addresses is described in [DISC]. **The Source/Target Link-layer Address option** has the following form when the link layer is IPv4. Since the length field is in units of 8 bytes, the value below is 1.

```
    +-------+-------+-------+-------+-------+-------+-------+-------+
    | Type  |Length | MUST be zero  |         IPv4 Address          |
    +-------+-------+-------+-------+-------+-------+-------+-------+
```

Type:
 1 for Source Link-layer address.
 2 for Target Link-layer address.

Length:
 1 (in units of 8 octets).

IPv4 Address:

The 32 bit IPv4 address, in network byte order. This is the address the interface currently responds to, and may be different from the Interface Identifier for stateless autoconfiguration.

Identifier: RQ_003_1017
RFC Clause: 6
Type: Mandatory
Applies to: Node

Requirement:
IPv4 multicast MUST be available.

Specification Text:
IPv4 multicast MUST be available. An IPv6 packet with a multicast destination address DST MUST be transmitted to the IPv4 multicast address of Organization-Local Scope using the mapping below. These IPv4 multicast addresses SHOULD be taken from the block 239.192.0.0/16, a sub-block of the Organization-Local Scope address block, or, if all of those are not available, from the expansion blocks defined in [ADMIN]. Note that when they are formed using the expansion blocks, they use only a /16 sized block.

```
        +-------+-------+-------+-------+
        |  239  |  OLS  | DST14 | DST15 |
        +-------+-------+-------+-------+

        DST14, DST15        last two bytes of IPv6 multicast address.

        OLS                 from the configured Organization-Local
                            Scope address block.  SHOULD be 192,
                            see [ADMIN] for details.
```

Identifier:	RQ_003_1018
RFC Clause:	6
Type:	Mandatory
Applies to:	Node

Requirement:

An IPv6 packet with a multicast destination address DST MUST be transmitted to the IPv4 multicast address of Organization-Local Scope using the following mapping:

Bit	Field
1 - 8	239
9 - 16	OLS
17 - 24	DST14
25 - 32	DST15

Specification Text:

IPv4 multicast MUST be available. An IPv6 packet with a multicast destination address DST MUST be transmitted to the IPv4 multicast address of Organization-Local Scope using the mapping below. These IPv4 multicast addresses SHOULD be taken from the block 239.192.0.0/16, a sub-block of the Organization-Local Scope address block, or, if all of those are not available, from the expansion blocks defined in [ADMIN]. Note that when they are formed using the expansion blocks, they use only a /16 sized block.

```
+-------+-------+-------+-------+
|  239  |  OLS  | DST14 | DST15 |
+-------+-------+-------+-------+
```

```
DST14, DST15        last two bytes of IPv6 multicast address.

OLS                 from the configured Organization-Local
                    Scope address block.  SHOULD be 192,
                    see [ADMIN] for details.
```

Identifier:	RQ_003_1019
RFC Clause:	6
Type:	Recommendation
Applies to:	Node

Requirement:

When an IPv6 packet with a multicast destination address DST is transmitted to the IPv4 multicast address of Organization-Local Scope, the IPv4 multicast addresses SHOULD be taken from the block 239.192.0.0/16 or, if all of those are not available, from the expansion blocks defined in RFC 2365.

Specification Text:

IPv4 multicast MUST be available. An IPv6 packet with a multicast destination address DST MUST be transmitted to the IPv4 multicast address of Organization-Local Scope using the mapping below. **These IPv4 multicast addresses SHOULD be taken from the block 239.192.0.0/16, a sub-block of the Organization-Local Scope address block, or, if all of those are not available, from the expansion blocks defined in [ADMIN].** Note that when they are formed using the expansion blocks, they use only a /16 sized block.

```
+-------+-------+-------+-------+
|  239  |  OLS  | DST14 | DST15 |
+-------+-------+-------+-------+
```

```
DST14, DST15        last two bytes of IPv6 multicast address.

OLS                 from the configured Organization-Local
                    Scope address block.  SHOULD be 192,
                    see [ADMIN] for details.
```

Identifier: RQ_003_1020
RFC Clause: 6
Type: Mandatory
Applies to: Node

Requirement:
When an IPv6 packet with a multicast destination address DST is transmitted to the IPv4 multicast address of Organization-Local Scope, the DST14 and DST15 Fields should contain the last two bytes of IPv6 multicast address.

Specification Text:
IPv4 multicast MUST be available. **An IPv6 packet with a multicast destination address DST MUST be transmitted to the IPv4 multicast address of Organization-Local Scope** using the mapping below. These IPv4 multicast addresses SHOULD be taken from the block 239.192.0.0/16, a sub-block of the Organization-Local Scope address block, or, if all of those are not available, from the expansion blocks defined in [ADMIN]. Note that when they are formed using the expansion blocks, they use only a /16 sized block.

```
+-------+-------+-------+-------+
|  239  |  OLS  | DST14 | DST15 |
+-------+-------+-------+-------+
```

DST14, DST15   **last two bytes of IPv6 multicast address.**

OLS   from the configured Organization-Local
Scope address block. SHOULD be 192,
see [ADMIN] for details.

Identifier: RQ_003_1021
RFC Clause: 6
Type: Recommendation
Applies to: Node

Requirement:
When an IPv6 packet with a multicast destination address DST is transmitted to the IPv4 multicast address of Organization-Local Scope, the OLS Field should contain 192.

Specification Text:
IPv4 multicast MUST be available. **An IPv6 packet with a multicast destination address DST MUST be transmitted to the IPv4 multicast address of Organization-Local Scope** using the mapping below. These IPv4 multicast addresses SHOULD be taken from the block 239.192.0.0/16, a sub-block of the Organization-Local Scope address block, or, if all of those are not available, from the expansion blocks defined in [ADMIN]. Note that when they are formed using the expansion blocks, they use only a /16 sized block.

```
+-------+-------+-------+-------+
|  239  |  OLS  | DST14 | DST15 |
+-------+-------+-------+-------+
```

DST14, DST15   last two bytes of IPv6 multicast address.

OLS   **from the configured Organization-Local
Scope address block. SHOULD be 192,
see [ADMIN] for details.**

Identifier: RQ_003_1027
RFC Clause: 6
Type: Mandatory
Applies to: Router

Requirement:
The following multicast groups MUST be joined to support Neighbor Discovery:

- all-nodes multicast address
- all-routers multicast address
- solicited-node multicast address

Specification Text:
No new IANA registration procedures are REQUIRED for the above. **See appendix A. for a list of all the multicast groups that MUST be joined to support Neighbor Discovery.**

APPENDIX A: IPv4 Multicast Addresses for Neighbor Discovery

The following IPv4 multicast groups are used to support Neighbor Discovery with this specification. The IPv4 addresses listed in this section were obtained by looking at the IPv6 multicast addresses that Neigbour Discovery uses, and deriving the resulting IPv4 "virtual link-layer" addresses that are generated from them using the algorithm given in Section 6.multicast groups that MUST be joined to support Neighbor Discovery.

all-nodes multicast address
- **the administratively-scoped IPv4 multicast address used to reach all nodes in the local IPv4 domain supporting this specification. 239.OLS.0.1**

all-routers multicast address
- **the administratively-scoped IPv4 multicast address to reach all routers in the local IPv4 domain supporting this specification. 239.OLS.0.2**

solicited-node multicast address
- **an administratively scoped multicast address that is computed as a function of the solicited target's address by taking the low-order 24 bits of the IPv4 address used to form the IPv6 address, and prepending the prefix FF02:0:0:0:0:1:FF00::/104 [AARCH]. This is then mapped to the IPv4 multicast address by the method described in this document. For example, if the IPv4 address used to form the IPv6 address is W.X.Y.Z, then the IPv6 solicited node multicast address is FF02::1:255.X.Y.Z and the corresponding IPv4 multicast address is 239.OLS.Y.Z**

Identifier: RQ_003_1022
RFC Clause: 7
Type: Mandatory
Applies to: Router

Requirement:
The "6over4" IPv6 prefix MUST be unique within its scope, whether site-local or global addressing is used.

Specification Text:
During transition, routers may need to advertise at least two IPv6 prefixes, one for the native LAN (e.g. Ethernet) and one for "6over4". **As with any IPv6 prefix assigned to an IPv6 subnet, the latter MUST be unique within its scope, whether site-local or global addressing is used.**

Identifier: RQ_003_1023
RFC Clause: 7
Type: Mandatory
Applies to: Router

Requirement:
When a router is handling both native LAN and "6over4" on the same physical interface, during stateless autoconfiguration, there is a period when IPv6 link-local addresses are used, in both cases the prefix-length for these link-local adddress MUST then be 128 so that the two cases can be distinguished.

Specification Text:
Also note that when a router is handling both native LAN and "6over4" on the same physical interface, during stateless autoconfiguration, there is a period when IPv6 link-local addresses are used, in both cases with the prefix FE80::/64. **Note that the prefix-length for these link-local adddress MUST then be 128 so that the two cases can be distinguished.**

Identifier: RQ_003_1024
RFC Clause: 9
Type: Mandatory
Applies to: Router

Requirement:
Boundary routers MUST discard multicast IPv4 packets with source or destination multicast addresses of organisation local scope (OLS), if they arrive on physical interfaces outside that scope.

Specification Text:
There is a possible spoofing attack in which spurious 6over4 packets are injected into a 6over4 domain from outside. **Thus, boundary routers MUST discard multicast IPv4 packets with source or destination multicast addresses of organisation local scope as defined in section 6 above, if they arrive on physical interfaces outside that scope.** To defend against spurious unicast 6over4 packets, boundary routers MUST discard incoming IPv4 packets with protocol type 41 from unknown sources, i.e. IPv6-in-IPv4 tunnels MUST only be accepted from trusted sources. Unless IPSEC authentication is available, the RECOMMENDED technique for this is to configure the boundary router only to accept protocol type 41 packets from source addresses within a trusted range or ranges.

Identifier: RQ_003_1025
RFC Clause: 9
Type: Mandatory
Applies to: Router

Requirement:
To defend against spurious unicast 6over4 packets, boundary routers MUST discard incoming IPv4 packets with protocol type 41 from unknown sources, i.e. IPv6-in-IPv4 tunnels MUST only be accepted from trusted sources.

Specification Text:
There is a possible spoofing attack in which spurious 6over4 packets are injected into a 6over4 domain from outside. Thus, boundary routers MUST discard multicast IPv4 packets with source or destination multicast addresses of organisation local scope as defined in section 6 above, if they arrive on physical interfaces outside that scope. **To defend against spurious unicast 6over4 packets, boundary routers MUST discard incoming IPv4 packets with protocol type 41 from unknown sources, i.e. IPv6-in-IPv4 tunnels MUST only be accepted from trusted sources.** Unless IPSEC authentication is available, the RECOMMENDED technique for this is to configure the boundary router only to accept protocol type 41 packets from source addresses within a trusted range or ranges.

Identifier: RQ_003_1026
RFC Clause: 9
Type: Recommendation
Applies to: Router

Requirement:
Unless IPSEC authentication is available, the RECOMMENDED technique for boundary routers to discard incoming IPv4 packets with protocol type 41 from unknown sources, is to configure the boundary router only to accept protocol type 41 packets from source addresses within a trusted range or ranges.

Specification Text:
There is a possible spoofing attack in which spurious 6over4 packets are injected into a 6over4 domain from outside. Thus, boundary routers MUST discard multicast IPv4 packets with source or destination multicast addresses of organisation local scope as defined in section 6 above, if they arrive on physical interfaces outside that scope. To defend against spurious unicast 6over4 packets, boundary routers MUST discard incoming IPv4 packets with protocol type 41 from unknown sources, i.e. IPv6-in-IPv4 tunnels MUST only be accepted from trusted sources. **Unless IPSEC authentication is available, the RECOMMENDED technique for this is to configure the boundary router only to accept protocol type 41 packets from source addresses within a trusted range or ranges.**

Requirements extracted from RFC 2765

Identifier: RQ_003_3001
RFC Clause: 1.2
Type: Mandatory
Applies to: Host

Requirement:
The IPv6 nodes using the translator MUST have an IPv4-translated IPv6 address while it is communicating with IPv4-only nodes.

Specification Text:
The IPv6 nodes using the translator MUST have an IPv4-translated IPv6 address while it is communicating with IPv4-only nodes.

Identifier: RQ_003_3002
RFC Clause: 1.2
Type: Mandatory
Applies to: Node

Requirement:
The address pool, used to generate IPv4-translated addresse, can not be assigned to subnets but MUST be separated from the IPv4 subnets used on the "inside" of the translator.

Specification Text:
The use of the algorithm assumes that there is an IPv4 address pool used to generate IPv4-translated addresses. Routing needs to be able to route any IPv4 packets, whether generated "outside" or "inside" the translator, destined to addresses in this pool towards the translator. **This implies that the address pool can not be assigned to subnets but MUST be separated from the IPv4 subnets used on the "inside" of the translator.**

Identifier:	RQ_003_3131
RFC Clause:	1.3
Type:	Recommendation
Applies to:	Host

Requirement:

In order to take advantage of translators a node should send an IPv6 packet where the destination address is the IPv4-mapped address and the source address is the node's temporarily assigned IPv4-translated address.

Specification Text:

The network layer in an IPv6-only node, when presented by the application with either an IPv4 destination address or an IPv4-mapped IPv6 destination address, is likely to drop the packet and return some error message to the application. **In order to take advantage of translators such a node should instead send an IPv6 packet where the destination address is the IPv4-mapped address and the source address is the node's temporarily assigned IPv4-translated address.** If the node does not have a temporarily assigned IPv4-translated address it should acquire one using mechanisms that are not discussed in this document.

Identifier:	RQ_003_3132
RFC Clause:	1.3
Type:	Recommendation
Applies to:	Host

Requirement:

If the node does not have a temporarily assigned IPv4-translated address it should acquire one using mechanisms that are not discussed in this document.

Specification Text:

The network layer in an IPv6-only node, when presented by the application with either an IPv4 destination address or an IPv4-mapped IPv6 destination address, is likely to drop the packet and return some error message to the application. In order to take advantage of translators such a node should instead send an IPv6 packet where the destination address is the IPv4-mapped address and the source address is the node's temporarily assigned IPv4-translated address. **If the node does not have a temporarily assigned IPv4-translated address it should acquire one using mechanisms that are not discussed in this document.**

Identifier:	RQ_003_3005
RFC Clause:	3
Type:	Mandatory
Applies to:	Router

Requirement:

When either IPv4 or IPv6 routers send back ICMP "packet too big" messages to the sender, an IPv6 fragment header SHALL only be included if the IPv4 packet is already fragmented.

Specification Text:

When the IPv4 node performs path MTU discovery (by setting the DF bit in the header) the path MTU discovery can operate end-to-end i.e. across the translator. In this case either IPv4 or IPv6 routers might send back ICMP "packet too big" messages to the sender. When these ICMP errors are sent by the IPv6 routers they will pass through a translator which will translate the ICMP error to a form that the IPv4 sender can understand. In this case an IPv6 fragment header is only included if the IPv4 packet is already fragmented.

Identifier: RQ_003_3003
RFC Clause: 3
Type: Mandatory
Applies to: Router

Requirement:
When the IPv4 sender does not perform path MTU discovery the translator SHALL fragment the IPv4 packet so that it fits in 1280 byte IPv6 packet.

Specification Text:
However, when the IPv4 sender does not perform path MTU discovery the translator has to ensure that the packet does not exceed the path MTU on the IPv6 side. This is done by fragmenting the IPv4 packet so that it fits in 1280 byte IPv6 packet since IPv6 guarantees that 1280 byte packets never need to be fragmented. Also, when the IPv4 sender does not perform path MTU discovery the translator MUST always include an IPv6 fragment header to indicate that the sender allows fragmentation. That is needed should the packet pass through an IPv6-to-IPv4 translator.

Identifier: RQ_003_3004
RFC Clause: 3
Type: Mandatory
Applies to: Router

Requirement:
When the IPv4 sender does not perform path MTU discovery the translator MUST always include an IPv6 fragment header to indicate that the sender allows fragmentation. That is needed should the packet pass through an IPv6-to-IPv4 translator.

Specification Text:
However, when the IPv4 sender does not perform path MTU discovery the translator has to ensure that the packet does not exceed the path MTU on the IPv6 side. This is done by fragmenting the IPv4 packet so that it fits in 1280 byte IPv6 packet since IPv6 guarantees that 1280 byte packets never need to be fragmented. **Also, when the IPv4 sender does not perform path MTU discovery the translator MUST always include an IPv6 fragment header to indicate that the sender allows fragmentation. That is needed should the packet pass through an IPv6-to-IPv4 translator.**

Identifier: RQ_003_3006
RFC Clause: 3.1
Type: Mandatory
Applies to: Router

Requirement:
If the DF flag is not set and the IPv4 packet will result in an IPv6 packet larger than 1280 bytes the IPv4 packet MUST be fragmented prior to translating it.

Specification Text:
If the DF flag is not set and the IPv4 packet will result in an IPv6 packet larger than 1280 bytes the IPv4 packet MUST be fragmented prior to translating it. Since IPv4 packets with DF not set will always result in a fragment header being added to the packet the IPv4 packets MUST be fragmented so that their length, excluding the IPv4 header, is at most 1232 bytes (1280 minus 40 for the IPv6 header and 8 for the Fragment header). The resulting fragments are then translated independently using the logic described below.

Identifier:	RQ_003_3007
RFC Clause:	3.1
Type:	Mandatory
Applies to:	Router

Requirement:

Since IPv4 packets with DF not set MUST be fragmented so that their length, excluding the IPv4 header, is at most 1232 bytes.

Specification Text:

If the DF flag is not set and the IPv4 packet will result in an IPv6 packet larger than 1280 bytes the IPv4 packet MUST be fragmented prior to translating it. **Since IPv4 packets with DF not set will always result in a fragment header being added to the packet the IPv4 packets MUST be fragmented so that their length, excluding the IPv4 header, is at most 1232 bytes (1280 minus 40 for the IPv6 header and 8 for the Fragment header).** The resulting fragments are then translated independently using the logic described below.

Identifier:	RQ_003_3008
RFC Clause:	3.1
Type:	Mandatory
Applies to:	Router

Requirement:

When translating the IPv4 packet to IPV6, the resulting IPv6 header's version field SHALL be set to 6.

Specification Text:

If the DF bit is set and the packet is not a fragment (i.e., the MF flag is not set and the Fragment Offset is zero) then there is no need to add a fragment header to the packet. **The IPv6 header fields are set as follows:**

```
   Version:
           6

   Traffic Class:
           By default, copied from IP Type Of Service and
           Precedence field (all 8 bits are copied).  According
           to [DIFFSERV] the semantics of the bits are identical
           in IPv4 and IPv6.  However, in some IPv4 environments
           these fields might be used with the old semantics of
           "Type Of Service and Precedence".  An implementation
           of a translator SHOULD provide the ability to ignore
           the IPv4 "TOS" and always set the IPv6 traffic class
           to zero.

   Flow Label:
           0 (all zero bits)

   Payload Length:
           Total length value from IPv4 header, minus the size
           of the IPv4 header and IPv4 options, if present.
   Next Header:
           Protocol field copied from IPv4 header

   Hop Limit:
           TTL value copied from IPv4 header.  Since the
           translator is a router, as part of forwarding the
           packet it needs to decrement either the IPv4 TTL
           (before the translation) or the IPv6 Hop Limit (after
           the translation).  As part of decrementing the TTL or
           Hop Limit the translator (as any router) needs to
           check for zero and send the ICMPv4 or ICMPv6 "ttl
           exceeded" error.

   Source Address:
           The low-order 32 bits is the IPv4 source address.
           The high-order 96 bits is the IPv4-mapped prefix
           (::ffff:0:0/96)
   Destination Address:
           The low-order 32 bits is the IPv4 destination
           address.  The high-order 96 bits is the IPv4-
           translated prefix (0::ffff:0:0:0/96)
```

Identifier: RQ_003_3009
RFC Clause: 3.1
Type: Mandatory
Applies to: Router

Requirement:

When translating the IPv4 packet to IPV6, the resulting IPv6 header's Traffic Class Field SHALL by default, be copied from IP Type Of Service and Precedence field (all 8 bits are copied).

Specification Text:

If the DF bit is set and the packet is not a fragment (i.e., the MF flag is not set and the Fragment Offset is zero) then there is no need to add a fragment header to the packet. **The IPv6 header fields are set as follows:**

```
   Version:
           6

   Traffic Class:
           By default, copied from IP Type Of Service and
           Precedence field (all 8 bits are copied).  According
           to [DIFFSERV] the semantics of the bits are identical
           in IPv4 and IPv6.  However, in some IPv4 environments
           these fields might be used with the old semantics of
           "Type Of Service and Precedence".  An implementation
           of a translator SHOULD provide the ability to ignore
           the IPv4 "TOS" and always set the IPv6 traffic class
           to zero.

   Flow Label:
           0 (all zero bits)

   Payload Length:
           Total length value from IPv4 header, minus the size
           of the IPv4 header and IPv4 options, if present.
   Next Header:
           Protocol field copied from IPv4 header

   Hop Limit:
           TTL value copied from IPv4 header.  Since the
           translator is a router, as part of forwarding the
           packet it needs to decrement either the IPv4 TTL
           (before the translation) or the IPv6 Hop Limit (after
           the translation).  As part of decrementing the TTL or
           Hop Limit the translator (as any router) needs to
           check for zero and send the ICMPv4 or ICMPv6 "ttl
           exceeded" error.

   Source Address:
           The low-order 32 bits is the IPv4 source address.
           The high-order 96 bits is the IPv4-mapped prefix
           (::ffff:0:0/96)

   Destination Address:
           The low-order 32 bits is the IPv4 destination
           address.  The high-order 96 bits is the IPv4-
           translated prefix (0::ffff:0:0:0/96)
```

Identifier: RQ_003_3010
RFC Clause: 3.1
Type: Recommendation
Applies to: Router

Requirement:

An implementation of a translator SHOULD provide the ability to ignore the IPv4 "TOS" and always set the IPv6 traffic class to zero.

Specification Text:

If the DF bit is set and the packet is not a fragment (i.e., the MF flag is not set and the Fragment Offset is zero) then there is no need to add a fragment header to the packet. The IPv6 header fields are set as follows:

Version:
6

Traffic Class:
By default, copied from IP Type Of Service and Precedence field (all 8 bits are copied). According to [DIFFSERV] the semantics of the bits are identical in IPv4 and IPv6. However, in some IPv4 environments these fields might be used with the old semantics of "Type Of Service and Precedence". **An implementation of a translator SHOULD provide the ability to ignore the IPv4 "TOS" and always set the IPv6 traffic class to zero.**

Flow Label:
0 (all zero bits)

Payload Length:
Total length value from IPv4 header, minus the size of the IPv4 header and IPv4 options, if present.

Next Header:
Protocol field copied from IPv4 header

Hop Limit:
TTL value copied from IPv4 header. Since the translator is a router, as part of forwarding the packet it needs to decrement either the IPv4 TTL (before the translation) or the IPv6 Hop Limit (after the translation). As part of decrementing the TTL or Hop Limit the translator (as any router) needs to check for zero and send the ICMPv4 or ICMPv6 "ttl exceeded" error.

Source Address:
The low-order 32 bits is the IPv4 source address. The high-order 96 bits is the IPv4-mapped prefix (::ffff:0:0/96)

Destination Address:
The low-order 32 bits is the IPv4 destination address. The high-order 96 bits is the IPv4-translated prefix (0::ffff:0:0:0/96)

Identifier: RQ_003_3011
RFC Clause: 3.1
Type: Mandatory
Applies to: Router

Requirement:

When translating the IPv4 packet to IPV6, the resulting IPv6 header's Flow Label field SHALL be set to 0 (all zero bits).

Specification Text:

If the DF bit is set and the packet is not a fragment (i.e., the MF flag is not set and the Fragment Offset is zero) then there is no need to add a fragment header to the packet. **The IPv6 header fields are set as follows:**

Version:

6

Traffic Class:

By default, copied from IP Type Of Service and Precedence field (all 8 bits are copied). According to [DIFFSERV] the semantics of the bits are identical in IPv4 and IPv6. However, in some IPv4 environments these fields might be used with the old semantics of "Type Of Service and Precedence". An implementation of a translator SHOULD provide the ability to ignore the IPv4 "TOS" and always set the IPv6 traffic class to zero.

Flow Label:

0 (all zero bits)

Payload Length:

Total length value from IPv4 header, minus the size of the IPv4 header and IPv4 options, if present.

Next Header:

Protocol field copied from IPv4 header

Hop Limit:

TTL value copied from IPv4 header. Since the translator is a router, as part of forwarding the packet it needs to decrement either the IPv4 TTL (before the translation) or the IPv6 Hop Limit (after the translation). As part of decrementing the TTL or Hop Limit the translator (as any router) needs to check for zero and send the ICMPv4 or ICMPv6 "ttl exceeded" error.

Source Address:

The low-order 32 bits is the IPv4 source address. The high-order 96 bits is the IPv4-mapped prefix (::ffff:0:0/96)

Destination Address:

The low-order 32 bits is the IPv4 destination address. The high-order 96 bits is the IPv4-translated prefix (0::ffff:0:0:0/96)

Identifier: RQ_003_3012
RFC Clause: 3.1
Type: Mandatory
Applies to: Router

Requirement:

If the DF bit is set and the packet is not a fragment, after translating the IPv4 packet to IPV6, the resulting IPv6 header's Payload Length Label field SHALL be set to the total length value from IPv4 header, minus the size of the IPv4 header and IPv4 options, if present.

Specification Text:

If the DF bit is set and the packet is not a fragment (i.e., the MF flag is not set and the Fragment Offset is zero) then there is no need to add a fragment header to the packet. The IPv6 header fields are set as follows:

Version:

6

Traffic Class:

By default, copied from IP Type Of Service and Precedence field (all 8 bits are copied). According to [DIFFSERV] the semantics of the bits are identical in IPv4 and IPv6. However, in some IPv4 environments these fields might be used with the old semantics of "Type Of Service and Precedence". An implementation of a translator SHOULD provide the ability to ignore the IPv4 "TOS" and always set the IPv6 traffic class to zero.

Flow Label:

0 (all zero bits)

Payload Length:
Total length value from IPv4 header, minus the size of the IPv4 header and IPv4 options, if present.

Next Header:
Protocol field copied from IPv4 header

Hop Limit:
TTL value copied from IPv4 header. Since the translator is a router, as part of forwarding the packet it needs to decrement either the IPv4 TTL (before the translation) or the IPv6 Hop Limit (after the translation). As part of decrementing the TTL or Hop Limit the translator (as any router) needs to check for zero and send the ICMPv4 or ICMPv6 "ttl exceeded" error.

Source Address:
The low-order 32 bits is the IPv4 source address. The high-order 96 bits is the IPv4-mapped prefix (::ffff:0:0/96)

Destination Address:
The low-order 32 bits is the IPv4 destination address. The high-order 96 bits is the IPv4-translated prefix (0::ffff:0:0:0/96)

Identifier: RQ_003_3013
RFC Clause: 3.1
Type: Mandatory
Applies to: Router

Requirement:
If the DF bit is set and the packet is not a fragment, after translating the IPv4 packet to IPV6, the resulting IPv6 header's Next Header field SHALL be set to the Protocol field copied from IPv4 header.

Specification Text:
If the DF bit is set and the packet is not a fragment (i.e., the MF flag is not set and the Fragment Offset is zero) then there is no need to add a fragment header to the packet. The IPv6 header fields are set as follows:

Version:
6

Traffic Class:
By default, copied from IP Type Of Service and Precedence field (all 8 bits are copied). According to [DIFFSERV] the semantics of the bits are identical in IPv4 and IPv6. However, in some IPv4 environments these fields might be used with the old semantics of "Type Of Service and Precedence". An implementation of a translator SHOULD provide the ability to ignore the IPv4 "TOS" and always set the IPv6 traffic class to zero.

Flow Label:
0 (all zero bits)

Payload Length:
Total length value from IPv4 header, minus the size of the IPv4 header and IPv4 options, if present.
Next Header:
Protocol field copied from IPv4 header

Hop Limit:
TTL value copied from IPv4 header. Since the translator is a router, as part of forwarding the packet it needs to decrement either the IPv4 TTL (before the translation) or the IPv6 Hop Limit (after the translation). As part of decrementing the TTL or Hop Limit the translator (as any router) needs to check for zero and send the ICMPv4 or ICMPv6 "ttl exceeded" error.

Source Address:

The low-order 32 bits is the IPv4 source address. The high-order 96 bits is the IPv4-mapped prefix (::ffff:0:0/96)

Destination Address:

The low-order 32 bits is the IPv4 destination address. The high-order 96 bits is the IPv4-translated prefix (0::ffff:0:0:0/96)

Identifier: RQ_003_3014
RFC Clause: 3.1
Type: Mandatory
Applies to: Router

Requirement:

When translating the IPv4 packet to IPV6, the resulting IPv6 header's Hop Limit field SHALL be set to the TTL value copied from IPv4 header.

Specification Text:

If the DF bit is set and the packet is not a fragment (i.e., the MF flag is not set and the Fragment Offset is zero) then there is no need to add a fragment header to the packet. **The IPv6 header fields are set as follows:**

Version:

6

Traffic Class:

By default, copied from IP Type Of Service and Precedence field (all 8 bits are copied). According to [DIFFSERV] the semantics of the bits are identical in IPv4 and IPv6. However, in some IPv4 environments these fields might be used with the old semantics of "Type Of Service and Precedence". An implementation of a translator SHOULD provide the ability to ignore the IPv4 "TOS" and always set the IPv6 traffic class to zero.

Flow Label:

0 (all zero bits)

Payload Length:

Total length value from IPv4 header, minus the size of the IPv4 header and IPv4 options, if present.

Next Header:

Protocol field copied from IPv4 header

Hop Limit:

TTL value copied from IPv4 header. Since the translator is a router, as part of forwarding the packet it needs to decrement either the IPv4 TTL (before the translation) or the IPv6 Hop Limit (after the translation). As part of decrementing the TTL or Hop Limit the translator (as any router) needs to check for zero and send the ICMPv4 or ICMPv6 "ttl exceeded" error.

Source Address:

The low-order 32 bits is the IPv4 source address. The high-order 96 bits is the IPv4-mapped prefix (::ffff:0:0/96)

Destination Address:

The low-order 32 bits is the IPv4 destination address. The high-order 96 bits is the IPv4-translated prefix (0::ffff:0:0:0/96)

Identifier: RQ_003_3015
RFC Clause: 3.1
Type: Mandatory
Applies to: Router

Requirement:
As part of forwarding the packet if the translator has not decremented the IPv4 TTL (before the translation) it SHALL decrement the IPv6 Hop Limit (after the translation).

Specification Text:
If the DF bit is set and the packet is not a fragment (i.e., the MF flag is not set and the Fragment Offset is zero) then there is no need to add a fragment header to the packet**. The IPv6 header fields are set as follows:**

Version:
6

Traffic Class:
By default, copied from IP Type Of Service and Precedence field (all 8 bits are copied). According to [DIFFSERV] the semantics of the bits are identical in IPv4 and IPv6. However, in some IPv4 environments these fields might be used with the old semantics of "Type Of Service and Precedence". An implementation of a translator SHOULD provide the ability to ignore the IPv4 "TOS" and always set the IPv6 traffic class to zero.

Flow Label:
0 (all zero bits)

Payload Length:
Total length value from IPv4 header, minus the size of the IPv4 header and IPv4 options, if present.

Next Header:
Protocol field copied from IPv4 header

Hop Limit:
TTL value copied from IPv4 header**. Since the translator is a router, as part of forwarding the packet it needs to decrement either the IPv4 TTL (before the translation) or the IPv6 Hop Limit (after the translation).** As part of decrementing the TTL or Hop Limit the translator (as any router) needs to check for zero and send the ICMPv4 or ICMPv6 "ttl exceeded" error.

Source Address:
The low-order 32 bits is the IPv4 source address. The high-order 96 bits is the IPv4-mapped prefix (::ffff:0:0/96)

Destination Address:
The low-order 32 bits is the IPv4 destination address. The high-order 96 bits is the IPv4-translated prefix (0::ffff:0:0:0/96)

Identifier: RQ_003_3016
RFC Clause: 3.1
Type: Mandatory
Applies to: Router

Requirement:
As part of forwarding the packet if the translator has decremented the IPv4 TTL (before the translation) it SHALL not decrement the IPv6 Hop Limit (after the translation).

Specification Text:

If the DF bit is set and the packet is not a fragment (i.e., the MF flag is not set and the Fragment Offset is zero) then there is no need to add a fragment header to the packet. The IPv6 header fields are set as follows:

Version:
6

Traffic Class:
By default, copied from IP Type Of Service and Precedence field (all 8 bits are copied). According to [DIFFSERV] the semantics of the bits are identical in IPv4 and IPv6. However, in some IPv4 environments these fields might be used with the old semantics of "Type Of Service and Precedence". An implementation of a translator SHOULD provide the ability to ignore the IPv4 "TOS" and always set the IPv6 traffic class to zero.

Flow Label:
0 (all zero bits)

Payload Length:
Total length value from IPv4 header, minus the size of the IPv4 header and IPv4 options, if present.

Next Header:
Protocol field copied from IPv4 header

Hop Limit:
TTL value copied from IPv4 header. **Since the translator is a router, as part of forwarding the packet it needs to decrement either the IPv4 TTL (before the translation) or the IPv6 Hop Limit (after the translation).** As part of decrementing the TTL or Hop Limit the translator (as any router) needs to check for zero and send the ICMPv4 or ICMPv6 "ttl exceeded" error.

Source Address:
The low-order 32 bits is the IPv4 source address. The high-order 96 bits is the IPv4-mapped prefix (::ffff:0:0/96)

Destination Address:
The low-order 32 bits is the IPv4 destination address. The high-order 96 bits is the IPv4-translated prefix (0::ffff:0:0:0/96)

Identifier: RQ_003_3017
RFC Clause: 3.1
Type: Mandatory
Applies to: Router

Requirement:

As part of decrementing the TTL the translator needs to check for zero and if present, send the ICMPv4 "ttl exceeded" error.

Specification Text:

If the DF bit is set and the packet is not a fragment (i.e., the MF flag is not set and the Fragment Offset is zero) then there is no need to add a fragment header to the packet. The IPv6 header fields are set as follows:

Version:
6

Traffic Class:

By default, copied from IP Type Of Service and Precedence field (all 8 bits are copied). According to [DIFFSERV] the semantics of the bits are identical in IPv4 and IPv6. However, in some IPv4 environments these fields might be used with the old semantics of "Type Of Service and Precedence". An implementation of a translator SHOULD provide the ability to ignore the IPv4 "TOS" and always set the IPv6 traffic class to zero.

Flow Label:

0 (all zero bits)

Payload Length:

Total length value from IPv4 header, minus the size of the IPv4 header and IPv4 options, if present.

Next Header:

Protocol field copied from IPv4 header

Hop Limit:

TTL value copied from IPv4 header. Since the translator is a router, as part of forwarding the packet it needs to decrement either the IPv4 TTL (before the translation) or the IPv6 Hop Limit (after the translation). **As part of decrementing the TTL or Hop Limit the translator (as any router) needs to check for zero and send the ICMPv4 or ICMPv6 "ttl exceeded" error.**

Source Address:

The low-order 32 bits is the IPv4 source address. The high-order 96 bits is the IPv4-mapped prefix (::ffff:0:0/96)

Destination Address:

The low-order 32 bits is the IPv4 destination address. The high-order 96 bits is the IPv4-translated prefix (0::ffff:0:0:0/96)

Identifier:	RQ_003_3018
RFC Clause:	3.1
Type:	Mandatory
Applies to:	Router

Requirement:

When translating the IPv4 packet to IPV6, the resulting IPv6 header's Source Address field SHALL be constructed with the low-order 32 bits as the IPv4 source address and the high-order 96 bits as the IPv4-mapped prefix (::ffff:0:0/96).

Specification Text:

If the DF bit is set and the packet is not a fragment (i.e., the MF flag is not set and the Fragment Offset is zero) then there is no need to add a fragment header to the packet. **The IPv6 header fields are set as follows:**

Version:

6

Traffic Class:

By default, copied from IP Type Of Service and Precedence field (all 8 bits are copied). According to [DIFFSERV] the semantics of the bits are identical in IPv4 and IPv6. However, in some IPv4 environments these fields might be used with the old semantics of "Type Of Service and Precedence". An implementation of a translator SHOULD provide the ability to ignore the IPv4 "TOS" and always set the IPv6 traffic class to zero.

Flow Label:

0 (all zero bits)

Payload Length:

Total length value from IPv4 header, minus the size of the IPv4 header and IPv4 options, if present.

Next Header:

Protocol field copied from IPv4 header

Hop Limit:

TTL value copied from IPv4 header. Since the translator is a router, as part of forwarding the packet it needs to decrement either the IPv4 TTL (before the translation) or the IPv6 Hop Limit (after the translation). As part of decrementing the TTL or Hop Limit the translator (as any router) needs to check for zero and send the ICMPv4 or ICMPv6 "ttl exceeded" error.

Source Address:

The low-order 32 bits is the IPv4 source address. The high-order 96 bits is the IPv4-mapped prefix (::ffff:0:0/96)

Destination Address:

The low-order 32 bits is the IPv4 destination address. The high-order 96 bits is the IPv4-translated prefix (0::ffff:0:0:0/96)

Identifier: RQ_003_3019
RFC Clause: 3.1
Type: Mandatory
Applies to: Router

Requirement:

When translating the IPv4 packet to IPV6, the resulting IPv6 header's Destination Address field SHALL be constructed with the low-order 32 bits is the IPv4 destination address and the high-order 96 bits is the IPv4-translated prefix (0::ffff:0:0:0/96)

Specification Text:

If the DF bit is set and the packet is not a fragment (i.e., the MF flag is not set and the Fragment Offset is zero) then there is no need to add a fragment header to the packet. **The IPv6 header fields are set as follows:**

Version:

6

Traffic Class:

By default, copied from IP Type Of Service and Precedence field (all 8 bits are copied). According to [DIFFSERV] the semantics of the bits are identical in IPv4 and IPv6. However, in some IPv4 environments these fields might be used with the old semantics of "Type Of Service and Precedence". An implementation of a translator SHOULD provide the ability to ignore the IPv4 "TOS" and always set the IPv6 traffic class to zero.

Flow Label:

0 (all zero bits)

Payload Length:

Total length value from IPv4 header, minus the size of the IPv4 header and IPv4 options, if present.

Next Header:

Protocol field copied from IPv4 header

Hop Limit:

TTL value copied from IPv4 header. Since the translator is a router, as part of forwarding the packet it needs to decrement either the IPv4 TTL (before the translation) or the IPv6 Hop Limit (after the translation). As part of decrementing the TTL or Hop Limit the translator (as any router) needs to check for zero and send the ICMPv4 or ICMPv6 "ttl exceeded" error.

Source Address:
The low-order 32 bits is the IPv4 source address.
The high-order 96 bits is the IPv4-mapped prefix
(::ffff:0:0/96)

Destination Address:
The low-order 32 bits is the IPv4 destination
address. The high-order 96 bits is the IPv4-
translated prefix (0::ffff:0:0:0/96)

Identifier:	RQ_003_3020
RFC Clause:	3.1
Type:	Mandatory
Applies to:	Router

Requirement:

With the exception of the unexpired source route option, if IPv4 options are present in the IPv4 packet they SHALL be ignored by the translator.

Specification Text:

If IPv4 options are present in the IPv4 packet, they are ignored i.e., there is no attempt to translate them. However, if an unexpired source route option is present then the packet MUST instead be discarded, and an ICMPv4 "destination unreachable/source route failed" (Type 3/Code 5) error message SHOULD be returned to the sender.

Identifier:	RQ_003_3021
RFC Clause:	3.1
Type:	Mandatory
Applies to:	Router

Requirement:

If an unexpired source route option is present in an IPv4 packet then the packet MUST be discarded by the translator.

Specification Text:

If IPv4 options are present in the IPv4 packet, they are ignored i.e., there is no attempt to translate them**. However, if an unexpired source route option is present then the packet MUST instead be discarded,** and an ICMPv4 "destination unreachable/source route failed" (Type 3/Code 5) error message SHOULD be returned to the sender.

Identifier:	RQ_003_3022
RFC Clause:	3.1
Type:	Recommendation
Applies to:	Router

Requirement:

If an unexpired source route option is present in an IPv4 packet and the packet has been discarded by the translator, an ICMPv4 "destination unreachable/source route failed" (Type 3/Code 5) error message SHOULD be returned to the sender.

Specification Text:

If IPv4 options are present in the IPv4 packet, they are ignored i.e., there is no attempt to translate them. However, if an unexpired source route option is present then the packet MUST instead be discarded, **and an ICMPv4 "destination unreachable/source route failed" (Type 3/Code 5) error message SHOULD be returned to the sender.**

Identifier: RQ_003_3023
RFC Clause: 3.1
Type: Mandatory
Applies to: Router

Requirement:
If the DF bit is not set or the packet is a fragment, after translating the IPv4 packet to IPV6, the resulting IPv6 header's Payload Length Label field SHALL be set to the total length value from IPv4 header, plus 8 for the fragment header, minus the size of the IPv4 header and IPv4 options, if present.

Specification Text:
If there is need to add a fragment header (the DF bit is not set or the packet is a fragment) the header fields are set as above with the following exceptions:

IPv6 fields:

Payload Length:
Total length value from IPv4 header, plus 8 for the fragment header, minus the size of the IPv4 header and IPv4 options, if present.

Next Header:
Fragment Header (44).

Fragment header fields:

Next Header:
Protocol field copied from IPv4 header.
Fragment Offset:
Fragment Offset copied from the IPv4 header.

M flag:
More Fragments bit copied from the IPv4 header.

Identification:
The low-order 16 bits copied from the Identification field in the IPv4 header. The high-order 16 bits set to zero.

Identifier: RQ_003_3024
RFC Clause: 3.1
Type: Mandatory
Applies to: Router

Requirement:
If the DF bit is not set or the packet is a fragment, after translating the IPv4 packet to IPV6, the resulting IPv6 header's Next Header field SHALL be set to Fragment Header (44).

Specification Text:
If there is need to add a fragment header (the DF bit is not set or the packet is a fragment) the header fields are set as above with the following exceptions:

IPv6 fields:

Payload Length:
Total length value from IPv4 header, plus 8 for the fragment header, minus the size of the IPv4 header and IPv4 options, if present.

Next Header:
Fragment Header (44).

Fragment header fields:

Next Header:
Protocol field copied from IPv4 header.
Fragment Offset:
Fragment Offset copied from the IPv4 header.

M flag:
More Fragments bit copied from the IPv4 header.

Identification:
The low-order 16 bits copied from the Identification field in the IPv4 header. The high-order 16 bits set to zero.

Identifier: RQ_003_3025
RFC Clause: 3.1
Type: Mandatory
Applies to: Router

Requirement:
If the DF bit is not set or the packet is a fragment, after translating the IPv4 packet to IPV6, the resulting IPv6 Fragment header field's Next Header field SHALL be set to the protocol field copied from IPv4 header.

Specification Text:
If there is need to add a fragment header (the DF bit is not set or the packet is a fragment) the header fields are set as above with the following exceptions:

IPv6 fields:

Payload Length:
Total length value from IPv4 header, plus 8 for the fragment header, minus the size of the IPv4 header and IPv4 options, if present.

Next Header:
Fragment Header (44).

Fragment header fields:

Next Header:
Protocol field copied from IPv4 header.
Fragment Offset:
Fragment Offset copied from the IPv4 header.

M flag:
More Fragments bit copied from the IPv4 header.

Identification:
The low-order 16 bits copied from the Identification field in the IPv4 header. The high-order 16 bits set to zero.

Identifier: RQ_003_3026
RFC Clause: 3.1
Type: Mandatory
Applies to: Router

Requirement:
If the DF bit is not set or the packet is a fragment, after translating the IPv4 packet to IPV6, the resulting IPv6 Fragment header field's Fragment Offset field SHALL be set to the fragment Offset copied from the IPv4 header.

Specification Text:
If there is need to add a fragment header (the DF bit is not set or the packet is a fragment) the header fields are set as above with the following exceptions:

IPv6 fields:

Payload Length:
Total length value from IPv4 header, plus 8 for the fragment header, minus the size of the IPv4 header and IPv4 options, if present.

Next Header:
Fragment Header (44).

Fragment header fields:

Next Header:
Protocol field copied from IPv4 header.
Fragment Offset:
Fragment Offset copied from the IPv4 header.

M flag:
More Fragments bit copied from the IPv4 header.

Identification:
The low-order 16 bits copied from the Identification
field in the IPv4 header. The high-order 16 bits set
to zero.

Identifier: RQ_003_3027
RFC Clause: 3.1
Type: Mandatory
Applies to: Router

Requirement:
If the DF bit is not set or the packet is a fragment, after translating the IPv4 packet to IPV6, the resulting IPv6 Fragment header field's M flag field SHALL be set to the more Fragments bit copied from the IPv4 header.

Specification Text:
If there is need to add a fragment header (the DF bit is not set or the packet is a fragment) the header fields are set as above with the following exceptions:

IPv6 fields:

Payload Length:
Total length value from IPv4 header, plus 8 for the
fragment header, minus the size of the IPv4 header
and IPv4 options, if present.

Next Header:
Fragment Header (44).

Fragment header fields:

Next Header:
Protocol field copied from IPv4 header.
Fragment Offset:
Fragment Offset copied from the IPv4 header.

M flag:
More Fragments bit copied from the IPv4 header.

Identification:
The low-order 16 bits copied from the Identification
field in the IPv4 header. The high-order 16 bits set
to zero.

Identifier: RQ_003_3028
RFC Clause: 3.1
Type: Mandatory
Applies to: Router

Requirement:
If the DF bit is not set or the packet is a fragment, after translating the IPv4 packet to IPV6, the resulting IPv6 Fragment header field's Identification field SHALL be set to the low-order 16 bits copied from the Identification field in the IPv4 header. The high-order 16 bits set to zero.

Specification Text:
If there is need to add a fragment header (the DF bit is not set or the packet is a fragment) the header fields are set as above with the following exceptions:

IPv6 fields:

Payload Length:
Total length value from IPv4 header, plus 8 for the
fragment header, minus the size of the IPv4 header
and IPv4 options, if present.

Next Header:
Fragment Header (44).

Fragment header fields:

Next Header:
Protocol field copied from IPv4 header.

Fragment Offset:
Fragment Offset copied from the IPv4 header.

M flag:
More Fragments bit copied from the IPv4 header.

Identification:
**The low-order 16 bits copied from the Identification
field in the IPv4 header. The high-order 16 bits set
to zero.**

Identifier: RQ_003_3029
RFC Clause: 3.2
Type: Recommendation
Applies to: Router

Requirement:
When a translator receives the first fragment of a fragmented UDP IPv4 packet and the checksum field is zero the translator SHOULD drop the packet.

Specification Text:
When a translator receives the first fragment of a fragmented UDP IPv4 packet and the checksum field is zero the translator SHOULD drop the packet and generate a system management event specifying at least the IP addresses and port numbers in the packet. When it receives fragments other than the first it SHOULD silently drop the packet, since there is no port information to log.

Identifier: RQ_003_3030
RFC Clause: 3.2
Type: Recommendation
Applies to: Router

Requirement:
When a translator receives the first fragment of a fragmented UDP IPv4 packet and the checksum field is zero the translator SHOULD generate a system management event specifying at least the IP addresses and port numbers in the packet.

Specification Text:
When a translator receives the first fragment of a fragmented UDP IPv4 packet and the checksum field is zero the translator SHOULD drop the packet and **generate a system management event specifying at least the IP addresses and port numbers in the packet**. When it receives fragments other than the first it SHOULD silently drop the packet, since there is no port information to log.

Identifier: RQ_003_3031
RFC Clause: 3.2
Type: Recommendation
Applies to: Router

Requirement:
When a translator receives the fragments other than the first fragment of a fragmented UDP IPv4 packet and the checksum field is zero the translator SHOULD silently drop the packet.

Specification Text:
When a translator receives the first fragment of a **fragmented UDP IPv4 packet and the checksum field is zero the translator** SHOULD drop the packet and generate a system management event specifying at least the IP addresses and port numbers in the packet**.** **When it receives fragments other than the first it SHOULD silently drop the packet, since there is no port information to log.**

Identifier:	RQ_003_3032
RFC Clause:	3.2
Type:	Mandatory
Applies to:	Router

Requirement:
When a translator receives an unfragmented UDP IPv4 packet and the checksum field is zero the translator MUST compute the missing UDP checksum as part of translating the packet.

Specification Text:
When a translator receives an unfragmented UDP IPv4 packet and the checksum field is zero the translator MUST compute the missing UDP checksum as part of translating the packet. Also, the translator SHOULD maintain a counter of how many UDP checksums are generated in this manner.

Identifier:	RQ_003_3033
RFC Clause:	3.2
Type:	Recommendation
Applies to:	Router

Requirement:
When a translator receives an unfragmented UDP IPv4 packet and the checksum field is zero the translator SHOULD maintain a counter of how many UDP checksums are generated in this manner.

Specification Text:
When a translator receives an unfragmented UDP IPv4 packet and the checksum field is zero the translator MUST compute the missing UDP checksum as part of translating the packet. Also, the translator **SHOULD maintain a counter of how many UDP checksums are generated in this manner.**

Identifier:	RQ_003_3034
RFC Clause:	3.3
Type:	Mandatory
Applies to:	Router

Requirement:
All ICMP messages that are to be translated require that the ICMP checksum field be updated as part of the translation.

Specification Text:
All ICMP messages that are to be translated require that the ICMP checksum field be updated as part of the translation since ICMPv6, unlike ICMPv4, has a pseudo-header checksum just like UDP and TCP. In addition all ICMP packets need to have the Type value translated and for ICMP error messages the included IP header also needs translation.

Identifier:	RQ_003_3035
RFC Clause:	3.3
Type:	Mandatory
Applies to:	Router

Requirement:
All ICMP packets that are to be translated require that the Type value be translated.

Specification Text:
All ICMP messages that are to be translated require that the ICMP checksum field be updated as part of the translation since ICMPv6, unlike ICMPv4, has a pseudo-header checksum just like UDP and TCP. **In addition all ICMP packets need to have the Type value translated** and for ICMP error messages the included IP header also needs translation.

Identifier: RQ_003_3036
RFC Clause: 3.3
Type: Mandatory
Applies to: Router

Requirement:
All ICMP packets that are to be translated require that within ICMP error messages the included IP header also needs translation.

Specification Text:
All ICMP messages that are to be translated require that the ICMP checksum field be updated as part of the translation since ICMPv6, unlike ICMPv4, has a pseudo-header checksum just like UDP and TCP. **In addition all ICMP packets need to have the Type value translated and for ICMP error messages the included IP header also needs translation.**

Identifier: RQ_003_3037
RFC Clause: 3.3
Type: Mandatory
Applies to: Router

Requirement:
In order to translate ICMPv4 query messages, for the Echo (Type 8) message, adjust the type to 128.

Specification Text:
```
The actions needed to translate various ICMPv4 messages are:

ICMPv4 query messages:

  Echo and Echo Reply (Type 8 and Type 0)
     Adjust the type to 128 and 129, respectively, and adjust the
     ICMP checksum both to take the type change into account and
     to include the ICMPv6 pseudo-header.

  Information Request/Reply (Type 15 and Type 16)
     Obsoleted in ICMPv4.  Silently drop.

  Timestamp and Timestamp Reply (Type 13 and Type 14)
     Obsoleted in ICMPv6.  Silently drop.

  Address Mask Request/Reply (Type 17 and Type 18)
     Obsoleted in ICMPv6.  Silently drop.

  ICMP Router Advertisement (Type 9)
     Single hop message.  Silently drop.

  ICMP Router Solicitation (Type 10)
     Single hop message.  Silently drop.

  Unknown ICMPv4 types
     Silently drop.
```

Identifier: RQ_003_3038
RFC Clause: 3.3
Type: Mandatory
Applies to: Router

Requirement:
In order to translate ICMPv4 query messages, after adjusting the Echo (Type 8) message to 128, adjust the ICMP checksum both to take the type change into account and to include the ICMPv6 pseudo-header.

Specification Text:
The actions needed to translate various ICMPv4 messages are:

ICMPv4 query messages:

  Echo and Echo Reply (**Type 8** and Type 0)
    Adjust the type to 128 and 129, respectively, and **adjust the**
    ICMP checksum both to take the type change into account and
    to include the ICMPv6 pseudo-header.

  Information Request/Reply (Type 15 and Type 16)
    Obsoleted in ICMPv4. Silently drop.

  Timestamp and Timestamp Reply (Type 13 and Type 14)
    Obsoleted in ICMPv6. Silently drop.

  Address Mask Request/Reply (Type 17 and Type 18)
    Obsoleted in ICMPv6. Silently drop.

  ICMP Router Advertisement (Type 9)
    Single hop message. Silently drop.

  ICMP Router Solicitation (Type 10)
    Single hop message. Silently drop.

  Unknown ICMPv4 types
    Silently drop.

Identifier: RQ_003_3039
RFC Clause: 3.3
Type: Mandatory
Applies to: Router

Requirement:
In order to translate ICMPv4 query messages, for the Echo Reply (Type 0) message, adjust the type to 129.

Specification Text:
The actions needed to translate various ICMPv4 messages are:

ICMPv4 query messages:

  Echo and **Echo Reply** (Type 8 and **Type 0**)
    Adjust the type to 128 and **129,** respectively, and adjust the
    ICMP checksum both to take the type change into account and
    to include the ICMPv6 pseudo-header.

  Information Request/Reply (Type 15 and Type 16)
    Obsoleted in ICMPv4. Silently drop.

  Timestamp and Timestamp Reply (Type 13 and Type 14)
    Obsoleted in ICMPv6. Silently drop.

  Address Mask Request/Reply (Type 17 and Type 18)
    Obsoleted in ICMPv6. Silently drop.

  ICMP Router Advertisement (Type 9)
    Single hop message. Silently drop.

  ICMP Router Solicitation (Type 10)
    Single hop message. Silently drop.

  Unknown ICMPv4 types
    Silently drop.

Identifier: RQ_003_3040
RFC Clause: 3.3
Type: Mandatory
Applies to: Router

Requirement:
In order to translate ICMPv4 query messages, after adjusting the Echo Reply (Type 0) message to 129, adjust the ICMP checksum both to take the type change into account and to include the ICMPv6 pseudo-header.

Specification Text:

The actions needed to translate various ICMPv4 messages are:

ICMPv4 query messages:

Echo and **Echo Reply** (Type 8 and **Type 0**)
Adjust the type to 128 and 129, respectively, and adjust the ICMP checksum both to take the type change into account and to include the ICMPv6 pseudo-header.

Information Request/Reply (Type 15 and Type 16)
Obsoleted in ICMPv4. Silently drop.

Timestamp and Timestamp Reply (Type 13 and Type 14)
Obsoleted in ICMPv6. Silently drop.

Address Mask Request/Reply (Type 17 and Type 18)
Obsoleted in ICMPv6. Silently drop.

ICMP Router Advertisement (Type 9)
Single hop message. Silently drop.

ICMP Router Solicitation (Type 10)
Single hop message. Silently drop.

Unknown ICMPv4 types
Silently drop.

Identifier: RQ_003_3041
RFC Clause: 3.3
Type: Mandatory
Applies to: Router

Requirement:

In order to translate ICMPv4 query messages, Silently drop Information Request (Type 15).

Specification Text:

The actions needed to translate various ICMPv4 messages are:

ICMPv4 query messages:

Echo and Echo Reply (Type 8 and Type 0)
Adjust the type to 128 and 129, respectively, and adjust the ICMP checksum both to take the type change into account and to include the ICMPv6 pseudo-header.

Information Request/Reply (**Type 15** and Type 16)
Obsoleted in ICMPv4. **Silently drop.**

Timestamp and Timestamp Reply (Type 13 and Type 14)
Obsoleted in ICMPv6. Silently drop.

Address Mask Request/Reply (Type 17 and Type 18)
Obsoleted in ICMPv6. Silently drop.

ICMP Router Advertisement (Type 9)
Single hop message. Silently drop.

ICMP Router Solicitation (Type 10)
Single hop message. Silently drop.

Unknown ICMPv4 types
Silently drop.

Identifier: RQ_003_3042
RFC Clause: 3.3
Type: Mandatory
Applies to: Router

Requirement:
In order to translate ICMPv4 query messages, Silently drop Information Reply (Type 16).

Specification Text:

```
The actions needed to translate various ICMPv4 messages are:

ICMPv4 query messages:

  Echo and Echo Reply (Type 8 and Type 0)
     Adjust the type to 128 and 129, respectively, and adjust the
     ICMP checksum both to take the type change into account and
     to include the ICMPv6 pseudo-header.

  Information Request/Reply (Type 15 and Type 16)
     Obsoleted in ICMPv4.  Silently drop.

  Timestamp and Timestamp Reply (Type 13 and Type 14)
     Obsoleted in ICMPv6.  Silently drop.

  Address Mask Request/Reply (Type 17 and Type 18)
     Obsoleted in ICMPv6.  Silently drop.

  ICMP Router Advertisement (Type 9)
     Single hop message.  Silently drop.

  ICMP Router Solicitation (Type 10)
     Single hop message.  Silently drop.

  Unknown ICMPv4 types
     Silently drop.
```

Identifier: RQ_003_3043
RFC Clause: 3.3
Type: Mandatory
Applies to: Router

Requirement:
In order to translate ICMPv4 query messages, Silently drop Timestamp (Type 13).

Specification Text:

```
   The actions needed to translate various ICMPv4 messages are:

ICMPv4 query messages:

  Echo and Echo Reply (Type 8 and Type 0)
     Adjust the type to 128 and 129, respectively, and adjust the
     ICMP checksum both to take the type change into account and
     to include the ICMPv6 pseudo-header.

  Information Request/Reply (Type 15 and Type 16)
     Obsoleted in ICMPv4.  Silently drop.

  Timestamp and Timestamp Reply (Type 13 and Type 14)
     Obsoleted in ICMPv6.  Silently drop.

  Address Mask Request/Reply (Type 17 and Type 18)
     Obsoleted in ICMPv6.  Silently drop.

  ICMP Router Advertisement (Type 9)
     Single hop message.  Silently drop.

  ICMP Router Solicitation (Type 10)
     Single hop message.  Silently drop.

  Unknown ICMPv4 types
     Silently drop.
```

Identifier: RQ_003_3044
RFC Clause: 3.3
Type: Mandatory
Applies to: Router

Requirement:
In order to translate ICMPv4 query messages, Silently drop Timestamp Reply (Type 14).

Specification Text:
The actions needed to translate various ICMPv4 messages are:

ICMPv4 query messages:

Echo and Echo Reply (Type 8 and Type 0)
Adjust the type to 128 and 129, respectively, and adjust the ICMP checksum both to take the type change into account and to include the ICMPv6 pseudo-header.

Information Request/Reply (Type 15 and Type 16)
Obsoleted in ICMPv4. Silently drop.

Timestamp and **Timestamp Reply** (Type 13 and **Type 14**)
Obsoleted in ICMPv6. **Silently drop.**

Address Mask Request/Reply (Type 17 and Type 18)
Obsoleted in ICMPv6. Silently drop.

ICMP Router Advertisement (Type 9)
Single hop message. Silently drop.

ICMP Router Solicitation (Type 10)
Single hop message. Silently drop.

Unknown ICMPv4 types
Silently drop.

Identifier: RQ_003_3045
RFC Clause: 3.3
Type: Mandatory
Applies to: Router

Requirement:
In order to translate ICMPv4 query messages, Silently drop Address Mask Request (Type 17).

Specification Text:
The actions needed to translate various ICMPv4 messages are:

ICMPv4 query messages:

Echo and Echo Reply (Type 8 and Type 0)
Adjust the type to 128 and 129, respectively, and adjust the ICMP checksum both to take the type change into account and to include the ICMPv6 pseudo-header.

Information Request/Reply (Type 15 and Type 16)
Obsoleted in ICMPv4. Silently drop.

Timestamp and Timestamp Reply (Type 13 and Type 14)
Obsoleted in ICMPv6. Silently drop.

Address Mask Request/Reply (**Type 17** and Type 18)
Obsoleted in ICMPv6. **Silently drop.**

ICMP Router Advertisement (Type 9)
Single hop message. Silently drop.

ICMP Router Solicitation (Type 10)
Single hop message. Silently drop.

Unknown ICMPv4 types
Silently drop.

Identifier: RQ_003_3046
RFC Clause: 3.3
Type: Mandatory
Applies to: Router

Requirement:
In order to translate ICMPv4 query messages, Silently drop Address Mask Reply (Type 18).

Specification Text:
The actions needed to translate various ICMPv4 messages are:

ICMPv4 query messages:

Echo and Echo Reply (Type 8 and Type 0)
Adjust the type to 128 and 129, respectively, and adjust the
ICMP checksum both to take the type change into account and
to include the ICMPv6 pseudo-header.

Information Request/Reply (Type 15 and Type 16)
Obsoleted in ICMPv4. Silently drop.

Timestamp and Timestamp Reply (Type 13 and Type 14)
Obsoleted in ICMPv6. Silently drop.

Address Mask Request/**Reply** (Type 17 and **Type 18**)
Obsoleted in ICMPv6. Silently drop.

ICMP Router Advertisement (Type 9)
Single hop message. Silently drop.

ICMP Router Solicitation (Type 10)
Single hop message. Silently drop.

Unknown ICMPv4 types
Silently drop.

Identifier: RQ_003_3047
RFC Clause: 3.3
Type: Mandatory
Applies to: Router

Requirement:
In order to translate ICMPv4 query messages, Silently drop ICMP Router Advertisement (Type 9)

Specification Text:
The actions needed to translate various ICMPv4 messages are:

ICMPv4 query messages:

Echo and Echo Reply (Type 8 and Type 0)
Adjust the type to 128 and 129, respectively, and adjust the
ICMP checksum both to take the type change into account and
to include the ICMPv6 pseudo-header.

Information Request/Reply (Type 15 and Type 16)
Obsoleted in ICMPv4. Silently drop.

Timestamp and Timestamp Reply (Type 13 and Type 14)
Obsoleted in ICMPv6. Silently drop.

Address Mask Request/Reply (Type 17 and Type 18)
Obsoleted in ICMPv6. Silently drop.

ICMP Router Advertisement (Type 9)
Single hop message. Silently drop.

ICMP Router Solicitation (Type 10)
Single hop message. Silently drop.

Unknown ICMPv4 types
Silently drop.

Identifier:	RQ_003_3048
RFC Clause:	3.3
Type:	Mandatory
Applies to:	Router

Requirement:
In order to translate ICMPv4 query messages, Silently drop ICMP Router Solicitation (Type 10)

Specification Text:
The actions needed to translate various ICMPv4 messages are:

ICMPv4 query messages:

Echo and Echo Reply (Type 8 and Type 0)
Adjust the type to 128 and 129, respectively, and adjust the
ICMP checksum both to take the type change into account and
to include the ICMPv6 pseudo-header.

Information Request/Reply (Type 15 and Type 16)
Obsoleted in ICMPv4. Silently drop.

Timestamp and Timestamp Reply (Type 13 and Type 14)
Obsoleted in ICMPv6. Silently drop.

Address Mask Request/Reply (Type 17 and Type 18)
Obsoleted in ICMPv6. Silently drop.

ICMP Router Advertisement (Type 9)
Single hop message. Silently drop.

ICMP Router Solicitation (Type 10)
Single hop message. Silently drop.

Unknown ICMPv4 types
Silently drop.

Identifier:	RQ_003_3049
RFC Clause:	3.3
Type:	Mandatory
Applies to:	Router

Requirement:
In order to translate ICMPv4 query messages, Silently drop Unknown ICMPv4 types

Specification Text:
The actions needed to translate various ICMPv4 messages are:

ICMPv4 query messages:

Echo and Echo Reply (Type 8 and Type 0)
Adjust the type to 128 and 129, respectively, and adjust the
ICMP checksum both to take the type change into account and
to include the ICMPv6 pseudo-header.

Information Request/Reply (Type 15 and Type 16)
Obsoleted in ICMPv4. Silently drop.

Timestamp and Timestamp Reply (Type 13 and Type 14)
Obsoleted in ICMPv6. Silently drop.

Address Mask Request/Reply (Type 17 and Type 18)
Obsoleted in ICMPv6. Silently drop.

ICMP Router Advertisement (Type 9)
Single hop message. Silently drop.

ICMP Router Solicitation (Type 10)
Single hop message. Silently drop.

Unknown ICMPv4 types
Silently drop.

Identifier:	RQ_003_3050
RFC Clause:	3.3
Type:	Recommendation
Applies to:	Router

Requirement:

All the IGMP messages should be silently dropped by the translator.

Specification Text:

IGMP messages:

While the MLD messages [MLD] are the logical IPv6 counterparts for the IPv4 IGMP messages all the "normal" IGMP messages are single-hop messages and should be silently dropped by the translator. Other IGMP messages might be used by multicast routing protocols and, since it would be a configuration error to try to have router adjacencies across IPv4/IPv6 translators those packets should also be silently dropped.

Identifier:	RQ_003_3051
RFC Clause:	3.3
Type:	Mandatory
Applies to:	Router

Requirement:

In order to translate ICMPv4 error messages, for Destination Unreachable (Type 3), if the Code field is set to 0 then translate Code to 0 (no route to destination).

Specification Text:

ICMPv4 error messages:

Destination Unreachable (Type 3)

For all that are not explicitly listed below set the Type to 1.

Translate the code field as follows:

Code 0, 1 (net, host unreachable):
Set Code to 0 (no route to destination).

Code 2 (protocol unreachable):
Translate to an ICMPv6 Parameter Problem (Type 4, Code 1) and make the Pointer point to the IPv6 Next Header field.

Code 3 (port unreachable):
Set Code to 4 (port unreachable).

Code 4 (fragmentation needed and DF set):
Translate to an ICMPv6 Packet Too Big message (Type 2) with code 0. The MTU field needs to be adjusted for the difference between the IPv4 and IPv6 header sizes. Note that if the IPv4 router did not set the MTU field i.e. the router does not implement [PMTUv4], then the translator MUST use the plateau values specified in [PMTUv4] to determine a likely path MTU and include that path MTU in the ICMPv6 packet. (Use the greatest plateau value that is less than the returned Total Length field.)

Code 5 (source route failed):
Set Code to 0 (no route to destination). Note that this error is unlikely since source routes are not translated.

Code 6,7:
Set Code to 0 (no route to destination).

Code 8:
Set Code to 0 (no route to destination).

Code 9, 10 (communication with destination host administratively prohibited):
Set Code to 1 (communication with destination administratively prohibited)

Code 11, 12:
Set Code to 0 (no route to destination).

Identifier: RQ_003_3052
RFC Clause: 3.3
Type: Mandatory
Applies to: Router

Requirement:

In order to translate ICMPv4 error messages, for Destination Unreachable (Type 3), if the Code field is set to Code 2 (protocol unreachable) then translate to an ICMPv6 Parameter Problem (Type 4, Code 1) and make the Pointer point to the IPv6 Next Header field.

Specification Text:

ICMPv4 error messages:

Destination Unreachable (Type 3)

For all that are not explicitly listed below set the Type to 1.

Translate the code field as follows:

Code 0, 1 (net, host unreachable):
Set Code to 0 (no route to destination).

Code 2 (protocol unreachable):
Translate to an ICMPv6 Parameter Problem (Type 4, Code 1) and make the Pointer point to the IPv6 Next Header field.

Code 3 (port unreachable):
Set Code to 4 (port unreachable).

Code 4 (fragmentation needed and DF set):
Translate to an ICMPv6 Packet Too Big message (Type 2) with code 0. The MTU field needs to be adjusted for the difference between the IPv4 and IPv6 header sizes. Note that if the IPv4 router did not set the MTU field i.e. the router does not implement [PMTUv4], then the translator MUST use the plateau values specified in [PMTUv4] to determine a likely path MTU and include that path MTU in the ICMPv6 packet. (Use the greatest plateau value that is less than the returned Total Length field.)

Code 5 (source route failed):
Set Code to 0 (no route to destination). Note that this error is unlikely since source routes are not translated.

Code 6,7:
Set Code to 0 (no route to destination).

Code 8:
Set Code to 0 (no route to destination).

Code 9, 10 (communication with destination host administratively prohibited):
Set Code to 1 (communication with destination administratively prohibited)

Code 11, 12:
Set Code to 0 (no route to destination).

Identifier: RQ_003_3053
RFC Clause: 3.3
Type: Mandatory
Applies to: Router

Requirement:
In order to translate ICMPv4 error messages, for Destination Unreachable (Type 3), if the Code field is set to Code 3 (port unreachable) then translate to Code to 4 (port unreachable).

Specification Text:

```
ICMPv4 error messages:

  Destination Unreachable (Type 3)
     For all that are not explicitly listed below set the Type to
     1.

     Translate the code field as follows:
        Code 0, 1 (net, host unreachable):
                     Set Code to 0 (no route to destination).
        Code 2 (protocol unreachable):
               Translate to an ICMPv6 Parameter Problem (Type 4,
               Code 1) and make the Pointer point to the IPv6 Next
               Header field.

        Code 3 (port unreachable):
               Set Code to 4 (port unreachable).

        Code 4 (fragmentation needed and DF set):
               Translate to an ICMPv6 Packet Too Big message (Type
               2) with code 0.  The MTU field needs to be adjusted
               for the difference between the IPv4 and IPv6 header
               sizes.  Note that if the IPv4 router did not set
               the MTU field i.e. the router does not implement
               [PMTUv4], then the translator MUST use the plateau
               values specified in [PMTUv4] to determine a likely
               path MTU and include that path MTU in the ICMPv6
               packet. (Use the greatest plateau value that is
               less than the returned Total Length field.)

        Code 5 (source route failed):
               Set Code to 0 (no route to destination).  Note that
               this error is unlikely since source routes are not
               translated.

        Code 6,7:
               Set Code to 0 (no route to destination).

        Code 8:
               Set Code to 0 (no route to destination).

        Code 9, 10 (communication with destination host
        administratively prohibited):
               Set Code to 1 (communication with destination
               administratively prohibited)

        Code 11, 12:
               Set Code to 0 (no route to destination).
```

Identifier: RQ_003_3054
RFC Clause: 3.3
Type: Mandatory
Applies to: Router

Requirement:
In order to translate ICMPv4 error messages, for Destination Unreachable (Type 3), if the Code field is set to Code 4 (fragmentation needed and DF set) then translate to to an ICMPv6 Packet Too Big message (Type 2) with code 0.

Specification Text:

```
ICMPv4 error messages:

  Destination Unreachable (Type 3)
     For all that are not explicitly listed below set the Type to
     1.

     Translate the code field as follows:
        Code 0, 1 (net, host unreachable):
                     Set Code to 0 (no route to destination).
        Code 2 (protocol unreachable):
               Translate to an ICMPv6 Parameter Problem (Type 4,
               Code 1) and make the Pointer point to the IPv6 Next
               Header field.

        Code 3 (port unreachable):
               Set Code to 4 (port unreachable).

        Code 4 (fragmentation needed and DF set):
               Translate to an ICMPv6 Packet Too Big message (Type
               2) with code 0.  The MTU field needs to be adjusted
               for the difference between the IPv4 and IPv6 header
               sizes.  Note that if the IPv4 router did not set
               the MTU field i.e. the router does not implement
               [PMTUv4], then the translator MUST use the plateau
               values specified in [PMTUv4] to determine a likely
               path MTU and include that path MTU in the ICMPv6
               packet. (Use the greatest plateau value that is
               less than the returned Total Length field.)

        Code 5 (source route failed):
               Set Code to 0 (no route to destination).  Note that
               this error is unlikely since source routes are not
               translated.

        Code 6,7:
               Set Code to 0 (no route to destination).

        Code 8:
               Set Code to 0 (no route to destination).

        Code 9, 10 (communication with destination host
        administratively prohibited):
               Set Code to 1 (communication with destination
               administratively prohibited)

        Code 11, 12:
               Set Code to 0 (no route to destination).
```

Identifier: RQ_003_3055
RFC Clause: 3.3
Type: Mandatory
Applies to: Router

Requirement:

In order to translate ICMPv4 error messages, for Destination Unreachable (Type 3), if the Code field is set to Code 4 (fragmentation needed and DF set) then the MTU field, if set, needs to be adjusted for the difference between the IPv4 and IPv6 header sizes.

Specification Text:

```
ICMPv4 error messages:

  Destination Unreachable (Type 3)
     For all that are not explicitly listed below set the Type to
     1.

     Translate the code field as follows:
        Code 0, 1 (net, host unreachable):
                     Set Code to 0 (no route to destination).
        Code 2 (protocol unreachable):
               Translate to an ICMPv6 Parameter Problem (Type 4,
               Code 1) and make the Pointer point to the IPv6 Next
               Header field.
```

Code 3 (port unreachable):
Set Code to 4 (port unreachable).

Code 4 (fragmentation needed and DF set):
Translate to an ICMPv6 Packet Too Big message (Type 2) with code 0. **The MTU field needs to be adjusted for the difference between the IPv4 and IPv6 header sizes.** Note that if the IPv4 router did not set the MTU field i.e. the router does not implement [PMTUv4], then the translator MUST use the plateau values specified in [PMTUv4] to determine a likely path MTU and include that path MTU in the ICMPv6 packet. (Use the greatest plateau value that is less than the returned Total Length field.)

Code 5 (source route failed):
Set Code to 0 (no route to destination). Note that this error is unlikely since source routes are not translated.

Code 6,7:
Set Code to 0 (no route to destination).

Code 8:
Set Code to 0 (no route to destination).

Code 9, 10 (communication with destination host administratively prohibited):
Set Code to 1 (communication with destination administratively prohibited)

Code 11, 12:
Set Code to 0 (no route to destination).

Identifier: RQ_003_3056
RFC Clause: 3.3
Type: Mandatory
Applies to: Router

Requirement:
In order to translate ICMPv4 error messages, for Destination Unreachable (Type 3), if the Code field is set to Code 4 (fragmentation needed and DF set) and the MTU field is not set, then the translator MUST use the plateau values specified in [RFC 1191] to determine a likely path MTU and include that path MTU in the ICMPv6 packet. Use the greatest plateau value that is less than the returned Total Length field.

Specification Text:
ICMPv4 error messages:

Destination Unreachable (Type 3)
For all that are not explicitly listed below set the Type to 1.

Translate the code field as follows:
Code 0, 1 (net, host unreachable):
Set Code to 0 (no route to destination).
Code 2 (protocol unreachable):
Translate to an ICMPv6 Parameter Problem (Type 4, Code 1) and make the Pointer point to the IPv6 Next Header field.

Code 3 (port unreachable):
Set Code to 4 (port unreachable).

Code 4 (fragmentation needed and DF set):
Translate to an ICMPv6 Packet Too Big message (Type 2) with code 0. The MTU field needs to be adjusted for the difference between the IPv4 and IPv6 header sizes. **Note that if the IPv4 router did not set the MTU field i.e. the router does not implement [PMTUv4], then the translator MUST use the plateau values specified in [PMTUv4] to determine a likely path MTU and include that path MTU in the ICMPv6 packet. (Use the greatest plateau value that is less than the returned Total Length field.)**

```
        Code 5 (source route failed):
               Set Code to 0 (no route to destination).  Note that
               this error is unlikely since source routes are not
               translated.

        Code 6,7:
               Set Code to 0 (no route to destination).

        Code 8:
               Set Code to 0 (no route to destination).

        Code 9, 10 (communication with destination host
        administratively prohibited):
               Set Code to 1 (communication with destination
               administratively prohibited)

        Code 11, 12:
               Set Code to 0 (no route to destination).
```

Identifier: RQ_003_3057
RFC Clause: 3.3
Type: Mandatory
Applies to: Router

Requirement:

In order to translate ICMPv4 error messages, for Destination Unreachable (Type 3), if the Code field is set to Code 9 or 10 (communication with destination host administratively prohibited) then the translator MUST set the Code to 1 (communication with destination administratively prohibited)

Specification Text:

ICMPv4 error messages:

```
  Destination Unreachable (Type 3)
     For all that are not explicitly listed below set the Type to
     1.

     Translate the code field as follows:
        Code 0, 1 (net, host unreachable):
                     Set Code to 0 (no route to destination).
        Code 2 (protocol unreachable):
               Translate to an ICMPv6 Parameter Problem (Type 4,
               Code 1) and make the Pointer point to the IPv6 Next
               Header field.

        Code 3 (port unreachable):
               Set Code to 4 (port unreachable).

        Code 4 (fragmentation needed and DF set):
               Translate to an ICMPv6 Packet Too Big message (Type
               2) with code 0.  The MTU field needs to be adjusted
               for the difference between the IPv4 and IPv6 header
               sizes.  Note that if the IPv4 router did not set
               the MTU field i.e. the router does not implement
               [PMTUv4], then the translator MUST use the plateau
               values specified in [PMTUv4] to determine a likely
               path MTU and include that path MTU in the ICMPv6
               packet. (Use the greatest plateau value that is
               less than the returned Total Length field.)

        Code 5 (source route failed):
               Set Code to 0 (no route to destination).  Note that
               this error is unlikely since source routes are not
               translated.

        Code 6,7:
               Set Code to 0 (no route to destination).

        Code 8:
               Set Code to 0 (no route to destination).

        Code 9, 10 (communication with destination host
        administratively prohibited):
               Set Code to 1 (communication with destination
               administratively prohibited)
        Code 11, 12:
               Set Code to 0 (no route to destination).
```

Identifier: RQ_003_3058
RFC Clause: 3.3
Type: Mandatory
Applies to: Router

Requirement:
In order to translate ICMPv4 error messages, the translator SHALL Silently drop Redirect (Type 5) and Source Quench (Type 4) messages.

Specification Text:

```
Redirect (Type 5)
     Single hop message.  Silently drop.

  Source Quench (Type 4)
     Obsoleted in ICMPv6.  Silently drop.

  Time Exceeded (Type 11)
           Set the Type field to 3.  The Code field is unchanged.
  Parameter Problem (Type 12)
     Set the Type field to 4.  The Pointer needs to be updated to
     point to the corresponding field in the translated include
     IP header.
```

Identifier: RQ_003_3059
RFC Clause: 3.3
Type: Mandatory
Applies to: Router

Requirement:
In order to translate ICMPv4 error messages, the translator SHALL for Time Exceeded (Type 11)messages, Set the Type field to 3.

Specification Text:

```
Redirect (Type 5)
     Single hop message.  Silently drop.

  Source Quench (Type 4)
     Obsoleted in ICMPv6.  Silently drop.

  Time Exceeded (Type 11)
           Set the Type field to 3.  The Code field is unchanged.
  Parameter Problem (Type 12)
     Set the Type field to 4.  The Pointer needs to be updated to
     point to the corresponding field in the translated include
     IP header.
```

Identifier: RQ_003_3060
RFC Clause: 3.3
Type: Mandatory
Applies to: Router

Requirement:
In order to translate ICMPv4 error messages, for Time Exceeded (Type 11) messages, the Code field is unchanged.

Specification Text:

```
Redirect (Type 5)
     Single hop message.  Silently drop.

  Source Quench (Type 4)
     Obsoleted in ICMPv6.  Silently drop.

  Time Exceeded (Type 11)
           Set the Type field to 3.  The Code field is unchanged.
  Parameter Problem (Type 12)
     Set the Type field to 4.  The Pointer needs to be updated to
     point to the corresponding field in the translated include
     IP header.
```

Identifier: RQ_003_3061
RFC Clause: 3.3
Type: Mandatory
Applies to: Router

Requirement:
In order to translate ICMPv4 error messages, for Parameter Problem (Type 12) messages, the Pointer needs to be updated to point to the corresponding field in the translated include IP header.

Specification Text:

```
Redirect (Type 5)
     Single hop message.  Silently drop.

  Source Quench (Type 4)
     Obsoleted in ICMPv6.  Silently drop.

  Time Exceeded (Type 11)
           Set the Type field to 3.  The Code field is unchanged.
  Parameter Problem (Type 12)
     Set the Type field to 4.  The Pointer needs to be updated to
     point to the corresponding field in the translated include
     IP header.
```

Identifier: RQ_003_3062
RFC Clause: 3.3
Type: Mandatory
Applies to: Router

Requirement:
In order to translate ICMPv4 error messages, for Destination Unreachable (Type 3) messages, with code field other than 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 or 12 the translator shall set the Type to 1.

Specification Text:

```
ICMPv4 error messages:

  Destination Unreachable (Type 3)
     For all that are not explicitly listed below set the Type to
     1.

     Translate the code field as follows:
        Code 0, 1 (net, host unreachable):
                 Set Code to 0 (no route to destination).
        Code 2 (protocol unreachable):
               Translate to an ICMPv6 Parameter Problem (Type 4,
               Code 1) and make the Pointer point to the IPv6 Next
               Header field.

        Code 3 (port unreachable):
               Set Code to 4 (port unreachable).

        Code 4 (fragmentation needed and DF set):
               Translate to an ICMPv6 Packet Too Big message (Type
               2) with code 0.  The MTU field needs to be adjusted
               for the difference between the IPv4 and IPv6 header
               sizes.  Note that if the IPv4 router did not set
               the MTU field i.e. the router does not implement
               [PMTUv4], then the translator MUST use the plateau
               values specified in [PMTUv4] to determine a likely
               path MTU and include that path MTU in the ICMPv6
               packet. (Use the greatest plateau value that is
               less than the returned Total Length field.)

        Code 5 (source route failed):
               Set Code to 0 (no route to destination).  Note that
               this error is unlikely since source routes are not
               translated.

        Code 6,7:
               Set Code to 0 (no route to destination).

        Code 8:
               Set Code to 0 (no route to destination).

        Code 9, 10 (communication with destination host
        administratively prohibited):
               Set Code to 1 (communication with destination
               administratively prohibited)

        Code 11, 12:
               Set Code to 0 (no route to destination).
```

Identifier: RQ_003_3064
RFC Clause: 3.3
Type: Mandatory
Applies to: Router

Requirement:
In order to translate ICMPv4 error messages, for Destination Unreachable (Type 3), if the Code field is set to 1 then translate Code to 0 (no route to destination).

Specification Text:

```
ICMPv4 error messages:

  Destination Unreachable (Type 3)
     For all that are not explicitly listed below set the Type to
     1.

     Translate the code field as follows:
        Code 0, 1 (net, host unreachable):
                     Set Code to 0 (no route to destination).
        Code 2 (protocol unreachable):
               Translate to an ICMPv6 Parameter Problem (Type 4,
               Code 1) and make the Pointer point to the IPv6 Next
               Header field.

        Code 3 (port unreachable):
               Set Code to 4 (port unreachable).

        Code 4 (fragmentation needed and DF set):
               Translate to an ICMPv6 Packet Too Big message (Type
               2) with code 0.  The MTU field needs to be adjusted
               for the difference between the IPv4 and IPv6 header
               sizes.  Note that if the IPv4 router did not set
               the MTU field i.e. the router does not implement
               [PMTUv4], then the translator MUST use the plateau
               values specified in [PMTUv4] to determine a likely
               path MTU and include that path MTU in the ICMPv6
               packet. (Use the greatest plateau value that is
               less than the returned Total Length field.)

        Code 5 (source route failed):
               Set Code to 0 (no route to destination).  Note that
               this error is unlikely since source routes are not
               translated.

        Code 6,7:
               Set Code to 0 (no route to destination).

        Code 8:
               Set Code to 0 (no route to destination).

        Code 9, 10 (communication with destination host
        administratively prohibited):
               Set Code to 1 (communication with destination
               administratively prohibited)

        Code 11, 12:
               Set Code to 0 (no route to destination).
```

Identifier: RQ_003_3065
RFC Clause: 3.3
Type: Mandatory
Applies to: Router

Requirement:

In order to translate ICMPv4 error messages, for Destination Unreachable (Type 3), if the Code field is set to 5 then translate Code to 0 (no route to destination).

Specification Text:

```
ICMPv4 error messages:

  Destination Unreachable (Type 3)
     For all that are not explicitly listed below set the Type to
     1.

     Translate the code field as follows:
        Code 0, 1 (net, host unreachable):
                     Set Code to 0 (no route to destination).
        Code 2 (protocol unreachable):
               Translate to an ICMPv6 Parameter Problem (Type 4,
               Code 1) and make the Pointer point to the IPv6 Next
               Header field.

        Code 3 (port unreachable):
               Set Code to 4 (port unreachable).

        Code 4 (fragmentation needed and DF set):
               Translate to an ICMPv6 Packet Too Big message (Type
               2) with code 0.  The MTU field needs to be adjusted
               for the difference between the IPv4 and IPv6 header
               sizes.  Note that if the IPv4 router did not set
               the MTU field i.e. the router does not implement
               [PMTUv4], then the translator MUST use the plateau
               values specified in [PMTUv4] to determine a likely
               path MTU and include that path MTU in the ICMPv6
               packet. (Use the greatest plateau value that is
               less than the returned Total Length field.)

        Code 5 (source route failed):
               Set Code to 0 (no route to destination).  Note that
               this error is unlikely since source routes are not
               translated.

        Code 6,7:
               Set Code to 0 (no route to destination).

        Code 8:
               Set Code to 0 (no route to destination).

        Code 9, 10 (communication with destination host
        administratively prohibited):
               Set Code to 1 (communication with destination
               administratively prohibited)

        Code 11, 12:
               Set Code to 0 (no route to destination).
```

Identifier: RQ_003_3066
RFC Clause: 3.3
Type: Mandatory
Applies to: Router

Requirement:

In order to translate ICMPv4 error messages, for Destination Unreachable (Type 3), if the Code field is set to 6 then translate Code to 0 (no route to destination).

Specification Text:

```
ICMPv4 error messages:

  Destination Unreachable (Type 3)
     For all that are not explicitly listed below set the Type to
     1.

     Translate the code field as follows:
        Code 0, 1 (net, host unreachable):
                     Set Code to 0 (no route to destination).
        Code 2 (protocol unreachable):
               Translate to an ICMPv6 Parameter Problem (Type 4,
               Code 1) and make the Pointer point to the IPv6 Next
               Header field.

        Code 3 (port unreachable):
               Set Code to 4 (port unreachable).

        Code 4 (fragmentation needed and DF set):
               Translate to an ICMPv6 Packet Too Big message (Type
               2) with code 0.  The MTU field needs to be adjusted
               for the difference between the IPv4 and IPv6 header
               sizes.  Note that if the IPv4 router did not set
               the MTU field i.e. the router does not implement
               [PMTUv4], then the translator MUST use the plateau
               values specified in [PMTUv4] to determine a likely
               path MTU and include that path MTU in the ICMPv6
               packet. (Use the greatest plateau value that is
               less than the returned Total Length field.)

        Code 5 (source route failed):
               Set Code to 0 (no route to destination).  Note that
               this error is unlikely since source routes are not
               translated.

        Code 6,7:
               Set Code to 0 (no route to destination).

        Code 8:
               Set Code to 0 (no route to destination).

        Code 9, 10 (communication with destination host
        administratively prohibited):
               Set Code to 1 (communication with destination
               administratively prohibited)

        Code 11, 12:
               Set Code to 0 (no route to destination).

----------------
```

Identifier: RQ_003_3067
RFC Clause: 3.3
Type: Mandatory
Applies to: Router

Requirement:

In order to translate ICMPv4 error messages, for Destination Unreachable (Type 3), if the Code field is set to 7 then translate Code to 0 (no route to destination).

Specification Text:

```
ICMPv4 error messages:

  Destination Unreachable (Type 3)
     For all that are not explicitly listed below set the Type to
     1.

     Translate the code field as follows:
        Code 0, 1 (net, host unreachable):
                     Set Code to 0 (no route to destination).
        Code 2 (protocol unreachable):
               Translate to an ICMPv6 Parameter Problem (Type 4,
               Code 1) and make the Pointer point to the IPv6 Next
               Header field.
        Code 3 (port unreachable):
               Set Code to 4 (port unreachable).

        Code 4 (fragmentation needed and DF set):
               Translate to an ICMPv6 Packet Too Big message (Type
               2) with code 0.  The MTU field needs to be adjusted
               for the difference between the IPv4 and IPv6 header
               sizes.  Note that if the IPv4 router did not set
               the MTU field i.e. the router does not implement
               [PMTUv4], then the translator MUST use the plateau
               values specified in [PMTUv4] to determine a likely
               path MTU and include that path MTU in the ICMPv6
               packet. (Use the greatest plateau value that is
               less than the returned Total Length field.)

        Code 5 (source route failed):
               Set Code to 0 (no route to destination).  Note that
               this error is unlikely since source routes are not
               translated.

        Code 6,7:
               Set Code to 0 (no route to destination).

        Code 8:
               Set Code to 0 (no route to destination).

        Code 9, 10 (communication with destination host
        administratively prohibited):
               Set Code to 1 (communication with destination
               administratively prohibited)

        Code 11, 12:
               Set Code to 0 (no route to destination).

----------------
```

Identifier: RQ_003_3068
RFC Clause: 3.3
Type: Mandatory
Applies to: Router

Requirement:

In order to translate ICMPv4 error messages, for Destination Unreachable (Type 3), if the Code field is set to 8 then translate Code to 0 (no route to destination).

Specification Text:

```
ICMPv4 error messages:

  Destination Unreachable (Type 3)
     For all that are not explicitly listed below set the Type to
     1.

     Translate the code field as follows:
        Code 0, 1 (net, host unreachable):
                     Set Code to 0 (no route to destination).
        Code 2 (protocol unreachable):
               Translate to an ICMPv6 Parameter Problem (Type 4,
               Code 1) and make the Pointer point to the IPv6 Next
               Header field.

        Code 3 (port unreachable):
               Set Code to 4 (port unreachable).

        Code 4 (fragmentation needed and DF set):
               Translate to an ICMPv6 Packet Too Big message (Type
               2) with code 0.  The MTU field needs to be adjusted
               for the difference between the IPv4 and IPv6 header
               sizes.  Note that if the IPv4 router did not set
               the MTU field i.e. the router does not implement
               [PMTUv4], then the translator MUST use the plateau
               values specified in [PMTUv4] to determine a likely
               path MTU and include that path MTU in the ICMPv6
               packet. (Use the greatest plateau value that is
               less than the returned Total Length field.)
        Code 5 (source route failed):
               Set Code to 0 (no route to destination).  Note that
               this error is unlikely since source routes are not
               translated.

        Code 6,7:
               Set Code to 0 (no route to destination).

        Code 8:
               Set Code to 0 (no route to destination).

        Code 9, 10 (communication with destination host
        administratively prohibited):
               Set Code to 1 (communication with destination
               administratively prohibited)

        Code 11, 12:
               Set Code to 0 (no route to destination).
```

Identifier: RQ_003_3069
RFC Clause: 3.3
Type: Mandatory
Applies to: Router

Requirement:

In order to translate ICMPv4 error messages, for Destination Unreachable (Type 3), if the Code field is set to 11 then translate Code to 0 (no route to destination).

Specification Text:

```
ICMPv4 error messages:

  Destination Unreachable (Type 3)
     For all that are not explicitly listed below set the Type to
     1.

     Translate the code field as follows:
        Code 0, 1 (net, host unreachable):
                     Set Code to 0 (no route to destination).
        Code 2 (protocol unreachable):
               Translate to an ICMPv6 Parameter Problem (Type 4,
               Code 1) and make the Pointer point to the IPv6 Next
               Header field.

        Code 3 (port unreachable):
               Set Code to 4 (port unreachable).

        Code 4 (fragmentation needed and DF set):
               Translate to an ICMPv6 Packet Too Big message (Type
               2) with code 0.  The MTU field needs to be adjusted
               for the difference between the IPv4 and IPv6 header
               sizes.  Note that if the IPv4 router did not set
               the MTU field i.e. the router does not implement
               [PMTUv4], then the translator MUST use the plateau
               values specified in [PMTUv4] to determine a likely
               path MTU and include that path MTU in the ICMPv6
               packet. (Use the greatest plateau value that is
               less than the returned Total Length field.)

        Code 5 (source route failed):
               Set Code to 0 (no route to destination).  Note that
               this error is unlikely since source routes are not
               translated.

        Code 6,7:
               Set Code to 0 (no route to destination).

        Code 8:
               Set Code to 0 (no route to destination).

        Code 9, 10 (communication with destination host
        administratively prohibited):
               Set Code to 1 (communication with destination
               administratively prohibited)
```

```
      Code 11, 12:
             Set Code to 0 (no route to destination).
```

Identifier: RQ_003_3070
RFC Clause: 3.3
Type: Mandatory
Applies to: Router

Requirement:

In order to translate ICMPv4 error messages, for Destination Unreachable (Type 3), if the Code field is set to 12 then translate Code to 0 (no route to destination).

Specification Text:

```
ICMPv4 error messages:

  Destination Unreachable (Type 3)
     For all that are not explicitly listed below set the Type to
     1.

     Translate the code field as follows:
        Code 0, 1 (net, host unreachable):
                     Set Code to 0 (no route to destination).
        Code 2 (protocol unreachable):
               Translate to an ICMPv6 Parameter Problem (Type 4,
               Code 1) and make the Pointer point to the IPv6 Next
               Header field.

        Code 3 (port unreachable):
               Set Code to 4 (port unreachable).

        Code 4 (fragmentation needed and DF set):
               Translate to an ICMPv6 Packet Too Big message (Type
               2) with code 0.  The MTU field needs to be adjusted
               for the difference between the IPv4 and IPv6 header
               sizes.  Note that if the IPv4 router did not set
               the MTU field i.e. the router does not implement
               [PMTUv4], then the translator MUST use the plateau
               values specified in [PMTUv4] to determine a likely
               path MTU and include that path MTU in the ICMPv6
               packet. (Use the greatest plateau value that is
               less than the returned Total Length field.)

        Code 5 (source route failed):
               Set Code to 0 (no route to destination).  Note that
               this error is unlikely since source routes are not
               translated.

        Code 6,7:
               Set Code to 0 (no route to destination).

        Code 8:
               Set Code to 0 (no route to destination).

        Code 9, 10 (communication with destination host
        administratively prohibited):
               Set Code to 1 (communication with destination
               administratively prohibited)

        Code 11, 12:
               Set Code to 0 (no route to destination).
```

Identifier: RQ_003_3063
RFC Clause: 3.5
Type: Mandatory
Applies to: Router

Requirement:

If the IPv4 destination field contains an address that falls in the pool(s) of IPv4 address that are used to represent the internal IPv6-only nodes, the packet needs to be translated to IPv6.

Specification Text:

The translator is assumed to know the pool(s) of IPv4 address that are used to represent the internal IPv6-only nodes. Thus if the IPv4 destination field contains an address that falls in these configured sets of prefixes the packet needs to be translated to IPv6.

Identifier: RQ_003_3071
RFC Clause: 4.1
Type: Mandatory
Applies to: Router

Requirement:

When translating the IPv6 packet to IPv4, the resulting IPv4 header's version field SHALL be set to 4.

Specification Text:

If there is no IPv6 Fragment header the IPv4 header fields are set as follows:

Version:
4

Internet Header Length:
5 (no IPv4 options)

Type of Service and Precedence:
By default, copied from the IPv6 Traffic Class (all 8 bits). According to [DIFFSERV] the semantics of the bits are identical in IPv4 and IPv6. However, in some IPv4 environments these bits might be used with the old semantics of "Type Of Service and Precedence". An implementation of a translator SHOULD provide the ability to ignore the IPv6 traffic class and always set the IPv4 "TOS" to zero.

Total Length:
Payload length value from IPv6 header, plus the size of the IPv4 header.

Identification:
All zero.

Flags:
The More Fragments flag is set to zero. The Don't Fragments flag is set to one.

Fragment Offset:
All zero.
Time to Live:
Hop Limit value copied from IPv6 header. Since the translator is a router, as part of forwarding the packet it needs to decrement either the IPv6 Hop Limit (before the translation) or the IPv4 TTL (after the translation). As part of decrementing the TTL or Hop Limit the translator (as any router) needs to check for zero and send the ICMPv4 or ICMPv6 "ttl exceeded" error.

Protocol:
Next Header field copied from IPv6 header.

Header Checksum:
Computed once the IPv4 header has been created.

Source Address:
If the IPv6 source address is an IPv4-translated address then the low-order 32 bits of the IPv6 source address is copied to the IPv4 source address. Otherwise, the source address is set to 0.0.0.0. The use of 0.0.0.0 is to avoid completely dropping e.g. ICMPv6 error messages sent by IPv6-only routers which makes e.g. traceroute present something for the IPv6-only hops.

Destination Address:

IPv6 packets that are translated have an IPv4-mapped destination address. Thus the low-order 32 bits of the IPv6 destination address is copied to the IPv4 destination address.

Identifier: RQ_003_3072
RFC Clause: 4.1
Type: Mandatory
Applies to: Router

Requirement:

When translating the IPv6 packet to IPv4, the resulting IPv4 header's Internet Header Length field SHALL be set to 5.

Specification Text:

If there is no IPv6 Fragment header the IPv4 header fields are set as follows:

Version:

4

Internet Header Length:

5 (no IPv4 options)

Type of Service and Precedence:

By default, copied from the IPv6 Traffic Class (all 8 bits). According to [DIFFSERV] the semantics of the bits are identical in IPv4 and IPv6. However, in some IPv4 environments these bits might be used with the old semantics of "Type Of Service and Precedence". An implementation of a translator SHOULD provide the ability to ignore the IPv6 traffic class and always set the IPv4 "TOS" to zero.

Total Length:

Payload length value from IPv6 header, plus the size of the IPv4 header.

Identification:

All zero.

Flags:

The More Fragments flag is set to zero. The Don't Fragments flag is set to one.

Fragment Offset:

All zero.

Time to Live:

Hop Limit value copied from IPv6 header. Since the translator is a router, as part of forwarding the packet it needs to decrement either the IPv6 Hop Limit (before the translation) or the IPv4 TTL (after the translation). As part of decrementing the TTL or Hop Limit the translator (as any router) needs to check for zero and send the ICMPv4 or ICMPv6 "ttl exceeded" error.

Protocol:

Next Header field copied from IPv6 header.

Header Checksum:

Computed once the IPv4 header has been created.

Source Address:

If the IPv6 source address is an IPv4-translated address then the low-order 32 bits of the IPv6 source address is copied to the IPv4 source address. Otherwise, the source address is set to 0.0.0.0. The use of 0.0.0.0 is to avoid completely dropping e.g. ICMPv6 error messages sent by IPv6-only routers which makes e.g. traceroute present something for the IPv6-only hops.

```
        Destination Address:
                IPv6 packets that are translated have an IPv4-mapped
                destination address.  Thus the low-order 32 bits of
                the IPv6 destination address is copied to the IPv4
                destination address.
```

Identifier: RQ_003_3073
RFC Clause: 4.1
Type: Mandatory
Applies to: Router

Requirement:
When translating the IPv6 packet to IPv4, the resulting IPv4 header's Type of Service and Precedence field SHALL, by default, be copied from the IPv6 Traffic Class (all 8 bits).

Specification Text:

If there is no IPv6 Fragment header the IPv4 header fields are set as follows:

```
        Version:
                4

        Internet Header Length:
                5 (no IPv4 options)

        Type of Service and Precedence:
                By default, copied from the IPv6 Traffic Class (all 8
                bits).  According to [DIFFSERV] the semantics of the
                bits are identical in IPv4 and IPv6.  However, in
                some IPv4 environments these bits might be used with
                the old semantics of "Type Of Service and
                Precedence".  An implementation of a translator
                SHOULD provide the ability to ignore the IPv6 traffic
                class and always set the IPv4 "TOS" to zero.

        Total Length:
                Payload length value from IPv6 header, plus the size
                of the IPv4 header.

        Identification:
                All zero.

        Flags:
                The More Fragments flag is set to zero.  The Don't
                Fragments flag is set to one.

        Fragment Offset:
                All zero.
        Time to Live:
                Hop Limit value copied from IPv6 header.  Since the
                translator is a router, as part of forwarding the
                packet it needs to decrement either the IPv6 Hop
                Limit (before the translation) or the IPv4 TTL (after
                the translation).  As part of decrementing the TTL or
                Hop Limit the translator (as any router) needs to
                check for zero and send the ICMPv4 or ICMPv6 "ttl
                exceeded" error.

        Protocol:
                Next Header field copied from IPv6 header.

        Header Checksum:
                Computed once the IPv4 header has been created.

        Source Address:
                If the IPv6 source address is an IPv4-translated
                address then the low-order 32 bits of the IPv6 source
                address is copied to the IPv4 source address.
                Otherwise, the source address is set to 0.0.0.0.  The
                use of 0.0.0.0 is to avoid completely dropping e.g.
                ICMPv6 error messages sent by IPv6-only routers which
                makes e.g. traceroute present something for the
                IPv6-only hops.
```

Destination Address:

IPv6 packets that are translated have an IPv4-mapped destination address. Thus the low-order 32 bits of the IPv6 destination address is copied to the IPv4 destination address.

Identifier: RQ_003_3074
RFC Clause: 4.1
Type: Optional
Applies to: Router

Requirement:

Although, when translating the IPv6 packet to IPv4, the resulting IPv4 header's Type of Service and Precedence field is, by default, copied from the IPv6 Traffic Class (all 8 bits), an implementation of a translator SHOULD provide the ability to ignore the IPv6 traffic class and always set the IPv4 "TOS" to zero.

Specification Text:

If there is no IPv6 Fragment header the IPv4 header fields are set as follows:

Version:

4

Internet Header Length:

5 (no IPv4 options)

Type of Service and Precedence:

By default, copied from the IPv6 Traffic Class (all 8 bits). According to [DIFFSERV] the semantics of the bits are identical in IPv4 and IPv6. However, in some IPv4 environments these bits might be used with the old semantics of "Type Of Service and Precedence". An implementation of a translator SHOULD provide the ability to ignore the IPv6 traffic class and always set the IPv4 "TOS" to zero.

Total Length:

Payload length value from IPv6 header, plus the size of the IPv4 header.

Identification:

All zero.

Flags:

The More Fragments flag is set to zero. The Don't Fragments flag is set to one.

Fragment Offset:

All zero.

Time to Live:

Hop Limit value copied from IPv6 header. Since the translator is a router, as part of forwarding the packet it needs to decrement either the IPv6 Hop Limit (before the translation) or the IPv4 TTL (after the translation). As part of decrementing the TTL or Hop Limit the translator (as any router) needs to check for zero and send the ICMPv4 or ICMPv6 "ttl exceeded" error.

Protocol:

Next Header field copied from IPv6 header.

Header Checksum:

Computed once the IPv4 header has been created.

Source Address:

If the IPv6 source address is an IPv4-translated address then the low-order 32 bits of the IPv6 source address is copied to the IPv4 source address. Otherwise, the source address is set to 0.0.0.0. The use of 0.0.0.0 is to avoid completely dropping e.g. ICMPv6 error messages sent by IPv6-only routers which makes e.g. traceroute present something for the IPv6-only hops.

Destination Address:
IPv6 packets that are translated have an IPv4-mapped
destination address. Thus the low-order 32 bits of
the IPv6 destination address is copied to the IPv4
destination address.

Identifier: RQ_003_3075
RFC Clause: 4.1
Type: Mandatory
Applies to: Router

Requirement:

When translating the IPv6 packet to IPv4 and if there is no IPv6 Fragment header, the resulting IPv4 header's Total Length field SHALL be set to the Payload length value from IPv6 header, plus the size of the IPv4 header.

Specification Text:

If there is no IPv6 Fragment header the IPv4 header fields are set as follows:

Version:
4

Internet Header Length:
5 (no IPv4 options)

Type of Service and Precedence:
By default, copied from the IPv6 Traffic Class (all 8
bits). According to [DIFFSERV] the semantics of the
bits are identical in IPv4 and IPv6. However, in
some IPv4 environments these bits might be used with
the old semantics of "Type Of Service and
Precedence". An implementation of a translator
SHOULD provide the ability to ignore the IPv6 traffic
class and always set the IPv4 "TOS" to zero.

Total Length:
Payload length value from IPv6 header, plus the size of the IPv4 header.

Identification:
All zero.

Flags:
The More Fragments flag is set to zero. The Don't
Fragments flag is set to one.

Fragment Offset:
All zero.
Time to Live:
Hop Limit value copied from IPv6 header. Since the
translator is a router, as part of forwarding the
packet it needs to decrement either the IPv6 Hop
Limit (before the translation) or the IPv4 TTL (after
the translation). As part of decrementing the TTL or
Hop Limit the translator (as any router) needs to
check for zero and send the ICMPv4 or ICMPv6 "ttl
exceeded" error.

Protocol:
Next Header field copied from IPv6 header.

Header Checksum:
Computed once the IPv4 header has been created.

Source Address:
If the IPv6 source address is an IPv4-translated
address then the low-order 32 bits of the IPv6 source
address is copied to the IPv4 source address.
Otherwise, the source address is set to 0.0.0.0. The
use of 0.0.0.0 is to avoid completely dropping e.g.
ICMPv6 error messages sent by IPv6-only routers which
makes e.g. traceroute present something for the
IPv6-only hops.

```
        Destination Address:
                IPv6 packets that are translated have an IPv4-mapped
                destination address.  Thus the low-order 32 bits of
                the IPv6 destination address is copied to the IPv4
                destination address.
```

Identifier: RQ_003_3076
RFC Clause: 4.1
Type: Mandatory
Applies to: Router

Requirement:
When translating the IPv6 packet to IPv4 and if there is no IPv6 Fragment header, the resulting IPv4 header's Identification field SHALL be set to All zero.

Specification Text:

```
   If there is no IPv6 Fragment header the IPv4 header fields are set as
   follows:

         Version:
                 4

         Internet Header Length:
                 5 (no IPv4 options)

         Type of Service and Precedence:
                 By default, copied from the IPv6 Traffic Class (all 8
                 bits).  According to [DIFFSERV] the semantics of the
                 bits are identical in IPv4 and IPv6.  However, in
                 some IPv4 environments these bits might be used with
                 the old semantics of "Type Of Service and
                 Precedence".  An implementation of a translator
                 SHOULD provide the ability to ignore the IPv6 traffic
                 class and always set the IPv4 "TOS" to zero.

         Total Length:
                 Payload length value from IPv6 header, plus the size
                 of the IPv4 header.

         Identification:
                 All zero.

         Flags:
                 The More Fragments flag is set to zero.  The Don't
                 Fragments flag is set to one.

         Fragment Offset:
                 All zero.
         Time to Live:
                 Hop Limit value copied from IPv6 header.  Since the
                 translator is a router, as part of forwarding the
                 packet it needs to decrement either the IPv6 Hop
                 Limit (before the translation) or the IPv4 TTL (after
                 the translation).  As part of decrementing the TTL or
                 Hop Limit the translator (as any router) needs to
                 check for zero and send the ICMPv4 or ICMPv6 "ttl
                 exceeded" error.

         Protocol:
                 Next Header field copied from IPv6 header.

         Header Checksum:
                 Computed once the IPv4 header has been created.

         Source Address:
                 If the IPv6 source address is an IPv4-translated
                 address then the low-order 32 bits of the IPv6 source
                 address is copied to the IPv4 source address.
                 Otherwise, the source address is set to 0.0.0.0.  The
                 use of 0.0.0.0 is to avoid completely dropping e.g.
                 ICMPv6 error messages sent by IPv6-only routers which
                 makes e.g. traceroute present something for the
                 IPv6-only hops.
```

Destination Address:
IPv6 packets that are translated have an IPv4-mapped destination address. Thus the low-order 32 bits of the IPv6 destination address is copied to the IPv4 destination address.

Identifier: RQ_003_3077
RFC Clause: 4.1
Type: Mandatory
Applies to: Router

Requirement:
When translating the IPv6 packet to IPv4 and if there is no IPv6 Fragment header, the resulting IPv4 header's More Fragments flag field SHALL be set to zero.

Specification Text:
If there is no IPv6 Fragment header the IPv4 header fields are set as follows:

Version:
4

Internet Header Length:
5 (no IPv4 options)

Type of Service and Precedence:
By default, copied from the IPv6 Traffic Class (all 8 bits). According to [DIFFSERV] the semantics of the bits are identical in IPv4 and IPv6. However, in some IPv4 environments these bits might be used with the old semantics of "Type Of Service and Precedence". An implementation of a translator SHOULD provide the ability to ignore the IPv6 traffic class and always set the IPv4 "TOS" to zero.

Total Length:
Payload length value from IPv6 header, plus the size of the IPv4 header.

Identification:
All zero.

Flags:
The More Fragments flag is set to zero. The Don't Fragments flag is set to one.

Fragment Offset:
All zero.
Time to Live:
Hop Limit value copied from IPv6 header. Since the translator is a router, as part of forwarding the packet it needs to decrement either the IPv6 Hop Limit (before the translation) or the IPv4 TTL (after the translation). As part of decrementing the TTL or Hop Limit the translator (as any router) needs to check for zero and send the ICMPv4 or ICMPv6 "ttl exceeded" error.

Protocol:
Next Header field copied from IPv6 header.

Header Checksum:
Computed once the IPv4 header has been created.

Source Address:
If the IPv6 source address is an IPv4-translated address then the low-order 32 bits of the IPv6 source address is copied to the IPv4 source address. Otherwise, the source address is set to 0.0.0.0. The use of 0.0.0.0 is to avoid completely dropping e.g. ICMPv6 error messages sent by IPv6-only routers which makes e.g. traceroute present something for the IPv6-only hops.

Destination Address:

IPv6 packets that are translated have an IPv4-mapped destination address. Thus the low-order 32 bits of the IPv6 destination address is copied to the IPv4 destination address.

Identifier: RQ_003_3078
RFC Clause: 4.1
Type: Mandatory
Applies to: Router

Requirement:

When translating the IPv6 packet to IPv4 and if there is no IPv6 Fragment header, the resulting IPv4 header's Don't Fragments flag field SHALL be set to one.

Specification Text:

If there is no IPv6 Fragment header the IPv4 header fields are set as follows:

Version:

4

Internet Header Length:

5 (no IPv4 options)

Type of Service and Precedence:

By default, copied from the IPv6 Traffic Class (all 8 bits). According to [DIFFSERV] the semantics of the bits are identical in IPv4 and IPv6. However, in some IPv4 environments these bits might be used with the old semantics of "Type Of Service and Precedence". An implementation of a translator SHOULD provide the ability to ignore the IPv6 traffic class and always set the IPv4 "TOS" to zero.

Total Length:

Payload length value from IPv6 header, plus the size of the IPv4 header.

Identification:

All zero.

Flags:

The More Fragments flag is set to zero. **The Don't Fragments flag is set to one.**

Fragment Offset:

All zero.

Time to Live:

Hop Limit value copied from IPv6 header. Since the translator is a router, as part of forwarding the packet it needs to decrement either the IPv6 Hop Limit (before the translation) or the IPv4 TTL (after the translation). As part of decrementing the TTL or Hop Limit the translator (as any router) needs to check for zero and send the ICMPv4 or ICMPv6 "ttl exceeded" error.

Protocol:

Next Header field copied from IPv6 header.

Header Checksum:

Computed once the IPv4 header has been created.

Source Address:

If the IPv6 source address is an IPv4-translated address then the low-order 32 bits of the IPv6 source address is copied to the IPv4 source address. Otherwise, the source address is set to 0.0.0.0. The use of 0.0.0.0 is to avoid completely dropping e.g. ICMPv6 error messages sent by IPv6-only routers which makes e.g. traceroute present something for the IPv6-only hops.

```
        Destination Address:
                IPv6 packets that are translated have an IPv4-mapped
                destination address.  Thus the low-order 32 bits of
                the IPv6 destination address is copied to the IPv4
                destination address.
```

Identifier: RQ_003_3079
RFC Clause: 4.1
Type: Mandatory
Applies to: Router

Requirement:
When translating the IPv6 packet to IPv4 and if there is no IPv6 Fragment header, the resulting IPv4 header's Fragment Offset field SHALL be set to all zero.

Specification Text:

```
If there is no IPv6 Fragment header the IPv4 header fields are set as
follows:

      Version:
              4

      Internet Header Length:
              5 (no IPv4 options)

      Type of Service and Precedence:
              By default, copied from the IPv6 Traffic Class (all 8
              bits).  According to [DIFFSERV] the semantics of the
              bits are identical in IPv4 and IPv6.  However, in
              some IPv4 environments these bits might be used with
              the old semantics of "Type Of Service and
              Precedence".  An implementation of a translator
              SHOULD provide the ability to ignore the IPv6 traffic
              class and always set the IPv4 "TOS" to zero.

      Total Length:
              Payload length value from IPv6 header, plus the size
              of the IPv4 header.

      Identification:
              All zero.

      Flags:
              The More Fragments flag is set to zero.  The Don't
              Fragments flag is set to one.

      Fragment Offset:
              All zero.
    Time to Live:
              Hop Limit value copied from IPv6 header.  Since the
              translator is a router, as part of forwarding the
              packet it needs to decrement either the IPv6 Hop
              Limit (before the translation) or the IPv4 TTL (after
              the translation).  As part of decrementing the TTL or
              Hop Limit the translator (as any router) needs to
              check for zero and send the ICMPv4 or ICMPv6 "ttl
              exceeded" error.

      Protocol:
              Next Header field copied from IPv6 header.

      Header Checksum:
              Computed once the IPv4 header has been created.

      Source Address:
              If the IPv6 source address is an IPv4-translated
              address then the low-order 32 bits of the IPv6 source
              address is copied to the IPv4 source address.
              Otherwise, the source address is set to 0.0.0.0.  The
              use of 0.0.0.0 is to avoid completely dropping e.g.
              ICMPv6 error messages sent by IPv6-only routers which
              makes e.g. traceroute present something for the
              IPv6-only hops.
```

Destination Address:

IPv6 packets that are translated have an IPv4-mapped destination address. Thus the low-order 32 bits of the IPv6 destination address is copied to the IPv4 destination address.

Identifier: RQ_003_3080
RFC Clause: 4.1
Type: Mandatory
Applies to: Router

Requirement:

When translating the IPv6 packet to IPv4, the resulting IPv4 header's Time to Live field SHALL be set to the Hop Limit value copied from IPv6 header.

Specification Text:

If there is no IPv6 Fragment header the IPv4 header fields are set as follows:

Version:

4

Internet Header Length:

5 (no IPv4 options)

Type of Service and Precedence:

By default, copied from the IPv6 Traffic Class (all 8 bits). According to [DIFFSERV] the semantics of the bits are identical in IPv4 and IPv6. However, in some IPv4 environments these bits might be used with the old semantics of "Type Of Service and Precedence". An implementation of a translator SHOULD provide the ability to ignore the IPv6 traffic class and always set the IPv4 "TOS" to zero.

Total Length:

Payload length value from IPv6 header, plus the size of the IPv4 header.

Identification:

All zero.

Flags:

The More Fragments flag is set to zero. The Don't Fragments flag is set to one.

Fragment Offset:

All zero.

Time to Live:

Hop Limit value copied from IPv6 header. Since the translator is a router, as part of forwarding the packet it needs to decrement either the IPv6 Hop Limit (before the translation) or the IPv4 TTL (after the translation). As part of decrementing the TTL or Hop Limit the translator (as any router) needs to check for zero and send the ICMPv4 or ICMPv6 "ttl exceeded" error.

Protocol:

Next Header field copied from IPv6 header.

Header Checksum:

Computed once the IPv4 header has been created.

Source Address:

If the IPv6 source address is an IPv4-translated address then the low-order 32 bits of the IPv6 source address is copied to the IPv4 source address. Otherwise, the source address is set to 0.0.0.0. The use of 0.0.0.0 is to avoid completely dropping e.g. ICMPv6 error messages sent by IPv6-only routers which makes e.g. traceroute present something for the IPv6-only hops.

Destination Address:
IPv6 packets that are translated have an IPv4-mapped destination address. Thus the low-order 32 bits of the IPv6 destination address is copied to the IPv4 destination address.

Identifier: RQ_003_3081
RFC Clause: 4.1
Type: Mandatory
Applies to: Router

Requirement:
As part of forwarding the packet if the translator has not decremented the IPv6 Hop Limit (before the translation) it SHALL decrement the IPv4 TTL (after the translation).

Specification Text:
If there is no IPv6 Fragment header the IPv4 header fields are set as follows:

Version:
4

Internet Header Length:
5 (no IPv4 options)

Type of Service and Precedence:
By default, copied from the IPv6 Traffic Class (all 8 bits). According to [DIFFSERV] the semantics of the bits are identical in IPv4 and IPv6. However, in some IPv4 environments these bits might be used with the old semantics of "Type Of Service and Precedence". An implementation of a translator SHOULD provide the ability to ignore the IPv6 traffic class and always set the IPv4 "TOS" to zero.

Total Length:
Payload length value from IPv6 header, plus the size of the IPv4 header.

Identification:
All zero.

Flags:
The More Fragments flag is set to zero. The Don't Fragments flag is set to one.

Fragment Offset:
All zero.

Time to Live:
Hop Limit value copied from IPv6 header. **Since the translator is a router, as part of forwarding the packet it needs to decrement either the IPv6 Hop Limit (before the translation) or the IPv4 TTL (after the translation).** As part of decrementing the TTL or Hop Limit the translator (as any router) needs to check for zero and send the ICMPv4 or ICMPv6 "ttl exceeded" error.

Protocol:
Next Header field copied from IPv6 header.

Header Checksum:
Computed once the IPv4 header has been created.

Source Address:
If the IPv6 source address is an IPv4-translated address then the low-order 32 bits of the IPv6 source address is copied to the IPv4 source address. Otherwise, the source address is set to 0.0.0.0. The use of 0.0.0.0 is to avoid completely dropping e.g. ICMPv6 error messages sent by IPv6-only routers which makes e.g. traceroute present something for the IPv6-only hops.

```
        Destination Address:
                IPv6 packets that are translated have an IPv4-mapped
                destination address.  Thus the low-order 32 bits of
                the IPv6 destination address is copied to the IPv4
                destination address.
```

Identifier: RQ_003_3082
RFC Clause: 4.1
Type: Mandatory
Applies to: Router

Requirement:

As part of forwarding the packet if the translator has decremented the IPv6 Hop Limit (before the translation) it SHALL NOT decrement the IPv4 TTL (after the translation).

Specification Text:

If there is no IPv6 Fragment header the IPv4 header fields are set as follows:

```
        Version:
                4

        Internet Header Length:
                5 (no IPv4 options)

        Type of Service and Precedence:
                By default, copied from the IPv6 Traffic Class (all 8
                bits).  According to [DIFFSERV] the semantics of the
                bits are identical in IPv4 and IPv6.  However, in
                some IPv4 environments these bits might be used with
                the old semantics of "Type Of Service and
                Precedence".  An implementation of a translator
                SHOULD provide the ability to ignore the IPv6 traffic
                class and always set the IPv4 "TOS" to zero.

        Total Length:
                Payload length value from IPv6 header, plus the size
                of the IPv4 header.

        Identification:
                All zero.

        Flags:
                The More Fragments flag is set to zero.  The Don't
                Fragments flag is set to one.

        Fragment Offset:
                All zero.

      Time to Live:
                Hop Limit value copied from IPv6 header.  Since the
                translator is a router, as part of forwarding the
                packet it needs to decrement either the IPv6 Hop
                Limit (before the translation) or the IPv4 TTL (after
                the translation).  As part of decrementing the TTL or
                Hop Limit the translator (as any router) needs to
                check for zero and send the ICMPv4 or ICMPv6 "ttl
                exceeded" error.

        Protocol:
                Next Header field copied from IPv6 header.

        Header Checksum:
                Computed once the IPv4 header has been created.

        Source Address:
                If the IPv6 source address is an IPv4-translated
                address then the low-order 32 bits of the IPv6 source
                address is copied to the IPv4 source address.
                Otherwise, the source address is set to 0.0.0.0.  The
                use of 0.0.0.0 is to avoid completely dropping e.g.
                ICMPv6 error messages sent by IPv6-only routers which
                makes e.g. traceroute present something for the
                IPv6-only hops.
```

Destination Address:
IPv6 packets that are translated have an IPv4-mapped destination address. Thus the low-order 32 bits of the IPv6 destination address is copied to the IPv4 destination address.

Identifier: RQ_003_3083
RFC Clause: 4.1
Type: Mandatory
Applies to: Router

Requirement:

As part of decrementing the Hop Limit the translator needs to check for zero and if present, send the ICMPv6 "ttl exceeded" error.

Specification Text:

If there is no IPv6 Fragment header the IPv4 header fields are set as follows:

Version:
4

Internet Header Length:
5 (no IPv4 options)

Type of Service and Precedence:
By default, copied from the IPv6 Traffic Class (all 8 bits). According to [DIFFSERV] the semantics of the bits are identical in IPv4 and IPv6. However, in some IPv4 environments these bits might be used with the old semantics of "Type Of Service and Precedence". An implementation of a translator SHOULD provide the ability to ignore the IPv6 traffic class and always set the IPv4 "TOS" to zero.

Total Length:
Payload length value from IPv6 header, plus the size of the IPv4 header.

Identification:
All zero.

Flags:
The More Fragments flag is set to zero. The Don't Fragments flag is set to one.

Fragment Offset:
All zero.

Time to Live:
Hop Limit value copied from IPv6 header. Since the translator is a router, as part of forwarding the packet it needs to decrement either the IPv6 Hop Limit (before the translation) or the IPv4 TTL (after the translation). **As part of decrementing the TTL or Hop Limit the translator (as any router) needs to check for zero and send the ICMPv4 or ICMPv6 "ttl exceeded" error.**

Protocol:
Next Header field copied from IPv6 header.

Header Checksum:
Computed once the IPv4 header has been created.

Source Address:
If the IPv6 source address is an IPv4-translated address then the low-order 32 bits of the IPv6 source address is copied to the IPv4 source address. Otherwise, the source address is set to 0.0.0.0. The use of 0.0.0.0 is to avoid completely dropping e.g. ICMPv6 error messages sent by IPv6-only routers which makes e.g. traceroute present something for the IPv6-only hops.

```
        Destination Address:
                IPv6 packets that are translated have an IPv4-mapped
                destination address.  Thus the low-order 32 bits of
                the IPv6 destination address is copied to the IPv4
                destination address.
```

Identifier: RQ_003_3084
RFC Clause: 4.1
Type: Mandatory
Applies to: Router

Requirement:
When translating the IPv6 packet to IPv4 and if there is no IPv6 Fragment header, the resulting IPv4 header's Protocol field SHALL be set to the Next Header field copied from IPv6 header.

Specification Text:
If there is no IPv6 Fragment header the IPv4 header fields are set as follows:

```
        Version:
                4

        Internet Header Length:
                5 (no IPv4 options)

        Type of Service and Precedence:
                By default, copied from the IPv6 Traffic Class (all 8
                bits).  According to [DIFFSERV] the semantics of the
                bits are identical in IPv4 and IPv6.  However, in
                some IPv4 environments these bits might be used with
                the old semantics of "Type Of Service and
                Precedence".  An implementation of a translator
                SHOULD provide the ability to ignore the IPv6 traffic
                class and always set the IPv4 "TOS" to zero.

        Total Length:
                Payload length value from IPv6 header, plus the size
                of the IPv4 header.

        Identification:
                All zero.

        Flags:
                The More Fragments flag is set to zero.  The Don't
                Fragments flag is set to one.

        Fragment Offset:
                All zero.
      Time to Live:
                Hop Limit value copied from IPv6 header.  Since the
                translator is a router, as part of forwarding the
                packet it needs to decrement either the IPv6 Hop
                Limit (before the translation) or the IPv4 TTL (after
                the translation).  As part of decrementing the TTL or
                Hop Limit the translator (as any router) needs to
                check for zero and send the ICMPv4 or ICMPv6 "ttl
                exceeded" error.
```

Protocol:
Next Header field copied from IPv6 header.

```
        Header Checksum:
                Computed once the IPv4 header has been created.

        Source Address:
                If the IPv6 source address is an IPv4-translated
                address then the low-order 32 bits of the IPv6 source
                address is copied to the IPv4 source address.
                Otherwise, the source address is set to 0.0.0.0.  The
                use of 0.0.0.0 is to avoid completely dropping e.g.
                ICMPv6 error messages sent by IPv6-only routers which
                makes e.g. traceroute present something for the
                IPv6-only hops.
        Destination Address:
                IPv6 packets that are translated have an IPv4-mapped
                destination address.  Thus the low-order 32 bits of
                the IPv6 destination address is copied to the IPv4
                destination address.
```

Identifier: RQ_003_3085
RFC Clause: 4.1
Type: Mandatory
Applies to: Router

Requirement:

When translating the IPv6 packet to IPv4, the resulting IPv4 header's Header Checksum field SHALL be computed once the IPv4 header has been created.

Specification Text:

```
   If there is no IPv6 Fragment header the IPv4 header fields are set as
   follows:

        Version:
                4

        Internet Header Length:
                5 (no IPv4 options)

        Type of Service and Precedence:
                By default, copied from the IPv6 Traffic Class (all 8
                bits).  According to [DIFFSERV] the semantics of the
                bits are identical in IPv4 and IPv6.  However, in
                some IPv4 environments these bits might be used with
                the old semantics of "Type Of Service and
                Precedence".  An implementation of a translator
                SHOULD provide the ability to ignore the IPv6 traffic
                class and always set the IPv4 "TOS" to zero.

        Total Length:
                Payload length value from IPv6 header, plus the size
                of the IPv4 header.

        Identification:
                All zero.

        Flags:
                The More Fragments flag is set to zero.  The Don't
                Fragments flag is set to one.

        Fragment Offset:
                All zero.
      Time to Live:
                Hop Limit value copied from IPv6 header.  Since the
                translator is a router, as part of forwarding the
                packet it needs to decrement either the IPv6 Hop
                Limit (before the translation) or the IPv4 TTL (after
                the translation).  As part of decrementing the TTL or
                Hop Limit the translator (as any router) needs to
                check for zero and send the ICMPv4 or ICMPv6 "ttl
                exceeded" error.

        Protocol:
                Next Header field copied from IPv6 header.

        Header Checksum:
                Computed once the IPv4 header has been created.

        Source Address:
                If the IPv6 source address is an IPv4-translated
                address then the low-order 32 bits of the IPv6 source
                address is copied to the IPv4 source address.
                Otherwise, the source address is set to 0.0.0.0.  The
                use of 0.0.0.0 is to avoid completely dropping e.g.
                ICMPv6 error messages sent by IPv6-only routers which
                makes e.g. traceroute present something for the
                IPv6-only hops.
```

Destination Address:

IPv6 packets that are translated have an IPv4-mapped destination address. Thus the low-order 32 bits of the IPv6 destination address is copied to the IPv4 destination address.

Identifier: RQ_003_3086
RFC Clause: 4.1
Type: Mandatory
Applies to: Router

Requirement:

When translating the IPv6 packet to IPv4, the resulting IPv4 header's Source Address field, if the IPv6 source address is an IPv4-translated address, SHALL be the low-order 32 bits of the IPv6 source address is copied to the IPv4 source address.

Specification Text:

If there is no IPv6 Fragment header the IPv4 header fields are set as follows:

Version:

4

Internet Header Length:

5 (no IPv4 options)

Type of Service and Precedence:

By default, copied from the IPv6 Traffic Class (all 8 bits). According to [DIFFSERV] the semantics of the bits are identical in IPv4 and IPv6. However, in some IPv4 environments these bits might be used with the old semantics of "Type Of Service and Precedence". An implementation of a translator SHOULD provide the ability to ignore the IPv6 traffic class and always set the IPv4 "TOS" to zero.

Total Length:

Payload length value from IPv6 header, plus the size of the IPv4 header.

Identification:

All zero.

Flags:

The More Fragments flag is set to zero. The Don't Fragments flag is set to one.

Fragment Offset:

All zero.

Time to Live:

Hop Limit value copied from IPv6 header. Since the translator is a router, as part of forwarding the packet it needs to decrement either the IPv6 Hop Limit (before the translation) or the IPv4 TTL (after the translation). As part of decrementing the TTL or Hop Limit the translator (as any router) needs to check for zero and send the ICMPv4 or ICMPv6 "ttl exceeded" error.

Protocol:

Next Header field copied from IPv6 header.

Header Checksum:

Computed once the IPv4 header has been created.

Source Address:

If the IPv6 source address is an IPv4-translated address then the low-order 32 bits of the IPv6 source address is copied to the IPv4 source address. Otherwise, the source address is set to 0.0.0.0. The use of 0.0.0.0 is to avoid completely dropping e.g. ICMPv6 error messages sent by IPv6-only routers which makes e.g. traceroute present something for the IPv6-only hops.

```
        Destination Address:
                IPv6 packets that are translated have an IPv4-mapped
                destination address.  Thus the low-order 32 bits of
                the IPv6 destination address is copied to the IPv4
                destination address.
```

Identifier: RQ_003_3087
RFC Clause: 4.1
Type: Mandatory
Applies to: Router

Requirement:

When translating the IPv6 packet to IPv4, the resulting IPv4 header's Source Address field, if the IPv6 source address is NOT an IPv4-translated address, SHALL be set to 0.0.0.0.

Specification Text:

If there is no IPv6 Fragment header the IPv4 header fields are set as follows:

```
        Version:
                4

        Internet Header Length:
                5 (no IPv4 options)

        Type of Service and Precedence:
                By default, copied from the IPv6 Traffic Class (all 8
                bits).  According to [DIFFSERV] the semantics of the
                bits are identical in IPv4 and IPv6.  However, in
                some IPv4 environments these bits might be used with
                the old semantics of "Type Of Service and
                Precedence".  An implementation of a translator
                SHOULD provide the ability to ignore the IPv6 traffic
                class and always set the IPv4 "TOS" to zero.

        Total Length:
                Payload length value from IPv6 header, plus the size
                of the IPv4 header.

        Identification:
                All zero.

        Flags:
                The More Fragments flag is set to zero.  The Don't
                Fragments flag is set to one.

        Fragment Offset:
                All zero.
      Time to Live:
                Hop Limit value copied from IPv6 header.  Since the
                translator is a router, as part of forwarding the
                packet it needs to decrement either the IPv6 Hop
                Limit (before the translation) or the IPv4 TTL (after
                the translation).  As part of decrementing the TTL or
                Hop Limit the translator (as any router) needs to
                check for zero and send the ICMPv4 or ICMPv6 "ttl
                exceeded" error.

        Protocol:
                Next Header field copied from IPv6 header.

        Header Checksum:
                Computed once the IPv4 header has been created.

        Source Address:
                If the IPv6 source address is an IPv4-translated
                address then the low-order 32 bits of the IPv6 source
                address is copied to the IPv4 source address.
                Otherwise, the source address is set to 0.0.0.0.  The
                use of 0.0.0.0 is to avoid completely dropping e.g.
                ICMPv6 error messages sent by IPv6-only routers which
                makes e.g. traceroute present something for the
                IPv6-only hops.
        Destination Address:
                IPv6 packets that are translated have an IPv4-mapped
                destination address.  Thus the low-order 32 bits of
                the IPv6 destination address is copied to the IPv4
                destination address.
```

Identifier: RQ_003_3088
RFC Clause: 4.1
Type: Mandatory
Applies to: Router

Requirement:

When translating the IPv6 packet to IPv4, the resulting IPv4 header's Destination Address field, SHALL contain the low-order 32 bits of the IPv6 destination address.

Specification Text:

If there is no IPv6 Fragment header the IPv4 header fields are set as follows:

```
        Version:
                4

        Internet Header Length:
                5 (no IPv4 options)

        Type of Service and Precedence:
                By default, copied from the IPv6 Traffic Class (all 8
                bits).  According to [DIFFSERV] the semantics of the
                bits are identical in IPv4 and IPv6.  However, in
                some IPv4 environments these bits might be used with
                the old semantics of "Type Of Service and
                Precedence".  An implementation of a translator
                SHOULD provide the ability to ignore the IPv6 traffic
                class and always set the IPv4 "TOS" to zero.

        Total Length:
                Payload length value from IPv6 header, plus the size
                of the IPv4 header.

        Identification:
                All zero.

        Flags:
                The More Fragments flag is set to zero.  The Don't
                Fragments flag is set to one.

        Fragment Offset:
                All zero.
      Time to Live:
                Hop Limit value copied from IPv6 header.  Since the
                translator is a router, as part of forwarding the
                packet it needs to decrement either the IPv6 Hop
                Limit (before the translation) or the IPv4 TTL (after
                the translation).  As part of decrementing the TTL or
                Hop Limit the translator (as any router) needs to
                check for zero and send the ICMPv4 or ICMPv6 "ttl
                exceeded" error.

        Protocol:
                Next Header field copied from IPv6 header.

        Header Checksum:
                Computed once the IPv4 header has been created.

        Source Address:
                If the IPv6 source address is an IPv4-translated
                address then the low-order 32 bits of the IPv6 source
                address is copied to the IPv4 source address.
                Otherwise, the source address is set to 0.0.0.0.  The
                use of 0.0.0.0 is to avoid completely dropping e.g.
                ICMPv6 error messages sent by IPv6-only routers which
                makes e.g. traceroute present something for the
                IPv6-only hops.
        Destination Address:
                IPv6 packets that are translated have an IPv4-mapped
                destination address.  Thus the low-order 32 bits of
                the IPv6 destination address is copied to the IPv4
                destination address.
```

Identifier: RQ_003_3089
RFC Clause: 4.1
Type: Mandatory
Applies to: Router

Requirement:

If an IPv6 hop-by-hop options header is present in the IPv6 packet, it is ignored by the translator.

Specification Text:

If any of an IPv6 hop-by-hop options header, destination options header, or routing header with the Segments Left field equal to zero are present in the IPv6 packet, they are ignored i.e., there is no attempt to translate them. However, the Total Length field and the Protocol field would have to be adjusted to "skip" these extension headers.

Identifier: RQ_003_3090
RFC Clause: 4.1
Type: Mandatory
Applies to: Router

Requirement:

If an IPv6 hop-by-hop destination options header is present in the IPv6 packet, it is ignored by the translator.

Specification Text:

If any of an IPv6 hop-by-hop options header, destination options header, or routing header with the Segments Left field equal to zero are present in the IPv6 packet, they are ignored i.e., there is no attempt to translate them. However, the Total Length field and the Protocol field would have to be adjusted to "skip" these extension headers.

Identifier: RQ_003_3091
RFC Clause: 4.1
Type: Mandatory
Applies to: Router

Requirement:

If an IPv6 routing header with the Segments Left field equal to zero is present in the IPv6 packet, it is ignored by the translator.

Specification Text:

If any of an IPv6 hop-by-hop options header, destination options header, or routing header with the Segments Left field equal to zero are present in the IPv6 packet, they are ignored i.e., there is no attempt to translate them. However, the Total Length field and the Protocol field would have to be adjusted to "skip" these extension headers.

Identifier: RQ_003_3092
RFC Clause: 4.1
Type: Mandatory
Applies to: Router

Requirement:

AS a result of the translator dropping any IPv6 hop-by-hop options header, destination options header, or routing header with the Segments Left field equal to zero are present in the IPv6 packet, the Total Length field and the Protocol field SHALL be adjusted to "skip" these extension headers.

Specification Text:

If any of an IPv6 hop-by-hop options header, destination options header, or routing header with the Segments Left field equal to zero are present in the IPv6 packet, they are ignored i.e., there is no attempt to translate them. However, the Total Length field and the Protocol field would have to be adjusted to "skip" these extension headers.

Identifier: RQ_003_3093
RFC Clause: 4.1
Type: Mandatory
Applies to: Router

Requirement:

If a routing header with a non-zero Segments Left field is present then the packet MUST NOT be translated.

Specification Text:

If a routing header with a non-zero Segments Left field is present then the packet MUST NOT be translated, and an ICMPv6 "parameter problem/ erroneous header field encountered" (Type 4/Code 0) error message, with the Pointer field indicating the first byte of the Segments Left field, SHOULD be returned to the sender.

Identifier: RQ_003_3094
RFC Clause: 4.1
Type: Recommendation
Applies to: Router

Requirement:

If a routing header with a non-zero Segments Left field is present then an ICMPv6 "parameter problem/ erroneous header field encountered" (Type 4/Code 0) error message, with the Pointer field indicating the first byte of the Segments Left field, SHOULD be returned to the sender.

Specification Text:

If a routing header with a non-zero Segments Left field is present then the packet MUST NOT be translated, and an ICMPv6 "parameter problem/ erroneous header field encountered" (Type 4/Code 0) error message, with the Pointer field indicating the first byte of the Segments Left field, SHOULD be returned to the sender.

Identifier: RQ_003_3095
RFC Clause: 4.1
Type: Mandatory
Applies to: Router

Requirement:

When translating the IPv6 packet to IPv4, if there is an IPv6 Fragment header, the resulting IPv4 header's Total Length field SHALL be set to the Payload length value from IPv6 header, minus 8 for the Fragment header, plus the size of the IPv4 header.

Specification Text:

If the IPv6 packet contains a Fragment header the header fields are set as above with the following exceptions:

Total Length:
Payload length value from IPv6 header, minus 8 for the Fragment header, plus the size of the IPv4 header.

Identification:
Copied from the low-order 16-bits in the Identification field in the Fragment header.

Flags:
The More Fragments flag is copied from the M flag in the Fragment header. The Don't Fragments flag is set to zero allowing this packet to be fragmented by IPv4 routers.

```
        Fragment Offset:
                Copied from the Fragment Offset field in the Fragment
                Header.

        Protocol:
                Next Header value copied from Fragment header.
```

Identifier: RQ_003_3096
RFC Clause: 4.1
Type: Mandatory
Applies to: Router

Requirement:

When translating the IPv6 packet to IPv4, if there is an IPv6 Fragment header, the resulting IPv4 header's Identification field SHALL be copied from the low-order 16-bits in the Identification field in the Fragment header.

Specification Text:

```
If the IPv6 packet contains a Fragment header the header fields are
   set as above with the following exceptions:

        Total Length:
                Payload length value from IPv6 header, minus 8 for
                the Fragment header, plus the size of the IPv4
                header.

        Identification:
                Copied from the low-order 16-bits in the
                Identification field in the Fragment header.

        Flags:
                The More Fragments flag is copied from the M flag in
                the Fragment header.  The Don't Fragments flag is set
                to zero allowing this packet to be fragmented by IPv4
                routers.

        Fragment Offset:
                Copied from the Fragment Offset field in the Fragment
                Header.

        Protocol:
                Next Header value copied from Fragment header.
```

Identifier: RQ_003_3097
RFC Clause: 4.1
Type: Mandatory
Applies to: Router

Requirement:

When translating the IPv6 packet to IPv4, if there is an IPv6 Fragment header, the resulting IPv4 header's More Fragments flag SHALL be copied from the M flag in the Fragment header.

Specification Text:

```
If the IPv6 packet contains a Fragment header the header fields are
   set as above with the following exceptions:

        Total Length:
                Payload length value from IPv6 header, minus 8 for
                the Fragment header, plus the size of the IPv4
                header.

        Identification:
                Copied from the low-order 16-bits in the
                Identification field in the Fragment header.

        Flags:
                The More Fragments flag is copied from the M flag in
                the Fragment header.  The Don't Fragments flag is set
                to zero allowing this packet to be fragmented by IPv4
                routers.
```

Fragment Offset:
Copied from the Fragment Offset field in the Fragment Header.

Protocol:
Next Header value copied from Fragment header.

Identifier: RQ_003_3098
RFC Clause: 4.1
Type: Mandatory
Applies to: Router

Requirement:
When translating the IPv6 packet to IPv4, if there is an IPv6 Fragment header, the resulting IPv4 header's Don't Fragments flag is set to zero.

Specification Text:
If the IPv6 packet contains a Fragment header the header fields are set as above with the following exceptions:

Total Length:
Payload length value from IPv6 header, minus 8 for the Fragment header, plus the size of the IPv4 header.

Identification:
Copied from the low-order 16-bits in the Identification field in the Fragment header.

Flags:
The More Fragments flag is copied from the M flag in the Fragment header. **The Don't Fragments flag is set to zero allowing this packet to be fragmented by IPv4 routers.**

Fragment Offset:
Copied from the Fragment Offset field in the Fragment Header.

Protocol:
Next Header value copied from Fragment header.

Identifier: RQ_003_3099
RFC Clause: 4.1
Type: Mandatory
Applies to: Router

Requirement:
When translating the IPv6 packet to IPv4, if there is an IPv6 Fragment header, the resulting IPv4 header's Fragment Offset field SHALL be copied from the Fragment Offset field in the Fragment Header.

Specification Text:
If the IPv6 packet contains a Fragment header the header fields are set as above with the following exceptions:

Total Length:
Payload length value from IPv6 header, minus 8 for the Fragment header, plus the size of the IPv4 header.

Identification:
Copied from the low-order 16-bits in the Identification field in the Fragment header.

Flags:
The More Fragments flag is copied from the M flag in the Fragment header. The Don't Fragments flag is set to zero allowing this packet to be fragmented by IPv4 routers.

Fragment Offset:
Copied from the Fragment Offset field in the Fragment Header.

Protocol:
Next Header value copied from Fragment header.

Identifier:	RQ_003_3100
RFC Clause:	4.1
Type:	Mandatory
Applies to:	Router

Requirement:
When translating the IPv6 packet to IPv4, if there is an IPv6 Fragment header, the resulting IPv4 header's Protocol field SHALL be the Next Header value copied from Fragment header.

Specification Text:
If the IPv6 packet contains a Fragment header the header fields are set as above with the following exceptions:

Total Length:
Payload length value from IPv6 header, minus 8 for the Fragment header, plus the size of the IPv4 header.

Identification:
Copied from the low-order 16-bits in the Identification field in the Fragment header.

Flags:
The More Fragments flag is copied from the M flag in the Fragment header. The Don't Fragments flag is set to zero allowing this packet to be fragmented by IPv4 routers.

Fragment Offset:
Copied from the Fragment Offset field in the Fragment Header.

Protocol:
Next Header value copied from Fragment header.

Identifier:	RQ_003_3101
RFC Clause:	4.2
Type:	Mandatory
Applies to:	Router

Requirement:
All ICMP messages that are to be translated from IPv6 to IPv4 require that the ICMP checksum field be updated as part of the translation.

Specification Text:
All ICMP messages that are to be translated require that the ICMP checksum field be updated as part of the translation since ICMPv6, unlike ICMPv4, has a pseudo-header checksum just like UDP and TCP.

Identifier:	RQ_003_3102
RFC Clause:	4.2
Type:	Mandatory
Applies to:	Router

Requirement:

All ICMP packets that are to be translated from IPv6 to IPv4 need to have the Type value translated.

Specification Text:
In addition all ICMP packets need to have the Type value translated and for ICMP error messages the included IP header also needs translation.

Identifier: RQ_003_3103
RFC Clause: 4.2
Type: Mandatory
Applies to: Router

Requirement:
All ICMP error messages that are to be translated from IPv6 to IPv4 need to have the included IP header translated.

Specification Text:
In addition all ICMP packets need to have the Type value translated and **for ICMP error messages the included IP header also needs translation.**

Identifier: RQ_003_3104
RFC Clause: 4.2
Type: Mandatory
Applies to: Router

Requirement:
In order to translate ICMPv6 informational messages, for the Echo (Type 128), adjust the type to 0.

Specification Text:

```
The actions needed to translate various ICMPv6 messages are:

      ICMPv6 informational messages:

        Echo Request and Echo Reply (Type 128 and 129)
           Adjust the type to 0 and 8, respectively, and adjust the ICMP
           checksum both to take the type change into account and to
           exclude the ICMPv6 pseudo-header.

        MLD Multicast Listener Query/Report/Done (Type 130, 131, 132)
           Single hop message.  Silently drop.
        Neighbor Discover messages (Type 133 through 137)
           Single hop message.  Silently drop.

        Unknown informational messages
           Silently drop.

      ICMPv6 error messages:

        Destination Unreachable (Type 1)
           Set the Type field to 3.  Translate the code field as
           follows:
              Code 0 (no route to destination):
                     Set Code to 1 (host unreachable).

              Code 1 (communication with destination administratively
              prohibited):
                     Set Code to 10 (communication with destination host
                     administratively prohibited).

              Code 2 (beyond scope of source address):
                     Set Code to 1 (host unreachable).  Note that this
                     error is very unlikely since the IPv4-translatable
                     source address is considered to have global scope.

              Code 3 (address unreachable):
                     Set Code to 1 (host unreachable).

              Code 4 (port unreachable):
                     Set Code to 3 (port unreachable).

        Packet Too Big (Type 2)
           Translate to an ICMPv4 Destination Unreachable with code 4.
           The MTU field needs to be adjusted for the difference between
           the IPv4 and IPv6 header sizes taking into account whether or
           not the packet in error includes a Fragment header.

        Time Exceeded (Type 3)
           Set the Type to 11.  The Code field is unchanged.
        Parameter Problem (Type 4)
           If the Code is 1 translate this to an ICMPv4 protocol
           unreachable (Type 3, Code 2).  Otherwise set the Type to 12
           and the Code to zero.  The Pointer needs to be updated to
           point to the corresponding field in the translated include IP
           header.

        Unknown error messages
           Silently drop.

----------------
```

Identifier: RQ_003_3105
RFC Clause: 4.2
Type: Mandatory
Applies to: Router

Requirement:
In order to translate ICMPv4 query messages, after adjusting the Echo (Type 128) to 0, adjust the ICMP checksum both to take the type change into account and to include the ICMPv6 pseudo-header.

Specification Text:

```
The actions needed to translate various ICMPv6 messages are:

      ICMPv6 informational messages:

        Echo Request and Echo Reply (Type 128 and 129)
           Adjust the type to 0 and 8, respectively, and adjust the ICMP
           checksum both to take the type change into account and to
           exclude the ICMPv6 pseudo-header.

        MLD Multicast Listener Query/Report/Done (Type 130, 131, 132)
           Single hop message.  Silently drop.
        Neighbor Discover messages (Type 133 through 137)
           Single hop message.  Silently drop.

        Unknown informational messages
           Silently drop.

      ICMPv6 error messages:

        Destination Unreachable (Type 1)
           Set the Type field to 3.  Translate the code field as
           follows:
              Code 0 (no route to destination):
                     Set Code to 1 (host unreachable).

              Code 1 (communication with destination administratively
              prohibited):
                     Set Code to 10 (communication with destination host
                     administratively prohibited).

              Code 2 (beyond scope of source address):
                     Set Code to 1 (host unreachable).  Note that this
                     error is very unlikely since the IPv4-translatable
                     source address is considered to have global scope.

              Code 3 (address unreachable):
                     Set Code to 1 (host unreachable).

              Code 4 (port unreachable):
                     Set Code to 3 (port unreachable).

        Packet Too Big (Type 2)
           Translate to an ICMPv4 Destination Unreachable with code 4.
           The MTU field needs to be adjusted for the difference between
           the IPv4 and IPv6 header sizes taking into account whether or
           not the packet in error includes a Fragment header.

        Time Exceeded (Type 3)
           Set the Type to 11.  The Code field is unchanged.
```

Parameter Problem (Type 4)
If the Code is 1 translate this to an ICMPv4 protocol unreachable (Type 3, Code 2). Otherwise set the Type to 12 and the Code to zero. The Pointer needs to be updated to point to the corresponding field in the translated include IP header.

Unknown error messages
Silently drop.

Identifier: RQ_003_3106
RFC Clause: 4.2
Type: Mandatory
Applies to: Router

Requirement:
In order to translate ICMPv6 informational messages, for the Echo Reply (Type 128), adjust the type to 8.

Specification Text:
The actions needed to translate various ICMPv6 messages are:

ICMPv6 informational messages:

Echo Request and **Echo Reply** (**Type** 128 and **129**)
Adjust the type to 0 and **8**, respectively, and adjust the ICMP checksum both to take the type change into account and to exclude the ICMPv6 pseudo-header.

MLD Multicast Listener Query/Report/Done (Type 130, 131, 132)
Single hop message. Silently drop.
Neighbor Discover messages (Type 133 through 137)
Single hop message. Silently drop.

Unknown informational messages
Silently drop.

ICMPv6 error messages:

Destination Unreachable (Type 1)
Set the Type field to 3. Translate the code field as follows:
Code 0 (no route to destination):
Set Code to 1 (host unreachable).

Code 1 (communication with destination administratively prohibited):
Set Code to 10 (communication with destination host administratively prohibited).

Code 2 (beyond scope of source address):
Set Code to 1 (host unreachable). Note that this error is very unlikely since the IPv4-translatable source address is considered to have global scope.

Code 3 (address unreachable):
Set Code to 1 (host unreachable).

Code 4 (port unreachable):
Set Code to 3 (port unreachable).

Packet Too Big (Type 2)
Translate to an ICMPv4 Destination Unreachable with code 4. The MTU field needs to be adjusted for the difference between the IPv4 and IPv6 header sizes taking into account whether or not the packet in error includes a Fragment header.

Time Exceeded (Type 3)
Set the Type to 11. The Code field is unchanged.

Parameter Problem (Type 4)

If the Code is 1 translate this to an ICMPv4 protocol unreachable (Type 3, Code 2). Otherwise set the Type to 12 and the Code to zero. The Pointer needs to be updated to point to the corresponding field in the translated include IP header.

Unknown error messages

Silently drop.

Identifier: RQ_003_3107
RFC Clause: 4.2
Type: Mandatory
Applies to: Router

Requirement:

In order to translate ICMPv4 query messages, after adjusting the Echo Reply (Type 129) to 8, adjust the ICMP checksum both to take the type change into account and to include the ICMPv6 pseudo-header.

Specification Text:

The actions needed to translate various ICMPv6 messages are:

ICMPv6 informational messages:

Echo Request and **Echo Reply** (Type 128 and **129**)

Adjust the type to 0 and **8**, respectively, **and adjust the ICMP checksum both to take the type change into account and to exclude the ICMPv6 pseudo-header.**

MLD Multicast Listener Query/Report/Done (Type 130, 131, 132)

Single hop message. Silently drop.

Neighbor Discover messages (Type 133 through 137)

Single hop message. Silently drop.

Unknown informational messages

Silently drop.

ICMPv6 error messages:

Destination Unreachable (Type 1)

Set the Type field to 3. Translate the code field as follows:

Code 0 (no route to destination):

Set Code to 1 (host unreachable).

Code 1 (communication with destination administratively prohibited):

Set Code to 10 (communication with destination host administratively prohibited).

Code 2 (beyond scope of source address):

Set Code to 1 (host unreachable). Note that this error is very unlikely since the IPv4-translatable source address is considered to have global scope.

Code 3 (address unreachable):

Set Code to 1 (host unreachable).

Code 4 (port unreachable):

Set Code to 3 (port unreachable).

Packet Too Big (Type 2)

Translate to an ICMPv4 Destination Unreachable with code 4. The MTU field needs to be adjusted for the difference between the IPv4 and IPv6 header sizes taking into account whether or not the packet in error includes a Fragment header.

Time Exceeded (Type 3)

Set the Type to 11. The Code field is unchanged.

```
        Parameter Problem (Type 4)
           If the Code is 1 translate this to an ICMPv4 protocol
           unreachable (Type 3, Code 2).  Otherwise set the Type to 12
           and the Code to zero.  The Pointer needs to be updated to
           point to the corresponding field in the translated include IP
           header.

        Unknown error messages
           Silently drop.
```

Identifier: RQ_003_3108
RFC Clause: 4.2
Type: Mandatory
Applies to: Router

Requirement:

In order to translate ICMPv6 informational messages, MLD Multicast Listener Query (Type 130) Single hop message SHOULD be silently dropped.

Specification Text:

```
The actions needed to translate various ICMPv6 messages are:

      ICMPv6 informational messages:

        Echo Request and Echo Reply (Type 128 and 129)
           Adjust the type to 0 and 8, respectively, and adjust the ICMP
           checksum both to take the type change into account and to
           exclude the ICMPv6 pseudo-header.

        MLD Multicast Listener Query/Report/Done (Type 130, 131, 132)
           Single hop message.  Silently drop.
        Neighbor Discover messages (Type 133 through 137)
           Single hop message.  Silently drop.

        Unknown informational messages
           Silently drop.

      ICMPv6 error messages:

        Destination Unreachable (Type 1)
           Set the Type field to 3.  Translate the code field as
           follows:
              Code 0 (no route to destination):
                     Set Code to 1 (host unreachable).

              Code 1 (communication with destination administratively
              prohibited):
                     Set Code to 10 (communication with destination host
                     administratively prohibited).

              Code 2 (beyond scope of source address):
                     Set Code to 1 (host unreachable).  Note that this
                     error is very unlikely since the IPv4-translatable
                     source address is considered to have global scope.

              Code 3 (address unreachable):
                     Set Code to 1 (host unreachable).

              Code 4 (port unreachable):
                     Set Code to 3 (port unreachable).

        Packet Too Big (Type 2)
           Translate to an ICMPv4 Destination Unreachable with code 4.
           The MTU field needs to be adjusted for the difference between
           the IPv4 and IPv6 header sizes taking into account whether or
           not the packet in error includes a Fragment header.

        Time Exceeded (Type 3)
           Set the Type to 11.  The Code field is unchanged.
        Parameter Problem (Type 4)
           If the Code is 1 translate this to an ICMPv4 protocol
           unreachable (Type 3, Code 2).  Otherwise set the Type to 12
           and the Code to zero.  The Pointer needs to be updated to
           point to the corresponding field in the translated include IP
           header.

        Unknown error messages
           Silently drop.

----------------
```

Identifier: RQ_003_3109
RFC Clause: 4.2
Type: Mandatory
Applies to: Router

Requirement:

In order to translate ICMPv6 informational messages, MLD Multicast Listener Report (Type 131) Single hop message SHOULD be silently dropped.

Specification Text:

```
The actions needed to translate various ICMPv6 messages are:

      ICMPv6 informational messages:

        Echo Request and Echo Reply (Type 128 and 129)
           Adjust the type to 0 and 8, respectively, and adjust the ICMP
           checksum both to take the type change into account and to
           exclude the ICMPv6 pseudo-header.

        MLD Multicast Listener Query/Report/Done (Type 130, 131, 132)
           Single hop message.  Silently drop.
        Neighbor Discover messages (Type 133 through 137)
           Single hop message.  Silently drop.

        Unknown informational messages
           Silently drop.

      ICMPv6 error messages:

        Destination Unreachable (Type 1)
           Set the Type field to 3.  Translate the code field as
           follows:
              Code 0 (no route to destination):
                     Set Code to 1 (host unreachable).

              Code 1 (communication with destination administratively
              prohibited):
                     Set Code to 10 (communication with destination host
                     administratively prohibited).

              Code 2 (beyond scope of source address):
                     Set Code to 1 (host unreachable).  Note that this
                     error is very unlikely since the IPv4-translatable
                     source address is considered to have global scope.

              Code 3 (address unreachable):
                     Set Code to 1 (host unreachable).

              Code 4 (port unreachable):
                     Set Code to 3 (port unreachable).

        Packet Too Big (Type 2)
           Translate to an ICMPv4 Destination Unreachable with code 4.
           The MTU field needs to be adjusted for the difference between
           the IPv4 and IPv6 header sizes taking into account whether or
           not the packet in error includes a Fragment header.

        Time Exceeded (Type 3)
           Set the Type to 11.  The Code field is unchanged.
```

```
     Parameter Problem (Type 4)
        If the Code is 1 translate this to an ICMPv4 protocol
        unreachable (Type 3, Code 2).  Otherwise set the Type to 12
        and the Code to zero.  The Pointer needs to be updated to
        point to the corresponding field in the translated include IP
        header.

     Unknown error messages
        Silently drop.
```

Identifier:	RQ_003_3110
RFC Clause:	4.2
Type:	Mandatory
Applies to:	Router

Requirement:

In order to translate ICMPv6 informational messages, MLD Multicast Listener Done (Type 132)Single hop message SHOULD be silently dropped.

Specification Text:

```
The actions needed to translate various ICMPv6 messages are:

      ICMPv6 informational messages:

        Echo Request and Echo Reply (Type 128 and 129)
           Adjust the type to 0 and 8, respectively, and adjust the ICMP
           checksum both to take the type change into account and to
           exclude the ICMPv6 pseudo-header.

        MLD Multicast Listener Query/Report/Done (Type 130, 131, 132)
           Single hop message.  Silently drop.
        Neighbor Discover messages (Type 133 through 137)
           Single hop message.  Silently drop.

        Unknown informational messages
           Silently drop.

      ICMPv6 error messages:

        Destination Unreachable (Type 1)
           Set the Type field to 3.  Translate the code field as
           follows:
              Code 0 (no route to destination):
                     Set Code to 1 (host unreachable).

              Code 1 (communication with destination administratively
              prohibited):
                     Set Code to 10 (communication with destination host
                     administratively prohibited).

              Code 2 (beyond scope of source address):
                     Set Code to 1 (host unreachable).  Note that this
                     error is very unlikely since the IPv4-translatable
                     source address is considered to have global scope.

              Code 3 (address unreachable):
                     Set Code to 1 (host unreachable).

              Code 4 (port unreachable):
                     Set Code to 3 (port unreachable).

        Packet Too Big (Type 2)
           Translate to an ICMPv4 Destination Unreachable with code 4.
           The MTU field needs to be adjusted for the difference between
           the IPv4 and IPv6 header sizes taking into account whether or
           not the packet in error includes a Fragment header.

        Time Exceeded (Type 3)
           Set the Type to 11.  The Code field is unchanged.
        Parameter Problem (Type 4)
           If the Code is 1 translate this to an ICMPv4 protocol
           unreachable (Type 3, Code 2).  Otherwise set the Type to 12
           and the Code to zero.  The Pointer needs to be updated to
           point to the corresponding field in the translated include IP
           header.

        Unknown error messages
           Silently drop.
```

Identifier: RQ_003_3111
RFC Clause: 4.2
Type: Mandatory
Applies to: Router

Requirement:
In order to translate ICMPv6 informational messages, Neighbor Discover - Router Solicitation (Type 133) Single hop message SHOULD be silently dropped.

Specification Text:

The actions needed to translate various ICMPv6 messages are:

```
      ICMPv6 informational messages:

        Echo Request and Echo Reply (Type 128 and 129)
           Adjust the type to 0 and 8, respectively, and adjust the ICMP
           checksum both to take the type change into account and to
           exclude the ICMPv6 pseudo-header.

        MLD Multicast Listener Query/Report/Done (Type 130, 131, 132)
           Single hop message.  Silently drop.
        Neighbor Discover messages (Type 133 through 137)
           Single hop message.  Silently drop.

        Unknown informational messages
           Silently drop.

      ICMPv6 error messages:

        Destination Unreachable (Type 1)
           Set the Type field to 3.  Translate the code field as
           follows:
              Code 0 (no route to destination):
                     Set Code to 1 (host unreachable).

              Code 1 (communication with destination administratively
              prohibited):
                     Set Code to 10 (communication with destination host
                     administratively prohibited).

              Code 2 (beyond scope of source address):
                     Set Code to 1 (host unreachable).  Note that this
                     error is very unlikely since the IPv4-translatable
                     source address is considered to have global scope.

              Code 3 (address unreachable):
                     Set Code to 1 (host unreachable).

              Code 4 (port unreachable):
                     Set Code to 3 (port unreachable).

        Packet Too Big (Type 2)
           Translate to an ICMPv4 Destination Unreachable with code 4.
           The MTU field needs to be adjusted for the difference between
           the IPv4 and IPv6 header sizes taking into account whether or
           not the packet in error includes a Fragment header.

        Time Exceeded (Type 3)
           Set the Type to 11.  The Code field is unchanged.
        Parameter Problem (Type 4)
           If the Code is 1 translate this to an ICMPv4 protocol
           unreachable (Type 3, Code 2).  Otherwise set the Type to 12
           and the Code to zero.  The Pointer needs to be updated to
           point to the corresponding field in the translated include IP
           header.

        Unknown error messages
           Silently drop.
```

Identifier: RQ_003_3112
RFC Clause: 4.2
Type: Mandatory
Applies to: Router

Requirement:

In order to translate ICMPv6 informational messages, Neighbor Discover - Router Advertisement (Type 134) Single hop message SHOULD be silently dropped.

Specification Text:

The actions needed to translate various ICMPv6 messages are:

```
      ICMPv6 informational messages:

        Echo Request and Echo Reply (Type 128 and 129)
           Adjust the type to 0 and 8, respectively, and adjust the ICMP
           checksum both to take the type change into account and to
           exclude the ICMPv6 pseudo-header.

        MLD Multicast Listener Query/Report/Done (Type 130, 131, 132)
           Single hop message.  Silently drop.
        Neighbor Discover messages (Type 133 through 137)
           Single hop message.  Silently drop.

        Unknown informational messages
           Silently drop.

      ICMPv6 error messages:

        Destination Unreachable (Type 1)
           Set the Type field to 3.  Translate the code field as
           follows:
              Code 0 (no route to destination):
                     Set Code to 1 (host unreachable).

              Code 1 (communication with destination administratively
              prohibited):
                     Set Code to 10 (communication with destination host
                     administratively prohibited).

              Code 2 (beyond scope of source address):
                     Set Code to 1 (host unreachable).  Note that this
                     error is very unlikely since the IPv4-translatable
                     source address is considered to have global scope.

              Code 3 (address unreachable):
                     Set Code to 1 (host unreachable).

              Code 4 (port unreachable):
                     Set Code to 3 (port unreachable).

        Packet Too Big (Type 2)
           Translate to an ICMPv4 Destination Unreachable with code 4.
           The MTU field needs to be adjusted for the difference between
           the IPv4 and IPv6 header sizes taking into account whether or
           not the packet in error includes a Fragment header.

        Time Exceeded (Type 3)
           Set the Type to 11.  The Code field is unchanged.
        Parameter Problem (Type 4)
           If the Code is 1 translate this to an ICMPv4 protocol
           unreachable (Type 3, Code 2).  Otherwise set the Type to 12
           and the Code to zero.  The Pointer needs to be updated to
           point to the corresponding field in the translated include IP
           header.

        Unknown error messages
           Silently drop.

----------------
```

Identifier: RQ_003_3113
RFC Clause: 4.2
Type: Mandatory
Applies to: Router

Requirement:

In order to translate ICMPv6 informational messages, Neighbor Discover - Neighbor Solicitation (Type 135) Single hop message SHOULD be silently dropped.

Specification Text:

```
The actions needed to translate various ICMPv6 messages are:

      ICMPv6 informational messages:

        Echo Request and Echo Reply (Type 128 and 129)
           Adjust the type to 0 and 8, respectively, and adjust the ICMP
           checksum both to take the type change into account and to
           exclude the ICMPv6 pseudo-header.

        MLD Multicast Listener Query/Report/Done (Type 130, 131, 132)
           Single hop message.  Silently drop.
        Neighbor Discover messages (Type 133 through 137)
           Single hop message.  Silently drop.

        Unknown informational messages
           Silently drop.

      ICMPv6 error messages:

        Destination Unreachable (Type 1)
           Set the Type field to 3.  Translate the code field as
           follows:
              Code 0 (no route to destination):
                     Set Code to 1 (host unreachable).

              Code 1 (communication with destination administratively
              prohibited):
                     Set Code to 10 (communication with destination host
                     administratively prohibited).

              Code 2 (beyond scope of source address):
                     Set Code to 1 (host unreachable).  Note that this
                     error is very unlikely since the IPv4-translatable
                     source address is considered to have global scope.

              Code 3 (address unreachable):
                     Set Code to 1 (host unreachable).

              Code 4 (port unreachable):
                     Set Code to 3 (port unreachable).

        Packet Too Big (Type 2)
           Translate to an ICMPv4 Destination Unreachable with code 4.
           The MTU field needs to be adjusted for the difference between
           the IPv4 and IPv6 header sizes taking into account whether or
           not the packet in error includes a Fragment header.

        Time Exceeded (Type 3)
           Set the Type to 11.  The Code field is unchanged.
```

Parameter Problem (Type 4)

If the Code is 1 translate this to an ICMPv4 protocol unreachable (Type 3, Code 2). Otherwise set the Type to 12 and the Code to zero. The Pointer needs to be updated to point to the corresponding field in the translated include IP header.

Unknown error messages

Silently drop.

Identifier: RQ_003_3114
RFC Clause: 4.2
Type: Mandatory
Applies to: Router

Requirement:

In order to translate ICMPv6 informational messages, Neighbor Discover - Neighbor Adverisement (Type 136) Single hop message SHOULD be silently dropped.

Specification Text:

The actions needed to translate various ICMPv6 messages are:

ICMPv6 informational messages:

Echo Request and Echo Reply (Type 128 and 129)

Adjust the type to 0 and 8, respectively, and adjust the ICMP checksum both to take the type change into account and to exclude the ICMPv6 pseudo-header.

MLD Multicast Listener Query/Report/Done (Type 130, 131, 132)

Single hop message. Silently drop.

Neighbor Discover messages (Type 133 through 137)

Single hop message. Silently drop.

Unknown informational messages

Silently drop.

ICMPv6 error messages:

Destination Unreachable (Type 1)

Set the Type field to 3. Translate the code field as follows:

Code 0 (no route to destination):

Set Code to 1 (host unreachable).

Code 1 (communication with destination administratively prohibited):

Set Code to 10 (communication with destination host administratively prohibited).

Code 2 (beyond scope of source address):

Set Code to 1 (host unreachable). Note that this error is very unlikely since the IPv4-translatable source address is considered to have global scope.

Code 3 (address unreachable):

Set Code to 1 (host unreachable).

Code 4 (port unreachable):

Set Code to 3 (port unreachable).

Packet Too Big (Type 2)

Translate to an ICMPv4 Destination Unreachable with code 4. The MTU field needs to be adjusted for the difference between the IPv4 and IPv6 header sizes taking into account whether or not the packet in error includes a Fragment header.

Time Exceeded (Type 3)

Set the Type to 11. The Code field is unchanged.

Parameter Problem (Type 4)

If the Code is 1 translate this to an ICMPv4 protocol unreachable (Type 3, Code 2). Otherwise set the Type to 12 and the Code to zero. The Pointer needs to be updated to point to the corresponding field in the translated include IP header.

Unknown error messages

Silently drop.

Identifier: RQ_003_3115
RFC Clause: 4.2
Type: Mandatory
Applies to: Router

Requirement:

In order to translate ICMPv6 informational messages, Neighbor Discover - Redirect Message (Type 137) Single hop message SHOULD be silently dropped.

Specification Text:

The actions needed to translate various ICMPv6 messages are:

ICMPv6 informational messages:

Echo Request and Echo Reply (Type 128 and 129)

Adjust the type to 0 and 8, respectively, and adjust the ICMP checksum both to take the type change into account and to exclude the ICMPv6 pseudo-header.

MLD Multicast Listener Query/Report/Done (Type 130, 131, 132)

Single hop message. Silently drop.

Neighbor Discover messages (Type 133 through 137)

Single hop message. Silently drop.

Unknown informational messages

Silently drop.

ICMPv6 error messages:

Destination Unreachable (Type 1)

Set the Type field to 3. Translate the code field as follows:

Code 0 (no route to destination):

Set Code to 1 (host unreachable).

Code 1 (communication with destination administratively prohibited):

Set Code to 10 (communication with destination host administratively prohibited).

Code 2 (beyond scope of source address):

Set Code to 1 (host unreachable). Note that this error is very unlikely since the IPv4-translatable source address is considered to have global scope.

Code 3 (address unreachable):

Set Code to 1 (host unreachable).

Code 4 (port unreachable):

Set Code to 3 (port unreachable).

Packet Too Big (Type 2)

Translate to an ICMPv4 Destination Unreachable with code 4. The MTU field needs to be adjusted for the difference between the IPv4 and IPv6 header sizes taking into account whether or not the packet in error includes a Fragment header.

Time Exceeded (Type 3)

Set the Type to 11. The Code field is unchanged.

```
        Parameter Problem (Type 4)
           If the Code is 1 translate this to an ICMPv4 protocol
           unreachable (Type 3, Code 2).  Otherwise set the Type to 12
           and the Code to zero.  The Pointer needs to be updated to
           point to the corresponding field in the translated include IP
           header.

        Unknown error messages
           Silently drop.

----------------
```

Identifier: RQ_003_3116
RFC Clause: 4.2
Type: Mandatory
Applies to: Router

Requirement:

In order to translate ICMPv6 informational messages, Unknown informational messages SHOULD be silently dropped.

Specification Text:

```
The actions needed to translate various ICMPv6 messages are:

      ICMPv6 informational messages:

        Echo Request and Echo Reply (Type 128 and 129)
           Adjust the type to 0 and 8, respectively, and adjust the ICMP
           checksum both to take the type change into account and to
           exclude the ICMPv6 pseudo-header.

        MLD Multicast Listener Query/Report/Done (Type 130, 131, 132)
           Single hop message.  Silently drop.
        Neighbor Discover messages (Type 133 through 137)
           Single hop message.  Silently drop.

        Unknown informational messages
           Silently drop.

      ICMPv6 error messages:

        Destination Unreachable (Type 1)
           Set the Type field to 3.  Translate the code field as
           follows:
              Code 0 (no route to destination):
                     Set Code to 1 (host unreachable).

              Code 1 (communication with destination administratively
              prohibited):
                     Set Code to 10 (communication with destination host
                     administratively prohibited).

              Code 2 (beyond scope of source address):
                     Set Code to 1 (host unreachable).  Note that this
                     error is very unlikely since the IPv4-translatable
                     source address is considered to have global scope.

              Code 3 (address unreachable):
                     Set Code to 1 (host unreachable).

              Code 4 (port unreachable):
                     Set Code to 3 (port unreachable).

        Packet Too Big (Type 2)
           Translate to an ICMPv4 Destination Unreachable with code 4.
           The MTU field needs to be adjusted for the difference between
           the IPv4 and IPv6 header sizes taking into account whether or
           not the packet in error includes a Fragment header.

        Time Exceeded (Type 3)
           Set the Type to 11.  The Code field is unchanged.
        Parameter Problem (Type 4)
           If the Code is 1 translate this to an ICMPv4 protocol
           unreachable (Type 3, Code 2).  Otherwise set the Type to 12
           and the Code to zero.  The Pointer needs to be updated to
           point to the corresponding field in the translated include IP
           header.

        Unknown error messages
           Silently drop.

----------------
```

Identifier: RQ_003_3117
RFC Clause: 4.2
Type: Mandatory
Applies to: Router

Requirement:

In order to translate ICMPv6 error messages, for Destination Unreachable (Type 1) messages Set the Type field to 3.

Specification Text:

```
The actions needed to translate various ICMPv6 messages are:

      ICMPv6 informational messages:

        Echo Request and Echo Reply (Type 128 and 129)
           Adjust the type to 0 and 8, respectively, and adjust the ICMP
           checksum both to take the type change into account and to
           exclude the ICMPv6 pseudo-header.

        MLD Multicast Listener Query/Report/Done (Type 130, 131, 132)
           Single hop message.  Silently drop.
        Neighbor Discover messages (Type 133 through 137)
           Single hop message.  Silently drop.

        Unknown informational messages
           Silently drop.

      ICMPv6 error messages:

        Destination Unreachable (Type 1)
           Set the Type field to 3.  Translate the code field as
           follows:
              Code 0 (no route to destination):
                     Set Code to 1 (host unreachable).

              Code 1 (communication with destination administratively
              prohibited):
                     Set Code to 10 (communication with destination host
                     administratively prohibited).

              Code 2 (beyond scope of source address):
                     Set Code to 1 (host unreachable).  Note that this
                     error is very unlikely since the IPv4-translatable
                     source address is considered to have global scope.

              Code 3 (address unreachable):
                     Set Code to 1 (host unreachable).

              Code 4 (port unreachable):
                     Set Code to 3 (port unreachable).

        Packet Too Big (Type 2)
           Translate to an ICMPv4 Destination Unreachable with code 4.
           The MTU field needs to be adjusted for the difference between
           the IPv4 and IPv6 header sizes taking into account whether or
           not the packet in error includes a Fragment header.

        Time Exceeded (Type 3)
           Set the Type to 11.  The Code field is unchanged.
        Parameter Problem (Type 4)
           If the Code is 1 translate this to an ICMPv4 protocol
           unreachable (Type 3, Code 2).  Otherwise set the Type to 12
           and the Code to zero.  The Pointer needs to be updated to
           point to the corresponding field in the translated include IP
           header.

        Unknown error messages
           Silently drop.

----------------
```

Identifier: RQ_003_3118
RFC Clause: 4.2
Type: Mandatory
Applies to: Router

Requirement:

```
In order to translate ICMPv6 error messages, for Destination Unreachable (Type 1) messages
Translate the Code Field, Code 0 (no route to destination), to 1 (host unreachable).
```

Specification Text:

```
The actions needed to translate various ICMPv6 messages are:

      ICMPv6 informational messages:

        Echo Request and Echo Reply (Type 128 and 129)
           Adjust the type to 0 and 8, respectively, and adjust the ICMP
           checksum both to take the type change into account and to
           exclude the ICMPv6 pseudo-header.

        MLD Multicast Listener Query/Report/Done (Type 130, 131, 132)
           Single hop message.  Silently drop.
        Neighbor Discover messages (Type 133 through 137)
           Single hop message.  Silently drop.

        Unknown informational messages
           Silently drop.

      ICMPv6 error messages:

        Destination Unreachable (Type 1)
           Set the Type field to 3.  Translate the code field as
           follows:
              Code 0 (no route to destination):
                     Set Code to 1 (host unreachable).

              Code 1 (communication with destination administratively
              prohibited):
                     Set Code to 10 (communication with destination host
                     administratively prohibited).

              Code 2 (beyond scope of source address):
                     Set Code to 1 (host unreachable).  Note that this
                     error is very unlikely since the IPv4-translatable
                     source address is considered to have global scope.

              Code 3 (address unreachable):
                     Set Code to 1 (host unreachable).

              Code 4 (port unreachable):
                     Set Code to 3 (port unreachable).

        Packet Too Big (Type 2)
           Translate to an ICMPv4 Destination Unreachable with code 4.
           The MTU field needs to be adjusted for the difference between
           the IPv4 and IPv6 header sizes taking into account whether or
           not the packet in error includes a Fragment header.

        Time Exceeded (Type 3)
           Set the Type to 11.  The Code field is unchanged.
```

Parameter Problem (Type 4)

If the Code is 1 translate this to an ICMPv4 protocol unreachable (Type 3, Code 2). Otherwise set the Type to 12 and the Code to zero. The Pointer needs to be updated to point to the corresponding field in the translated include IP header.

Unknown error messages

Silently drop.

Identifier: RQ_003_3119
RFC Clause: 4.2
Type: Mandatory
Applies to: Router

Requirement:

In order to translate ICMPv6 error messages, for Destination Unreachable (Type 1) messages

Translate the Code Field, Code 1 (communication with destination administratively prohibited), to 10 communication with destination host administratively prohibited).

Specification Text:

The actions needed to translate various ICMPv6 messages are:

ICMPv6 informational messages:

Echo Request and Echo Reply (Type 128 and 129)

Adjust the type to 0 and 8, respectively, and adjust the ICMP checksum both to take the type change into account and to exclude the ICMPv6 pseudo-header.

MLD Multicast Listener Query/Report/Done (Type 130, 131, 132)

Single hop message. Silently drop.

Neighbor Discover messages (Type 133 through 137)

Single hop message. Silently drop.

Unknown informational messages

Silently drop.

ICMPv6 error messages:

Destination Unreachable (Type 1)

Set the Type field to 3. **Translate the code field as follows:**

Code 0 (no route to destination):

Set Code to 1 (host unreachable).

Code 1 (communication with destination administratively prohibited):

Set Code to 10 (communication with destination host administratively prohibited).

Code 2 (beyond scope of source address):

Set Code to 1 (host unreachable). Note that this error is very unlikely since the IPv4-translatable source address is considered to have global scope.

Code 3 (address unreachable):

Set Code to 1 (host unreachable).

Code 4 (port unreachable):

Set Code to 3 (port unreachable).

Packet Too Big (Type 2)

Translate to an ICMPv4 Destination Unreachable with code 4. The MTU field needs to be adjusted for the difference between the IPv4 and IPv6 header sizes taking into account whether or not the packet in error includes a Fragment header.

Time Exceeded (Type 3)

Set the Type to 11. The Code field is unchanged.

Parameter Problem (Type 4)
If the Code is 1 translate this to an ICMPv4 protocol unreachable (Type 3, Code 2). Otherwise set the Type to 12 and the Code to zero. The Pointer needs to be updated to point to the corresponding field in the translated include IP header.

Unknown error messages
Silently drop.

Identifier: RQ_003_3120
RFC Clause: 4.2
Type: Mandatory
Applies to: Router

Requirement:
In order to translate ICMPv6 error messages, for Destination Unreachable (Type 1) messages Translate the Code Field, Code 2 (beyond scope of source address), to 1 (host unreachable).

Specification Text:
The actions needed to translate various ICMPv6 messages are:

ICMPv6 informational messages:

Echo Request and Echo Reply (Type 128 and 129)
Adjust the type to 0 and 8, respectively, and adjust the ICMP checksum both to take the type change into account and to exclude the ICMPv6 pseudo-header.

MLD Multicast Listener Query/Report/Done (Type 130, 131, 132)
Single hop message. Silently drop.
Neighbor Discover messages (Type 133 through 137)
Single hop message. Silently drop.

Unknown informational messages
Silently drop.

ICMPv6 error messages:

Destination Unreachable (Type 1)
Set the Type field to 3. Translate the code field as follows:

Code 0 (no route to destination):
Set Code to 1 (host unreachable).

Code 1 (communication with destination administratively prohibited):
Set Code to 10 (communication with destination host administratively prohibited).

Code 2 (beyond scope of source address):
Set Code to 1 (host unreachable). Note that this error is very unlikely since the IPv4-translatable source address is considered to have global scope.

Code 3 (address unreachable):
Set Code to 1 (host unreachable).

Code 4 (port unreachable):
Set Code to 3 (port unreachable).

Packet Too Big (Type 2)
Translate to an ICMPv4 Destination Unreachable with code 4. The MTU field needs to be adjusted for the difference between the IPv4 and IPv6 header sizes taking into account whether or not the packet in error includes a Fragment header.

Time Exceeded (Type 3)
Set the Type to 11. The Code field is unchanged.

```
        Parameter Problem (Type 4)
           If the Code is 1 translate this to an ICMPv4 protocol
           unreachable (Type 3, Code 2).  Otherwise set the Type to 12
           and the Code to zero.  The Pointer needs to be updated to
           point to the corresponding field in the translated include IP
           header.

        Unknown error messages
           Silently drop.
```

Identifier: RQ_003_3121
RFC Clause: 4.2
Type: Mandatory
Applies to: Router

Requirement:
In order to translate ICMPv6 error messages, for Destination Unreachable (Type 1) messages Translate the Code Field, Code 3 (address unreachable), to 1 (host unreachable).

Specification Text:
The actions needed to translate various ICMPv6 messages are:

```
      ICMPv6 informational messages:

        Echo Request and Echo Reply (Type 128 and 129)
           Adjust the type to 0 and 8, respectively, and adjust the ICMP
           checksum both to take the type change into account and to
           exclude the ICMPv6 pseudo-header.

        MLD Multicast Listener Query/Report/Done (Type 130, 131, 132)
           Single hop message.  Silently drop.
        Neighbor Discover messages (Type 133 through 137)
           Single hop message.  Silently drop.

        Unknown informational messages
           Silently drop.

      ICMPv6 error messages:

        Destination Unreachable (Type 1)
           Set the Type field to 3.  Translate the code field as
           follows:
              Code 0 (no route to destination):
                     Set Code to 1 (host unreachable).

              Code 1 (communication with destination administratively
              prohibited):
                     Set Code to 10 (communication with destination host
                     administratively prohibited).

              Code 2 (beyond scope of source address):
                     Set Code to 1 (host unreachable).  Note that this
                     error is very unlikely since the IPv4-translatable
                     source address is considered to have global scope.

              Code 3 (address unreachable):
                     Set Code to 1 (host unreachable).

              Code 4 (port unreachable):
                     Set Code to 3 (port unreachable).

        Packet Too Big (Type 2)
           Translate to an ICMPv4 Destination Unreachable with code 4.
           The MTU field needs to be adjusted for the difference between
           the IPv4 and IPv6 header sizes taking into account whether or
           not the packet in error includes a Fragment header.

        Time Exceeded (Type 3)
           Set the Type to 11.  The Code field is unchanged.
        Parameter Problem (Type 4)
           If the Code is 1 translate this to an ICMPv4 protocol
           unreachable (Type 3, Code 2).  Otherwise set the Type to 12
           and the Code to zero.  The Pointer needs to be updated to
           point to the corresponding field in the translated include IP
           header.

        Unknown error messages
           Silently drop.
```

Identifier:	RQ_003_3122
RFC Clause:	4.2
Type:	Mandatory
Applies to:	Router

Requirement:

In order to translate ICMPv6 error messages, for Destination Unreachable (Type 1) messages
Translate the Code Field, Code 4 (port unreachable), to 3 (port unreachable).

Specification Text:

The actions needed to translate various ICMPv6 messages are:

```
      ICMPv6 informational messages:

        Echo Request and Echo Reply (Type 128 and 129)
           Adjust the type to 0 and 8, respectively, and adjust the ICMP
           checksum both to take the type change into account and to
           exclude the ICMPv6 pseudo-header.

        MLD Multicast Listener Query/Report/Done (Type 130, 131, 132)
           Single hop message.  Silently drop.
        Neighbor Discover messages (Type 133 through 137)
           Single hop message.  Silently drop.

        Unknown informational messages
           Silently drop.

      ICMPv6 error messages:

        Destination Unreachable (Type 1)
           Set the Type field to 3.  Translate the code field as
           follows:
              Code 0 (no route to destination):
                     Set Code to 1 (host unreachable).

              Code 1 (communication with destination administratively
              prohibited):
                     Set Code to 10 (communication with destination host
                     administratively prohibited).

              Code 2 (beyond scope of source address):
                     Set Code to 1 (host unreachable).  Note that this
                     error is very unlikely since the IPv4-translatable
                     source address is considered to have global scope.

              Code 3 (address unreachable):
                     Set Code to 1 (host unreachable).

              Code 4 (port unreachable):
                     Set Code to 3 (port unreachable).

        Packet Too Big (Type 2)
           Translate to an ICMPv4 Destination Unreachable with code 4.
           The MTU field needs to be adjusted for the difference between
           the IPv4 and IPv6 header sizes taking into account whether or
           not the packet in error includes a Fragment header.

        Time Exceeded (Type 3)
           Set the Type to 11.  The Code field is unchanged.
```

Parameter Problem (Type 4)

If the Code is 1 translate this to an ICMPv4 protocol unreachable (Type 3, Code 2). Otherwise set the Type to 12 and the Code to zero. The Pointer needs to be updated to point to the corresponding field in the translated include IP header.

Unknown error messages

Silently drop.

Identifier:	RQ_003_3123
RFC Clause:	4.2
Type:	Mandatory
Applies to:	Router

Requirement:

In order to translate ICMPv6 error messages, for Packet Too Big (Type 2) messages Translate to an ICMPv4 Destination Unreachable with code 4.

Specification Text:

The actions needed to translate various ICMPv6 messages are:

ICMPv6 informational messages:

Echo Request and Echo Reply (Type 128 and 129)

Adjust the type to 0 and 8, respectively, and adjust the ICMP checksum both to take the type change into account and to exclude the ICMPv6 pseudo-header.

MLD Multicast Listener Query/Report/Done (Type 130, 131, 132)

Single hop message. Silently drop.

Neighbor Discover messages (Type 133 through 137)

Single hop message. Silently drop.

Unknown informational messages

Silently drop.

ICMPv6 error messages:

Destination Unreachable (Type 1)

Set the Type field to 3. Translate the code field as follows:

Code 0 (no route to destination):

Set Code to 1 (host unreachable).

Code 1 (communication with destination administratively prohibited):

Set Code to 10 (communication with destination host administratively prohibited).

Code 2 (beyond scope of source address):

Set Code to 1 (host unreachable). Note that this error is very unlikely since the IPv4-translatable source address is considered to have global scope.

Code 3 (address unreachable):

Set Code to 1 (host unreachable).

Code 4 (port unreachable):

Set Code to 3 (port unreachable).

Packet Too Big (Type 2)

Translate to an ICMPv4 Destination Unreachable with code 4.

The MTU field needs to be adjusted for the difference between the IPv4 and IPv6 header sizes taking into account whether or not the packet in error includes a Fragment header.

Time Exceeded (Type 3)

Set the Type to 11. The Code field is unchanged.

```
        Parameter Problem (Type 4)
           If the Code is 1 translate this to an ICMPv4 protocol
           unreachable (Type 3, Code 2).  Otherwise set the Type to 12
           and the Code to zero.  The Pointer needs to be updated to
           point to the corresponding field in the translated include IP
           header.

        Unknown error messages
           Silently drop.
```

Identifier:	RQ_003_3124
RFC Clause:	4.2
Type:	Mandatory
Applies to:	Router

Requirement:

When translating an ICMPv6 error messages, for Packet Too Big (Type 2) messages to an ICMPv4 Destination Unreachable with code 4, the MTU field needs to be adjusted for the difference between the IPv4 and IPv6 header sizes taking into account whether or not the packet in error includes a Fragment header.

Specification Text:

The actions needed to translate various ICMPv6 messages are:

```
      ICMPv6 informational messages:

        Echo Request and Echo Reply (Type 128 and 129)
           Adjust the type to 0 and 8, respectively, and adjust the ICMP
           checksum both to take the type change into account and to
           exclude the ICMPv6 pseudo-header.

        MLD Multicast Listener Query/Report/Done (Type 130, 131, 132)
           Single hop message.  Silently drop.
        Neighbor Discover messages (Type 133 through 137)
           Single hop message.  Silently drop.

        Unknown informational messages
           Silently drop.

      ICMPv6 error messages:

        Destination Unreachable (Type 1)
           Set the Type field to 3.  Translate the code field as
           follows:
              Code 0 (no route to destination):
                     Set Code to 1 (host unreachable).

              Code 1 (communication with destination administratively
              prohibited):
                     Set Code to 10 (communication with destination host
                     administratively prohibited).

              Code 2 (beyond scope of source address):
                     Set Code to 1 (host unreachable).  Note that this
                     error is very unlikely since the IPv4-translatable
                     source address is considered to have global scope.

              Code 3 (address unreachable):
                     Set Code to 1 (host unreachable).

              Code 4 (port unreachable):
                     Set Code to 3 (port unreachable).

        Packet Too Big (Type 2)
           Translate to an ICMPv4 Destination Unreachable with code 4.
           The MTU field needs to be adjusted for the difference between
           the IPv4 and IPv6 header sizes taking into account whether or
           not the packet in error includes a Fragment header.

        Time Exceeded (Type 3)
           Set the Type to 11.  The Code field is unchanged.
```

```
     Parameter Problem (Type 4)
        If the Code is 1 translate this to an ICMPv4 protocol
        unreachable (Type 3, Code 2).  Otherwise set the Type to 12
        and the Code to zero.  The Pointer needs to be updated to
        point to the corresponding field in the translated include IP
        header.

     Unknown error messages
        Silently drop.

----------------
```

Identifier: RQ_003_3125
RFC Clause: 4.2
Type: Mandatory
Applies to: Router

Requirement:

In order to translate ICMPv6 error messages, for Time Exceeded (Type 3) messages set the Type to 11.

Specification Text:

The actions needed to translate various ICMPv6 messages are:

```
   ICMPv6 informational messages:

     Echo Request and Echo Reply (Type 128 and 129)
        Adjust the type to 0 and 8, respectively, and adjust the ICMP
        checksum both to take the type change into account and to
        exclude the ICMPv6 pseudo-header.

     MLD Multicast Listener Query/Report/Done (Type 130, 131, 132)
        Single hop message.  Silently drop.
     Neighbor Discover messages (Type 133 through 137)
        Single hop message.  Silently drop.

     Unknown informational messages
        Silently drop.

   ICMPv6 error messages:

     Destination Unreachable (Type 1)
        Set the Type field to 3.  Translate the code field as
        follows:
           Code 0 (no route to destination):
                  Set Code to 1 (host unreachable).

           Code 1 (communication with destination administratively
           prohibited):
                  Set Code to 10 (communication with destination host
                  administratively prohibited).

           Code 2 (beyond scope of source address):
                  Set Code to 1 (host unreachable).  Note that this
                  error is very unlikely since the IPv4-translatable
                  source address is considered to have global scope.

           Code 3 (address unreachable):
                  Set Code to 1 (host unreachable).

           Code 4 (port unreachable):
                  Set Code to 3 (port unreachable).

     Packet Too Big (Type 2)
        Translate to an ICMPv4 Destination Unreachable with code 4.
        The MTU field needs to be adjusted for the difference between
        the IPv4 and IPv6 header sizes taking into account whether or
        not the packet in error includes a Fragment header.

     Time Exceeded (Type 3)
        Set the Type to 11.  The Code field is unchanged.

     Parameter Problem (Type 4)
        If the Code is 1 translate this to an ICMPv4 protocol
        unreachable (Type 3, Code 2).  Otherwise set the Type to 12
        and the Code to zero.  The Pointer needs to be updated to
        point to the corresponding field in the translated include IP
        header.
     Unknown error messages
        Silently drop.
```

Identifier: RQ_003_3126
RFC Clause: 4.2
Type: Mandatory
Applies to: Router

Requirement:

In order to translate ICMPv6 error messages, for Parameter Problem (Type 4) messages if the Code is 1 translate this to an ICMPv4 protocol unreachable (Type 3, Code 2).

Specification Text:

The actions needed to translate various ICMPv6 messages are:

```
      ICMPv6 informational messages:

        Echo Request and Echo Reply (Type 128 and 129)
           Adjust the type to 0 and 8, respectively, and adjust the ICMP
           checksum both to take the type change into account and to
           exclude the ICMPv6 pseudo-header.

        MLD Multicast Listener Query/Report/Done (Type 130, 131, 132)
           Single hop message.  Silently drop.
        Neighbor Discover messages (Type 133 through 137)
           Single hop message.  Silently drop.

        Unknown informational messages
           Silently drop.

      ICMPv6 error messages:

        Destination Unreachable (Type 1)
           Set the Type field to 3.  Translate the code field as
           follows:
              Code 0 (no route to destination):
                     Set Code to 1 (host unreachable).

              Code 1 (communication with destination administratively
              prohibited):
                     Set Code to 10 (communication with destination host
                     administratively prohibited).

              Code 2 (beyond scope of source address):
                     Set Code to 1 (host unreachable).  Note that this
                     error is very unlikely since the IPv4-translatable
                     source address is considered to have global scope.

              Code 3 (address unreachable):
                     Set Code to 1 (host unreachable).

              Code 4 (port unreachable):
                     Set Code to 3 (port unreachable).

        Packet Too Big (Type 2)
           Translate to an ICMPv4 Destination Unreachable with code 4.
           The MTU field needs to be adjusted for the difference between
           the IPv4 and IPv6 header sizes taking into account whether or
           not the packet in error includes a Fragment header.

        Time Exceeded (Type 3)
           Set the Type to 11.  The Code field is unchanged.

        Parameter Problem (Type 4)
           If the Code is 1 translate this to an ICMPv4 protocol
           unreachable (Type 3, Code 2).  Otherwise set the Type to 12
           and the Code to zero.  The Pointer needs to be updated to
           point to the corresponding field in the translated include IP
           header.

        Unknown error messages
           Silently drop.
```

Identifier: RQ_003_3127
RFC Clause: 4.2
Type: Mandatory
Applies to: Router

Requirement:
In order to translate ICMPv6 error messages, for Parameter Problem (Type 4) messages if the Code is NOT 1 set the Type to 12 and the Code to zero.

Specification Text:

```
The actions needed to translate various ICMPv6 messages are:

      ICMPv6 informational messages:

        Echo Request and Echo Reply (Type 128 and 129)
           Adjust the type to 0 and 8, respectively, and adjust the ICMP
           checksum both to take the type change into account and to
           exclude the ICMPv6 pseudo-header.

        MLD Multicast Listener Query/Report/Done (Type 130, 131, 132)
           Single hop message.  Silently drop.
        Neighbor Discover messages (Type 133 through 137)
           Single hop message.  Silently drop.

        Unknown informational messages
           Silently drop.

      ICMPv6 error messages:

        Destination Unreachable (Type 1)
           Set the Type field to 3.  Translate the code field as
           follows:
              Code 0 (no route to destination):
                     Set Code to 1 (host unreachable).

              Code 1 (communication with destination administratively
              prohibited):
                     Set Code to 10 (communication with destination host
                     administratively prohibited).

              Code 2 (beyond scope of source address):
                     Set Code to 1 (host unreachable).  Note that this
                     error is very unlikely since the IPv4-translatable
                     source address is considered to have global scope.

              Code 3 (address unreachable):
                     Set Code to 1 (host unreachable).

              Code 4 (port unreachable):
                     Set Code to 3 (port unreachable).

        Packet Too Big (Type 2)
           Translate to an ICMPv4 Destination Unreachable with code 4.
           The MTU field needs to be adjusted for the difference between
           the IPv4 and IPv6 header sizes taking into account whether or
           not the packet in error includes a Fragment header.

        Time Exceeded (Type 3)
           Set the Type to 11.  The Code field is unchanged.

        Parameter Problem (Type 4)
           If the Code is 1 translate this to an ICMPv4 protocol
           unreachable (Type 3, Code 2).  Otherwise set the Type to 12
           and the Code to zero.  The Pointer needs to be updated to
           point to the corresponding field in the translated include IP
           header.

        Unknown error messages
           Silently drop.
```

Identifier: RQ_003_3128
RFC Clause: 4.2
Type: Mandatory
Applies to: Router

Requirement:

In order to translate ICMPv6 error messages, for Parameter Problem (Type 4) messages the Pointer needs to be updated to point to the corresponding field in the translated include IP header.

Specification Text:

The actions needed to translate various ICMPv6 messages are:

ICMPv6 informational messages:

Echo Request and Echo Reply (Type 128 and 129)
Adjust the type to 0 and 8, respectively, and adjust the ICMP checksum both to take the type change into account and to exclude the ICMPv6 pseudo-header.

MLD Multicast Listener Query/Report/Done (Type 130, 131, 132)
Single hop message. Silently drop.
Neighbor Discover messages (Type 133 through 137)
Single hop message. Silently drop.

Unknown informational messages
Silently drop.

ICMPv6 error messages:

Destination Unreachable (Type 1)
Set the Type field to 3. Translate the code field as follows:

Code 0 (no route to destination):
Set Code to 1 (host unreachable).

Code 1 (communication with destination administratively prohibited):
Set Code to 10 (communication with destination host administratively prohibited).

Code 2 (beyond scope of source address):
Set Code to 1 (host unreachable). Note that this error is very unlikely since the IPv4-translatable source address is considered to have global scope.

Code 3 (address unreachable):
Set Code to 1 (host unreachable).

Code 4 (port unreachable):
Set Code to 3 (port unreachable).

Packet Too Big (Type 2)
Translate to an ICMPv4 Destination Unreachable with code 4. The MTU field needs to be adjusted for the difference between the IPv4 and IPv6 header sizes taking into account whether or not the packet in error includes a Fragment header.

Time Exceeded (Type 3)
Set the Type to 11. The Code field is unchanged.

Parameter Problem (Type 4)

If the Code is 1 translate this to an ICMPv4 protocol unreachable (Type 3, Code 2). Otherwise set the Type to 12 and the Code to zero. **The Pointer needs to be updated to point to the corresponding field in the translated include IP header.**

Unknown error messages
Silently drop.

Identifier: RQ_003_3129
RFC Clause: 4.2
Type: Mandatory
Applies to: Router

Requirement:
In order to translate ICMPv6 error messages, silently drop Unknown error messages.

.

Specification Text:
The actions needed to translate various ICMPv6 messages are:

ICMPv6 informational messages:

Echo Request and Echo Reply (Type 128 and 129)
Adjust the type to 0 and 8, respectively, and adjust the ICMP checksum both to take the type change into account and to exclude the ICMPv6 pseudo-header.

MLD Multicast Listener Query/Report/Done (Type 130, 131, 132)
Single hop message. Silently drop.
Neighbor Discover messages (Type 133 through 137)
Single hop message. Silently drop.

Unknown informational messages
Silently drop.

ICMPv6 error messages:

Destination Unreachable (Type 1)
Set the Type field to 3. Translate the code field as follows:
Code 0 (no route to destination):
Set Code to 1 (host unreachable).

Code 1 (communication with destination administratively prohibited):
Set Code to 10 (communication with destination host administratively prohibited).

Code 2 (beyond scope of source address):
Set Code to 1 (host unreachable). Note that this error is very unlikely since the IPv4-translatable source address is considered to have global scope.

Code 3 (address unreachable):
Set Code to 1 (host unreachable).

Code 4 (port unreachable):
Set Code to 3 (port unreachable).

Packet Too Big (Type 2)
Translate to an ICMPv4 Destination Unreachable with code 4. The MTU field needs to be adjusted for the difference between the IPv4 and IPv6 header sizes taking into account whether or not the packet in error includes a Fragment header.

Time Exceeded (Type 3)
Set the Type to 11. The Code field is unchanged.

Parameter Problem (Type 4)

If the Code is 1 translate this to an ICMPv4 protocol unreachable (Type 3, Code 2). Otherwise set the Type to 12 and the Code to zero. The Pointer needs to be updated to point to the corresponding field in the translated include IP header.

Unknown error messages
Silently drop.

Identifier: RQ_003_3130
RFC Clause: 4.4
Type: Mandatory
Applies to: Router

Requirement:
When the translator receives an IPv6 packet with an IPv4-mapped destination address the packet will be translated to IPv4.

Specification Text:
When the translator receives an IPv6 packet with an IPv4-mapped destination address the packet will be translated to IPv4.

Identifier: RQ_003_3133
RFC Clause: 5
Type: Mandatory
Applies to: Host

Requirement:
The application protocols need to handle operation on a dual stack node.

Specification Text:
As specified in Section 1.3 the application protocols need to handle operation on a dual stack node.
In addition the protocol stack needs to be able to:

```
   o  Determine when an IPv4-translatable address needs to be allocated
      and the allocation needs to be refreshed/renewed.  This can
      presumably be done without involving the applications by e.g.
      handling this under the socket API.  For instance, when the
      connect or sendto socket calls are invoked they could check if the
      destination is an IPv4-mapped address and in that case
      allocate/refresh the IPv4-translatable address.

   o  Ensure, as part of the source address selection mechanism, that
      when the destination address is an IPv4-mapped address the source
      address MUST be an IPv4-translatable address.  And an IPv4-
      translatable address MUST NOT be used with other forms of IPv6
      destination addresses.

   o  Should the peer have AAAA/A6 address records the application (or
      resolver) SHOULD never fall back to looking for A address records
      even if communication fails using the available AAAA/A6 records.
      The reason for this restriction is to prevent traffic between two
      IPv6 nodes (which AAAA/A6 records in the DNS) from accidentally
      going through SIIT translators twice; from IPv6 to IPv4 and to
      IPv6 again.  It is considered preferable to instead signal a
      failure to communicate to the application.
```

Identifier: RQ_003_3134
RFC Clause: 5
Type: Mandatory
Applies to: Host

Requirement:
The protocol stack needs to be able to determine when an IPv4-translatable address needs to be allocated.

Specification Text:
As specified in Section 1.3 the application protocols need to handle operation on a dual stack node.
In addition the protocol stack needs to be able to:

```
   o  Determine when an IPv4-translatable address needs to be allocated
      and the allocation needs to be refreshed/renewed.  This can
      presumably be done without involving the applications by e.g.
      handling this under the socket API.  For instance, when the
      connect or sendto socket calls are invoked they could check if the
      destination is an IPv4-mapped address and in that case
      allocate/refresh the IPv4-translatable address.
```

o Ensure, as part of the source address selection mechanism, that when the destination address is an IPv4-mapped address the source address MUST be an IPv4-translatable address. And an IPv4-translatable address MUST NOT be used with other forms of IPv6 destination addresses.

o Should the peer have AAAA/A6 address records the application (or resolver) SHOULD never fall back to looking for A address records even if communication fails using the available AAAA/A6 records. The reason for this restriction is to prevent traffic between two IPv6 nodes (which AAAA/A6 records in the DNS) from accidentally going through SIIT translators twice; from IPv6 to IPv4 and to IPv6 again. It is considered preferable to instead signal a failure to communicate to the application.

Identifier: RQ_003_3135
RFC Clause: 5
Type: Mandatory
Applies to: Host

Requirement:
The protocol stack needs to be able to determine when an IPv4-translatable address allocation needs to be refreshed/renewed.

Specification Text:
As specified in Section 1.3 the application protocols need to handle operation on a dual stack node. In addition the protocol stack needs to be able to:

o **Determine when an IPv4-translatable address needs to be** allocated and the allocation needs to be **refreshed/renewed.** This can presumably be done without involving the applications by e.g. handling this under the socket API. For instance, when the connect or sendto socket calls are invoked they could check if the destination is an IPv4-mapped address and in that case allocate/refresh the IPv4-translatable address.

o Ensure, as part of the source address selection mechanism, that when the destination address is an IPv4-mapped address the source address MUST be an IPv4-translatable address. And an IPv4-translatable address MUST NOT be used with other forms of IPv6 destination addresses.

o Should the peer have AAAA/A6 address records the application (or resolver) SHOULD never fall back to looking for A address records even if communication fails using the available AAAA/A6 records. The reason for this restriction is to prevent traffic between two IPv6 nodes (which AAAA/A6 records in the DNS) from accidentally going through SIIT translators twice; from IPv6 to IPv4 and to IPv6 again. It is considered preferable to instead signal a failure to communicate to the application.

Identifier: RQ_003_3136
RFC Clause: 5
Type: Mandatory
Applies to: Host

Requirement:
The protocol stack needs to ensure, as part of the source address selection mechanism, that when the destination address is an IPv4-mapped address the source address MUST be an IPv4-translatable address.

Specification Text:

As specified in Section 1.3 the application protocols need to handle operation on a dual stack node. In addition the protocol stack needs to be able to:

- Determine when an IPv4-translatable address needs to be allocated and the allocation needs to be refreshed/renewed. This can presumably be done without involving the applications by e.g. handling this under the socket API. For instance, when the connect or sendto socket calls are invoked they could check if the destination is an IPv4-mapped address and in that case allocate/refresh the IPv4-translatable address.

- **Ensure, as part of the source address selection mechanism, that when the destination address is an IPv4-mapped address the source address MUST be an IPv4-translatable address.** And an IPv4-translatable address MUST NOT be used with other forms of IPv6 destination addresses.

- Should the peer have AAAA/A6 address records the application (or resolver) SHOULD never fall back to looking for A address records even if communication fails using the available AAAA/A6 records. The reason for this restriction is to prevent traffic between two IPv6 nodes (which AAAA/A6 records in the DNS) from accidentally going through SIIT translators twice; from IPv6 to IPv4 and to IPv6 again. It is considered preferable to instead signal a failure to communicate to the application.

Identifier:	RQ_003_3137
RFC Clause:	5
Type:	Mandatory
Applies to:	Host

Requirement:

An IPv4-translatable address MUST NOT be used with other forms of IPv6 destination addresses.

Specification Text:

As specified in Section 1.3 the application protocols need to handle operation on a dual stack node. In addition the protocol stack needs to be able to:

- Determine when an IPv4-translatable address needs to be allocated and the allocation needs to be refreshed/renewed. This can presumably be done without involving the applications by e.g. handling this under the socket API. For instance, when the connect or sendto socket calls are invoked they could check if the destination is an IPv4-mapped address and in that case allocate/refresh the IPv4-translatable address.

- Ensure, as part of the source address selection mechanism, that when the destination address is an IPv4-mapped address the source address MUST be an IPv4-translatable address. **And an IPv4-translatable address MUST NOT be used with other forms of IPv6 destination addresses.**

- Should the peer have AAAA/A6 address records the application (or resolver) SHOULD never fall back to looking for A address records even if communication fails using the available AAAA/A6 records. The reason for this restriction is to prevent traffic between two IPv6 nodes (which AAAA/A6 records in the DNS) from accidentally going through SIIT translators twice; from IPv6 to IPv4 and to IPv6 again. It is considered preferable to instead signal a failure to communicate to the application.

Identifier:	RQ_003_3138
RFC Clause:	5
Type:	Recommendation
Applies to:	Host

Requirement:

Should the peer have AAAA/A6 address records the application (or resolver) SHOULD never fall back to looking for A address records even if communication fails using the available AAAA/A6 records.

Specification Text:

As specified in Section 1.3 the application protocols need to handle operation on a dual stack node. In addition the protocol stack needs to be able to:

o Determine when an IPv4-translatable address needs to be allocated and the allocation needs to be refreshed/renewed. This can presumably be done without involving the applications by e.g. handling this under the socket API. For instance, when the connect or sendto socket calls are invoked they could check if the destination is an IPv4-mapped address and in that case allocate/refresh the IPv4-translatable address.

o Ensure, as part of the source address selection mechanism, that when the destination address is an IPv4-mapped address the source address MUST be an IPv4-translatable address. And an IPv4-translatable address MUST NOT be used with other forms of IPv6 destination addresses.

o **Should the peer have AAAA/A6 address records the application (or resolver) SHOULD never fall back to looking for A address records even if communication fails using the available AAAA/A6 records.** The reason for this restriction is to prevent traffic between two IPv6 nodes (which AAAA/A6 records in the DNS) from accidentally going through SIIT translators twice; from IPv6 to IPv4 and to IPv6 again. It is considered preferable to instead signal a failure to communicate to the application.

Identifier:	RQ_003_3139
RFC Clause:	5
Type:	Recommendation
Applies to:	Host

Requirement:

Should the peer have AAAA/A6 address records, if communication fails using the available AAAA/A6 records, the application (or resolver) SHOULD signal a failure to communicate to the application.

Specification Text:

As specified in Section 1.3 the application protocols need to handle operation on a dual stack node. In addition the protocol stack needs to be able to:

o Determine when an IPv4-translatable address needs to be allocated and the allocation needs to be refreshed/renewed. This can presumably be done without involving the applications by e.g. handling this under the socket API. For instance, when the connect or sendto socket calls are invoked they could check if the destination is an IPv4-mapped address and in that case allocate/refresh the IPv4-translatable address.

o Ensure, as part of the source address selection mechanism, that when the destination address is an IPv4-mapped address the source address MUST be an IPv4-translatable address. And an IPv4-translatable address MUST NOT be used with other forms of IPv6 destination addresses.

o **Should the peer have AAAA/A6 address records the application (or resolver**) SHOULD never fall back to looking for A address records even if communication fails using the available AAAA/A6 records. The reason for this restriction is to prevent traffic between two IPv6 nodes (which AAAA/A6 records in the DNS) from accidentally going through SIIT translators twice; from IPv6 to IPv4 and to IPv6 again. **It is considered preferable to instead signal a failure to communicate to the application.**

Requirements extracted from RFC 2766

Identifier: RQ_003_6001
RFC Clause: 1
Type: Recommendation
Applies to: Router

Requirement:
NAT Protocol Translation (NAT-PT) is only to be use when no other native IPv6 or IPv6 over IPv4 tunneled means of communication is possible.

Specification Text:
A fundamental assumption for NAT-PT is only to be use when no other native IPv6 or IPv6 over IPv4 tunneled means of communication is possible. In other words the aim is to only use translation between IPv6 only nodes and IPv4 only nodes, while translation between IPv6 only nodes and the IPv4 part of a dual stack node should be avoided over other alternatives.

Identifier: RQ_003_6002
RFC Clause: 2.2.2
Type: Mandatory
Applies to: Router

Requirement:
If Bi-Directional-NAT-PT is implemented, a DNS-ALG [RFC2694] MUST be employed to facilitate name to address mapping.

Specification Text:
With Bi-directional-NAT-PT, sessions can be initiated from hosts in V4 network as well as the V6 network. V6 network addresses are bound to V4 addresses, statically or dynamically as connections are established in either direction. The name space (i.e., their Fully Qualified Domain Names) between hosts in V4 and V6 networks is assumed to be end-to-end unique. Hosts in V4 realm access V6-realm hosts by using DNS for address resolution. **A DNS-ALG [DNS-ALG] MUST be employed in conjunction with Bi-Directional-NAT-PT to facilitate name to address mapping**. Specifically, the DNS-ALG MUST be capable of translating V6 addresses in DNS Queries and responses into their V4-address bindings, and vice versa, as DNS packets traverse between V6 and V4 realms.

Identifier: RQ_003_6003
RFC Clause: 2.2.2
Type: Mandatory
Applies to: Router

Requirement:
When Bi-Directional-NAT-PT is implemented,the DNS-ALG MUST be capable of translating V6 addresses in DNS Queries and responses into their V4-address bindings, and vice versa, as DNS packets traverse between V6 and V4 realms.

Specification Text:
With Bi-directional-NAT-PT, sessions can be initiated from hosts in V4 network as well as the V6 network. V6 network addresses are bound to V4 addresses, statically or dynamically as connections are established in either direction. The name space (i.e., their Fully Qualified Domain Names) between hosts in V4 and V6 networks is assumed to be end-to-end unique. Hosts in V4 realm access V6-realm hosts by using DNS for address resolution. A DNS-ALG [DNS-ALG] MUST be employed in conjunction with Bi-Directional-NAT-PT to facilitate name to address mapping. **Specifically, the DNS-ALG MUST be capable of translating V6 addresses in DNS Queries and responses into their V4-address bindings, and vice versa, as DNS packets traverse between V6 and V4 realms.**

Identifier: RQ_003_6004
RFC Clause: 3.1
Type: Mandatory
Applies to: Router

Requirement:
When implementing Basic NAT Protocol Translation (Basic-NAT-PT), if the V6 network has less V4 addresses than V6 end nodes, dynamic address allocation is REQUIRED for at least some of them.

Specification Text:
The V4 addresses in the address pool could be allocated one-to-one to the V6 addresses of the V6 end nodes in which case one needs as many V4 addresses as V6 end points. **In this document we assume that the V6 network has less V4 addresses than V6 end nodes and thus dynamic address allocation is REQUIRED for at least some of them.**

Identifier: RQ_003_6005
RFC Clause: 3.1
Type: Mandatory
Applies to: Router

Requirement:
When implementing Basic NAT Protocol Translation (Basic-NAT-PT), the prefix PREFIX::/96 SHALL be advertised in the stub domain by the NAT-PT.

Specification Text:
NOTE: **The prefix PREFIX::/96 is advertised in the stub domain by the NAT-PT, and packets addressed to this PREFIX will be routed to the NAT-PT. The pre-configured PREFIX only needs to be routable within the IPv6 stub domain and as such it can be any routable prefix that the network administrator chooses.**

Identifier: RQ_003_6006
RFC Clause: 3.1
Type: Recommendation
Applies to: Router

Requirement:
When implementing Basic NAT Protocol Translation (Basic-NAT-PT), if the outgoing packet is not a session initialisation packet and the NAT-PT does not already have stored some state about the related session, including assigned IPv4 address and other parameters for the translation, the packet SHOULD be silently discarded.

Specification Text:
If the outgoing packet is not a session initialisation packet, the NAT-PT SHOULD already have stored some state about the related session, including assigned IPv4 address and other parameters for the translation. If this state does not exist, the packet SHOULD be silently discarded.

Identifier: RQ_003_6007
RFC Clause: 3.1
Type: Mandatory
Applies to: Router

Requirement:
When implementing Basic NAT Protocol Translation (Basic-NAT-PT), if the outgoing packet is a session initialisation packet, the NAT-PT SHALL locally allocate an address from its pool of addresses.

Specification Text:
If the packet is a session initialisation packet, the NAT-PT locally allocates an address (e.g: 120.130.26.10) from its pool of addresses and the packet is translated to IPv4. The translation parameters are cached for the duration of the session and the IPv6 to IPv4 mapping is retained by NAT-PT.

Identifier: RQ_003_6008
RFC Clause: 3.1
Type: Mandatory
Applies to: Router

Requirement:
When implementing Basic NAT Protocol Translation (Basic-NAT-PT), if the outgoing packet is a session initialisation packet, the NAT-PT SHALL translate the packet to IPv4.

Specification Text:
If the packet is a session initialisation packet, the NAT-PT locally allocates an address (e.g: 120.130.26.10) from its pool of addresses **and the packet is translated to IPv4.** The translation parameters are cached for the duration of the session and the IPv6 to IPv4 mapping is retained by NAT-PT.

Identifier: RQ_003_6009
RFC Clause: 3.1
Type: Mandatory
Applies to: Router

Requirement:
When implementing Basic NAT Protocol Translation (Basic-NAT-PT), if the outgoing packet is a session initialisation packet, the translation parameters SHALL BE cached for the duration of the session.

Specification Text:
If the packet is a session initialisation packet, the NAT-PT locally allocates an address (e.g: 120.130.26.10) from its pool of addresses and the packet is translated to IPv4. The translation parameters are cached for the duration of the session and the IPv6 to IPv4 mapping is retained by NAT-PT.

Identifier: RQ_003_6010
RFC Clause: 3.1
Type: Mandatory
Applies to: Router

Requirement:
When implementing Basic NAT Protocol Translation (Basic-NAT-PT), if the outgoing packet is a session initialisation packet, the IPv6 to IPv4 mapping SHALL BE retained by NAT-PT.

Specification Text:
If the packet is a session initialisation packet, the NAT-PT locally allocates an address (e.g: 120.130.26.10) from its pool of addresses and the packet is translated to IPv4. The translation parameters are cached for the duration of the session and **the IPv6 to IPv4 mapping is retained by NAT-PT.**

Identifier: RQ_003_6011
RFC Clause: 3.1
Type: Mandatory
Applies to: Router

Requirement:
When implementing Basic NAT Protocol Translation (Basic-NAT-PT), the NAT-PT shall examine the SA and DA of any returning traffic to determine if it belongs to the same session.

Specification Text:
The resulting IPv4 packet has SA=120.130.26.10 and DA=132.146.243.30. Any returning traffic will be recognised as belonging to the same session by NAT-PT. NAT-PT will use the state information to translate the packet, and the resulting addresses will be SA=PREFIX::132.146.243.30, DA=FEDC:BA98::7654:3210. Note that this packet can now be routed inside the IPv6-only stub network as normal.

Identifier: RQ_003_6012
RFC Clause: 3.1
Type: Mandatory
Applies to: Router

Requirement:
When implementing Basic NAT Protocol Translation (Basic-NAT-PT), the NAT-PT SHALL use the state information to translate returning packets belonging to the same session.

Specification Text:
The resulting IPv4 packet has SA=120.130.26.10 and DA=132.146.243.30. Any returning traffic will be recognised as belonging to the same session by NAT-PT. **NAT-PT will use the state information to translate the packet, and the resulting addresses will be SA=PREFIX::132.146.243.30, DA=FEDC:BA98::7654:3210**. Note that this packet can now be routed inside the IPv6-only stub network as normal.

Identifier: RQ_003_6013
RFC Clause: 3.2
Type: Mandatory
Applies to: Router

Requirement:
When implementing Network Address Port Translation + Protocol Translation (NAPT-PT), between IPv6 and IPv4 realms, on receipt of a local IPv6 packet, the NAPT-PT SHALL assign one of the TCP ports from the assigned V4 address to translate the tuple of (Source Address, Source TCP port)

Specification Text:
IPv6 Node A would establish a TCP session with the IPv4 Node C as follows:

Node A creates a packet with:

Source Address, SA=FEDC:BA98::7654:3210 , source TCP port = 3017 and Destination Address, DA = PREFIX::132.146.243.30, destination TCP port = 23.

When the packet reaches the NAPT-PT box, NAPT-PT would assign one of the TCP ports from the assigned V4 address to translate the tuple of (Source Address, Source TCP port) as follows:

```
  SA=120.130.26.10, source TCP port = 1025 and
  DA=132.146.243.30, destination TCP port = 23.
```

The returning traffic from 132.146.243.30, TCP port 23 will be recognised as belonging to the same session and will be translated back to V6 as follows:

```
  SA = PREFIX::132.146.243.30, source TCP port = 23;
  DA = FEDC:BA98::7654:3210 , destination TCP port = 3017
```

Inbound NAPT-PT sessions are restricted to one server per service, assigned via static TCP/UDP port mapping. For example, the Node [IPv6-A] in figure 1 may be the only HTTP server (port 80) in the V6 domain. Node [IPv4-C] sends a packet:

```
  SA=132.146.243.30, source TCP port = 1025  and
  DA=120.130.26.10, destination TCP port = 80
```

NAPT-PT will translate this packet to:

```
  SA=PREFIX::132.146.243.30, source TCP port = 1025
  DA=FEDC:BA98::7654:3210, destination TCP port = 80
```

Identifier: RQ_003_6014
RFC Clause: 3.2
Type: Mandatory
Applies to: Router

Requirement:
When implementing Network Address Port Translation + Protocol Translation (NAPT-PT), between IPv6 and IPv4 realms, on receipt of a return IPv4 packet, the NAPT-PT SHALL, on recognition of the TCP port, translated the packet back to V6.

Specification Text:

IPv6 Node A would establish a TCP session with the IPv4 Node C as follows:

Node A creates a packet with:

Source Address, SA=FEDC:BA98::7654:3210 , source TCP port = 3017 and Destination Address, DA = PREFIX::132.146.243.30, destination TCP port = 23.

When the packet reaches the NAPT-PT box, NAPT-PT would assign one of the TCP ports from the assigned V4 address to translate the tuple of (Source Address, Source TCP port) as follows:

  SA=120.130.26.10, source TCP port = 1025 and
  DA=132.146.243.30, destination TCP port = 23.

The returning traffic from 132.146.243.30, TCP port 23 will be recognised as belonging to the same session and will be translated back to V6 as follows:

  SA = PREFIX::132.146.243.30, source TCP port = 23;
  DA = FEDC:BA98::7654:3210 , destination TCP port = 3017

Inbound NAPT-PT sessions are restricted to one server per service, assigned via static TCP/UDP port mapping. For example, the Node [IPv6-A] in figure 1 may be the only HTTP server (port 80) in the V6 domain. Node [IPv4-C] sends a packet:

  SA=132.146.243.30, source TCP port = 1025 and
  DA=120.130.26.10, destination TCP port = 80

NAPT-PT will translate this packet to:

  SA=PREFIX::132.146.243.30, source TCP port = 1025
  DA=FEDC:BA98::7654:3210, destination TCP port = 80

Identifier: RQ_003_6015
RFC Clause: 3.2
Type: Mandatory
Applies to: Router

Requirement:

When implementing Network Address Port Translation + Protocol Translation (NAPT-PT), between IPv6 and IPv4 realms, inbound NAPT-PT sessions SHALL BE restricted to one server per service, assigned via static TCP/UDP port mapping.

Specification Text:

IPv6 Node A would establish a TCP session with the IPv4 Node C as follows:

Node A creates a packet with:

Source Address, SA=FEDC:BA98::7654:3210 , source TCP port = 3017 and Destination Address, DA = PREFIX::132.146.243.30, destination TCP port = 23.

When the packet reaches the NAPT-PT box, NAPT-PT would assign one of the TCP ports from the assigned V4 address to translate the tuple of (Source Address, Source TCP port) as follows:

  SA=120.130.26.10, source TCP port = 1025 and
  DA=132.146.243.30, destination TCP port = 23.

The returning traffic from 132.146.243.30, TCP port 23 will be recognised as belonging to the same session and will be translated back to V6 as follows:

  SA = PREFIX::132.146.243.30, source TCP port = 23;
  DA = FEDC:BA98::7654:3210 , destination TCP port = 3017

Inbound NAPT-PT sessions are restricted to one server per service, assigned via static TCP/UDP port mapping. For example, the Node [IPv6-A] in figure 1 may be the only HTTP server (port 80) in the V6 domain. Node [IPv4-C] sends a packet:

  SA=132.146.243.30, source TCP port = 1025 and
  DA=120.130.26.10, destination TCP port = 80

NAPT-PT will translate this packet to:

  SA=PREFIX::132.146.243.30, source TCP port = 1025
  DA=FEDC:BA98::7654:3210, destination TCP port = 80

Identifier: RQ_003_6016
RFC Clause: 4
Type: Recommendation
Applies to: Router

Requirement:
When using DNS Application Level Gateway (DNS-ALG) address assignement using "A6" records, the DNS-ALG SHOULD, track all the replies in the transaction before translating an "A6" record to an "A" record.

Specification Text:
In any case, the DNS-ALG's principle of operation described in this section is the same with either "AAAA" or "A6" records. **The only difference is that a name resolution using "A6" records may require more than one query - reply pairs. The DNS-ALG SHOULD, in that case, track all the replies in the transaction before translating an "A6" record to an "A" record.**

Identifier: RQ_003_6017
RFC Clause: 4.1
Type: Mandatory
Applies to: Router

Requirement:
When using DNS Application Level Gateway (DNS-ALG) address assignement for incoming connections (IPv4 to IPv6), The DNS-ALG on the NAT-PT device SHALL modify DNS Queries for for A records going into the V6 domain by changing the Query type in "Node Name to Node Address Query requests" from "A" to "AAAA" or "A6".

Specification Text:
In figure 2 above, when Node C's name resolver sends a name look up request for Node A, the lookup query is directed to the DNS server on the V6 network. Considering that NAT-PT is residing on the border router between V4 and V6 networks, this request datagram would traverse through the NAT-PT router. **The DNS-ALG on the NAT-PT device would modify DNS Queries for A records going into the V6 domain as follows:** (Note that a TCP/UDP DNS packet is recognised by the fact that its source or destination port number is 53)

a) **For Node Name to Node Address Query requests: Change the Query type from "A" to "AAAA" or "A6".**

b) For Node address to Node name query requests: Replace the string "IN-ADDR.ARPA" with the string "IP6.INT". Replace the V4 address octets (in reverse order) preceding the string "IN-ADDR.ARPA" with the corresponding V6 address (if there exists a map) octets in reverse order.

Identifier: RQ_003_6018
RFC Clause: 4.1
Type: Mandatory
Applies to: Router

Requirement:
When using DNS Application Level Gateway (DNS-ALG) address assignement for incoming connections (IPv4 to IPv6), The DNS-ALG on the NAT-PT device SHALL modify DNS Queries for for A records going into the V6 domain by replacing the string "IN-ADDR.ARPA" with the string "IP6.ARPA" in "Node address to Node name query requests".

NOTE: RFC2766 states "Replace the string "IN-ADDR.ARPA" with the string "IP6.INT" however this reference to IP6.INT is depricated and replaced by IP6.ARPA in RFC3152 which is in turn superceeded by RFC3596.

Specification Text:
In figure 2 above, when Node C's name resolver sends a name look up request for Node A, the lookup query is directed to the DNS server on the V6 network. Considering that NAT-PT is residing on the border router between V4 and V6 networks, this request datagram would traverse through the NAT-PT router. **The DNS-ALG on the NAT-PT device would modify DNS Queries for A records going into the V6 domain as follows:** (Note that a TCP/UDP DNS packet is recognised by the fact that its source or destination port number is 53)

a) For Node Name to Node Address Query requests: Change the Query type from "A" to "AAAA" or "A6".

b) **For Node address to Node name query requests: Replace the string "IN-ADDR.ARPA" with the string "IP6.INT".** Replace the V4 address octets (in reverse order) preceding the string "IN-ADDR.ARPA" with the corresponding V6 address (if there exists a map) octets in reverse order.

Identifier:	RQ_003_6019
RFC Clause:	4.1
Type:	Mandatory
Applies to:	Router

Requirement:

When using DNS Application Level Gateway (DNS-ALG) address assignement for incoming connections (IPv4 to IPv6), The DNS-ALG on the NAT-PT device SHALL modify DNS Queries for for A records going into the V6 domain by replacing the V4 address octets (in reverse order) preceding the string "IN-ADDR.ARPA" with the corresponding V6 address (if there exists a map) octets in reverse order.

Specification Text:

In figure 2 above, when Node C's name resolver sends a name look up request for Node A, the lookup query is directed to the DNS server on the V6 network. Considering that NAT-PT is residing on the border router between V4 and V6 networks, this request datagram would traverse through the NAT-PT router. **The DNS-ALG on the NAT-PT device would modify DNS Queries for A records going into the V6 domain as follows:** (Note that a TCP/UDP DNS packet is recognised by the fact that its source or destination port number is 53)

a) For Node Name to Node Address Query requests: Change the Query type from "A" to "AAAA" or "A6".

b) For Node address to Node name query requests: Replace the string "IN-ADDR.ARPA" with the string "IP6.INT". Replace the V4 address octets (in reverse order) preceding the string "IN-ADDR.ARPA" with the corresponding V6 address (if there exists a map) octets in reverse order.

Identifier:	RQ_003_6020
RFC Clause:	4.1
Type:	Mandatory
Applies to:	Router

Requirement:

When using DNS Application Level Gateway (DNS-ALG) address assignement for incoming connections (IPv4 to IPv6), The DNS-ALG on the NAT-PT device SHALL, when a DNS response traverses from the DNS server on the V6 network to the V4 node, intercept the DNS packet and translate DNS responses for "AAAA" records into "A" records

Specification Text:

In the opposite direction, when a DNS response traverses from the DNS server on the V6 network to the V4 node, the DNS-ALG once again intercepts the DNS packet and would:

a) **Translate DNS responses for "AAAA"** or "A6" **records into "A" records,** (only translate "A6" records when the name has completely been resolved)
b) Replace the V6 address resolved by the V6 DNS with the V4 address internally assigned by the NAT-PT router.

Identifier:	RQ_003_6021
RFC Clause:	4.1
Type:	Mandatory
Applies to:	Router

Requirement:

When using DNS Application Level Gateway (DNS-ALG) address assignement for incoming connections (IPv4 to IPv6), The DNS-ALG on the NAT-PT device SHALL, when a DNS response traverses from the DNS server on the V6 network to the V4 node, intercept the DNS packet and, when the name has been completely resolved, translate DNS responses for "A6" records into "A" records.

Specification Text:

In the opposite direction, when a DNS response traverses from the DNS server on the V6 network to the V4 node, the DNS-ALG once again intercepts the DNS packet and would:

a) **Translate DNS responses for** "AAAA" or **"A6" records into "A" records, (only translate "A6" records when the name has completely been resolved)**

b) Replace the V6 address resolved by the V6 DNS with the V4 address internally assigned by the NAT-PT router.

Identifier: RQ_003_6022
RFC Clause: 4.1
Type: Mandatory
Applies to: Router

Requirement:

When using DNS Application Level Gateway (DNS-ALG) address assignement for incoming connections (IPv4 to IPv6), The DNS-ALG on the NAT-PT device SHALL, when a DNS response traverses from the DNS server on the V6 network to the V4 node, intercept the DNS packet and replace the V6 address resolved by the V6 DNS with the V4 address internally assigned by the NAT-PT router.

Specification Text:

In the opposite direction, when a DNS response traverses from the DNS server on the V6 network to the V4 node, the DNS-ALG once again intercepts the DNS packet and would:

a) Translate DNS responses for "AAAA" or "A6" records into "A" records, (only translate "A6" records when the name has completely been resolved)

b) **Replace the V6 address resolved by the V6 DNS with the V4 address internally assigned by the NAT-PT router.**

Identifier: RQ_003_6023
RFC Clause: 4.1
Type: Mandatory
Applies to: Router

Requirement:

When using DNS Application Level Gateway (DNS-ALG) address assignement for incoming connections (IPv4 to IPv6), The NAT-PT device SHALL, if an V4 address is not previously assigned to the V6 node, assign one at this time.

Specification Text:

If a V4 address is not previously assigned to this V6 node, NAT-PT would assign one at this time. As an example say IPv4-C attempts to initialise a session with node IPv6-A by making a name lookup ("A" record) for Node-A . The name query goes to the local DNS and from there it is propagated to the DNS server of the IPv6 network. The DNS-ALG intercepts and translates the "A" query to "AAAA" or "A6" query and then forwards it to the DNS server in the IPv6 network which replies as follows: (The example uses AAAA records for convenience)

```
Node-A    AAAA     FEDC:BA98::7654:3210,
```

this is returned by the DNS server and gets intercepted and translated by the DNS-ALG to:

```
Node-A     A      120.130.26.1
```

The DNS-ALG also holds the mapping between FEDC:BA98::7654:3210 and 120.130.26.1 in NAT-PT. The "A" record is then returned to Node-C. Node-C can now initiate a session as follows:

```
SA=132.146.243.30, source TCP port = 1025  and
DA=120.130.26.1, destination TCP port = 80
```

```
   the packet will be routed to NAT-PT, which since it already holds a
   mapping between  FEDC:BA98::7654:3210 and 120.130.26.1 can translate
   the packet to:

      SA=PREFIX::132.146.243.30, source TCP port = 1025
      DA=FEDC:BA98::7654:3210, destination TCP port = 80

   the communication can now proceed as normal.
```

Identifier: RQ_003_6024
RFC Clause: 4.1
Type: Recommendation
Applies to: Router

Requirement:
When using DNS Application Level Gateway (DNS-ALG) address assignement for incoming connections (IPv4 to IPv6), the TTL values on all DNS resource records (RRs) passing through NAT-PT SHOULD be set to 0.

Specification Text:
The TTL values on all DNS resource records (RRs) passing through NAT-PT SHOULD be set to 0 so that DNS servers/clients do not cache temporarily assigned RRs. Note, however, that due to some buggy DNS client implementations a value of 1 might in some cases work better. The TTL values should be left unchanged for statically mapped addresses.

Identifier: RQ_003_6025
RFC Clause: 4.1
Type: Recommendation
Applies to: Router

Requirement:
When using DNS Application Level Gateway (DNS-ALG) address assignement for incoming connections (IPv4 to IPv6), the TTL values on all DNS resource records (RRs) passing through NAT-PT SHOULD be left unchanged for statically mapped addresses.

Specification Text:
The TTL values on all DNS resource records (RRs) passing through NAT-PT SHOULD be set to 0 so that DNS servers/clients do not cache temporarily assigned RRs. Note, however, that due to some buggy DNS client implementations a value of 1 might in some cases work better. **The TTL values should be left unchanged for statically mapped addresses.**

Identifier: RQ_003_6026
RFC Clause: 4.1
Type: Recommendation
Applies to: Router

Requirement:
When using DNS Application Level Gateway (DNS-ALG) address assignement for incoming connections (IPv4 to IPv6), address mappings for incoming sessions should time out to minimise the effect of denial of service attacks.

Specification Text:
Address mappings for incoming sessions, as described above, are subject to denial of service attacks since one can make multiple queries for nodes residing in the V6 network causing the DNS-ALG to map all V4 addresses in NAT-PT and thus block legitimate incoming sessions. **Thus, address mappings for incoming sessions should time out to minimise the effect of denial of service attacks.** Additionally, one IPv4 address (using NAPT-PT, see 3.2) could be reserved for outgoing sessions only to minimise the effect of such attacks to outgoing sessions.

Identifier: RQ_003_6027
RFC Clause: 4.1
Type: Optional
Applies to: Router

Requirement:
When using DNS Application Level Gateway (DNS-ALG) address assignement for incoming connections (IPv4 to IPv6), one IPv4 address (using NAPT-PT) could be reserved for outgoing sessions only.

Specification Text:
Address mappings for incoming sessions, as described above, are subject to denial of service attacks since one can make multiple queries for nodes residing in the V6 network causing the DNS-ALG to map all V4 addresses in NAT-PT and thus block legitimate incoming sessions. Thus, address mappings for incoming sessions should time out to minimise the effect of denial of service attacks. **Additionally, one IPv4 address (using NAPT-PT, see 3.2) could be reserved for outgoing sessions only to minimise the effect of such attacks to outgoing sessions.**

Identifier: RQ_003_6028
RFC Clause: 4.2
Type: Recommendation
Applies to: Router

Requirement:
It is recommend that DNS servers internal to V6 domains maintain a mapping of names to IPv6 addresses for internal nodes.

Specification Text:
V6 nodes learn the address of V4 nodes from the DNS server in the V4 domain or from the DNS server internal to the V6 network. **We recommend that DNS servers internal to V6 domains maintain a mapping of names to IPv6 addresses for internal nodes** and possibly cache mappings for some external nodes. In the case where the DNS server in the v6 domain contains the mapping for external V4 nodes, the DNS queries will not cross the V6 domain and that would obviate the need for DNS-ALG intervention. Otherwise, the queries will cross the V6 domain and are subject to DNS-ALG intervention. We recommend external DNS servers in the V4 domain cache name mapping for external nodes (i.e., V4 nodes) only. Zone transfers across IPv4 - IPv6 boundaries are strongly discouraged.

Identifier: RQ_003_6029
RFC Clause: 4.2
Type: Recommendation
Applies to: Router

Requirement:
It is recommend that DNS servers internal to V6 domains cache mappings for some external nodes.

Specification Text:
V6 nodes learn the address of V4 nodes from the DNS server in the V4 domain or from the DNS server internal to the V6 network. **We recommend that DNS servers internal to V6 domains** maintain a mapping of names to IPv6 addresses for internal nodes and **possibly cache mappings for some external nodes.** In the case where the DNS server in the v6 domain contains the mapping for external V4 nodes, the DNS queries will not cross the V6 domain and that would obviate the need for DNS-ALG intervention. Otherwise, the queries will cross the V6 domain and are subject to DNS-ALG intervention. We recommend external DNS servers in the V4 domain cache name mapping for external nodes (i.e., V4 nodes) only. Zone transfers across IPv4 - IPv6 boundaries are strongly discouraged.

Identifier: RQ_003_6030
RFC Clause: 4.2
Type: Recommendation
Applies to: Router

Requirement:
It is recommend that external DNS servers in the V4 domain cache name mapping for external nodes (i.e., V4 nodes) only.

Specification Text:

V6 nodes learn the address of V4 nodes from the DNS server in the V4 domain or from the DNS server internal to the V6 network. We recommend that DNS servers internal to V6 domains maintain a mapping of names to IPv6 addresses for internal nodes and possibly cache mappings for some external nodes. In the case where the DNS server in the v6 domain contains the mapping for external V4 nodes, the DNS queries will not cross the V6 domain and that would obviate the need for DNS-ALG intervention. Otherwise, the queries will cross the V6 domain and are subject to DNS-ALG intervention. **We recommend external DNS servers in the V4 domain cache name mapping for external nodes (i.e., V4 nodes) only.** Zone transfers across IPv4 - IPv6 boundaries are strongly discouraged.

Identifier:	RQ_003_6031
RFC Clause:	4.2
Type:	Mandatory
Applies to:	Router

Requirement:

In the case of NAPT-PT, a TCP/UDP source port SHALL BE assigned from the registered V4 address upon detection of each new outbound session.

Specification Text:

In the case of NAPT-PT, a TCP/UDP source port is assigned from the registered V4 address upon detection of each new outbound session.

Identifier:	RQ_003_6032
RFC Clause:	5
Type:	Recommendation
Applies to:	Router

Requirement:

NAT-PT SHOULD translate all IP/ICMP headers from v4 to v6 in order to make end-to-end IPv6 to IPv4 communication possible.

Specification Text:

The IPv4 and ICMPv4 headers are similar to their V6 counterparts but a number of field are either missing, have different meaning or different length. **NAT-PT SHOULD translate all IP/ICMP headers from v4 to v6 and vice versa in order to make end-to-end IPv6 to IPv4 communication possible**. Due to the address translation function and possible port multiplexing, NAT-PT SHOULD also make appropriate adjustments to the upper layer protocol (TCP/UDP) headers. A separate section on FTP-ALG describes the changes FTP-ALG would make to FTP payload as an FTP packet traverses from V4 to V6 realm or vice versa.

Identifier:	RQ_003_6033
RFC Clause:	5
Type:	Recommendation
Applies to:	Router

Requirement:

NAT-PT SHOULD translate all IP/ICMP headers from v6 to v4 in order to make end-to-end IPv4 to IPv6 communication possible.

Specification Text:

The IPv4 and ICMPv4 headers are similar to their V6 counterparts but a number of field are either missing, have different meaning or different length. **NAT-PT SHOULD translate all IP/ICMP headers from v4 to v6 and vice versa in order to make end-to-end IPv6 to IPv4 communication possible**. Due to the address translation function and possible port multiplexing, NAT-PT SHOULD also make appropriate adjustments to the upper layer protocol (TCP/UDP) headers. A separate section on FTP-ALG describes the changes FTP-ALG would make to FTP payload as an FTP packet traverses from V4 to V6 realm or vice versa.

Identifier:	RQ_003_6034
RFC Clause:	5
Type:	Recommendation
Applies to:	Router

Requirement:
NAT-PT SHOULD make appropriate adjustments to the upper layer protocol (TCP/UDP) headers.

Specification Text:
The IPv4 and ICMPv4 headers are similar to their V6 counterparts but a number of field are either missing, have different meaning or different length. NAT-PT SHOULD translate all IP/ICMP headers from v4 to v6 and vice versa in order to make end-to-end IPv6 to IPv4 communication possible. Due to the address translation function and possible port multiplexing, **NAT-PT SHOULD also make appropriate adjustments to the upper layer protocol (TCP/UDP) headers.** A separate section on FTP-ALG describes the changes FTP-ALG would make to FTP payload as an FTP packet traverses from V4 to V6 realm or vice versa.

Identifier:	RQ_003_6035
RFC Clause:	5.1
Type:	Mandatory
Applies to:	Router

Requirement:
With the exception of the "Source Address" and "Destination Address" fields, the translation of the IPv4 headers to IPv6 headers in the NAT-PT SHALL be as specified for the Stateless IP/IPCM Translator (SIIT) [RFC 2765].

Specification Text:
Translating IPv4 headers to IPv6 headers

This is done exactly the same as in SIIT apart from the following fields:

Source Address:
The low-order 32 bits is the IPv4 source address. The high-order 96 bits is the designated PREFIX for all v4 communications. Addresses using this PREFIX will be routed to the NAT-PT gateway (PREFIX::/96)

Destination Address:
NAT-PT retains a mapping between the IPv4 destination address and the IPv6 address of the destination node. The IPv4 destination address is replaced by the IPv6 address retained in that mapping.

Identifier:	RQ_003_6036
RFC Clause:	5.1
Type:	Mandatory
Applies to:	Router

Requirement:
The NAT-PT SHALL translate the Source Address field in the IPv4 header to IPv6 as follows:

- The low-order 32 bits is the IPv4 source address.
- The high-order 96 bits is the designated PREFIX for all v4 communications.

Specification Text:
Translating IPv4 headers to IPv6 headers

This is done exactly the same as in SIIT apart from the following fields:

Source Address:
The low-order 32 bits is the IPv4 source address. The high-order 96 bits is the designated PREFIX for all v4 communications. Addresses using this PREFIX will be routed to the NAT-PT gateway (PREFIX::/96)

Destination Address:

NAT-PT retains a mapping between the IPv4 destination address and the IPv6 address of the destination node. The IPv4 destination address is replaced by the IPv6 address retained in that mapping.

Identifier:	RQ_003_6037
RFC Clause:	5.1
Type:	Mandatory
Applies to:	Router

Requirement:

The NAT-PT SHALL replace the Destination Address field in the IPv4 header by the IPv6 address retained in the NAT-PT mapping between the IPv4 destination address and the IPv6 address of the destination node.

Specification Text:

Translating IPv4 headers to IPv6 headers

This is done exactly the same as in SIIT apart from the following fields:

Source Address:

The low-order 32 bits is the IPv4 source address. The high-order 96 bits is the designated PREFIX for all v4 communications. Addresses using this PREFIX will be routed to the NAT-PT gateway (PREFIX::/96)

Destination Address:

NAT-PT retains a mapping between the IPv4 destination address and the IPv6 address of the destination node. The IPv4 destination address is replaced by the IPv6 address retained in that mapping.

Identifier:	RQ_003_6038
RFC Clause:	5.2
Type:	Mandatory
Applies to:	Router

Requirement:

With the exception of the "Source Address" and "Destination Address" fields, the translation of the IPv6 headers to IPv4 headers in the NAT-PT SHALL be as specified for the Stateless IP/IPCM Translator (SIIT) [RFC 2765].

Specification Text:

Translating IPv6 headers to IPv4 headers

This is done exactly the same as in SIIT apart from the Source Address which should be determined as follows:

Source Address:

The NAT-PT retains a mapping between the IPv6 source address and an IPv4 address from the pool of IPv4 addresses available. The IPv6 source address is replaced by the IPv4 address retained in that mapping.

Destination Address:

IPv6 packets that are translated have a destination address of the form PREFIX::IPv4/96. Thus the low-order 32 bits of the IPv6 destination address is copied to the IPv4 destination address.

Identifier: RQ_003_6039
RFC Clause: 5.2
Type: Mandatory
Applies to: Router

Requirement:
The NAT-PT SHALL replace the Source Address field in the IPv6 header by the IPv4 address retained in the NAT-PT mapping between the IPv6 Source address and the IPv4 Source address.

Specification Text:
Translating IPv6 headers to IPv4 headers

This is done exactly the same as in SIIT apart from the Source Address which should be determined as follows:

```
      **Source Address:**
         **The NAT-PT retains a mapping between the IPv6 source address**
         **and an IPv4 address from the pool of IPv4 addresses**
         **available. The IPv6 source address is replaced by the IPv4**
         **address retained in that mapping.**

      Destination Address:
         IPv6 packets that are translated have a destination address
         of the form PREFIX::IPv4/96. Thus the low-order 32 bits of
         the IPv6 destination address is copied to the IPv4
         destination address.
```

Identifier: RQ_003_6040
RFC Clause: 5.2
Type: Mandatory
Applies to: Router

Requirement:
The NAT-PT SHALL translate the Destination Address field in the IPv6 header to IPv4 by copying the low-order 32 bits of the IPv6 destination address to the IPv4 destination address.

Specification Text:
Translating IPv6 headers to IPv4 headers

This is done exactly the same as in SIIT apart from the Source Address which should be determined as follows:

```
      Source Address:
         The NAT-PT retains a mapping between the IPv6 source address
         and an IPv4 address from the pool of IPv4 addresses
         available. The IPv6 source address is replaced by the IPv4
         address retained in that mapping.

      **Destination Address:**
         **IPv6 packets that are translated have a destination address**
         **of the form PREFIX::IPv4/96. Thus the low-order 32 bits of**
         **the IPv6 destination address is copied to the IPv4**
         **destination address.**
```

Identifier: RQ_003_6041
RFC Clause: 5.3.1
Type: Recommendation
Applies to: Router

Requirement:
UDP checksums, when set to a non-zero value, SHOULD be recalculated to reflect the address change from v4 to v6.

Specification Text:
UDP checksums, when set to a non-zero value, and TCP checksum **SHOULD be recalculated to reflect the address change from v4 to v6.** The incremental checksum adjustment algorithm may be borrowed from [NAT]. In the case of NAPT-PT, TCP/UDP checksum should be adjusted to account for the address and TCP/UDP port changes, going from V4 to V6 address.

Identifier:	RQ_003_6042
RFC Clause:	5.3.1
Type:	Recommendation
Applies to:	Router

Requirement:
The TCP checksum SHOULD be recalculated to reflect the address change from v4 to v6.

Specification Text:
UDP checksums, when set to a non-zero value, and **TCP checksum SHOULD be recalculated to reflect the address change from v4 to v6**. The incremental checksum adjustment algorithm may be borrowed from [NAT]. In the case of NAPT-PT, TCP/UDP checksum should be adjusted to account for the address and TCP/UDP port changes, going from V4 to V6 address.

Identifier:	RQ_003_6043
RFC Clause:	5.3.1
Type:	Recommendation
Applies to:	Router

Requirement:
In the case of NAPT-PT, TCP/UDP checksum should be adjusted to account for the address and TCP/UDP port changes, going from V4 to V6 address.

Specification Text:
UDP checksums, when set to a non-zero value, and TCP checksum SHOULD be recalculated to reflect the address change from v4 to v6. The incremental checksum adjustment algorithm may be borrowed from [NAT]. **In the case of NAPT-PT, TCP/UDP checksum should be adjusted to account for the address and TCP/UDP port changes, going from V4 to V6 address.**

Identifier:	RQ_003_6044
RFC Clause:	5.3.1
Type:	Mandatory
Applies to:	Router

Requirement:
When the checksum of a V4 UDP packet is set to zero, NAT-PT MUST evaluate the checksum in its entirety for the V6-translated UDP packet.

Specification Text:
When the checksum of a V4 UDP packet is set to zero, NAT-PT MUST evaluate the checksum in its entirety for the V6-translated UDP packet. If a V4 UDP packet with a checksum of zero arrives in fragments, NAT-PT MUST await all the fragments until they can be assembled into a single non-fragmented packet and evaluate the checksum prior to forwarding the translated V6 UDP packet.

Identifier:	RQ_003_6045
RFC Clause:	5.3.1
Type:	Mandatory
Applies to:	Router

Requirement:
If a V4 UDP packet with a checksum of zero arrives in fragments, NAT-PT MUST await all the fragments until they can be assembled into a single non-fragmented packet and evaluate the checksum prior to forwarding the translated V6 UDP packet.

Specification Text:
When the checksum of a V4 UDP packet is set to zero, NAT-PT MUST evaluate the checksum in its entirety for the V6-translated UDP packet. **If a V4 UDP packet with a checksum of zero arrives in fragments, NAT-PT MUST await all the fragments until they can be assembled into a single non-fragmented packet and evaluate the checksum prior to forwarding the translated V6 UDP packet.**

Identifier:	RQ_003_6046
RFC Clause:	5.3.2
Type:	Recommendation
Applies to:	Router

Requirement:

UDP checksums SHOULD be recalculated to reflect the address change from v6 to v4.

Specification Text:

TCP and **UDP checksums SHOULD be recalculated to reflect the address change from v6 to v4.** The incremental checksum adjustment algorithm may be borrowed from [NAT]. In the case of NAPT-PT, TCP/UDP checksums should be adjusted to account for the address and TCP/UDP port changes, going from V6 to V4 addresses. For UDP packets, optionally, the checksum may simply be changed to zero.

Identifier:	RQ_003_6047
RFC Clause:	5.3.2
Type:	Recommendation
Applies to:	Router

Requirement:

TCP checksums SHOULD be recalculated to reflect the address change from v6 to v4.

Specification Text:

TCP and UDP **checksums SHOULD be recalculated to reflect the address change from v6 to v4.** The incremental checksum adjustment algorithm may be borrowed from [NAT]. In the case of NAPT-PT, TCP/UDP checksums should be adjusted to account for the address and TCP/UDP port changes, going from V6 to V4 addresses. For UDP packets, optionally, the checksum may simply be changed to zero.

Identifier:	RQ_003_6048
RFC Clause:	5.3.2
Type:	Recommendation
Applies to:	Router

Requirement:

In the case of NAPT-PT, TCP checksums should be adjusted to account for the address change from v6 to v4.

Specification Text:

TCP and UDP checksums SHOULD be recalculated to reflect the address change from v6 to v4. The incremental checksum adjustment algorithm may be borrowed from [NAT]. **In the case of NAPT-PT, TCP/UDP checksums should be adjusted to account for the address and TCP/UDP port changes, going from V6 to V4 addresses.** For UDP packets, optionally, the checksum may simply be changed to zero.

Identifier:	RQ_003_6049
RFC Clause:	5.3.2
Type:	Optional
Applies to:	Router

Requirement:

For UDP packets, optionally, the checksum may simply be changed to zero.

Specification Text:

TCP and UDP checksums SHOULD be recalculated to reflect the address change from v6 to v4. The incremental checksum adjustment algorithm may be borrowed from [NAT]. In the case of NAPT-PT, TCP/UDP checksums should be adjusted to account for the address and TCP/UDP port changes, going from V6 to V4 addresses. **For UDP packets, optionally, the checksum may simply be changed to zero.**

Identifier: RQ_003_6050
RFC Clause: 5.3.2
Type: Recommendation
Applies to: Router

Requirement:
In the case of NAPT-PT, UDP checksums should be adjusted to account for the the address change from v6 to v4.

Specification Text:
TCP and UDP checksums SHOULD be recalculated to reflect the address change from v6 to v4. The incremental checksum adjustment algorithm may be borrowed from [NAT]. **In the case of NAPT-PT**, TCP/**UDP checksums should be adjusted to account for the address and TCP/UDP port changes, going from V6 to V4 addresses.** For UDP packets, optionally, the checksum may simply be changed to zero.

Identifier: RQ_003_6051
RFC Clause: 5.3.2
Type: Mandatory
Applies to: Router

Requirement:

The checksum calculation for a V4 ICMP header needs to be derived from the V6 ICMP header by running the checksum adjustment algorithm [RFC 1631] to remove the V6 pseudo header from the computation and to take into account changes to the checksum as a result of updates to the source and destination addresses (and transport ports in the case of NAPT-PT) made to the payload carried within ICMP.

Specification Text:
The checksum calculation for a V4 ICMP header needs to be derived from the V6 ICMP header by running the checksum adjustment algorithm [NAT] to remove the V6 pseudo header from the computation. Note, the adjustment MUST additionally take into account changes to the checksum as a result of updates to the source and destination addresses (and transport ports in the case of NAPT-PT) made to the payload carried within ICMP.

Identifier: RQ_003_6052
RFC Clause: 6
Type: Mandatory
Applies to: Router

Requirement:
Because an FTP control session carries, in its payload, the IP address and TCP port information for the data session, an FTP-ALG is REQUIRED to provide application level transparency for FTP.

Specification Text:
Because an FTP control session carries, in its payload, the IP address and TCP port information for the data session, an FTP-ALG is REQUIRED to provide application level transparency for this popular Internet application.

Identifier: RQ_003_6084
RFC Clause: 6
Type: Optional
Applies to: Node

Requirement:
V4 or V6 nodes MAY implement EPRT and EPSV command extensions to FTP.

Specification Text:
In the FTP application running on a legacy V4 node, arguments to the FTP PORT command and arguments in PASV response(successful) include an IP V4 address and a TCP port, both represented in ASCII as h1,h2,h3,h4,p1,p2. **However, [FTP-IPV6] suggests EPRT and EPSV command extensions to FTP, with an intent to eventually retire the use of PORT and PASV commands.** These extensions may be used on a V4 or V6 node. FTP-ALG, facilitating transparent FTP between V4 and V6 nodes, works as follows.

Identifier:	RQ_003_6053
RFC Clause:	6.1
Type:	Mandatory
Applies to:	Router

Requirement:
If a V4 host originates the FTP session and uses PORT command, the FTP-ALG will translate this command into EPRT command prior to forwarding to the V6 node.

Specification Text:
A V4 host may or may not have the EPRT and EPSV command extensions implemented in its FTP application. **If a V4 host originates the FTP session and uses PORT** or **PASV command, the FTP-ALG will translate these commands into EPRT** and EPSV **commands respectively prior to forwarding to the V6 node**. Likewise, EPSV response from V6 nodes will be translated into PASV response prior to forwarding to V4 nodes. The format of EPRT and EPSV commands and EPSV response may be specified as follows[FTP-IPV6].

Identifier:	RQ_003_6054
RFC Clause:	6.1
Type:	Mandatory
Applies to:	Router

Requirement:
If a V4 host originates the FTP session and uses PASV command, the FTP-ALG will translate this command into EPSV command prior to forwarding to the V6 node.

Specification Text:
A V4 host may or may not have the EPRT and EPSV command extensions implemented in its FTP application. **If a V4 host originates the FTP session and uses** PORT or **PASV command, the FTP-ALG will translate these commands into** EPRT and **EPSV commands respectively prior to forwarding to the V6 node.** Likewise, EPSV response from V6 nodes will be translated into PASV response prior to forwarding to V4 nodes. The format of EPRT and EPSV commands and EPSV response may be specified as follows[FTP-IPV6].

Identifier:	RQ_003_6055
RFC Clause:	6.1
Type:	Mandatory
Applies to:	Router

Requirement:
EPSV response from V6 nodes will be translated into PASV response prior to forwarding to V4 nodes.

Specification Text:
A V4 host may or may not have the EPRT and EPSV command extensions implemented in its FTP application. If a V4 host originates the FTP session and uses PORT or PASV command, the FTP-ALG will translate these commands into EPRT and EPSV commands respectively prior to forwarding to the V6 node. Likewise, **EPSV response from V6 nodes will be translated into PASV response prior to forwarding to V4 nodes**. The format of EPRT and EPSV commands and EPSV response may be specified as follows[FTP-IPV6].

Identifier:	RQ_003_6081
RFC Clause:	6.1
Type:	Mandatory
Applies to:	Router

Requirement:
If the FTP-ALG translates a PORT command command into EPRT command, the EPRT command [RFC 2766] SHALL be structured as follows:

```
EPRT<space><d><net-prt><d><net-addr><d><tcp-port><d>
```

Specification Text:

A V4 host may or may not have the EPRT and EPSV command extensions implemented in its FTP application. **If a V4 host originates the FTP session and uses PORT or PASV command, the FTP-ALG will translate these commands into EPRT and EPSV commands respectively prior to forwarding to the V6 node.** Likewise, EPSV response from V6 nodes will be translated into PASV response prior to forwarding to V4 nodes. The format of EPRT and EPSV commands and EPSV response may be specified as follows[FTP-IPV6].

EPRT<space><d><net-prt><d><net-addr><d><tcp-port><d>
EPSV<space><net-prt>
(or)
EPSV<space>ALL

Format of EPSV response(Positive): 229 <text indicating extended passive mode> (<d><d><d><tcp-port><d>)

Identifier: RQ_003_6082
RFC Clause: 6.1
Type: Mandatory
Applies to: Router

Requirement:

If the FTP-ALG translates a PASV command command into EPSV command, the EPRT command [RFC 2766] SHALL be structured as follows:

```
EPSV<space><net-prt>
(or)
EPSV<space>ALL
```

Specification Text:

A V4 host may or may not have the EPRT and EPSV command extensions implemented in its FTP application. **If a V4 host originates the FTP session and uses PORT or PASV command, the FTP-ALG will translate these commands into EPRT and EPSV commands respectively prior to forwarding to the V6 node.** Likewise, EPSV response from V6 nodes will be translated into PASV response prior to forwarding to V4 nodes. The format of EPRT and EPSV commands and EPSV response may be specified as follows[FTP-IPV6].

EPRT<space><d><net-prt><d><net-addr><d><tcp-port><d>
EPSV<space><net-prt>
(or)
EPSV<space>ALL

Format of EPSV response(Positive): 229 <text indicating extended passive mode> (<d><d><d><tcp-port><d>)

Identifier: RQ_003_6083
RFC Clause: 6.1
Type: Mandatory
Applies to: Router

Requirement:

If the FTP-ALG translates a PASV response command into EPSV response command, the EPSV response (Positive) [RFC 2766] SHALL be structured as follows:

```
229 <text indicating extended passive mode> (<d><d><d><tcp-port><d>)
```

Specification Text:

A V4 host may or may not have the EPRT and EPSV command extensions implemented in its FTP application. If a V4 host originates the FTP session and uses PORT or PASV command, the FTP-ALG will translate these commands into EPRT and EPSV commands respectively prior to forwarding to the V6 node. **Likewise, EPSV response from V6 nodes will be translated into PASV response prior to forwarding to V4 nodes.** The format of EPRT and EPSV commands and EPSV response may be specified as follows[FTP-IPV6].

```
EPRT<space><d><net-prt><d><net-addr><d><tcp-port><d>
EPSV<space><net-prt>
     (or)
EPSV<space>ALL
```

Format of EPSV response(Positive): 229 <text indicating extended passive mode> (<d><d><d><tcp-port><d>)

Identifier:	RQ_003_6056
RFC Clause:	6.1
Type:	Mandatory
Applies to:	Router

Requirement:

To translate a PORT command from a v4 node to EPRT, the FTP-ALG SHALL set the EPRT protocol <net-prt> field to AF #2 (IPV6).

Specification Text:

PORT command from a V4 node is translated into EPRT command, by setting the protocol <net-prt> field to AF #2 (IPV6) and translating the V4 host Address (represented as h1,h2,h3,h4) into its NAT-PT assigned V6 address in string notation, as defined in [V6ADDR] in the <net-addr> field. TCP port represented by p1,p2 in PORT command MUST be specified as a decimal <tcp-port> in the EPRT command. Further, <tcp-port> translation may also be REQUIRED in the case of NAPT-PT. PASV command from a V4 node is be translated into a EPSV command with the <net-prt> argument set to AF #2. EPSV response from a V6 node is translated into PASV response prior to forwarding to the target V4 host.

Identifier:	RQ_003_6057
RFC Clause:	6.1
Type:	Mandatory
Applies to:	Router

Requirement:

To translate a PORT command from a v4 node to EPRT, the FTP-ALG SHALL translate the V4 host Address into its NAT-PT assigned V6 address in string notation in the <net-addr> field.

Specification Text:

PORT command from a V4 node is translated into EPRT command, by setting the protocol <net-prt> field to AF #2 (IPV6) and **translating the V4 host Address (represented as h1,h2,h3,h4) into its NAT-PT assigned V6 address in string notation, as defined in [V6ADDR] in the <net-addr> field.** TCP port represented by p1,p2 in PORT command MUST be specified as a decimal <tcp-port> in the EPRT command. Further, <tcp-port> translation may also be REQUIRED in the case of NAPT-PT. PASV command from a V4 node is be translated into a EPSV command with the <net-prt> argument set to AF #2. EPSV response from a V6 node is translated into PASV response prior to forwarding to the target V4 host.

Identifier:	RQ_003_6058
RFC Clause:	6.1
Type:	Mandatory
Applies to:	Router

Requirement:

The TCP port represented by p1,p2 in PORT command MUST be specified as a decimal <tcp-port> in the EPRT command by the FTP-ALG.

Specification Text:

PORT command from a V4 node is translated into EPRT command, by setting the protocol <net-prt> field to AF #2 (IPV6) and translating the V4 host Address (represented as h1,h2,h3,h4) into its NAT-PT assigned V6 address in string notation, as defined in [V6ADDR] in the <net-addr> field. **TCP port represented by p1,p2 in PORT command MUST be specified as a decimal <tcp-port> in the EPRT command.** Further, <tcp-port> translation may also be REQUIRED in the case of NAPT-PT. PASV command from a V4 node is be translated into a EPSV command with the <net-prt> argument set to AF #2. EPSV response from a V6 node is translated into PASV response prior to forwarding to the target V4 host.

Identifier:	RQ_003_6059
RFC Clause:	6.1
Type:	Optional
Applies to:	Router

Requirement:

The FTP-ALG SHALL translate the PASV command from a V4 node into a EPSV command with the <net-prt> argument set to AF #2.

Specification Text:

PORT command from a V4 node is translated into EPRT command, by setting the protocol <net-prt> field to AF #2 (IPV6) and translating the V4 host Address (represented as h1,h2,h3,h4) into its NAT-PT assigned V6 address in string notation, as defined in [V6ADDR] in the <net-addr> field. TCP port represented by p1,p2 in PORT command MUST be specified as a decimal <tcp-port> in the EPRT command. Further, <tcp-port> translation may also be REQUIRED in the case of NAPT-PT. **PASV command from a V4 node is be translated into a EPSV command with the <net-prt> argument set to AF #2.** EPSV response from a V6 node is translated into PASV response prior to forwarding to the target V4 host.

Identifier: RQ_003_6060
RFC Clause: 6.1
Type: Mandatory
Applies to: Router

Requirement:

The FTP-ALG SHALL translate the EPSV response from a V6 node into the PASV response prior to forwarding to the target V4 host.

Specification Text:

PORT command from a V4 node is translated into EPRT command, by setting the protocol <net-prt> field to AF #2 (IPV6) and translating the V4 host Address (represented as h1,h2,h3,h4) into its NAT-PT assigned V6 address in string notation, as defined in [V6ADDR] in the <net-addr> field. TCP port represented by p1,p2 in PORT command MUST be specified as a decimal <tcp-port> in the EPRT command. Further, <tcp-port> translation may also be REQUIRED in the case of NAPT-PT. PASV command from a V4 node is be translated into a EPSV command with the <net-prt> argument set to AF #2. **EPSV response from a V6 node is translated into PASV response prior to forwarding to the target V4 host.**

Identifier: RQ_003_6061
RFC Clause: 6.1
Type: Mandatory
Applies to: Router

Requirement:

If a V4 host originated the FTP session and was using EPRT and EPSV commands, the FTP-ALG will translate the protocol Number <net-prt> field from AF #1 to AF #2.

Specification Text:

If a V4 host originated the FTP session and was using EPRT and EPSV commands, the FTP-ALG will simply translate the parameters to these commands, without altering the commands themselves. The protocol Number <net-prt> field will be translated from AF #1 to AF #2. <net-addr> will be translated from the V4 address in ASCII to its NAT-PT assigned V6 address in string notation as defined in [V6ADDR]. <tcp-port> argument in EPSV response requires translation only in the case of NAPT-PT.

Identifier: RQ_003_6062
RFC Clause: 6.1
Type: Mandatory
Applies to: Router

Requirement:

If a V4 host originated the FTP session and was using EPRT and EPSV commands, the FTP-ALG will translate <net-addr> from the V4 address in ASCII to its NAT-PT assigned V6 address in string notation as defined in [RFC2373].

Specification Text:

If a V4 host originated the FTP session and was using EPRT and EPSV commands, the FTP-ALG will simply translate the parameters to these commands, without altering the commands themselves. The protocol Number <net-prt> field will be translated from AF #1 to AF #2. **<net-addr> will be translated from the V4 address in ASCII to its NAT-PT assigned V6 address in string notation as defined in [V6ADDR].** <tcp-port> argument in EPSV response requires translation only in the case of NAPT-PT.

Identifier: RQ_003_6063
RFC Clause: 6.2
Type: Mandatory
Applies to: Router

Requirement:
If the FTP-ALG leaves the command strings "EPRT" and "EPSV" unaltered it SHALL translate the <net-prt>, <net-addr> and <tcp-port> arguments from V6 to its NAT-PT (or NAPT-PT) assigned V4 information.

Specification Text:
If a V6 host originates the FTP session, however, the FTP-ALG has two approaches to pursue. In the first approach, **the FTP-ALG will leave the command strings "EPRT" and "EPSV" unaltered and simply translate the <net-prt>, <net-addr> and <tcp-port> arguments from V6 to its NAT-PT (or NAPT-PT) assigned V4 information.** <tcp-port> is translated only in the case of NAPT-PT. Same goes for EPSV response from V4 node. This is the approach we recommend to ensure forward support for RFC 2428. However, with this approach, the V4 hosts are mandated to have their FTP application upgraded to support EPRT and EPSV extensions to allow access to V4 and V6 hosts, alike.

Identifier: RQ_003_6064
RFC Clause: 6.2
Type: Mandatory
Applies to: Router

Requirement:
If the FTP-ALG leaves the command strings "EPRT" and "EPSV" unaltered it SHALL also leave the ESPV respnse unaltered.

Specification Text:
If a V6 host originates the FTP session, however, the FTP-ALG has two approaches to pursue. **In the first approach, the FTP-ALG will leave the command strings "EPRT" and "EPSV" unaltered** and simply translate the <net-prt>, <net-addr> and <tcp-port> arguments from V6 to its NAT-PT (or NAPT-PT) assigned V4 information. <tcp-port> is translated only in the case of NAPT-PT. **Same goes for EPSV response from V4 node.** This is the approach we recommend to ensure forward support for RFC 2428. However, with this approach, the V4 hosts are mandated to have their FTP application upgraded to support EPRT and EPSV extensions to allow access to V4 and V6 hosts, alike.

Identifier: RQ_003_6065
RFC Clause: 6.2
Type: Mandatory
Applies to: Host

Requirement:
If the FTP-ALG leaves the command strings "EPRT" and "EPSV" unaltered, the V4 hosts are mandated to have their FTP application upgraded to support EPRT and EPSV extensions to allow access to V4 and V6 hosts, alike.

Specification Text:
If a V6 host originates the FTP session, however, the FTP-ALG has two approaches to pursue. In the first approach, **the FTP-ALG will leave the command strings "EPRT" and "EPSV" unaltered** and simply translate the <net-prt>, <net-addr> and <tcp-port> arguments from V6 to its NAT-PT (or NAPT-PT) assigned V4 information. <tcp-port> is translated only in the case of NAPT-PT. Same goes for EPSV response from V4 node. This is the approach we recommend to ensure forward support for RFC 2428. **However, with this approach, the V4 hosts are mandated to have their FTP application upgraded to support EPRT and EPSV extensions to allow access to V4 and V6 hosts, alike.**

Identifier: RQ_003_6066
RFC Clause: 6.2
Type: Recommendation
Applies to: Router

Requirement:
It is RECOMMENDED that the FTP-ALG leaves the command strings "EPRT" and "EPSV" unaltered.

Specification Text:

If a V6 host originates the FTP session, however, the FTP-ALG has two approaches to pursue. In the first approach, the FTP-ALG will leave the command strings "EPRT" and "EPSV" unaltered and simply translate the <net-prt>, <net-addr> and <tcp-port> arguments from V6 to its NAT-PT (or NAPT-PT) assigned V4 information. <tcp-port> is translated only in the case of NAPT-PT. Same goes for EPSV response from V4 node. **This is the approach we recommend to ensure forward support for RFC 2428.** However, with this approach, the V4 hosts are mandated to have their FTP application upgraded to support EPRT and EPSV extensions to allow access to V4 and V6 hosts, alike.

Identifier: RQ_003_6067
RFC Clause: 6.2
Type: Optional
Applies to: Router

Requirement:

If a V6 host originates the FTP session, the FTP-ALG MAY chose to translate the EPRT and EPSV commands and their parameters to "PORT" and "PASV" repectively.

Specification Text:

In the second approach, **the FTP-ALG will translate the command strings "EPRT" and "EPSV" and their parameters from the V6 node into their equivalent NAT-PT assigned V4 node info and attach to "PORT" and "PASV" commands prior to forwarding to V4 node.** Likewise, PASV response from V4 nodes is translated into EPSV response prior to forwarding to the target V6 nodes. However, the FTP-ALG would be unable to translate the command "EPSV<space>ALL" issued by V6 nodes. In such a case, the V4 host, which receives the command, may return an error code indicating unsupported function. This error response may cause many RFC 2428 compliant FTP applications to simply fail, because EPSV support is mandated by RFC 2428. The benefit of this approach, however, is that is does not impose any FTP upgrade requirements on V4 hosts.

Identifier: RQ_003_6068
RFC Clause: 6.2
Type: Mandatory
Applies to: Router

Requirement:

If a V6 host originates the FTP session and the FTP-ALG choses to translate the EPRT and EPSV commands and their parameters to "PORT" and "PASV" repectively, the PASV response from V4 nodes SHALL BE translated into EPSV response prior to forwarding to the target V6 nodes.

Specification Text:

In the second approach, the FTP-ALG will translate the command strings "EPRT" and "EPSV" and their parameters from the V6 node into their equivalent NAT-PT assigned V4 node info and attach to "PORT" and "PASV" commands prior to forwarding to V4 node. **Likewise, PASV response from V4 nodes is translated into EPSV response prior to forwarding to the target V6 nodes.** However, the FTP-ALG would be unable to translate the command "EPSV<space>ALL" issued by V6 nodes. In such a case, the V4 host, which receives the command, may return an error code indicating unsupported function. This error response may cause many RFC 2428 compliant FTP applications to simply fail, because EPSV support is mandated by RFC 2428. The benefit of this approach, however, is that is does not impose any FTP upgrade requirements on V4 hosts.

Identifier: RQ_003_6069
RFC Clause: 6.2
Type: Optional
Applies to: Host

Requirement:

If the FTP-ALG does not translate the command "EPSV<space>ALL" issued by V6 nodes, the V4 host, which receives the command, may return an error code indicating unsupported function.

Specification Text:

In the second approach, the FTP-ALG will translate the command strings "EPRT" and "EPSV" and their parameters from the V6 node into their equivalent NAT-PT assigned V4 node info and attach to "PORT" and "PASV" commands prior to forwarding to V4 node. Likewise, PASV response from V4 nodes is translated into EPSV response prior to forwarding to the target V6 nodes. **However, the FTP-ALG would be unable to translate the command "EPSV<space>ALL" issued by V6 nodes. In such a case, the V4 host, which receives the command, may return an error code indicating unsupported function.** This error response may cause many RFC 2428 compliant FTP applications to simply fail, because EPSV support is mandated by RFC 2428. The benefit of this approach, however, is that is does not impose any FTP upgrade requirements on V4 hosts.

Identifier: RQ_003_6070
RFC Clause: 6.3
Type: Mandatory
Applies to: Router

Requirement:

As a result of payload translation by the FTP-ALG, the TCP checksum SHALL require adjustment.

Specification Text:

If the new size is the same as the previous, only the TCP checksum needs adjustment as a result of the payload translation. If the new size is different from the previous, TCP sequence numbers should also be changed to reflect the change in the length of the FTP control session payload. The IP packet length field in the V4 header or the IP payload length field in the V6 header should also be changed to reflect the new payload size. A table is used by the FTP-ALG to correct the TCP sequence and acknowledgement numbers in the TCP header for control packets in both directions.

Identifier: RQ_003_6071
RFC Clause: 6.3
Type: Mandatory
Applies to: Router

Requirement:

If, as a result of payload translation by the FTP-ALG, the new size of the FTP control packet is different from the previous, the TCP sequence numbers should be changed to reflect the change in the length of the FTP control session payload.

Specification Text:

If the new size is the same as the previous, only the TCP checksum needs adjustment as a result of the payload translation. **If the new size is different from the previous, TCP sequence numbers should also be changed to reflect the change in the length of the FTP control session payload.** The IP packet length field in the V4 header or the IP payload length field in the V6 header should also be changed to reflect the new payload size. A table is used by the FTP-ALG to correct the TCP sequence and acknowledgement numbers in the TCP header for control packets in both directions.

Identifier: RQ_003_6072
RFC Clause: 6.3
Type: Recommendation
Applies to: Router

Requirement:

If, as a result of payload translation by the FTP-ALG, the new size of the V4 FTP control packet is different from the previous, the IP packet length field in the V4 header should be changed to reflect the new payload size.

Specification Text:

If the new size is the same as the previous, only the TCP checksum needs adjustment as a result of the payload translation. If the new size is different from the previous, TCP sequence numbers should also be changed to reflect the change in the length of the FTP control session payload. **The IP packet length field in the V4 header or the IP payload length field in the V6 header should also be changed to reflect the new payload size.** A table is used by the FTP-ALG to correct the TCP sequence and acknowledgement numbers in the TCP header for control packets in both directions.

Identifier: RQ_003_6073
RFC Clause: 6.3
Type: Recommendation
Applies to: Router

Requirement:
If, as a result of payload translation by the FTP-ALG, the new size of the V6 FTP control packet is different from the previous, the or the IP payload length field in the V6 header should be changed to reflect the new payload size.

Specification Text:
If the new size is the same as the previous, only the TCP checksum needs adjustment as a result of the payload translation. If the new size is different from the previous, TCP sequence numbers should also be changed to reflect the change in the length of the FTP control session payload. **The IP packet length field in the V4 header or the IP payload length field in the V6 header should also be changed to reflect the new payload size.** A table is used by the FTP-ALG to correct the TCP sequence and acknowledgement numbers in the TCP header for control packets in both directions.

Identifier: RQ_003_6074
RFC Clause: 6.3
Type: Recommendation
Applies to: Router

Requirement:
Table entries, used by the FTP-ALG to correct the TCP sequence and acknowledgement numbers in the TCP header for control packets in both directions, should have:
- the source address,
- source data port,
- destination address for V4 and V6 portions of the session,
- destination data port for V4 and V6 portions of the session,
- sequence number delta for outbound control packets
- sequence number delta for inbound control packets.

Specification Text:
The table entries should have the source address, source data port, destination address and destination data port for V4 and V6 portions of the session, sequence number delta for outbound control packets and sequence number delta for inbound control packets.

Identifier: RQ_003_6075
RFC Clause: 6.3
Type: Mandatory
Applies to: Router

Requirement:
As a result of payload translation by the FTP-ALG, the sequence number for an outbound control packet SHALL BE increased by the outbound sequence number delta.

Specification Text:
The sequence number for an outbound control packet is increased by the outbound sequence number delta, and the acknowledgement number for the same outbound packet is decreased by the inbound sequence number delta. Likewise, the sequence number for an inbound packet is increased by the inbound sequence number delta and the acknowledgement number for the same inbound packet is decreased by the outbound sequence number delta.

Identifier: RQ_003_6076
RFC Clause: 6.3
Type: Mandatory
Applies to: Router

Requirement:
As a result of payload translation by the FTP-ALG, once the sequence number for an outbound control packet has been increased by the outbound sequence number delta, the acknowledgement number for the same outbound packet SHALL BE decreased by the inbound sequence number delta.

Specification Text:
The sequence number for an outbound control packet is increased by the outbound sequence number delta, **and the acknowledgement number for the same outbound packet is decreased by the inbound sequence number delta.** Likewise, the sequence number for an inbound packet is increased by the inbound sequence number delta and the acknowledgement number for the same inbound packet is decreased by the outbound sequence number delta.

Identifier: RQ_003_6077
RFC Clause: 6.3
Type: Mandatory
Applies to: Router

Requirement:
As a result of payload translation by the FTP-ALG, the sequence number for an inbound packet SHALL BE increased by the inbound sequence number delta.

Specification Text:
The sequence number for an outbound control packet is increased by the outbound sequence number delta, and the acknowledgement number for the same outbound packet is decreased by the inbound sequence number delta. **Likewise, the sequence number for an inbound packet is increased by the inbound sequence number delta** and the acknowledgement number for the same inbound packet is decreased by the outbound sequence number delta.

Identifier: RQ_003_6078
RFC Clause: 6.3
Type: Mandatory
Applies to: Router

Requirement:
As a result of payload translation by the FTP-ALG, once the sequence number for an inbound packet is increased by the inbound sequence number delta, the acknowledgement number for the same inbound packet is decreased by the outbound sequence number delta.

Specification Text:
The sequence number for an outbound control packet is increased by the outbound sequence number delta, and the acknowledgement number for the same outbound packet is decreased by the inbound sequence number delta. Likewise, the sequence number for an inbound packet is increased by the inbound sequence number delta and **the acknowledgement number for the same inbound packet is decreased by the outbound sequence number delta.**

Identifier: RQ_003_6079
RFC Clause: 7.1
Type: Mandatory
Applies to: Router

Requirement:
Except for packets originating from, or directed to, dual-stack nodes that do not require packet translation, it is mandatory that all requests and responses pertaining to a session be routed via the same NAT-PT router.

Specification Text:
There are limitations to using the NAT-PT translation method. **It is mandatory that all requests and responses pertaining to a session be routed via the same NAT-PT router.** One way to guarantee this would be to have NAT-PT based on a border router that is unique to a stub domain, where all IP packets are either originated from the domain or destined to the domain. This is a generic problem with NAT and it is fully described in [NAT-TERM].

Note, this limitation does not apply to packets originating from or directed to dual-stack nodes that do not require packet translation. This is because in a dual-stack set-up, IPv4 addresses implied in a V6 address can be identified from the address format PREFIX::x.y.z.w and a dual-stack router can accordingly route a packet between v4 and dual-stack nodes without tracking state information.

Identifier: RQ_003_6080
RFC Clause: 7.1
Type: Recommendation
Applies to: Node

Requirement:
A native IPv6 connection and/or some kind of tunneled IPv6 connection should be preferred over translation when possible.

Specification Text:
This should also not affect IPv6 to IPv6 communication and in fact only actually use translation when no other means of communication is possible. **For example NAT-PT may also have a native IPv6 connection and/or some kind of tunneled IPv6 connection. Both of the above connections should be preferred over translation when possible.** The above makes sure that NAT-PT is a tool only to be used to assist transition to native IPv6 to IPv6 communication.

Requirements extracted from RFC 3056

Identifier: RQ_003_0001
RFC Clause: 2
Type: Mandatory
Applies to: Node

Requirement:
To enable communication between IPv6 sites over the IPv4 network using the mechanism described in RFC 3056, IPv6 Addresses shall conform to the following structure:

Bit	Field
1 - 3	Format Prefix
4 - 16	Top-Level Aggregation Identifier
17 - 48	Next-Level Aggregation Identifier
49 - 64	Site-Level Agregation Identifier
65 - 128	Interface Identifier

Specification Text:
The subscriber site is then deemed to have the following IPv6 address prefix, without any further assignment procedures being necessary:

Prefix length: 48 bits
Format prefix: 001
TLA value: 0x0002
NLA value: V4ADDR

This is illustrated as follows:

```
| 3 |  13  |    32     |   16   |          64 bits               |
+---+------+-----------+--------+--------------------------------+
|FP | TLA  | V4ADDR    | SLA ID |         Interface ID           |
|001|0x0002|           |        |                                |
+---+------+-----------+--------+--------------------------------+
```

Thus, this prefix has exactly the same format as normal /48 prefixes assigned according to [AGGR]. It can be abbreviated as 2002:V4ADDR::/48. Within the subscriber site it can be used exactly like any other valid IPv6 prefix, e.g., for automated address assignment and discovery according to the normal mechanisms such as [CONF, DISC], for native IPv6 routing, or for the "6over4" mechanism [6OVER4].

Identifier: RQ_003_0002
RFC Clause: 2
Type: Mandatory
Applies to: Node

Requirement:
To enable communication between IPv6 sites over the IPv4 network using the mechanism described in RFC 3056, the Format Prefix field of the IPv6 Addresses shall be 001.

Specification Text:
The subscriber site is then deemed to have the following IPv6 address prefix, without any further assignment procedures being necessary:

Prefix length: 48 bits
Format prefix: 001
TLA value: 0x0002
NLA value: V4ADDR

This is illustrated as follows:

```
| 3 |  13  |    32     |   16   |          64 bits               |
+---+------+-----------+--------+--------------------------------+
|FP | TLA  | V4ADDR    | SLA ID |         Interface ID           |
|001|0x0002|           |        |                                |
+---+------+-----------+--------+--------------------------------+
```

Thus, this prefix has exactly the same format as normal /48 prefixes assigned according to [AGGR]. It can be abbreviated as 2002:V4ADDR::/48. Within the subscriber site it can be used exactly like any other valid IPv6 prefix, e.g., for automated address assignment and discovery according to the normal mechanisms such as [CONF, DISC], for native IPv6 routing, or for the "6over4" mechanism [6OVER4].

Identifier: RQ_003_0003
RFC Clause: 2
Type: Mandatory
Applies to: Node

Requirement:
To enable communication between IPv6 sites over the IPv4 network using the mechanism described in RFC 3056, the Top-Level Aggregation Identifier field of the IPv6 Addresses shall be 0x002.

Specification Text:
The subscriber site is then deemed to have the following IPv6 address prefix, without any further assignment procedures being necessary:

Prefix length: 48 bits
Format prefix: 001
TLA value: 0x0002
NLA value: V4ADDR

This is illustrated as follows:

```
| 3 |  13  |    32     |   16   |          64 bits               |
+---+------+-----------+--------+--------------------------------+
|FP | TLA  | V4ADDR    | SLA ID |         Interface ID           |
|001|0x0002|           |        |                                |
+---+------+-----------+--------+--------------------------------+
```

Thus, this prefix has exactly the same format as normal /48 prefixes assigned according to [AGGR]. It can be abbreviated as 2002:V4ADDR::/48. Within the subscriber site it can be used exactly like any other valid IPv6 prefix, e.g., for automated address assignment and discovery according to the normal mechanisms such as [CONF, DISC], for native IPv6 routing, or for the "6over4" mechanism [6OVER4].

Identifier: RQ_003_0004
RFC Clause: 2
Type: Mandatory
Applies to: Node

Requirement:
To enable communication between IPv6 sites over the IPv4 network using the mechanism described in RFC 3056, the Next-Level Aggregation Identifier field of the IPv6 Addresses shall be a Globally Unique 32-bit IPv4 address.

Specification Text:
The subscriber site is then deemed to have the following IPv6 address prefix, without any further assignment procedures being necessary:

```
   Prefix length: 48 bits
   Format prefix: 001
   TLA value: 0x0002
   NLA value: V4ADDR
```

This is illustrated as follows:

```
| 3 |  13  |    32     |   16   |          64 bits               |
+---+------+-----------+--------+--------------------------------+
|FP | TLA  | V4ADDR    | SLA ID |         Interface ID           |
|001|0x0002|           |        |                                |
+---+------+-----------+--------+--------------------------------+
```

Thus, this prefix has exactly the same format as normal /48 prefixes assigned according to [AGGR]. It can be abbreviated as 2002:V4ADDR::/48. Within the subscriber site it can be used exactly like any other valid IPv6 prefix, e.g., for automated address assignment and discovery according to the normal mechanisms such as [CONF, DISC], for native IPv6 routing, or for the "6over4" mechanism [6OVER4].

Identifier: RQ_003_0005
RFC Clause: 2.1
Type: Recommendation
Applies to: Host

Requirement:
If one host has only a 6to4 address, and the other one has both a 6to4 and a native IPv6 address, then the 6to4 address should be used for both.

Specification Text:
If one host has only a 6to4 address, and the other one has both a 6to4 and a native IPv6 address, then the 6to4 address should be used for both.

Identifier: RQ_003_0006
RFC Clause: 2.1
Type: Recommendation
Applies to: Host

Requirement:
If both hosts have a 6to4 address and a native IPv6 address, the default configuration should be native IPv6 for both.

Specification Text:
If both hosts have a 6to4 address and a native IPv6 address, then either the 6to4 address should be used for both, or the native IPv6 address should be used for both. The choice should be configurable. **The default configuration should be native IPv6 for both.**

Identifier: RQ_003_0007
RFC Clause: 2.1
Type: Recommendation
Applies to: Host

Requirement:
If both hosts have a 6to4 address and a native IPv6 address, then either the 6to4 address should be used for both, or the native IPv6 address should be used for both.

Specification Text:
If both hosts have a 6to4 address and a native IPv6 address, then either the 6to4 address should be used for both, or the native IPv6 address should be used for both. The choice should be configurable. The default configuration should be native IPv6 for both.

Identifier: RQ_003_0008
RFC Clause: 2.1
Type: Recommendation
Applies to: Host

Requirement:
If both hosts have a 6to4 address and a native IPv6 address, then the choice to use, either the 6to4 address for both, or the native IPv6 address for both, should be configurable.

Specification Text:
If both hosts have a 6to4 address and a native IPv6 address, then either the 6to4 address should be used for both, or the native IPv6 address should be used for both. The choice should be configurable. The default configuration should be native IPv6 for both.

Identifier: RQ_003_0010
RFC Clause: 3
Type: Mandatory
Applies to: Router

Requirement:
IPv6 packets SHALL be transmitted in IPv4 packets [RFC 791] with either, or both, of its Destination and Source IPv4 addresses identical to the V4ADDR field of an IPv6 prefix.

Specification Text:
IPv6 packets are transmitted in IPv4 packets [RFC 791] with an IPv4 protocol type of 41, the same as has been assigned [MECH] for IPv6 packets that are tunneled inside of IPv4 frames. **The IPv4 header contains the Destination and Source IPv4 addresses. One or both of these will be identical to the V4ADDR field of an IPv6 prefix formed as specified above (see section 5 for more details).** The IPv4 packet body contains the IPv6 header and payload.

Identifier: RQ_003_0011
RFC Clause: 3
Type: Mandatory
Applies to: Router

Requirement:
IPv6 packets SHALL be transmitted in IPv4 packets [RFC 791] with the IPv4 packet body containing the IPv6 header and payload.

Specification Text:
IPv6 packets are transmitted in IPv4 packets [RFC 791] with an IPv4 protocol type of 41, the same as has been assigned [MECH] for IPv6 packets that are tunneled inside of IPv4 frames. The IPv4 header contains the Destination and Source IPv4 addresses. One or both of these will be identical to the V4ADDR field of an IPv6 prefix formed as specified above (see section 5 for more details). **The IPv4 packet body contains the IPv6 header and payload.**

Identifier: RQ_003_0009
RFC Clause: 3
Type: Mandatory
Applies to: Router

Requirement:
IPv6 packets which are transmitted in IPv4 packets [RFC 791] SHALL use an IPv4 protocol type of 41,

Specification Text:
IPv6 packets are transmitted in IPv4 packets [RFC 791] with an IPv4 protocol type of 41, the same as has been assigned [MECH] for IPv6 packets that are tunneled inside of IPv4 frames. The IPv4 header contains the Destination and Source IPv4 addresses. One or both of these will be identical to the V4ADDR field of an IPv6 prefix formed as specified above (see section 5 for more details). The IPv4 packet body contains the IPv6 header and payload.

Identifier: RQ_003_0012
RFC Clause: 4
Type: Mandatory
Applies to: Router

Requirement:
The IPv4 "do not fragment" bit SHOULD NOT be set in the encapsulating IPv4 header.

Specification Text:
If the IPv6 MTU size proves to be too large for some intermediate IPv4 subnet, IPv4 fragmentation will ensue. While undesirable, this is not necessarily disastrous, unless the fragments are delivered to different IPv4 destinations due to some form of IPv4 anycast. **The IPv4 "do not fragment" bit SHOULD NOT be set in the encapsulating IPv4 header.**

Identifier: RQ_003_0013
RFC Clause: 5.1
Type: Mandatory
Applies to: Router

Requirement:
Any of the 6to4 sites SHALL able to send IPv4 packets with protocol type 41 to any of the other 6to4 sites.

Specification Text:
The simplest deployment scenario for 6to4 is to use it between a number of sites, each of which has at least one connection to a shared IPv4 Internet. This could be the global Internet, or it could be a corporate IP network. In the case of the global Internet, there is no requirement that the sites all connect to the same Internet service provider. **The only requirement is that any of the sites is able to send IPv4 packets with protocol type 41 to any of the others.** By definition, each site has an IPv6 prefix in the format defined in Section 2. It will therefore create DNS records for these addresses. For example, site A which owns IPv4 address 192.1.2.3 will create DNS records with the IPv6 prefix {FP=001,TLA=0x0002,NLA=192.1.2.3}/48 (i.e., 2002:c001:0203::/48). Site B which owns address 9.254.253.252 will create DNS records with the IPv6 prefix {FP=001,TLA=0x0002,NLA=9.254.253.252}/48 (i.e., 2002:09fe:fdfc::/48).

Identifier: RQ_003_0014
RFC Clause: 5.1
Type: Mandatory
Applies to: Router

Requirement:
Within a 6to4 site, addresses with the 2002::/16 prefix, apart from those with the local 2002:V4ADDR::/48 prefix, will be handled like any other non-local IPv6 address, i.e., by a default or explicit route towards the 6to4 border router.

Specification Text:
Within a 6to4 site, addresses with the 2002::/16 prefix, apart from those with the local 2002:V4ADDR::/48 prefix, will be handled like any other non-local IPv6 address, i.e., by a default or explicit route towards the 6to4 border router.

Identifier: RQ_003_0015
RFC Clause: 5.1
Type: Recommendation
Applies to: Router

Requirement:
The 6to4 router SHOULD also generate the appropriate IPv6 prefix announcements [RFC2462, RFC 2461].

Specification Text:
In this scenario, any number of 6to4 sites can interoperate with no tunnel configuration, and no special requirements from the IPv4 service. All that is REQUIRED is the appropriate DNS entries and the additional sending and decapsulation rules configured in the 6to4 router. **This router SHOULD also generate the appropriate IPv6 prefix announcements [CONF, DISC].**

Identifier: RQ_003_0016
RFC Clause: 5.1
Type: Recommendation
Applies to: Router

Requirement:
It is RECOMMENDED that each site should only use one IPv4 address per 6to4 routerant that should be the address assigned to the external interface of that 6to4 router.

Specification Text:
It is RECOMMENDED that in any case each site should use only one IPv4 address per 6to4 router, and that should be the address assigned to the external interface of the 6to4 router. Single-homed sites therefore SHOULD use only one IPv4 address for 6to4 routing. Multi- homed sites are discussed briefly in section 5.6.

Identifier: RQ_003_0040
RFC Clause: 5.11
Type: Mandatory
Applies to: Router

Requirement:
The 2002::/16 prefix MUST NOT be advertised to a 6to4 exterior routing domain.

Specification Text:
The 2002::/16 routing prefix may be legitimately advertised into the native IPv6 routing domain by a relay router, and into an IPv6 site's local IPv6 routing domain; hence there is a risk of misconfiguration causing it to be advertised into a 6to4 exterior routing domain.

To summarize, **the 2002::/16 prefix MUST NOT be advertised to a 6to4 exterior routing domain.**

Identifier: RQ_003_0017
RFC Clause: 5.2
Type: Optional
Applies to: Router

Requirement:
The relay router MAY apply source address based filters to accept traffic only from specific 6to4 routers.

Specification Text:
We now have three distinct classes of routing domain to consider:

1. the internal IPv6 routing domain of each 6to4 site;
2. an exterior IPv6 routing domain interconnecting a given set of 6to4 border routers,
including relay routers, among themselves, i.e., a 6to4 exterior routing domain;
3. the exterior IPv6 routing domain of each native IPv6 island.

1. The internal routing domain of a 6to4 site behaves as described in section 5.1.

2. There are two deployment options for a 6to4 exterior routing domain:

2.1 No IPv6 exterior routing protocol is used. The 6to4 routers using a given relay router each have a default IPv6 route pointing to the relay router. **The relay router MAY apply source address based filters to accept traffic only from specific 6to4 routers.**

2.2 An IPv6 exterior routing protocol is used. The set of 6to4 routers using a given relay router obtain native IPv6 routes from the relay router using a routing protocol such as BGP4+ [RFC 2283, BGP4+]. The relay router will advertise whatever native IPv6 routing prefixes are appropriate on its 6to4 pseudo-interface. These prefixes will indicate the regions of native IPv6 topology that the relay router is willing to relay to. Their choice is a matter of routing policy. It is necessary for network operators to carefully consider desirable traffic patterns and topology when choosing the scope of such routing advertisements. The relay router will establish BGP peering only with specific 6to4 routers whose traffic it is willing to accept.

Although this solution is more complex, it provides effective policy control, i.e., BGP4+ policy determines which 6to4 routers are able to use which relay router.

3. A relay router MUST advertise a route to 2002::/16 into the native IPv6 exterior routing domain. It is a matter of routing policy how far this routing advertisement of 2002::/16 is propagated in the native IPv6 routing system. Since there will in general be multiple relay routers advertising it, network operators will require to filter it in a managed way. Incorrect policy in this area will lead to potential unreachability or to perverse traffic patterns.

Identifier:	RQ_003_0018
RFC Clause:	5.2
Type:	Mandatory
Applies to:	Router

Requirement:

A relay router MUST advertise a route to 2002::/16 into the native IPv6 exterior routing domain.

Specification Text:

We now have three distinct classes of routing domain to consider:

1. the internal IPv6 routing domain of each 6to4 site;
2. an exterior IPv6 routing domain interconnecting a given set of 6to4 border routers, including relay routers, among themselves, i.e., a 6to4 exterior routing domain;
3. the exterior IPv6 routing domain of each native IPv6 island.

1. The internal routing domain of a 6to4 site behaves as described in section 5.1.

2. There are two deployment options for a 6to4 exterior routing domain:

2.1 No IPv6 exterior routing protocol is used. The 6to4 routers using a given relay router each have a default IPv6 route pointing to the relay router. The relay router MAY apply source address based filters to accept traffic only from specific 6to4 routers.

2.2 An IPv6 exterior routing protocol is used. The set of 6to4 routers using a given relay router obtain native IPv6 routes from the relay router using a routing protocol such as BGP4+ [RFC 2283, BGP4+]. The relay router will advertise whatever native IPv6 routing prefixes are appropriate on its 6to4 pseudo-interface. These prefixes will indicate the regions of native IPv6 topology that the relay router is willing to relay to. Their choice is a matter of routing policy. It is necessary for network operators to carefully consider desirable traffic patterns and topology when choosing the scope of such routing advertisements. The relay router will establish BGP peering only with specific 6to4 routers whose traffic it is willing to accept.

Although this solution is more complex, it provides effective policy control, i.e., BGP4+ policy determines which 6to4 routers are able to use which relay router.

3**. A relay router MUST advertise a route to 2002::/16 into the native IPv6 exterior routing domain.** It is a matter of routing policy how far this routing advertisement of 2002::/16 is propagated in the native IPv6 routing system. Since there will in general be multiple relay routers advertising it, network operators will require to filter it in a managed way. Incorrect policy in this area will lead to potential unreachability or to perverse traffic patterns.

Identifier: RQ_003_0019
RFC Clause: 5.2
Type: Mandatory
Applies to: Router

Requirement:
6to4 prefixes more specific than 2002::/16 MUST NOT be propagated in native IPv6 routing.

Specification Text:
6to4 prefixes more specific than 2002::/16 MUST NOT be propagated in native IPv6 routing, to prevent pollution of the IPv6 routing table by elements of the IPv4 routing table. Therefore, a 6to4 site which also has a native IPv6 connection MUST NOT advertise its 2002::/48 routing prefix on that connection, and all native IPv6 network operators MUST filter out and discard any 2002:: routing prefix advertisements longer than /16.

Identifier: RQ_003_0020
RFC Clause: 5.2
Type: Mandatory
Applies to: Router

Requirement:
A 6to4 site which also has a native IPv6 connection MUST NOT advertise its 2002::/48 routing prefix on that connection.

Specification Text:
6to4 prefixes more specific than 2002::/16 MUST NOT be propagated in native IPv6 routing, to prevent pollution of the IPv6 routing table by elements of the IPv4 routing table. Therefore, **a 6to4 site which also has a native IPv6 connection MUST NOT advertise its 2002::/48 routing prefix on that connection,** and all native IPv6 network operators MUST filter out and discard any 2002:: routing prefix advertisements longer than /16.

Identifier: RQ_003_0021
RFC Clause: 5.2
Type: Mandatory
Applies to: Router

Requirement:
All native IPv6 network operators MUST filter out and discard any 2002:: routing prefix advertisements longer than /16.

Specification Text:
6to4 prefixes more specific than 2002::/16 MUST NOT be propagated in native IPv6 routing, to prevent pollution of the IPv6 routing table by elements of the IPv4 routing table. Therefore, a 6to4 site which also has a native IPv6 connection MUST NOT advertise its 2002::/48 routing prefix on that connection, and **all native IPv6 network operators MUST filter out and discard any 2002:: routing prefix advertisements longer than /16.**

Identifier: RQ_003_0022
RFC Clause: 5.2.2
Type: Mandatory
Applies to: Router

Requirement:
On its native IPv6 interface, the relay router MUST advertise a route to 2002::/16.

Specification Text:
On its native IPv6 interface, the relay router MUST advertise a route to 2002::/16. It MUST NOT advertise a longer 2002:: routing prefix on that interface. Routing policy within the native IPv6 routing domain determines the scope of that advertisement, thereby limiting the visibility of the relay router in that domain.

Identifier: RQ_003_0023
RFC Clause: 5.2.2
Type: Mandatory
Applies to: Router

Requirement:
On its native IPv6 interface, the relay router It MUST NOT advertise a longer 2002:: routing prefix on that interface.

Specification Text:
On its native IPv6 interface, the relay router MUST advertise a route to 2002::/16. **It MUST NOT advertise a longer 2002:: routing prefix on that interface**. Routing policy within the native IPv6 routing domain determines the scope of that advertisement, thereby limiting the visibility of the relay router in that domain.

Identifier: RQ_003_0024
RFC Clause: 5.2.2.3
Type: Recommendation
Applies to: Router

Requirement:
A relay router should not attempt to serve more sites than any other transit router, allowing for the encapsulation overhead.

Specification Text:
Relay routers introduce the potential for scaling issues. In general a relay router should not attempt to serve more sites than any other transit router, allowing for the encapsulation overhead.

Identifier: RQ_003_0025
RFC Clause: 5.2.3
Type: Mandatory
Applies to: Router

Requirement:
A router with both 6to4 pseudo- interfaces and native IPv6 interfaces, which is unwilling to act as a relay router, MUST NOT advertise any 2002:: routing prefix into the native IPv6 domain.

Specification Text:
It may arise that a site has a router with both 6to4 pseudo- interfaces and native IPv6 interfaces, but is unwilling to act as a relay router. Such a site MUST NOT advertise any 2002:: routing prefix into the native IPv6 domain and MUST NOT advertise any native IPv6 routing prefixes or a default IPv6 route into the 6to4 domain. Within the 6to4 domain it will behave exactly as in the basic 6to4 scenario of Section 5.1.

Identifier: RQ_003_0026
RFC Clause: 5.2.3
Type: Mandatory
Applies to: Router

Requirement:
A router with both 6to4 pseudo- interfaces and native IPv6 interfaces, which is unwilling to act as a relay router, MUST NOT advertise any native IPv6 routing prefixes or a default IPv6 route into the 6to4 domain.

Specification Text:
It may arise that a site has a router with both 6to4 pseudo- interfaces and native IPv6 interfaces, but is unwilling to act as a relay router. Such a site MUST NOT advertise any 2002:: routing prefix into the native IPv6 domain and **MUST NOT advertise any native IPv6 routing prefixes or a default IPv6 route into the 6to4 domain.** Within the 6to4 domain it will behave exactly as in the basic 6to4 scenario of Section 5.1.

Identifier: RQ_003_0027
RFC Clause: 5.3
Type: Mandatory
Applies to: Router

Requirement:
Every 6to4 router, if the next hop IPv6 address for an IPv6 packet does match the prefix 2002::/16, and does not match any prefix of the local site, first MUST apply any security checks (see Section 8).

Specification Text:
ADDITIONAL SENDING RULE for 6to4 routers

```
if the next hop IPv6 address for an IPv6 packet
   does match the prefix 2002::/16, and
   does not match any prefix of the local site
       then
       apply any security checks (see Section 8);

               encapsulate the packet in IPv4 as in Section 3,
       with IPv4 destination address = the NLA value V4ADDR
       extracted from the next hop IPv6 address;
       queue the packet for IPv4 forwarding.
```

Identifier: RQ_003_0028
RFC Clause: 5.3
Type: Mandatory
Applies to: Router

Requirement:
Every 6to4 router, if the next hop IPv6 address for an IPv6 packet does match the prefix 2002::/16, and does not match any prefix of the local site, after applying any security checks, MUST encapsulate the packet in IPv4 as in Section 3, with IPv4 destination address = the NLA value V4ADDR extracted from the next hop IPv6 address.

Specification Text:
ADDITIONAL SENDING RULE for 6to4 routers

```
if the next hop IPv6 address for an IPv6 packet
   does match the prefix 2002::/16, and
   does not match any prefix of the local site
       then
       apply any security checks (see Section 8);
               encapsulate the packet in IPv4 as in Section 3,
       with IPv4 destination address = the NLA value V4ADDR
       extracted from the next hop IPv6 address;
       queue the packet for IPv4 forwarding.
```

Identifier: RQ_003_0029
RFC Clause: 5.3
Type: Mandatory
Applies to: Router

Requirement:
Every 6to4 router, if the next hop IPv6 address for an IPv6 packet does match the prefix 2002::/16, and does not match any prefix of the local site, after applying any security checks, and encapsulating the packet, MUST queue the packet for IPv4 forwarding.

Specification Text:

```
ADDITIONAL SENDING RULE for 6to4 routers

if the next hop IPv6 address for an IPv6 packet
   does match the prefix 2002::/16, and
   does not match any prefix of the local site
       then
       apply any security checks (see Section 8);
               encapsulate the packet in IPv4 as in Section 3,
       with IPv4 destination address = the NLA value V4ADDR
       extracted from the next hop IPv6 address;
       queue the packet for IPv4 forwarding.
```

Identifier: RQ_003_0030
RFC Clause: 5.3
Type: Mandatory
Applies to: Router

Requirement:

Every 6to4 router, for incoming IPv4 packets with protocol type 41, MUST apply any security checks (see Section 8).

Specification Text:

```
 A simple decapsulation rule for incoming IPv4 packets with protocol type 41 MUST be implemented:

 ADDITIONAL DECAPSULATION RULE for 6to4 routers

   apply any security checks (see Section 8);
   remove the IPv4 header;
   submit the packet to local IPv6 routing.
```

Identifier: RQ_003_0031
RFC Clause: 5.3
Type: Mandatory
Applies to: Router

Requirement:

Every 6to4 router, for incoming IPv4 packets with protocol type 41, after applying any security checks, MUST remove the IPv4 header.

Specification Text:

```
 A simple decapsulation rule for incoming IPv4 packets with protocol type 41 MUST be implemented:

 ADDITIONAL DECAPSULATION RULE for 6to4 routers

   apply any security checks (see Section 8);
   remove the IPv4 header;
   submit the packet to local IPv6 routing.
```

Identifier: RQ_003_0032
RFC Clause: 5.3
Type: Mandatory
Applies to: Router

Requirement:

Every 6to4 router, for incoming IPv4 packets with protocol type 41, after applying any security checks and removing the IPv4 header MUST submit the packet to local IPv6 routing.

Specification Text:

```
A simple decapsulation rule for incoming IPv4 packets with protocol type 41 MUST be implemented:

 ADDITIONAL DECAPSULATION RULE for 6to4 routers

   apply any security checks (see Section 8);
   remove the IPv4 header;
   submit the packet to local IPv6 routing.
```

Identifier: RQ_003_0033
RFC Clause: 5.5
Type: Mandatory
Applies to: Node

Requirement:
within the native IPv6 world, the scope of 2002::/16 routing advertisements MUST be correctly defined by routing policy to ensure that traffic to 2002::/16 follows the intended paths.

Specification Text:

If there are multiple relay routers between native IPv6 and the 6to4 world, different parts of the 6to4 world will be served by different relays. **The only complexity that this introduces is in the scoping of 2002::/16 routing advertisements within the native IPv6 world. Like any BGP4+ advertisements, their scope MUST be correctly defined by routing policy to ensure that traffic to 2002::/16 follows the intended paths.**

Identifier: RQ_003_0034
RFC Clause: 5.5
Type: Mandatory
Applies to: Node

Requirement:
If multiple IPv6 stubs are interconnected through multiple, disjoint IPv4 networks then the 6to4 world is also fragmented; this is the one scenario that MUST be avoided.

Specification Text:
If multiple IPv6 stubs are interconnected through multiple, disjoint IPv4 networks (i.e., a fragmented IPv4 world) then the 6to4 world is also fragmented; this is the one scenario that MUST be avoided. It is illustrated below to show why it does not work, since the 2002::/16 advertisement from Relay1 will be invisible to Relay2, and vice versa. Sites A and B therefore have no connectivity to sites C and D.

Identifier: RQ_003_0035
RFC Clause: 5.6
Type: Optional
Applies to: Node

Requirement:
Sites which are multihomed on IPv4 MAY extend the 6to4 scenario by using a 2002:: prefix for each IPv4 border router.

Specification Text:
Sites which are multihomed on IPv4 MAY extend the 6to4 scenario by using a 2002:: prefix for each IPv4 border router, thereby obtaining a simple form of IPv6 multihoming by using multiple simultaneous IPv6 prefixes and multiple simultaneous relay routers.

Identifier: RQ_003_0036
RFC Clause: 5.8
Type: Mandatory
Applies to: Router

Requirement:
If the site concerned has very limited global IPv4 address space, and is running an IPv4 network address translator (NAT). The NAT box MUST also contain a fully functional IPv6 router including the 6to4 mechanism.

Specification Text:
If the site concerned has very limited global IPv4 address space, and is running an IPv4 network address translator (NAT), all of the above mechanisms remain valid. The NAT box MUST also contain a fully functional IPv6 router including the 6to4 mechanism. The address used for V4ADDR will simply be a globally unique IPv4 address allocated to the NAT. In the example of Section 5.1 above, the 6to4 routers would also be the sites' IPv4 NATs, which would own the globally unique IPv4 addresses 192.1.2.3 and 9.254.253.252.

Identifier: RQ_003_0037
RFC Clause: 5.8
Type: Optional
Applies to: Router

Requirement:
If a 6to4 border router is combined with an RSIP border router, it can support IPv6 hosts using 6to4 addresses, IPv4 hosts using RSIP, or dual stack hosts using both.

Specification Text:
The Realm-Specific IP (RSIP) mechanism [RSIP] can also co-exist with 6to4. If a 6to4 border router is combined with an RSIP border router, it can support IPv6 hosts using 6to4 addresses, IPv4 hosts using RSIP, or dual stack hosts using both. The RSIP function provides fine-grained management of dynamic global IPv4 address allocation and the 6to4 function provides a stable IPv6 global address to each host. As with NAT, the IPv4 address used to construct the site's 2002: prefix will be one of the global addresses of the RSIP border router.

Identifier: RQ_003_0038
RFC Clause: 5.9
Type: Mandatory
Applies to: Node

Requirement:
Within a private corporate network as part of its internal transition to IPv6, the V4ADDR MUST be a duly allocated global IPv4 address.

Specification Text:
There is nothing to stop the above scenario being deployed within a private corporate network as part of its internal transition to IPv6; the corporate IPv4 backbone would serve as the virtual link layer for individual corporate sites using 2002:: prefixes. **The V4ADDR MUST be a duly allocated global IPv4 address,** which MUST be unique within the private network. The Intranet thereby obtains globally unique IPv6 addresses even if it is internally using private IPv4 addresses [RFC 1918].

Identifier: RQ_003_0039
RFC Clause: 5.9
Type: Mandatory
Applies to: Node

Requirement:
Within a private corporate network as part of its internal transition to IPv6 the V4ADDR MUST be unique within the private network.

Specification Text:
There is nothing to stop the above scenario being deployed within a private corporate network as part of its internal transition to IPv6; the corporate IPv4 backbone would serve as the virtual link layer for individual corporate sites using 2002:: prefixes. **The V4ADDR MUST be a duly allocated global IPv4 address, which MUST be unique within the private network. The Intranet thereby obtains globally unique IPv6 addresses even if it is internally using private IPv4 addresses [RFC 1918].**

Identifier: RQ_003_0041
RFC Clause: 6
Type: Mandatory
Applies to: Node

Requirement:
The 6to4 mechanism MUST assume only unicast capability in its underlying IPv4 carrier network.

Specification Text:
It is not possible to assume the general availability of wide-area IPv4 multicast, so (unlike [6OVER4]) **the 6to4 mechanism MUST assume only unicast capability in its underlying IPv4 carrier network.** An IPv6 multicast routing protocol is needed [MULTI].

Identifier: RQ_003_0042
RFC Clause: 6
Type: Optional
Applies to: Node

Requirement:
Anycast addresses formed with 2002:: prefixes may be used inside a 6to4 site.

Specification Text:
The allocated anycast address space [ANYCAST] is compatible with 2002:: prefixes, i.e., anycast addresses formed with such prefixes may be used inside a 6to4 site.

Identifier: RQ_003_0043
RFC Clause: 9
Type: Recommendation
Applies to: Node

Requirement:
The use of IP security at both IPv4 and IPv6 levels should be avoided, for efficiency reasons.

Specification Text:
Implementors should be aware that, in addition to possible attacks against IPv6, security attacks against IPv4 MUST also be considered. **Use of IP security at both IPv4 and IPv6 levels should nevertheless be avoided, for efficiency reasons.** For example, if IPv6 is running encrypted, encryption of IPv4 would be redundant except if traffic analysis is felt to be a threat. If IPv6 is running authenticated, then authentication of IPv4 will add little. Conversely, IPv4 security will not protect IPv6 traffic once it leaves the 6to4 domain. Therefore, implementing IPv6 security is REQUIRED even if IPv4 security is available.

Identifier: RQ_003_0044
RFC Clause: 9
Type: Mandatory
Applies to: Node

Requirement:
Implementing IPv6 security is REQUIRED even if IPv4 security is available.

Specification Text:
Implementors should be aware that, in addition to possible attacks against IPv6, security attacks against IPv4 MUST also be considered. Use of IP security at both IPv4 and IPv6 levels should nevertheless be avoided, for efficiency reasons. For example, if IPv6 is running encrypted, encryption of IPv4 would be redundant except if traffic analysis is felt to be a threat. If IPv6 is running authenticated, then authentication of IPv4 will add little. Conversely, IPv4 security will not protect IPv6 traffic once it leaves the 6to4 domain. **Therefore, implementing IPv6 security is REQUIRED even if IPv4 security is available.**

Identifier: RQ_003_0045
RFC Clause: 9
Type: Mandatory
Applies to: Node

Requirement:
In addition to possible attacks against IPv6, security attacks against IPv4 MUST also be considered.

Specification Text:
Implementors should be aware that, in addition to possible attacks against IPv6, security attacks against IPv4 MUST also be considered. Use of IP security at both IPv4 and IPv6 levels should nevertheless be avoided, for efficiency reasons. For example, if IPv6 is running encrypted, encryption of IPv4 would be redundant except if traffic analysis is felt to be a threat. If IPv6 is running authenticated, then authentication of IPv4 will add little. Conversely, IPv4 security will not protect IPv6 traffic once it leaves the 6to4 domain. Therefore, implementing IPv6 security is REQUIRED even if IPv4 security is available.

Identifier: RQ_003_0046
RFC Clause: 9
Type: Optional
Applies to: Host

Requirement:
If the acceptance of 6to4 traffic from any source from which regular IPv4 traffic is accepted is felt to be a security risk, additional source address based packet filtering could be applied.

Specification Text:
By default, 6to4 traffic will be accepted and decapsulated from any source from which regular IPv4 traffic is accepted. If this is for any reason felt to be a security risk (for example, if IPv6 spoofing is felt to be more likely than IPv4 spoofing), then additional source address based packet filtering could be applied. A possible plausibility check is whether the encapsulating IPv4 address is consistent with the encapsulated 2002:: address. If this check is applied, exceptions to it MUST be configured to admit traffic from relay routers (Section 5). 2002:: traffic MUST also be excepted from checks applied to prevent spoofing of "6 over 4" traffic [6OVER4].

Identifier: RQ_003_0047
RFC Clause: 9
Type: Mandatory
Applies to: Host

Requirement:
If additional source address based packet filtering is applied to check whether the encapsulating IPv4 address is consistent with the encapsulated 2002:: address, then exceptions to it MUST be configured to admit traffic from relay routers (Section 5).

Specification Text:
By default, 6to4 traffic will be accepted and decapsulated from any source from which regular IPv4 traffic is accepted. If this is for any reason felt to be a security risk (for example, if IPv6 spoofing is felt to be more likely than IPv4 spoofing), then additional source address based packet filtering could be applied. A possible plausibility check is whether the encapsulating IPv4 address is consistent with the encapsulated 2002:: address. If this check is applied, exceptions to it MUST be configured to admit traffic from relay routers (Section 5). 2002:: traffic MUST also be excepted from checks applied to prevent spoofing of "6 over 4" traffic [6OVER4].

Identifier: RQ_003_0048
RFC Clause: 9
Type: Mandatory
Applies to: Host

Requirement:
2002:: traffic MUST be excepted from checks applied to prevent spoofing of "6 over 4" traffic [6OVER4].

Specification Text:
By default, 6to4 traffic will be accepted and decapsulated from any source from which regular IPv4 traffic is accepted. If this is for any reason felt to be a security risk (for example, if IPv6 spoofing is felt to be more likely than IPv4 spoofing), then additional source address based packet filtering could be applied. A possible plausibility check is whether the encapsulating IPv4 address is consistent with the encapsulated 2002:: address. If this check is applied, exceptions to it MUST be configured to admit traffic from relay routers (Section 5). **2002:: traffic MUST also be excepted from checks applied to prevent spoofing of "6 over 4" traffic [6OVER4].**

Identifier: RQ_003_0049
RFC Clause: 9
Type: Mandatory
Applies to: Router

Requirement:
Any 6to4 traffic whose source address embeds a V4ADDR which is not in the format of a global unicast address MUST be silently discarded by both encapsulators and decapsulators.

Specification Text:
In any case, **any 6to4 traffic whose source** or destination **address embeds a V4ADDR which is not in the format of a global unicast address MUST be silently discarded by both encapsulators and decapsulators.** Specifically, this means that IPv4 addresses defined in [RFC 1918], broadcast, subnet broadcast, multicast and loopback addresses are unacceptable.

Identifier: RQ_003_0050
RFC Clause: 9
Type: Mandatory
Applies to: Router

Requirement:
Any 6to4 traffic whose destination address embeds a V4ADDR which is not in the format of a global unicast address MUST be silently discarded by both encapsulators and decapsulators.

Specification Text:
In any case, any 6to4 traffic whose source or **destination address embeds a V4ADDR which is not in the format of a global unicast address MUST be silently discarded by both encapsulators and decapsulators.** Specifically, this means that IPv4 addresses defined in [RFC 1918], broadcast, subnet broadcast, multicast and loopback addresses are unacceptable.

Requirements extracted from RFC 3596

Identifier: RQ_003_5001
RFC Clause: 2
Type: Mandatory
Applies to: Host

Requirement:
A record type is defined to store a host's IPv6 address. A host that has more than one IPv6 address MUST have more than one such record.

Specification Text:
A record type is defined to store a host's IPv6 address. A host that has more than one IPv6 address MUST have more than one such record.

Identifier: RQ_003_5002
RFC Clause: 2.1
Type: Mandatory
Applies to: Host

Requirement:
The AAAA resource record type is a record specific to the Internet class that stores a single IPv6 address.

Specification Text:
The AAAA resource record type is a record specific to the Internet class that stores a single IPv6 address

----

Identifier: RQ_003_5003
RFC Clause: 2.1
Type: Mandatory
Applies to: Host

Requirement:
The AAAA resource record type has the IANA assigned value of 28 (decimal).

Specification Text:
The IANA assigned value of the type is 28 (decimal).

Identifier: RQ_003_5004
RFC Clause: 2.2
Type: Mandatory
Applies to: Host

Requirement:
A 128 bit IPv6 address is encoded in the data portion of an AAAA resource record in network byte order (high-order byte first).

Specification Text:
A 128 bit IPv6 address is encoded in the data portion of an AAAA resource record in network byte order (high-order byte first).

Identifier: RQ_003_5005
RFC Clause: 2.3
Type: Mandatory
Applies to: Host

Requirement:
An AAAA query for a specified domain name in the Internet class returns all associated AAAA resource records in the answer section of a response.

Specification Text:
An AAAA query for a specified domain name in the Internet class returns all associated AAAA resource records in the answer section of a response.

Identifier: RQ_003_5006
RFC Clause: 2.3
Type: Mandatory
Applies to: Host

Requirement:
A type AAAA query does not trigger additional section processing.

Specification Text:
A type AAAA query does not trigger additional section processing.

Identifier: RQ_003_5007
RFC Clause: 2.4
Type: Mandatory
Applies to: Host

Requirement:
The textual representation of the data portion of the AAAA resource record used in a master database file is the textual representation of an IPv6 address as defined in [RFC 3513].

Specification Text:
The textual representation of the data portion of the AAAA resource record used in a master database file is the textual representation of an IPv6 address as defined in [RFC 3513].

Identifier: RQ_003_5008
RFC Clause: 2.5
Type: Mandatory
Applies to: Host

Requirement:
An IPv6 address is represented as a name in the IP6.ARPA domain by a sequence of nibbles, each represented by a hexadecimal digit, separated by dots, encoded in reverse order and with the suffix ".IP6.ARPA".

Specification Text:
An IPv6 address is represented as a name in the IP6.ARPA domain by a sequence of nibbles separated by dots with the suffix ".IP6.ARPA". The sequence of nibbles is encoded in reverse order, i.e., the low-order nibble is encoded first, followed by the next low-order nibble and so on. Each nibble is represented by a hexadecimal digit. For example, the reverse lookup domain name corresponding to the address 4321:0:1:2:3:4:567:89ab would be

```
  b.a.9.8.7.6.5.0.4.0.0.0.3.0.0.0.2.0.0.0.1.0.0.0.0.0.0.0.1.2.3.4.IP6.ARPA.
```

Identifier: RQ_003_5009
RFC Clause: 3
Type: Mandatory
Applies to: Host

Requirement:
All existing query types that perform type A additional section processing, MUST be redefined to perform both type A and type AAAA additional section processing.

Specification Text:
All existing query types that perform type A additional section processing, i.e., name server (NS), location of services (SRV) and mail exchange (MX) query types, MUST be redefined to perform both type A and type AAAA additional section processing. These definitions mean that a name server MUST add any relevant IPv4 addresses and any relevant IPv6 addresses available locally to the additional section of a response when processing any one of the above queries.

Identifier: RQ_003_5010
RFC Clause: 3
Type: Mandatory
Applies to: Host

Requirement:
A name server MUST add any relevant IPv4 addresses and any relevant IPv6 addresses available locally to the additional section of a response when processing name server (NS), location of services (SRV) and mail exchange (MX) queries.

Specification Text:
All existing query types that perform type A additional section processing, i.e., name server (NS), location of services (SRV) and mail exchange (MX) query types, MUST be redefined to perform both type A and type AAAA additional section processing. **These definitions mean that a name server MUST add any relevant IPv4 addresses and any relevant IPv6 addresses available locally to the additional section of a response when processing any one of the above queries.**

Identifier: RQ_003_5011
RFC Clause: 4
Type: Mandatory
Applies to: Host

Requirement:
Any information obtained from the DNS MUST be regarded as unsafe unless techniques specified in [RFC 2535] or [RFC 2845] are used.

Specification Text:
Any information obtained from the DNS MUST be regarded as unsafe unless techniques specified in [7] or [8] are used. The definitions of the AAAA record type and of the IP6.ARPA domain do not change the model for use of these techniques.

Requirements extracted from RFC 4213

Identifier: RQ_003_4001
RFC Clause: 2
Type: Optional
Applies to: Node

Requirement:
Even though a node MAY be equipped to support both IPv4 and IPv6 protocols, one or the other stack MAY be disabled for operational reasons.

Specification Text:
Even though a node MAY be equipped to support both protocols, one or the other stack MAY be disabled for operational reasons. Here we use a rather loose notion of "stack". A stack being enabled has IP addresses assigned, but whether or not any particular application is available on the stacks is explicitly not defined. Thus, IPv6/IPv4 nodes MAY be operated in one of three modes:

- With their IPv4 stack enabled and their IPv6 stack disabled.
- With their IPv6 stack enabled and their IPv4 stack disabled.
- With both stacks enabled.

Identifier: RQ_003_4002
RFC Clause: 2
Type: Optional
Applies to: Node

Requirement:
IPv6/IPv4 nodes MAY provide a configuration switch to disable either their IPv4 or IPv6 stack.

Specification Text:
IPv6/IPv4 nodes with their IPv6 stack disabled will operate like IPv4-only nodes. Similarly, IPv6/IPv4 nodes with their IPv4 stacks disabled will operate like IPv6-only nodes. **IPv6/IPv4 nodes MAY provide a configuration switch to disable either their IPv4 or IPv6 stack.**

Identifier: RQ_003_4003
RFC Clause: 2
Type: Optional
Applies to: Node

Requirement:
The configured tunneling technique, MAY be used in addition to the dual IP layer operation.

Specification Text:
The configured tunneling technique, which is described in Section 3, MAY or MAY not be used in addition to the dual IP layer operation.

Identifier:	RQ_003_4004
RFC Clause:	2.2
Type:	Mandatory
Applies to:	Node

Requirement:
IPv6/IPv4 nodes MUST provide resolver libraries capable of dealing with IPv4 "A" records as well as IPv6 "AAAA" records.

Specification Text:
The Domain Naming System (DNS) is used in both IPv4 and IPv6 to map between hostnames and IP addresses. A new resource record type named "AAAA" has been defined for IPv6 addresses [RFC3596]**. Since IPv6/IPv4 nodes MUST be able to interoperate directly with both IPv4 and IPv6 nodes, they MUST provide resolver libraries capable of dealing with IPv4 "A" records as well as IPv6 "AAAA" records.** Note that the lookup of A versus AAAA records is independent of whether the DNS packets are carried in IPv4 or IPv6 packets and that there is no assumption that the DNS servers know the IPv4/IPv6 capabilities of the requesting node.

Identifier:	RQ_003_4005
RFC Clause:	2.2
Type:	Optional
Applies to:	Node

Requirement:
DNS resolver libraries on IPv6/IPv4 nodes MAY order the results returned to the application in order to influence the version of IP packets used to communicate with that specific node.

Specification Text:
DNS resolver libraries on IPv6/IPv4 nodes MUST be capable of handling both AAAA and A records. However, when a query locates an AAAA record holding an IPv6 address, and an A record holding an IPv4 address, **the resolver library MAY order the results returned to the application in order to influence the version of IP packets used to communicate with that specific node -- IPv6 first, or IPv4 first.**

Identifier:	RQ_003_4006
RFC Clause:	2.2
Type:	Recommendation
Applies to:	Node

Requirement:
The applications SHOULD be able to specify whether they want IPv4, IPv6, or both records [RFC3493].

Specification Text:
The applications SHOULD be able to specify whether they want IPv4, IPv6, or both records [RFC3493]. That defines which address families the resolver looks up. If there is not an application choice, or if the application has requested both, the resolver library MUST NOT filter out any records.

Identifier:	RQ_003_4007
RFC Clause:	2.2
Type:	Mandatory
Applies to:	Node

Requirement:
If the applications does not specify whether they want IPv4, IPv6, or both records, or if the application has requested both, the resolver library MUST NOT filter out any records.

Specification Text:
The applications SHOULD be able to specify whether they want IPv4, IPv6, or both records [RFC3493]. That defines which address families the resolver looks up. **If there is not an application choice, or if the application has requested both, the resolver library MUST NOT filter out any records.**

Identifier: RQ_003_4008
RFC Clause: 3.1
Type: Mandatory
Applies to: Node

Requirement:
The encapsulation of an IPv6 datagram in IPv4 SHALL be acheived by prepending an IPv4 header to the IPv6 datagram.

Specification Text:
The encapsulation of an IPv6 datagram in IPv4 is shown below:

```
                                +-------------+
                                |    IPv4     |
                                |   Header    |
+-------------+                 +-------------+
|    IPv6     |                 |    IPv6     |
|   Header    |                 |   Header    |
+-------------+                 +-------------+
|  Transport  |                 |  Transport  |
|   Layer     |      ===>       |   Layer     |
|   Header    |                 |   Header    |
+-------------+                 +-------------+
|             |                 |             |
~    Data     ~                 ~    Data     ~
|             |                 |             |
+-------------+                 +-------------+
           Encapsulating IPv6 in IPv4
```

Identifier: RQ_003_4009
RFC Clause: 3.1
Type: Mandatory
Applies to: Node

Requirement:
The encapsulator SHALL determine when to fragment and when to report an ICMPv6 "packet too big" error back to the source.

Specification Text:
In addition to adding an IPv4 header, the encapsulator also has to handle some more complex issues:

- **Determine when to fragment and when to report an ICMPv6 "packet too big" error back to the source.**

- How to reflect ICMPv4 errors from routers along the tunnel path back to the source as ICMPv6 errors.

Identifier: RQ_003_4010
RFC Clause: 3.1
Type: Mandatory
Applies to: Node

Requirement:
The encapsulator SHALL determine how to reflect ICMPv4 errors from routers along the tunnel path back to the source as ICMPv6 errors.

Specification Text:
In addition to adding an IPv4 header, the encapsulator also has to handle some more complex issues:

- Determine when to fragment and when to report an ICMPv6 "packet too big" error back to the source.

- **How to reflect ICMPv4 errors from routers along the tunnel path back to the source as ICMPv6 errors.**

Identifier: RQ_003_4011
RFC Clause: 3.2
Type: Mandatory
Applies to: Node

Requirement:
The encapsulator MUST NOT encapsulate IPv6 by using IPv4 as a link layer with a very large MTU (65535-20 bytes at most).

Specification Text:
Naively, the encapsulator could view encapsulation as IPv6 using IPv4 as a link layer with a very large MTU (65535-20 bytes at most; 20 bytes "extra" are needed for the encapsulating IPv4 header). The encapsulator would only need to report ICMPv6 "packet too big" errors back to the source for packets that exceed this MTU. However, such a scheme would be inefficient or non-interoperable for three reasons and therefore MUST NOT be used:

1) It would result in more fragmentation than needed. IPv4 layer fragmentation SHOULD be avoided due to the performance problems caused by the loss unit being smaller than the retransmission unit [KM97].

2) Any IPv4 fragmentation occurring inside the tunnel, i.e., between the encapsulator and the decapsulator, would have to be reassembled at the tunnel endpoint. For tunnels that terminate at a router, this would require additional memory and other resources to reassemble the IPv4 fragments into a complete IPv6 packet before that packet could be forwarded.

3) The encapsulator has no way of knowing that the decapsulator is able to defragment such IPv4 packets (see Section 3.6 for details), and has no way of knowing that the decapsulator is able to handle such a large IPv6 Maximum Receive Unit (MRU).

Identifier: RQ_003_4012
RFC Clause: 3.2
Type: Recommendation
Applies to: Node

Requirement:
IPv4 layer fragmentation SHOULD be avoided due to the performance problems caused by the loss unit being smaller than the retransmission unit.

Specification Text:
Naively, the encapsulator could view encapsulation as IPv6 using IPv4 as a link layer with a very large MTU (65535-20 bytes at most; 20 bytes "extra" are needed for the encapsulating IPv4 header). The encapsulator would only need to report ICMPv6 "packet too big" errors back to the source for packets that exceed this MTU. However, such a scheme would be inefficient or non-interoperable for three reasons and therefore MUST NOT be used:

1) It would result in more fragmentation than needed. IPv4 layer fragmentation SHOULD be avoided due to the performance problems caused by the loss unit being smaller than the retransmission unit [KM97].

2) Any IPv4 fragmentation occurring inside the tunnel, i.e., between the encapsulator and the decapsulator, would have to be reassembled at the tunnel endpoint. For tunnels that terminate at a router, this would require additional memory and other resources to reassemble the IPv4 fragments into a complete IPv6 packet before that packet could be forwarded.

3) The encapsulator has no way of knowing that the decapsulator is able to defragment such IPv4 packets (see Section 3.6 for details), and has no way of knowing that the decapsulator is able to handle such a large IPv6 Maximum Receive Unit (MRU).

Identifier: RQ_003_4013
RFC Clause: 3.2
Type: Mandatory
Applies to: Node

Requirement:
The encapsulator MUST use either:
- the fixed static MTU
- or OPTIONAL dynamic MTU determination based on the IPv4 path MTU to the tunnel endpoint.

Specification Text:

Hence, the encapsulator MUST NOT treat the tunnel as an interface with an MTU of 64 kilobytes, but instead either use the fixed static MTU or OPTIONAL dynamic MTU determination based on the IPv4 path MTU to the tunnel endpoint.

Identifier: RQ_003_4014
RFC Clause: 3.2
Type: Recommendation
Applies to: Node

Requirement:

If both the the fixed static MTU and OPTIONAL dynamic MTU mechanisms are implemented, the decision of which to use SHOULD be configurable on a per-tunnel endpoint basis.

Specification Text:

If both the mechanisms are implemented, the decision of which to use SHOULD be configurable on a per-tunnel endpoint basis.

Identifier: RQ_003_4015
RFC Clause: 3.2.1
Type: Mandatory
Applies to: Node

Requirement:

A node using static tunnel MTU treats the tunnel interface as having a fixed-interface MTU with a default of between 1280 and 1480 bytes (inclusive),

Specification Text:

A node using static tunnel MTU treats the tunnel interface as having a fixed-interface MTU. By default, the MTU MUST be between 1280 and 1480 bytes (inclusive), but it SHOULD be 1280 bytes. If the default is not 1280 bytes, the implementation MUST have a configuration knob that can be used to change the MTU value.

Identifier: RQ_003_4016
RFC Clause: 3.2.1
Type: Recommendation
Applies to: Node

Requirement:

A node using static tunnel MTU treats the tunnel interface as having a fixed-interface MTU, with a RECOMMENDED default value of 1280 bytes.

Specification Text:

A node using static tunnel MTU treats the tunnel interface as having a fixed-interface MTU. By default, the MTU MUST be between 1280 and 1480 bytes (inclusive), but it SHOULD be 1280 bytes. If the default is not 1280 bytes, the implementation MUST have a configuration knob that can be used to change the MTU value.

Identifier: RQ_003_4017
RFC Clause: 3.2.1
Type: Mandatory
Applies to: Node

Requirement:

If a node is using static tunnel MTU where the default is not 1280 bytes, the implementation MUST have a configuration knob that can be used to change the MTU value.

Specification Text:

A node using static tunnel MTU treats the tunnel interface as having a fixed-interface MTU. By default, the MTU MUST be between 1280 and 1480 bytes (inclusive), but it SHOULD be 1280 bytes. **If the default is not 1280 bytes, the implementation MUST have a configuration knob that can be used to change the MTU value.**

Identifier: RQ_003_4018
RFC Clause: 3.2.1
Type: Mandatory
Applies to: Node

Requirement:
A node, using static tunnel MTU , MUST be able to accept a fragmented IPv6 packet that, after reassembly, is as large as 1500 octets [RFC2460].

Specification Text:
A node MUST be able to accept a fragmented IPv6 packet that, after reassembly, is as large as 1500 octets [RFC2460]. This memo also includes requirements (see Section 3.6) for the amount of IPv4 reassembly and IPv6 MRU that MUST be supported by all the decapsulators. These ensure correct interoperability with any fixed MTUs between 1280 and 1480 bytes.

Identifier: RQ_003_4019
RFC Clause: 3.2.1
Type: Mandatory
Applies to: Node

Requirement:
A node, using static tunnel MTU, MUST NOT configure an MTU of more than 1480 bytes unless it has been administratively ensured that the decapsulator can reassemble or receive packets of that size.

Specification Text:
A larger fixed MTU than supported by these requirements MUST NOT be configured unless it has been administratively ensured that the decapsulator can reassemble or receive packets of that size.

Identifier: RQ_003_4020
RFC Clause: 3.2.1
Type: Mandatory
Applies to: Node

Requirement:
When using the static tunnel MTU, the Don't Fragment bit MUST NOT be set in the encapsulating IPv4 header.

Specification Text:
When using the static tunnel MTU, the Don't Fragment bit MUST NOT be set in the encapsulating IPv4 header. As a result, the encapsulator SHOULD NOT receive any ICMPv4 "packet too big" messages as a result of the packets it has encapsulated.

Identifier: RQ_003_4021
RFC Clause: 3.2.2
Type: Optional
Applies to: Node

Requirement:
The use of Dynamic MTU determination is OPTIONAL.

Specification Text:
The dynamic MTU determination is OPTIONAL. However, if it is implemented, it SHOULD have the behavior described in this document.

Identifier: RQ_003_4022
RFC Clause: 3.2.2
Type: Recommendation
Applies to: Node

Requirement:
If it is implemented, Dynamic MTU determination SHOULD have the behavior described in this document (RFC 4213).

Specification Text:
The dynamic MTU determination is OPTIONAL. **However, if it is implemented, it SHOULD have the behavior described in this document.**

Identifier: RQ_003_4023
RFC Clause: 3.2.2
Type: Recommendation
Applies to: Node

Requirement:
When using dynamic MTU determination, the encapsulator SHOULD employ the following algorithm to determine when to forward an IPv6 packet that is larger than the tunnel's path MTU using IPv4 fragmentation, and when to return an ICMPv6 "packet too big" message per [RFC1981]:

```
if (IPv4 path MTU - 20) is less than 1280
        if packet is larger than 1280 bytes
                Send ICMPv6 "packet too big" with MTU = 1280.
                Drop packet.
        else
                Encapsulate but do not set the Don't Fragment
                flag in the IPv4 header.  The resulting IPv4
                packet might be fragmented by the IPv4 layer
                on the encapsulator or by some router along
                the IPv4 path.
        endif
else
        if packet is larger than (IPv4 path MTU - 20)
                Send ICMPv6 "packet too big" with
                MTU = (IPv4 path MTU - 20).
                Drop packet.
        else
                Encapsulate and set the Don't Fragment flag
                in the IPv4 header.
        endif
endif
```

Specification Text:
The encapsulator SHOULD employ the following algorithm to determine when to forward an IPv6 packet that is larger than the tunnel's path MTU using IPv4 fragmentation, and when to return an ICMPv6 "packet too big" message per [RFC1981]:

```
if (IPv4 path MTU - 20) is less than 1280
        if packet is larger than 1280 bytes
                Send ICMPv6 "packet too big" with MTU = 1280.
                Drop packet.
        else
                Encapsulate but do not set the Don't Fragment
                flag in the IPv4 header.  The resulting IPv4
                packet might be fragmented by the IPv4 layer
                on the encapsulator or by some router along
                the IPv4 path.
        endif
else
        if packet is larger than (IPv4 path MTU - 20)
                Send ICMPv6 "packet too big" with
                MTU = (IPv4 path MTU - 20).
                Drop packet.
        else
                Encapsulate and set the Don't Fragment flag
                in the IPv4 header.
        endif
endif
```

Identifier: RQ_003_4024
RFC Clause: 3.2.2
Type: Optional
Applies to: Node

Requirement:
Encapsulators that have a large number of tunnels MAY choose between dynamic versus static tunnel MTUs on a per-tunnel endpoint basis.

Specification Text:
Encapsulators that have a large number of tunnels MAY choose between dynamic versus static tunnel MTUs on a per-tunnel endpoint basis. In cases where the number of tunnels that any one node is using is large, it is helpful to observe that this state information can be cached and discarded when not in use.

Identifier: RQ_003_4025
RFC Clause: 3.3
Type: Mandatory
Applies to: Node

Requirement:
The TTL of the encapsulating IPv4 header SHALL be selected in an implementation-dependent manner.

Specification Text:
The TTL of the encapsulating IPv4 header is selected in an implementation-dependent manner. The current suggested value is published in the "Assigned Numbers" RFC [RFC3232][ASSIGNED]. Implementations MAY provide a mechanism to allow the administrator to configure the IPv4 TTL as the IP Tunnel MIB [RFC4087].

Identifier: RQ_003_4026
RFC Clause: 3.3
Type: Recommendation
Applies to: Node

Requirement:
The current suggested value for the TTL of the encapsulating IPv4 header is published in the "Assigned Numbers" RFC [RFC3232][ASSIGNED].

Specification Text:
The TTL of the encapsulating IPv4 header is selected in an implementation-dependent manner. **The current suggested value is published in the "Assigned Numbers" RFC [RFC3232][ASSIGNED].** Implementations MAY provide a mechanism to allow the administrator to configure the IPv4 TTL as the IP Tunnel MIB [RFC4087].

Identifier: RQ_003_4027
RFC Clause: 3.3
Type: Optional
Applies to: Node

Requirement:
Implementations MAY provide a mechanism to allow the administrator to configure the IPv4 TTL as the IP Tunnel MIB [RFC4087].

Specification Text:
The TTL of the encapsulating IPv4 header is selected in an implementation-dependent manner. The current suggested value is published in the "Assigned Numbers" RFC [RFC3232][ASSIGNED]. **Implementations MAY provide a mechanism to allow the administrator to configure the IPv4 TTL as the IP Tunnel MIB [RFC4087].**

Identifier: RQ_003_4028
RFC Clause: 3.4
Type: Mandatory
Applies to: Node

Requirement:
ICMPv4 error handling is NOT applicable to static MTU tunnels.

Specification Text:
ICMPv4 error handling is only applicable to dynamic MTU determination, even though the functions could be used with static MTU tunnels as well.

Identifier: RQ_003_4029
RFC Clause: 3.4
Type: Mandatory
Applies to: Node

Requirement:
The recorded path MTU SHALL BE used by IPv6 to determine if an ICMPv6 "packet too big" error has to be generated.

Specification Text:
The ICMPv4 "packet too big" error messages are handled according to IPv4 Path MTU Discovery [RFC1191] and the resulting path MTU is recorded in the IPv4 layer. **The recorded path MTU is used by IPv6 to determine if an ICMPv6 "packet too big" error has to be generated as described in Section 3.2.2.**

Identifier: RQ_003_4030
RFC Clause: 3.4
Type: Optional
Applies to: Node

Requirement:
If sufficient data bytes from the offending packet are available, the encapsulator MAY extract the encapsulated IPv6 packet and use it to generate an ICMPv6 message directed back to the originating IPv6 node,

Specification Text:
If sufficient data bytes from the offending packet are available, the encapsulator MAY extract the encapsulated IPv6 packet and use it to generate an ICMPv6 message directed back to the originating IPv6 node, as shown below:

```
        +--------------+
        | IPv4 Header  |
        | dst = encaps |
        |       node   |
        +--------------+
        |    ICMPv4    |
        |    Header    |
 - -    +--------------+
        | IPv4 Header  |
        | src = encaps |
IPv4    |       node   |
        +--------------+   - -
Packet  |    IPv6      |
        |    Header    |   Original IPv6
 in     +--------------+   Packet -
        |   Transport  |   Can be used to
Error   |    Header    |   generate an
        +--------------+   ICMPv6
        |              |   error message
        ~     Data     ~   back to the source.
        |              |
 - -    +--------------+   - -
```

ICMPv4 Error Message Returned to Encapsulating Node

Identifier: RQ_003_4031
RFC Clause: 3.4
Type: Recommendation
Applies to: Node

Requirement:
When receiving ICMPv4 errors as above and the errors are not "packet too big", it would be useful to log the error as an error related to the tunnel.

Specification Text:
When receiving ICMPv4 errors as above and the errors are not "packet too big", it would be useful to log the error as an error related to the tunnel. Also, if sufficient headers are available, then the originating node MAY send an ICMPv6 error of type "unreachable" with code "address unreachable" to the IPv6 source. (The "address unreachable" code is appropriate since, from the perspective of IPv6, the tunnel is a link and that code is used for link-specific errors [RFC2463]).

Identifier: RQ_003_4032
RFC Clause: 3.5
Type: Mandatory
Applies to: Node

Requirement:
When encapsulating an IPv6 packet in an IPv4 datagram, the IPv4 header's version field SHALL be set to 4.

Specification Text:
When encapsulating an IPv6 packet in an IPv4 datagram, the IPv4 header fields are set as follows:

Version:

4

IP Header Length in 32-bit words:

5 (There are no IPv4 options in the encapsulating header.)

Type of Service:

0 unless otherwise specified. (See [RFC2983] and [RFC3168] Section 9.1 for issues relating to the Type-of-Service byte and tunneling.)

Total Length:

Payload length from IPv6 header plus length of IPv6 and IPv4 headers (i.e., IPv6 payload length plus a constant 60 bytes).

Identification:

Generated uniquely as for any IPv4 packet transmitted by the system.

Flags:

Set the Don't Fragment (DF) flag as specified in Section 3.2. Set the More Fragments (MF) bit as necessary if fragmenting.

Fragment Offset:

Set as necessary if fragmenting.

Time to Live:

Set in an implementation-specific manner, as described in
Section 3.3.

Protocol:

41 (Assigned payload type number for IPv6).

Header Checksum:

Calculate the checksum of the IPv4 header [RFC791].

Source Address:

An IPv4 address of the encapsulator: either configured by the administrator or an address of the outgoing interface.

Destination Address:

IPv4 address of the tunnel endpoint.

Identifier: RQ_003_4033
RFC Clause: 3.5
Type: Mandatory
Applies to: Node

Requirement:
When encapsulating an IPv6 packet in an IPv4 datagram, the IPv4 header's IP Header Length field SHALL be set to 5.

Specification Text:
When encapsulating an IPv6 packet in an IPv4 datagram, the IPv4 header fields are set as follows:

Version:

4

IP Header Length in 32-bit words:

5 (There are no IPv4 options in the encapsulating header.)

Type of Service:

0 unless otherwise specified. (See [RFC2983] and [RFC3168] Section 9.1 for issues relating to the Type-of-Service byte and tunneling.)

Total Length:

Payload length from IPv6 header plus length of IPv6 and IPv4 headers (i.e., IPv6 payload length plus a constant 60 bytes).

Identification:

Generated uniquely as for any IPv4 packet transmitted by the system.

Flags:

Set the Don't Fragment (DF) flag as specified in Section 3.2. Set the More Fragments (MF) bit as necessary if fragmenting.

Fragment Offset:

Set as necessary if fragmenting.

Time to Live:

Set in an implementation-specific manner, as described in
Section 3.3.

Protocol:

41 (Assigned payload type number for IPv6).

Header Checksum:

Calculate the checksum of the IPv4 header [RFC791].

Source Address:

An IPv4 address of the encapsulator: either configured by the administrator or an address of the outgoing interface.

Destination Address:

IPv4 address of the tunnel endpoint.

Identifier: RQ_003_4034
RFC Clause: 3.5
Type: Mandatory
Applies to: Node

Requirement:

When encapsulating an IPv6 packet in an IPv4 datagram, the IPv4 header's Type of Service field SHALL be set to 0 unless otherwise specified.

Specification Text:

```
 When encapsulating an IPv6 packet in an IPv4 datagram, the IPv4 header fields are set as follows:

    Version:

       4

    IP Header Length in 32-bit words:

       5 (There are no IPv4 options in the encapsulating header.)

    Type of Service:

       0 unless otherwise specified. (See [RFC2983] and [RFC3168] Section 9.1 for issues relating to
the Type-of-Service byte and tunneling.)

    Total Length:

       Payload length from IPv6 header plus length of IPv6 and IPv4 headers (i.e., IPv6 payload
length plus a constant 60 bytes).

    Identification:

       Generated uniquely as for any IPv4 packet transmitted by the system.

    Flags:

       Set the Don't Fragment (DF) flag as specified in Section 3.2. Set the More Fragments (MF) bit
as necessary if fragmenting.

    Fragment Offset:

       Set as necessary if fragmenting.

    Time to Live:

       Set in an implementation-specific manner, as described in
       Section 3.3.

    Protocol:

       41 (Assigned payload type number for IPv6).

    Header Checksum:

       Calculate the checksum of the IPv4 header [RFC791].

    Source Address:

       An IPv4 address of the encapsulator: either configured by the administrator or an address of
the outgoing interface.

    Destination Address:

       IPv4 address of the tunnel endpoint.
```

Identifier: RQ_003_4035
RFC Clause: 3.5
Type: Mandatory
Applies to: Node

Requirement:
When encapsulating an IPv6 packet in an IPv4 datagram, the IPv4 header's Total Length field SHALL be set to the Payload length from IPv6 header plus length of IPv6 and IPv4 headers (i.e., IPv6 payload length plus a constant 60 bytes).

Specification Text:
When encapsulating an IPv6 packet in an IPv4 datagram, the IPv4 header fields are set as follows:

```
    Version:

       4

    IP Header Length in 32-bit words:

       5 (There are no IPv4 options in the encapsulating header.)

    Type of Service:

       0 unless otherwise specified. (See [RFC2983] and [RFC3168] Section 9.1 for issues relating to
the Type-of-Service byte and tunneling.)
```

Total Length:

Payload length from IPv6 header plus length of IPv6 and IPv4 headers (i.e., IPv6 payload length plus a constant 60 bytes).

Identification:

```
       Generated uniquely as for any IPv4 packet transmitted by the system.

    Flags:

       Set the Don't Fragment (DF) flag as specified in Section 3.2. Set the More Fragments (MF) bit
as necessary if fragmenting.

    Fragment Offset:

       Set as necessary if fragmenting.

    Time to Live:

       Set in an implementation-specific manner, as described in
       Section 3.3.

    Protocol:

       41 (Assigned payload type number for IPv6).

    Header Checksum:

       Calculate the checksum of the IPv4 header [RFC791].

    Source Address:

       An IPv4 address of the encapsulator: either configured by the administrator or an address of
the outgoing interface.

    Destination Address:

       IPv4 address of the tunnel endpoint.
```

Identifier: RQ_003_4036
RFC Clause: 3.5
Type: Mandatory
Applies to: Node

Requirement:

When encapsulating an IPv6 packet in an IPv4 datagram, the IPv4 header's Identification field SHALL be generated uniquely as for any IPv4 packet transmitted by the system.

Specification Text:

When encapsulating an IPv6 packet in an IPv4 datagram, the IPv4 header fields are set as follows:

```
    Version:

       4

    IP Header Length in 32-bit words:

       5 (There are no IPv4 options in the encapsulating header.)

    Type of Service:

       0 unless otherwise specified. (See [RFC2983] and [RFC3168] Section 9.1 for issues relating to
the Type-of-Service byte and tunneling.)

    Total Length:

       Payload length from IPv6 header plus length of IPv6 and IPv4 headers (i.e., IPv6 payload
length plus a constant 60 bytes).

    Identification:

       Generated uniquely as for any IPv4 packet transmitted by the system.

    Flags:

       Set the Don't Fragment (DF) flag as specified in Section 3.2. Set the More Fragments (MF) bit
as necessary if fragmenting.

    Fragment Offset:

       Set as necessary if fragmenting.

    Time to Live:

       Set in an implementation-specific manner, as described in
       Section 3.3.

    Protocol:

       41 (Assigned payload type number for IPv6).

    Header Checksum:

       Calculate the checksum of the IPv4 header [RFC791].

    Source Address:

       An IPv4 address of the encapsulator: either configured by the administrator or an address of
the outgoing interface.

    Destination Address:

       IPv4 address of the tunnel endpoint.
```

Identifier: RQ_003_4037
RFC Clause: 3.5
Type: Mandatory
Applies to: Node

Requirement:
When encapsulating an IPv6 packet in an IPv4 datagram, the IPv4 header's Don't Fragment (DF) flag SHALL be SET.

Specification Text:
When encapsulating an IPv6 packet in an IPv4 datagram, the IPv4 header fields are set as follows:

Version:

4

IP Header Length in 32-bit words:

5 (There are no IPv4 options in the encapsulating header.)

Type of Service:

0 unless otherwise specified. (See [RFC2983] and [RFC3168] Section 9.1 for issues relating to the Type-of-Service byte and tunneling.)

Total Length:

Payload length from IPv6 header plus length of IPv6 and IPv4 headers (i.e., IPv6 payload length plus a constant 60 bytes).

Identification:

Generated uniquely as for any IPv4 packet transmitted by the system.

Flags:

Set the Don't Fragment (DF) flag as specified in Section 3.2. Set the More Fragments (MF) bit as necessary if fragmenting.

Fragment Offset:

Set as necessary if fragmenting.

Time to Live:

Set in an implementation-specific manner, as described in
Section 3.3.

Protocol:

41 (Assigned payload type number for IPv6).

Header Checksum:

Calculate the checksum of the IPv4 header [RFC791].

Source Address:

An IPv4 address of the encapsulator: either configured by the administrator or an address of the outgoing interface.

Destination Address:

IPv4 address of the tunnel endpoint.

Identifier: RQ_003_4038
RFC Clause: 3.5
Type: Mandatory
Applies to: Node

Requirement:

When encapsulating an IPv6 packet in an IPv4 datagram, the IPv4 header's More Fragments (MF) flag SHALL be SET as necessary if fragmenting.

Specification Text:

When encapsulating an IPv6 packet in an IPv4 datagram, the IPv4 header fields are set as follows:

```
    Version:

       4

    IP Header Length in 32-bit words:

       5 (There are no IPv4 options in the encapsulating header.)

    Type of Service:
```

0 unless otherwise specified. (See [RFC2983] and [RFC3168] Section 9.1 for issues relating to the Type-of-Service byte and tunneling.)

```
    Total Length:
```

Payload length from IPv6 header plus length of IPv6 and IPv4 headers (i.e., IPv6 payload length plus a constant 60 bytes).

```
    Identification:

       Generated uniquely as for any IPv4 packet transmitted by the system.

    Flags:
```

Set the Don't Fragment (DF) flag as specified in Section 3.2. **Set the More Fragments (MF) bit as necessary if fragmenting.**

```
    Fragment Offset:

       Set as necessary if fragmenting.

    Time to Live:

       Set in an implementation-specific manner, as described in
       Section 3.3.

    Protocol:

       41 (Assigned payload type number for IPv6).

    Header Checksum:

       Calculate the checksum of the IPv4 header [RFC791].

    Source Address:
```

An IPv4 address of the encapsulator: either configured by the administrator or an address of the outgoing interface.

```
    Destination Address:

       IPv4 address of the tunnel endpoint.
```

Identifier: RQ_003_4039
RFC Clause: 3.5
Type: Mandatory
Applies to: Node

Requirement:

When encapsulating an IPv6 packet in an IPv4 datagram, the IPv4 header's Fragment Offset field SHALL be SET as necessary if fragmenting.

Specification Text:

When encapsulating an IPv6 packet in an IPv4 datagram, the IPv4 header fields are set as follows:

```
    Version:

       4

    IP Header Length in 32-bit words:

       5 (There are no IPv4 options in the encapsulating header.)

    Type of Service:

       0 unless otherwise specified. (See [RFC2983] and [RFC3168] Section 9.1 for issues relating to
the Type-of-Service byte and tunneling.)

    Total Length:

       Payload length from IPv6 header plus length of IPv6 and IPv4 headers (i.e., IPv6 payload
length plus a constant 60 bytes).

    Identification:

       Generated uniquely as for any IPv4 packet transmitted by the system.

    Flags:

       Set the Don't Fragment (DF) flag as specified in Section 3.2. Set the More Fragments (MF) bit
as necessary if fragmenting.

    Fragment Offset:

       Set as necessary if fragmenting.

    Time to Live:

       Set in an implementation-specific manner, as described in
       Section 3.3.

    Protocol:

       41 (Assigned payload type number for IPv6).

    Header Checksum:

       Calculate the checksum of the IPv4 header [RFC791].

    Source Address:

       An IPv4 address of the encapsulator: either configured by the administrator or an address of
the outgoing interface.

    Destination Address:

       IPv4 address of the tunnel endpoint.
```

Identifier: RQ_003_4040
RFC Clause: 3.5
Type: Mandatory
Applies to: Node

Requirement:

When encapsulating an IPv6 packet in an IPv4 datagram, the IPv4 header's Time to Live field SHALL be SET in an implementation-specific manner.

Specification Text:

When encapsulating an IPv6 packet in an IPv4 datagram, the IPv4 header fields are set as follows:

Version:

4

IP Header Length in 32-bit words:

5 (There are no IPv4 options in the encapsulating header.)

Type of Service:

0 unless otherwise specified. (See [RFC2983] and [RFC3168] Section 9.1 for issues relating to the Type-of-Service byte and tunneling.)

Total Length:

Payload length from IPv6 header plus length of IPv6 and IPv4 headers (i.e., IPv6 payload length plus a constant 60 bytes).

Identification:

Generated uniquely as for any IPv4 packet transmitted by the system.

Flags:

Set the Don't Fragment (DF) flag as specified in Section 3.2. Set the More Fragments (MF) bit as necessary if fragmenting.

Fragment Offset:

Set as necessary if fragmenting.

Time to Live:

Set in an implementation-specific manner, as described in Section 3.3.

Protocol:

41 (Assigned payload type number for IPv6).

Header Checksum:

Calculate the checksum of the IPv4 header [RFC791].

Source Address:

An IPv4 address of the encapsulator: either configured by the administrator or an address of the outgoing interface.

Destination Address:

IPv4 address of the tunnel endpoint.

Identifier: RQ_003_4041
RFC Clause: 3.5
Type: Mandatory
Applies to: Node

Requirement:
When encapsulating an IPv6 packet in an IPv4 datagram, the IPv4 header's Protocol field SHALL be SET to 41.

Specification Text:
When encapsulating an IPv6 packet in an IPv4 datagram, the IPv4 header fields are set as follows:

```
    Version:

       4

    IP Header Length in 32-bit words:

       5 (There are no IPv4 options in the encapsulating header.)

    Type of Service:

       0 unless otherwise specified. (See [RFC2983] and [RFC3168] Section 9.1 for issues relating to
the Type-of-Service byte and tunneling.)

    Total Length:

       Payload length from IPv6 header plus length of IPv6 and IPv4 headers (i.e., IPv6 payload
length plus a constant 60 bytes).

    Identification:

       Generated uniquely as for any IPv4 packet transmitted by the system.

    Flags:

       Set the Don't Fragment (DF) flag as specified in Section 3.2. Set the More Fragments (MF) bit
as necessary if fragmenting.

    Fragment Offset:

       Set as necessary if fragmenting.

    Time to Live:

       Set in an implementation-specific manner, as described in
       Section 3.3.

    Protocol:

       41 (Assigned payload type number for IPv6).

    Header Checksum:

       Calculate the checksum of the IPv4 header [RFC791].

    Source Address:

       An IPv4 address of the encapsulator: either configured by the administrator or an address of
the outgoing interface.

    Destination Address:

       IPv4 address of the tunnel endpoint.
```

Identifier: RQ_003_4042
RFC Clause: 3.5
Type: Mandatory
Applies to: Node

Requirement:
When encapsulating an IPv6 packet in an IPv4 datagram, the IPv4 header's Header Checksum field SHALL be the checksum of the IPv4 header [RFC791].

Specification Text:
When encapsulating an IPv6 packet in an IPv4 datagram, the IPv4 header fields are set as follows:

```
    Version:

       4

    IP Header Length in 32-bit words:

       5 (There are no IPv4 options in the encapsulating header.)

    Type of Service:

       0 unless otherwise specified. (See [RFC2983] and [RFC3168] Section 9.1 for issues relating to
the Type-of-Service byte and tunneling.)

    Total Length:

       Payload length from IPv6 header plus length of IPv6 and IPv4 headers (i.e., IPv6 payload
length plus a constant 60 bytes).

    Identification:

       Generated uniquely as for any IPv4 packet transmitted by the system.

    Flags:

       Set the Don't Fragment (DF) flag as specified in Section 3.2. Set the More Fragments (MF) bit
as necessary if fragmenting.

    Fragment Offset:

       Set as necessary if fragmenting.

    Time to Live:

       Set in an implementation-specific manner, as described in
       Section 3.3.

    Protocol:

       41 (Assigned payload type number for IPv6).

    Header Checksum:

       Calculate the checksum of the IPv4 header [RFC791].

    Source Address:

       An IPv4 address of the encapsulator: either configured by the administrator or an address of
the outgoing interface.

    Destination Address:

       IPv4 address of the tunnel endpoint.
```

Identifier: RQ_003_4043
RFC Clause: 3.5
Type: Mandatory
Applies to: Node

Requirement:
When encapsulating an IPv6 packet in an IPv4 datagram, the IPv4 header's Header Source Address field SHALL be an IPv4 address of the encapsulator, either configured by the administrator or an address of the outgoing interface.

Specification Text:
When encapsulating an IPv6 packet in an IPv4 datagram, the IPv4 header fields are set as follows:

Version:

4

IP Header Length in 32-bit words:

5 (There are no IPv4 options in the encapsulating header.)

Type of Service:

0 unless otherwise specified. (See [RFC2983] and [RFC3168] Section 9.1 for issues relating to the Type-of-Service byte and tunneling.)

Total Length:

Payload length from IPv6 header plus length of IPv6 and IPv4 headers (i.e., IPv6 payload length plus a constant 60 bytes).

Identification:

Generated uniquely as for any IPv4 packet transmitted by the system.

Flags:

Set the Don't Fragment (DF) flag as specified in Section 3.2. Set the More Fragments (MF) bit as necessary if fragmenting.

Fragment Offset:

Set as necessary if fragmenting.

Time to Live:

Set in an implementation-specific manner, as described in
Section 3.3.

Protocol:

41 (Assigned payload type number for IPv6).

Header Checksum:

Calculate the checksum of the IPv4 header [RFC791].

Source Address:

An IPv4 address of the encapsulator: either configured by the administrator or an address of the outgoing interface.

Destination Address:

IPv4 address of the tunnel endpoint.

Identifier: RQ_003_4044
RFC Clause: 3.5
Type: Mandatory
Applies to: Node

Requirement:

When encapsulating an IPv6 packet in an IPv4 datagram, the IPv4 header's Header Destination Address field SHALL be an IPv4 address of the tunnel endpoint.

Specification Text:

When encapsulating an IPv6 packet in an IPv4 datagram, the IPv4 header fields are set as follows:

```
    Version:

       4

    IP Header Length in 32-bit words:

       5 (There are no IPv4 options in the encapsulating header.)

    Type of Service:

       0 unless otherwise specified. (See [RFC2983] and [RFC3168] Section 9.1 for issues relating to
the Type-of-Service byte and tunneling.)

    Total Length:

       Payload length from IPv6 header plus length of IPv6 and IPv4 headers (i.e., IPv6 payload
length plus a constant 60 bytes).

    Identification:

       Generated uniquely as for any IPv4 packet transmitted by the system.

    Flags:

       Set the Don't Fragment (DF) flag as specified in Section 3.2. Set the More Fragments (MF) bit
as necessary if fragmenting.

    Fragment Offset:

       Set as necessary if fragmenting.

    Time to Live:

       Set in an implementation-specific manner, as described in
       Section 3.3.

    Protocol:

       41 (Assigned payload type number for IPv6).

    Header Checksum:

       Calculate the checksum of the IPv4 header [RFC791].

    Source Address:

       An IPv4 address of the encapsulator: either configured by the administrator or an address of
the outgoing interface.

    Destination Address:

       IPv4 address of the tunnel endpoint.
```

Identifier: RQ_003_4045
RFC Clause: 3.5
Type: Mandatory
Applies to: Node

Requirement:
When encapsulating the packets, the node MUST ensure that it will use the correct source address so that the packets are acceptable to the decapsulator as described in Section 3.6.

Specification Text:
When encapsulating the packets, the node MUST ensure that it will use the correct source address so that the packets are acceptable to the decapsulator as described in Section 3.6. Configuring the source address is appropriate particularly in cases in which automatic selection of source address MAY produce different results in a certain period of time. This is often the case with multiple addresses, and multiple interfaces, or when routes MAY change frequently. Therefore, it SHOULD be possible to administratively specify the source address of a tunnel.

Identifier: RQ_003_4046
RFC Clause: 3.5
Type: Recommendation
Applies to: Node

Requirement:
When encapsulating the packets, the node SHOULD be able to administratively specify the source address of a tunnel.

Specification Text:
When encapsulating the packets, the node MUST ensure that it will use the correct source address so that the packets are acceptable to the decapsulator as described in Section 3.6. Configuring the source address is appropriate particularly in cases in which automatic selection of source address MAY produce different results in a certain period of time. This is often the case with multiple addresses, and multiple interfaces, or when routes MAY change frequently. **Therefore, it SHOULD be possible to administratively specify the source address of a tunnel.**

Identifier: RQ_003_4047
RFC Clause: 3.6
Type: Mandatory
Applies to: Node

Requirement:
When an IPv6/IPv4 host or a router receives an IPv4 datagram that is addressed to one of its own IPv4 addresses or a joined multicast group address, and the value of the protocol field is 41, the packet needs to be verified as belonging to one of the configured tunnel interfaces, reassembled (if fragmented at the IPv4 level), have the IPv4 header removed and the resulting IPv6 datagram be submitted to the IPv6 layer code on the node.

Specification Text:
When an IPv6/IPv4 host or a router receives an IPv4 datagram that is addressed to one of its own IPv4 addresses or a joined multicast group address, and the value of the protocol field is 41, the packet is potentially a tunnel packet and needs to be verified to belong to one of the configured tunnel interfaces (by checking source/destination addresses), reassembled (if fragmented at the IPv4 level), and have the IPv4 header removed and the resulting IPv6 datagram be submitted to the IPv6 layer code on the node.

Identifier: RQ_003_4050
RFC Clause: 3.6
Type: Optional
Applies to: Node

Requirement:
When packets for which the IPv4 source address does not match are discarded and if the implementation normally sends an ICMP message when receiving an unknown protocol packet, such an error message MAY be sent (e.g., ICMPv4 Protocol 41 Unreachable).

Specification Text:

The decapsulator MUST verify that the tunnel source address is correct before further processing packets, to mitigate the problems with address spoofing (see Section 4). This check also applies to packets that are delivered to transport protocols on the decapsulator. **This is done by verifying that the source address is the IPv4 address of the encapsulator, as configured on the decapsulator. Packets for which the IPv4 source address does not match MUST be discarded** and an ICMP message SHOULD NOT be generated; however, **if the implementation normally sends an ICMP message when receiving an unknown protocol packet, such an error message MAY be sent (e.g., ICMPv4 Protocol 41 Unreachable).**

Identifier: RQ_003_4063
RFC Clause: 3.6
Type: Mandatory
Applies to: Node

Requirement:

When decapsulating the packet, the IPv6 header SHALL NOT be modified.

Specification Text:

When decapsulating the packet, the IPv6 header is not modified. (However, see [RFC2983] and [RFC3168] section 9.1 for issues relating to the Type of Service byte and tunneling.) If the packet is subsequently forwarded, its hop limit is decremented by one.

Identifier: RQ_003_4064
RFC Clause: 3.6
Type: Mandatory
Applies to: Node

Requirement:

If the packet is subsequently forwarded, by the decapsulator, its hop limit is decremented by one.

Specification Text:

When decapsulating the packet, the IPv6 header is not modified. (However, see [RFC2983] and [RFC3168] section 9.1 for issues relating to the Type of Service byte and tunneling.) **If the packet is subsequently forwarded, its hop limit is decremented by one.**

Identifier: RQ_003_4065
RFC Clause: 3.6
Type: Mandatory
Applies to: Node

Requirement:

Once the encapsulating IPv4 header is discarded, the resulting packet is checked for validity when submitted to the IPv6 layer.

Specification Text:

The encapsulating IPv4 header is discarded, and the resulting packet is checked for validity when submitted to the IPv6 layer. When reconstructing the IPv6 packet, the length MUST be determined from the IPv6 payload length since the IPv4 packet might be padded (thus have a length that is larger than the IPv6 packet plus the IPv4 header being removed).

Identifier: RQ_003_4066
RFC Clause: 3.6
Type: Mandatory
Applies to: Node

Requirement:

When reconstructing the IPv6 packet, the length MUST be determined from the IPv6 payload length since the IPv4 packet might be padded (thus have a length that is larger than the IPv6 packet plus the IPv4 header being removed).

Specification Text:
The encapsulating IPv4 header is discarded, and the resulting packet is checked for validity when submitted to the IPv6 layer. **When reconstructing the IPv6 packet, the length MUST be determined from the IPv6 payload length since the IPv4 packet might be padded (thus have a length that is larger than the IPv6 packet plus the IPv4 header being removed).**

Identifier: RQ_003_4067
RFC Clause: 3.6
Type: Mandatory
Applies to: Node

Requirement:
After the decapsulation, the node MUST silently discard a packet with an invalid IPv6 source address.

Specification Text:
After the decapsulation, the node MUST silently discard a packet with an invalid IPv6 source address. The list of invalid source addresses SHOULD include at least:

- all multicast addresses (FF00::/8)

- the loopback address (::1)

- all the IPv4-compatible IPv6 addresses [RFC3513] (::/96), excluding the unspecified address for Duplicate Address Detection (::/128)

- all the IPv4-mapped IPv6 addresses (::ffff:0:0/96)

Identifier: RQ_003_4068
RFC Clause: 3.6
Type: Recommendation
Applies to: Node

Requirement:
The list of invalid source addresses SHOULD include at least:
- all multicast addresses (FF00::/8)
- the loopback address (::1)
- all the IPv4-compatible IPv6 addresses [RFC3513] (::/96), excluding the unspecified address for Duplicate Address Detection (::/128)
- all the IPv4-mapped IPv6 addresses (::ffff:0:0/96)

Specification Text:
After the decapsulation, the node MUST silently discard a packet with an invalid IPv6 source address. **The list of invalid source addresses SHOULD include at least:**

- **all multicast addresses (FF00::/8)**

- **the loopback address (::1)**

- **all the IPv4-compatible IPv6 addresses [RFC3513] (::/96), excluding the unspecified address for Duplicate Address Detection (::/128)**

- **all the IPv4-mapped IPv6 addresses (::ffff:0:0/96)**

Identifier: RQ_003_4069
RFC Clause: 3.6
Type: Recommendation
Applies to: Node

Requirement:
The node SHOULD be configured to perform ingress filtering [RFC2827][RFC3704] on the IPv6 source address, similar to on any of its interfaces.

Specification Text:
In addition, the node SHOULD be configured to perform ingress filtering [RFC2827][RFC3704] on the IPv6 source address, similar to on any of its interfaces, e.g.:

1) if the tunnel is toward the Internet, the node SHOULD be configured to check that the site's IPv6 prefixes are not used as the source addresses, or

2) if the tunnel is toward an edge network, the node SHOULD be configured to check that the source address belongs to that edge network.

Identifier:	RQ_003_4070
RFC Clause:	3.6
Type:	Recommendation
Applies to:	Node

Requirement:
It is RECOMMENDED that the implementations provide a single knob to make it easier to for the administrators to enable strict ingress filtering toward edge networks.

Specification Text:
It is RECOMMENDED that the implementations provide a single knob to make it easier to for the administrators to enable strict ingress filtering toward edge networks.

Identifier:	RQ_003_4048
RFC Clause:	3.6
Type:	Mandatory
Applies to:	Node

Requirement:
The decapsulator MUST verify that the source address is the IPv4 address of the encapsulator, as configured on the decapsulator. Packets for which the IPv4 source address does not match MUST be discarded.

Specification Text:
The decapsulator MUST verify that the tunnel source address is correct before further processing packets, to mitigate the problems with address spoofing (see Section 4). This check also applies to packets that are delivered to transport protocols on the decapsulator. This is done by verifying that the source address is the IPv4 address of the encapsulator, as configured on the decapsulator. Packets for which the IPv4 source address does not match MUST be discarded and an ICMP message SHOULD NOT be generated; however, if the implementation normally sends an ICMP message when receiving an unknown protocol packet, such an error message MAY be sent (e.g., ICMPv4 Protocol 41 Unreachable).

Identifier:	RQ_003_4049
RFC Clause:	3.6
Type:	Recommendation
Applies to:	Node

Requirement:
When packets for which the IPv4 source address does not match are discarded and unless the implementation normally sends an ICMP message when receiving an unknown protocol packet, an ICMP message SHOULD NOT be generated;

Specification Text:
The decapsulator MUST verify that the tunnel source address is correct before further processing packets, to mitigate the problems with address spoofing (see Section 4). This check also applies to packets that are delivered to transport protocols on the decapsulator. **This is done by verifying that the source address is the IPv4 address of the encapsulator, as configured on the decapsulator. Packets for which the IPv4 source address does not match MUST be discarded and an ICMP message SHOULD NOT be generated;** however, if the implementation normally sends an ICMP message when receiving an unknown protocol packet, such an error message MAY be sent (e.g., ICMPv4 Protocol 41 Unreachable).

Identifier:	RQ_003_4051
RFC Clause:	3.6
Type:	Optional
Applies to:	Node

Requirement:

Independent of any other forms of IPv4 ingress filtering the administrator of the node may have configured, the implementation MAY perform ingress filtering, i.e. check that the packet is arriving from the interface in the direction of the route toward the tunnel end-point.

Specification Text:

Independent of any other forms of IPv4 ingress filtering the administrator of the node MAY have configured, the implementation MAY perform ingress filtering, i.e., check that the packet is arriving from the interface in the direction of the route toward the tunnel end-point, similar to a Strict Reverse Path Forwarding (RPF) check [RFC3704]. As this MAY cause problems on tunnels that are routed through multiple links, it is RECOMMENDED that this check, if done, is disabled by default. The packets caught by this check SHOULD be discarded; an ICMP message SHOULD NOT be generated by default.

Identifier:	RQ_003_4052
RFC Clause:	3.6
Type:	Recommendation
Applies to:	Node

Requirement:

If the implementation performs ingress filtering. it is RECOMMENDED that this check, if done, is disabled by default.

Specification Text:

Independent of any other forms of IPv4 ingress filtering the administrator of the node MAY have configured, the implementation MAY perform ingress filtering, i.e., check that the packet is arriving from the interface in the direction of the route toward the tunnel end-point, similar to a Strict Reverse Path Forwarding (RPF) check [RFC3704]. As this MAY cause problems on tunnels that are routed through multiple links, **it is RECOMMENDED that this check, if done, is disabled by default.** The packets caught by this check SHOULD be discarded; an ICMP message SHOULD NOT be generated by default.

Identifier:	RQ_003_4053
RFC Clause:	3.6
Type:	Recommendation
Applies to:	Node

Requirement:

If the implementation performs ingress filtering. it is RECOMMENDED that The packets caught by this check SHOULD be discarded.

Specification Text:

Independent of any other forms of IPv4 ingress filtering the administrator of the node MAY have configured, the implementation MAY perform ingress filtering, i.e., check that the packet is arriving from the interface in the direction of the route toward the tunnel end-point, similar to a Strict Reverse Path Forwarding (RPF) check [RFC3704]. As this MAY cause problems on tunnels that are routed through multiple links, it is RECOMMENDED that this check, if done, is disabled by default. **The packets caught by this check SHOULD be discarded;** an ICMP message SHOULD NOT be generated by default.

Identifier:	RQ_003_4054
RFC Clause:	3.6
Type:	Recommendation
Applies to:	Node

Requirement:

If the implementation performs ingress filtering. it is RECOMMENDED that for packets caught by this check, an ICMP message SHOULD NOT be generated by default.

Specification Text:
Independent of any other forms of IPv4 ingress filtering the administrator of the node MAY have configured, the implementation MAY perform ingress filtering, i.e., check that the packet is arriving from the interface in the direction of the route toward the tunnel end-point, similar to a Strict Reverse Path Forwarding (RPF) check [RFC3704]. As this MAY cause problems on tunnels that are routed through multiple links, it is RECOMMENDED that this check, if done, is disabled by default. **The packets caught by this check** SHOULD be discarded; an **ICMP message SHOULD NOT be generated by default.**

Identifier: RQ_003_4055
RFC Clause: 3.6
Type: Mandatory
Applies to: Node

Requirement:
The decapsulator MUST be capable of having, on the tunnel interfaces, an IPv6 MRU of at least the maximum of 1500 bytes.

Specification Text:
The decapsulator MUST be capable of having, on the tunnel interfaces, an IPv6 MRU of at least the maximum of 1500 bytes and the largest (IPv6) interface MTU on the decapsulator.

Identifier: RQ_003_4056
RFC Clause: 3.6
Type: Mandatory
Applies to: Node

Requirement:
The decapsulator MUST be capable of having, on the tunnel interfaces, the largest (IPv6) interface MTU on the decapsulator.

Specification Text:
The decapsulator MUST be capable of having, on the tunnel interfaces, an IPv6 MRU of at least the maximum of 1500 bytes and **the largest (IPv6) interface MTU on the decapsulator.**

Identifier: RQ_003_4057
RFC Clause: 3.6
Type: Mandatory
Applies to: Node

Requirement:
The decapsulator MUST be capable of reassembling an IPv4 packet that is (after the reassembly) the maximum of 1500 bytes.

Specification Text:
The decapsulator MUST be capable of reassembling an IPv4 packet that is (after the reassembly) the maximum of 1500 bytes and the largest (IPv4) interface MTU on the decapsulator. The 1500-byte number is a result of encapsulators that use the static MTU scheme in Section 3.2.1, while encapsulators that use the dynamic scheme in Section 3.2.2 can cause up to the largest interface MTU on the decapsulator to be received. (Note that it is strictly the interface MTU on the last IPv4 router *before* the decapsulator that matters, but for most links the MTU is the same between all neighbors.)

Identifier: RQ_003_4058
RFC Clause: 3.6
Type: Mandatory
Applies to: Node

Requirement:
The decapsulator MUST be capable of reassembling the largest (IPv4) interface MTU on the decapsulator.

Specification Text:

The decapsulator MUST be capable of reassembling an IPv4 packet that is (after the reassembly) the maximum of 1500 bytes and **the largest (IPv4) interface MTU on the decapsulator**. The 1500-byte number is a result of encapsulators that use the static MTU scheme in Section 3.2.1, while encapsulators that use the dynamic scheme in Section 3.2.2 can cause up to the largest interface MTU on the decapsulator to be received. (Note that it is strictly the interface MTU on the last IPv4 router *before* the decapsulator that matters, but for most links the MTU is the same between all neighbors.)

Identifier:	RQ_003_4059
RFC Clause:	3.6
Type:	Optional
Applies to:	Node

Requirement:

An implementation MAY have a configuration knob that can be used to set a larger value of the tunnel reassembly buffers than an IPv4 packet that is (after the reassembly) the maximum of 1500 bytes and the largest (IPv4) interface MTU on the decapsulator.

Specification Text:

This reassembly limit allows dynamic tunnel MTU determination by the encapsulator to take advantage of larger IPv4 path MTUs. **An implementation MAY have a configuration knob that can be used to set a larger value of the tunnel reassembly buffers than the above number,** but it MUST NOT be set below the above number.

Identifier:	RQ_003_4060
RFC Clause:	3.6
Type:	Mandatory
Applies to:	Node

Requirement:

An implementation MUST NOT have a configuration knob that can be used to set a lower value of the tunnel reassembly buffers than an IPv4 packet that is (after the reassembly) the maximum of 1500 bytes and the largest (IPv4) interface MTU on the decapsulator.

Specification Text:

This reassembly limit allows dynamic tunnel MTU determination by the encapsulator to take advantage of larger IPv4 path MTUs. An implementation MAY have **a configuration knob that can be used to set** a larger value of **the tunnel reassembly buffers** than the above number, but it **MUST NOT be set below the above number.**

Identifier:	RQ_003_4061
RFC Clause:	3.6
Type:	Mandatory
Applies to:	Node

Requirement:

The decapsulation of an IPv6 datagram from IPv4 SHALL be acheived by removing the IPv4 header from the IPv6 datagram.

Specification Text:
The decapsulation is shown below:

```
+-------------+
|    IPv4     |
|   Header    |
+-------------+                 +-------------+
|    IPv6     |                 |    IPv6     |
|   Header    |                 |   Header    |
+-------------+                 +-------------+
|  Transport  |                 |  Transport  |
|   Layer     |      ===>       |   Layer     |
|   Header    |                 |   Header    |
+-------------+                 +-------------+
|             |                 |             |
~    Data     ~                 ~    Data     ~
|             |                 |             |
+-------------+                 +-------------+

         Decapsulating IPv6 from IPv4
```

Identifier: RQ_003_4062
RFC Clause: 3.6
Type: Mandatory
Applies to: Node

Requirement:
The decapsulator SHALL perform IPv4 reassembly before decapsulating the IPv6 packet.

Specification Text:
The decapsulator performs IPv4 reassembly before decapsulating the IPv6 packet.

Identifier: RQ_003_4071
RFC Clause: 3.7
Type: Mandatory
Applies to: Node

Requirement:
The configured tunnels are IPv6 interfaces (over the IPv4 "link layer") and thus MUST have link-local addresses.

Specification Text:
The configured tunnels are IPv6 interfaces (over the IPv4 "link layer") and thus MUST have link-local addresses. The link-local addresses are used by, e.g., routing protocols operating over the tunnels.

Identifier: RQ_003_4072
RFC Clause: 3.7
Type: Optional
Applies to: Node

Requirement:
The IPv6 interface identifier [RFC3513] for configured tunnels MAY be based on the 32-bit IPv4 address of an underlying interface, or formed using some other means, as long as it is different from the other tunnel endpoint with a reasonably high probability.

Specification Text:
The interface identifier [RFC3513] for such an interface MAY be based on the 32-bit IPv4 address of an underlying interface, or formed using some other means, as long as it is unique from the other tunnel endpoint with a reasonably high probability.

Identifier:	RQ_003_4073
RFC Clause:	3.7
Type:	Mandatory
Applies to:	Node

Requirement:

If an IPv4 address is used for forming the IPv6 link-local address, the interface identifier is the IPv4 address, prepended by zeros.

Specification Text:

If an IPv4 address is used for forming the IPv6 link-local address, the interface identifier is the IPv4 address, prepended by zeros. Note that the "Universal/Local" bit is zero, indicating that the interface identifier is not globally unique. The link-local address is formed by appending the interface identifier to the prefix FE80::/64.

Identifier:	RQ_003_4074
RFC Clause:	3.7
Type:	Mandatory
Applies to:	Node

Requirement:

If an IPv4 address is used for forming the IPv6 link-local address, the link-local address is formed by appending the interface identifier to the prefix FE80::/64.

Specification Text:

If an IPv4 address is used for forming the IPv6 link-local address, the interface identifier is the IPv4 address, prepended by zeros. Note that the "Universal/Local" bit is zero, indicating that the interface identifier is not globally unique. **The link-local address is formed by appending the interface identifier to the prefix FE80::/64.**

Identifier:	RQ_003_4075
RFC Clause:	3.8
Type:	Mandatory
Applies to:	Node

Requirement:

Configured tunnel implementations MUST at least accept and respond to the probe packets used by Neighbor Unreachability Detection (NUD) [RFC2461].

Specification Text:

Configured tunnel implementations MUST at least accept and respond to the probe packets used by Neighbor Unreachability Detection (NUD) [RFC2461]. The implementations SHOULD also send NUD probe packets to detect when the configured tunnel fails at which point the implementation can use an alternate path to reach the destination. Note that Neighbor Discovery allows that the sending of NUD probes be omitted for router-to-router links if the routing protocol tracks bidirectional reachability.

Identifier:	RQ_003_4076
RFC Clause:	3.8
Type:	Recommendation
Applies to:	Node

Requirement:

Configured tunnel implementations SHOULD send NUD probe packets to detect when the configured tunnel fails, at which point the implementation can use an alternate path to reach the destination.

Specification Text:

Configured tunnel implementations MUST at least accept and respond to the probe packets used by Neighbor Unreachability Detection (NUD) [RFC2461]. **The implementations SHOULD also send NUD probe packets to detect when the configured tunnel fails at which point the implementation can use an alternate path to reach the destination.** Note that Neighbor Discovery allows that the sending of NUD probes be omitted for router-to-router links if the routing protocol tracks bidirectional reachability.

Identifier: RQ_003_4077
RFC Clause: 3.8
Type: Recommendation
Applies to: Node

Requirement:
The sender of Neighbor Discovery packets SHOULD NOT include Source Link Layer Address options or Target Link Layer Address options on the tunnel link.

Specification Text:
For the purposes of Neighbor Discovery, the configured tunnels specified in this document are assumed to NOT have a link-layer address, even though the link-layer (IPv4) does have an address. This means that:

- the sender of Neighbor Discovery packets SHOULD NOT include Source Link Layer Address options or Target Link Layer Address options on the tunnel link.

- the receiver MUST, while otherwise processing the Neighbor Discovery packet, silently ignore the content of any Source Link Layer Address options or Target Link Layer Address options received on the tunnel link.

Identifier: RQ_003_4078
RFC Clause: 3.8
Type: Mandatory
Applies to: Node

Requirement:
The receiver MUST, while otherwise processing the Neighbor Discovery packet, silently ignore the content of any Source Link Layer Address options or Target Link Layer Address options received on the tunnel link.

Specification Text:
For the purposes of Neighbor Discovery, the configured tunnels specified in this document are assumed to NOT have a link-layer address, even though the link-layer (IPv4) does have an address. This means that:

- the sender of Neighbor Discovery packets SHOULD NOT include Source Link Layer Address options or Target Link Layer Address options on the tunnel link.

- the receiver MUST, while otherwise processing the Neighbor Discovery packet, silently ignore the content of any Source Link Layer Address options or Target Link Layer Address options received on the tunnel link.

Identifier: RQ_003_4079
RFC Clause: 5
Type: Recommendation
Applies to: Node

Requirement:
If the remainder threats of tunnel source verification are considered to be significant, a tunneling scheme with authentication SHOULD be used.

Specification Text:
If the remainder threats of tunnel source verification are considered to be significant, a tunneling scheme with authentication SHOULD be used instead, e.g., IPsec [RFC2401] (preferable) or Generic Routing Encapsulation with a pre-configured secret key [RFC2890]. As the configured tunnels are set up more or less manually, setting up the keying material is probably not a problem. However, setting up secure IPsec IPv6-in-IPv4 tunnels is described in another document [V64IPSEC].

Requirements extracted from RFC 4214

Identifier: RQ_003_2001
RFC Clause: 6.1
Type: Mandatory
Applies to: Node

Requirement:
ISATAP interface identifiers are constructed in Modified EUI-64format ([RFC3513], Section 2.5.1 and Appendix A) by concatenating the24-bit IANA OUI (00-00-5E), the 8-bit hexadecimal value 0xFE, and a32-bit IPv4 address in network byte order as follows:

Bit	Field Content
1 - 16	000000ug00000000
17 - 32	0101111011111110
33 - 64	Ipv4 address

Specification Text:
ISATAP interface identifiers are constructed in Modified EUI-64format ([RFC3513], Section 2.5.1 and Appendix A) by concatenating the24-bit IANA OUI (00-00-5E), the 8-bit hexadecimal value 0xFE, and a32-bit IPv4 address in network byte order as follows:

```
   |0              1|1              3|3                              6|
   |0              5|6              1|2                              3|
   +----------------+----------------+--------------------------------+
   |000000ug00000000|0101111011111110|mmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmm|
   +----------------+----------------+--------------------------------+
```

When the IPv4 address is known to be globally unique, the "u" bit(universal/local) is set to 1; otherwise, the "u" bit is set to 0."g" is the individual/group bit, **and "m" are the bits of the IPv4address.**

Identifier: RQ_003_2002
RFC Clause: 6.1
Type: Mandatory
Applies to: Node

Requirement:
In ISATAP interface identifiers, when the IPv4 address is known to be globally unique, the "u" bit(universal/local) is set to 1.

Specification Text:
ISATAP interface identifiers are constructed in Modified EUI-64format ([RFC3513], Section 2.5.1 and Appendix A) by concatenating the24-bit IANA OUI (00-00-5E), the 8-bit hexadecimal value 0xFE, and a32-bit IPv4 address in network byte order as follows:

```
   |0              1|1              3|3                              6|
   |0              5|6              1|2                              3|
   +----------------+----------------+--------------------------------+
   |000000ug00000000|0101111011111110|mmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmm|
   +----------------+----------------+--------------------------------+
```

When the IPv4 address is known to be globally unique, the "u" bit(universal/local) is set to 1; otherwise, the "u" bit is set to 0."g" is the individual/group bit, and "m" are the bits of the IPv4address.

Identifier: RQ_003_2003
RFC Clause: 6.1
Type: Mandatory
Applies to: Node

Requirement:
In ISATAP interface identifiers, when the IPv4 address is NOT known to be globally unique, the "u" bit(universal/local) is set to 0.

Specification Text:
ISATAP interface identifiers are constructed in Modified EUI-64format ([RFC3513], Section 2.5.1 and Appendix A) by concatenating the24-bit IANA OUI (00-00-5E), the 8-bit hexadecimal value 0xFE, and a32-bit IPv4 address in network byte order as follows:

```
   |0              1|1              3|3                              6|
   |0              5|6              1|2                              3|
   +----------------+----------------+--------------------------------+
   |000000ug00000000|0101111011111110|mmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmm|
   +----------------+----------------+--------------------------------+
```

When the IPv4 address is known to be globally unique, the "u" bit(universal/local) is set to 1; otherwise, the "u" bit is set to 0."g" is the individual/group bit, and "m" are the bits of the IPv4address.

Identifier: RQ_003_2004
RFC Clause: 6.1
Type: Mandatory
Applies to: Node

Requirement:
In ISATAP interface identifiers, the "g" is the individual/group bit, it SHALL be set to 0 if the address is a unicast address.

Specification Text:
ISATAP interface identifiers are constructed in Modified EUI-64format ([RFC3513], Section 2.5.1 and Appendix A) by concatenating the24-bit IANA OUI (00-00-5E), the 8-bit hexadecimal value 0xFE, and a32-bit IPv4 address in network byte order as follows:

```
   |0              1|1              3|3                              6|
   |0              5|6              1|2                              3|
   +----------------+----------------+--------------------------------+
   |000000ug00000000|0101111011111110|mmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmm|
   +----------------+----------------+--------------------------------+
```

When the IPv4 address is known to be globally unique, the "u" bit(universal/local) is set to 1; otherwise, the "u" bit is set to 0.**"g" is the individual/group bit,** and "m" are the bits of the IPv4address.

Identifier: RQ_003_2005
RFC Clause: 6.1
Type: Mandatory
Applies to: Node

Requirement:
In ISATAP interface identifiers, the "g" is the individual/group bit, it SHALL be set to 1 if the address is a multicast address.

Specification Text:
ISATAP interface identifiers are constructed in Modified EUI-64format ([RFC3513], Section 2.5.1 and Appendix A) by concatenating the24-bit IANA OUI (00-00-5E), the 8-bit hexadecimal value 0xFE, and a32-bit IPv4 address in network byte order as follows:

```
   |0              1|1              3|3                              6|
   |0              5|6              1|2                              3|
   +----------------+----------------+--------------------------------+
   |000000ug00000000|0101111011111110|mmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmm|
   +----------------+----------------+--------------------------------+
```

When the IPv4 address is known to be globally unique, the "u" bit(universal/local) is set to 1; otherwise, the "u" bit is set to 0.**"g" is the individual/group bit**, and "m" are the bits of the IPv4address.

Identifier: RQ_003_2006
RFC Clause: 6.2
Type: Mandatory
Applies to: Node

Requirement:
An ISATAP interface's locator set MUST NOT span multiple sites.

Specification Text:
Each ISATAP interface configures a set of locators consisting of IPv4address-to-interface mappings from a single site; i.e., an ISATAP interface's locator set MUST NOT span multiple sites.

Identifier: RQ_003_2007
RFC Clause: 6.2
Type: Recommendation
Applies to: Node

Requirement:
When an IPv4 address is removed from an interface, the corresponding locator SHOULD be removed from its associated locator set(s).

Specification Text:
When an IPv4 address is removed from an interface, the corresponding locator SHOULD be removed from its associated locator set(s). When anew IPv4 address is assigned to an interface, the corresponding locator MAY be added to the appropriate locator set(s).

Identifier: RQ_003_2008
RFC Clause: 6.2
Type: Optional
Applies to: Node

Requirement:
When anew IPv4 address is assigned to an interface, the corresponding locator MAY be added to the appropriate locator set(s).

Specification Text:
When an IPv4 address is removed from an interface, the corresponding locator SHOULD be removed from its associated locator set(s). **When anew IPv4 address is assigned to an interface, the corresponding locator MAY be added to the appropriate locator set(s).**

Identifier: RQ_003_2009
RFC Clause: 6.2
Type: Mandatory
Applies to: Node

Requirement:
ISATAP interfaces form ISATAP interface identifiers from IPv4addresses in their locator set and SHALL use them to create link-local ISATAP addresses ([RFC2462], Section 5.3).

Specification Text:
ISATAP interfaces form ISATAP interface identifiers from IPv4addresses in their locator set and use them to create link-local ISATAP addresses ([RFC2462], Section 5.3).

Identifier: RQ_003_2010
RFC Clause: 6.3
Type: Mandatory
Applies to: Node

Requirement:
ISATAP MUST assume that its underlying IPv4 carrier network only has unicast capability.

Specification Text:
It is not possible to assume the general availability of wide-areaIPv4 multicast, so (unlike 6over4 [RFC2529]) **ISATAP MUST assume that its underlying IPv4 carrier network only has unicast capability.** Support for IPv6 multicast over ISATAP interfaces is not described in this document.

Identifier: RQ_003_2011
RFC Clause: 7.1
Type: Mandatory
Applies to: Node

Requirement:
ISATAP addresses shall be mapped to a link-layer address by a static computation; i.e., the last four octets are treated as an IPv4address.

Specification Text:
ISATAP addresses are mapped to a link-layer address by a static computation; i.e., the last four octets are treated as an IPv4address.

Identifier: RQ_003_2012
RFC Clause: 7.2
Type: Recommendation
Applies to: Node

Requirement:
ISATAP interfaces SHOULD process ARP failures and persistent ICMPv4errors as link-specific information indicating that a path to a neighbor MAY have failed ([RFC2461], Section 7.3.3).

Specification Text:
ISATAP interfaces SHOULD process ARP failures and persistent ICMPv4errors as link-specific information indicating that a path to a neighbor MAY have failed ([RFC2461], Section 7.3.3).

Identifier: RQ_003_2013
RFC Clause: 7.2
Type: Mandatory
Applies to: Node

Requirement:
When an ISATAP node receives an IPv4 protocol 41 datagram that does not belong to a configured tunnel interface, it SHALL determine whether the packet's IPv4 destination address and arrival interface match a locator configured in an ISATAP interface's locator set.

Specification Text:
The specification in ([MECH], Section 3.6) is used. Additionally, when an ISATAP node receives an IPv4 protocol 41 datagram that doesnot belong to a configured tunnel interface, it determines whether the packet's IPv4 destination address and arrival interface match a locator configured in an ISATAP interface's locator set.

Identifier: RQ_003_2014
RFC Clause: 7.3
Type: Mandatory
Applies to: Node

Requirement:
If an ISATAP interface that configures a matching locator is found, then the decapsulator MUST verify that the packet's IPv4 source address is correct for the encapsulated IPv6 source address. The IPv4 source address is correct if:

- the IPv6 source address is an ISATAP address that embeds the IPv4 source address in its interface identifier, or

- the IPv4 source address is a member of the Potential Router List (see Section 8.1).

Specification Text:
If an ISATAP interface that configures a matching locator is found, the decapsulator MUST verify that the packet's IPv4 source address is correct for the encapsulated IPv6 source address. The IPv4 source address is correct if:

- **the IPv6 source address is an ISATAP address that embeds the IPv4 source address in its interface identifier, or**

- **the IPv4 source address is a member of the Potential Router List (see Section 8.1).**

Identifier: RQ_003_2015
RFC Clause: 7.3
Type: Mandatory
Applies to: Node

Requirement:
If an ISATAP interface that configures a matching locator is found, then the decapsulator MUST verify that the packet's IPv4 source address is correct for the encapsulated IPv6 source address. Packets for which the IPv4 source address is incorrect for this ISATAP interface SHALL also be checked to determine whether they belong to another tunnel interface.

Specification Text:
Packets for which the IPv4 source address is incorrect for this ISATAP interface are checked to determine whether they belong to another tunnel interface.

Identifier: RQ_003_2016
RFC Clause: 8.1
Type: Mandatory
Applies to: Host

Requirement:
To the list of Conceptual Data Structures (specified in RFC2461, Section 5.1), ISATAP interfaces add a Potential Router List (PRL).

Specification Text:
To the list of Conceptual Data Structures ([RFC2461], Section 5.1),ISATAP interfaces add the following:

> **Potential Router List (PRL)**
> A set of entries about potential routers; used to support router and prefix discovery. Each entry ("PRL(i)") has an associated timer ("TIMER(i)"), and an IPv4 address ("V4ADDR(i)") that represents a router's advertising ISATAP interface.

Identifier: RQ_003_2017
RFC Clause: 8.1
Type: Mandatory
Applies to: Host

Requirement:
The Potential Router List (PRL) SHALL comprise a set of entries about potential routers; used to support router and prefix discovery. Each entry ("PRL(i)") has an associated timer ("TIMER(i)"), and an IPv4 address ("V4ADDR(i)") that represents a router's advertising ISATAP interface.

Specification Text:
To the list of Conceptual Data Structures ([RFC2461], Section 5.1),ISATAP interfaces add the following:

Potential Router List (PRL)
A set of entries about potential routers; used to support router and prefix discovery. Each entry ("PRL(i)") has an associated timer ("TIMER(i)"), and an IPv4 address ("V4ADDR(i)") that represents a router's advertising ISATAP interface.

Identifier: RQ_003_2018
RFC Clause: 8.2
Type: Mandatory
Applies to: Router

Requirement:
Advertising ISATAP interfaces send Solicited Router Advertisement messages as specified in ([RFC2461], Section 6.2.6) except that the messages SHALL be sent directly to the soliciting node and not multicast to the all-nodes group, which is the usual case.

Specification Text:
Advertising ISATAP interfaces send Solicited Router Advertisement messages as specified in ([RFC2461], Section 6.2.6) except that the messages are sent directly to the soliciting node; i.e., they might not be received by other nodes on the link.

History

Document history		
V1.1.1	September 2007	Publication