IPv6 over Fibre Channel

Status of this Memo

This document is an Internet Draft and is in full conformance with all provisions of Section 10 of RFC 2026.

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

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

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

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

Abstract

Fibre Channel (FC) is a high speed serial interface technology that supports several Upper Layer Protocols including Small Computer System Interface (SCSI) and IPv4, as specified in [IPFC].
The purpose of this document is to specify a way of encapsulating IP version 6 [IPv6] over Fibre Channel and to describe a method of forming IPv6 link-local addresses [AARCH] and statelessly autoconfigured addresses on Fibre Channel networks. This document also describes the content of the Source/Target Link-layer Address option used in Neighbor Discovery [DISC] when the messages are transmitted on a Fibre Channel network.

Conventions used in this document

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

Table Of Contents

Warning to readers familiar with Fibre Channel: both Fibre Channel and IETF standards use the same byte transmission order. However, the bit numbering is different. See Appendix C for guidance.

1. Introduction to Fibre Channel

1.1 Overview

Fibre Channel (FC) is a gigabit speed network technology primarily used for Storage Networking. Fibre Channel is standardized in the T11 Technical Committee of the InterNational Committee for Information Technology Standards (INCITS), an American National Standard Institute (ANSI) accredited standards committee.

Fibre Channel devices are called Nodes. Each Node has one or more Ports that connect to Ports of other devices. Fibre Channel may be implemented using any combination of the following three topologies:
- a point-to-point link between two Ports;
- a set of Ports interconnected by a switching network called a Fabric, as defined in [FC-FS];
- a set of Ports interconnected with a loop topology, as defined in [FC-AL-2].

A Node Port is more precisely called an N_Port. A Node Port that is capable of operating in a loop topology using the loop specific protocols is designated as an NL_Port. The term Nx_Port is used to generically indicate these two kinds of Node Port.

A Fabric Port is more precisely called an F_Port. A Fabric Port that is capable of operating in a loop topology using the loop specific protocols is designated as an FL_Port. The term Fx_Port is used to generically indicate these two kinds of Fabric Port.

From an IPv6 point of view, a Fibre Channel network, built with any combination of the FC topologies described above, is an IPv6 Link [IPv6] connecting any IPv6-capable Nx_Port acting as an IPv6 Interface.

1.2 Identifiers and Login

Fibre Channel entities are identified by permanent 64 bit long Name_Identifiers. [FC-FS] defines several formats of Name_Identifiers. The value of the first four bits defines the format of a Name_Identifier. These names are referred to in a more precise manner as follows:
- an Nx_Port's Name_Identifier is called N_Port_Name;
- an Fx_Port's Name_Identifier is called F_Port_Name;
- a Node's Name_Identifier is called Node_Name;
- a Fabric's Name_Identifier is called Fabric_Name.

An Nx_Port connected to a Fibre Channel network is associated with two identifiers, its permanent N_Port_Name and a volatile 24 bit

address called N_Port_ID. The N_Port_Name is used to identify the Nx_Port, while the N_Port_ID is used for communications among Nx_Ports.

Each Nx_Port acquires an N_Port_ID from the Fabric by performing a process called Fabric Login or FLOGI. The FLOGI process is used also to negotiate several communications parameters between the Nx_Port and the Fabric, such as the receive data field size, which determines the maximum size of the Fibre Channel frames that may be transferred between the Nx_Port and the Fabric.

Before effective communication may take place between two Nx_Ports, they must complete a process called Port Login or PLOGI. The PLOGI process provides each Nx_Port with the other Nx_Port's N_Port_Name, and negotiates several communication parameters, such as the receive data field size, which determines the maximum size of the Fibre Channel frames that may be transferred between the two Nx_Ports.

Both Fabric Login and Port Login may be explicit, i.e. performed using specific FC control messages (called Extended Link Services or ELS), or implicit, in which the parameters are specified by configuration or other methods.

1.3 FC Levels and Frame Format

[FC-FS] describes the Fibre Channel protocol using 5 different levels. The FC-2 and FC-4 levels are relevant for this specification. The FC-2 level defines the FC frame format, the transport services, and control functions necessary for information transfer. The FC-4 level supports Upper Level Protocols, such as IPv4, IPv6 or SCSI. The Fibre Channel frame format is depicted in figure 1.

```
+-----+-----------+-----------+--------//-------+-----+-----+
|     |           |         Data Field          |     |     |
| SOF | FC Header |<--------------------------->| CRC | EOF |
|     |           | Optional  | Frame           |     |     |
|     |           | Header(s) | Payload         |     |     |
+-----+-----------+-----------+--------//-------+-----+-----+
```

Fig. 1: Fibre Channel Frame Format

The Start of Frame (SOF) and End of Frame (EOF) are special FC transmission words that act as frame delimiters. The CRC is 4-octets long and uses the same 32-bit polynomial used in FDDI.

The FC Header is 24-octets long and contains several fields associated with the identification and control of the Data Field.

The Data Field is of variable size, ranging from 0 to 2112 octets, and includes the user data in the Frame Payload field, and Optional Headers. The currently defined Optional Headers are:
- ESP_Header;
- Network_Header;
- Association_Header;
- Device_Header.

The value of the SOF field determines the FC Class of service associated with the frame. Five Classes of service are specified in [FC-FS]. They are distinguished primarily by the method of flow control between the communicating Nx_Ports and by the level of data integrity provided. A given Fabric or Nx_Port may support one or more of the following Classes of service:
- Class 1: Dedicated physical connection with delivery confirmation;
- Class 2: Frame multiplexed service with delivery confirmation;
- Class 3: Datagram service;
- Class 4: Fractional bandwidth;
- Class 6: Reliable multicast via dedicated connections.

1.4 Sequences and Exchanges

An application level payload such as IPv6 is called Information Unit at the FC-4 level of Fibre Channel. Each FC-4 Information Unit is mapped to an FC Sequence by the FC-2 level. An FC Sequence consists of one or more FC frames related by the value of the Sequence_ID (SEQ_ID) field of the FC Header.

The maximum data that may be carried by an FC frame is 2112 octets. The maximum usable frame size depends on the Fabric and Nx_Port implementations and is negotiated during the Login process. Whenever an Information Unit to be transmitted exceeds this value, the FC-2 level segments it into multiple FC frames, sent as a single Sequence. The receiving Nx_Port reassembles the Sequence of frames and delivers a reassembled Information Unit to the FC-4 level. The Sequence Count (SEQ_CNT) field of the FC Header may be used to ensure frame ordering.

Multiple Sequences may be related together as belonging to the same FC Exchange. The Exchange is a mechanism used by two Nx_Ports to identify and manage an operation between them. The Exchange is opened when the operation is started between the two Nx_Ports, and closed when the operation ends. FC frames belonging to the same Exchange are related by the value of the Exchange_ID fields in the FC Header. An Originator Exchange_ID (OX_ID) and a Responder Exchange_ID (RX_ID) uniquely identify the Exchange.

2. IPv6 Capable Nx_Ports

This specification requires an IPv6 capable Nx_Port to have the following properties:

- The format of its N_Port_Name MUST be one of 0x1, 0x2, 0x5, 0xC, 0xD, 0xE, 0xF. Other Name_Identifier formats are not acceptable to support IPv6;
- It MUST support Class 3;
- It MUST support continuously increasing SEQ_CNT [FC-FS];
- It MUST be able to transmit and receive an FC-4 Information Unit at least 1304 octets long;
- It SHOULD support a receive data field size for Device_Data FC frames of at least 1024 octets.

3. IPv6 Encapsulation

3.1 FC Sequence Format

An IPv6 packet is an Information Unit at the FC-4 level of Fibre Channel, and is mapped to an FC Sequence by the FC-2 level. An FC Sequence containing an IPv6 packet MUST carry the FC Network_Header [FC-FS] and the LLC/SNAP header [IEEE-LLC], resulting to the FC Sequence format depicted in figure 2.

```
+---------------+---------------+---------------+---------------+
|                                                               |
+-                                                             -+
|                        Network_Header                         |
+-                        (16 octets)                          -+
|                                                               |
+-                                                             -+
|                                                               |
+---------------+---------------+---------------+---------------+
|                        LLC/SNAP header                        |
+-                         (8 octets)                          -+
|                                                               |
+---------------+---------------+---------------+---------------+
|                                                               |
+-                                                             -+
/                         IPv6 Packet                           /
/                                                               /
+-                                                             -+
|                                                               |
+---------------+---------------+---------------+---------------+
```

Fig. 2: FC Sequence Carrying an IPv6 Packet

The FC ESP_Header [FC-FS] MAY be used to secure the FC frames composing the FC Sequence. [AH] or [ESP] may be used to provide security at the IPv6 layer. Other types of FC Optional Header MUST NOT be used in an IPv6 FC Sequence.

Typically, a Sequence consists of more than one frame. Only the first frame of the Sequence MUST include the FC Network_Header and the LLC/SNAP header. The other frames MUST NOT include them, as depicted in figure 3.

```
                   First Frame of an IPv6 FC Sequence
+-----------+------------------+-----------------+-------//--------+
| FC Header | FC Network_Header | LLC/SNAP header | First chunk of  |
|           |                  |                 | the IPv6 Packet |
+-----------+------------------+-----------------+-------//--------+

               Subsequent Frames of an IPv6 FC Sequence
          +-----------+----------------//------------------+
          | FC Header | Additional chunk of the IPv6 Packet |
          +-----------+---------------//-------------------+
```

Fig. 3: Optional Headers in an IPv6 FC Sequence

3.2 Classes of Service

This specification uses FC Class 3. IPv6 packets carrying Neighbor Discovery [DISC] messages MUST be encapsulated in Class 3 FC frames. Other IPv6 packets SHOULD use Class 3 as well. The use of other Classes of service is outside the scope of this specification.

3.3 FC Header Code Points

The fields of the Fibre Channel Header are depicted in figure 4. To encapsulate IPv6 over Fibre Channel the following code points MUST be used:

- R_CTL: 0x04 (Device_Data frame with Unsolicited Data Information Category [FC-FS])
- TYPE: 0x05 (IP over Fibre Channel)
- CS_CTL/Prio: 0x0
- DF_CTL: 0x20 (Network_Header) for the first FC frame of an IPv6 Sequence, 0x00 for the following FC frames. If the FC ESP_Header is used, then 0x60 for the first FC frame of an IPv6 Sequence, 0x40 for the following FC frames.
- F_CTL, SEQ_ID, SEQ_CNT, OX_ID, RX_ID: see section 9, and [FC-FS] for additional requirements.

```
     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
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |     R_CTL     |                     D_ID                      |
    +---------------+---------------+---------------+---------------+
    |  CS_CTL/Prio  |                     S_ID                      |
    +---------------+---------------+---------------+---------------+
    |     TYPE      |                    F_CTL                      |
    +---------------+---------------+---------------+---------------+
    |    SEQ_ID     |    DF_CTL     |            SEQ_CNT            |
    +---------------+---------------+---------------+---------------+
    |             OX_ID             |             RX_ID             |
    +---------------+---------------+---------------+---------------+
    |                           Parameter                           |
    +---------------+---------------+---------------+---------------+
```

Fig. 4: FC Header Format

3.4 FC Network_Header

The fields of the FC Network_Header are depicted in figure 5. For use with IPv6 the N_Port_Names formats MUST be one of 0x1, 0x2, 0x5, 0xC, 0xD, 0xE, 0xF. Other Name_Identifier formats are not acceptable to support IPv6.

```
     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
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    +-                  Destination N_Port_Name                    -+
    |                                                               |
    +---------------------------------------------------------------+
    |                                                               |
    +-                    Source N_Port_Name                       -+
    |                                                               |
    +---------------------------------------------------------------+
```

Fig. 5: FC Network_Header Format

3.5 LLC/SNAP Header

The fields of the LLC/SNAP Header [IEEE-LLC] are depicted in figure 6. To encapsulate IPv6 over Fibre Channel the following code points MUST be used:
- DSAP: 0xAA
- SSAP: 0xAA
- CTRL: 0x03

- OUI: 0x00-00-00
- PID: 0x86-DD

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     DSAP      |     SSAP      |     CTRL      |      OUI      |
+---------------+---------------+---------------+---------------+
|             OUI               |              PID              |
+---------------+---------------+---------------+---------------+
```

Fig. 6: LLC/SNAP Header Format

3.6 Bit and Byte Ordering

IPv6 packets are mapped to the FC-4 level using the big-endian byte ordering, that corresponds to the standard network byte order or canonical form.

4. Maximum Transfer Unit

The default MTU size for IPv6 [IPv6] packets over Fibre Channel is 65280 octets. This size may be reduced by a Router Advertisement [DISC] containing an MTU option that specifies a smaller MTU, or by manual configuration of each Nx_Port. However, as required by [IPv6], the MTU MUST NOT be lower than 1280 octets. If a Router Advertisement received on an Nx_Port has an MTU option specifying an MTU larger than 65280, or larger than a manually configured value, that MTU option MAY be logged to system management but MUST be otherwise ignored.

5. Stateless Autoconfiguration

5.1 IPv6 Interface Identifier and Address Prefix

The IPv6 Interface ID [AARCH] for an Nx_Port is based on the EUI-64 address [EUI64] derived from the Nx_Port's N_Port_Name. The IPv6 Interface Identifier is obtained by complementing the Universal/Local bit of the OUI field of the derived EUI-64 address.

[FC-FS] specifies a method to map format 0x1 (IEEE 48 bit address), or 0x2 (IEEE Extended), or 0x5 (IEEE Registered) FC Name_Identifiers in EUI-64 addresses. This allows the usage of these Name_Identifiers to support IPv6. [FC-FS] also defines EUI-64 mapped FC Name_Identifiers (formats 0xC, 0xD, 0xE, and 0xF), that are derived from an EUI-64 address. It is possible to reverse this address

mapping to obtain the original EUI-64 address in order to support IPv6.

An IPv6 Address Prefix used for stateless autoconfiguration [ACONF] of an Nx_Port MUST have a length of 64 bits.

5.2 Generating an Interface ID from a Format 1 N_Port_Name

The Name_Identifier format 0x1 is depicted in figure 7.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|0 0 0 1|         0x000         |              OUI              |
+-------+-------+---------------+---------------+---------------+
|      OUI      |                     VSID                      |
+---------------+---------------+---------------+---------------+
```

Fig. 7: Format 0x1 Name_Identifier

The EUI-64 address derived from this Name_Identifier has the format depicted in figure 8 [FC-FS].

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         OUI with complemented U/L bit         |0 0 0 1|  VSID |
+---------------+---------------+-------+-------+-------+-------+
|                     VSID              |         0x000         |
+---------------+---------------+-------+-------+---------------+
```

Fig. 8: EUI-64 Address from a Format 0x1 Name_Identifier

The IPv6 Interface Identifier is obtained from this EUI-64 address by complementing the U/L bit in the OUI field. So the OUI in the IPv6 Interface ID is exactly as in the FC Name_Identifier. The resulting IPv6 Interface Identifier has local scope [AARCH] and the format depicted in figure 9.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      OUI                      |0 0 0 1|  VSID |
+---------------+---------------+-------+-------+-------+-------+
|                     VSID              |         0x000         |
+---------------+---------------+-------+-------+---------------+
```

Fig. 9: IPv6 Interface ID from a Format 0x1 Name_Identifier

As an example, the FC Name_Identifier 0x10-00-34-63-46-AB-CD-EF generates the IPv6 Interface Identifier 3463:461A:BCDE:F000.

5.3 Generating an Interface ID from a Format 2 N_Port_Name

The Name_Identifier format 0x2 is depicted in figure 10.

```
  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
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |0 0 1 0|    Vendor Specific    |              OUI              |
 +-------+-------+---------------+---------------+---------------+
 |      OUI      |                     VSID                      |
 +---------------+---------------+---------------+---------------+
```

Fig. 10: Format 0x2 Name_Identifier

The EUI-64 address derived from this Name_Identifier has the format depicted in figure 11 [FC-FS].

```
  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
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |         OUI with complemented U/L bit         |0 0 1 0|  VSID |
 +---------------+-----------------------+-------+-------+-------+
 |                   VSID                |    Vendor Specific    |
 +---------------+-----------------------+-------+---------------+
```

Fig. 11: EUI-64 Address from a Format 0x2 Name_Identifier

The IPv6 Interface Identifier is obtained from this EUI-64 address by complementing the U/L bit in the OUI field. So the OUI in the IPv6 Interface ID is exactly as in the FC Name_Identifier. The resulting IPv6 Interface Identifier has local scope [AARCH] and the format depicted in figure 12.

```
  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
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                      OUI                      |0 0 1 0|  VSID |
 +---------------+-----------------------+-------+-------+-------+
 |                   VSID                |    Vendor Specific    |
 +---------------+-----------------------+-------+---------------+
```

Fig. 12: IPv6 Interface ID from a Format 0x2 Name_Identifier

As an example, the FC Name_Identifier 0x27-89-34-63-46-AB-CD-EF generates the IPv6 Interface Identifier 3463:462A:BCDE:F789.

5.4 Generating an Interface ID from a Format 5 N_Port_Name

The Name_Identifier format 0x5 is depicted in figure 13.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|0 1 0 1|                     OUI                       |  VSID |
+-------+-------+---------------+---------------+-------+-------+
|                              VSID                             |
+---------------+---------------+---------------+---------------+
```

Fig. 13: Format 0x5 Name_Identifier

The EUI-64 address derived from this Name_Identifier has the format depicted in figure 14 [FC-FS].

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         OUI with complemented U/L bit         |0 1 0 1|  VSID |
+---------------+---------------+---------------+-------+-------+
|                              VSID                             |
+---------------+---------------+---------------+---------------+
```

Fig. 14: EUI-64 Address from a Format 0x5 Name_Identifier

The IPv6 Interface Identifier is obtained from this EUI-64 address complementing the U/L bit in the OUI field. So the OUI in the IPv6 Interface ID is exactly as in the FC Name_Identifier. The resulting IPv6 Interface Identifier has local scope [AARCH] and the format depicted in figure 15.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      OUI                      |0 1 0 1|  VSID |
+---------------+---------------+---------------+-------+-------+
|                              VSID                             |
+---------------+---------------+---------------+---------------+
```

Fig. 15: IPv6 Interface ID from a Format 0x5 Name_Identifier

As an example, the FC Name_Identifier 0x53-46-34-6A-BC-DE-F7-89 generates the IPv6 Interface Identifier 3463:465A:BCDE:F789.

5.5 Generating an Interface ID from a EUI-64 mapped N_Port_Name

The EUI-64 mapped Name_Identifiers formats (formats 0xC through 0xF) are derived from a EUI-64 address by compressing the OUI field of such addresses. The compression is performed by removing from the OUI the Universal/Local and Individual/Group bits, and by putting bits 0 to 5 of the OUI in the first octet of the Name_Identifier, and bits 8 to 23 of the OUI in the second and third octet of the Name_Identifier, as shown in figure 16.

```
  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
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |1 1| OUI[0..5] |           OUI[8..23]          |      VSID     |
 +---+-----------+---------------+---------------+---------------+
 |                              VSID                             |
 +---------------+---------------+---------------+---------------+
```

Fig. 16: EUI-64 Mapped Name_Identifiers Format

The EUI-64 address used to generate the Name_Identifier shown in figure 16 has the format depicted in figure 17.

```
  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
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 | OUI[0..5] |0 0|           OUI[8..23]          |      VSID     |
 +-----------+---+---------------+---------------+---------------+
 |                              VSID                             |
 +---------------+---------------+---------------+---------------+
```

Fig. 17: EUI-64 Address from a EUI-64 Mapped Name_Identifier

The IPv6 Interface Identifier is obtained from this EUI-64 address by complementing the U/L bit in the OUI field. The resulting IPv6 Interface Identifier has global scope [AARCH] and the format depicted in figure 18.

```
  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
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 | OUI[0..5] |1 0|           OUI[8..23]          |      VSID     |
 +-----------+---+---------------+---------------+---------------+
 |                              VSID                             |
 +---------------+---------------+---------------+---------------+
```

Fig. 18: IPv6 Interface ID from a EUI-64 Mapped Name_Identifier

As an example, the FC Name_Identifier 0xCD-63-46-AB-01-25-78-9A generates the IPv6 Interface Identifier 3663:46AB:0125:789A.

6. Link-Local Addresses

The IPv6 link-local address [AARCH] for an Nx_Port is formed by appending the Interface Identifier, as defined in section 5, to the prefix FE80::/64. The resulting address is depicted in figure 19.

```
  10 bits            54 bits                  64 bits
+----------+-----------------------+---------------------------+
|1111111010|         (zeros)       |    Interface Identifier   |
+----------+-----------------------+---------------------------+
```

Fig. 19: IPv6 link-local Address Format

7. Address Mapping for Unicast

An Nx_Port has two kinds of Fibre Channel addresses:
- a non-volatile 64-bit address, called N_Port_Name;
- a volatile 24-bit address, called N_Port_ID.

The N_Port_Name is used to uniquely identify the Nx_Port, while the N_Port_ID is used to route frames to the Nx_Port. Both FC addresses are required to resolve an IPv6 unicast address. The fact that the N_Port_ID is volatile implies that the mapping between N_Port_Name and N_Port_ID MUST be valid before use. Appendix B discusses the validation process.

The procedure for mapping IPv6 unicast addresses into Fibre Channel link-layer addresses uses the Neighbor Discovery Protocol [DISC]. The Source/Target Link-layer Address option has the format depicted in figure 20 when the link layer is Fibre Channel.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |  Length = 2   |           Reserved            |
+---------------+---------------+---------------+---------------+
|                                                               |
+-                         N_Port_Name                         -+
|                                                               |
+---------------+---------------+---------------+---------------+
|                    N_Port_ID                  |   Reserved    |
+---------------+---------------+---------------+---------------+
```

Fig. 20: Source/Target Link-layer Address option for Fibre Channel

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

Length: 2 (in units of 8 octets).

N_Port_Name: This field contains the Nx_Port's N_Port_Name.
N_Port_ID: This field contains the Nx_Port's N_Port_ID.

8. Address Mapping for Multicast

By default, all best-effort IPv6 multicast packets MUST be mapped to FC Sequences addressed to the broadcast N_Port_ID 0xFF-FF-FF. In particular, datagrams addressed to all-nodes multicast address, all-routers multicast address, and solicited-node multicast addresses [AARCH] MUST be sent as Class 3 FC Sequences addressed to the broadcast N_Port_ID 0xFF-FF-FF. In this case, the Destination N_Port_Name field of the FC Network_Header MUST be set to the value 0x10-00-FF-FF-FF-FF-FF-FF. Appendix A specifies how to transmit a Class 3 broadcast FC Sequence over various Fibre Channel topologies.

An Nx_Port supporting IPv6 MUST be able to map a received broadcast Class 3 Device_Data FC frame to an implicit Port Login context in order to handle IPv6 multicast packets. The receive data field size of this implicit Port Login MUST be the same across all the Nx_Ports connected to the same Fabric, otherwise FC broadcast transmission does not work. In order to reduce the need for FC Sequence segmentation, the receive data field size of this implicit Port Login SHOULD be 1024 octets. This receive data field size requirement applies to broadcast Device_Data FC frames, not to ELSs.

Receiving an FC Sequence carrying an IPv6 multicast packet MAY trigger some additional processing by the Nx_Port if that IPv6 packet requires a reply. In this case, if a valid Port Login to the Nx_Port that sent the IPv6 multicast packet does not exist, the Nx_Port MUST perform such a Port Login, and then use it for the unicast IPv6 reply. In the case of Neighbor Discovery messages [DISC], the N_Port_ID to which the Port Login is directed is taken from the N_Port_ID field of the Source/Target Link-layer Address option.

As an example, an Nx_Port processes a received broadcast FC Sequence carrying an IPv6 multicast unsolicited router advertisement [DISC] simply by passing the carried IPv6 packet to the IPv6 layer. Instead, if a received broadcast FC Sequence carries an IPv6 multicast solicitation message [DISC] requiring a reply, and no valid Port Login exists with the Nx_Port sender of the multicast packet, then a Port Login MUST be performed in order to send the unicast reply message.

Best-effort IPv6 multicast for other multicast group addresses MAY use Fibre Channel Multicast Groups [FC-FS], if supported by the particular FC topology and implementation.

9. Sequence and Exchange Management

9.1 Sequence Management

FC Sequences are REQUIRED to be non-streamed. In order to avoid missing FC frame aliasing by Sequence_ID reuse, an Nx_Port supporting IPv6 is REQUIRED to use continuously increasing SEQ_CNT [FC-FS]. Each Exchange MUST start with SEQ_CNT = 0 in the first frame, and every frame transmitted after that MUST increment the previous SEQ_CNT by one. Any frames received from the other N_Port in the Exchange shall have no effect on the transmitted SEQ_CNT.

9.2 Exchange Management

To transfer IPv6 packets, each Nx_Port MUST have a dedicated Exchange for sending data to each Nx_Port in the network and a dedicated Exchange for receiving data from each Nx_Port.

An Exchange Responder is not required to assign RX_IDs. If a RX_ID of 0xFFFF is assigned, the Exchange Responder is identifying Exchanges based on S_ID / D_ID / OX_ID only.

When an Exchange is created between two Nx_Ports for unicast IPv6 packets, it remains active while the Nx_Ports are logged in with each other. Each FC broadcast and ELS [FC-FS] SHOULD use a separate short lived Exchange.

For IPv6, Exchanges MUST NOT transfer Sequence Initiative, because they are used in a unidirectional mode. The Sequence Initiative bit in the F_CTL field of the FC Header [FC-FS] MUST be set to 0.

The mechanism for aging or expiring exchanges based on activity, timeout, or other methods is outside the scope of this document.

The Exchange Originator MAY terminate Exchanges by setting the F_CTL LS bit [FC-FS]. Exchanges MAY be torn down by the Exchange Originator or Exchange Responder by using the ABTS protocol. IPv6 Exchanges SHOULD NOT be terminated by Logout, since this may terminate active Exchanges on other FC-4s [FC-FS].

10. Security Considerations

IPv6 does not introduce any additional security concerns beyond those that already exist within the Fibre Channel protocols. Zoning techniques based on FC Name Server masking (soft zoning) do not work with IPv6, because IPv6 over Fibre Channel does not use the FC Name Server. All the techniques defined to secure IPv6 traffic may be used in a Fibre Channel Environment.

11. IANA Considerations

None.

12. Acknowledgment

The author would like to acknowledge the authors of [IPFC], [ETHER], and [IPv6-1394], since some part of this document has been derived from them, as well as the ANSI INCITS T11.3 Task Group members who reviewed this document.

13. Normative References

[FC-FS] ANSI INCITS 373-2003, "Fibre Channel - Framing and Signaling (FC-FS)", Rev 1.90, April 2003.

[FC-AL-2] ANSI INCITS 332-1999, "Fibre Channel - Arbitrated Loop-2 (FC-AL-2)".

[IPv6] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

[AARCH] Hinden, R. and S. Deering, "Internet Protocol Version 6 (IPv6) Addressing Architecture", RFC 3513, April 2003.

[ACONF] Thomson, S. and T. Narten, "IPv6 Stateless Address Autoconfiguration", RFC 2462, December 1998.

[DISC] Narten, T., Nordmark, E., and W. Simpson, "Neighbor Discovery for IP Version 6 (IPv6)", RFC 2461, December 1998.

[IEEE-LLC] IEEE Standards for Local Area Networks: Logical Link Control", IEEE, New York, 1985.

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

14. Informative References

[IPFC] Rajagopal, M., Bhagwat, R., and W. Rickard, "IP and ARP over Fibre Channel", RFC 2625, June 1999.

[AH] Kent, S. and R. Atkinson, "IP Authentication Header", RFC 2402, November 1998.

[ESP] Kent, S. and R. Atkinson, "IP Encapsulating Security Payload (ESP)", RFC 2406, November 1998.

[EUI64] "Guidelines For 64-bit Global Identifier (EUI-64)", http://standards.ieee.org/db/oui/tutorials/EUI64.html

[ETHER] Crawford, M., "Transmission of IPv6 Packets over Ethernet Networks", RFC 2464, December 1998.

[IPv6-1394] Fujisawa, K. and A. Onoe, "Transmission of IPv6 Packets over IEEE 1394 Networks", RFC 3146, October 2001.

15. Author's Address

Claudio DeSanti
Andiamo Systems, Inc.
375 E. Tasman Dr.
San Jose, CA 95134
USA

Phone: +1 408 853-9172
EMail: cds@andiamo.com

A. Transmission of a Broadcast FC Sequence over FC Topologies

A.1 Point-to-Point Topology

No particular mechanisms are required for this case. The Nx_Port connected at the other side of the cable receives the broadcast FC Sequence having D_ID 0xFFFFFF.

A.2 Private Loop Topology

An NL_Port attached to a private loop MUST transmit a Class 3 broadcast FC Sequence by using the OPN(fr) primitive signal [FC-AL-2].

a) The source NL_Port first sends an Open Broadcast Replicate (OPN(fr)) primitive signal, forcing all the NL_Ports in the loop (except itself) to replicate the frames that they receive while examining the FC Header's D_ID field.
b) The source NL_Port then removes the OPN(fr) signal when it returns to it.
c) The source NL_Port then sends the Class 3 broadcast FC Sequence having D_ID 0xFFFFFF.

A.3 Public Loop Topology

An NL_Port attached to a public loop MUST NOT use the OPN(fr) primitive signal. Rather, it MUST send the Class 3 broadcast FC Sequence having D_ID 0xFFFFFF to the FL_Port at AL_PA = 0x00 [FC-AL-2].

The Fabric propagates the broadcast to all other FC_Ports [FC-FS], including the FL_Port which the broadcast arrived on. This includes all F_Ports, and other FL_Ports.

Each FL_Port propagates the broadcast by using the primitive signal OPN(fr), in order to prepare the loop to receive the broadcast sequence.

A.4 Fabric Topology

An N_Port connected to an F_Port MUST transmit the Class 3 broadcast FC Sequence having D_ID 0xFFFFFF to the F_Port. The Fabric propagates the broadcast to all other FC_Ports [FC-FS].

B. Validation of the <N_Port_Name, N_Port_ID> mapping

B.1 Overview

At all times, the <N_Port_Name, N_Port_ID> mapping must be valid before use.

After an FC link interruption occurs, the N_Port_ID of an Nx_Port may change, as well as the N_Port_IDs of all other Nx_Ports that have previously performed Port Login with this Nx_Port. Because of this, address validation is required after a LIP in a loop topology [FC-AL-2] or after NOS/OLS in a point-to-point topology [FC-FS].

N_Port_IDs do not change as a result of Link Reset (LR) [FC-FS], thus address validation is not required in this case.

B.2 FC Layer Address Validation in a Point-to-Point Topology

No validation is required after LR. In a point-to-point topology, NOS/OLS causes implicit Logout of each N_Port and after a NOS/OLS each N_Port must again perform a Port Login [FC-FS].

B.3 FC Layer Address Validation in a Private Loop Topology

After a LIP [FC-AL-2], an NL_Port must not transmit any data to another NL_Port until the address of the other port has been validated. The validation consists of completing either ADISC or PDISC [FC-FS].

For a requester, this specification prohibits PDISC and requires ADISC. As a responder, an implementation may need to respond to both ADISC and PDISC for compatibility with other FC specifications.

If the three FC addresses (N_Port_ID, N_Port_Name, Node_Name) of a logged remote NL_Port exactly match the values prior to the LIP, then any active Exchange with that NL_Port may continue.

If any of the three FC addresses has changed, then the remote NL_Port must be logged out.

If an NL_Port's N_Port_ID changes after a LIP, then all active logged in NL_Ports must be logged out.

B.4 FC Layer Address Validation in a Public Loop Topology

A FAN ELS may be sent by the Fabric to all known previously logged in NL_Ports following an initialization event. Therefore, after a LIP [FC-AL-2], NL_Ports may wait for this notification to arrive, or they may perform an FLOGI.

If the F_Port_Name and Fabric_Name contained in the FAN ELS or FLOGI response exactly match the values before the LIP and if the AL_PA [FC-AL-2] obtained by the NL_Port is the same as the one before the LIP, then the port may resume all Exchanges. If not, then FLOGI must be performed with the Fabric and all logged in Nx_Ports must be logged out.

A public loop NL_Port must perform the private loop validation as specified in section B.3 to any NL_Port on the local loop that has an N_Port_ID of the form 0x00-00-XX.

B.5 FC Layer Address Validation in a Fabric Topology

No validation is required after LR (link reset).

After NOS/OLS, an N_Port must perform FLOGI. If, after FLOGI, the N_Port's N_Port_ID, the F_Port_Name, and the Fabric_Name are the same as before the NOS/OLS, then the N_Port may resume all Exchanges. If not, all logged in Nx_Ports must be logged out [FC-FS].

C. Fibre Channel Bit and Byte Numbering Guidance

Both Fibre Channel and IETF standards use the same byte transmission order. However, the bit numbering is different.

Fibre Channel bit numbering can be observed if the data structure heading shown in figure 21 is cut and pasted at the top of the figures present in this document.

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

Fig. 21: Fibre Channel Bit Numbering