Network Working Group
Request for Comments: 3821
Category: Standards Track

M. Rajagopal
E. Rodriguez
R. Weber

July 2004

Fibre Channel Over TCP/IP (FCIP)

Status of this Memo

Copyright Notice

Abstract

Fibre Channel Over TCP/IP (FCIP) describes mechanisms that allow the interconnection of islands of Fibre Channel storage area networks over IP-based networks to form a unified storage area network in a single Fibre Channel fabric. FCIP relies on IP-based network services to provide the connectivity between the storage area network islands over local area networks, metropolitan area networks, or wide area networks.

Table Of Contents

1. Purpose, Motivation, and Objectives

Warning to Readers Familiar With Fibre Channel: Both Fibre Channel and IETF standards use the same byte transmission order. However, the bit and byte numbering is different. See appendix A for guidance.

Fibre Channel (FC) is a gigabit or multi-gigabit speed networking technology primarily used to implement Storage Area Networks (SANs). See section 2 for information about how Fibre Channel is standardized and the relationship of this specification to Fibre Channel standards. An overview of Fibre Channel can be found in [34].

This specification describes mechanisms that allow the interconnection of islands of Fibre Channel SANs over IP Networks to form a unified SAN in a single Fibre Channel fabric. The motivation behind defining these interconnection mechanisms is a desire to connect physically remote FC sites allowing remote disk access, tape backup, and live mirroring.

Fibre Channel standards have chosen nominal distances between switch elements that are less than the distances available in an IP Network. Since Fibre Channel and IP Networking technologies are compatible, it is logical to turn to IP Networking for extending the allowable distances between Fibre Channel switch elements.

The fundamental assumption made in this specification is that the Fibre Channel traffic is carried over the IP Network in such a manner that the Fibre Channel Fabric and all Fibre Channel devices on the Fabric are unaware of the presence of the IP Network. This means that the FC datagrams must be delivered in such time as to comply with existing Fibre Channel specifications. The FC traffic may span LANs, MANs, and WANs, so long as this fundamental assumption is adhered to.

The objectives of this document are to:

1) specify the encapsulation and mapping of Fibre Channel (FC) frames employing FC Frame Encapsulation [19].

2) apply the mechanism described in 1) to an FC Fabric using an IP network as an interconnect between two or more islands in an FC Fabric.

3) address any FC concerns arising from tunneling FC traffic over an IP-based network, including security, data integrity (loss), congestion, and performance. This will be accomplished by utilizing the existing IETF-specified suite of protocols.

4) be compatible with the referenced FC standards. While new work may be undertaken in T11 to optimize and enhance FC Fabrics, this specification REQUIRES conformance only to the referenced FC standards.

5) be compatible with all applicable IETF standards so that the IP Network used to extend an FC Fabric can be used concurrently for other reasonable purposes.

The objectives of this document do not include using an IP Network as a replacement for the Fibre Channel Arbitrated Loop interconnect. No definition is provided for encapsulating loop primitive signals for transmission over an IP 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 BCP 14, RFC 2119 [1].

2. Relationship to Fibre Channel Standards

2.1. Relevant Fibre Channel Standards

FC is standardized as a family of American National Standards developed by the T11 technical committee of INCITS (InterNational Committee for Information Technology Standards). T11 has specified a number of documents describing FC protocols, operations, and services. T11 documents of interest to readers of this specification include (but are not limited to):

- FC-BB - Fibre Channel Backbone [2]
- FC-BB-2 - Fibre Channel Backbone -2 [3]
- FC-SW-2 - Fibre Channel Switch Fabric -2 [4]
- FC-FS - Fibre Channel Framing and Signaling [5]

FC-BB and FC-BB-2 describe the relationship between an FC Fabric and interconnect technologies not defined by Fibre Channel standards (e.g., ATM and SONET). FC-BB-2 is the Fibre Channel document describing the relationships between FC and TCP/IP, including the FC use of FCIP.

FC-SW-2 describes the switch components of an FC Fabric and FC-FS describes the FC Frame format and basic control features of Fibre Channel.

Additional information regarding T11 activities is available on the committee’s web site www.t11.org.

2.2. This Specification and Fibre Channel Standards

When considering the challenge of transporting FC Frames over an IP Network, it is logical to divide the standardization effort between TCP/IP requirements and Fibre Channel requirements. This specification covers the TCP/IP requirements for transporting FC Frames; the Fibre Channel documents described in section 2.1 cover the Fibre Channel requirements.

This specification addresses only the requirements necessary to properly utilize an IP Network as a conduit for FC Frames. The result is a specification for an FCIP Entity (see section 5.4).

A product that tunnels an FC Fabric through an IP Network MUST combine the FCIP Entity with an FC Entity (see section 5.3) using an implementation specific interface. The requirements placed on an FC Entity by this specification to achieve proper delivery of FC Frames are summarized in appendix H. More information about FC Entities can be found in the Fibre Channel standards and an example of an FC Entity can be found in FC-BB-2 [3].

No attempt is being made to define a specific API between an FCIP Entity and an FC Entity. The approach is to specify required functional interactions between an FCIP Entity and an FC Entity (both of which are required to forward FC frames across an IP Network), but allow implementers to choose how these interactions will be realized.

3. Terminology

Terms used to describe FCIP concepts are defined in this section.

FC End Node - An FC device that uses the connection services provided by the FC Fabric.

FC Entity - The Fibre Channel specific functional component that combines with an FCIP Entity to form an interface between an FC Fabric and an IP Network (see section 5.3).

FC Fabric - An entity that interconnects various Nx_Ports (see [5]) attached to it, and is capable of routing FC Frames using only the destination ID information in an FC Frame header (see appendix F).

FC Fabric Entity - A Fibre Channel specific element containing one or more Interconnect_Ports (see FC-SW-2 [4]) and one or more FC/FCIP Entity pairs. See FC-BB-2 [3] for details about FC Fabric Entities.

FC Frame - The basic unit of Fibre Channel data transfer (see appendix F).

FC Frame Receiver Portal - The access point through which an FC Frame and time stamp enter an FCIP Data Engine from the FC Entity.

FC Frame Transmitter Portal - The access point through which a reconstituted FC Frame and time stamp leave an FCIP Data Engine to the FC Entity.

FC/FCIP Entity pair - The combination of one FC Entity and one FCIP entity.

FCIP Data Engine (FCIP_DE) - The component of an FCIP Entity that handles FC Frame encapsulation, de-encapsulation, and transmission FCIP Frames through a single TCP Connection (see section 5.6).

FCIP Entity - The entity responsible for the FCIP protocol exchanges on the IP Network and encompasses FCIP_LEP(s) and FCIP Control and Services module (see section 5.4).

FCIP Frame - An FC Frame plus the FC Frame Encapsulation [19] header, encoded SOF and encoded EOF that contains the FC Frame (see section 5.6.1).

FCIP Link - One or more TCP Connections that connect one FCIP_LEP to another (see section 5.2).

FCIP Link Endpoint (FCIP_LEP) - The component of an FCIP Entity that handles a single FCIP Link and contains one or more FCIP_DEs (see section 5.5).

Encapsulated Frame Receiver Portal - The TCP access point through which an FCIP Frame is received from the IP Network by an FCIP Data Engine.

Encapsulated Frame Transmitter Portal - The TCP access point through which an FCIP Frame is transmitted to the IP Network by an FCIP Data Engine.

FCIP Special Frame (FSF) - A specially formatted FC Frame containing information used by the FCIP protocol (see section 7).

4. Protocol Summary

The FCIP protocol is summarized as follows:

1) The primary function of an FCIP Entity is forwarding FC Frames, employing FC Frame Encapsulation described in [19].

2) Viewed from the IP Network perspective, FCIP Entities are peers and communicate using TCP/IP. Each FCIP Entity contains one or more TCP endpoints in the IP-based network.

3) Viewed from the FC Fabric perspective, pairs of FCIP Entities, in combination with their associated FC Entities, forward FC Frames between FC Fabric elements. The FC End Nodes are unaware of the existence of the FCIP Link.

4) FC Primitive Signals, Primitive Sequences, and Class 1 FC Frames are not transmitted across an FCIP Link because they cannot be encoded using FC Frame Encapsulation [19].

5) The path (route) taken by an encapsulated FC Frame follows the normal routing procedures of the IP Network.

6) An FCIP Entity MAY contain multiple FCIP Link Endpoints, but each FCIP Link Endpoint (FCIP_LEP) communicates with exactly one other FCIP_LEP.

7) When multiple FCIP_LEPs with multiple FCIP_DEs are in use, selection of which FCIP_DE to use for encapsulating and transmitting a given FC Frame is covered in FC-BB-2 [3]. FCIP Entities do not actively participate in FC Frame routing.

8) The FCIP Control and Services module MAY use TCP/IP quality of service features (see section 10.2).

9) It is necessary to statically or dynamically configure each FCIP entity with the IP addresses and TCP port numbers corresponding to FCIP Entities with which it is expected to initiate communication. If dynamic discovery of participating FCIP Entities is supported, the function SHALL be performed using the Service Location Protocol (SLPv2) [17]. It is outside the scope of this specification to describe any static configuration method for participating FCIP Entity discovery. Refer to section 8.1.2.2 for a detailed description of dynamic discovery of participating FCIP Entities using SLPv2.

10) Before creating a TCP Connection to a peer FCIP Entity, the FCIP Entity attempting to create the TCP connection SHALL statically or dynamically determine the IP address, TCP port, expected FC Fabric Entity World Wide Name, TCP Connection Parameters, and Quality of Service Information.

11) FCIP Entities do not actively participate in the discovery of FC source and destination identifiers. Discovery of FC addresses (accessible via the FCIP Entity) is provided by techniques and protocols within the FC architecture as described in FC-FS [5] and FC-SW-2 [4].

12) To support IP Network security (see section 9), FCIP Entities MUST:
    1) implement cryptographically protected authentication and cryptographic data integrity keyed to the authentication process, and
    2) implement data confidentiality security features.

13) On an individual TCP Connection, this specification relies on TCP/IP to deliver a byte stream in the same order that it was sent.

14) This specification assumes the presence of and requires the use of TCP and FC data loss and corruption mechanisms. The error detection and recovery features described in this specification complement and support these existing mechanisms.

5. The FCIP Model

5.1. FCIP Protocol Model

The relationship between FCIP and other protocols is illustrated in figure 1.

```
+------------------------+ FCIP Link +------------------------+
|          FCIP          |===========|          FCIP          |
+--------+------+--------+           +--------+------+--------+
|  FC-2  |      |  TCP   |           |  TCP   |      |  FC-2  |
+--------+      +--------+           +--------+      +--------+
|  FC-1  |      |   IP   |           |   IP   |      |  FC-1  |
+--------+      +--------+           +--------+      +--------+
|  FC-0  |      |  LINK  |           |  LINK  |      |  FC-0  |
+--------+      +--------+           +--------+      +--------+
     |          |   PHY  |           |   PHY  |           |
     |          +--------+           +--------+           |
     |               |                    |               |
     |               |     IP Network     |               |
     V               +--------------------+               V
  to Fibre                                             to Fibre
  Channel                                              Channel
  Fabric                                               Fabric
```

```
Key: FC-0 - Fibre Channel Physical Media Layer
     FC-1 - Fibre Channel Encode and Decode Layer
     FC-2 - Fibre Channel Framing and Flow Control Layer
     TCP  - Transmission Control Protocol
     IP   - Internet Protocol
     LINK - IP Link Layer
     PHY  - IP Physical Layer
```

Figure 1: FCIP Protocol Stack Model

Note that the objective of the FCIP Protocol is to create and maintain one or more FCIP Links to transport data.

5.2. FCIP Link

The FCIP Link is the basic unit of service provided by the FCIP Protocol to an FC Fabric. As shown in figure 2, an FCIP Link connects two portions of an FC Fabric using an IP Network as a transport to form a single FC Fabric.

```
/\/\/\/\/\/\         /\/\/\/\/\/\         /\/\/\/\/\/\
\    FC    /         \    IP    /         \    FC    /
/  Fabric  \=========/  Network \=========/  Fabric  \
\/\/\/\/\/\/         \/\/\/\/\/\/         \/\/\/\/\/\/
           |                               |
           |<--------- FCIP Link -------->|
```

Figure: 2 FCIP Link Model

At the points where the ends of the FCIP Link meet portions of the FC Fabric, an FCIP Entity (see section 5.4) combines with an FC Entity as described in section 5.3 to serve as the interface between FC and IP.

An FCIP Link SHALL contain at least one TCP Connection and MAY contain more than one TCP Connection. The endpoints of a single TCP Connection are FCIP Data Engines (see section 5.6). The endpoints of a single FCIP Link are FCIP Link Endpoints (see section 5.5).

5.3. FC Entity

An implementation that tunnels an FC Fabric through an IP Network MUST combine an FC Entity with an FCIP Entity (see section 5.4) to form a complete interface between the FC Fabric and IP Network as shown in figure 3. An FC Fabric Entity may contain multiple instances of the FC/FCIP Entity pair shown on either the right-hand or left-hand side of figure 3.

```
           |<--------- FCIP Link -------->|
           |                              |
+----------+         /\/\/\/\/\/\         +----------+
|   FCIP   |         \    IP    /         |   FCIP   |
|  Entity  |=========/  Network \=========|  Entity  |
+----------+         \/\/\/\/\/\/         +----------+
|    FC    |                              |    FC    |
|  Entity  |                              |  Entity  |
+----------+                              +----------+
     |                                         |
/\/\/\/\/\/\                              /\/\/\/\/\/\
\    FC    /                              \    FC    /
/  Fabric  \                              /  Fabric  \
\/\/\/\/\/\/                              \/\/\/\/\/\/
```

Figure 3: Model for Two Connected FC/FCIP Entity Pairs

In general, the combination of an FCIP Link and two FC/FCIP Entity pairs is intended to provide a non-Fibre Channel backbone transport between Fibre Channel components. For example, this combination can be used to function as the hard-wire connection between two Fibre Channel switches.

The interface between the FC and FCIP Entities is implementation specific. The functional requirements placed on an FC Entity by this specification are listed in appendix H. More information about FC Entities can be found in the Fibre Channel standards and an example of an FC Entity can be found in FC-BB-2 [3].

5.4. FCIP Entity

The model for an FCIP Entity is shown in figure 4.

```
    .......................................................
    : FCIP Entity                                         :
    :                                                     :
    :  +-----------+                                      :
    :  |   FCIP    |                                      :
    :  |Control and|------------------------------------+ :
    :  | Services  |                                    | :
    :  |  Module   |                                    | :
    :  +-----------+                                    | :
    :        |            +--------------------+        | :
    :        |   +-------+--------------------+|----+   | :
    :        |   |+-----+--------------------+|----+|   | :
    :        |   ||+----| FCIP Link Endpoint |----+||   | :
    :        |   |||    +--------------------+    |||   | :
    :.............................................|||.....:
             |   |||                              |||   |
             |   |||                              |||   o<--+
             |   |||                unique TCP    |||   |   |
             |   |||                connections-->|||   |   |
             |   |||                              |||   |   |
          +----------+                         /\/\/\/\/\/\ |
          |    FC    |                         \    IP    / |
          |  Entity  |                         /  Network \ |
          +----------+                         \/\/\/\/\/\/ |
               |                                            |
          /\/\/\/\/\/\                   +------------------+
          \    FC    /                   +->TCP port for
          /  Fabric  \                      incoming
          \/\/\/\/\/\/                      connections
```

Figure 4: FCIP Entity Model

The FCIP Entity receives TCP connect requests on behalf of the FCIP_LEPs that it manages. In support of this, the FCIP Entity is the sole owner of at least one TCP port/IP Address combination used to form TCP Connections. The TCP port may be the FCIP well known port at a given IP Address. An FC Fabric to IP Network interface product SHALL provide each FC/FCIP Entity pair contained in the product with a unique combination of FC Fabric Entity World Wide Identifier and FC/FCIP Entity Identifier values (see section 7).

An FCIP Entity contains an FCIP Control and Services Module to control FCIP link initialization, FCIP link dissolution, and to provide the FC Entity with an interface to key IP Network features.

The interfaces to the IP Network features are implementation specific, however, REQUIRED TCP/IP functional support is specified in this document, including:

- TCP Connections - see section 8
- Security - see section 9
- Performance - see section 10
- Dynamic Discovery - see section 8.1.2.2

The FCIP Link Endpoints in an FCIP Entity provide the FC Frame encapsulation and transmission features of FCIP.

5.5. FCIP Link Endpoint (FCIP_LEP)

As shown in figure 5, the FCIP Link Endpoint contains one FCIP Data Engine for each TCP Connection in the FCIP Link.

```
 ................................................
 : FCIP Link Endpoint                           :
 :                  +------------------+        :
 :         +-------+------------------+|----+   :
 :         |+-----+------------------+|----+|   :
 :         ||+----| FCIP Data Engine |----+||   :
 :         |||    +------------------+    |||   :
 :..............................................:
           |||                            |||
      +----------+                    /\/\/\/\/\/\
      |    FC    |                    \    IP    /
      |  Entity  |                    /  Network \
      +----------+                    \/\/\/\/\/\/
            |
      /\/\/\/\/\/\
      \    FC    /
      /  Fabric  \
      \/\/\/\/\/\/
```

Figure 5: FCIP Link Endpoint Model

Each time a TCP Connection is formed with a new FC/FCIP Entity pair (including all the actions described in section 8.1), the FCIP Entity SHALL create a new FCIP Link Endpoint containing one FCIP Data Engine.

An FCIP_LEP is a transparent data translation point between an FC Entity and an IP Network. A pair of FCIP_LEPs communicating over one or more TCP Connections create an FCIP Link to join two islands of an FC Fabric, producing a single FC Fabric.

The IP Network over which the two FCIP_LEPs communicate is not aware of the FC payloads that it is carrying. Likewise, the FC End Nodes connected to the FC Fabric are unaware of the TCP/IP based transport employed in the structure of the FC Fabric.

An FCIP_LEP uses normal TCP based flow control mechanisms for managing its internal resources and matching them with the advertised TCP Receiver Window Size (see sections 8.3.2, 8.5). An FCIP_LEP MAY communicate with its local FC Entity counterpart to coordinate flow control.

5.6. FCIP Data Engine (FCIP_DE)

The model for one of the multiple FCIP_DEs that MAY be present in an FCIP_LEP is shown in figure 6.

```
     +--------------------------------+
     |                                |
F    |-+    +------------------+    +-|
C    |p|    |  Encapsulation   |    |p|    N
  -->|1|--->|     Engine       |--->|2|--> e
E    |-+    +------------------+    +-|    t
n    |                                |  I w
t    |-+    +------------------+    +-|  P o
i    |p|    | De-Encapsulation |    |p|    r
t <--|4|<---|     Engine       |<---|3|<-- k
y    |-+    +------------------+    +-|
     |                                |
     +--------------------------------+
```

Figure 6: FCIP Data Engine Model

Data enters and leaves the FCIP_DE through four portals (p1 - p4). The portals do not process or examine the data that passes through them. They are only the named access points where the FCIP_DE interfaces with the external world. The names of the portals are as follows:

p1) FC Frame Receiver Portal - The interface through which an FC Frame and time stamp enters an FCIP_DE from the FC Entity.

p2) Encapsulated Frame Transmitter Portal - The TCP interface through which an FCIP Frame is transmitted to the IP Network by an FCIP_DE.

p3) Encapsulated Frame Receiver Portal - The TCP interface through which an FCIP Frame is received from the IP Network by an FCIP_DE.

p4) FC Frame Transmitter Portal - The interface through which a reconstituted FC Frame and time stamp exits an FCIP_DE to the FC Entity.

The work of the FCIP_DE is done by the Encapsulation and De-Encapsulation Engines. The Engines have two functions:

1) Encapsulating and de-encapsulating FC Frames using the encapsulation format described in FC Frame Encapsulation [19] and in section 5.6.1 of this document, and

2) Detecting some data transmission errors and performing minimal error recovery as described in section 5.6.2.

Data flows through a pair of IP Network connected FCIP_DEs in the following seven steps:

1) An FC Frame and time stamp arrives at the FC Frame Receiver Portal and is passed to the Encapsulation Engine. The FC Frame is assumed to have been processed by the FC Entity according to the applicable FC rules and is not validated by the FCIP_DE. If the FC Entity is in the Unsynchronized state with respect to a time base as described in the FC Frame Encapsulation [19] specification, the time stamp delivered with the FC Frame SHALL be zero.

2) In the Encapsulation Engine, the encapsulation format described in FC Frame Encapsulation [19] and in section 5.6.1 of this document SHALL be applied to prepare the FC Frame and associated time stamp for transmission over the IP Network.

3) The entire encapsulated FC Frame (a.k.a. the FCIP Frame) SHALL be passed to the Encapsulated Frame Transmitter Portal where it SHALL be inserted in the TCP byte stream.

4) Transmission of the FCIP Frame over the IP Network follows all the TCP rules of operation. This includes, but is not limited to, the in-order delivery of bytes in the stream, as specified by TCP [6].

5) The FCIP Frame arrives at the partner FCIP Entity where it enters the FCIP_DE through the Encapsulated Frame Receiver Portal and is passed to the De-Encapsulation Engine for processing.

6) The De-Encapsulation Engine SHALL validate the incoming TCP byte stream as described in section 5.6.2.2 and SHALL de-encapsulate the FC Frame and associated time stamp according to the encapsulation format described in FC Frame Encapsulation [19] and in section 5.6.1 of this document.

7) In the absence of errors, the de-encapsulated FC Frame and time stamp SHALL be passed to the FC Frame Transmitter Portal for delivery to the FC Entity. Error handling is discussed in section 5.6.2.2.

Every FC Frame that arrives at the FC Frame Receiver Portal SHALL be transmitted on the IP Network as described in steps 1 through 4 above. In the absence of errors, data bytes arriving at the Encapsulated Frame Receiver Portal SHALL be de-encapsulated and forwarded to the FC Frame Transmitter Portal as described in steps 5 through 7.

5.6.1. FCIP Encapsulation of FC Frames

The FCIP encapsulation of FC Frames employs FC Frame Encapsulation [19].

The features from FC Frame Encapsulation that are unique to individual protocols SHALL be applied as follows for the FCIP encapsulation of FC Frames.

The Protocol# field SHALL contain 1 in accordance with the IANA Considerations annex of FC Frame Encapsulation [19].

The Protocol Specific field SHALL have the format shown in figure 7. Note: the word numbers in figure 7 are relative to the complete FC Frame Encapsulation header, not to the Protocol Specific field.

```
W|------------------------------Bit------------------------------|
o|                                                               |
r|                    1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3|
d|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|              replication of encapsulation word 0              |
 +---------------+---------------+---------------+---------------+
2|    pFlags     |    Reserved   |    -pFlags    |  -Reserved    |
 +---------------+---------------+---------------+---------------+
```

Figure 7: FCIP Usage of FC Frame Encapsulation Protocol Specific field

Word 1 of the Protocol Specific field SHALL contain an exact copy of word 0 in FC Frame Encapsulation [19].

The pFlags (protocol specific flags) field provides information about the protocol specific usage of the FC Encapsulation Header. Figure 8 shows the defined pFlags bits.

```
|----------------Bit--------------------|
|                                       |
|  0    1    2    3    4    5    6    7 |
+----+-----------------------------+----+
| Ch |          Reserved           | SF |
+----+-----------------------------+----+
```

Figure 8: pFlags Field Bits

The SF (Special Frame) bit indicates whether the FCIP Frame is an encapsulated FC Frame or an FSF (FCIP Special Frame, see section 7). When the FCIP Frame contains an encapsulated FC Frame, the SF bit SHALL be 0. When the FCIP Frame is an FSF, the SF bit SHALL be 1.

The FSF SHALL only be sent as the first bytes transmitted in each direction on a newly formed TCP Connection and only one FSF SHALL be transmitted in each direction at that time (see section 8.1). After that all FCIP Frames SHALL have the SF bit set to 0.

The Ch (Changed) bit indicates whether an echoed FSF has been intentionally altered (see section 8.1.3). The Ch bit SHALL be 0 unless the FSF bit is 1. When the initial TCP Connection FSF is sent, the Ch bit SHALL be 0. If the recipient of a TCP connect request echoes the FSF without any changes, then the Ch bit SHALL continue to be 0. If the recipient of a TCP connect request alters the FSF before echoing it, then the Ch bit SHALL be changed to 1.

The -pFlags field SHALL contain the ones complement of the contents of the pFlags field.

Table 1 summarizes the usage of the pFlags SF and Ch bits.

SF	Ch	Originated or Echoed	Validity/Description
0	0	n/a	Encapsulated FC Frame
0	1	n/a	Always Illegal
1	0	Originated	Originated FSF
1	1	Originated	Always Illegal
1	0	Echoed	Echoed FSF without changes
1	1	Echoed	Echoed FSF with changes
Note 1: Echoed FSFs may contain changes resulting from transmission errors, necessitating the comparison between sent and received FSF bytes by the FSF originator described in section 8.1.2.3. Note 2: Column positions in this table do not reflect the bit positions of the SF and Ch bits in the pFlags field.			

Table 1: pFlags SF and Ch bit usage summary

The Reserved pFlags bits SHALL be 0.

The Reserved field (bits 23-16 in word 2): SHALL contain 0.

The -Reserved field (bits 7-0 in word 2): SHALL contain 255 (or 0xFF).

The CRCV (CRC Valid) Flag SHALL be set to 0.

The CRC field SHALL be set to 0.

In FCIP, the SOF and EOF codes listed as Class 2, Class 3, and Class 4 in the FC Frame Encapsulation [19] are legal.

5.6.2. FCIP Data Engine Error Detection and Recovery

5.6.2.1. TCP Assistance With Error Detection and Recovery

TCP [6] requires in order delivery, generation of TCP checksums, and checking of TCP checksums. Thus, the byte stream passed from TCP to the FCIP_LEP will be in order and free of errors detectable by the TCP checksum. The FCIP_LEP relies on TCP to perform these functions.

5.6.2.2. Errors in FCIP Headers and Discarding FCIP Frames

Bytes delivered through the Encapsulated Frame Receiver Portal that are not correctly delimited as defined by the FC Frame Encapsulation [19] are considered to be in error.

The failure of the Protocol# and Version fields in the FCIP Frame header to contain the values defined for an FCIP Frame SHALL be considered an error.

Further, some errors in the encapsulation will result in the FCIP_DE losing synchronization with the FC Frames in the byte stream entering through the Encapsulated Frame Receiver Portal.

The Frame Length field in the FC Frame Encapsulation header is used to determine where in the data stream the next FC Encapsulated Header is located. The following tests SHALL be performed to verify synchronization with the byte stream entering the Encapsulated Frame Receiver Portal, and synchronization SHALL be considered lost if any of the tests fail:

1) Frame Length field validation -- 15 < Frame Length < 545;

2) Comparison of Frame Length field to its ones complement; and

3) A valid EOF is found in the word preceding the start of the next FCIP header as indicated by the Frame Length field, to be tested as follows:

   1) Bits 24-31 and 16-23 contain identical legal EOF values (the list of legal EOF values is in the FC Frame Encapsulation [19]); and

   2) Bits 8-15 and 0-7 contain the ones complement of the EOF value found in bits 24-31.

Note: The range of valid Frame Length values is derived as follows. The FCIP Frame header is seven words, one word each is required for the encoded SOF and EOF values, the FC Frame header is six words, and the FC CRC requires one word, yielding a base Frame Length of 16 (7+1+1+6+1) words, if no FC Payload is present. Since the FC Payload is optional, any Frame Length value greater than 15 is valid. The maximum FC Payload size is 528 words, meaning that any Frame Length value up to and including 544 (528+16) is valid.

If synchronization is lost, the FC Frame SHALL NOT be forwarded on to the FC Entity and further recovery SHALL be handled as defined by section 5.6.2.3.

In addition to the tests above, the validity and positioning of the following FCIP Frame information SHOULD be used to detect encapsulation errors that may or may not affect synchronization:

a) Protocol# ones complement field (1 test);
b) Version ones complement field (1 test);
c) Replication of encapsulation word 0 in word 1 (1 test);
d) Reserved field and its ones complement (2 tests);
e) Flags field and its ones complement (2 tests);
f) CRC field is equal to zero (1 test);
g) SOF fields and ones complement fields (4 tests);
h) Format and values of FC header (1 test);
i) CRC of FC Frame (2 tests);
j) FC Frame Encapsulation header information in the next FCIP Frame (1 test).

At least 3 of the 16 tests listed above SHALL be performed. Failure of any of the above tests actually performed SHALL indicate an encapsulation error and the FC Frame SHALL NOT be forwarded on to the FC Entity. Further, such errors SHOULD be considered carefully, since some may be synchronization errors.

Whenever an FCIP_DE discards bytes delivered through the Encapsulated Frame Receiver Portal, it SHALL cause the FCIP Entity to notify the FC Entity of the condition and provide a suitable description of the reason bytes were discarded.

The burden for recovering from discarded data falls on the FC Entity and other components of the FC Fabric, and is outside the scope of this specification.

5.6.2.3. Synchronization Failures

If an FCIP_DE determines that it cannot find the next FCIP Frame header in the byte stream entering through the Encapsulated Frame Receiver Portal, the FCIP_DE SHALL do one of the following:

a) close the TCP Connection [6] [7] and notify the FC Entity with the reason for the closure;

b) recover synchronization by searching the bytes delivered by the Encapsulated Frame Receiver Portal for a valid FCIP Frame header having the correct properties (see section 5.6.2.2), and discarding bytes delivered by the Encapsulated Frame Receiver Portal until a valid FCIP Frame header is found; or

c) attempt to recover synchronization as described in b) and if synchronization cannot be recovered, close the TCP Connection as described in a), including notification of the FC Entity with the reason for the closure.

If the FCIP_DE attempts to recover synchronization, the resynchronization algorithm used SHALL meet the following requirements:

a) discard or identify with an EOFa (see appendix section F.1) those FC Frames and fragments of FC Frames identified before synchronization has again been completely verified. The number of FC Frames not forwarded may vary based on the algorithm used;

b) return to forwarding FC Frames through the FC Frame Transmitter Portal only after synchronization on the transmitted FCIP Frame stream has been verified; and

c) close the TCP/IP connection if the algorithm ends without verifying successful synchronization. The probability of failing to synchronize successfully and the time necessary to determine whether or not synchronization was successful may vary with the algorithm used.

An example algorithm meeting these requirements can be found in appendix D.

The burden for recovering from the discarding of FCIP Frames during the optional resynchronization process described in this section falls on the FC Entity and other components of the FC Fabric, and is outside the scope of this specification.

6. Checking FC Frame Transit Times in the IP Network

FC-BB-2 [3] defines how the measurement of IP Network transit time is performed, based on the requirements stated in the FC Frame Encapsulation [19] specification. The choice to place this implementation requirement on the FC Entity is based on a desire to include the transit time through the FCIP Entities when computing the IP Network transit time experienced by the FC Frames.

Each FC Frame that enters the FCIP_DE through the FC Frame Receiver Portal SHALL be accompanied by a time stamp value that the FCIP_DE SHALL place in the Time Stamp [integer] and Time Stamp [fraction] fields of the encapsulation header of the FCIP Frame that contains the FC Frame. If no synchronized time stamp value is available to accompany the entering FC Frame, a value of zero SHALL be used.

Each FC Frame that exits the FCIP_DE through the FC Frame Transmitter Portal SHALL be accompanied by the time stamp value taken from the FCIP Frame that encapsulated the FC Frame.

The FC Entity SHALL use suitable internal clocks and either Fibre Channel services or an SNTP Version 4 server [26] to establish and maintain the required synchronized time value. The FC Entity SHALL verify that the FC Entity it is communicating with on an FCIP Link is using the same synchronized time source, either Fibre Channel services or SNTP server.

Note that since the FC Fabric is expected to have a single synchronized time value throughout, reliance on the Fibre Channel services means that only one synchronized time value is needed for all FCIP_DEs regardless of their connection characteristics.

7. The FCIP Special Frame (FSF)

7.1. FCIP Special Frame Format

Figure 9 shows the FSF format.

```
 W|------------------------------Bit------------------------------|
 o|                                                               |
 r|                    1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3|
 d|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|   Protocol#   |    Version    |  -Protocol#   |   -Version    |
  |    (0x01)     |    (0x01)     |    (0xFE)     |    (0xFE)     |
  +---------------+---------------+---------------+---------------+
 1|   Protocol#   |    Version    |  -Protocol#   |   -Version    |
  |    (0x01)     |    (0x01)     |    (0xFE)     |    (0xFE)     |
  +---------------+---------------+---------------+---------------+
 2|    pFlags     |   Reserved    |    -pFlags    |  -Reserved    |
  |               |    (0x00)     |               |    (0xFF)     |
  +-----------+---+---------------+-----------+---+---------------+
 3|   Flags   |  Frame Length     |  -Flags   |  -Frame Length    |
  | (0b000000)| (0b0000010011)    | (0b111111)|  (0b1111101100)   |
  +-----------+-------------------+-----------+-------------------+
 4|                     Time Stamp [integer]                      |
  +---------------------------------------------------------------+
 5|                     Time Stamp [fraction]                     |
  +---------------------------------------------------------------+
 6|                    CRC (Reserved in FCIP)                     |
  |                       (0x00-00-00-00)                         |
  +-------------------------------+-------------------------------+
 7|           Reserved            |           -Reserved           |
  |           (0x00-00)           |           (0xFF-FF)           |
  +-------------------------------+-------------------------------+
 8|                                                               |
  +-----       Source FC Fabric Entity World Wide Name       -----+
 9|                                                               |
  +---------------------------------------------------------------+
10|                                                               |
  +-----          Source FC/FCIP Entity Identifier           -----+
11|                                                               |
  +---------------------------------------------------------------+
12|                                                               |
  +-----                  Connection Nonce                   -----+
13|                                                               |
  +---------------+---------------+-------------------------------+
                            (Continued)
```

Figure 9: FSF Format (part 1 of 2)

```
   W|------------------------------Bit------------------------------|
   o|                                                               |
   r|                    1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3|
   d|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|
    |                                                               |
    |                          (Concluded)                          |
    +---------------------------------------------------------------+
  14|   Connection   |    Reserved    |    Connection Usage Code    |
    |  Usage Flags   |     (0x00)     |    <defined in FC-BB-2>     |
    +----------------+----------------+-----------------------------+
  15|                                                               |
    +-----    Destination FC Fabric Entity World Wide Name     -----+
  16|                                                               |
    +---------------------------------------------------------------+
  17|                            K_A_TOV                            |
    +-------------------------------+-------------------------------+
  18|           Reserved            |           -Reserved           |
    |          (0x00-00)            |           (0xFF-FF)           |
    +-------------------------------+-------------------------------+
```

Figure 9: FSF Format (part 2 of 2)

The FSF SHALL only be sent as the first bytes transmitted in each direction on a newly formed TCP Connection, and only one FSF SHALL be transmitted in each direction.

The contents of the FSF SHALL be as described for encapsulated FC Frames, except for the fields described in this section.

All FSFs SHALL have the pFlags SF bit set to 1 (see section 5.6.1).

The Source FC Fabric Entity World Wide Name field SHALL contain the Fibre Channel Name_Identifier [5] for the FC Fabric entity associated with the FC/FCIP Entity pair that generates (as opposed to echoes) the FSF. For example, if the FC Fabric entity is a FC Switch, the FC Fabric Entity World Wide Name field SHALL contain the Switch_Name [4]. The Source FC Fabric Entity World Wide Name SHALL be world wide unique.

The Source FC/FCIP Entity Identifier field SHALL contain a unique identifier for the FC/FCIP Entity pair that generates (as opposed to echoes) the FSF. The value is assigned by the FC Fabric entity whose world wide name appears in the Source FC Fabric Entity World Wide Name field.

Note: The combination of the Source FC Entity World Wide Name and Source FC/FCIP Entity Identifier fields uniquely identifies every FC/FCIP Entity pair in the IP Network.

The Connection Nonce field shall contain a 64-bit random number generated to uniquely identify a single TCP connect request. In order to provide sufficient security for the connection nonce, the Randomness Recommendations for Security [9] SHOULD be followed.

The Connection Usage Flags field identifies the types of SOF values [19] to be carried on the connection as shown in figure 10.

```
|------------------------------Bit-------------------------------|
|                                                                |
|    0      1       2       3       4       5       6       7    |
+-------+-------+-------+-------+--------------------------------+
|  SOFf | SOF?2 | SOF?3 | SOF?4 |            Reserved            |
+-------+-------+-------+-------+--------------------------------+
```

Figure 10: Connection Usage Flags Field Format

If the SOFf bit is one, then FC Frames containing SOFf are intended to be carried on the connection.

If the SOF?2 bit is one, then FC Frames containing SOFi2 and SOFn2 are intended to be carried on the connection.

If the SOF?3 bit is one, then FC Frames containing SOFi3 and SOFn3 are intended to be carried on the connection.

If the SOF?4 bit is one, then FC Frames containing SOFi4, SOFn4, and SOFc4 are intended to be carried on the connection.

All or none of the SOFf, SOF?2, SOF?3, and SOF?4 bits MAY be set to one. If all of the SOFf, SOF?2, SOF?3, and SOF?4 bits are zero, then the types of FC Frames intended to be carried on the connection have no specific relationship to the SOF code.

The FCIP Entity SHALL NOT enforce the SOF usage described by the Connection Usage Flags field and SHALL only use the contents of the field as described below.

The Connection Usage Code field contains Fibre Channel defined information regarding the intended usage of the connection as specified in FC-BB-2 [3].

The FCIP Entity SHALL use the contents of the Connection Usage Flags and Connection Usage Code fields to locate appropriate QoS settings in the "shared" database of TCP Connection information (see section 8.1.1) and apply those settings to a newly formed connection.

The Destination FC Fabric Entity World Wide Name field MAY contain the Fibre Channel Name_Identifier [5] for the FC Fabric entity associated with the FC/FCIP Entity pair that echoes (as opposed to generates) the Special Frame.

The K_A_TOV field SHALL contain the FC Keep Alive Timeout value to be applied to the new TCP Connection as specified in FC-BB-2 [3].

For each new incoming TCP connect request and subsequent FSF received, the FCIP Entity SHALL send the contents of the Source FC Fabric Entity World Wide Name, Source FC/FCIP Identifier, Connection Usage Flags and Connection Usage Code fields to the FC Entity along with the other connection information (e.g., FCIP_LEP and FCIP_DE information).

7.2. Overview of FSF Usage in Connection Establishment

When a new TCP Connection is established, an FCIP Special Frame makes one round trip from the FCIP Entity initiating the TCP connect operation to the FCIP Entity receiving the TCP connect request and back. This FSF usage serves three functions:

- Identification of the FCIP Link endpoints

- Conveyance of a few critical parameters shared by the FC/FCIP Entity pairs involved in the FCIP Link

- Configuration discovery (used in place of SLP only when allowed by site security policies)

The specific format and protocol requirements for this usage of the FSF are found in sections 7.1 and 8.1.2.3. This section provides an overview of the FSF usage without stating requirements.

Because FCIP is only a tunnel for a Fibre Channel Fabric and because the Fabric has its own complex link setup algorithm that can be employed for many FCIP link setup needs, it is desirable to minimize the complexity of the FSF usage during TCP Connection setup. With this in mind, this FSF usage is not a login or parameter negotiation mechanism. A single FSF transits each newly established TCP connection as the first bytes sent in each direction.

Note: This usage of the FSF cannot be eliminated entirely because a newly created TCP Connection must be associated with the correct FCIP Link before FC Fabric initialization of the connection can commence.

The first bytes sent from the TCP connect request initiator to the receiver are an FSF identifying both the sender and who the sender thinks is the receiver. If the contents of this FSF are correct and acceptable to the receiver, the unchanged FSF is echoed back to the sender. This send/echo process is the only set of actions that allows the TCP Connection to be used to carry FC Fabric traffic. If the send and unchanged echo process does not occur, the algorithm followed at one or both ends of the TCP Connection results in the closure of the TCP Connection (see section 8.1 for specific algorithm requirements).

Note: Owing to the limited manner in which the FSF is used and the requirement that the FSF be echoed without changes before a TCP Connection is allowed to carry user data, no error checking beyond that provided by TCP is deemed necessary.

As described above, the primary purpose of the FSF usage during TCP Connection setup is identifying the FCIP Link to which the new TCP Connection belongs. From these beginnings, it is only a small stretch to envision using the FSF as a simplified configuration discovery tool, and the mechanics of such a usage are described in section 8.1.

However, use of the FSF for configuration discovery lacks the broad range of capabilities provided by SLPv2 and most particularly lacks the security capabilities of SLPv2. For these reasons, using the FSF for configuration discovery is not appropriate for all environments. Thus the choice to use the FSF for discovery purposes is a policy choice to be included in the TCP Connection Establishment "shared" database described in section 8.1.1.

When FSF-based configuration discovery is enabled, the normal TCP Connection setup rules outlined above are modified as follows.

Normally, the algorithm executed by an FCIP Entity receiving an FSF includes verifying that its own identification information in the arriving FSF is correct and closing the TCP Connection if it is not. This can be viewed as requiring the initiator of a TCP connect request to know in advance the identity of the FCIP Entity that is the target of that request (using SLP, for example), and through the FSF effectively saying, "I think I’m talking to X." If the party at the other end of the TCP connect request is really Y, then it simply hangs up.

FSF-based discovery allows the "I think I'm talking to X" to be replaced with "Please tell me who I am talking to?", which is accomplished by replacing an explicit value in the Destination FC Fabric Entity World Wide Name field with zero.

If the policy at the receiving FCIP Entity allows FSF-based discovery, the zero is replaced with the correct Destination FC Fabric Entity World Wide Name value in the echoed FSF. This is still subject to the rules of sending with unchanged echo, and so closure of TCP Connection occurs after the echoed FSF is received by the TCP connect initiator.

Despite the TCP Connection closure, however, the TCP connect initiator now knows the correct Destination FC Fabric Entity World Wide Name identity of the FCIP Entity at a given IP Address and a subsequent TCP Connection setup sequence probably will be successful.

The Ch bit in the pFlags field (see section 5.6.1) allows for differentiation between changes in the FSF resulting from transmission errors and changes resulting from intentional acts by the FSF recipient.

8. TCP Connection Management

8.1. TCP Connection Establishment

8.1.1. Connection Establishment Model

The description of the connection establishment process is a model for the interactions between an FC Entity and an FCIP Entity during TCP Connection establishment. The model is written in terms of a "shared" database that the FCIP Entity consults to determine the properties of the TCP Connections to be formed combined with routine calls to the FC Entity when connections are successfully established. Whether the FC Entity contributes information to the "shared" database is not critical to this model. However, the fact that the FCIP Entity MAY consult the database at any time to determine its actions relative to TCP Connection establishment is important.

It is important to remember that this description is only a model for the interactions between an FC Entity and an FCIP Entity. Any implementation that has the same effects on the FC Fabric and IP Network as those described using the model meets the requirements of this specification. For example, an implementation might replace the "shared" database with a routine interface between the FC and FCIP Entities.

8.1.2. Creating New TCP Connections

8.1.2.1. Non-Dynamic Creation of New TCP Connections

When an FCIP Entity discovers that a new TCP Connection needs to be established, it SHALL determine the IP Address to which the TCP Connection is to be made and establish all enabled IP security features for that IP Address as described in section 9. Then the FCIP Entity SHALL determine the following information about the new connection in addition to the IP Address:

- The expected Destination FC Fabric Entity World Wide Name of the FC/FCIP Entity pair to which the TCP Connection is being made

- TCP Connection Parameters (see section 8.3)

- Quality of Service Information (see section 10)

Based on this information, the FCIP Entity SHALL generate a TCP connect request [6] to the FCIP Well-Known Port of 3225 (or other configuration specific port number) at the specified IP Address.

If the TCP connect request is rejected, the FCIP Entity SHALL act to limit unnecessary repetition of attempts to establish similar connections. For example, the FCIP Entity might wait 60 seconds before trying to re-establish the connection.

If the TCP connect request is accepted, the FCIP Entity SHALL follow the steps described in section 8.1.2.3 to complete the establishment of a new FCIP_DE.

It is RECOMMENDED that an FCIP Entity not initiate TCP connect requests to another FCIP Entity if incoming TCP connect requests from that FCIP Entity have already been accepted.

8.1.2.2. Dynamic Creation of New TCP Connections

If dynamic discovery of participating FCIP Entities is supported, the function SHALL be performed using the Service Location Protocol (SLPv2) [17] in the manner defined for FCIP usage [20].

Upon discovering that dynamic discovery is to be used, the FCIP Entity SHALL enable IP security features for the SLP discovery process as described in [20] and then:

1) Determine the one or more FCIP Discovery Domain(s) to be used in the dynamic discovery process;

2) Establish an SLPv2 Service Agent to advertise the availability of this FCIP Entity to peer FCIP Entities in the identified FCIP Discovery Domain(s); and

3) Establish an SLPv2 User Agent to locate service advertisements for peer FCIP Entities in the identified FCIP Discovery Domain(s).

For each peer FCIP Entity dynamically discovered through the SLPv2 User Agent, the FCIP Entity SHALL establish all enabled IP security features for the discovered IP Address as described in section 9 and then determine the following information about the new connection:

- The expected Destination FC Fabric Entity World Wide Name of the FC/FCIP Entity pair to which the TCP Connection is being made

- TCP Connection Parameters (see section 8.3)

- Quality of Service Information (see section 10)

Based on this information, the FCIP Entity SHALL generate a TCP connect request [6] to the FCIP Well-Known Port of 3225 (or other configuration specific port number) at the IP Address specified by the service advertisement. If the TCP connect request is rejected, act to limit unnecessary repetition of attempts to establish similar connections. If the TCP connect request is accepted, the FCIP Entity SHALL follow the steps described in section 8.1.2.3 to complete the establishment of a new FCIP_DE.

It is recommended that an FCIP Entity not initiate TCP connect requests to another FCIP Entity if incoming TCP connect requests from that FCIP Entity have already been accepted.

8.1.2.3. Connection Setup After a Successful TCP Connect Request

Whether Non-Dynamic TCP Connection creation (see section 8.1.2.1) or Dynamic TCP Connection creation (see section 8.1.2.2) is used, the steps described in this section SHALL be followed to take the TCP Connection setup process to completion.

After the TCP connect request has been accepted, the FCIP Entity SHALL send an FCIP Special Frame (FSF, see section 7) as the first bytes transmitted on the newly formed connection, and retain a copy of those bytes for later comparisons. All fields in the FSF SHALL be filled in as described in section 7, particularly:

- The Source FC Fabric Entity World Wide Name field SHALL contain the FC Fabric Entity World Wide Name for the FC/FCIP Entity pair that is originating the TCP connect request;

- The Source FC/FCIP Entity Identifier field SHALL contain a unique identifier that is assigned by the FC Fabric entity whose world wide name appears in the Source FC Fabric Entity World Wide Name field;

- The Connection Nonce field SHALL contain a 64-bit random number that differs in value from any recently used Connection Nonce value. In order to provide sufficient security for the connection nonce, the Randomness Recommendations for Security [9] SHOULD be followed; and

- The Destination FC Fabric Entity World Wide Name field SHALL contain 0 or the expected FC Fabric Entity World Wide Name for the FC/FCIP Entity pair whose destination is the TCP connect request.

After the FSF is sent on the newly formed connection, the FCIP Entity SHALL wait for the FSF to be echoed as the first bytes received on the newly formed connection.

The FCIP Entity MAY apply a timeout of not less than 90 seconds while waiting for the echoed FSF bytes. If the timeout expires, the FCIP Entity SHALL close the TCP Connection and notify the FC Entity with the reason for the closure.

If the echoed FSF bytes do not exactly match the FSF bytes sent (words 7 through 17 inclusive) or if the echoed Destination FC Fabric Entity World Wide Name field contains zero, the FCIP Entity SHALL close the TCP Connection and notify the FC Entity with the reason for the closure.

The FCIP Entity SHALL only perform the following steps if the echoed FSF bytes exactly match the FSF bytes sent (words 7 through 17 inclusive).

1) Instantiate the appropriate Quality of Service (see section 10) conditions on the newly created TCP Connection,

2) If the IP Address and TCP Port to which the TCP Connection was made is not associated with any other FCIP_LEP, create a new FCIP_LEP for the new FCIP Link,

3) Create a new FCIP_DE within the newly created FCIP_LEP to service the new TCP Connection, and

4) Inform the FC Entity of the new FCIP_LEP, FCIP_DE, Destination FC Fabric Entity World Wide Name, Connection Usage Flags, and Connection Usage Code.

8.1.3. Processing Incoming TCP Connect Requests

The FCIP Entity SHALL listen for new TCP Connection requests [6] on the FCIP Well-Known Port (3225). An FCIP Entity MAY also accept and establish TCP Connections to a TCP port number other than the FCIP Well-Known Port, as configured by the network administrator in a manner outside the scope of this specification.

The FCIP Entity SHALL determine the following information about the requested connection:

- Whether the "shared" database (see section 8.1.1) allows the requested connection

- Whether IP security setup has been performed for the IP security features enabled on the connection (see section 9)

If the requested connection is not allowed, the FCIP Entity SHALL reject the connect request using appropriate TCP means. If the requested connection is allowed, the FC Entity SHALL ensure that required IP security features are enabled and accept the TCP connect request.

After the TCP connect request has been accepted, the FCIP Entity SHALL wait for the FSF sent by the originator of the TCP connect request (see section 8.1.2) as the first bytes received on the accepted connection.

The FCIP Entity MAY apply a timeout of no less than 90 seconds while waiting for the FSF bytes. If the timeout expires, the FCIP Entity SHALL close the TCP Connection and notify the FC Entity with the reason for the closure.

Note: One method for attacking the security of the FCIP Link formation process (detailed in section 9.1) depends on keeping a TCP connect request open without sending an FSF. Implementations should bear this in mind in the handling of TCP connect requests where the FSF is not sent in a timely manner.

Upon receipt of the FSF sent by the originator of the TCP connect request, the FCIP Entity SHALL inspect the contents of the following fields:

- Connection Nonce,
- Destination FC Fabric Entity World Wide Name,
- Connection Usage Flags, and
- Connection Usage Code.

If the Connection Nonce field contains a value identical to the most recently received Connection Nonce from the same IP Address, the FCIP Entity SHALL close the TCP Connection and notify the FC Entity with the reason for the closure.

If an FCIP Entity receives a duplicate FSF during the FCIP Link formation process, it SHALL close that TCP Connection and notify the FC Entity with the reason for the closure.

If the Destination FC Fabric Entity World Wide Name contains 0, the FCIP Entity SHALL take one of the following three actions:

1) Leave the Destination FC Fabric Entity World Wide Name field and Ch bit both 0;

2) Change the Destination FC Fabric Entity World Wide Name field to match FC Fabric Entity World Wide Name associated with the FCIP Entity that received the TCP connect request and change the Ch bit to 1; or

3) Close the TCP Connection without sending any response.

The choice between the above actions depends on the anticipated usage of the FCIP Entity. The FCIP Entity may consult the "shared" database when choosing between the above actions.

If:

a) The Destination FC Fabric Entity World Wide Name contains a non-zero value that does not match the FC Fabric Entity World Wide Name associated with the FCIP Entity that received the TCP connect request, or

b) The contents of the Connection Usage Flags and Connection Usage Code fields is not acceptable to the FCIP Entity that received the TCP connect request, then the FCIP Entity SHALL take one of the following two actions:

   1) Change the contents of the unacceptable fields to correct/ acceptable values and set the Ch bit to 1; or

   2) Close the TCP Connection without sending any response.

If the FCIP Entity makes any changes in the content of the FSF, it SHALL also set the Ch bit to 1.

If any changes have been made in the received FSF during the processing described above, the following steps SHALL be performed:

1) The changed FSF SHALL be echoed to the originator of the TCP connect request as the only bytes transmitted on the accepted connection;

2) The TCP Connection SHALL be closed (the FC Entity need not be notified of the TCP Connection closure in this case because it is not indicative of an error); and

3) All of the additional processing described in this section SHALL be skipped.

The remaining steps in this section SHALL be performed only if the FCIP Entity has not changed the contents of the above mentioned fields to correct/acceptable values.

If the Source FC Fabric Entity World Wide Name and Source FC/FCIP Entity Identifier field values in the FSF do not match the Source FC Fabric Entity World Wide Name and Source FC/FCIP Entity Identifier associated with any other FCIP_LEP, the FCIP Entity SHALL:

1) Echo the unchanged FSF to the originator of the TCP connect request as the first bytes transmitted on the accepted connection;

2) Instantiate the appropriate Quality of Service (see section 10.2) conditions on the newly created TCP Connection, considering the Connection Usage Flags and Connection Usage Code fields, and "shared" database information (see section 8.1.1) as appropriate,

3) Create a new FCIP_LEP for the new FCIP Link,

4) Create a new FCIP_DE within the newly created FCIP_LEP to service the new TCP Connection, and

5) Inform the FC Entity of the new FCIP_LEP, FCIP_DE, Source FC Fabric Entity World Wide Name, Source FC/FCIP Entity Identifier, Connection Usage Flags, and Connection Usage Code.

If the Source FC Fabric Entity World Wide Name and Source FC/FCIP Entity Identifier field values in the FCIP Special Frame match the Source FC Fabric Entity World Wide Name and Source FC/FCIP Entity Identifier associated with an existing FCIP_LEP, the FCIP Entity SHALL:

1) Request that the FC Entity authenticate the source of the TCP connect request (see FC-BB-2 [3]), providing the following information to the FC Entity for authentication purposes:

a) Source FC Fabric Entity World Wide Name,
b) Source FC/FCIP Entity Identifier, and
c) Connection Nonce.

The FCIP Entity SHALL NOT use the new TCP Connection for any purpose until the FC Entity authenticates the source of the TCP connect request. If the FC Entity indicates that the TCP connect request cannot be properly authenticated, the FCIP Entity SHALL close the TCP Connection and skip all of the remaining steps in this section.

The definition of the FC Entity SHALL include an authentication mechanism for use in response to a TCP connect request source that communicates with the partner FC/FCIP Entity pair on an existing FCIP Link. This authentication mechanism should use a previously authenticated TCP Connection in the existing FCIP Link to authenticate the Connection Nonce sent in the new TCP Connection setup process. The FCIP Entity SHALL treat failure of this authentication as an authentication failure for the new TCP Connection setup process.

2) Echo the unchanged FSF to the originator of the TCP connect request as the first bytes transmitted on the accepted connection;

3) Instantiate the appropriate Quality of Service (see section 10.2) conditions on the newly created TCP Connection, considering the Connection Usage Flags and Connection Usage Code fields, and "shared" database information (see section 8.1.1) as appropriate,

4) Create a new FCIP_DE within the existing FCIP_LEP to service the new TCP Connection, and

5) Inform the FC Entity of the FCIP_LEP, Source FC Fabric Entity World Wide Name, Source FC/FCIP Entity Identifier, Connection Usage Flags, Connection Usage Code, and new FCIP_DE.

Note that the originator of TCP connect requests uses the IP Address and TCP Port to identify which TCP Connections belong to which FCIP_LEPs while the recipient of TCP connect requests uses the Source FC Fabric Entity World Wide Name, and Source FC/FCIP Entity Identifier fields from the FSF to identify which TCP Connection belong to which FCIP_LEPs. For this reason, an FCIP Entity that both originates and receives TCP connect requests is unable to match the FCIP_LEPs associated with originated TCP connect requests to the FCIP_LEPs associated with received TCP connect requests.

8.1.4. Simultaneous Connection Establishment

If two FCIP Entities perform simultaneous open operations, then two TCP Connections are formed and the SF originates at one end on one connection and at the other end on the other. Connection setup proceeds as described above on both connections, and the steps described above properly result in the formation of two FCIP Links between the same FCIP Entities.

This is not an error. Fibre Channel is perfectly capable of handling two approximately equal connections between FC Fabric elements.

The decision to setup pairs of FCIP Links in this manner is considered to be a site policy decision that can be covered in the "shared" database described in section 8.1.1.

8.2. Closing TCP Connections

The FCIP Entity SHALL provide a mechanism with acknowledgement by which the FC Entity is able to cause the closing of an existing TCP Connection at any time. This allows the FC Entity to close TCP Connections that are producing too many errors, etc.

8.3. TCP Connection Parameters

In order to provide efficient management of FCIP_LEP resources as well as FCIP Link resources, consideration of certain TCP Connection parameters is recommended.

8.3.1. TCP Selective Acknowledgement Option

The Selective Acknowledgement option RFC 2883 [18] allows the receiver to acknowledge multiple lost packets in a single ACK, enabling faster recovery. An FCIP Entity MAY negotiate use of TCP SACK and use it for faster recovery from lost packets and holes in TCP sequence number space.

8.3.2. TCP Window Scale Option

The TCP Window Scale option [8] allows TCP window sizes larger than 16-bit limits to be advertised by the receiver. It is necessary to allow data in long fat networks to fill the available pipe. This also implies buffering on the TCP sender that matches the (bandwidth*delay) product of the TCP Connection. An FCIP_LEP uses locally available mechanisms to set a window size that matches the available local buffer resources and the desired throughput.

8.3.3. Protection Against Sequence Number Wrap

It is RECOMMENDED that FCIP Entities implement protection against wrapped sequence numbers PAWS [8]. It is quite possible that within a single connection, TCP sequence numbers wrap within a timeout window.

8.3.4. TCP_NODELAY Option

FCIP Entities should disable the Nagle Algorithm as described in RFC 1122 [7] section 4.2.3.4. By tradition, this can be accomplished by setting the TCP_NODELAY option to one at the local TCP interface.

8.4. TCP Connection Considerations

In idle mode, a TCP Connection "keep alive" option of TCP is normally used to keep a connection alive. However, this timeout is fairly large and may prevent early detection of loss of connectivity. In order to facilitate faster detection of loss of connectivity, FC Entities SHOULD implement some form of Fibre Channel connection failure detection (see FC-BB-2 [3]).

When an FCIP Entity discovers that TCP connectivity has been lost, the FCIP Entity SHALL notify the FC Entity of the failure including information about the reason for the failure.

8.5. Flow Control Mapping between TCP and FC

The FCIP Entity and FC Entity are connected to the IP Network and FC Fabric, respectively, and they need to follow the flow control mechanisms of both TCP and FC, which work independently of each other.

This section provides guidelines as to how the FCIP Entity can map TCP flow control to status notifications to the FC Entity.

There are two scenarios in which the flow control management becomes crucial:

1) When there is line speed mismatch between the FC and IP interfaces.

   Even though it is RECOMMENDED that both of the FC and IP interfaces to the FC Entity and FCIP Entity, respectively, be of comparable speeds, it is possible to carry FC traffic over an IP Network that has a different line speed and bit error rate.

2) When the FC Fabric or IP Network encounters congestion.

   Even when both the FC Fabric or IP network are of comparable speeds, during the course of operation, the FC Fabric or the IP Network could encounter congestion due to transient conditions.

The FC Entity uses Fibre Channel mechanisms for flow control at the FC Frame Receiver Portal based on information supplied by the FCIP Entity regarding flow constraints at the Encapsulated Frame Transmitter Portal. The FCIP Entity uses TCP mechanisms for flow control at the Encapsulated Frame Receiver Portal based on information supplied by the FC Entity regarding flow constraints at the FC Frame Transmitter Portal.

Coordination of these flow control mechanisms, one of which is credit based and the other of which is window based, depends on a painstaking design that is outside the scope of this specification.

9. Security

FCIP utilizes the IPsec protocol suite to provide data confidentiality and authentication services, and IKE as the key management protocol. This section describes the requirements for various components of these protocols as used by FCIP, based on FCIP operating environments. Additional consideration for use of IPsec and IKE with the FCIP protocol can be found in [21]. In the event that requirements in [21] conflict with requirements stated in this document, the requirements in this document SHALL prevail.

9.1. Threat Models

Using a general purpose, wide-area network, such as an IP Network, as a functional replacement for physical cabling introduces some security problems not normally encountered in Fibre Channel Fabrics. FC interconnect cabling is typically protected physically from outside access. Public IP Networks allow hostile parties to impact the security of the transport infrastructure.

The general effect is that the security of an FC Fabric is only as good as the security of the entire IP Network that carries the FCIP Links used by that FC Fabric. The following broad classes of attacks are possible:

1) Unauthorized Fibre Channel elements can gain access to resources through normal Fibre Channel Fabric and processes. Although this is a valid threat, securing the Fibre Channel Fabrics is outside the scope of this document. Securing the IP Network is the issue considered in this specification.

2) Unauthorized agents can monitor and manipulate Fibre Channel traffic flowing over physical media used by the IP Network and accessible to the agent.

3) TCP Connections may be hijacked and used to instantiate an invalid FCIP Link between two peer FCIP Entities.

4) Valid and invalid FCIP Frames may be injected on the TCP Connections.

5) The payload of an FCIP Frame may be altered or transformed. The TCP checksum, FCIP ones complement checks, and FC frame CRC do not protect against this because all of them can be modified or regenerated by a malicious and determined adversary.

6) Unauthorized agents can masquerade as valid FCIP Entities and disturb proper operation of the Fibre Channel Fabric.

7) Denial of Service attacks can be mounted by injecting TCP Connection requests and other resource exhaustion operations.

8) An adversary may launch a variety of attacks against the discovery process [17].

9) An attacker may exploit the FSF authentication mechanism of the FCIP Link formation process (see section 8.1.3). The attacker could observe the FSF contents sent on an initial connection of an FCIP Link and use the observed nonce, Source FC/FCIP Entity Identifier, and other FSF contents to form an FCIP Link using the attacker’s own previously established connection, while resetting/blocking the observed connection. Although the use of timeout for reception of FSF reduces the risk of this attack, such an attack is possible. See section 9.3.1 to protect against this specific attack.

The existing IPsec Security Architecture and protocol suite [10] offers protection from these threats. An FCIP Entity MUST implement portions of the IPsec protocol suite as described in this section.

9.2. FC Fabric and IP Network Deployment Models

In the context of enabling a secure FCIP tunnel between FC SANs, the following characteristics of the IP Network deployment are useful to note.

1) The FCIP Entities share a peer-to-peer relationship. Therefore, the administration of security policies applies to all FCIP Entities in an equal manner. This differs from a true Client-Server relationship, where there is an inherent difference in how security policies are administered.

2) Policy administration as well as security deployment and configuration are constrained to the set of FCIP Entities, thereby posing less of a requirement on a scalable mechanism. For example, the validation of credentials can be relaxed to the point where deploying a set of pre-shared keys is a viable technique.

3) TCP Connections and the IP Network are terminated at the FCIP Entity. The granularity of security implementation is at the level of the FCIP tunnel endpoint (or FCIP Entity), unlike other applications where there is a user-level termination of TCP Connections. User-level objects are not controllable by or visible to FCIP Entities. All user-level security related to FCIP is the responsibility of the Fibre Channel standards and is outside the scope of this specification.

4) When an FCIP Entity is deployed, its IP addresses will typically be statically assigned. However, support for dynamic IP address assignment, as described in [33], while typically not required, cannot be ruled out.

9.3. FCIP Security Components

FCIP Security compliant implementations MUST implement ESP and the IPsec protocol suite based cryptographic authentication and data integrity [10], as well as confidentiality using algorithms and transforms as described in this section. Also, FCIP implementations MUST meet the secure key management requirements of IPsec protocol suite.

9.3.1. IPsec ESP Authentication and Confidentiality

FCIP Entities MUST implement IPsec ESP [12] in Tunnel Mode for providing Data Integrity and Confidentiality. FCIP Entities MAY implement IPsec ESP in Transport Mode, if deployment considerations require use of Transport Mode. When ESP is utilized, per-packet data origin authentication, integrity, and replay protection MUST be used.

If Confidentiality is not enabled but Data Integrity is enabled, ESP with NULL Encryption [15] MUST be used.

IPsec ESP for message authentication computes a cryptographic hash over the payload that is protected. While IPsec ESP mandates compliant implementations to support certain algorithms for deriving this hash, FCIP implementations:

- MUST implement HMAC with SHA-1 [11]
- SHOULD implement AES in CBC MAC mode with XCBC extensions [23]
- DES in CBC mode SHOULD NOT be used due to inherent weaknesses

For ESP Confidentiality, FCIP Entities:

- MUST implement 3DES in CBC mode [16]
- SHOULD implement AES in CTR mode [22]
- MUST implement NULL Encryption [15]

9.3.2. Key Management

FCIP Entities MUST support IKE [14] for peer authentication, negotiation of Security Associations (SA), and Key Management using the IPsec DOI [13]. Manual keying SHALL NOT be used for establishing an SA since it does not provide the necessary elements for rekeying (see section 9.3.3). Conformant FCIP implementations MUST support peer authentication using pre-shared keys and MAY support peer authentication using digital certificates. Peer authentication using public key encryption methods outlined in IKE [14] sections 5.2 and 5.3 SHOULD NOT be used.

IKE Phase 1 establishes a secure, MAC-authenticated channel for communications for use by IKE Phase 2. FCIP implementations MUST support IKE Main Mode and SHOULD support Aggressive Mode.

IKE Phase 1 exchanges MUST explicitly carry the Identification Payload fields (IDii and IDir). Conformant FCIP implementations MUST use ID_IPV4_ADDR, ID_IPV6_ADDR (if the protocol stack supports IPv6), or ID_FQDN Identification Type values. The ID_USER_FQDN, IP Subnet, IP Address Range, ID_DER_ASN1_DN, and ID_DER_ASN1_GN Identification Type values SHOULD NOT be used. The ID_KEY_ID Identification Type values MUST NOT be used. As described in [13], the port and protocol fields in the Identification Payload MUST be set to zero or UDP port 500.

FCIP Entities negotiate parameters for SA during IKE Phase 2 only using "Quick Mode". For FCIP Entities engaged in IKE "Quick Mode", there is no requirement for PFS (Perfect Forward Secrecy). FCIP implementations MUST use either ID_IPV4_ADDR or ID_IPV6_ADDR Identification Type values (based on the version of IP supported). Other Identification Type values MUST NOT be used.

Since the number of Phase 2 SAs may be limited, Phase 2 delete messages may be sent for idle SAs. The receipt of a Phase 2 delete message SHOULD NOT be interpreted as a reason for tearing down an FCIP Link or any of its TCP connections. When there is new activity on that idle link, a new Phase 2 SA MUST be re-established.

For a given pair of FCIP Entities, the same IKE Phase 1 negotiation can be used for all Phase 2 negotiations; i.e., all TCP Connections that are bundled into the single FCIP Link can share the same Phase 1 results.

Repeated rekeying using "Quick Mode" on the same shared secret will reduce the cryptographic properties of that secret over time. To overcome this, Phase 1 SHOULD be invoked periodically to create a new set of IKE shared secrets and related security parameters.

IKE Phase 1 establishment requires the following key distribution and FCIP Entities:

- MUST support pre-shared IKE keys.
- MAY support certificate-based peer authentication using digital signatures.
- SHOULD NOT use peer authentication using the public key encryption methods outlined in sections 5.2 and 5.3 of [14].

When pre-shared keys are used, IKE Main Mode is usable only when both peers of an FCIP Link use statically assigned IP addresses. When support for dynamically assigned IP Addresses is attempted in conjunction with Main Mode, use of group pre-shared keys would be forced, and the use of group pre-shared keys in combination with Main Mode is not recommended as it exposes the deployed environment to man-in-the-middle attacks. Therefore, if either peer of an FCIP Link uses dynamically assigned addresses, Aggressive Mode SHOULD be used and Main Mode SHOULD NOT be used.

When Digital Signatures are used, either IKE Main Mode or IKE Aggressive Mode may be used. In all cases, access to locally stored secret information (pre-shared key, or private key for digital signing) MUST be suitably restricted, since compromise of secret information nullifies the security properties of IKE/IPsec protocols. Such mechanisms are outside the scope of this document. Support for IKE Oakley Groups [27] is not required.

For the purpose of establishing a secure FCIP Link, the two participating FCIP Entities consult a Security Policy Database (SPD). The SPD is described in IPsec [10] Section 4.4.1. FCIP Entities may have more than one interface and IP Address, and it is possible for an FCIP Link to contain multiple TCP connections whose FCIP endpoint IP Addresses are different. In this case, an IKE Phase 1 SA is established for each FCIP endpoint IP Address pair. Within IKE Phase 1, FCIP implementations must support the ID_IPV4_ADDR, ID_IPV6_ADDR (if the protocol stack supports IPv6), and ID_FQDN Identity Payloads. If FCIP Endpoint addresses are dynamically assigned, it may be beneficial to use ID_FQDN, and for this reason, IP_FQDN Identity Payload MUST be supported. Other identity payloads (ID_USER_FQDN, ID_DER_ASN1_GN, ID_KEY_ID) SHOULD NOT be used.

At the end of successful IKE negotiations both FCIP Entities store the SA parameters in their SA database (SAD). The SAD is described in IPsec [10] Section 4.4.3. The SAD contains the set of active SA entries, each entry containing Sequence Counter Overflow, Sequence Number Counter, Anti-replay Window, and the Lifetime of the SA. FCIP Entities SHALL employ a default SA Lifetime of one hour and a default Anti-replay window of 32 sequence numbers.

When a TCP Connection is established between two FCIP_DEs, two unidirectional SAs are created for that connection and each SA is identified in the form of a Security Parameter Index (SPI). One SA is associated with the incoming traffic flow and the other SA is associated with the outgoing traffic flow. The FCIP_DEs at each end of the TCP connection MUST maintain the SPIs for both its incoming and outgoing FCIP Encapsulated Frames.

FCIP Entities MAY provide administrative management of Confidentiality usage. These management interfaces SHOULD be provided in a secure manner, so as to prevent an attacker from subverting the security process by attacking the management interface.

9.3.3. ESP Replay Protection and Rekeying Issues

FCIP Entities MUST implement Replay Protection against ESP Sequence Number wrap, as described in [14]. In addition, based on the cipher algorithm and the number of bits in the cipher block size, the validity of the key may become compromised. In both cases, the SA needs to be re-established.

FCIP Entities MUST use the results of IKE Phase 1 negotiation for initiating an IKE Phase 2 "Quick Mode" exchange and establish new SAs.

To enable smooth transition of SAs, it is RECOMMENDED that both FCIP Entities refresh the SPI when the sequence number counter reaches 2^31 (i.e., half the sequence number space). It also is RECOMMENDED that the receiver operate with multiple SPIs for the same TCP Connection for a period of 2^31 sequence number packets before aging out an SPI.

When a new SPI is created for the outgoing direction, the sending side SHALL begin using it for all new FCIP Encapsulated Frames. Frames that are either in-flight, or re-sent due to TCP retransmissions, etc. MAY use either the new SPI or the one being replaced.

9.4. Secure FCIP Link Operation

9.4.1. FCIP Link Initialization Steps

FCIP implementations may allow enabling and disabling security mechanisms at the granularity of an FCIP Link. If enabled, the following FCIP Link Initialization steps MUST be followed.

When an FCIP Link is initialized, before any FCIP TCP Connections are established, the local SPD is consulted to determine if IKE Phase 1 has been completed with the FCIP Entity in the peer FCIP Entity, as identified by the WWN.

If Phase 1 is already completed, IKE Phase 2 proceeds. Otherwise, IKE Phase 1 MUST be completed before IKE Phase 2 can start. Both IKE Phase 1 and Phase 2 transactions use UDP Port 500. If IKE Phase 1 fails, the FCIP Link initialization terminates and notifies the FC entity with the reason for the termination. Otherwise, the FCIP Link initialization moves to TCP Connection Initialization.

As described in section 8.1, FCIP Entities exchange an FSF for forming an FCIP Link. The use of ESP Confidentiality is an effective countermeasure against any perceived security risks of FSF.

9.4.2. TCP Connection Security Associations (SAs)

Each TCP connection MUST be protected by an IKE Phase 2 SA. Traffic from one or more than one TCP connection may flow within each IPsec Phase 2 SA. While it is possible for an IKE Phase 2 SA to protect multiple TCP connections, all packets of a TCP connection are protected using only one IKE Phase 2 SA.

If different Quality of Service settings are applied to TCP connections, it is advisable to use a different IPsec SA for these connections. Attempting to apply a different quality of service to connections handled by the same IPsec SA can result in reordering, and falling outside the replay window. For additional details, see [21].

FCIP implementations need not verify that the IP addresses and port numbers in the packet match any locally stored per-connection values, leaving this check to be performed by the IPsec layer.

An implementation is free to perform several IKE Phase 2 negotiations and cache them in its local SPIs, although entries in such a cache can be flushed per current SA Lifetime settings.

9.4.3. Handling Data Integrity and Confidentiality Violations

Upon datagram reception, when the ESP packet fails an integrity check, the receiver MUST drop the datagram, which will trigger TCP retransmission. If many such datagrams are dropped, a receiving FCIP Entity MAY close the TCP Connection and notify the FC Entity with the reason for the closure.

An implementation SHOULD follow guidelines for auditing all auditable ESP events per IPsec [10] Section 7.

Integrity checks MUST be performed if Confidentiality is enabled.

10. Performance

10.1. Performance Considerations

Traditionally, the links between FC Fabric components have been characterized by low latency and high throughput. The purpose of FCIP is to provide functionality equivalent to these links using an IP Network, where low latency and high throughput are not as certain. It follows that FCIP Entities and their counterpart FC Entities probably will be interested in optimal use of the IP Network.

Many options exist for ensuring high throughput and low latency appropriate for the distances involved in an IP Network. For example, a private IP Network might be constructed for the sole use of FCIP Entities. The options that are within the scope of this specification are discussed here.

One option for increasing the probability that FCIP data streams will experience low latency and high throughput is the IP QoS techniques discussed in section 10.2. This option can have value when applied to a single TCP Connection. Depending on the sophistication of the FC Entity, further value may be obtained by having multiple TCP Connections with differing QoS characteristics.

There are many reasons why an FC Entity might request the creation of multiple TCP Connections within an FCIP_LEP. These reasons include a desire to provide differentiated services for different TCP data connections between FCIP_LEPs, or a preference to separately queue different streams of traffic not having a common in-order delivery requirement.

At the time a new TCP Connection is created, the FC Entity SHALL specify to the FCIP Entity the QoS characteristics (including but not limited to IP per-hop-behavior) to be used for the lifetime of that connection. This MAY be achieved by having:

a) only one set of QoS characteristics for all TCP Connections;

b) a default set of QoS characteristics that the FCIP Entity applies in the absence of differing instructions from the FC Entity; or

c) a sophisticated mechanism for exchanging QoS requirements information between the FC Entity and FCIP Entity each time a new TCP Connection is created.

Once established, the QoS characteristics of a TCP Connection SHALL NOT be changed, since this specification provides no mechanism for the FC Entity to control such changes. The mechanism for providing different QoS characteristics in FCIP is the establishment of a different TCP Connections and associated FCIP_DEs.

When FCIP is used with a network with a large (bandwidth*delay) product, it is RECOMMENDED that FCIP_LEPs use the TCP mechanisms (window scaling and wrapped sequence protection) for Long Fat Networks (LFNs) as defined in RFC 1323 [24].

10.2. IP Quality of Service (QoS) Support

Many methods of providing QoS have been devised or proposed. These include (but are not limited to) the following:

- Multi-Protocol Label Switching (MPLS) -- RFC 3031 [32]
- Differentiated Services Architecture (diffserv) -- RFC 2474 [28], RFC 2475 [29], RFC 2597 [30], and RFC 2598 [31] -- and other forms of per-hop-behavior (PHB)
- Integrated Services, RFC 1633 [25]
- IEEE 802.1p

The purpose of this specification is not to specify any particular form of IP QoS, but rather to specify only those issues that must be addressed in order to maximize interoperability between FCIP equipment that has been manufactured by different vendors.

It is RECOMMENDED that some form of preferential QoS be used for FCIP traffic to minimize latency and packet drops. No particular form of QoS is recommended.

If a PHB IP QoS is implemented, it is RECOMMENDED that it interoperate with diffserv (see RFC 2474 [28], RFC 2475 [29], RFC 2597 [30], and RFC 2598 [31]).

If no form of preferential QoS is implemented, the DSCP field SHOULD be set to '000000' to avoid negative impacts on other network components and services that may be caused by uncontrolled usage of non-zero values of the DSCP field.

11. References

11.1. Normative References

The references in this section were current as of the time this specification was approved. This specification is intended to operate with newer versions of the referenced documents and looking for newer reference documents is recommended.

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

[2] Fibre Channel Backbone (FC-BB), ANSI INCITS.342:2001, December 12, 2001.

[3] Fibre Channel Backbone -2 (FC-BB-2), ANSI INCITS.372:2003, July 25, 2003.

[4] Fibre Channel Switch Fabric -2 (FC-SW-2), ANSI INCITS.355:2001, December 12, 2001.

[5] Fibre Channel Framing and Signaling (FC-FS), ANSI INCITS.373:2003, October 27, 2003.

[6] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, September 1981.

[7] Braden, R., "Requirements for Internet Hosts -- Communication Layers", STD 3, RFC 1122, October 1989.

[8] Jacobson, V., Braden, R. and D. Borman, "TCP Extensions for High Performance", RFC 1323, May 1992.

[9] Eastlake, D., Crocker, S. and J. Schiller, "Randomness Recommendations for Security", RFC 1750, December 1994.

[10] Kent, S. and R. Atkinson, "Security Architecture for the Internet Protocol", RFC 2401, November 1998.

[11] Krawczyk, H., Bellare, M. and R. Canetti, "HMAC: Keyed- Hashing for Message Authentication", RFC 2104, February 1997.

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

[13] Piper, D., "The Internet IP Security Domain of Interpretation of ISAKMP", RFC 2407, November 1998.

[14] Harkins, D. and D. Carrel, "The Internet Key Exchange (IKE)", RFC 2409, November 1998.

[15] Glenn, R. and S. Kent, "The NULL Encryption Algorithm and Its Use With IPsec", RFC 2410, November 1998.

[16] Pereira, R. and R. Adams, "The ESP CBC-Mode Cipher Algorithms", RFC 2451, November 1998.

[17] Guttman, E., Perkins, C., Veizades, J. and M. Day, "Service Location Protocol, version 2", RFC 2608, July 1999.

[18] Floyd, S., Mahdavi, J., Mathis, M. and M. Podolsky, "SACK Extension", RFC 2883, July 2000.

[19] Weber, R., Rajagopal, M., Travostino, F., O'Donnell, M., Monia, C. and M. Merhar, "Fibre Channel (FC) Frame Encapsulation", RFC 3643, December 2003.

[20] Peterson, D., "Finding Fibre Channel over TCP/IP (FCIP) Entities Using Service Location Protocol version 2 (SLPv2)", RFC 3822, July 2004.

[21] Aboba, B., Tseng, J., Walker, J., Rangan, V. and F. Travostino, "Securing Block Storage Protocols over IP", RFC 3723, April 2004.

[22] Frankel, S., Glenn, R. and S. Kelly, "The AES-CBC Cipher Algorithm and Its Use with IPsec", RFC 3602, September 2003.

[23] Frankel, S. and H. Herbert, "The AES-XCBC-MAC-96 Algorithm and Its Use With IPsec", RFC 3566, September 2003.

11.2. Informative References

[24] Jacobson, V., Braden, R. and D. Borman, "TCP Extensions for High Performance", RFC 1323, May 1992.

[25] Braden, R., Clark, D. and S. Shenker, "Integrated Services in the Internet Architecture: an Overview", RFC 1633, June 1994.

[26] Mills, D., "Simple Network Time Protocol (SNTP) Version 4 for IPv4, IPv6 and OSI", RFC 2030, October 1996.

[27] Orman, H., "The OAKLEY Key Determination Protocol", RFC 2412, November 1998.

[28] Nichols, K., Blake, S., Baker, F. and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and Ipv6 Headers", RFC 2474, December 1998.

[29] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z. and W. Weiss, "An Architecture for Differentiated Services", RFC 2475, December 1998.

[30] Heinanen, J., Baker, F., Weiss, W. and J. Wroclawski, "An Assured Forwarding PHB", RFC 2597, June 1999.

[31] Jacobson, V., Nichols, K. and K. Poduri, "An Expedited Forwarding PHB Group", RFC 2598, June 1999.

[32] Rosen, E., Viswanathan, A. and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, January, 2001.

[33] Patel, B., Aboba, B., Kelly, S. and V. Gupta, "Dynamic Host Configuration Protocol (DHCPv4) Configuration of IPsec Tunnel Mode", RFC 3456, January 2003.

[34] Kembel, R., "The Fibre Channel Consultant: A Comprehensive Introduction", Northwest Learning Associates, 1998.

12. Acknowledgments

The developers of this specification thank Mr. Jim Nelson for his assistance with FC-FS related issues.

The developers of this specification express their appreciation to Mr. Mallikarjun Chadalapaka and Mr. David Black for their detailed and helpful reviews.

Appendix A - Fibre Channel Bit and Byte Numbering Guidance

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

Fibre Channel bit and byte numbering can be observed if the data structure heading, shown in figure 11, is cut and pasted at the top of figure 7, figure 9, and figure 17.

```
W|------------------------------Bit------------------------------|
o|                                                               |
r|3 3 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1                     |
d|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|
```

Figure 11: Fibre Channel Data Structure Bit and Byte Numbering

Fibre Channel bit numbering for the pFlags field can be observed if the data structure heading, shown in figure 12, is cut and pasted at the top of figure 8.

```
|----------------Bit--------------------|
|                                       |
| 31   30   29   28   27   26   25   24 |
```

Figure 12: Fibre Channel pFlags Bit Numbering

Fibre Channel bit numbering for the Connection Usage Flags field can be observed if the data structure heading, shown in figure 13, is cut and pasted at the top of figure 10.

```
|------------------------------Bit------------------------------|
|                                                               |
|   31      30      29      28      27      26      25      24  |
```

Figure 13: Fibre Channel Connection Usage Flags Bit Numbering

Appendix B - IANA Considerations

IANA has made the following port assignments to FCIP:

- fcip-port 3225/tcp FCIP
- fcip-port 3225/udp FCIP

IANA has changed the authority for these port allocations to reference this RFC.

Use of UDP with FCIP is prohibited even though IANA has allocated a port.

The FC Frame encapsulation used by this specification employs Protocol# value 1, as described in the IANA Considerations appendix of the FC Frame Encapsulation [19] specification.

Appendix C - FCIP Usage of Addresses and Identifiers

In support of network address translators, FCIP does not use IP Addresses to identify FCIP Entities or FCIP_LEPs. The only use of IP Addresses for identification occurs when initiating new TCP connect requests (see section 8.1.2.3) where the IP Address destination of the TCP connect request is used to answer the question: "Have previous TCP connect requests been made to the same destination FCIP Entity?" The correctness of this assumption is further checked by sending the Destination FC Fabric Entity World Wide Name in the FCIP Special Frame (FSF) and having the value checked by the FCIP Entity that receives the TCP connect request and FSF (see section 8.1.3).

For the purposes of processing incoming TCP connect requests, the source FCIP Entity is identified by the Source FC Fabric Entity World Wide Name and Source FC/FCIP Entity Identifier fields in the FSF sent from the TCP connect requestor to the TCP connect recipient as the first bytes following the TCP connect request (see section 8.1.2.3 and section 8.1.3).

FC-BB-2 [3] provides the definitions for each of the following FSF fields:

- Source FC Fabric Entity World Wide Name,
- Source FC/FCIP Entity Identifier, and
- Destination FC Fabric Entity World Wide Name.

As described in section 8.1.3, FCIP Entities segregate their FCIP_LEPs between:

- Connections resulting from TCP connect requests initiated by the FCIP Entity, and

- Connections resulting from TCP connect requests received by the FCIP Entity.

Within each of these two groups, the following information is used to further identify each FCIP_LEP:

- Source FC Fabric Entity World Wide Name,
- Source FC/FCIP Entity Identifier, and
- Destination FC Fabric Entity World Wide Name.

Appendix D - Example of Synchronization Recovery Algorithm

The contents of this annex are informative.

Synchronization may be recovered as specified in section 5.6.2.3. An example of an algorithm for searching the bytes delivered to the Encapsulated Frame Receiver Portal for a valid FCIP Frame header is provided in this annex.

This resynchronization uses the principle that a valid FCIP data stream must contain at least one valid header every 2176 bytes (the maximum length of an encapsulated FC Frame). Although other data patterns containing apparently valid headers may be contained in the stream, the FC CRC or FCIP Frame validity of the data patterns contained in the data stream will always be either interrupted by or resynchronized with the valid FCIP Frame headers.

Consider the case shown in figure 14. A series of short FCIP Frames, perhaps from a trace, are embedded in larger FCIP Frames, say as a result of a trace file being transferred from one disk to another. The headers for the short FCIP Frames are denoted SFH and the long FCIP Frame headers are marked as LFH.

```
+-+--+-+----+-+----+-+----+-+-+-+---+-+---
|L|  |S|    |S|    |S|    |S| |L|   |S|
|F|  |F|    |F|    |F|    |F| |F|   |F|...
|H|  |H|    |H|    |H|    |H| |H|   |H|
+-+--+-+----+-+----+-+----+-+-+-+---+-+---
|                            |
|<---------2176 bytes-------->|
```

Figure 14: Example of resynchronization data stream

A resynchronization attempt that starts just to the right of an LFH will find several SFH FCIP Frames before discovering that they do not represent the transmitted stream of FCIP Frames. Within 2176 bytes plus or minus, however, the resynchronization attempt will encounter an SFH whose length does not match up with the next SFH because the LFH will fall in the middle of the short FCIP Frame pushing the next header farther out in the byte stream.

Note that the resynchronization algorithm cannot forward any prospective FC Frames to the FC Frame Transmitter Portal because, until synchronization is completely established, there is no certainty that anything that looked like an FCIP Frame really was one. For example, an SFH might fortuitously contain a length that points exactly to the beginning of an LFH. The LFH would identify the correct beginning of a transmitted FCIP Frame, but that in no way guarantees that the SFH was also a correct FCIP Frame header.

There exist some data streams that cannot be resynchronized by this algorithm. If such a data stream is encountered, the algorithm causes the TCP Connection to be closed.

The resynchronization assumes that security and authentication procedures outside the FCIP Entity are protecting the valid data stream from being replaced by an intruding data stream containing valid FCIP data.

The following steps are one example of how an FCIP_DE might resynchronize with the data stream entering the Encapsulated Frame Receiver Portal.

1) Search for candidate and strong headers:

   The data stream entering the Encapsulated Frame Receiver Portal is searched for 12 bytes in a row containing the required values for:

   a) Protocol field,
   b) Version field,
   c) ones complement of the Protocol field,
   d) ones complement of the Version field,
   e) replication of encapsulation word 0 in word 1, and
   f) pFlags field and its ones complement.

   If such a 12-byte grouping is found, the FCIP_DE assumes that it has identified bytes 0-2 of a candidate FCIP encapsulation header.

   All bytes up to and including the candidate header byte are discarded.

   If no candidate header has been found after searching a specified number of bytes greater than some multiple of 2176 (the maximum length of an FCIP Frame), resynchronization has failed and the TCP/IP connection is closed.

   Word 3 of the candidate header contains the Frame Length and Flags fields and their ones complements. If the fields are consistent with their ones complements, the candidate header is considered a strong candidate header. The Frame Length field is used to determine where in the byte stream the next strong candidate header should be and processing continues at step 2).

2) Use multiple strong candidate headers to locate a verified candidate header:

   The Frame Length in one strong candidate header is used to skip incoming bytes until the expected location of the next strong candidate header is reached. Then the tests described in step 1) are applied to see if another strong candidate header has successfully been located.

   All bytes skipped and all bytes in all strong candidate headers processed are discarded.

   Strong candidate headers continue to be verified in this way for at least 4352 bytes (twice the maximum length of an FCIP Frame). If at any time a verification test fails, processing restarts at step 1 and a retry counter is incremented. If the retry counter exceeds 3 retries, resynchronization has failed and the TCP Connection is closed, and the FC entity is notified with the reason for the closure.

   After strong candidate headers have been verified for at least 4352 bytes, the next header identified is a verified candidate header, and processing continues at step 3).

   Note: If a strong candidate header was part of the data content of an FCIP Frame, the FCIP Frame defined by that or a subsequent strong candidate header will eventually cross an actual header in the byte stream. As a result it will either identify the actual header as a strong candidate header or it will lose synchronization again because of the extra 28 bytes in the length, returning to step 1 as described above.

3) Use multiple strong candidate headers to locate a verified candidate header:

   Incoming bytes are inspected and discarded until the next verified candidate header is reached. Inspection of the incoming bytes includes testing for other candidate headers using the criteria described in step 1. Each verified candidate header is tested against the tests listed in section 5.6.2.2 as would normally be the case.

   Verified candidate headers continue to be located and tested in this way for a minimum of 4352 bytes (twice the maximum length of an FCIP Frame). If all verified candidate headers encountered are valid, the last verified candidate header is a valid header. At this point the FCIP_DE stops discarding bytes and begins normal

FCIP de-encapsulation, including for the first time since synchronization was lost, delivery of FC Frames through the FC Frame Transmitter Portal according to normal FCIP rules.

If any verified candidate headers are invalid but meet all the requirements of a strong candidate header, increment the retry counter and return to step 2). If any verified candidate headers are invalid and fail to meet the tests for a strong candidate header, or if inspection of the bytes between verified candidate headers discovers any candidate headers, increment the retry counter and return to step 1. If the retry counter exceeds 4 retries, resynchronization has failed and the TCP/IP connection is closed.

A flowchart for this algorithm can be found in figure 15.

```
                        Synchronization is lost
                                 |
                    _____________v_______________
                   |                             |
                   | Search for candidate header |
      +----------->|                             |
      |            |   Found           Not Found |
      |            | (Strong candidate)          |
      |            |_____________________________|
      |                    |              |
      |                    |              + --------->close TCP
      |             _______v_____________________     Connection
      |            |                             |    and notify
      |            |   Enough strong candidate   |    the FC Entity
      |      +---->|     headers identified?     |    with the reason
      |      |     |                             |    for closure
      |      |     |     No               Yes    |
      |      |     |        (Verified candidate) |
      |      |     |_____________________________|
      |___________________|                |
      ^      |                             |
      |      |                             |
      |      |      _______________________v_____
      |      |     |                             |
      |      |     | Enough verified candidate   |
      |      |     |   headers validated?        |
      |      |     |                             |
      |      |     |     No               Yes    |
      |      |     |            (Resynchronized) |
      |      |     |_____________________________|
      |      |            |                |
      |      |      ______v__________      |      Resume
      |      |     |                 |     + ---> Normal
      |      |     | Synchronization |            De-encapsulation
      |      |     |      Lost?      |
      |      |     |                 |
      |      |     | No          Yes |
      |      |     |_________________|
      |      |        |           |
      |      |________|           |
      |___________________________|
```

Figure 15: Flow diagram of simple synchronization example

Appendix E - Relationship between FCIP and IP over FC (IPFC)

The contents of this annex are informative.

IPFC (RFC 2625) describes the encapsulation of IP packets in FC Frames. It is intended to facilitate IP communication over an FC network.

FCIP describes the encapsulation of FC Frames in TCP segments, which in turn are encapsulated inside IP packets for transporting over an IP network. It gives no consideration to the type of FC Frame that is being encapsulated. Therefore, the FC Frame may actually contain an IP packet as described in the IP over FC specification (RFC 2625). In such a case, the data packet would have:

- Data Link Header
- IP Header
- TCP Header
- FCIP Header
- FC Header
- IP Header

Note: The two IP headers would not be identical to each other. One would have information pertaining to the final destination, while the other would have information pertaining to the FCIP Entity.

The two documents focus on different objectives. As mentioned above, implementation of FCIP will lead to IP encapsulation within IP. While perhaps inefficient, this should not lead to issues with IP communication. One caveat: if a Fibre Channel device is encapsulating IP packets in an FC Frame (e.g., an IPFC device), and that device is communicating with a device running IP over a non-FC medium, a second IPFC device may need to act as a gateway between the two networks. This scenario is not specifically addressed by FCIP.

There is nothing in either of the specifications to prevent a single device from implementing both FCIP and IP-over-FC (IPFC), but this is implementation specific, and is beyond the scope of this document.

Appendix F - FC Frame Format

Note: All users of the words "character" or "characters" in this section refer to 8bit/10bit link encoding wherein each 8 bit "character" within a link frame is encoded as a 10 bit "character" for link transmission. These words do not refer to ASCII, Unicode, or any other form of text characters, although octets from such characters will occur as 8 bit "characters" for this encoding. This usage is employed here for consistency with the ANSI T11 standards that specify Fibre Channel.

The contents of this annex are informative.

All FC Frames have a standard format (see FC-FS [5]) much like LAN's 802.x protocols. However, the exact size of each FC Frame varies depending on the size of the variable fields. The size of the variable field ranges from 0 to 2112-bytes as shown in the FC Frame Format in figure 16, resulting in the minimum size FC Frame of 36 bytes and the maximum size FC Frame of 2148 bytes. Valid FC Frame lengths are always a multiple of four bytes.

```
+------+--------+-----------+----//-------+------+------+
| SOF  |Frame   |Optional   |  Frame      | CRC  |  EOF |
| (4B) |Header  |Header     | Payload     | (4B) | (4B) |
|      |(24B)   |<----------------------->|      |      |
|      |        | Data Field = (0-2112B)  |      |      |
+------+--------+-----------+----//-------+------+------+
```

Figure 16: FC Frame Format

SOF and EOF Delimiters

On an FC link, Start-of-Frame (SOF) and End-Of-Frame (EOF) are called Ordered Sets and are sent as special words constructed from the 8B/10B comma character (K28.5) followed by three additional 8B/10B data characters making them uniquely identifiable in the data stream.

On an FC link, the SOF delimiter serves to identify the beginning of an FC Frame and prepares the receiver for FC Frame reception. The SOF contains information about the FC Frame's Class of Service, position within a sequence, and in some cases, connection status.

The EOF delimiter identifies the end of the FC Frame and the final FC Frame of a sequence. In addition, it serves to force the running disparity to negative. The EOF is used to end the connection in connection-oriented classes of service.

A special EOF delimiter called EOFa (End Of Frame - Abort) is used to terminate a partial FC Frame resulting from a malfunction in a link facility during transmission. Since an FCIP Entity functions like a transmission link with respect to the rest of the FC Fabric, FCIP_DEs may use EOFa in their error recovery procedures.

It is therefore important to preserve the information conveyed by the delimiters across the IP-based network, so that the receiving FCIP Entity can correctly reconstruct the FC Frame in its original SOF and EOF format before forwarding it to its ultimate FC destination on the FC link.

When an FC Frame is encapsulated and sent over a byte-oriented interface, the SOF and EOF delimiters are represented as sequences of four consecutive bytes, which carry the equivalent Class of Service and FC Frame termination information as the FC ordered sets.

The representation of SOF and EOF in an encapsulation FC Frame is described in FC Frame Encapsulation [19].

Frame Header

The FC Frame Header is transparent to the FCIP Entity. The FC Frame Header is 24 bytes long and has several fields that are associated with the identification and control of the payload. Current FC Standards allow up to 3 Optional Header fields [5]:

- Network_Header (16-bytes)
- Association_Header (32-bytes)
- Device_Header (up to 64-bytes).

Frame Payload

The FC Frame Payload is transparent to the FCIP Entity. An FC application level payload is called an Information Unit at the FC-4 Level. This is mapped into the FC Frame Payload of the FC Frame. A large Information Unit is segmented using a structure consisting of FC Sequences. Typically, a Sequence consists of more than one FC Frame. FCIP does not maintain any state information regarding the relationship of FC Frames within an FC Sequence.

CRC

The FC CRC is 4 bytes long and uses the same 32-bit polynomial used in FDDI and is specified in ANSI X3.139 Fiber Distributed Data Interface. This CRC value is calculated over the entire FC header and the FC payload; it does not include the SOF and EOF delimiters.

Note: When FC Frames are encapsulated into FCIP Frames, the FC Frame CRC is untouched by the FCIP Entity.

Appendix G - FC Encapsulation Format

This annex contains a reproduction of the FC Encapsulation Format [19] as it applies to FCIP Frames that encapsulate FC Frames. The information in this annex is not intended to represent the FCIP Special Frame (FSF) that is described in section 7.

The information in this annex was correct as of the time this specification was approved. The information in this annex is informative only.

If there are any differences between the information here and the FC Encapsulation Format specification [19], the FC Encapsulation Format specification takes precedence.

If there are any differences between the information here and the contents of section 5.6.1, then the contents of section 5.6.1 take precedence.

Figure 17 applies the requirements stated in section 5.6.1 and in the FC Encapsulation Frame format resulting in a summary of the FC Frame format. Where FCIP requires specific values, those values are shown in hexadecimal in parentheses. Detailed requirements for the FCIP usage of the FC Encapsulation Format are in section 5.6.1.

```
     W|------------------------------Bit------------------------------|
     o|                                                               |
     r|                    1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3|
     d|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|   Protocol#   |    Version    |  -Protocol#   |   -Version    |
      |    (0x01)     |    (0x01)     |    (0xFE)     |    (0xFE)     |
      +---------------+---------------+---------------+---------------+
     1|   Protocol#   |    Version    |  -Protocol#   |   -Version    |
      |    (0x01)     |    (0x01)     |    (0xFE)     |    (0xFE)     |
      +---------------+---------------+---------------+---------------+
     2|    pFlags     |   Reserved    |    -pFlags    |  -Reserved    |
      |     (0x00)    |    (0x00)     |    (0xFF)     |    (0xFF)     |
      +-----------+---+---------------+-----------+---+---------------+
     3|   Flags   |   Frame Length    |   -Flags  |  -Frame Length    |
      |  (0x00)   |                   |   (0x3F)  |                   |
      +-----------+-------------------+-----------+-------------------+
     4|                     Time Stamp [integer]                      |
      +---------------------------------------------------------------+
     5|                     Time Stamp [fraction]                     |
      +---------------------------------------------------------------+
     6|                     CRC (Reserved in FCIP)                    |
      |                        (0x00-00-00-00)                        |
      +---------------+---------------+---------------+---------------+
     7|      SOF      |      SOF      |     -SOF      |     -SOF      |
      +---------------+---------------+---------------+---------------+
     8|                                                               |
      +-----            FC Frame content (see appendix F)        -----+
      |                                                               |
      +---------------+---------------+---------------+---------------+
     n|      EOF      |      EOF      |     -EOF      |     -EOF      |
      +---------------+---------------+---------------+---------------+
```

Figure 17: FCIP Frame Format

The names of fields are generally descriptive on their contents and the FC Encapsulation Format specification [19] is referenced for details. Field names preceded by a minus sign are ones complement values of the named field.

Note: Figure 17 does not represent the FSF that is described in section 7.

Appendix H - FCIP Requirements on an FC Entity

The contents of this annex are informative for FCIP but might be considered normative on FC-BB-2.

The capabilities that FCIP requires of an FC Entity include:

1) The FC Entity must deliver FC Frames to the correct FCIP Data Engine (in the correct FCIP Link Endpoint).

2) Each FC Frame delivered to an FCIP_DE must be accompanied by a time value synchronized with the clock maintained by the FC Entity at the other end of the FCIP Link (see section 6). If a synchronized time value is not available, a value of zero must accompany the FC Frame.

3) When FC Frames exit FCIP Data Engine(s) via the FC Frame Transmitter Portal(s), the FC Entity should forward them to the FC Fabric. However, before forwarding an FC Frame, the FC Entity must compute the end-to-end transit time for the FC Frame using the time value supplied by the FCIP_DE (taken from the FCIP header) and a synchronized time value (see section 6). If the end-to-end transit time exceeds the requirements of the FC Fabric, the FC Entity is responsible for discarding the FC Frame.

4) The only delivery ordering guarantee provided by FCIP is correctly ordered delivery of FC Frames between a pair of FCIP Data Engines. FCIP expects the FC Entity to implement all other FC Frame delivery ordering requirements.

5) When a TCP connect request is received and that request would add a new TCP Connection to an existing FCIP_LEP, the FC Entity must authenticate the source of the TCP connect request before use of the new TCP connection is allowed.

6) The FC Entity may participate in determining allowed TCP Connections, TCP Connection parameters, quality of service usage, and security usage by modifying interactions with the FCIP Entity that are modelled as a "shared" database in section 8.1.1.

7) The FC Entity may require the FCIP Entity to perform TCP close requests.

8) The FC Entity may recover from connection failures.

9) The FC Entity must recover from events that the FCIP Entity cannot handle, such as:

a) loss of synchronization with FCIP Frame headers from the Encapsulated Frame Receiver Portal requiring resetting the TCP Connection; and

b) recovering from FCIP Frames that are discarded as a result of synchronization problems (see section 5.6.2.2 and section 5.6.2.3).

10) The FC Entity must work cooperatively with the FCIP Entity to manage flow control problems in either the IP Network or FC Fabric.

11) The FC Entity may test for failed TCP Connections.

Note that the Fibre Channel standards must be consulted for a complete understanding of the requirements placed on an FC Entity.

Table 2 shows the explicit interactions between the FCIP Entity and the FC Entity.

Reference Section	Condition	Information/Parameter Passed and Direction	
		FCIP Entity--->	<---FC Entity
5.6 FCIP Data Engine	FC Frame ready for IP transfer		Provide FC Frame and time stamp at FC Frame Receiver Portal
WWN = World Wide Name			
continued			

Table 2: FC/FCIP Entity pair interactions (part 1 of 5)

Reference Section	Condition	Information/Parameter Passed and Direction	
		FCIP Entity--->	<---FC Entity
continued			
5.6 FCIP Data Engine	FCIP Frame received from IP Network	Provide FC Frame and time stamp at FC Frame Trans- mitter Portal	
5.6.2.2 Errors in FCIP Headers and Discarding FCIP Frames	FCIP_DE discards bytes delivered through Encapsulated Frame Receiver Portal	Inform FC Entity that bytes have been discarded with reason	
5.6.2.3 Synchron- ization Failures	FCIP Entity closes TCP Connection due to synchron- ization failure	Inform FC Entity that TCP Connection has been closed with reason for closure	
8.1.2.3 Connection Setup Following a Successful TCP Connect Request	Receipt of the echoed FSF takes too long or the FSF contents have changed	Inform FC Entity that TCP Connection has been closed with reason for closure	
WWN = World Wide Name			
continued			

Table 2: FC/FCIP Entity pair interactions (part 2 of 5)

Reference Section	Condition	Information/Parameter Passed and Direction	
		FCIP Entity--->	<---FC Entity
continued			
8.1.2.1 Non-Dynamic Creation of a New TCP Connections	New TCP Connection created based on "shared" database information	Inform FC Entity of new or existing FCIP_LEP and new FCIP_DE along with Destination FC Fabric Entity WWN, Connection Usage Flags, Connection Usage Code and Connection Nonce	
8.1.2.2 Dynamic Creation of a New TCP Connections	New TCP Connection created based on SLP service advertisement and "shared" database information	Inform FC Entity of new or existing FCIP_LEP and new FCIP_DE along with Destination FC Fabric Entity WWN, Connection Usage Flags, Connection Usage Code and Connection Nonce	
WWN = World Wide Name			
continued			

Table 2: FC/FCIP Entity pair interactions (part 3 of 5)

Reference Section	Condition	Information/Parameter Passed and Direction: FCIP Entity--->	Information/Parameter Passed and Direction: <---FC Entity
continued			
8.1.3 Processing Incoming TCP Connect Requests	New TCP Connection created based on incoming TCP Connect request and "shared" database information	Inform FC Entity of new or existing FCIP_LEP and new FCIP_DE along with Source FC Fabric Entity WWN, Source FC/FCIP Entity Identifier, Connection Usage Flags, Connection Usage Code and Connection Nonce	
8.1.3 Processing Incoming TCP Connect Requests	TCP Connect Request wants to add a new TCP Connection to an existing FCIP_LEP	Request FC Entity to authenticate the source of the TCP Connect Request	Yes or No answer about whether the source of the TCP Connect Request can be authenticated
8.1.3 Processing Incoming TCP Connect Requests	Receipt of the FSF takes too long or duplicate Connection Nonce value	Inform FC Entity that TCP Connection has been closed with reason for closure	
WWN = World Wide Name			
continued			

Table 2: FC/FCIP Entity pair interactions (part 4 of 5)

Reference Section	Condition	Information/Parameter Passed and Direction: FCIP Entity--->	Information/Parameter Passed and Direction: <---FC Entity
concluded			
8.2 Closing TCP Connections	FC Entity determines that a TCP Connection needs to be closed	Acknowledgement of TCP Connection closure	Identification of the FCIP_DE whose TCP Connection needs to be closed
8.4 TCP Connection Considera- tions	Discovery that TCP connectiv- ity has been lost	Inform FC Entity that TCP Connection has been closed with reason for closure	
9.4.1 FCIP Link Initializ- ation Steps	IKE phase 1 failed, result- ing in termin- ation of link initialization	Inform FC Entity that TCP Connection can not be opened with reason for failure	
9.4.3 Handling data integrity and confi- dentiality violations	Excessive numbers of dropped datagrams detected and TCP Connection closed	Inform FC Entity that TCP Connection has been closed with reason for closure	
RFC 3723 Handling SA parameter mismatches	TCP Connection closed due to SA parameter mismatch problems	Inform FC Entity that TCP Connection has been closed with reason for closure	
WWN = World Wide Name			

Table 2: FC/FCIP Entity pair interactions (part 5 of 5)

Editors and Contributors Acknowledgements

During the development of this specification, Murali Rajagopal, Elizabeth Rodriguez, Vi Chau, and Ralph Weber served consecutively as editors. Raj Bhagwat contributed substantially to the initial basic FCIP concepts.

Venkat Rangan contributed the Security section and continues to coordinate security issues with the ips Working Group and IETF.

Andy Helland contributed a substantial revision of Performance section, aligning it with TCP/IP QoS concepts.

Dave Peterson contributed the dynamic discovery section and edits to RFC 3822.

Anil Rijhsinghani contributed material related to the FCIP MIB and edits the FCIP MIB document.

Bob Snively contributed material related to error detection and recovery including the bulk of the synchronization recovery example annex.

Lawrence J. Lamers contributed numerous ideas focused on keeping FCIP compatible with B_Port devices.

Milan Merhar contributed several of the FCIP conceptual modifications necessary to support NATs.

Don Fraser contributed material related to link failure detection and reporting.

Bill Krieg contributed a restructuring of the TCP Connection setup sections that made them more linear with respect to time and more readable.

Several T11 leaders supported this effort and advised the editors of this specification regarding coordination with T11 documents and projects. These T11 leaders are: Jim Nelson (Framing and Signaling), Neil Wanamaker (Framing and Signaling), Craig Carlson (Generic Services), Ken Hirata (Switch Fabric), Murali Rajagopal (Backbone), Steve Wilson (Switch Fabric), and Michael O'Donnell (Security Protocols).

Editors and Contributors Addresses

Neil Wanamaker
Akara
10624 Icarus Court
Austin, TX 78726
USA

Phone: +1 512 257 7633
Fax: +1 512 257 7877
EMail: nwanamaker@akara.com

Ralph Weber
ENDL Texas, representing Brocade
Suite 102 PMB 178
18484 Preston Road
Dallas, TX 75252
USA

Phone: +1 214 912 1373
EMail: roweber@ieee.org

Elizabeth G. Rodriguez
Dot Hill Systems Corp.
6305 El Camino Real
Carlsbad, CA 92009
USA

Phone: +1 760 431 4435
EMail: elizabeth.rodriguez@dothill.com

Steve Wilson
Brocade Comm. Systems, Inc.
1745 Technology Drive
San Jose, CA. 95110
USA

Phone: +1 408 333 8128
EMail: swilson@brocade.com

Bob Snively
Brocade Comm. Systems, Inc.
1745 Technology Drive
San Jose, CA 95110
USA

Phone: +1 408 303 8135
EMail: rsnively@brocade.com

David Peterson
Cisco Systems - SRBU
6450 Wedgwood Road
Maple Grove, MN 55311
USA

Phone: +1 763 398 1007
Cell: +1 612 802 3299
EMail: dap@cisco.com

Donald R. Fraser
Hewlett-Packard
301 Rockrimmon Blvd., Bldg. 5
Colorado Springs, CO 80919
USA

Phone: +1 719 548 3272
EMail: Don.Fraser@HP.com

R. Andy Helland
LightSand Communications, Inc.
375 Los Coches Street
Milpitas, CA 95035
USA

Phone: +1 408 404 3119
Fax: +1 408 941 2166
EMail: andyh@lightsand.com

Raj Bhagwat
LightSand Communications, Inc.
24411 Ridge Route Dr.
Suite 135
Laguna Hills, CA 92653
USA

Phone: +1 949 837 1733 x104
EMail: rajb@lightsand.com

Bill Krieg
Lucent Technologies
200 Lucent Lane
Cary, NC 27511
USA

Phone: +1 919 463 4020
Fax: +1 919 463 4041
EMail: bkrieg@lucent.com

Michael E. O'Donnell
McDATA Corporation
310 Interlocken Parkway
Broomfield, CO 80021
USA

Phone: +1 303 460 4142
Fax: +1 303 465 4996
EMail: modonnell@mcdata.com

Anil Rijhsinghani
McDATA Corporation
310 Interlocken Parkway
Broomfield, CO 80021
USA

Phone: +1 508 870 6593
EMail: anil.rijhsinghani@mcdata.com

Milan J. Merhar
43 Nagog Park
Pirus Networks
Acton, MA 01720
USA

Phone: +1 978 206 9124
EMail: Milan@pirus.com

Craig W. Carlson
QLogic Corporation
6321 Bury Drive
Eden Prairie, MN 55346
USA

Phone: +1 952 932 4064
EMail: craig.carlson@qlogic.com

Venkat Rangan
Rhapsody Networks Inc.
3450 W. Warren Ave.
Fremont, CA 94538
USA

Phone: +1 510 743 3018
Fax: +1 510 687 0136
EMail: venkat@rhapsodynetworks.com

Lawrence J. Lamers
SAN Valley Systems, Inc.
6320 San Ignacio Ave.
San Jose, CA 95119-1209
USA

Phone: +1 408 234 0071
EMail: ljlamers@ieee.org

Murali Rajagopal
Broadcom Corporation
16215 Alton Parkway
Irvine,CA 92619
USA

Phone: +1 949 450 8700
EMail: muralir@broadcom.com

Ken Hirata
Vixel Corporation
15245 Alton Parkway, Suite 100
Irvine, CA 92618
USA

Phone: +1 949 788 6368
Fax: +1 949 753 9500
EMail: ken.hirata@vixel.com

Vi Chau
USA
Email: vchau1@cox.net

Full Copyright Statement

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.