

0.1 FC-BB_FCoE definitions

insert here.

0.2 List of commonly used acronyms and abbreviations

ACE	Access Control Entry
ACL	Access Control List
D_A_TOV	Discovery Advertisement Time-Out Value
FCF	Fibre Channel Forwarder
FIP	FCoE Initialization Protocol
FPMA	Fabric Provided MAC Address
MAC	Media Access Control
VLAN	Virtual Local Area Network

0.3 Approved references

IEEE Std. 802.1DTM-2004, *IEEE Standard for Local and metropolitan area networks, Media Access Control (MAC) Bridges*

0.4 Defined Constants

This standard utilizes several defined constants. The value for each of these constants is specified in Table 1.

Table 1 – Defined Constants

Constant	Value	Description
FIP_TYPE	8914h	Value to be used in the Type field of the 802.3 frame to indicate a FIP payload
FCoE_Type	8906h	Value to be used in the type field of the 802.3 frame to indicate an FCoE payload
ALL_FCF_MACS	TBD	Group address for all FCFs
ALL_ENODE_MACS	TBD	Group address for all ENodes
DEFAULT_FIP_PRIORITY	128	Default value to be used in the Priority Descriptor
DEFAULT_FC-MAP	0EFC00h	Default value to be used for the FC-MAP
D_A_TOV	Default: 5 May be administratively configured to any integer value between 1 and 60, inclusive	Number of seconds between Discovery Advertisements sent by FCFs

1 Increasing FCoE robustness using FIP Snooping and FPMA (*Informative*)

Editor's Note: This annex is specific to FPMA. It is assumed that other interested parties will either provide a similar annex for SPMA, or will enhance this annex to cover both FPMA and SPMA.

1.1 Introduction

In Fibre Channel networks, Fibre Channel switches are generally considered trusted devices. Other Fibre Channel devices must log into the switch before they can communicate with the rest of the fabric. Given that Fibre Channel links are point-to-point, the Fibre Channel switch has complete control over the traffic a device injects into the fabric or is received from the fabric. As a result, the switch can enforce zoning configurations, ensure devices are using their assigned addresses, and prevent various types of anomalous behaviors (both erroneous and malicious).

FCoE provides increased flexibility; however, with this flexibility new challenges arise in assuring highly robust fabrics. Specifically, if Ethernet bridges exist between an ENode and the FCF, the point-to-point assurance between ENode and FCF is lost. Thus the FCF does not have the complete authority that a Fibre Channel switch has.

Equivalent robustness between FCoE and Fibre Channel is possible if one can ensure that all FCoE traffic to and from an ENode must pass through an FCF, and that if multiple devices can access an FCF through a single physical FCF port, that all those devices use their assigned MAC addresses. Doing so, in effect, creates the equivalent of a point-to-point link between the ENode and FCF.

One possible method of accomplishing this is to ensure every ENode is physically connected to an FCF with no intervening Ethernet bridges. Unfortunately, in many deployments this would prove impractical. For example, in large scale blade or 1U server environments, deploying an FCF in each blade system or top of rack switch creates the same scaling limitations in FCoE that are well known today in comparably configured Fibre Channel fabrics.

Ethernet bridges commonly provide a feature called Access Control Lists (ACLs). Properly configured ACLs can emulate a point-to-point link by providing the traffic enforcement previously discussed. Furthermore, the FIP protocol has been designed to enable Ethernet bridges to efficiently monitor FIP frames passing through them. With this data, FPMA simplifies the automatic configuration of these ACLs. In addition, the automatic configuration is possible independent of any other ACLs that might be in use in the fabric for other applications.

This clause discusses the ACLs, the required Access Control Entries (ACEs) within the ACL to provide equivalent FCoE robustness, and the process of generating these ACEs automatically via FIP Snooping.

1.2 ACL Introduction

The implementation of ACLs is not standardized and specifics vary between Ethernet bridges. However, certain features are available from a wide variety of Ethernet bridges.

In general, an Access Control List consists of an ordered list of rules that determine whether a given frame should be forwarded ("permit") or discarded ("deny"). Each rule is specified by matching bits within the received frame to a specified pattern. The pattern may require that the bit be a one, zero, or don't care. If a frame matches multiple patterns within the ACL, the first matching ACE determines whether the frame is permitted or denied. A default permit or deny may be specified to cover the case in which no patterns match.

Most ACL implementations operate on frames at ingress (referred to as an ingress ACL). Some implementations can also apply ACLs at egress (referred to as an egress ACL). For the purpose of this annex, all ACL processing is assumed to be done at ingress. These ACLs are applied to ports directly connected to ENodes and may also be applied to ports connected to other switches with ENodes connected downstream.

Editor's Note: Currently, when applied to a switch-to-switch link, these ACLs only operate properly if there are no FCFs down stream from the connection and with limitations related to multiple and/or redundant paths. This annex needs to be either expanded to describe these restrictions are to add capability to remove these restrictions. This is a work in progress.

In addition, ACL implementations vary in how deeply into a frame the patterns may be applied. For the purpose of the Annex, it is only necessary to examine the Source and Destination MAC address fields, the VLAN tag, and the Type fields. Most ACL implementations are capable of this.

Finally, implementations vary as to whether bridge learning is subjected to the ingress ACL. This annex assumes that bridge learning is, in fact, subjected to the ACL (i.e., if the frame is denied by the ACL, its source address will not be learned). For implementations that do learn source addresses of denied frames, a simple extension using Static Forwarding Entries (as defined in IEEE Std. 802.1DTM-2004) is discussed in clause 1.3 to provide equivalent functionality.

1.2.1 ACL Nomenclature

The exact method of specifying ACLs varies by implementation. A generalized nomenclature is used in this annex. In general, an ACE has the form of:

`[field = value],[field = value],...,permit || deny;`

The last ACE may contain only the keyword “permit” or “deny” to cover the case that no ACEs match.

The fields used in this Annex are:

- DA: Destination MAC Address (48 bits)
- DApr: 24 most significant bits of Destination MAC Address (24 bits)
- SA: Source MAC Address (48 bits)
- SApr: 24 most significant bits of the Source MAC Address (24 bits)
- VLAN: Value of the VLAN field within the VLAN tag (12 bits)
- Type: Value of the Type field (16 bits)

The following constants, defined in table 1, are used:

- FIP_TYPE
- FCoE_TYPE

FC-MAP is the 24-bit FPMA address prefix being used on the fabric.

Finally, “{FCFs}” is used to represent the set of MAC Addresses to FCFs to which given Ethernet Bridge port is to allow connectivity. For simplicity, a single ACE is illustrated using this set. In general, multiple ACEs may be required to represent all the members of the set.

1.2.2 ACL Construction

The following are the requirements of the ACL:

- a) Enable transmission of FIP frames from ENodes to FCFs
- b) Ensure that FIP frames can only be addressed to FCFs
- c) Prevent transmission of all FCoE frames from an ENode prior to its successful completion of FLOGI
- d) Prevent transmission from an ENode to any other ENode's FCoE MAC address
- e) Prevent transmission of frames using an FCF's MAC address as a source (except for FCFs, of course)
- f) After successful completion of FLOGI, ensure that only the only FCoE source addresses used by an ENode are the ones assigned by the FCF to that ENode
- g) After successful completion of FLOGI, ensure that the assigned FCoE source address is only used for FCoE traffic
- h) After successful completion of FLOGI, ensure that FCoE frames can only be addressed to the FCF
- i) Ensure that ACEs generated for FCoE do not conflict with other ACEs that might be used in the fabric for any other purpose
- j) Ensure that the ACEs do not inhibit non-FCoE traffic (i.e. traffic that does not contain the FCoE or FIP type value *and* does not utilize an FCoE source MAC address)

The following sub-paragraphs describe the ACEs required to achieve each of these requirements.

1.2.3 FIP Frame Transmission

An ENode must be allowed to send FIP frames to FCFs, and only to FCFs. These frames may be addressed to a specific FCF, or to the ALL_FCF_MACS group address. ACEs that accomplish this are:

```
DA = ALL_FCF_MACs, Type = FIP_TYPE, permit;
DA = {FCFs}, Type = FIP_TYPE, permit;
Type = FIP_TYPE, deny;
```

1.2.4 Prevention of the transmission of frames using an FCF's MAC address for the source

An ENode (or any non-FCF device) must not be allowed to transmit frames using an FCF's source address. This is necessary to prevent various of address learning and ACL spoofing attacks. The ACE that prevents this is:

```
SA = {FCFs}, deny;
```

1.2.5 Prevention of FCoE frames prior to successful completion of FLOGI

ENodes are not permitted to send any FCoE frames prior to the successful completion of FLOGI. FCoE frames are identified by the type field. In addition, ENodes are not permitted to use any FCoE source MAC address prior to the successful completion of FLOGI to ensure that bridge learning is not compromised. The ACEs to accomplish this are:

```
Type = FCoE_Type, deny;
SApre = FC_MAP, deny;
```

Note that the FC_MAP allows one to simply identify all FCoE MAC addresses with a single ACE without a priori knowledge of MAC addresses to be used for FCoE.

1.2.6 Prevention of ENode to ENode communication using FCoE addresses

ENodes are not permitted to send frames to other ENodes using FCoE ENode destination MAC addresses. This prevents an ENode interpreting traffic from another ENODE as being from an FCF (including the case of an ENode using an FCF's source address rendering checking of the frame by the receiving ENode insufficient). The ACE to accomplish this is:

DApre = FC-MAP, deny;

Again, note that all FCoE MAC addresses are identified with a single ACE utilizing the FC-MAP, again without any a priori knowledge of assigned FCoE MAC addresses.

1.2.7 Enabling traffic after successful completion of FLOGI

After successful completion of FLOGI, FCoE traffic between the ENode and the FCF that accepted the FLOGI using the assigned FCoE MAC address must be enabled. The following ACE accomplishes this:

SA = FCF assigned MAC address, DA = FCF MAC address, Type = FCoE, permit;

Note that these ACEs must be inserted anywhere prior to those in section 1.2.5 (it may be convenient to simply insert these at the top of the ACL).

1.2.8 ACL summary

Prior to receipt of any Discovery Advertisements, the initial ACL is:

DA = ALL_FCF_MACs, Type = FIP_TYPE, permit;
 Type = FIP_TYPE, deny;
 Type = FCoE_Type, deny;
 Any non-FCoE related ACEs.

After receipt of Discovery Advertisements (or as the result of administrative configuration), the ACL is expanded to:

DA = ALL_FCF_MACs, Type = FIP_TYPE, permit;
 DA = {FCFs}, Type = FIP_TYPE, permit;
 Type = FIP_TYPE, deny;
 Type = FCoE_Type, deny;
 SA = {FCFs}, deny;
 SApre = FC_MAP, deny;
 DApre = FC-MAP, deny;
 Any non-FCoE related ACEs.

For each successful FLOGI (or FDISC), an ACE is added (prior to SApre = FC-MAP, deny;) of the form:

SA = FCF assigned MAC address, DA = FCF MAC address, Type = FCoE, permit;

1.2.9 Automatic Configuration of ACLs

An Ethernet bridge may choose to snoop FIP frames from which all the information needed to automatically configure these ACLs may be determined.

The set of FCF MAC addresses {FCFs} may be determined by creating a list from all received FIP Discovery Advertisements. In addition, the FC-MAP may be determined from these advertisements.

Editor's Note: The above requires some way to ensure that only FCFs are able to inject discovery advertisements (i.e. securely determining that the device you are connected to is indeed an FCF). At a minimum, this can be achieved by specific configuration of the edge ports directly connected to FCFs (which could be automatic with in Ethernet switches embedded in FCFs). Additional work in, for example, FC-SP-2 may enable greater levels of automation in this area. This continues to be a work in progress.

The post FLOGI / FDISC ACEs may be constructed by examining the FIP FLOGI Accepts / FDISC Accepts transmitted from the FCFs. These contain the FCF assigned MAC address and the FCF's MAC address. It may be possible to obtain additional robustness by storing the FIP FLOGIs and FDISCs, and then ensuring valid matches with their accepts.

All other values required for the ACLs are defined constants.

Editor's Note: This section needs to be expanded to cover FIP Clear Virtual Link functionality.

1.3 Ethernet bridge learning considerations

Some implementations of ACLs allow a bridge to learn a source address even if the frame is denied by the ACL. This can leave a fabric vulnerable to certain Ethernet learning attacks. In such implementations, Static Forwarding Entries may be used to supplement the ACL.

To accomplish this, when a post FLOGI / FDISC ACE is created, also create a Static Forwarding Entry for the assigned MAC address. This entry should specify "forward" for the port on which the FLOGI / FDISC is received, and "filter" for all other ports.

1.4 VLAN considerations

It is possible for separate FCoE fabrics to exist on separate VLANs. Using a common FC-MAP for the entire physical infrastructure allows most of the ACEs to operate without any modification in this case. The only exception are the ACEs that are generated in response to snooping the FIP FLOGI / FDISC accepts. In this case, the ACE must be qualified with the appropriate VLAN (which also may be snooped from the FIP FLOGI / FDISC accept).