

0.1 Acknowledgements

The following have contributed to the generation of this Annex:

David Black	EMC
Claudio DeSanti	Cisco
Silvano Gai	Cisco
Roger Hathorn	IBM
Landon Noll	Cisco
Joe Pelissier	Cisco
David Peterson	Brocade

0.2 FC-BB_FCoE definitions

insert here.

0.3 List of commonly used acronyms and abbreviations

ACE	Access Control Entry
ACL	Access Control List
FCF	Fibre Channel Forwarder
FIP	FCoE Initialization Protocol
FPMA	Fabric Provided MAC Address
SPMA	Server Provided MAC Address
MAC	Media Access Control
VLAN	Virtual Local Area Network

0.4 Approved references

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

0.5 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	01-10-18-01-00-02	Group address for all FCFs
ALL_ENODE_MACS	01-10-18-01-00-01	Group address for all ENodes

1 Increasing FCoE robustness using Access Control Lists *(Informative)*

1.1 Introduction

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

FCoE provides increased flexibility, but 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 the ENode and FCF is lost. Thus the FCF does not have the complete control that a Fibre Channel switch has.

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

Note that the above are necessary, but not sufficient, conditions to ensure equivalent robustness. See annex TBD for a complete discussion on achieving equivalent robustness.

One possible method of accomplishing this is to ensure every ENode is physically connected to an FCF with no intervening Ethernet bridges, but in many deployments this is not practical.

Ethernet bridges commonly provide a feature called Access Control Lists (ACLs). Properly configured ACLs may 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. This data facilitates the automatic configuration of these ACLs. In addition, the automatic configuration is possible independent of any other ACLs that may be in use in the fabric for other applications.

This annex 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.

The particular set of ACEs to be used depend on the location of the port within the fabric and the traffic that is administratively configured to pass through it. Figure 1 illustrates a fabric along with the potential different ACEs that may apply.

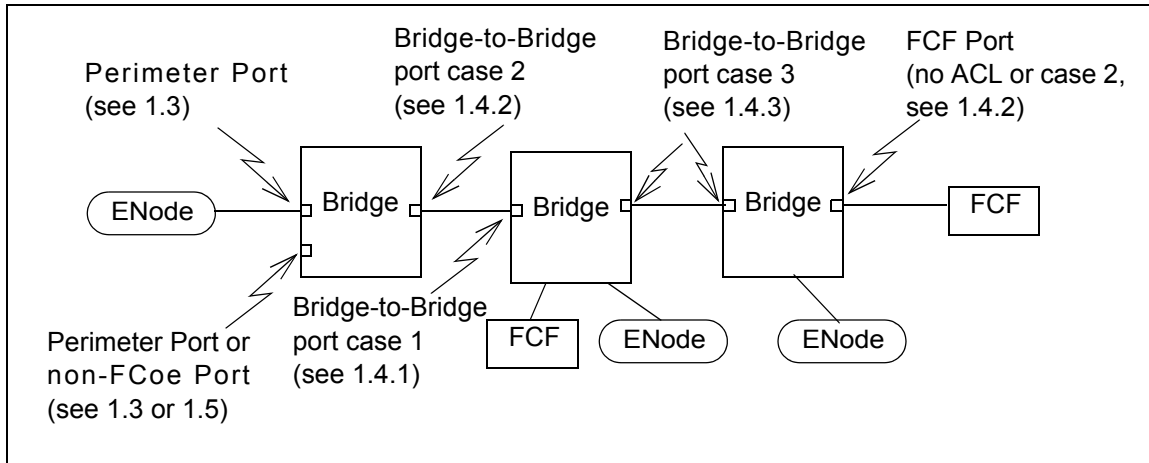


Figure 1 – Bridge Port to ACE Cross Reference

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 (i.e., “permit”) or discarded (i.e., “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 in the last entry to cover the case in which no patterns match.

Most ACL implementations allow specification of ACLs per bridge port and operate on frames as they enter the bridge using the ACL specified for the ingress port (referred to as an ingress ACL). Some implementations may also apply ACLs to frames as they exit from the bridge (referred to as an egress ACL). For the purpose of this annex, all ACLs are assumed to be ingress ACLs.

It is recommended that ACL protection be applied at the edge of the fabric, i.e., at the ports that connect directly to the ENodes. ACLs are provided in this Annex that are intended to be applied to these ports. It is also possible to construct ACLs that provide a lower level of protection on bridge-to-bridge ports. This annex also provides suggested ACLs for this purpose.

In addition, ACL implementations vary in how deeply into a frame the patterns may be applied. For the purpose of this annex, it is only necessary to examine the Source and Destination MAC address fields, the VLAN tag, and the Type (aka Ethertype) 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 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 (see IEEE Std. 802.1DTM-2004) is discussed to provide equivalent functionality (see 1.7).

1.2.1 ACL Nomenclature

The exact method of specifying ACLs varies by implementation. A generalized nomenclature is used in this annex. ACLs consist of an ordered list of access control entries. In general, an access control entry (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)
- 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) (aka Ethertype)

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

- FIP_TYPE
- FCoE_TYPE

FC-MAP applies only if FPMA addressing is in use and is the 24-bit FPMA address prefix being used on the fabric.

“{FCFs}” represents the set of FCF MAC Addresses to which a given ENode is allowed 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.

1.3 Perimeter ACL Construction

The ACL described in this clause should be used at the perimeter in the fabric. This includes bridge ports connected to ENodes and unconnected bridge ports. It may also be used on bridge to bridge ports to provide additional security in depth; however, doing so in certain deployments may exceed the ACE capacity of the bridge. In addition, deploying this ACL on bridge-to-bridge ports may limit the fabric’s ability to autonomously respond to link failures. See 1.4 for additional options to address these issues on bridge-to-bridge links.

The following are the requirements of the ACL and the subparagraphs that describe how these requirements are met using ACEs:

- a) Enable transmission of FIP frames from ENodes to FCFs (see 1.3.1);
- b) Ensure that FIP frames from ENodes may only be addressed to FCFs (see 1.3.1);
- c) Ensure no end device uses an FCF MAC address as its source (see 1.3.2);
- d) Prevent transmission of all FCoE frames from an ENode prior to its successful completion of FLOGI (see 1.3.3);
- e) After successful completion of FLOGI, ensure that only the FCoE source addresses used by an ENode are the ones assigned by the FCF to that ENode (see 1.3.4);

- f) After successful completion of FLOGI, ensure that the assigned FCoE source address is only used for FCoE traffic (see 1.3.4); and
- g) After successful completion of each FLOGI or FDISC, ensure that FCoE frames may only be addressed to the accepting FCFs (see 1.3.4).

These ACEs are constructed such that if they are inserted prior to any other non-FCoE and non-FIP related ACEs that may be in use, they will not conflict with those ACEs. In addition, these ACEs are constructed such that they do not inhibit non-FCoE and non-FIP traffic (i.e. traffic that does not contain the FCoE or FIP type value and does not utilize an FCoE source MAC address)

1.3.1 FIP Frame Transmission

An ENode is 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;
```

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

An ENode is not 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;
```

1.3.3 Prevention of frames using FCoE Type or FCoE Source Addresses 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 being equal to FCoE_Type. The ACE to accomplish this is:

```
Type = FCoE_Type, deny;
```

1.3.4 Enabling traffic after successful completion of FLOGI (or FDISC)

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

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

Note that for proper operation these ACEs are inserted anywhere prior to those in section 1.3.3 (it may be convenient to simply insert these at the top of the ACL).

1.3.5 Prevention of Duplicate VN_Port MAC Addresses

Duplicate VN_Port MAC addresses within a FC-BB-E network can result in various catastrophic failures including undetected corruption of data, denial of service, and undetected interception of data. Duplicate VN_Port MAC addresses may occur due to fabric configuration issues and due to malicious entities on the fabric. Duplicate VN_Port MAC addresses may be prevented with the use of FPMA and the appropriate use of ACEs. In general, it is not practical to construct an ACE to prevent

address duplication with SPMA since doing so requires a priori knowledge of all MAC Addresses that are being used for VN_Port MAC addresses everywhere in the fabric. As a result, duplicate VN_Port address prevention is beyond the scope of this annex for SPMA.

With FPMA, it is possible to identify all VN_Port MAC addresses. The following ACE causes a bridge port to discard any frame with a source address equal to a VN_Port MAC address:

SAPre = FC-MAP, deny;

1.3.6 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;
 SAPre = FC_MAP, deny; --Note: applies to FPMA only
 Any non-FCoE related ACEs.

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

SA = {FCFs}, deny;
 DA = ALL_FCF_MACs, Type = FIP_TYPE, permit;
 DA = {FCFs}, Type = FIP_TYPE, permit;
 Type = FIP_TYPE, deny;
 Type = FCoE_Type, deny;
 SAPre = FC_MAP, deny; --Note: applies to FPMA only
 Any non-FCoE related ACEs

For each successful FLOGI (or FDISC), an ACE is added prior to "Type=FCoE_Type, deny;" of the form:

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

1.4 Security in Depth

The ACL described in section 1.3, if properly deployed at the perimeter of the fabric (i.e. all bridge ports connected to all ENodes and unconnected ports), provide a high degree of fabric security. However, if a device is connected within this perimeter defense (e.g. a mis-configuration or an omitted port ACL), the level of security provided is diminished. Deployment of ACLs on bridge-to-bridge links provide additional defense in these situations. The ACL described in section 1.3 may be used for this purpose. However, as previously discussed, use of this ACL on bridge-to-bridge ports results in undesirable scalability and fabric resiliency characteristics.

This section provides alternative ACLs that may be used on bridge-to-bridge links that have better scalability and fabric resilience characteristics. Three sets of ACLs are provided that are designed for a specific use within the fabric (see figure 1 for an illustration of the following cases):

- (Case 1) A bridge port, connected to a bridge-to-bridge link, receiving frames from ENode(s) destined for FCF(s) (but not the other direction)
- (Case 2) A bridge port, connected to a bridge-to-bridge link or to an FCF, receiving frames from FCF(s) destined for ENode(s) (but not the other direction)

- (Case 3) A bridge port, connected to a bridge-to-bridge link, receiving frames from both FCF(s) and ENode(s)

1.4.1 Bridge-to-bridge link receiving ENode frames destined to FCF(s)

In the case of a bridge-to-bridge link receiving frames from ENode(s), which by definition are destined to FCF(s), the ingress bridge port should check for basic validity of the received frames. This includes:

Verification that FIP frames are addressed to FCFs

Verification that FCoE frames are addressed to FCFs

Verification that FCoE frames are sourced only by ENodes (see Note 1)

Prevention of FCoE frames from one FCF destined to another FCF (see note 2)

NOTE 1 – This protection is practical with FPMA only.

NOTE 2 – Normally, FCFs are permitted to transfer frames between one another. However, this is a special case of a bridge-to-bridge link on which it is administratively known that all of the traffic being received is supposed to be from ENode(s), therefore, it is proper to disallow FCF to FCF traffic. See 1.4.3 to address the case in which traffic may be flowing from FCF to FCF.

A set of ACEs that accomplish this are:

```
DA=ALL_FCF_MACS, Type=FIP_TYPE, permit;
DA={FCFs}, Type=FIP_TYPE, permit;
Type=FIP_TYPE, deny;
DA={FCFs}, SApre=FC_MAP, Type=FCoE_TYP, permit; -- FPMA only
DA={FCFs}, SA={FCFs}, Type=FCoE_TYPE, deny; -- SPMA only, but no harm for FPMA
DA={FCFs}, Type=FCoE_TYPE, permit; -- SPMA only
Type=FCoE_TYPE, deny;
```

1.4.2 Bridge-to-bridge link receiving FCF frames destined to ENode(s)

In the case of a bridge-to-bridge link receiving frames from FCF(s) with administrative knowledge that that these frames are destined only to ENodes (i.e. there are no FCFs downstream) the ingress bridge port should check for basic validity of the received frames. This includes:

- Verification that all FIP frames are sourced from an FCFs and are not destined to other FCFs
- Verification that all FCoE frames are sourced from an FCF
- Verification that all FCoE frames are destined to ENodes (see note 3)
- Prevention of FCoE frames destined for an FCF (see note 4)

NOTE 3 – This protection is practical with FPMA only.

NOTE 4 – Normally, FCFs are permitted to transfer frames between one another. However, this is a special case of a bridge-to-bridge link on which it is administratively known that all of the traffic being received is supposed to be destined ENode(s), therefore, it is proper to disallow FCF to FCF traffic. See 1.4.3 to address the case in which traffic may be flowing from FCF to FCF.

A set of ACEs that accomplish this are:

```
DA={FCFs}, Type=FIP_TYPE, deny;
DA=ALL_ENODE_MACS, Type=FIP_TYPE, permit --see note 5
SA={FCFs}, Type=FIP_TYPE, permit;
Type=FIP_TYPE, deny;
SA={FCFs}, DApr=FC_MAP, Type=FCoE_TYPE, permit; -- FPMA only
DA={FCFs}, SA={FCFs}, Type=FCoE_TYPE, deny; -- SPMA only, see note 6
SA={FCFs}, Type=FCoE_TYPE, permit; -- SPMA only
Type=FCoE_TYPE, deny;
```

NOTE 5 – This ACE should only be included if it is administratively known and trusted that only FCFs will/can inject frames onto the fabric destined to ALL_ENODE_MACS. Including this ACE enables automatic population of the {FCFs} set. If this cannot be trusted, this ACE should be included which has the side affect of disabling automatic population of the {FCFs} set, thus requiring that the set be populated administratively.

NOTE 6 – This ACE may result in scalability issues in some deployments. Given n FCFs in a fabric, this expands to n squared entries, which may exceed the ACL capability of the bridge. In this case it becomes impractical to ensure that FCFs are not sourcing these frames when SPMA is in use.

1.4.3 Bridge-to-bridge link receiving both FCF and ENode frames

In the case of a bridge-to-bridge link receiving frames from both FCF(s) and ENodes, including frames that are exchanged between FCFs, the ingress bridge port should check for basic validity of the received frames. This includes:

- Verify that all FIP frames are either sourced by or destined to an FCF
- Verify that all FCoE frames are sourced by an FCF and destined to either an ENode or FCF; or sourced by an ENode and destined to an FCF (see note 7)
- Verify that all FCoE frames or either sourced by are destined to an FCF

NOTE 7 – This protection is practical with FPMA only.

A set of ACEs that accomplish this are:

```
DA=ALL_FCF_MACS, Type=FIP_TYPE, permit;
DA={FCFs}, Type=FIP_TYPE, permit;
SA={FCFs}, Type=FIP_TYPE, permit;
Type=FIP_TYPE, deny;
DA={FCFs}, SApr=FC_MAP, Type=FCoE_TYPE, permit; -- FPMA only
SA={FCFs}, DApr=FC_MAP, Type=FCoE_TYPE, permit; -- FPMA only
DA={FCFs}, Type=FCoE_TYPE, permit; -- SPMA only
SA={FCFs}, Type=FCoE_TYPE, permit; -- SPMA only
Type=FCoE_TYPE, deny;
```

1.4.4 Additional FCF Crosstalk Protection

The preceding ACL recommendations have been based on the set of FCF MAC addresses, {FCFs}, that contain all FCF MAC addresses including MAC addresses used for VE_Ports, VF_Ports, and the FCFs' FCoE Controllers. It is possible that this MAC address set could be subdivided into their individual uses to create ACEs that provide greater protection against unintended FCF to FCF communication. For example, an ACE may be constructed that prevents VE_Ports from communicating with

VF_Ports, etc. While the details of achieving this is beyond the scope of this annex, this possibility may be considered for individual implementations and deployments.

1.5 Prevention of FCoE related traffic

It may be desirable to prevent the reception of all FCoE related traffic on a given bridge port. To do this, all frames with an Ethertype of FCoE_Type or FIP_Type must be denied. Additionally, it is desirable to deny all frames using any VN_Port's source MAC address (e.g. to prevent an attack from a rogue host or to prevent undetected data corruption due to an erroneous configuration). Denying such frames is generally practical only with FPMA. The ACEs to accomplish this are:

```
Type = FIP_Type, deny;
Type = FCoE_Type, deny;
SApre = FC_MAP, deny; --Note: applies to FPMA only
```

1.6 Automatic Configuration of ACLs

An Ethernet bridge may choose to snoop FIP frames from which all, or most in the case of SPMA, 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.

It is important to note that the ACLs recommended in this annex do not prevent a rogue host from advertising itself as an FCF using the ALL_FCFS_MACS group address. Preventing this was considered beyond the scope of this annex since this vulnerability exists in Fibre Channel and is not unique to FCoE. Within Fibre Channel and FCoE, this vulnerability may be addressed using FC-SP to prevent E_Ports from being formed with such hosts.

However, bridges that examine FIP traffic to determine a list of FCF MAC addresses would include such rogue hosts in their list of valid FCFs if such frames were considered. To avoid this vulnerability bridges should examine only the FIP advertisements addressed to the ALL_ENODES_MACS group address. Alternatively, the list of FCF MAC addresses may be configured administratively.

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.

The FLOGI/FDISC ACEs may be deleted from one port and re-created on another if by examining the FIP FLOGI Accepts / FDISC accepts transmitted from the FCFs, it is determined that an ENode has moved from one port to another.

Finally, the FIP Clear Virtual Link message may be examined to determine that one of these ACEs may be removed from a port.

To ensure that only valid FIP messages are examined, all ports except those known to be connected to an FCF (e.g. via administrative configuration) should contain an ACE to filter FIP frames (the ACLs described in this annex contain such an entry).

All other values required for the ACLs are defined constants.

1.7 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 may 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.

1.8 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 is qualified with the appropriate VLAN, which also may be snooped from the FIP FLOGI/FDISC accept.

The use of multiple FC-MAPs in a given physical infrastructure as well as the use of multiple Virtual Fabrics on a single VLAN is beyond the scope of this annex.