

Annex C: Increasing FC-BB_E Robustness Using Access Control Lists (Informative)

C.1 Overview

In Fibre Channel ~~networks~~Fabrics, 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-an~~ 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~~ may be achieved by ensuring that all FCoE traffic to and from an ~~Enode~~ ENode passes through an ~~FCF~~, FCF and that if multiple ~~Enodes may~~ ENodes access an FCF through a single physical FCF port, those ~~Enodes~~ 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 D for a complete discussion on achieving equivalent robustness.

~~One~~ A possible method of ~~accomplishing~~ achieving this robustness 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~~ independently of any other ACLs that may be in use in the ~~Fabrie~~ network for other applications.

This annex discusses the ACLs, the required Access Control Entries (ACEs) within the ACL to provide FCoE with equivalent ~~FCoE robustness~~ robustness as Fibre Channel, and the process of generating these ACEs automatically ~~via-by~~ examining FIP ~~snooping~~ messages.

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

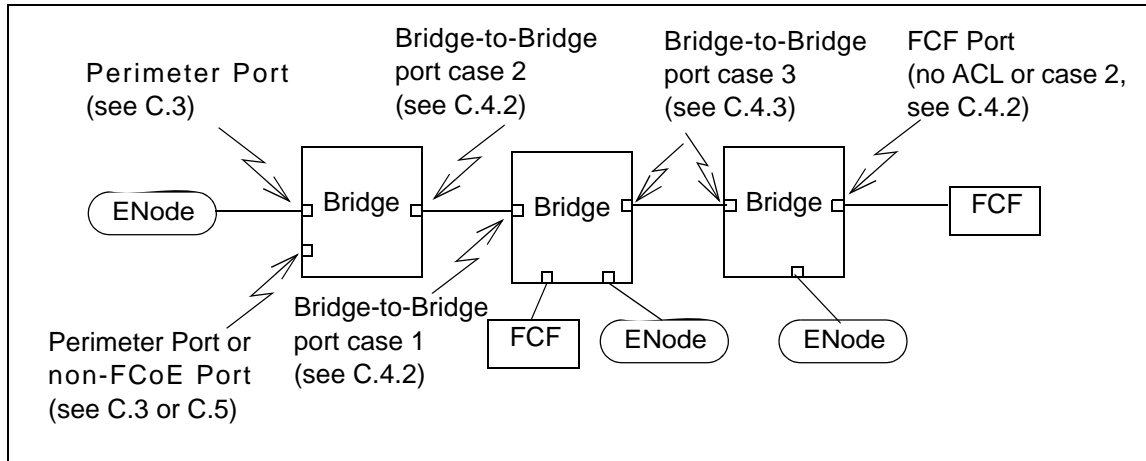


Figure C.1 – Bridge port to ACE cross reference

C.2 Access Control Lists

C.2.1 ACL overview

The implementation of ACLs is not standardized and specifics vary ~~between-among~~ 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-are set~~ to one, to zero, or to don’t care. If a frame matches multiple patterns ~~with-in-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 (i.e., an ingress ACL). Some implementations may also apply ACLs to frames as they exit from the bridge (i.e., 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, network~~ (i.e., at the ports that connect directly to the ENodes). ~~The 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 (i.e., Ethertype) fields. Most ACL implementations are capable of this.

~~Finally, implementations~~ ~~Implementations~~ vary as to whether bridge learning is ~~subjected-subject~~ to the ingress ACL. This annex assumes that bridge learning is ~~subjected-subject~~ to the ACL (i.e., if ~~the~~ a frame is denied by the ACL, its source address ~~will-is~~ not ~~be-~~learned). For implementations that ~~do~~ learn source addresses of denied frames, a simple extension using Static Forwarding Entries (see IEEE ~~Std.-802.4D™~~ 802.4D-2004) is discussed to provide equivalent functionality (see C.7).

C.2.2 ACL nomenclature

The exact method of specifying ACLs ~~varies by implementation~~ is implementation specific. 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:

- a) DA: Destination MAC Address (48 bits);
- b) SA: Source MAC Address (48 bits);
- c) SApr: 24 most significant bits of the Source MAC Address (24 bits);
- d) VLAN: Value of the VLAN field within the VLAN tag (12 bits); and
- e) Type: Value of the Type field (16 bits) (i.e., Ethertype).

The following constants, defined in table 45, are used:

- a) FIP_TYPE; and
- b) FCoE_TYPE.

FC-MAP applies only if ~~FPMA-addressing is~~ FPMAs are in use and is the 24-bit FPMA address prefix being used on the ~~Fabric~~ network (see 7.5a).

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

C.3 Perimeter ACL construction

The ACL described in this ~~clause-subclause~~ should be used at the perimeter ~~in-of~~ the ~~Fabric~~ network. 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-ACEs~~ capacity of the bridge. In addition, deploying this ACL on bridge-to-bridge ports may limit the ~~Fabric's-network~~ ability to autonomously respond to link failures. See C.4 for additional options to address these issues on bridge-to-bridge links.

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

- a) enable transmission of FIP frames from ENodes to FCFs (see C.3.1);
- b) ensure that FIP frames from ENodes may only be addressed to FCFs (see C.3.1);
- c) ensure no end device uses an ~~FCF-FCF~~-MAC address as its source (see C.3.2);
- d) prevent transmission of all FCoE frames from an ENode prior to its successful completion of FIP FLOGI (see C.3.3);
- e) after successful completion of FIP FLOGI, ensure that only the FCoE source addresses used by an ENode are the ones assigned by the FCF to that ENode (see C.3.4);
- f) after successful completion of FIP FLOGI, ensure that the assigned FCoE source address is only used for FCoE traffic (see C.3.4) and FIP traffic (i.e., VN_Port FIP Keep Alive messages); and
- g) after successful completion of each FIP FLOGI or FIP NPIV FDISC, ensure that FCoE frames may only be addressed to the accepting FCFs (see C.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 do~~ 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~~-Ethertype value and does not utilize an FCoE source MAC address).

C.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~~-All-FCF-MACs group address. ACEs that accomplish this are:

```
DA = ALL_FCF_MACSAll-FCF-MACs, Type = FIP_TYPE, permit;
DA = {FCFs}, Type = FIP_TYPE, permit;; -- see note 22
Type = FIP_TYPE, deny;
```

NOTE 22 – This ACE allows also VN_Port FIP Keep Alive messages.

C.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~~-FCF source address. This is necessary to prevent ~~various of~~ address learning and ~~ACL-spoofing~~-FCF impersonation attacks. The ACE that prevents this is:

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

C.3.3 Prevention of frames using FCoE Type or FCoE Source Addresses prior to successful completion of FIP FLOGI

ENodes are not permitted to send any FCoE frames prior to the successful completion of FIP FLOGI. FCoE frames are identified by the Type field being equal to FCoE_TYPE. The ACE to accomplish this is:

```
Type = FCoE_TYPE, deny;
```

C.3.4 Enabling traffic after successful completion of FIP FLOGI (or FIP NPIV FDISC)

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

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

For proper operation these ACEs are inserted anywhere prior to those in C.3.3 and it may be convenient to simply insert these at the top of the ACL.

C.3.5 Prevention of duplicate VN_Port MAC addresses

Duplicate VN_Port MAC addresses within a FC-~~BB-E~~-BB_E network ~~can result in~~-may lead to various catastrophic ~~failures~~-failures, including undetected corruption of data, denial of service, and undetected interception of data. Duplicate VN_Port MAC addresses may occur due to ~~Fabric~~-network configuration issues and ~~due~~ to malicious entities on the ~~Fabric~~network. Duplicate VN_Port MAC addresses may be prevented with the use of ~~FPMA~~-FPMAs and the appropriate use of ACEs. In general, it is not practical to construct an ACE to prevent address duplication with ~~SPMA~~-SPMAs since doing so requires a priori knowledge of all MAC Addresses that are being used ~~for~~-as VN_Port MAC

addresses everywhere in the **Fabric** network. As a result, duplicate VN_Port address prevention is beyond the scope of this annex for **SPMASPMAs**.

With **FPMAFPMAs**, 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;
```

This entry should be placed in the ACL after the entries described in C.3.4.

C.3.6 ACL summary

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

```
DA = ALL_FCF_MAGsAll-FCF-MACs, Type = FIP_TYPE, permit;
Type = FIP_TYPE, deny;
Type = FCoE_TYPE, deny;
SAPre = FC_MAPFC-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_MAGsAll-FCF-MACs, Type = FIP_TYPE, permit;
DA = {FCFs}, Type = FIP_TYPE, permit;
Type = FIP_TYPE, deny;
Type = FCoE_TYPE, deny;
SAPre = FC_MAPFC-MAP, deny; --- Note: applies to FPMA only
Any non-FCoE related ACEs
```

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

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

C.4 Security in depth

C.4.1 Overview

The ACL described in C.3, if properly deployed at the perimeter of the **Fabric** network (i.e., all bridge ports connected to all ENodes and unconnected ports), **provide** provides a high degree of **Fabric** network 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 C.3 may be used for this purpose. However, as previously discussed, use of this ACL on bridge-to-bridge ports **results in** has undesirable scalability and **Fabric** network resiliency characteristics.

This subclause provides alternative ACLs that may be used on bridge-to-bridge links that have better scalability and **Fabric** network resilience characteristics. Three sets of ACLs are **provided that are provided**, each designed for a specific use within the **Fabric** network. See figure C.1 for an illustration of the following cases:

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

NOTE 23 – Changes in routing (e.g., automatic spanning tree recalculation) may cause the assumptions on which the recommended ACLs are based to become invalid. This may have the result of weaker protection or Virtual Link timeouts, requiring relogins.

C.4.2 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 that~~ by definition are destined to FCF(s), the ingress bridge port should check for basic validity of the received frames. This includes:

- a) Verification that FIP frames are addressed to FCFs;
- b) Verification that FCoE frames are addressed to FCFs;
- c) Verification that FCoE frames are sourced only by ENodes (see ~~note 24~~note 24); and
- d) Prevention of FCoE frames from one FCF destined to another FCF (see ~~note 25~~note 25).

NOTE 24 – This protection is practical with FPMA only.

NOTE 25 – 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 C.4.4 to address the case in which traffic may be flowing from FCF to FCF.

A set of ACEs that accomplish this ~~are~~is:

```
DA=ALL_FCF_MACSALL-FCF-MACs, Type=FIP_TYPE, permit;
DA={FCFs}, Type=FIP_TYPE, permit;
Type=FIP_TYPE, deny;
DA={FCFs}, SApre=FC_MAPFC-MAP, Type=FCoE_TYPE, 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;
```

C.4.3 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:

- a) Verification that all FIP frames are sourced from an FCFs and are not destined to other FCFs;
- b) Verification that all FCoE frames are sourced from an FCF;
- c) Verification that all FCoE frames are destined to ENodes (see ~~note 26~~note 26); and
- d) Prevention of FCoE frames destined for an FCF (see ~~note 27~~note 27).

NOTE 26 – This protection is practical with FPMA only.

NOTE 27 – 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 sup-

posed to be destined ENode(s), therefore, it is proper to disallow FCF to FCF traffic. See C.4.4 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_MACSAll-ENode-MACs, Type=FIP_TYPE, permit --- see note 28note 28
SA={FCFs}, Type=FIP_TYPE, permit;
Type=FIP_TYPE, deny;
SA={FCFs}, DApr=FC_MAPFC-MAP, Type=FCoE_TYPE, permit; -- FPMA only
DA={FCFs}, SA={FCFs}, Type=FCoE_TYPE, deny; -- SPMA only, see note 29note 29
SA={FCFs}, Type=FCoE_TYPE, permit; -- SPMA only
Type=FCoE_TYPE, deny;
```

NOTE 28 – This ACE should only be included if it is administratively known and trusted that only FCFs ~~will/can~~ **are able to** inject frames ~~destined to All-ENode-MACs~~ **destined to All-ENode-MACs** onto the ~~Fabric~~ **destined to ALL_ENODE_MACS** network. Including this ACE enables automatic population of the {FCFs} set. If this cannot be trusted, this ACE should **not** be ~~included which~~ **included**. This has the side ~~effect~~ **effect** of disabling automatic population of the {FCFs} set, thus requiring that ~~the~~ **set** to be populated administratively.

NOTE 29 – This ACE may result in scalability issues in some deployments. Given n FCFs in a ~~Fabric~~ **network**, this expands to n squared entries, which may exceed the ACL capability of ~~the~~ **a** bridge. In this case it ~~be-~~ **comes** ~~is~~ impractical to ensure that FCFs are not sourcing these frames when ~~SPMA is~~ **SPMAs** ~~are~~ in use.

C.4.4 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:

- a) Verify that all FIP frames are either sourced by or destined to an FCF;
- b) 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 30~~note 30); **and**
- c) Verify that all FCoE frames or either sourced by are destined to an FCF.

NOTE 30 – This protection is practical with FPMA only.

A set of ACEs that accomplish this **are**:

```
DA=ALL_FCF_MACSAll-FCF-MACs, Type=FIP_TYPE, permit;
DA={FCFs}, Type=FIP_TYPE, permit;
SA={FCFs}, Type=FIP_TYPE, permit;
DA=All-ENode-MACsAll-ENode-MACs, Type=FIP_TYPE, permit -- see note 31;
Type=FIP_TYPE, deny;
DA={FCFs}, SApr=FC_MAPFC-MAP, Type=FCoE_TYPE, permit; -- FPMA only
SA={FCFs}, DApr=FC_MAPFC-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;
```

NOTE 31 – This ACE should only be included if it is administratively known and trusted that only FCFs are able to inject frames destined to All-ENode-MACs onto the network. Including this ACE enables automatic population of the {FCFs} set. If this cannot be trusted, this ACE should not be included. This has the side effect of disabling automatic population of the {FCFs} set, thus requiring that set to be populated administratively.

C.4.5 Additional FCF ~~crosstalk~~-protection

The preceding ACL recommendations have been based on the set of ~~FCF-FCF~~-MAC addresses, {FCFs}, that contain all ~~FCF-FCF~~-MAC addresses including MAC addresses used for VE_Ports, VF_Ports, and the ~~FCFs'~~ FCoE ~~Controllers~~Controllers of the FCF-MACs. ~~It is possible that this~~ This MAC address set ~~could~~ may be subdivided into ~~their~~ subsets for 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~~ VF_Ports may be constructed. While the details of achieving this ~~is~~ are beyond the scope of this annex, this possibility may be considered for individual implementations and deployments.

C.5 Prevention of FCoE related traffic

It may be desirable to prevent the reception of all FCoE related traffic on a given bridge ~~port~~port (e.g., on bridge ports connected to links that are not known to be Lossless). To do this, all frames with an Ethertype of FCoE_TYPE or FIP_TYPE ~~must~~ have to be denied. Additionally, it is desirable to deny all frames using any ~~VN_Port's~~ VN_Port MAC address as 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_MAPFC-MAP, deny; --- Note: applies to FPMA only
```

C.6 Automatic configuration of ACLs

An Ethernet bridge may choose to ~~snoop~~examine FIP frames from which all, or most in the case of ~~SPMA~~SPMAs, the information needed to automatically configure ~~these~~ ACLs may be determined.

The set of ~~FCF-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~~All-FCF-MACs group address. Preventing this ~~was considered~~ is 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~~VE_Ports from being formed with such hosts.

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

The post FLOGI/FDISC ACEs may be constructed by examining the FIP FLOGI ~~Accepts~~LS_ACC / FIP NPV FDISC ~~Accepts~~LS_ACC messages transmitted ~~from~~ by the FCFs. These ~~messages~~ contain the ~~FCF~~ assigned VN_Port 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~~LS_ACC / FIP NPV FDISC ~~accepts~~LS_ACC messages transmitted ~~from~~ by the FCFs, ~~if~~ it is determined that an ENode has moved from one port to another.

Finally, the FIP Clear Virtual ~~Link-Links~~ 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.~~

The list of FCF-MAC addresses may be configured administratively. Doing so with the ACEs provided in this annex prevents a rogue host from impersonating an FCF.

The ACEs provided in this annex have soft state (i.e., a bridge should remove them if FIP Keep Alive messages and Discovery Advertisements are not seen on a periodic basis).

C.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-network~~ vulnerable to certain Ethernet learning attacks. In such implementations, Static Forwarding Entries (see IEEE 802.1Q-2005) 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 ~~addressaddress is also created~~. This entry should specify "forward" for the port on which the FIP FLOGI / FIP NPIV FDISC is received, and "filter" for all other ports.

C.8 VLAN considerations

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

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 ~~annexstandard~~.