Technical Report

Ethernet Storage Best Practices

David Klem, Mike Worthen, Frank Pleshe, NetApp

January 2013 | TR-3802

TABLE OF CONTENTS

1 INTRODUCTION

Ethernet storage has existed for many years in the form of Network Attached Storage (NAS). Under this guise, there has always been the attitude that the Internet Protocol (IP) and Ethernet networking for storage systems is simple and can be left to the networking team. In the past, for smaller or less critical environments, this was acceptable. Today, Ethernet based storage systems are faster and much more powerful, servicing the needs of large critical applications and databases over 10 Gigabit Ethernet (10GbE) network infrastructures. For Ethernet based storage systems serving mission critical environments, it is essential that consideration be given to the network design based on the demands of the applications and the nature of the storage protocols serving them. This fundamental concept is critical to ensure performance, scalability, latency and redundancy requirements are met.

Ethernet storage systems have become a continuously increasing source of network traffic, requiring design considerations for maximizing performance of large servers to large storage systems. This new concept is quite different from a network catering to thousands of clients and servers connected across LAN and WAN networks. Proper Ethernet storage network designs can achieve the performance of Fibre Channel networks, provided that technologies such as jumbo frames, Virtual Interfaces (VIFs), Virtual LANs (VLANs), IP MultiPathing (IPMP), Spanning Tree Protocol (STP), Port Channeling and multilayer topologies are employed in the architecture of the system.

By not considering the specific requirements of Ethernet Storage Networking, the success of the platform or project is left to chance. One NetApp customer recently said, "Our CEO thinks the Ethernet attached storage is unreliable, but it is almost always the network". The network and the Ethernet storage systems, as independent technologies, are highly reliable. The design of the infrastructure that integrates them is critical to realizing this reliability.

This document outlines the best practice recommendations for building enterprise class Ethernet storage networks. It provides requirements of the networking infrastructure along with proposed designs to best meet these requirements.

2 USING VLANS FOR TRAFFIC SEPARATION

In the past, standard Ethernet hubs were used to aggregate servers, storage, and other devices on a network. By definition, all devices connected to a single hub were part of both the same collision and broadcast domain. A collision domain refers to a physical Ethernet segment including all cables, interfaces and network hardware which are apart of the same Ethernet signal timing region. If two or more devices transmit at the same time within this region, a collision will occur. In the event of a collision, systems report and discard the colliding frames then retransmit. As networks were created with high-density hubs and those hubs were interconnected to each other the timing region of the Ethernet segment was large. Large collision domains often caused network to run inefficiently and not realize the true bandwidth potential of the segment. Ethernet switches were introduced into networks to isolate timing and error correction in an effort to increase the efficiency of a device’s physical connection to the network by deploying a switch an administrator effectively isolates devices into their own collision domain which will reduce errors and increase the efficiency the network. While the Ethernet switch isolates collisions it does not isolate broadcasts.

While a collision domain is a physical division of a Ethernet segment, a broadcast domain is a logical isolation of the Ethernet segment. A broadcast is an important function of a Ethernet segment as it is the means to discover the Ethernet address of a device which is not known by a transmitting device in the same broadcast domain. This function ensures that devices do not need to store the Ethernet addresses of their neighbor but in the event they need to communicate with a neighbor, they simply transmit a broadcast frame asking to discover the address of a device. The intended device within the broadcast domain will respond with the address and data transmissions will begin. Large broadcast domains produce inefficiencies in network communications similar to how large collision domains did. However, the separation of broadcast domains typically required separate physical equipment and dedicated ports on large routers. Isolation of broadcast domains required investment in both hubs and router interfaces as the only means to connect two independent broadcast domains is through the use of a router. Many organizations did not isolate broadcast domains because of the costs of routers and router interfaces that would route traffic between the broadcast domains.

In the mid 90s, speeds of networks began to increase and the technology for routing between broadcast domains was introduced to switches. These switches were referred to as “layer 3 switches” and they simply provided the ability to isolate a port into its own collision domain while also allowing the administrator to assign ports to different broadcast domains while also providing the routing feature set required to route between the broadcast domains within the network.

With the introduction of this technology it now became possible to logically define port groupings by function as opposed to grouping by location.

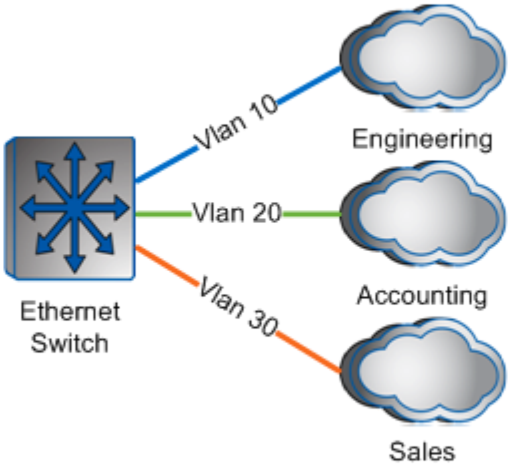

Figure 2-1) Logical VLAN separation by function

Because VLANs operate at the Data Link Layer (L2) of the OSI Model, traffic will be completely isolated between VLANs unless a bridge or a router (L3) is used to connect the networks together. Typically, a one-to-one relationship exists between an IP subnet and a VLAN. While there are exceptions, following this general rule will help simplify the network and ease in management.

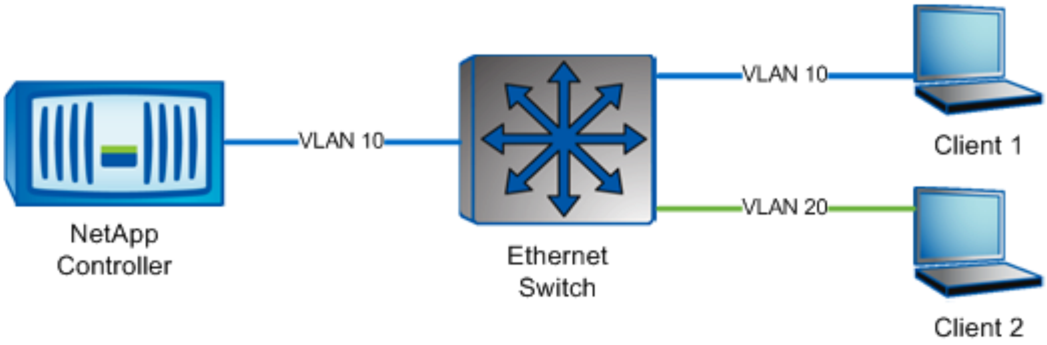

Figure 2-2) VLAN Description

Implementing VLANs provides many advantages within the network.

- Higher utilization of switch infrastructure. By allowing separate logical devices to live on the same physical switch, this eliminates the need for a completely separate hub or switch for each network. Higher density, more efficient switches can be used to aggregate devices.
- Lower management costs. Moving physical devices or cables no longer becomes required, as an administrator can logically assign devices from the management console of the switch to different networks if required.
- Security and stability of the network. Because a L3 device is required to communicate between VLANs, L3 access lists can be applied to prevent communication between certain networks. Broadcast storms and the effects of unicast flooding become isolated to a single VLAN. A malicious user with a packet capture tool will be unable to intercept traffic not destined for that user's host.

2.1 VLAN TRUNKING

Organizations typically require the use of multiple switches for redundancy and port density. Often times a logical VLAN might need to exist across multiple switches. With the standard access port configuration of only allowing a single VLAN to be defined on a port, multiple ports each assigned to a single VLAN would be required to pass VLANs across switches. This method does not scale well and is highly inefficient. The IEEE 802.1q standard provides a solution to this problem with a feature called VLAN Trunking.

VLAN Trunking allows a single link to carry multiple VLANs by "tagging" each packet with a 4-byte tag. This tag defines which VLAN each packet is assigned to as it travels throughout the network between VLAN- aware devices. Common VLAN-aware devices are network switches, routers, certain servers, and NetApp Storage Controllers. When the frame reaches an end-point or access port, the VLAN tag is removed and the original frame is then sent to the end device. The tag is simply a forwarding instruction to ensure that the frame is delivered to the proper broadcast domain or VLAN.

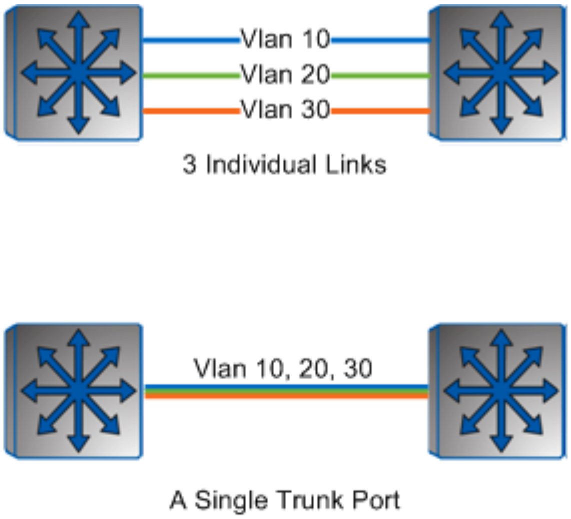

Figure 2-3) Vlan Trunking

One important concept used within 802.1q trunking is the *native VLAN feature*. Frames assigned to the VLAN configured as the native vlan will be sent untagged across the trunk link. All other VLANs that are configured in the trunk will be tagged with their respective VLAN IDs. For this reason, it is very important that the native vlan be configured the same on both ends of the connection. Improper configuration of the native vlan can result in limited or no connectivity for the administratively defined native VLANs.

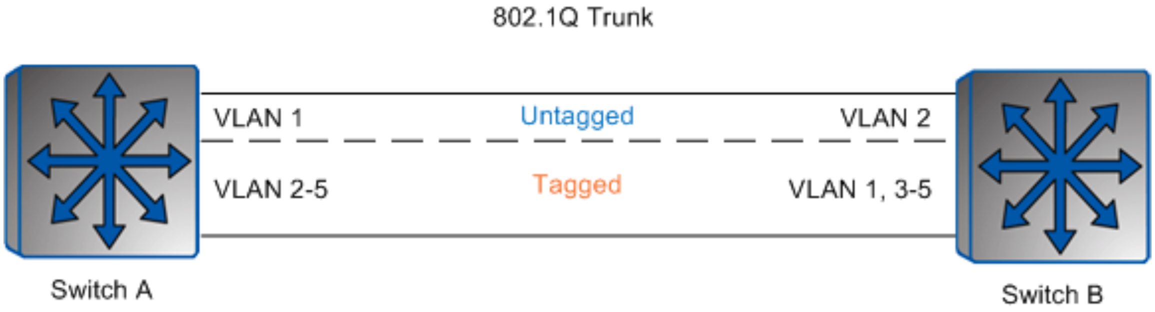

Figure 2-4) Native Vlan Misconfig

As stated earlier, NetApp Controllers also provide the ability to configure multiple VLANs and VLAN Trunking. This allows for greater flexibility and configuration options within the controller itself. For instance, a NetApp Controller with a 10Gb interfaces might need to serve multiple functions such as iSCSI boot traffic as well as standard NAS traffic. The iSCSI boot network might require additional security and control, but because of the relatively low traffic requirement and administrator might not want to dedicate an entire 10Gb link to this function. By implementing VLAN Trunking on the NetApp Controller and the switch, that single 10Gb link can be shared between the two functions while still maintaining isolation between VLANs.

When configuring VLANs on a NetApp controller, be aware that Data OnTap does not currently use the native VLAN feature. For this reason, the native VLAN on the switch port, which a NetApp controller is connected, must be assigned to an unused VLAN to ensure data is forwarded correctly.

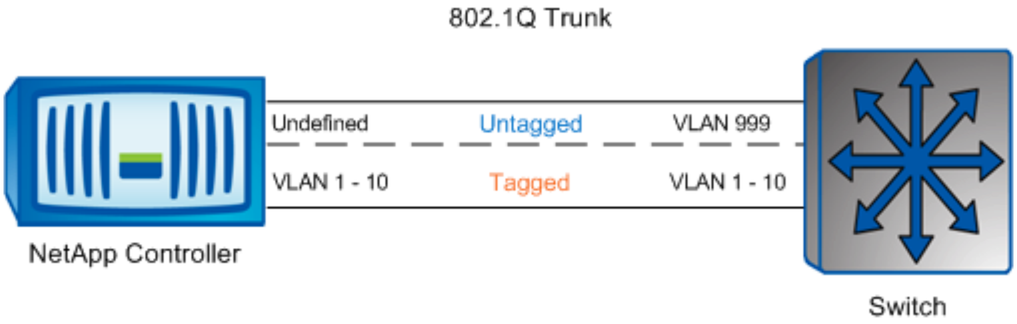

Figure 2-5) Native Vlan Netapp Config

Configuration Template – VLAN Trunking

NETAPP RC FILE

```
vlan create e0a 10 20 30
ifconfig e0a-10 192.168.10.10 netmask 255.255.255.0
ifconfig e0a-20 192.168.20.10 netmask 255.255.255.0
ifconfig e0a-30 192.168.30.10 netmask 255.255.255.0
```

CISCO CONFIG

```
interface GigabitEthernet1/1
 description NetApp e0a Trunk
 switchport mode trunk
 switchport trunk allowed vlan 10,20,30
 switchport trunk native vlan 123
 flowcontrol receive on
 no cdp enable
 spanning-tree guard loop
 spanning-tree portfast
end
```

2.2 VLAN RECOMMENDATIONS

It is critical that the network infrastructure and the traffic flowing over it be well understood. The use of VLANs can benefit Ethernet Storage networks by providing an effective mechanism to segment network traffic and at the same time provide better utilization of network switching resources. When configuring the network for VLANs remember the following key items:

- Ensure the Native VLAN is properly configured on both ends of switch connections, if those switches support native VLANs
- NetApp Controllers connected to a switch which supports native VLANs should have the native VLAN on that port assigned to a VLAN which will not be used on the NetApp controller
- Utilize VLAN trunking to allow multiple VLANs to travel over a single high speed interface such as 10GbE

3 PREVENTING LOOPS WITH SPANNING TREE

3.1 PROTOCOL INTRODUCTION

The Spanning Tree Protocol (STP) provides a mechanism for bridges to dynamically discover a topology that is loop-free. It provides just enough connectivity so that where physically possible, there is a path between every LAN segment. Ethernet Storage Networks that are designed with high availability in environments where mixtures of 10Gbps and 1Gbps interfaces are deployed must pay particular attention to the Spanning-Tree topology of your network. NetApp engineering is often engaged to troubleshoot Ethernet Storage performance problems found to be related to traffic flow caused by a network topology deployed without adequate attention to the spanning-tree environment.

In the following section we will refer to a bridge. A bridge is traditionally recognized as a physical device but in more recent years with switches being capable of running multiple VLANs a bridge will exist in software as a single VLAN. As the text refers to a bridge also consider this to be a process that is executed by each VLAN on each device which is configured to operate a particular VLAN.

A spanning-tree topology is calculated through bridges transmitting special messages to each other that allow them to calculate the spanning tree. These special messages are given the name *configuration Bridge Protocol Data Units* (configuration BPDUs).

NOTE: In a switch that is configured with 100 VLANs a spanning-tree topology must be constructed through this discovery process for each of the 100 VLANs.

Bridges exchange configuration BPDU messages to perform the following procedure.

- The bridges elect a single “root” bridge.
- Each bridge calculates the shortest-path distance to the root bridge. The port that provides this path is labeled the “root port”.
- Each LAN reviews the distance to the root of each of its bridges and selects the closest, labeling this the "designated bridge". The designated bridge will forward frames from that LAN toward the root bridge.

- Finally, all bridge ports that are not root ports or designated bridge links are disabled.

In environments where multiple VLANs are employed, Spanning Tree Protocol is run separately for each VLAN.

3.2 ELECTING A ROOT BRIDGE

The first step in creating a loop-free topology is electing a root bridge of the spanning tree. Imagine two physical switches running a single VLAN. When an Ethernet cable is used to connect the switches, the two ports immediately exchange configuration BPDUs. These messages contain the *local priority* and *MAC address.* The local priority is a user-configured value used to influence the election of the root bridge. The bridge with the smallest priority will be selected as the root. NOTE: Choosing the smallest MAC address will break Ties. Switches typically have a default priority and thus two switches that have not been configured from the same manufacturer will advertise the same priority. If an administrator would like to force a particular switch to be the root bridge, then that administrator would manually specify a priority lower than that of any other switch in the network.

There are many modern methods for connection switches and the spanning tree topology will be discovered differently in each method. The following are a few examples.

Connection	Spanning Tree implications
A **single cable** connects two ports configured for a **single VLAN**	The root bridge is elected normally for the given VLAN.
A **single cable** connects two ports configured for **VLAN trunking**	The root bridge is elected normally for the each VLAN on the trunk. It is possible to have one VLAN rooted on one switch and another VLAN rooted on a different switch.
Multiple cables connect ports configured for a **single VLAN**	The root bridge is elected normally for the given VLAN. The redundant links will be detected as a loop and only the lowest port ID (the root port) will be enabled.
Multiple cables connect ports configured for **VLAN trunking**	The root bridge is elected normally for the each VLAN on the trunk. The redundant links will be detected as a loop and only the lowest port ID (the root port) will be enabled.
Multiple cables connect ports configured for a **single VLAN** with **port bonding (Etherchannel)**	The root bridge is elected normally for the given VLAN, and all links will be employed together as one logical link. See section 4 for details on port bonding.
Multiple cables connect ports configured for **VLAN trunking** with **port bonding (Etherchannel)**	The root bridge is elected normally for the each VLAN on the trunk, and all links will be employed together as one logical link. See section 4 for details on port bonding.

3.3 PATH TO ROOT SELECTION

Once a root bridge has been elected and the spanning tree discovered, the next step is to use this information to determine the best paths and to block detected loops. The “best path” is the one with the fewest, fastest links to the root bridge.

In figure 1.1 we detail 3 switches. Switch B is the root bridge and connects to Switch A and Switch C with 1Gbps connections. Switch A and C also have a connection to each other which is 10Gbps. This 10Gbps connection creates a loop condition in the network, which requires the loop to be eliminated. The spanning tree topology is a tree and thus the root of the tree is the root bridge. Switch B is the root so the 1Gbps connections to the other switches are the ports, which are forwarding. The 10Gbps connection between A and C will be blocked because it is not the most direct path.

In figure 1.2 we detail the same 3 switches yet this time there are multiple links connecting Switch A & C to the root bridge B. The multiple links on the path to the root are loops in the topology. When there are loops in the topology on a path to the root then the fastest speed link is forwarded while the slower speed link is blocked.

3.4 FAST START MECHANISMS

Since Spanning Tree calculations can take several seconds to put a given port into a forwarding state, network equipment manufacturers like Cisco have created “Fast Start” mechanism to get traffic moving across network connections as quickly as possible. Features such as Portfast, UplinkFast and BackboneFast provide “FAST START” mechanisms for ports in various locations throughout the network.

PortFast

Spanning-tree PortFast causes a spanning-tree port to enter the forwarding state immediately, bypassing the listening and learning states. You can use PortFast on switch ports connected to a single workstation or server to allow those devices to connect to the network immediately, rather than waiting for spanning tree to converge.

UplinkFast

UplinkFast provides fast convergence after a spanning-tree topology change and achieves load balancing between redundant links using uplink groups. An uplink group is a set of ports (per VLAN), only one of which is forwarding at any given time. Specifically, an uplink group consists of the root port (which is forwarding) and a set of blocked ports, except for self-looping ports. The uplink group provides an alternate path in case the currently forwarding link fails.

BackboneFast

Backbone fast is a Cisco proprietary feature that, once enabled on all switches of a bridge network, can save a switch up to 20 seconds (max_age) when it recovers from an indirect link failure.

PVST+ (PER VLAN SPANNING-TREE)

The basic Spanning Tree protocol can be very slow and time consuming to get all the network ports into their correct state of operation. As networks have evolved, the need for immediate startup or failover to redundant links has driven the evolution of the Spanning Tree protocol operation. Per VLAN Spanning Tree (PVST)+ is based on IEEE802.1D standard and includes Cisco proprietary extensions such as BackboneFast, UplinkFast, and PortFast. Rapid-PVST+ is based on IEEE 802.1w standard and has a faster convergence than 802.1D. RSTP (IEEE 802.1w) natively includes most of the Cisco proprietary enhancements to the 802.1D Spanning Tree, such as BackboneFast and UplinkFast.

CISCO CONFIG

```
interface GigabitEthernet1/1
 description NetApp e0a Trunk
 switchport mode trunk
 switchport trunk allowed vlan 10,20,30
 switchport trunk native vlan 123
 flowcontrol receive on
 no cdp enable
spanning-tree guard root
spanning-tree guard loop
spanning-tree portfast
end
```

3.5 SPANNING-TREE RECOMMENDATIONS

Understanding the basics of how SpanningTree works is essential to knowing how traffic flows across your network. An improperly configured layer 2 Spanning Tree network can send traffic across additional links increasing latency. In the worst situations spanning-tree could send traffic over slower speed links, which can cause congestion and dramatically affect performance. The key elements to optimize network performance through a properly design Spanning Tree are:

- A complete understanding of the network topology and how traffic is supposed to flow for each Ethernet storage VLAN
- A understanding of how upstream network link failure will produce a spanning-tree event and ensure that this event does not impact a users perception of storage controller failover times
- Utilization of “fast start” mechanisms to ensure lowest possible time for ports to reach a forwarding state

4 IMPROVE PERFORMANCE AND RESILIENCY WITH PORT BONDING

Port bonding methods are available on both the network and storage components to aggregate multiple physical links into a single virtual interface. A load balancing algorithm distributes traffic across the physical links in the channel to provide additional performance and redundancy within the network.

The terminology for this technology sometimes differs between NetApp and the networking industry, which can cause confusion. While the network industry uses terms such as EtherChannel or port-channel, NetApp calls these Multi-Mode VIFs, or virtual interfaces. An incorrect term that’s sometimes used is “trunked interfaces” or “trunk”. The term “trunk” or “trunking” should be avoided, since the networking industry already uses the term trunk to describe VLAN trunking.

NetApp provides three types of VIFs, each with their own benefits and requirements. In many cases it will be recommended that different VIF types be used together to attain higher levels of redundancy.

- Single-mode VIF
- Static Multi-mode VIF
- Dynamic Multi-mode VIF (LACP)

4.1 SINGLE MODE VIF

A single-mode VIF contains a single active link and a number of links in passive mode. All links in the single-mode VIF share a single MAC address. Only one link is active and sending data therefore there is never a duplicate MAC address on the network.

In the example shown below, a single-mode VIF is configured with two links connected to redundant switches. Only link e0 is active, while e1 is in standby mode. If e0 fails (signaled by the link status of the interface going down), the remaining link will take over. In the case of multiple standby interfaces, the storage device will randomly pick one to bring up as the active link. This is commonly known as active-passive mode.

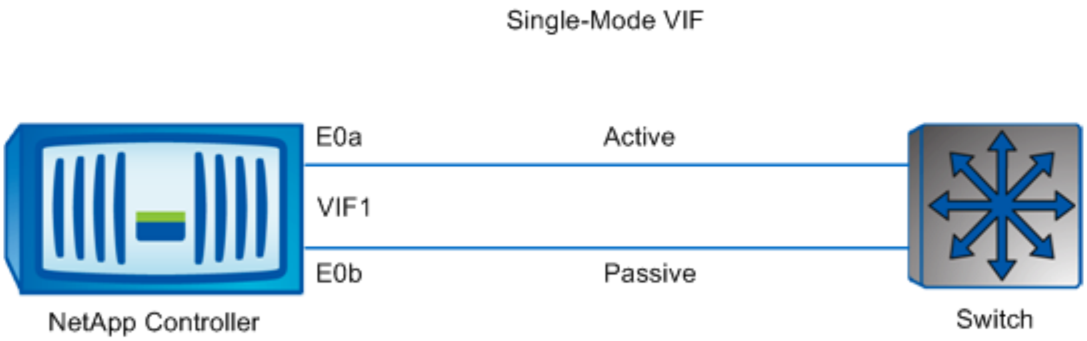

Figure 4-1) Single Mode Vif

By default, Data ONTAP will randomly select an interface in the single mode VIF to become active. In many cases, administrators may prefer to designate which interface will become active by using the "**`vif favor`**" command. This will eliminate the random selection and cause the device to always choose that particular link if available. Similarly, the "`vif nofavor`" command can be used to ensure that a particular port is *not* chosen in the random selection, unless it is the last port in the single mode VIF.

PROS

- No switch configuration necessary to support a single mode VIF from the filer side.
- Provides 1-1 redundancy with no decrease in bandwidth in the event of a failover.

CONS

- Requires twice the number of switch ports, and only utilizes half the bandwidth allocated by switch ports as a number of interfaces are functioning in passive mode.
- The passive interface shows a status of "down" from a management perspective, which means the administrator is unable to monitor failures on the passive links for preventative maintenance.
- Often, failover is never tested within the environment to ensure proper function during a link outage.
- If Spanning-Tree fast start is not used for on storage controller ports a spanning-tree calculation is required to process before the port will actively forward traffic

Configuration template – Single Mode VIF

Below are example configuration files for both the NetApp storage controller and a Cisco IOS-based switch.

NETAPP RC FILE

```
vif create single template-vif1 e0a e0b

vif favor e0a

ifconfig template-vif1 10.1.1.100 netmask 255.255.255.0 mtusize 1500
partner 10.1.1.200

route add default 10.1.1.1

routed on

options dns.domainname template.netapp.com

options dns.enable on

savecore
```

CISCO IOS SWITCH

```
interface GigabitEthernet1/1
 description NetApp e0a
 switchport access vlan 116
 switchport mode access
 flowcontrol receive on
 no cdp enable
 spanning-tree guard loop
 spanning-tree portfast
!
interface GigabitEthernet1/2
 description NetApp e0b
 switchport access vlan 116
 switchport mode access
 flowcontrol receive on
 no cdp enable
 spanning-tree guard loop
 spanning-tree portfast
end
```

4.2 STATIC MULTI-MODE VIF

A static multi-mode VIF contains two or more interfaces in which all links are active. This follows the 802.3ad (static) IEEE standard, and requires support and configuration on the switch. However, neither LACP nor PAgP are support in this configuration, so no negotiation or auto-detection of the aggregated ports occurs. The switch must be configured in "on" or "static" mode, which forces the interfaces to form a channel.

Static multi-mode VIFs are able to continue operating even if all but one link has been lost. This allows for higher throughput than a single mode VIF, and still provides redundancy. Multiple methods of load balancing are available for outgoing packets from the NetApp FAS device, including:

- Source and Destination IP Hash,
- Source and Destination MAC Address Hash, and
- Round Robin.

The sending device always determines which link is used to send traffic. Because of this, different settings can be used on both the switch and the end-device, which can result in uneven traffic distribution for both transmit and receive directions. Because of this, the load balancing hash algorithms should match as closely as possible in both directions.

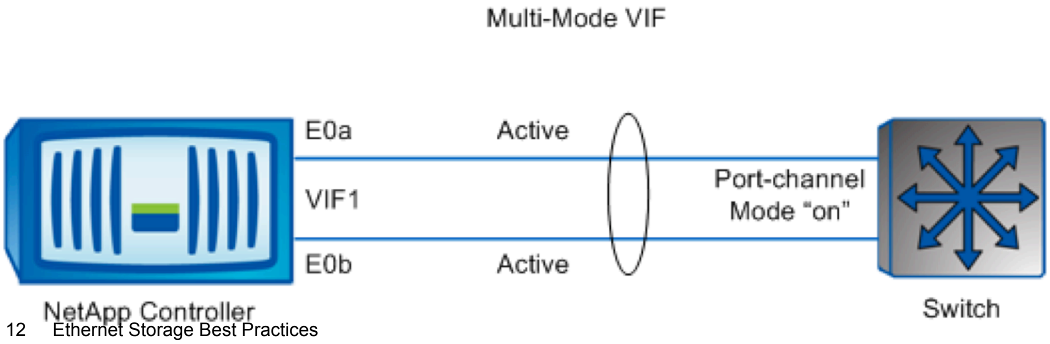

Figure 4-2) Static Multi-Mode VIF

PROS

- Allows for higher aggregate bandwidth based on the load-balancing algorithm chosen, since all ports in the channel are active
- Can detect a loss of link on either side of the connection and fail traffic over to the remaining link. This allows for greater redundancy without wasting ports on an active/passive configuration.

CONS

- Static mode forces the channel up, which can potentially lead to problems such as the traffic black hole example mentioned earlier.
- Because no negotiation occurs between the devices, static mode results in fewer log and error messages, which can make it difficult to troubleshoot.
- Achieving even load distribution across the links requires the use of multiple source and destination address pairs and proper load-balancing algorithm selection. Additional configuration on the NetApp FAS device (such as IP aliases, discussed in section XXXXXXXX) may be needed to reach the optimal distribution.

Configuration Template – Static Multi-mode VIF

NETAPP RC FILE

```
vif create multi template-vif1 –b ip e0a e0b

ifconfig template-vif1 10.1.1.100 netmask 255.255.255.0 mtusize 1500
partner 10.1.1.200 flowcontrol send

route add default 10.1.1.1
```

CISCO IOS SWITCH

```
interface GigabitEthernet1/1

 description NetApp e0a

 switchport access vlan 100

 switchport mode access

 flowcontrol receive on

 no cdp enable

 spanning-tree guard loop

 channel-group 5 mode on

!

interface GigabitEthernet1/2

 description NetApp e0b
 switchport access vlan 100
 switchport mode access
 flowcontrol receive on
 no cdp enable
 spanning-tree guard loop
 channel-group 5 mode on
!
interface Port-channel5
 description NetApp template-vif1
 switchport
 switchport access vlan 100
 switchport mode access
 flowcontrol receive on
 no cdp enable
 spanning-tree guard loop
end
```

4.3 DYNAMIC MULTI-MODE VIF

Dynamic multi-mode VIFs are based on Link Aggregation Control Protocol (LACP) as described by the IEEE 802.ad (dynamic) standard. They are similar to static multi-mode VIFs in that they contain two or more active interfaces, share a single MAC address, and require configuration on both ends of the connection. In a dynamic multi-mode VIF, however, additional negotiation parameters are passed between the devices using an LACP Protocol Data Unit (PDU). This allows for the two connected devices to dynamically add and remove links from the channel for reasons other than physical link failure. This is an important distinction, because dynamic multi-mode VIFs can not only detect and compensate for a loss of link, but also a loss of data flow. This leads to higher availability and can help prevent the packet black hole problem discussed in the single-mode VIF section.

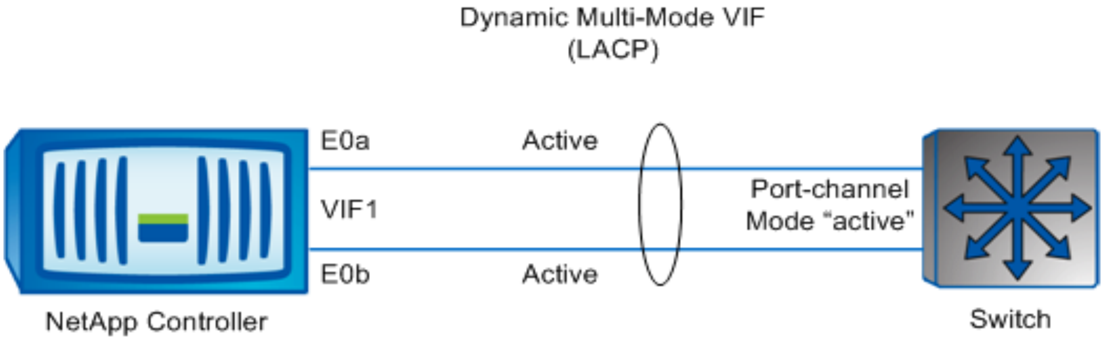

Figure 4-3) Dynamic Multi-Mode Vif (Lacp)

PROS

- Allows for higher aggregate bandwidth based on load-balancing algorithm chosen, since all ports in the channel are active.

- Because of the LACP PDUs used, not only can a dynamic multi-mode VIF detect a loss of link on either side, it can also detect a loss of data flow. This avoids the queue wedge / traffic black hole example explained earlier.
- Many newer switches support multi-chassis LACP, which eliminates the need for a second level single-mode VIF for redundancy across switches.

CONS

- Older switches might not support the LACP standard.
- Older switches might not support multi-chassis LACP, which would require a second level single-mode VIF for switch redundancy. (Cisco supports multichassis etherchannel in LACP or Static so we should mention Multichassis in static also)
- Achieving even load distribution across the links requires the use of multiple source and destination address pairs and proper load-balancing selection. Additional configuration on the NetApp FAS device (such as IP aliases, discussed later in this document) may be needed to reach the optimal distribution.

Configuration Template – Dynamic Multi-mode VIF

NETAPP RC FILE

```
vif create lacp template-vif1 –b ip e0a e0b

ifconfig template-vif1 10.1.1.100 netmask 255.255.255.0 mtusize 1500
partner 10.1.1.200 flowcontrol send

route add default 10.1.1.1
```

CISCO IOS SWITCH

```
interface GigabitEthernet1/1
 description NetApp e0a
 switchport access vlan 100
 switchport mode access
 flowcontrol receive on
 no cdp enable
 spanning-tree guard loop
 channel-group 5 mode active
!
interface GigabitEthernet1/2
 description NetApp e0b
 switchport access vlan 100
 switchport mode access
 flowcontrol receive on
 no cdp enable
 spanning-tree guard loop
 channel-group 5 mode active
!
interface Port-channel5
 description NetApp template-vif1
 switchport
 switchport access vlan 100
 switchport mode access
 flowcontrol receive on
 no cdp enable
 spanning-tree guard loop
end
```

4.4 VIF PERFORMANCE

Performance benefits are often cited as a reason to create and deploy VIFs in a storage environment. While it is true that VIFs can increase performance, there are many considerations that must be taken into account to take full advantage of them.

When creating a multi-mode VIF, a load-balancing algorithm is used to determine which link of the VIF will be used to send a particular traffic flow. Data ONTAP supports three different load-balancing methods for both static and dynamic multi-mode VIFs: *round robin*, *source/destination IP*, and *source/destination MAC*. For each mode, an *exclusive-or* (XOR) combination of the last byte of both the source and destination address are used to determine which link of the VIF to use for sending traffic, based on the following formula:

(SourceAddress XOR DestinationAddress) % NumberOfLinks

One common misconception is that a VIF containing N number of 1Gb links will have N*1Gb of bandwidth available. While in theory this is true with perfect frame distribution, it is not typically the case in practice due to the hashing method used. With large numbers of source and destination pairs, IP or MAC address-based hashing can approximate an even distribution. However, a single transmission (source and destination pair) can only transmit up to the speed of *one* of the physical links in the channel. Take the following example, which illustrates how VIFs may not provide any performance benefit for smaller numbers of source and destination pairs.

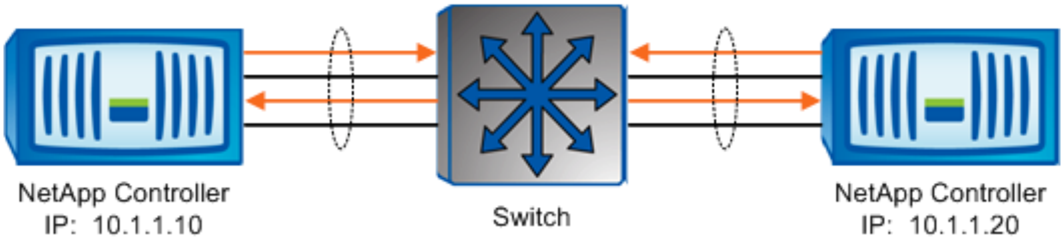

Figure 4-4) VIF Load Balancing

In this case, each FAS device has a single IP assigned and a 2-port VIF created. When initiating any type of data transfer between the two devices (such as a SnapMirror operation), the hash will always calculate the same value, causing all traffic to flow over the same link. This means for a one-to-one connection, no performance benefit is realized by adding links to the VIF. Depending on the IP address and MAC address schemes used, it is possible to get an uneven distribution across VIFs. Some trial and error might be required to determine the hashing algorithm best suited for the environment.

Another important distinction is that the device sending the traffic determines the frame distribution for each hop. In the example above, the sending filer determines the hash across the first VIF. Then, when that data traverses the links to the receiving filer, the switch actually determines the hash. This can get quite complicated if different hashing (frame distribution) algorithms are used throughout a large network.

4.5 VIF LOAD BALANCING

ROUND ROBIN

Round Robin was one of the first load balancing algorithms developed and implemented by the switch vendors. It essentially alternates Ethernet frames over the active links in the channel regardless of the source and destination mac or ip address. This provides a very even distribution across all links in the channel, but introduces a major problem with out of order packet delivery. Frame 1 sent across link 1 might arrive at the destination later than frame 2 sent over link 2. This causes the higher-level protocols and applications to recover, often resulting in a retransmission of the frames. Since this is highly inefficient, using round robin as a load-balancing algorithm is typically not recommended.

SOURCE AND DESTINATION MAC ADDRESS

Source and destination MAC address hashing is the least common algorithm used because it is likely that a disproportionate amount of traffic will rely on a single link. This algorithm performs an XOR calculation on the source and destination pair of the MAC addresses in a packet. If the source and destination both reside on the same subnet or VLAN, this method can work well. However, when traffic must pass through a router, the same host-to-router path will be used for all flows, regardless of the destination beyond the router. The following example demonstrates why this occurs.

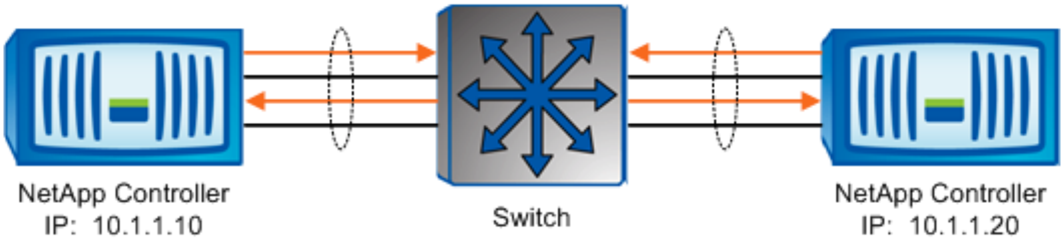

Figure 4-5) Round Robin

- Host1's IP address is 10.1.2.10/24 (Host1's default router is 10.1.2.1)
- Controller1's IP address is 10.1.2.10/24 (Controller1's default router is 10.1.2.1)

The host and filer defined above are located on two separate subnets. The only way these devices can communicate with each other is through the router, whose purpose is to route between L3 networks and suppress broadcasts. In the case of the example above, default router 10.10.1.1 and 10.10.3.1 are actually the same physical router, those addresses are simply two physical interfaces on the router.

As Host1 builds a frame destined for Controller1, it recognizes that 10.10.3.100 is an IP address not on its local network, so it forwards the frame to its default router.

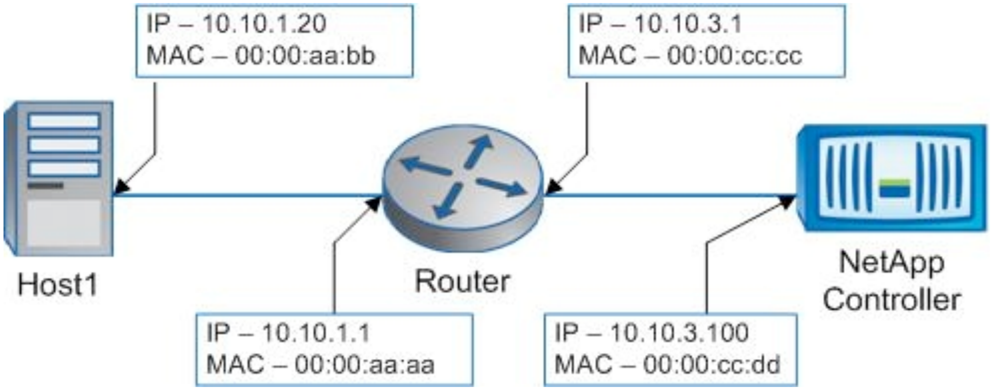

Figure 4-6) Traffic Flow

Host1 to Host1Router

- IP Source: Host1 (10.10.1.10)
- MAC Source: Host1
- IP Destination: VIFController1 (10.10.3.100)
- MAC Destination: Host1DefaultRouter

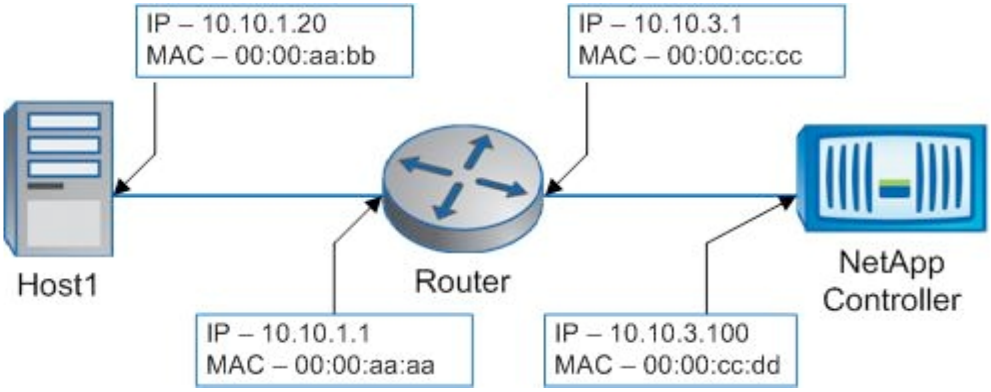

Figure 4-7) Traffic Flow

Host1Router Routing Packet to Controller1

- IP Source: Host1 (10.10.1.10)
- MAC Source: Controller1DefaultRouter
- IP Destination: VIFController1 (10.10.3.100)
- MAC Destination: VIFController1

The source and destination MAC addresses changed as the frame was forwarded through the network, but the source and destination IP stays the same. This impacts the effectiveness of MAC address based hashing algorithms: for packets originating on the NetApp controller, the source MAC address will always be the controller VIF and the destination MAC address will always be the router, resulting in only a single link being used for load balancing.

To fully understand how this creates a problem, consider a 4-link 1Gbps VIF on Controller1 and an additional 50 hosts on the same subnet as Host1. Packets sent by the controller to any of these hosts will always have the same source and destination MAC addresses, and therefore always use the same physical link.

SOURCE AND DESTINATION IP ADDRESS

IP Load-Balancing is the default setting for all NetApp multi-mode VIFs and is the most common type of multi-mode VIF in production today. The algorithm is no different than the MAC address algorithm defined above, except it is the last octet of the source and destination IP addresses that are hashed. This technique is more effective because the IP addresses never change as packets travel through the network. This means that unique pairs are more likely, and device-to-router links will not be restricted to a single link. This results in a more equal distribution of traffic across the physical links.

Note that because the last octet of the source and destination IP address are the only parts that are used in the load balancing calculation, devices that share a common last octet will end up hashing to the same physical link across a channel, even if they are in different subnets.

NOTE:

- With Data ONTAP versions up to 7.3.1 we perform an XOR calculation on the source and destination IP addresses.
- From 7.3.2 to 8.0.1 we use a hashing algorithm for port-based distribution and continue using the XOR calculation for source and destination IP addresses.
- Beginning with 8.0.1 we use a hashing algorithm for both IP and port based distribution.

4.6 IP ALIASING

Understanding load balancing algorithms allow an administrator to exploit them to provide more ideal traffic patterns. When load balancing by IP address hash, one can achieve better load distribution by assigning the VIF additional IP addresses. All NetApp VIFs and physical interfaces can be assigned multiple IP addresses; IP addresses beyond the first one are referred to as *IP aliases*. A typical recommendation is to match the number of addresses with the number of physical links in the port channel between the controller and the switch. Therefore, if a design calls for a 4-link 1Gbps multi-mode VIF between a NetApp controller and switch, assign the VIF its first IP address directly, then add three IP aliases.

Simply assigning additional addresses will not achieve more even traffic distribution. The hosts that transfer data from the NetApp controllers must utilize all of the addresses to fully take advantage of the IP-based load balancing algorithm. This can be achieved by a few different ways, depending on the protocol being used. Below are a few examples of NFS client configurations.

- **Oracle NFS**: Oracle hosts should mount NFS volumes by evenly distributing NFS mounts across the available controller IP address. If there are 4 different NFS mounts then mount them via the four different IP addresses on the controller. Each mount will have a different source and destination address pair and the communication from the host to controller will be efficiently utilized.
- **VMware NFS**: ESX hosts should mount each NFS data store via a different IP address on the NetApp controller. The host can use a single VMkernel interface (the source address) as long as the datastores are mounted with a different IP addresses on the controller. If there are more datastores than IP addresses, then simply distribute the datastore mounts evenly across the controller's IP addresses.

Finally, when administrators assign IP aliases to network interfaces on NetApp controllers that are configured for high availability, the IP alias will be moved to the partner controller in a down condition. The IP aliases do not need to be configured as a partner if the physical interface has already been partnered.

4.7 LAYERING VIFS

In some cases, additional redundancy may be required that cannot be accomplished by a traditional VIF configuration. With some switch vendors such as Cisco, features such as cross-stack EtherChannel, Virtual Switch System (VSS) or Virtual Port Channels (VPC) can allow a NetApp administrator to create a dynamic or static multi-mode VIF that spans multiple switches.

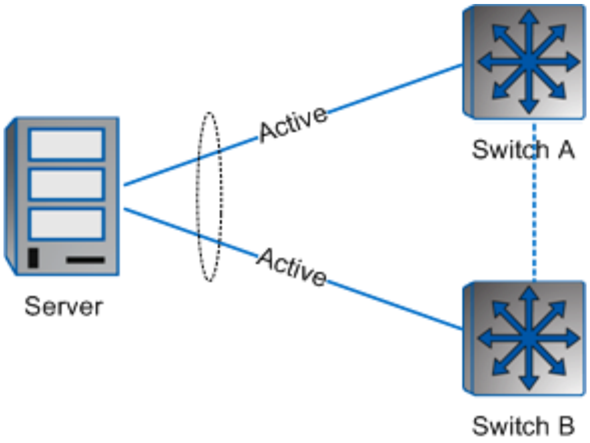

Figure 4-8) VSS Or VPC Example Diagram

For switches that do not support this feature, Data ONTAP allows for the layering of multiple VIFs, which is known as a second level VIF, for added redundancy. This allows the administrator to realize the performance gains of a multi-mode VIF, but still realize the redundancy of a single-mode VIF.

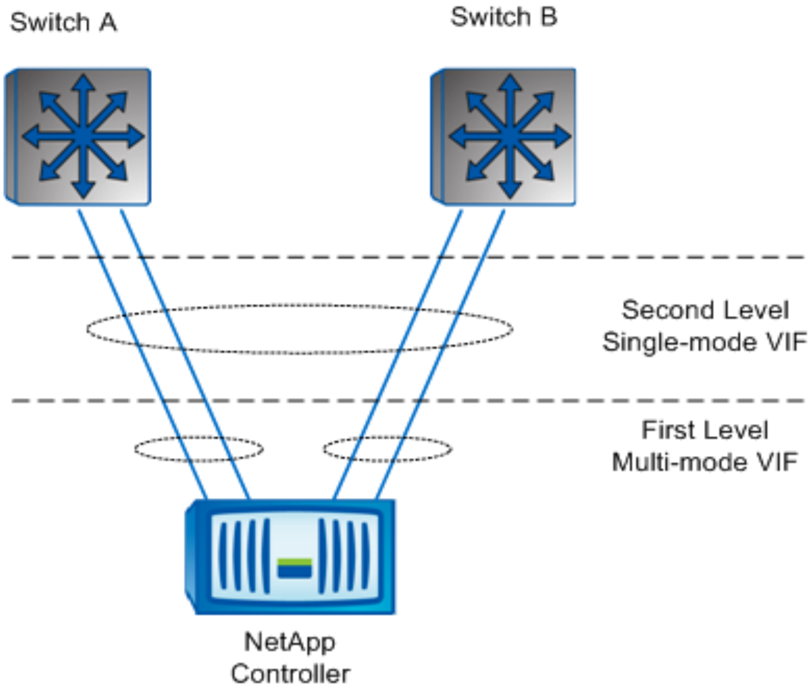

Figure 4-9) Layered VIF Example

Configuration Template – Layering VIFs

4.8 PORT BONDING RECOMMENDATIONS

SINGLE MODE VIF

Often, single mode VIFs are used to provide separate paths to redundant switches. This will preserve connectivity in case of a switch or uplink failure.

Many switch vendors now support what can be called “multi-chassis LACP”. This feature allows an end device to be configured as an LACP Ether Channel with members of the channel connected to redundant switches.

5 INCREASING PERFORMANCE WITH JUMBO FRAMES

By default, a standard Ethernet frame has a 1500 byte payload. Ethernet header and CRC checksum data adds 18 bytes for a total Ethernet frame size of 1518 bytes, or 1522 bytes with a VLAN tag (IEEE 802.3ac). Since the header and CRC checksum create a fixed amount of overhead per packet, efficiencies can be gained by sending a larger payload. By changing the MTU (Maximum Transmission Unit) from the standard 1500 bytes up to 9000 bytes, *jumbo frames* are created. Larger frames effectively reduce the number of packets processed for the same amount of data, which results in an increase in network throughput.

Care must be taken when implementing jumbo frames in an Ethernet storage network. Incorrect design or configuration can result in poor performance, and even limited or no connectivity at all. When configuring a NetApp FAS storage appliance for jumbo frames, the following elements must be properly configured:

- The FAS device’s network interface and any associated VIFs.
- The individual ports and port-channel interface, if applicable, on the switch.
- The VLAN and ports on all switches and layer 3 routers between the FAS device and its clients.

Ports with standard MTU size and jumbo MTU size should never be mixed on the same VLAN. Consider a host and a NetApp FAS device configured on the same VLAN, where the FAS controller is configured for jumbo frames and the host is not. The host will be able to communicate to the FAS device using its standard 1500 byte frames. However, the reply back from the FAS device will be a 9000 byte frame and because the two machines are located on the same VLAN, there is no device fragmenting this frame into the standard 1500 byte size for consumption by the host.

To allow the NetApp storage controller to support both standard and jumbo frame requests from hosts, one option is to place a router in between the FAS device and the hosts, as routers are able to fragment jumbo frames into smaller 1500 byte increments. Devices that can utilize jumbo frames can be placed onto a separate VLAN configured to directly pass jumbo frames to the NetApp storage controller, while hosts that can only accept standard frames are placed onto a VLAN whose traffic is passed through a router for fragmentation. This configuration allows all hosts to properly communicate with the NetApp storage controller.

Another method is to directly connect VLANs for standard frame traffic and jumbo frame traffic to separate ports on the NetApp storage controller. This has the advantage of allowing traffic with the DF bit (don’t fragment) set to always reach its destination. Here are some possible scenarios that make use of dedicated VLANs:

- Local management VLAN (MTU 1500): supports SNMP, Operations Manager, SSH, RLM, etc. This network never has storage traffic running across it.
- Storage traffic (MTU 9000): Isolated, non-routed VLAN for NFS, CIFS, or iSCSI data.
- Replication network (MTU 9000): Isolated, non-routed VLAN for high speed storage replication such as SnapMirror and SnapVault data. Separating this traffic allows more granular monitoring and the ability to support different WAN MTU sizes depending on the links used.

- Inter-site replication (MTU 1500 or lower): Useful for offsite backups where a WAN connection is required and can have different MTU maximums.

Configuration Template – Jumbo Frames

NOTE

The following config template illustrates the configuration of a single interface. When configuring the MTU on an interface that's part of a port channel, the port-channel interface itself must also contain the MTU statement on the switch.

NETAPP RC FILE

```
ifconfig e0a 10.1.1.100 netmask 255.255.255.0 mtusize 9000 partner
10.1.1.200 flowcontrol send
```

CISCO IOS SWITCH

```
interface GigabitEthernet1/1
 description NetApp e0a
 switchport access vlan 100
 switchport mode access
 flowcontrol receive on
 no cdp enable
 spanning-tree guard loop
 mtu 9198
interface Vlan 100
 ip address 10.1.1.1 255.255.255.0
 mtu 9198
```

JUMBO FRAME RECOMMENDATIONS

Significant performance increase can be realized through the use of Jumbo Frames in an Ethernet Storage network provided the following concepts are adhered to:

- Configure Jumbo Frames throughout the network, from Ethernet storage controller to the host
- Segment traffic with Jumbo frames on a different VLAN to ensure optimal network interface performance.

6 CONGESTION MANAGEMENT WITH FLOW CONTROL

Flow control mechanisms exist at many different OSI Layers including but not limited to NFS, TCP window, Ethernet XON/XOFF. In an Ethernet context, L2 flow control was unable to be implemented until the introduction of full duplex links, because a half duplex link is unable to send and receive traffic simultaneously. 802.3X allows a device on a point-to-point connection experiencing congestion to send a PAUSE frame to temporarily pause all flow of data. A reserved and defined multicast MAC address of 01-80-C2-00-00-01 is used to send the PAUSE frames, which also includes the length of pause requested.

In simple networks, this method may work well. However, with the introduction of larger and larger networks along with more advanced and faster network equipment and software, technologies such as TCP windowing, increased switch buffering, and end-to-end QoS better address the need for simple flow control throughout the network. Simple Ethernet Flow Control does not react granularly and fast enough to cope with these environments.

A TCP connection uses the end-to-end connection to determine the window size used, which can take into account the bandwidth, buffer space, and round trip time. As congestion or packet loss increases along that entire path, the window size will decrease to compensate essentially controlling the flow. Contrast this to PAUSE frames, which work on a point-to-point connection. The switch port or NIC decides when to send a PAUSE frame and for what duration while only taking into account this single link. No upper level protocols are considered. This can potentially affect TCP performance by introducing artificial delay between hops and causing TCP to decrease the window size due to dropped packets. In larger networks it is also possible that "congestion trees" start forming, severely limiting overall network capacity for all attached devices. **For these reasons, it's not recommended to enable flow control throughout the network (including switches, data ports, intracluster ports)**.

NOTE: When creating or configuring an interface on a NetApp Controller, the flow control settings default to "on" for both send and receive. Configuring "send off" and "receive off" will have to be explicitly configured from the **ifconfig** command.

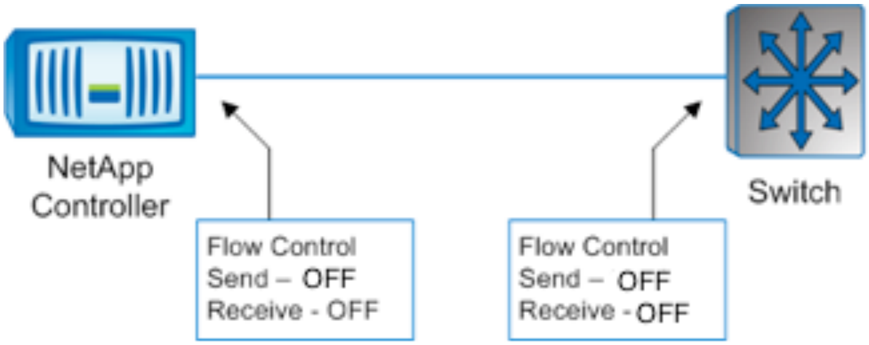
NetApp Controller
Switch
Flow Control Send – OFF Receive - OFF
Flow Control Send – OFF Receive - OFF

Figure 6-1) Flow Control

Configuration Template – Flow Control

NETAPP RC FILE

```
ifconfig e0a 10.1.1.100 netmask 255.255.255.0 mtusize 9000 partner
10.1.1.200 flowcontrol none
```

CISCO IOS SWITCH

```
interface GigabitEthernet1/1
 description NetApp e0a
 switchport access vlan 100
 switchport mode access
```

```
flowcontrol receive off
no cdp enable
spanning-tree guard loop
mtu 9198
```

6.1 FLOW CONTROL RECOMMENDATIONS

Ensure flow control is disabled on both the storage controller and the switch it is connected to.

7 CONCLUSION

With more and more mission critical data and applications relying on Ethernet Storage networks, it is critical that the network is completely understood from the Storage controller down to the host. Taking the network design and configuration for granted can lead to severe performance degradation and potential outages. Utilizing VLANs, fast start mechanisms in Spanning Tree, MutiMode LACP VIFs for link bundling, Jumbo Frames and Ethernet PAUSE (aka flow control) can provide the high availability and performance required with Ethernet Storage networks.

NetApp provides no representations or warranties regarding the accuracy, reliability, or serviceability of any information or recommendations provided in this publication, or with respect to any results that may be obtained by the use of the information or observance of any recommendations provided herein. The information in this document is distributed AS IS, and the use of this information or the implementation of any recommendations or techniques herein is a customer's responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. This document and the information contained herein may be used solely in connection with the NetApp products discussed in this document.

NetApp™

www.netapp.com