



[Home](#) [Archives](#) [Profile](#) [Subscribe](#)

« [Hard zoning versus soft zoning in a FC/FCoE SAN](#) | [Main](#) | [Partner Mac](#)
[Address: 00:00:00:00:00:00](#) »

01/16/2012

Introducing Target Driven Zoning (TDZ)

At 9:28 on Tuesday 12/6/2011, the T11 FC-GS-7 working group approved a motion to incorporate the text for “Peer Zoning” that was prepared by Claudio DeSanti (Cisco). The actual text can be found in the T11 document 11-411v2. To me this moment marked the culmination of a four year journey to take a concept, get it through a standards body and into a standards document.

Target Driven Zoning (TDZ) is a proposed application that utilizes Peer Zoning to **reduce the number of steps required to provision new storage by 50%**. Since many customers have been complaining about the task of zoning for years, I’m **proposing to achieve this reduction by eliminating the manual task of FC zoning**.

Just to be clear, while I view introduction of the Peer Zoning functionality as a game changer for FC, I don’t view the completion of this effort as some kind of extraordinary accomplishment, actually there’s nothing extraordinary about it. These sorts of efforts are started all the time in standards bodies. Sometimes they result in useful protocols that are implemented by many (e.g., FCoE and FIP). Other results are implemented by few but they provide a critical requirement for the industry (e.g., FC-SP) and yet others turn out to be interesting academic exercises but are never implemented by anyone (e.g., FC-SCM). I have hopes that Peer Zoning will fall into the category of “useful technology that is implemented (and deployed) by many” but it’s too early to tell as this point.

This post is intended to give you an overview of the technology and give you enough information to decide if TDZ is something that you’d like to use in your environment. I also feel it’s important to give credit where it’s due and with that in mind without the help of Mark Lippitt, David Black, Claudio DeSanti, Bob Nixon and Ralph Weber; this entire effort would still be hopelessly stalled.

Background

The story starts back in mid-2007. I was on a conference call with David Black and Mark Lippitt (both with EMC) discussing what would happen if zoning wasn’t



[Erik Smith](#)

[Follow](#)

ARCHIVES

[September 2022](#)

[August 2020](#)

[April 2019](#)

[December 2017](#)

[November 2017](#)

[July 2017](#)

[April 2017](#)

[September 2016](#)

[July 2016](#)

[February 2016](#)

SEARCH

[Search](#)

CATEGORIES

[Subscribe to this blog's feed](#)

DISCLAIMER

This is not an official EMC blog. The opinions expressed here are my personal opinions. Content published here is not read or approved in advance by EMC and does not necessarily reflect the views and opinions of EMC nor does it constitute any official communication of EMC.

used in an FC SAN (as was being proposed in the T11 FC-SCM working group). In case you don't know, EMC has a fairly strict best practice called "Single Initiator Zoning" that was created based on lab testing results. As the name suggests, the best practice states "zones should only contain a single initiator as well as the storage targets it needs to access." Because of the work that I had done with fabric scalability, I was pulled into the meeting (somewhat last minute if memory serves) and asked to provide my opinion on what was being proposed. Now, as most people who work with me will tell you, I typically call things as I see them and the intensity of my voice will vary with the level of conviction that I feel about a given topic. As a result, my reaction to eliminating zoning was something along the lines of "ARE YOU <bleep> KIDDING ME?" and I proceeded to provide specific reasons why I thought that this was the *SILLIEST* idea I had ever heard of. Perhaps it was the intensity of my reaction (or the simple fact that David simply didn't have sufficient bandwidth to create a presentation that contained all of my points), in either case, he eventually asked me to present my list of concerns to the T11 FC-SCM working group at the August 2007 meeting in Seattle.

Preparing for Seattle

In case you aren't familiar with T11, this is the group that literally wrote the book on FC. In fact, a better way to say it would be they ARE the book. From my point of view as an integration engineer who "grew up" with FC, the idea of merely attending (let alone presenting at) this meeting felt something akin to traveling to Mt Olympus and meeting Zeus and the gang. As a result, I spent much of the month before the meeting preparing by testing specific "unzoned" configurations in the lab, re-reading the various standards involved (FC-GS and FC-SW), iterating on the presentation with Mark Lippitt and working myself up into an emotional wreck in general. As luck would have it, the FC-SCM meeting was first on the schedule for the week, so off I went to present at my first T11 meeting without having actually ever attended one.

The FC-SCM meeting

During the FC-SCM meeting I presented "FC-SCM-OpenZoning". The main point of the presentation was **due to the way Initiators and Targets perform discovery, you can't simply eliminate zoning**. If you did, the resulting flood of name server queries every time something changed in the fabric would bring the fabric to its knees with as few as 255 N_Ports in the fabric. Consider the case where an N_Port is bouncing (logging in and out) and you are totally screwed. After speaking to my last slide and answering a couple of questions, I returned to my seat feeling like the weight of the world had been lifted off of my shoulders. I think Mark Lippitt and David Black said something to the effect of "Nice job" and my response was going to be "Boy, am I glad that's over with" but before I could even get the words out of my mouth, Claudio turned around and said something to the effect of "Ok, so you have shown us a problem, now what? Would you be willing to come back and present a solution?" I literally couldn't believe my ears, I was like "are you talking to me?" After briefly consulting with Mark, I said yes and committed to provide a proposal at the October 2007 T11 meeting in Coeur d'Alene, ID. The purpose of that proposal would be to describe an approach that would eliminate the manual task of zoning.

Inspiration strikes

After the FC-SCM meeting Mark, David and I went out to lunch. On the walk down to Pike's Place Market I was walking behind them while they were talking

about the concept of OLZ or “One Large Zone” (what would eventually become FC-SCM). I found OLZ interesting and would dedicate many hours over the next three years pushing it for it to be adopted, but I wanted to pursue a different approach. I believed (and still do) that the key to a solution was to solve the RSCN problem. Basically, the RSCN problem happens in an environment without zoning when a single device logs in or out. When the device’s status changes an RSCN is sent to all devices that are impacted by the change. Since all devices register for RSCN in a fabric without zoning, every device would receive an RSCN and would pile on the Name Server at the same time and basically cause a DoS attack. With this concern in mind my thinking process went something along the lines of:

What if somehow we could send RSCNs only to end devices that are actually impacted by the change (i.e., hosts that are logged into that target and vice versa)?

- Where would this information come from?

Maybe a storage port could tell the switch to only send RSCNs to end devices that are logged into it?

- You would probably identify these devices to the switch by their WWPN

But wait a second, our storage ports have a list of WWPNs that they care about (have been granted access) in the LUN masking database

- If a storage ports could somehow send these WWPNs to the switch...
- A switch could use this information to restrict RSCN distribution

But...switches already use the WWPNs in the definition of a zone

Why couldn’t the storage port just share the WWPNs in the LUN Masking database with the switch and then the switch could just create the zones based on this information???

The rest of the story

Describing all the ups and downs of the story goes way beyond the scope of this post, but a couple of excerpts from a cliff notes version would certainly mention; Claudio agrees to write the text at the August 2011 meeting in Edmonton, delivers the goods in Albuquerque and it all gets approved in St. Petersburg. Thanks again Claudio!

Target Driven Zoning technical details

In this section I describe how I envision the Peer Zoning functionality Claudio defined in 11-411v2 will be utilized by an application that I am calling Target Driven Zoning or TDZ.

The Target Driven Zoning concept is simple, since a storage port has all of the information required to create an FC zone, it should just go ahead and create one automatically.

Before I dive into the protocol level details that will explain how TDZ works, let me take step back and define the Storage Provisioning process as it is done today without TDZ and then show how it would work with TDZ.

Without TDZ

1. Physically attach hosts and storage to the fabric.

2. Using the switch zoning interface, create a zone that contains an initiator and its targets. Repeat for every initiator/target relationship being added to the environment.
3. Activate the new Zone Set / Configuration.
4. Using the Storage array management interface, add each initiator to a "Storage Group" on the array.

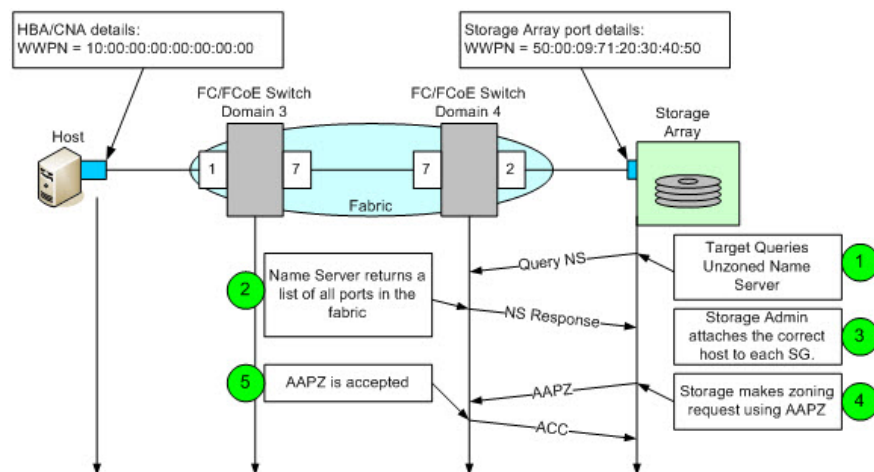
With TDZ

1. Physically attach hosts and storage to the fabric.
2. Using the Storage array management interface, add each initiator to a "Storage Group" on the array. **The storage will then automatically setup the appropriate zones to allow each initiator to access it.**

The nuts and bolts of Peer Zoning / TDZ

For the remainder of this section, I'll be referring to the following configuration. Starting from the left, the configuration consists of:

1. A Host containing a single HBA/CNA port that has a WWPN (World Wide Port Name) of 10:00:00:00:00:00:00.
2. A FC Fabric containing two FC switches:
 - a. FC/FCoE Switch Domain 3;
 - b. FC/FCoE Switch Domain 4
3. A Storage array containing a port that has a WWPN of 50:00:09:71:20:30:40:50



Step 1 - The Target queries the Unzoned Name Server

Due to an assumption on our part of "when creating a Storage Group, users would rather pick WWPNs from a list rather than manually type them in"; TDZ suffers from a causality dilemma that is solved by utilizing the unzoned name server.

The dilemma is:

How do you pick a host WWPN from a list in the array's management software when none of the interfaces on that array have been zoned to have access to that host WWPN?

As I describe in the [Hard zoning versus Soft zoning blog post](#), the response from the switch to a Name Server query will usually only contain those devices that share a common zone with the querying N_Port. When TDZ is being used, there

is a very good chance that the storage has not been zoned to have access to anything. As a result, it will not be possible to provide a list of initiator WWPNs to the storage admin via a traditional NS query.

The unzoned Name Server provides a solution to this dilemma because as the name implies, the unzoned name server allows an N_Port to query the Name Server and obtain a list of all N_Ports registered with the fabric without regard for zoning.

Step 2 - The Name Server returns a list of all ports registered in the Fabric

The response to the unzoned Name Server query will be a list of all N_Ports registered with the fabric.

As you think about this behavior, you may come to the conclusion that it represents a serious security flaw. As Peer Zoning was being defined, we felt the question we needed to answer was:

“If any N_Port can just query the unzoned Name Server, what’s to stop a host from using the unzoned name server to discover what targets are out there and then grant itself access to any storage port that it wants?”

We thought about this question (a lot actually) and we came up with a multi-pronged approach. As I’ll describe shortly each approach represents a different level of security that can be set on a fabric wide basis. However, before I get to the descriptions, let me state that no matter which one you choose, you are still protected by LUN Masking on the Storage array. In other words, if a rogue host were to grant itself access to the target using Peer Zoning commands, the rogue host would still be prevented from accessing data on that target due to LUN Masking. LUN Masking will only allow certain WWPNs to access certain LUNs on each target. So even if a host created a zone to allow itself to have access to a target port, it would also need to spoof the WWPN of a host that has been granted access to LUNs on that target. Let me point out that this is something that can easily be done today on all HBA and CNAs and does not represent a new security hole that is being introduced by peer zoning.

- Option 1: Peer Zoning disabled (default). If you don’t find manual zoning tasks to be too much trouble, then you can opt not to enable peer zoning.
- Option 2: Peer Zoning enabled - Authentication required. This option allows you to enable Peer Zoning but will require any port that wants to use the unzoned Name Server to authenticate with the switch using one of the mechanisms defined in FC-SP (e.g., DH-CHAP)
- Option 3: Peer zoning enabled - Port based. This option allows you to enable Peer Zoning but it will only allow unzoned name server queries on certain switch interfaces. You could simply enable the feature on interfaces where storage ports are located.
- Option 4: Peer zoning enabled - OUI based. This option allows you to specify that certain vendor OUI’s can use unzoned Name Server queries by default. (e.g., I trust <your storage vendor of choice> (hopefully EMC) OUIs but not others)
- Option 5: Peer zoning enable - open. This option allows you to specify that Peer Zoning can be used by any N_Port in the fabric.

Step 3 - The Storage Admin attaches the correct host WWPN to a "Storage Group"

This is more of a Storage Provisioning process step and does not require any FC protocol interaction. From within the Storage Array software application, the storage admin selects a host and associates it with a "Storage Group". When the user clicks "OK" or something similar, the next step will be performed.

Step 4 - The Storage port uses AAPZ

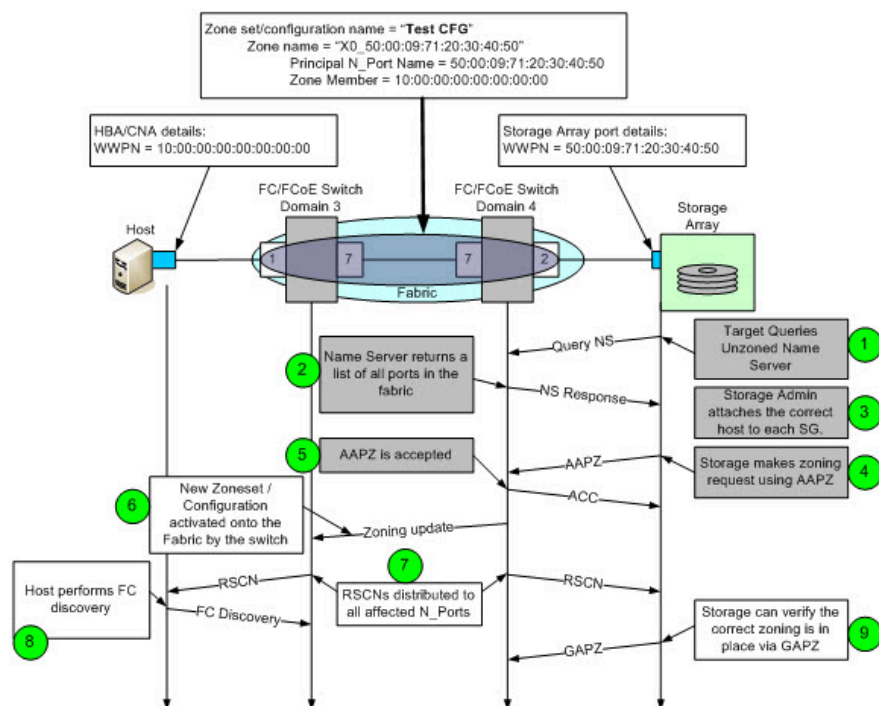
Once the Storage Administrator adds the WWPN of the host to the storage group and commits this change, the storage port(s) that are associated with this storage group will use AAPZ (Add/replace Active Peer Zone) to add the appropriate peer zones to the fabric. The AAPZ request will contain the zone name, the principal N_Port name as well as the zone members (the peers) that should be allowed to access the principal N_Port name.

Step 5 - The AAPZ is accepted by the switch

Assuming the switch is functioning properly and it is not too busy to process the AAPZ, it will accept the AAPZ.

Step 6 - New zoneset / configuration is activated on the fabric by the switch

Once the switch transmits the ACC to the AAPZ, it has one minute to actually update the zoning on the fabric. An example of a peer zone is shown at the top of the following diagram.



Step 7 - RSCNs distributed to all affected N_Ports

Once the zone set containing the new Peer Zone has been activated onto the fabric, each switch will transmit RSCNs to each affected N_Port. By the way, there is nothing new here. RSCNs are an existing part of the FC protocol and are widely used today.

Step 8 - The Host performs FC discovery

Once the hosts in the peer zone have received an RSCN, they will perform FC discovery as normal. This is the same discovery process that I described in the [Hard versus soft zoning blog post](#) with one big exception; the peer members in a peer zone are only allowed to access the principal zone member and not each other. This is a new behavior specific to peer zones and it is was put in place to ensure that our single initiator zoning best practice can remain intact.

Step 9 - Storage verifies zoning via GAPZ

At any point after using AAPZ, a target port can use GAPZ to verify that the zoning change is in place.

Conclusion

If you're interested in hearing more about TDZ or are an EMC Customer and interested in actually giving it a try, please let me know! A robust amount of Customer demand is all that is required to get this idea out of the prototype phase and into the ready to deploy phase.

Thanks for reading!

Posted at 12:03 PM | [Permalink](#)

 Reblog (0)

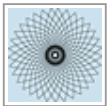
Comments

 You can follow this conversation by subscribing to the [comment feed](#) for this post.



Congratulations Erik - I know you put a time of energy and passion into this effort. As you said, hopefully customers will help move this from a good idea to real deployments.

Posted by: [Stuart Miniman](#) | [01/16/2012 at 12:33 PM](#)



Thanks Stu!

Posted by: [Erik Smith](#) | [01/16/2012 at 12:58 PM](#)



Congrats Erik (and others). Sounds very useful to me.

Posted by: [Chris Carter](#) | [01/16/2012 at 01:33 PM](#)



Thanks Chris!

Posted by: [Erik Smith](#) | [01/16/2012 at 02:09 PM](#)



Excellent and interesting work!

Posted by: [Victorforde](#) | [01/16/2012 at 05:06 PM](#)



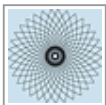
Hi Erik, quick question about this.

Will the storage port be able to remove peer zones when a WWPN is removed from the masking configuration?

We still have upper limits on the number of zones in the fabric, and since this doesn't actually remove the requirement for the zone, we may still need to remove unused zones. Especially in the case of a VMax, where every host WWPN is added to the masking configuration of every storage port in the masking view, this would actually increase the number of zones in each fabric.

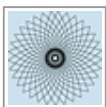
Great work by the way, this will be of great benefit once it becomes widely available.

Posted by: Anthony Padula | [01/16/2012 at 07:13 PM](#)



Thanks Victor!

Posted by: [Erik Smith](#) | [01/16/2012 at 07:20 PM](#)



Hi Anthony, great question! Yes, the peer zone will be updated each time the masking configuration changes. Each change could represent the addition or removal of a host WWPN.

Posted by: [Erik Smith](#) | [01/16/2012 at 07:23 PM](#)



Erik,

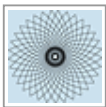
How often would the array query the NS if a host is masked but is not yet powered on? If the NS doesn't have an entry would the zone still be created or would the array wait until it received an NS response that included the masked WWN? I like the overall idea but I see many organizations looking for the network team to take over SAN switch operations and having a storage admin who may or may not keep a consistent clean environment decide who can talk to what storage could be a scary proposition. I would prefer to see a little more intelligence related to removal of unused or inactive zones.

Posted by: Scott DeShong | [01/17/2012 at 10:18 PM](#)



Great work! Congratulations Erik.

Posted by: Willington Echeverri | [01/18/2012 at 07:29 AM](#)



Hi Scott, thanks for the feedback! The array port pushes peer members out to the switch (using AAPZ) when either:

- 1) The LUN Mask on the storage port changes; or
- 2) The array port detects that that current peer zone does not match the current masking database (using GAPZ)

In the situation you are describing, there are at least four different scenarios that need to be considered:

- a) The host port has never logged into the array port and its WWPN is not associated with a storage group.
- b) The host port has never logged into the array port and its WWPN was manually added to the storage group.
- c) The host port was previously logged into the array port and its WWPN is not associated with a storage group.

d) The host port was previously logged into the array port and its WWPN is associated with a storage group.

Scenarios b and d would result in AAPZ being used as soon as the LUN Masking database was updated. See update case 1 above.

Scenarios a and c would result in no peer zone changes being made by the array port.

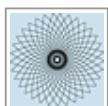
In regards to house keeping tasks:

When a host adapter is replaced, it's WWPN will change and this change will need to be reflected in the LUN Masking database, which will in turn be reflected in the peer zone members. In other words the house keeping is done automatically for them.

The peer zone will also be automatically updated if a storage group is destroyed due to a host being decommissioned.

Does the approach described above fully address your concerns?

Posted by: [Erik Smith](#) | [01/18/2012 at 08:44 AM](#)



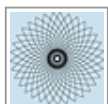
Thanks Willington!

Posted by: [Erik Smith](#) | [01/18/2012 at 08:46 AM](#)



Nice work Erik! This will resolve a lot of issues in storage networking.

Posted by: mike warner | [01/18/2012 at 09:31 AM](#)



Hi Mike, thanks!

Posted by: [Erik Smith](#) | [01/18/2012 at 11:29 AM](#)



Erik,

Thanks for the additional information. So if I'm understanding correctly, the array will always implement zones for any defined storage group (configured LUN masking) regardless of host state. The zones will only change when the array masking changes.

How would multiple arrays within the same fabric handle zoning? How would they agree on a common zoneset name? Would the GAPZ only check for zones in the active zoneset that match their respective LUN masking and ignore everything else? Would fabric locking be required to ensure only one array can update the zoneset at any one time?

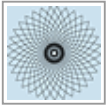
Thanks again for the great discussion!

Posted by: Scott DeShong | [01/18/2012 at 12:00 PM](#)



Great Post Erik, Let's hope that it becomes an approved AND accepted standard. This way anyone that finds current zoning in a SAN difficult could give it a try instead of trying to run OLTP Apps on NFS!

Posted by: [Chris Conniff](#) | [01/18/2012 at 01:01 PM](#)



Hi Scott, thanks for the great questions! These are points that I should have included in the original post.

You're right; the peer zone members will always match the LUN mask.

In regards to your other points:

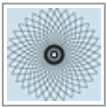
1. Multiple arrays: Multiple arrays are allowed and this situation isn't actually any different from the case where you have multiple ports from the same array attached to the same fabric because as of right now every array port will transmit its own AAPZ.

2. Zone set names: By design the target port doesn't have any knowledge of the zone set name. The AAPZ simply requests that the fabric add a peer zone to the active zone set.

3. Zone Names: The target can specify any zone name desired, but this could lead to conflicts. As a result, I'm pushing to have EMC Array ports just use the default zone name of 'X0_WWPN of principal device'. Since the WWPN of principal device is the same as the WWPN of the target port using the AAPZ and the WWPN is unique, we shouldn't run into any zone name conflicts.

4. Fabric Locking: The fabric has to be locked every time the active zone set is modified. This is an integral part of the three phase commit used today and was not something that could be changed. As a result, AAPZ commands are accepted by the switch and then added to the active zone set at a later time. The maximum amount of time between the AAPZ accept and when the change will actually take effect is one minute. This one minute time period will allow the switches to coalesce AAPZ requests and it avoids zone set activation storms. In practice, I am expecting the time between AAPZ and activation to be much less than one minute.

Posted by: [Erik Smith](#) | [01/18/2012 at 02:29 PM](#)



Hi Chris, thanks! Agreed, accepted and requested by our customers is the next hurdle..

Posted by: [Erik Smith](#) | [01/18/2012 at 02:35 PM](#)



Congrats :)....

Looking forward to start working on this ..

Posted by: [Sandeep MP](#) | [01/22/2012 at 09:56 PM](#)



Its perfect and too interesting..

Posted by: [Kiran Kumar](#) | [01/31/2012 at 09:37 AM](#)



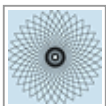
This is superb Stuff :) .Article was very informative.I have two questions

1.How this is going to work with auto provisioning in VMAX models ?

2. Since zoning is done at storage port level, does current storage processors support this technology, I am pretty sure this will create more problems if front end port is dead, either we can't access storage nor we can't do zoning?

Regards
Sreejith.C

Posted by: [sreejith](#) | [02/08/2012 at 01:02 PM](#)



Hi Sreejith, thanks!

In regards to question 1, the actual implementation details are still being worked out. As soon as I have a better feeling for how this will be integrated into the UI, I will be doing another post.

For question 2, there is nothing to prevent you from making the same storage device visible on multiple interfaces and then using something like PowerPath for load balancing and failover. Also, traditional fabric based zoning will still function as it does today, so you can always fallback to that if necessary.

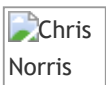
Regards, Erik

Posted by: [Erik Smith](#) | [02/10/2012 at 08:53 PM](#)



Thanks Eric,
I'm pretty sure this will be a turn around. I wish you all the best :)

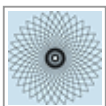
Posted by: [sreejith](#) | [02/13/2012 at 06:01 AM](#)



Great article Erik, I would very much like to see this technology adopted!

Has there been any thought given to how this could play into multi switch fabrics? Would ISL's still be configured by hand or does this contain provisions for switches to identify and proliferate zones to peer switches?

Posted by: [Chris Norris](#) | [02/15/2012 at 02:36 PM](#)



Hi Chris, thanks!

Since a Peer Zone is literally a zone tagged with a special attribute, it will be automatically distributed throughout the fabric as a part of the zone set / configuration just like a traditional zone would be.

Physical ISLs would still need to be present (and configured) but if you were using VSANs, one could easily imagine a way to automatically configure IVR to allow the connectivity specified by the target. This is actually a very interesting idea...

Posted by: [Erik Smith](#) | [02/16/2012 at 09:45 AM](#)

[Next »](#)

The comments to this entry are closed.

[Brass Tacks](#) [Powered by TypePad](#)



[Home](#) [Archives](#) [Profile](#) [Subscribe](#)

« [Hard zoning versus soft zoning in a FC/FCoE SAN](#) | [Main](#) | [Partner Mac](#)
[Address: 00:00:00:00:00:00](#) »

01/16/2012

Introducing Target Driven Zoning (TDZ)

At 9:28 on Tuesday 12/6/2011, the T11 FC-GS-7 working group approved a motion to incorporate the text for “Peer Zoning” that was prepared by Claudio DeSanti (Cisco). The actual text can be found in the T11 document 11-411v2. To me this moment marked the culmination of a four year journey to take a concept, get it through a standards body and into a standards document.

Target Driven Zoning (TDZ) is a proposed application that utilizes Peer Zoning to **reduce the number of steps required to provision new storage by 50%**. Since many customers have been complaining about the task of zoning for years, I’m **proposing to achieve this reduction by eliminating the manual task of FC zoning**.

Just to be clear, while I view introduction of the Peer Zoning functionality as a game changer for FC, I don’t view the completion of this effort as some kind of extraordinary accomplishment, actually there’s nothing extraordinary about it. These sorts of efforts are started all the time in standards bodies. Sometimes they result in useful protocols that are implemented by many (e.g., FCoE and FIP). Other results are implemented by few but they provide a critical requirement for the industry (e.g., FC-SP) and yet others turn out to be interesting academic exercises but are never implemented by anyone (e.g., FC-SCM). I have hopes that Peer Zoning will fall into the category of “useful technology that is implemented (and deployed) by many” but it’s too early to tell as this point.

This post is intended to give you an overview of the technology and give you enough information to decide if TDZ is something that you’d like to use in your environment. I also feel it’s important to give credit where it’s due and with that in mind without the help of Mark Lippitt, David Black, Claudio DeSanti, Bob Nixon and Ralph Weber; this entire effort would still be hopelessly stalled.

Background

The story starts back in mid-2007. I was on a conference call with David Black and Mark Lippitt (both with EMC) discussing what would happen if zoning wasn’t



[Erik Smith](#)

ARCHIVES

[August 2020](#)

[April 2019](#)

[December 2017](#)

[November 2017](#)

[July 2017](#)

[April 2017](#)

[September 2016](#)

[July 2016](#)

[February 2016](#)

[May 2015](#)

SEARCH

CATEGORIES

[Subscribe to this blog's feed](#)

DISCLAIMER

This is not an official EMC blog. The opinions expressed here are my personal opinions. Content published here is not read or approved in advance by EMC and does not necessarily reflect the views and opinions of EMC nor does it constitute any official communication of EMC.

used in an FC SAN (as was being proposed in the T11 FC-SCM working group). In case you don't know, EMC has a fairly strict best practice called "Single Initiator Zoning" that was created based on lab testing results. As the name suggests, the best practice states "zones should only contain a single initiator as well as the storage targets it needs to access." Because of the work that I had done with fabric scalability, I was pulled into the meeting (somewhat last minute if memory serves) and asked to provide my opinion on what was being proposed. Now, as most people who work with me will tell you, I typically call things as I see them and the intensity of my voice will vary with the level of conviction that I feel about a given topic. As a result, my reaction to eliminating zoning was something along the lines of "ARE YOU <bleep> KIDDING ME?" and I proceeded to provide specific reasons why I thought that this was the *SILLIEST* idea I had ever heard of. Perhaps it was the intensity of my reaction (or the simple fact that David simply didn't have sufficient bandwidth to create a presentation that contained all of my points), in either case, he eventually asked me to present my list of concerns to the T11 FC-SCM working group at the August 2007 meeting in Seattle.

Preparing for Seattle

In case you aren't familiar with T11, this is the group that literally wrote the book on FC. In fact, a better way to say it would be they ARE the book. From my point of view as an integration engineer who "grew up" with FC, the idea of merely attending (let alone presenting at) this meeting felt something akin to traveling to Mt Olympus and meeting Zeus and the gang. As a result, I spent much of the month before the meeting preparing by testing specific "unzoned" configurations in the lab, re-reading the various standards involved (FC-GS and FC-SW), iterating on the presentation with Mark Lippitt and working myself up into an emotional wreck in general. As luck would have it, the FC-SCM meeting was first on the schedule for the week, so off I went to present at my first T11 meeting without having actually ever attended one.

The FC-SCM meeting

During the FC-SCM meeting I presented "FC-SCM-OpenZoning". The main point of the presentation was **due to the way Initiators and Targets perform discovery, you can't simply eliminate zoning**. If you did, the resulting flood of name server queries every time something changed in the fabric would bring the fabric to its knees with as few as 255 N_Ports in the fabric. Consider the case where an N_Port is bouncing (logging in and out) and you are totally screwed. After speaking to my last slide and answering a couple of questions, I returned to my seat feeling like the weight of the world had been lifted off of my shoulders. I think Mark Lippitt and David Black said something to the effect of "Nice job" and my response was going to be "Boy, am I glad that's over with" but before I could even get the words out of my mouth, Claudio turned around and said something to the effect of "Ok, so you have shown us a problem, now what? Would you be willing to come back and present a solution?" I literally couldn't believe my ears, I was like "are you talking to me?" After briefly consulting with Mark, I said yes and committed to provide a proposal at the October 2007 T11 meeting in Coeur d'Alene, ID. The purpose of that proposal would be to describe an approach that would eliminate the manual task of zoning.

Inspiration strikes

After the FC-SCM meeting Mark, David and I went out to lunch. On the walk down to Pike's Place Market I was walking behind them while they were talking

about the concept of OLZ or “One Large Zone” (what would eventually become FC-SCM). I found OLZ interesting and would dedicate many hours over the next three years pushing it for it to be adopted, but I wanted to pursue a different approach. I believed (and still do) that the key to a solution was to solve the RSCN problem. Basically, the RSCN problem happens in an environment without zoning when a single device logs in or out. When the device’s status changes an RSCN is sent to all devices that are impacted by the change. Since all devices register for RSCN in a fabric without zoning, every device would receive an RSCN and would pile on the Name Server at the same time and basically cause a DoS attack. With this concern in mind my thinking process went something along the lines of:

What if somehow we could send RSCNs only to end devices that are actually impacted by the change (i.e., hosts that are logged into that target and vice versa)?

- Where would this information come from?

Maybe a storage port could tell the switch to only send RSCNs to end devices that are logged into it?

- You would probably identify these devices to the switch by their WWPN

But wait a second, our storage ports have a list of WWPNs that they care about (have been granted access) in the LUN masking database

- If a storage ports could somehow send these WWPNs to the switch...
- A switch could use this information to restrict RSCN distribution

But...switches already use the WWPNs in the definition of a zone

Why couldn’t the storage port just share the WWPNs in the LUN Masking database with the switch and then the switch could just create the zones based on this information???

The rest of the story

Describing all the ups and downs of the story goes way beyond the scope of this post, but a couple of excerpts from a cliff notes version would certainly mention; Claudio agrees to write the text at the August 2011 meeting in Edmonton, delivers the goods in Albuquerque and it all gets approved in St. Petersburg. Thanks again Claudio!

Target Driven Zoning technical details

In this section I describe how I envision the Peer Zoning functionality Claudio defined in 11-411v2 will be utilized by an application that I am calling Target Driven Zoning or TDZ.

The Target Driven Zoning concept is simple, since a storage port has all of the information required to create an FC zone, it should just go ahead and create one automatically.

Before I dive into the protocol level details that will explain how TDZ works, let me take step back and define the Storage Provisioning process as it is done today without TDZ and then show how it would work with TDZ.

Without TDZ

1. Physically attach hosts and storage to the fabric.

2. Using the switch zoning interface, create a zone that contains an initiator and its targets. Repeat for every initiator/target relationship being added to the environment.
3. Activate the new Zone Set / Configuration.
4. Using the Storage array management interface, add each initiator to a "Storage Group" on the array.

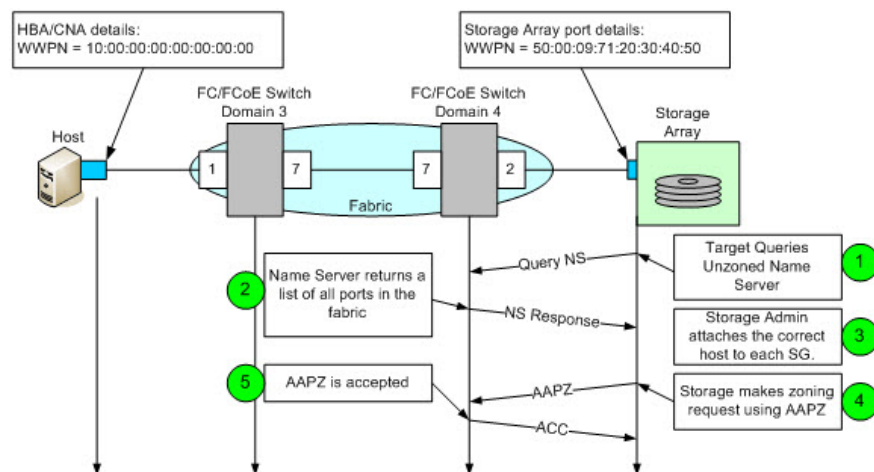
With TDZ

1. Physically attach hosts and storage to the fabric.
2. Using the Storage array management interface, add each initiator to a "Storage Group" on the array. **The storage will then automatically setup the appropriate zones to allow each initiator to access it.**

The nuts and bolts of Peer Zoning / TDZ

For the remainder of this section, I'll be referring to the following configuration. Starting from the left, the configuration consists of:

1. A Host containing a single HBA/CNA port that has a WWPN (World Wide Port Name) of 10:00:00:00:00:00:00:00.
2. A FC Fabric containing two FC switches:
 - a. FC/FCoE Switch Domain 3;
 - b. FC/FCoE Switch Domain 4
3. A Storage array containing a port that has a WWPN of 50:00:09:71:20:30:40:50



Step 1 - The Target queries the Unzoned Name Server

Due to an assumption on our part of "when creating a Storage Group, users would rather pick WWPNs from a list rather than manually type them in"; TDZ suffers from a causality dilemma that is solved by utilizing the unzoned name server.

The dilemma is:

How do you pick a host WWPN from a list in the array's management software when none of the interfaces on that array have been zoned to have access to that host WWPN?

As I describe in the [Hard zoning versus Soft zoning blog post](#), the response from the switch to a Name Server query will usually only contain those devices that share a common zone with the querying N_Port. When TDZ is being used, there

is a very good chance that the storage has not been zoned to have access to anything. As a result, it will not be possible to provide a list of initiator WWPNs to the storage admin via a traditional NS query.

The unzoned Name Server provides a solution to this dilemma because as the name implies, the unzoned name server allows an N_Port to query the Name Server and obtain a list of all N_Ports registered with the fabric without regard for zoning.

Step 2 - The Name Server returns a list of all ports registered in the Fabric

The response to the unzoned Name Server query will be a list of all N_Ports registered with the fabric.

As you think about this behavior, you may come to the conclusion that it represents a serious security flaw. As Peer Zoning was being defined, we felt the question we needed to answer was:

“If any N_Port can just query the unzoned Name Server, what’s to stop a host from using the unzoned name server to discover what targets are out there and then grant itself access to any storage port that it wants?”

We thought about this question (a lot actually) and we came up with a multi-pronged approach. As I’ll describe shortly each approach represents a different level of security that can be set on a fabric wide basis. However, before I get to the descriptions, let me state that no matter which one you choose, you are still protected by LUN Masking on the Storage array. In other words, if a rogue host were to grant itself access to the target using Peer Zoning commands, the rogue host would still be prevented from accessing data on that target due to LUN Masking. LUN Masking will only allow certain WWPNs to access certain LUNs on each target. So even if a host created a zone to allow itself to have access to a target port, it would also need to spoof the WWPN of a host that has been granted access to LUNs on that target. Let me point out that this is something that can easily be done today on all HBA and CNAs and does not represent a new security hole that is being introduced by peer zoning.

- Option 1: Peer Zoning disabled (default). If you don’t find manual zoning tasks to be too much trouble, then you can opt not to enable peer zoning.
- Option 2: Peer Zoning enabled - Authentication required. This option allows you to enable Peer Zoning but will require any port that wants to use the unzoned Name Server to authenticate with the switch using one of the mechanisms defined in FC-SP (e.g., DH-CHAP)
- Option 3: Peer zoning enabled - Port based. This option allows you to enable Peer Zoning but it will only allow unzoned name server queries on certain switch interfaces. You could simply enable the feature on interfaces where storage ports are located.
- Option 4: Peer zoning enabled - OUI based. This option allows you to specify that certain vendor OUI’s can use unzoned Name Server queries by default. (e.g., I trust <your storage vendor of choice> (hopefully EMC) OUIs but not others)
- Option 5: Peer zoning enable - open. This option allows you to specify that Peer Zoning can be used by any N_Port in the fabric.

Step 3 - The Storage Admin attaches the correct host WWPN to a "Storage Group"

This is more of a Storage Provisioning process step and does not require any FC protocol interaction. From within the Storage Array software application, the storage admin selects a host and associates it with a "Storage Group". When the user clicks "OK" or something similar, the next step will be performed.

Step 4 - The Storage port uses AAPZ

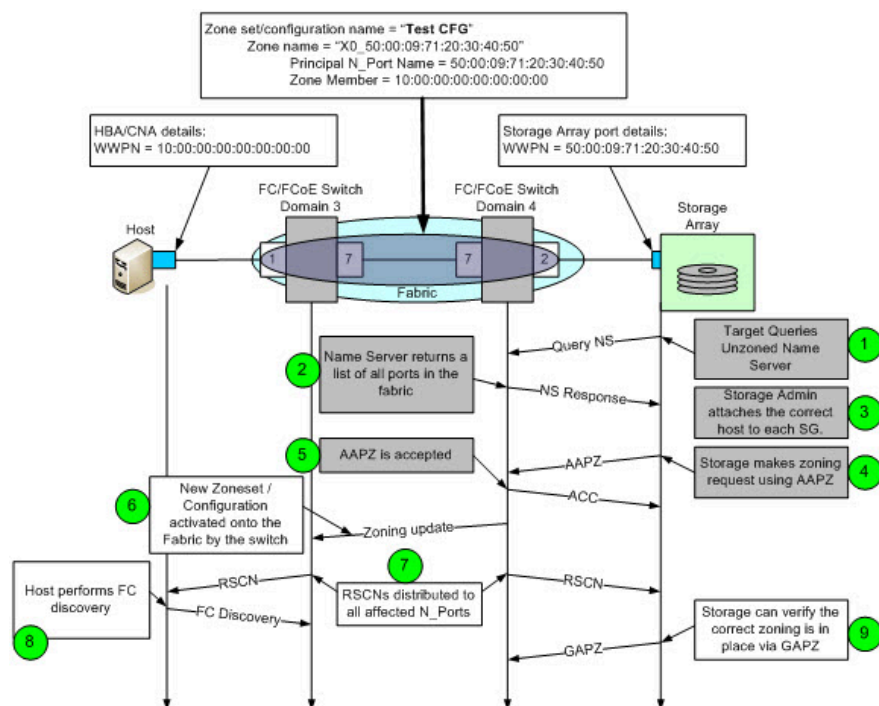
Once the Storage Administrator adds the WWPN of the host to the storage group and commits this change, the storage port(s) that are associated with this storage group will use AAPZ (Add/replace Active Peer Zone) to add the appropriate peer zones to the fabric. The AAPZ request will contain the zone name, the principal N_Port name as well as the zone members (the peers) that should be allowed to access the principal N_Port name.

Step 5 - The AAPZ is accepted by the switch

Assuming the switch is functioning properly and it is not too busy to process the AAPZ, it will accept the AAPZ.

Step 6 - New zoneset / configuration is activated on the fabric by the switch

Once the switch transmits the ACC to the AAPZ, it has one minute to actually update the zoning on the fabric. An example of a peer zone is shown at the top of the following diagram.



Step 7 - RSCNs distributed to all affected N_Ports

Once the zone set containing the new Peer Zone has been activated onto the fabric, each switch will transmit RSCNs to each affected N_Port. By the way, there is nothing new here. RSCNs are an existing part of the FC protocol and are widely used today.

Step 8 - The Host performs FC discovery

Once the hosts in the peer zone have received an RSCN, they will perform FC discovery as normal. This is the same discovery process that I described in the [Hard versus soft zoning blog post](#) with one big exception; the peer members in a peer zone are only allowed to access the principal zone member and not each other. This is a new behavior specific to peer zones and it is was put in place to ensure that our single initiator zoning best practice can remain intact.

Step 9 - Storage verifies zoning via GAPZ

At any point after using AAPZ, a target port can use GAPZ to verify that the zoning change is in place.

Conclusion

If you're interested in hearing more about TDZ or are an EMC Customer and interested in actually giving it a try, please let me know! A robust amount of Customer demand is all that is required to get this idea out of the prototype phase and into the ready to deploy phase.

Thanks for reading!

Posted at 12:03 PM | [Permalink](#)

 Reblog (0)

Comments

 You can follow this conversation by subscribing to the [comment feed](#) for this post.



Following up on sreejith's comments...

If you define masking in an autoprovisioning group (not exclusive to VMAX, other vendor's have initiator groups as well), would zoning be identical between fabrics?

For example:

- an IG of 2 initiators (1 per fabric)
- a PG of 2 ports (1 per fabric)
- a SG of x devs
- MV of the above

Would there be 4 zones in each fabric? Or would TDZ only create zones in it's fabric for the WWNs that it can locate? I believe it's the latter, but want to double check.

I may have missed something, I need to read the article again, lots of great info here.

Great article to be clear.

Posted by: Dave Silvestri | [02/16/2012 at 10:39 AM](#)



Hi Dave, thanks! Peer zoning would allow either of the behaviors you are describing. As a result, I view this decision as an implementation detail. I can see arguments for and against both approaches. Do you have a strong preference towards one or the other or will either work?

Posted by: [Erik Smith](#) | [02/16/2012 at 02:25 PM](#)



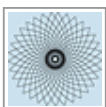
Personally I prefer to keep my zoning specific to the entities that exist in a fabric. I find it helps if someone does upgrades on a server and pulls the fibre cables, and then replaces them when complete, but gets them reversed.

Mostly that's for names/nicknames (which I also wonder if you could pull from the masking consistency is wonderful).

I suppose a benefit is that it wouldn't matter if the cables got reversed... however I prefer symmetry, and all my "a" labeled hbas on fabric A, and all my "b" labeled hbas on Fabric B...

To each his own though :)

Posted by: [Dave Silvestri](#) | [02/17/2012 at 11:59 AM](#)

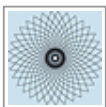


:-) Very nice to see it finally got accepted and written up by T11. Coincidence is that I had a similar proposal back in 2002 when doing consulting work for HP and had a long discussion with one of the HP engineer in Colorado Springs.

I asked once or twice how it went but when push came to shove other projects were given priorities and the idea lead to lala land. In this case your ears. :-)

Anyway, great job Eric and I hope to see this coming in array/switch implementations in the very near future. Will save many people a lot of headaches. :-)

Posted by: [Erwin van Londen](#) | [03/15/2012 at 01:40 AM](#)



Hi Erwin, thanks! For our customers sake, I wish you had succeeded earlier! I think it would have made FC much easier for them to use...

Regards, Erik

Posted by: [Erik Smith](#) | [03/15/2012 at 08:42 AM](#)



Thank you Erik :) Appreciate your reply

Posted by: [Sreejith.c](#) | [03/23/2012 at 09:20 AM](#)



Any progress on this really interesting topic?

Posted by: [MGK](#) | [01/09/2013 at 08:17 AM](#)



Hi Michal, yes we're making progress on TDZ. I hope to have something significant to share in the next few days.

Erik

Posted by: [Erik](#) | [01/17/2013 at 08:17 AM](#)



Hello Erick,

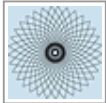
its a brilliant idea and significant savings from a operational perspective.

may i know how its making progress in terms of implementation by storage vendors (EMC, Cisco, Hitachi)?

Thanks,

Mallik.

Posted by: Mallik | [07/07/2014 at 01:44 AM](#)



Hi Mallik, thanks! I was pleasantly surprised with all of the attention TDZ recieved at EMC World this year. Although interest from our Customers (and competition) is way up compared to a year ago, I cannot share anything specific about our product plans at this point. I can say the amount of time I've been spending on it recently has dramatically increased but its too early to say what will come of that effort.

Posted by: [Erik Smith](#) | [07/07/2014 at 10:27 AM](#)



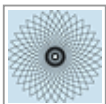
Hello Eric,

As the concept of TDZ is still interesting to me, now, about 4 Years after, I wonder what was the popularity and the adoption rate by the customers?

Thanks,

Igal

Posted by: [IgalK](#) | [09/25/2016 at 03:52 AM](#)



Hi Igal, although Brocade has supported TDZ since FOS 7.4 and I know at least one of our competitors support it as well, I don't have any information about actual adoption. Once our storage arrays support the feature, we'll have a better idea of actual adoption and I'll provide an update at that point.

Posted by: [Erik Smith](#) | [09/27/2016 at 09:51 AM](#)

[« Previous](#)

The comments to this entry are closed.

[Brass Tacks](#) [Powered by TypePad](#)