



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

[Main](#)

12/07/2017

NVMe over Fabrics' Discovery problem

TL;DR - Today, Fibre Channel provides an [NVMe](#) over Fabrics Discovery mechanism, Ethernet does not.

This blog post describes an investigation that was performed to determine if an enhancement to the NVMe over Fabrics Discovery protocol was needed in order for it to support a Network Centric connectivity model as described below. The investigation determined that a multi-layer discovery technique was called for and that FC already provided the required Discovery mechanisms; Ethernet currently does not.

Before I continue I need to thank Dave Peterson (Broadcom) for his assistance with the FC Discovery portion of this investigation.

Management models

In order for NVMe over Fabrics (NVMe-oF) to be widely adopted, I believe it must integrate into existing workflows and management tools (e.g., provisioning and monitoring) that existing enterprises use when managing external (e.g., Array based) storage. As of today, there are three primary approaches that can be utilized for this purpose:

1. A Centralized [Infrastructure as a Service \(IaaS\)](#) Management and Orchestration (M&O) layer. OpenStack is one example of an open source project that provides this functionality, but some users are opting to write their own custom M&O Layer by making use of workflow orchestration frameworks (e.g., Stackstorm), while leveraging tools (e.g., Ansible) and/or extensions to languages (e.g., Pymomi). This type of approach is typically used by larger customers with extensive development resources. This type of approach can be used with both the "End-node Centric" and "Network Centric" provisioning models described below. I should also point out that [the IaaS Missing Link](#) is something that complicates this kind of approach.



[Erik Smith](#)

[Follow](#)

[ARCHIVES](#)

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.

2. The “End-node Centric” management model. This approach requires every host and storage resource to be individually configured and is primarily used in Small and Medium Business (SMB) environments to manage access to storage when NFS/CIFS or iSCSI are being used. The benefit to this approach is “simplicity” and the fact that a centralized discovery service does not need to be installed nor configured. The downside to this model is that it doesn’t scale very well; as the number of nodes increases the administrative burden increases non-linearly.
3. The “Network Centric” management model. This approach utilizes a centralized Discovery Service to enable Hosts to discover the Storage ports that are available to it. This type of approach is used in Medium and Large Enterprise environments to manage connectivity in their Fibre Channel (FC) [Storage Area Networks](#) (SANs). The benefit to this approach is that it scales fairly well (10k+ ports with FC today) and supports State Change Notifications. The downside is a centralized service must be installed and in some cases configured. Another limitation is the requirement that all host and storage subsystem ports register with the centralized Discovery Service.

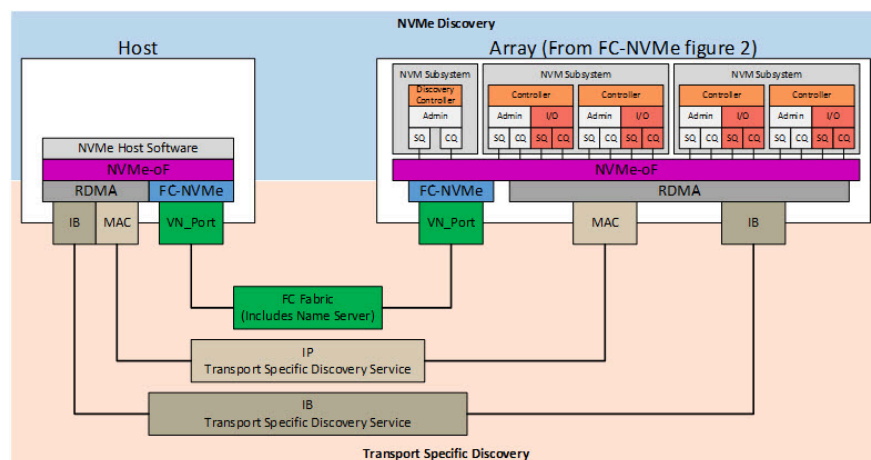
For more information about the “Network Centric” and End-node Centric” management models as well as the scalability attributes of each, please refer to [FC and FCoE versus iSCSI - “Network-centric” versus “End-Node-centric” provisioning](#)

Since NVMe over Fabrics will need to scale (e.g., beyond a single rack) I believe it makes sense for the Transports to at least support a “Network Centric” management model.

In addition, since NVMe over Fabrics will need to co-exist with SCSI and SCSI-FCP, it makes sense to follow a layered discovery approach that is similar to what has been done for discovery with those protocols.

The Network Centric management model

In order to support the “Network Centric” management model, this blog post breaks discovery into two different layers; NVMe and Transport Specific. This separation will allow for the existing NVMe Discovery process to remain intact while allowing each Transport to leverage existing transport specific approaches for the discovery of Transport Addresses that support a specific feature (i.e., a Discovery Service). This concept is illustrated in the diagram below.



The Network Centric approach and how the layered approach will impact each transport is described below.

Transport Specific Discovery

In order to automate the discovery of end devices (e.g., Arrays) that support NVMe over Fabrics, I believe a “centralized”, Transport Specific, Discovery Service is needed. Furthermore, this Discovery Service will need to be able to provide a list of Transport Specific addresses that provide an NVMe Discovery Controller (e.g., that reside on each NVMe Subsystem).

Side note: Transport Specific Discovery is totally optional. Outside of expected behavior, the implementation details regarding a particular Transport Specific Discovery Service will probably not be defined in the NVMe over Fabrics specification.

Although this may seem obvious to some of you, it’s important to explicitly state that each of the protocols would need to handle Transport Specific Discovery in a Transport independent way, for example:

Fibre Channel

The Transport Specific Discovery Service for FC is the Name Server and is defined in the Fibre Channel standard FC-GS. The Discovery Service capability is returned to the Host port in response to “Get N_Port Identifier for the specified FC-4 Feature” (GID_FF) with the FC-4 feature of 28h specified and the FC-4 Discovery Service Feature bit set to 1. When the FC Transport Discovery process is complete, the Host port will have a list of all FC WWPNs and N_Port IDs that provide an NVMe Discovery Controller. This requires that each VN_Port properly register its FC-4 features.

IP

Transport Specific Discovery for IP could be provided in a number of ways, including:

- manual configuration,
- via a configuration management tool (e.g., Ansible, Puppet), or
- a centralized service (as described below) could be deployed and then discovered via DHCP:
 1. The location of the Transport Discovery Service for IP (e.g., [iSNS](#) could be enhanced and used) can be discovered via DHCP.
 2. The Hosts and Discovery Controllers will need to register their NQN, Transport Address and Discovery Service capability with the Transport Discovery Service. These registration commands still need to be defined.
 3. The Hosts and Discovery Controllers will optionally be allowed to register a Discovery Domain (Similar to what is defined for iSNS) with the Transport Discovery Service. The purpose of the Discovery Domain is to limit the number of Discovery Services that each host can access and this serves two purposes:
 - a. It helps avoid scalability issues related to “Discovery and RSCN Storms”
 - b. It provides a coarse grained access control mechanism similar to the role that zoning plays in a FC SAN.

4. The Hosts and Discovery Controllers will be allowed to query the Transport Discovery Service to determine the transport addresses that are available and of those which of them support a Discovery Service. The commands used to query the Transport Discovery Service will also need to be defined.

IB

For IB I'm going to let Mellanox figure this one out. :-)

NVMe Registration and Discovery

After the Transport Specific Discovery process has completed, the Host Software can use the Fabric Connect command and specify the well-known Discovery Service NQN as the destination. For each Transport Specific address that registered as a Discovery Service and was discovered via the Transport Specific Discovery service, the host software will obtain a Discovery Log page as defined in section 5 of the NVMe over Fabrics 1.0 specification.

As shown in the diagram above, this Discovery Service can either be physically located in the same enclosure as the Controller used for IO (e.g., within the same Array enclosure) or it could be centrally located and accessible over an out-of-band network connection (e.g., to support a rack of JBOF). In order to support both of these cases, the information that is registered with the NVMe Discovery Service must allow for a Discovery Log page to be fully populated. Since this is the case, the Register Discovery Log Page command should contain one or more Discovery log pages.

The Register Discovery Log Page command can be used to directly register a log page but other methods not defined in the NVMe over Fabrics specification may also be used (i.e., when the Discovery Service is located within a storage array).

The format of the Register Discovery Log Page command is also TBD.

State Change Notification

With FC, State Change Notification has proven to be amazingly important for operational concerns. Essentially, you need to know when a device goes away ASAP and RSCN allows for this.

Similar to the approach used with Discovery, the mechanisms used to provide State Change Notification should also be layered into Transport specific and NVMe specific.

Transport Specific

Each Transport may support a mechanism to indicate that a change has occurred in the transport that may affect connectivity.

Fibre Channel - Fibre Channel provides Asynchronous state change notification via Registered State Change Notification (RSCN)

IP - I am honestly not sure what could be done here short of adding this functionality to an iSNS like centralized controller.

IB - Again, I'll defer to Mellanox.

NVMe Specific

State change notification can also be provided between NVMe host software and each controller through the use of the following approaches:

- Asynchronous Notification for optional events as defined in NVMe over Fabrics 1.0.
- Keepalive based monitoring between the NVMe host software and each controller.
- A combination of keepalive based monitoring between the NVMe host software/Controller/Discovery Controller and the Transport Discovery Service; with the Transport Discovery Service transmitting an asynchronous event when a keepalive timeout occurs

All three of these approaches will need to be defined.

Conclusion

As [we've said before](#), NVMe over Fabrics needs a bit more work before it will be ready for use by the majority of our Enterprise Customers. Along these lines, I believe that Discovery and State Change Notification are features that will be critically important to you as you start to evaluate different Transports.

Thanks for reading!

Posted at 10:56 AM | [Permalink](#)



Comments

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

The comments to this entry are closed.

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