D. Otis
Internet Draft SANlight
Document: draft-otis-scsi-ip-01.txt 10/9/00
Category: Informational Expires 4/9/01

SCSI over IP

Status of this Memo

1. Abstract

This is an overview of SCSI over IP considerations leading to the FC-SCTP-IP draft and to suggest possible implementations of this FC-SCTP-IP draft. With two basic architectures covered, it is the intent to illustrate decisions leading to both simple FC encapsulation as well as native IP access.

2. Basic Architectures

The optical delay between facilities is 8 microseconds per mile or 5 microseconds per kilometer. Limiting Metro Area Networks (MAN) to a distance of 125 fiber miles results in 1 millisecond point-to-point delay. A Wide Area Network (WAN) of 15K miles results in a point-to-point delay of 120 milliseconds and used as example distances.

Direct access to the storage device provides the best performance for a one-to-one association between a client access point and the target device otherwise benefit of caching is reduced by network delay. Another consideration is communication buffering which can add a significant delay and needs to be considered part of the architecture.

Network delays necessitate placing caching for storage adjacent to the client otherwise performance benefits are significantly reduced. Caching placed adjacent to storage may be justified if shared by multiple remote clients as a means to reduce load on the storage device. Such shared storage becomes problematic unless restricted to read-only, otherwise locking/releasing of associated records before and after access operations adds additional delay and complexity.

In reviewing the basic architectures, cache adjacent to storage will be considered for read-only volumes if used on a MAN or WAN because of their multi-client use, otherwise abstraction of the basic block device becomes appropriate. Such abstraction could be a file or database server as example. These abstractions simplify locking/releasing of disparate blocks into a simpler opening/closing of objects. This brings architectures down to the following:

- One-to-One Client Side Cache (CSC)
- Many-to-One Server Side Cache (SSC)(read-only unless on LAN)

3. Why not just use abstraction servers?

Abstraction servers in this case will be defined as any server abstracting blocks and partitions into objects that conjoin a combination of information into a presentation unrelated to the original block information. This could be as HTTP, file, database or authentication server. To ensure reliability of abstraction servers, the device level interface Storage Area Network(SAN) must be accessible to more than one such server. The amount of RAM required by abstraction servers typically run above 1% of volume space to prevent thrashing on references of extent, namespace and permissions as example. RAM is about 100 times the expense of hard storage, so abstracted data doubles cost. In addition, abstraction servers are sized for the number of clients and volume space making scaling difficult. By allowing clients remote access to the SAN in an exclusive fashion using their own abstraction servers, the best performance is achieved and servers are properly scaled.

Direct SAN access would be through a translation of fibre-channel at a switch aggregating FC nodes into a single SCTP IP connection. Simple encapsulation ensures no state information is remembered at any translation node. Ethernet provides a great deal of flexibility with respect to possible data rates on the IP connection and paths. 1G-bit or encapsulated as a single connection on four 1G-bit ports with load balanced redundant paths as example. Shortly, 10G-bit ports will be available. A review is required to determine if FC structures are suitable.

4. Does FC or FCP structures need to change?

In examining a minimal case of one device using FCP structures, the sequence rate is examined. Normal use will increase the number of devices reducing the demand of concurrent sequences. 10-kilometer fiber places only 3 frames of data in flight at 1G-bit. At a MAN length, this number is increased to 60-frames and, at WAN, it rises to 7K-frames. This number rises to 600/70K-frames should the connection run at 10G-bit. A 512 concurrent exchanges (256 each direction) sequence limit of FC or of commands appears to be a problem as only 1K-sequences/second could be supported at this WAN distance.

With such a WAN however, only 4 responses per second would be possible which would imply 256 concurrent threads would be required to satiate this concurrent sequence limit. Over such large distances, only applications demanding large transfers of 1.2M-bytes would a maximum bandwidth of 10G-bit be achieved from a single initiator to a single device. However, the reason for such a SSC being used would be to support multiple clients, and each additional client reduces the requisite transfer for maximum connection utilization. Should the average transfer be 8K-bytes, then 160 ATM clients would be required if all are at the example WAN distance running 10G-bit from the server supporting a single device.

Of course, there would be more than one device being accessed per client. Additional devices as well as additional clients ensure bandwidth is not constrained by the protocol or even rather limited outstanding commands.

For the MAN distance at 10G-bit, FC would be limited to 128K-sequences per second with 500 responses per second which again implies 256 concurrent threads would be required to satiate. Should the average transfer be 8K-bytes then only two clients limited by 40% would consume the 10G-bit connection. Once at a LAN distance, a concurrent sequence limit would never become significant even at 10G-bit, as there would be only 30 frames of data in flight at the 10-kilometer distance.

With practical considerations for a SSC, should the concurrent sequence constraint become a limit for a particular remote client, these parallel processes could be identified as either different initiators related to a sub-processor or simply isolated on a different stream should the cache support native SCTP access. Regardless of the concurrent sequence constraint of FC, a storage device is mechanically limited and can only provide about 200 accesses per second. As such, when directly accessing the drive, regardless of the data-rate or distance, the number of sequences is still bound by the drive. This mechanical limit will not be supplanted for dozens of generations, even with WAN use.

With most commerce, the cache hit rate is relatively low, so it is doubtful that even without sequence expansion by means of multiple streams, there would be an FC protocol limitation on a single initiator on a single stream. The prominent use of a SCSI-IP protocol however, will be to communicate directly to the storage device and so keeping state information within the domain of client and device server and not at an intermediary node is an important consideration for reliability. Re-engineering the FC header structures runs a significant risk by introducing stateful translations. In the end, it will be hard to justify modification of these structures should reliability suffer as a result.

5. Flow Control at the Sub-Node

It is important to prevent internal nodes within a SSC configuration from becoming congested or allowed to consume excessive resources. The communication buffer is a separate resource from the FIFO delivering to the sub-nodes that obtain access to the various RAID or JBOD. To provide a deterministic response from sub-nodes, the depth of the FIFO must be controlled. As little as 8M-byte of FIFO would introduce 100 milliseconds of delay at typical FC rates. As such, resolution of these nodes must be accommodated. By attaching each FIFO to a SCTP stream, this control then resolves to the FIFO. The stream can also have priorities assigned to allow urgent commands a means of bypassing lower priority commands or of being processed out of order.

The flow control works in a similar fashion to that of FC Class-3 and only Class-3 is indicated as being supported for FC media. Rather than single credit tokens "R_RDY" being sent, a token count relays the buffer credits being returned to the stream. Flow control resolves to the stream and not the initiator unless each initiator is mapped to a stream. This is left to the implementer. The typical case would be to have fewer than 20 drives connected to an FC node that feeds the SCTP stream. As there are 65,535 streams available, 1.3M-drives could be accommodated by the SCTP protocol with a single connection. It is doubtful that more than a few dozen FC nodes would be found on a single device however.

6. Why SCTP

The essential advantages for using SCTP compared to TCP:

- Headers contained within one frame.
- Objects aligned at 32-bit boundaries.
- Out of sequence frame processing.
- Standard authentication.
- Independent streams under common control.
- Session restart.
- Improved error detection.
- Prevention of blind spoofing and denial of service attacks.
- Standard Heartbeat and multi-homing. (Optional)

With VI promoted for new file systems, it is clear CPU and memory overhead are two areas receiving attention. SCSI does not require VI to allow zero copy data placement or out-of-sequence processing, however reliance on the encapsulation structure as well as the ability to identify object boundaries within the stream are highly important. To date, TCP does not allow for these features and with the sizeable install base, it is doubtful it ever will as SCTP provides requisite features for implementing new versions of VI, RPC, and SCSI allowing out-of-sequence delivery and meeting the congestion control requirement found with TCP. Those expecting to change TCP as a means to offer a solution for the problem of locating objects within a persistent stream may be surprised by the rapidity SCTP fills that void and preempts these efforts. The API for TCP simply does not provide the type of interface required nor does it provide for the reliability. When it comes to reliability of storage, little can be said to be more paramount. Perhaps only performance would receive as much scrutiny and SCTP wins on both accounts for several reasons.

Use of SCTP also allows a low overhead for simple encapsulation as well as providing the means for native IP access. Rather than divergent standards, SCTP offers a unified means of communicating to SAN. TCP does not provide that ability.

8. Bootstrapping

Dynamic Host Configuration Protocol is the first step in bootstrapping; Lightweight Directory Access Protocol should be the second. With these very powerful tools, all overhead needed to communicate with SCSI is done prior to making any connections. LDAP would contain knowledge of SCSI defined as a SCSI service schema. This technique avoids all real-time authentications to allow SCSI transport to scale. The SCSI schema naming conventions for the boot drive may take the form NETSCSIBOOT:XXXXXXXXXXXX where the hexadecimal text string of the MAC address of the booting machine is used as a user name to match against the special drive name.

A schema for a SCSI service may look something like:

```
   Object Class: SCSI IP Network Services
     Description: Used to define Network

     SIPNSMacro: SCSINET OBJECT-CLASS
         SUBCLASS Portal
         MUST CONTAIN {
             Primary_IP,
             T_PROT,
             E_PROT,
             Targets,
             Permission}
         MAY CONTAIN {
             Secondary_IP,
             Internal_IP}
     TARGET_DEF OBJECT-CLASS
         SUBCLASS OF Targets
         MAY CONTAIN {
                     Port_Identifier,
                     Port_WWN,
                     LUNS,
                     Link}

     LUN_DEF OBJECT-CLASS
```

SUBCLASS OF LUNS

```
         MAY CONTAIN {
                     HI_LUN,
                     WWNNS}...
```

Standardizing using LDAP rather than vendor specific tools ensures more rapid acceptance and use of this protocol both within Internet and in enterprise environments. In single user scenarios, a simple flat file may suffice in defining SCSI services either as registry entries or as /etc files.

The storage provider would only advertise the authentication server via a DNS to the public. If the client's browser had a plug-in that knew how to talk to a SCSI device, it could allow the user to type SCSI://my.storage.com/my_stuff and a pop-up would request a password or use a stored password to then access the authentication server at this location to look for the drives under my_stuff. Once the needed information was exchanged between the authentication server and the client, the SCSI driver would then have all the binary information required to access the SCSI portal (not advertised via DNS). The authentication server would return a structure as indicated prior together with a one-time secret for a cookie exchange. LDAP has a Java interface, so perhaps Java was used. There is sufficient documentation for accessing LDAP, whereas there is little if any for vendor specific management tools. Vendor specific management tools could easily construct a database exchange that would populate the documented LDAP database however.

To assist in allowing a bank of drives to be mounted in a fashion suitable for any contained applications, drive mounting information should be included as part of the access information. This mounting information could simply be a name associated with the user for each logical unit. Although at the device level, LUN is typically set to zero, there is no assurance this additional addressing will not be useful in the future. Presently, drives are partitioned using OS file system mapping. These partitions are not protected beyond the constructs of the OS. For marketing reasons, should a large drive be divided into smaller units, a partition is not a suitable unit of division. OS tools from different vendors often corrupt information within the partition tables. Making use of the logical unit offers a more secure and OS independent means of dividing the drive. The logical unit may be nothing more than a drive supported partitioning convention.

Should there be a third-party command that is required to transverse the IP, it should be a port on the backside of a portal that has already been connected to yet another portal. This connection may have been established in response to the authentication or done in a prior fashion. The port on the back of the portal would have a SCSI address and would map into yet another SCSI address within the realm of the other Portal. Again, the client would not handle this translation nor should it be as it would be in the domain of the provider. The provider would be required to make the permission and translation table prior to authentication. Perhaps the translation table was made at the time of installation. At no point in time, would the client be able to change this table. This SCSI space would be as defined in the permission list and remains static upon authentication. Allowing a dynamic mapping of target address would prevent any scalable means of authentication.

9. Authentication and Transparent Bridges

The SCSI Portal or SCSI device works within SCSI address space and not the IP address space. Upon authentication, only a permission list is checked against the client to validate a frame. The translation of SCSI address space through a transparent bridges or to different targets is not allowed to change for addresses already defined within an active permissions list. Such changes would negate permissions already assigned. The transport is not allowed to change any translations of SCSI addresses, as this would require extensive checking and authentication in real-time, which would be prohibitive and not scaleable.

```
Connection Diagram
                              (Target /bin:Foo)
                              (Target /:Foo)
                              (Target /<swap>:Foo)
                              (Target /usr:Foo)
                              (Target /home:Foo)
                                     |  \ (Third-Party Address)
        (SCSI Address Space)         |   \
(Accessible from Client Gateway) SCSI Portal 1
                                     |    \
                                     |     \
                                   (NAT)    \
                                     |       \  (IP Address Space)
                                 (GATEWAY)    \
                                     |         \
                                Client Foo      \
                                                 \
                                           SCSI Portal 2
        (SCSI Address Space)                     |
                                            (Target /opt:Foo)
```

In this diagram, the client Foo communicates to the gateway on the LAN. From this gateway, a non-routable IP may be swapped for a routable IP using a NAT. This could be the entry point for a tunnel as well. Once connected to the SCSI Portal 1, the SCSI transport only contains SCSI Addresses, (S_ID and_D_ID as example). In the case of a third-party command, a port (addressed by the third-party command) on the back of the SCSI Portal 1 travels through yet another IP connection to SCSI Portal 2. This connection is hidden from Client Foo. This creates a transparent bridge between Portal 1 and Portal 2. Client Foo was not given any information as to what was required to travel between Portal 1 and Portal 2. This information is within the purview of the provider and is not shared with the client. The client's view within Portal 1 is as a pure SCSI space defined by a permission list even though the third-party target ID may have been translated to conform to Portal 2 address space. The target ID remains a SCSI address throughout.

This technique prevents the Portal from doing any real-time authorizations, SCSI-IP name lookups, or dynamic address translations. Permissions, and addresses are fixed once a connection is made and any bridges are established as needed with information provide prior to connection acceptance.

10. Acknowledgments

Randall R. Stewart [randall@stewart.chicago.il.us]
For his timely inputs on SCTP making this far more enjoyable.

11. Author's Addresses

Douglas Otis
SANlight Inc.
160 Saratoga #46
Santa Clara, CA 95051

Phone: (408) 260-1400 x2001
Email: dotis@sanlight.net

Full Copyright Statement