IPS
Internet Draft
<draft-ietf-ips-isns-08.txt>
Standards Track
Expires August 2002

Josh Tseng
Kevin Gibbons
Charles Monia
Nishan Systems

Franco Travostino
Nortel Networks

Tom McSweeney
Curt Du Laney
John Dowdy
IBM

Chad Gregory
Intel

February 2002

Internet Storage Name Service (iSNS)

Status of this Memo

Acknowledgements

Numerous individuals contributed to the creation of this draft through their careful review and submissions of comments and recommendations. We acknowledge the following persons for their technical contributions to this document: Mark Bakke (Cisco), John Hufferd (IBM), Julian Satran (IBM), Kaladhar Voruganti(IBM), Joe Czap (IBM), Jim Hafner (IBM), Yaron Klein (Sanrad), Larry Lamers (SAN Valley), Jack Harwood (EMC), David Black (EMC), David Robinson

(Sun), Joe Souza (Microsoft), Alan Warwick (Microsoft), Bob Snead

(Microsoft), Fa Yeou (Nishan), Ken Hirata (Vixel), Howard Hall, and Marjorie Krueger (HP).

Comments

Comments should be sent to the IPS mailing list (ips@ece.cmu.edu) or to the authors.

Table of Contents

Status of this Memo....1
Acknowledgements....1
Comments....2
1. Abstract....6
2. Conventions used in this document....6
3. iSNS Overview....6
3.1 iSNS Architectural Components....6
3.1.1 iSNS Protocol (iSNSP)....6
3.1.2 iSNS Client....7
3.1.3 iSNS Server....7
3.1.4 iSNS Database....7
3.1.5 iSCSI....7
3.1.6 iFCP....7
3.2 iSNS Functional Overview....7
3.2.1 Name Registration Service....8
3.2.2 Discovery Domain and Login Control Service....8
3.2.3 State Change Notification Service....9
3.2.4 Open Mapping Between Fibre Channel and iSCSI Devices....10
3.3 iSNS and Domain Name System (DNS)....11
3.4 iSNS and LDAP....11
3.5 iSNS Server Discovery....12
3.5.1 Service Location Protocol (SLP)....12
3.5.2 Dynamic Host Configuration Protocol (DHCP)....12
3.5.3 iSNS Heartbeat Message....12
3.6 iSNS and NAT....12
3.7 Transfer of iSNS Database Records between iSNS Servers....13
3.8 Backup iSNS Servers....15
3.9 Deployment Architecture Diagram....16
4. iSNS Object Model....16

4.1 NETWORK ENTITY Object....17
4.2 PORTAL Object....17
4.3 STORAGE NODE Object....17
4.4 FC DEVICE Object (iFCP Only)....17
4.5 DISCOVERY DOMAIN Object....17
4.6 DISCOVERY DOMAIN SET Object....18
4.7 iSNS Database Model....18
5. iSNS Implementation Requirements....18
5.1 iSCSI Requirements....18
5.1.1 Required Attributes for Support of iSCSI....19
5.1.2 Example iSCSI Object Model Diagrams....20
5.1.3 Required Commands and Response Messages for Support of iSCSI..21
5.2 iFCP Requirements....22
5.2.1 Required Attributes for Support of iFCP....22
5.2.2 Example iFCP Object Model Diagram....23
5.2.3 Required Commands and Response Messages for Support of iFCP...24
5.3 Attribute Descriptions for Discovery Domain Registration....26
5.4 Use of TCP For iSNS Communication....27
5.5 Use of UDP For iSNS Communication....28
6. iSNS Message Attributes....28
6.1 iSNS Attribute Summary....28
6.2 Entity Identifier-Keyed Attributes....31
6.2.1 Entity Identifier (EID)....31

6.2.2 Entity Protocol....31
6.2.3 Management IP Address....32
6.2.4 Entity Registration Timestamp....32
6.2.5 Protocol Version Range....32
6.2.6 Registration Period....32
6.2.7 Entity Index....33
6.2.8 Entity ISAKMP Phase-1 Proposals....33
6.2.9 Entity Certificate....33
6.3 Portal-Keyed Attributes....34
6.3.1 Portal IP-Address....34
6.3.2 Portal TCP/UDP Port....34
6.3.3 Portal Symbolic Name....34
6.3.4 Entity Status Inquiry Interval....34
6.3.5 ESI Port....35
6.3.6 Portal Group....35
6.3.7 Portal Index....35
6.3.8 SCN Port....36
6.3.9 Security Bitmap....36
6.3.10Portal ISAKMP Phase-1 Proposals....36
6.3.11Portal ISAKMP Phase-2 Proposals....37
6.3.12Portal Certificate....37

6.4 iSCSI Node-Keyed Attributes....37
6.4.1 iSCSI Name....37
6.4.2 iSCSI Node Type....37
6.4.3 iSCSI Node Alias....38
6.4.4 iSCSI Node SCN Bitmap....38
6.4.5 iSCSI Node Index....39
6.4.6 EUI64 Token....39
6.4.7 iSCSI Node Certificate....40
6.5 FC Port-Keyed Attributes....40
6.5.1 Port Name (WWPN)....40
6.5.2 Port ID....40
6.5.3 Port Type....40
6.5.4 Symbolic Port Name....41
6.5.5 Fabric Port Name (FWWN)....41
6.5.6 Hard Address....41
6.5.7 Port IP Address....41
6.5.8 Class of Service (COS)....41
6.5.9 FC-4 Types....42
6.5.10FC-4 Descriptor....42
6.5.11FC-4 Features....42
6.5.12iFCP SCN Bitmap....42
6.5.13iFCP Port Type....42
6.5.14Port Certificate....43
6.6 Node-Keyed Attributes....43
6.6.1 Node Name (WWNN)....43
6.6.2 Symbolic Node Name....43
6.6.3 Node IP Address....44
6.6.4 Node IPA....44
6.6.5 Node Certificate....44
6.6.6 Proxy iSCSI Name....44
6.7 Other Attributes....44
6.7.1 FC-4 Type Code....44
6.7.2 iFCP Switch Name....44

6.7.3 Preferred ID....45
6.7.4 Assigned ID....45
6.7.5 Space_Identifier....45
6.7.6 Server-Specific Attributes....45
6.8 Discovery Domain Registration Attributes....45
6.8.1 iSNS Discovery Domain Attribute Summary....46
6.8.2 DD Set ID Keyed Attributes....46
6.8.3 DD ID Keyed Attributes....47
6.9 Vendor-Specific Attributes....48
6.10 Company OUI....49
6.11 Standards-Based Extensions....49

7. iSNSP Message Format....49
7.1 iSNSP PDU Header....49
7.1.1 iSNSP Version....49
7.1.2 iSNSP Function ID....49
7.1.3 iSNSP PDU Length....50
7.1.4 iSNSP Flags....50
7.1.5 iSNSP Transaction ID....50
7.1.6 iSNSP Sequence ID....50
7.2 iSNSP Message Segmentation and Reassembly....50
7.3 iSNSP Message Payload....51
7.3.1 Attribute Value 4-Byte Alignment....51
7.4 iSNSP Response Error Codes....51
7.5 iSNS Multicast Message Authentication....52
7.6 Registration and Query Messages....53
7.6.1 Source Attribute....54
7.6.2 Key Attributes....54
7.6.3 Delimiter Attribute....55
7.6.4 Operating Attributes....55
7.6.5 Registration and Query Message Types....55
7.7 Response Messages....66
7.7.1 Error Code....66
7.7.2 Key Attributes in Response....66
7.7.3 Delimiter Attribute in Response....67
7.7.4 Operating Attributes in Response....67
7.7.5 Registration and Query Message Types....67
7.8 Vendor Specific Messages....71
8. Security Considerations....71
8.1 iSNS Security Threat Analysis....71
8.2 iSNS Security Implementation Requirements....71
8.3 Using iSNS to Discover Security Requirements of Peer Devices...73
8.4 Using iSNS to Configure Security Policies of Client Devices....73
8.5 Resource Issues....73
8.6 iSNS Interaction with IKE and IPSec....74
9. Normative References....75
10. Informative References....76
11. Author's Addresses....77
Full Copyright Statement....78
Appendix A -- iSNS Examples....79
A.1 iSCSI Initialization Example....79
A.1.1 Simple iSCSI Target Registration....79
A.1.2 Target Registration and DD Configuration....80
A.1.3 Initiator Registration and Target Discovery....81

1. Abstract

This document provides a generic framework centering around use of the iSNS for discovery and management of iSCSI and Fibre Channel (FCP) storage devices in an enterprise-scale IP storage network. iSNS is an application that stores iSCSI and FC device attributes and monitors their availability and reachability in an integrated IP storage network. Due to its role as a consolidated information repository, iSNS provides for more efficient and scalable management of storage devices in an IP network.

2. Conventions used in this document

iSNS refers to the framework consisting of the storage network model and associated services.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

All frame formats are in big endian network byte order.

3. iSNS Overview

The objective of iSNS is to facilitate scalable configuration and management of iSCSI and Fibre Channel (FCP) storage devices in an IP network. iSNS allows the administrator to go beyond a simple device-by-device management model, where each storage device is manually and individually configured with its own list of known initiators and targets. Using the iSNS, each storage device subordinates its discovery and management responsibilities to the iSNS server. The iSNS server thereby serves as the consolidated management contact through which administrator workstations can configure and manage the entire storage network, including both iSCSI and Fibre Channel devices.

iSNS can be implemented to support iSCSI and/or iFCP protocols as needed; an iSNS implementation MAY provide support for one or both of these protocols as desired by the implementer. Implementation requirements within each of these protocols is further discussed in section 5. Use of iSNS is OPTIONAL for iSCSI, and REQUIRED for iFCP.

3.1 iSNS Architectural Components

3.1.1 iSNS Protocol (iSNSP)

The iSNS Protocol (iSNSP) is a flexible and lightweight protocol that specifies how iSNS clients and servers communicate. It is suitable for various platforms, including switches and targets as well as server hosts.

3.1.2 iSNS Client

iSNS clients initiate transactions with the iSNS server using the iSNSP. iSNS clients are applications that are co-resident in the storage device, and can register deviceÆs attribute information, download information about other registered clients in a common Discovery Domain (DD), and receive asynchronous notification of topology events that occur in their DD(s). Management stations are a special type of iSNS client that have access to all DDs stored in the iSNS.

3.1.3 iSNS Server

The iSNS server responds to iSNS protocol queries and requests, and initiates iSNS protocol State Change Notifications. Properly authenticated information submitted by a registration request is stored in an internal or external iSNS database.

3.1.4 iSNS Database

The iSNS database is the information repository for the iSNS server(s). It maintains information about iSNS client attributes. A directory-enabled implementation of iSNS may store client attributes in an LDAP directory infrastructure.

3.1.5 iSCSI

iSCSI (Internet SCSI) is an encapsulation of SCSI for a new generation of storage devices interconnected with TCP/IP.

3.1.6 iFCP

iFCP (Internet FCP) is a gateway-to-gateway protocol designed to interconnect existing Fibre Channel and SCSI devices using TCP/IP. iFCP maps the existing FCP standard and associated Fibre Channel services to TCP/IP.

3.2 iSNS Functional Overview

iSNS Protocol registration and query messages are sent by iSNS clients to servers, while notification messages are sent by iSNS servers to iSNS clients. Messages originating at the client are

sent to the iSNS server at the well-known iSNS TCP or UDP port number.

There are four main functions of the iSNS:

1) A Name Service Providing Storage Resource Discovery

2) Discovery Domain (DD) and Login Control Service

3) State Change Notification Service

4) Open Mapping of Fibre Channel and iSCSI Devices

3.2.1 Name Registration Service

The iSNS provides a registration function to allow all entities in a storage network to register and query the iSNS database. Both targets and initiators can register in the iSNS database, as well as query for information about other initiators and targets. This allows, for example, a client initiator to obtain information about target devices from the iSNS server. This service is modeled on the Fibre Channel Generic Services Name Server described in FC-GS-3, with extensions, operating within the context of an IP network.

The naming registration service also provides the ability to obtain a network unique Domain ID for iFCP gateways when required.

3.2.2 Discovery Domain and Login Control Service

The Discovery Domain (DD) Service facilitates the partitioning of iSNS client devices into more manageable groupings for administrative and login control purposes. This allows the administrator to limit the login process to the more appropriate subset of targets registered in the iSNS. iSNS clients must be in at least one common DD in order to obtain information about each

other. iSNS clients can be a member of multiple DD's simultaneously.

The DD information stored in the iSNS can be used by various enforcement points in the network to configure security and access control policy. For example, a DD-aware switch can block storage initiators from accessing targets that are not in the same DD, even if the initiator somehow obtained address information for a target outside of its DD. This functionality is the equivalent of the ôHard Zoningö functionality in a Fibre Channel network. Similarly, Discovery Domains are similar to VLANs in an Ethernet network. An implementation may decide to use the Discovery Domain functionality in iSNS to configure and establish VLANs that enforce DD access restrictions.

Login Control allows targets to subordinate their access control/authorization policy to the iSNS server. The target node or device downloads the list of authorized initiators from the iSNS. Each node or device is uniquely identified by an iSCSI Name or Port Name (iFCP). Only initiators that match the required identification and authentication information provided by the iSNS will be allowed access by that target node or device during session establishment.

If spoofing of initiator identities is a concern, the target may use the public key certificate of the authorized initiator, obtained from the iSNS server, to authenticate the initiator.

DD's can be managed offline by a separate management workstation, through the iSNSP or through SNMP. If the target opts to use the Login Control feature of the iSNS, the target subordinates management of access control policy (i.e., the list of initiators allowed to login to that target) to the management workstations that are manipulating information in the iSNS database.

If administratively authorized, a target can upload its own Login Control list. This is accomplished using the DDReg message and listing the iSCSI Name of each initiator to be registered in the Target's DD.

Depending on the implementation, newly registered devices that have not explicitly been placed into a DD by the management station MAY be placed into a "default DD" where they are visible to other devices in that DD. Other implementations MAY decide that they are registered with no DD, making them inaccessible to source-scoped iSNSP messages.

The iSNS server MUST use every iSNSP message containing the SOURCE field to determine the source of the request and scope the operation to the set of Discovery Domains that the iSNS client is a member of. In addition, the SOURCE field MAY also be used to determine whether the specified node is authorized to perform the specified iSNS operation. For example, an iSNS server implementation may decide that only CONTROL nodes (identified by the iFCP or iSCSI Node Type bitmap) are authorized to create or delete discovery domains.

Valid and active Discovery Domains (DD's) belong to at least one active Discovery Domain Sets (DDS's). Discovery Domains that do not belong to an activated DDS are not enabled.

3.2.3 State Change Notification Service

The State Change Notification (SCN) service allows the iSNS to issue notifications about network events that affect the operational state of iSNS clients. The iSNS client has the ability to register for these notifications of events detected by the iSNS. The types of events for which SCNs can be sent include change in Discovery Domain (DD) membership and device registration updates.

The State Change Notification service utilizes the Discovery Domain Service to control the distribution of notification messages. Notifications about changes within a DD are limited to members of that DD.

If the iSNS is unable to service an SCN registration it SHALL reject the SCN registration request, returning a SCN Registration Rejected error code. The rejection might occur in situations where the network size, or current level of SCN registrations, has passed an

implementation-specific threshold. A client not allowed to register for SCNs SHOULD monitor its sessions with other storage devices directly.

The specific notification mechanism by which the iSNS learns of the events is implementation-specific, but can include examples such as explicit notification messages from an iSNS client to the iSNS server, or a hardware interrupt to a switch-hosted iSNS as a result of link failure. The State Change Notification is equivalent to the Fibre Channel State Change Notification service, with extensions, operating within the context of an IP network.

3.2.4 Open Mapping Between Fibre Channel and iSCSI Devices

The iSNS database stores naming and discovery information about both Fibre Channel and iSCSI devices. This allows the iSNS to store mappings of a Fibre Channel device to a proxy iSCSI device "image" in the IP network. Similarly, mappings of an iSCSI device to a "proxy WWN" can be stored under the EUI64 field for that that iSCSI device.

Furthermore, Fibre Channel-aware management stations that interact with the iSNS server can retrieve information about Fibre Channel devices, and use this information to manage Fibre Channel devices as well as iSCSI devices. This allows management functions such as Discovery Domains and State Change Notifications to be seamlessly applied for both iSCSI and Fibre Channel devices, facilitating integration of IP networks with Fibre Channel devices and fabrics.

Note that Fibre Channel attributes are stored as iFCP attributes, and the ability to store this information in the iSNS server is useful even if the iFCP protocol is not implemented. In particular, tag 101 can be used to store a "Proxy iSCSI Name" for Fibre Channel devices registered in the iSNS. This field is used to associate the FC device with an iSCSI registration entry that is used for the Fibre Channel device to communicate with iSCSI devices in the IP network. Conversely, tag 37 contains an EUI64 token field, which can be used to store an FC Node Name (WWNN) value used by iSCSI-FC gateways to represent an iSCSI device in the Fibre Channel domain.

By storing the mapping between Fibre Channel and iSCSI devices in the iSNS, this information becomes open to any iSNS client wishing to retrieve and use this information. In many cases, this provides advantages over storing this information internally within an iSCSI-FC gateway, where the mapping is inaccessible to other devices except by proprietary mechanisms.

3.3 iSNS and Domain Name System (DNS)

A directory-enabled iSNS implementation may use LDAP to store iSNS client attributes. If this is the case, then LDAP can be used to support both the iSNS and DNS server infrastructures, maintaining consistency in Domain Name-to-IP address mappings used by DNS and iSNS.

A detailed description of the Domain Name System (DNS) protocol is found in [RFC 1035], and is beyond the scope of this document. If a common LDAP information base is used to support both DNS and iSNS servers, then Domain-Name-to-IP address mappings for storage devices can be obtained from either DNS servers or the iSNS.

3.4 iSNS and LDAP

LDAP is a generic protocol to access directory services through the network. It is a passive service designed to deliver scalable directory services using a get/set model. Applications designed and tailored to specific user requirements interact with LDAP for their generic directory service needs. On the other hand, iSNS is an application that goes beyond the simple get/set model, and provides specific capabilities needed to monitor and manage an enterprise-scale storage network. iSNS is one example of an application that can leverage the services of LDAP. By layering iSNS on top of LDAP, the capabilities of both iSNS and LDAP can be leveraged to manage and scale the enterprise IP storage network.

The iSNS application provides capabilities that LDAP alone is not designed to achieve. This includes the following:

1) Client Attribute Awareness - The iSNS server application interprets attribute values submitted by clients in registration messages, and can take appropriate action based upon specific registered attribute values. The iSNS server is conscious of the state of each client.

2) State Change Notification - An iSNS server may initiate notification messages to clients in the event of a change in the network, such as the non-availability or non-reachability of a storage device, or a specific change of a client attribute.

3) Monitoring of Clients - iSNS provides an Entity Status Inquiry message to verify the availability and reachability of storage devices.

4) Lightweight - iSNSP is a simple and lightweight protocol suitable for implementation on embedded devices such as switches and targets. There are no unused or "wasted" features that may bog down the performance of the host device.

LDAP provides important capabilities that can be used to increase the scalability of iSNS services. For example, LDAP provides

database replication capabilities (LDUP), which can be used to support iSNS deployments with multiple iSNS servers.

3.5 iSNS Server Discovery

3.5.1 Service Location Protocol (SLP)

The Service Location Protocol (SLP) provides a flexible and scalable framework for providing hosts with access to information about the existence, location, and configuration of networked services, including the iSNS server. SLP MAY be used by iSNS clients to discover the IP address of the iSNS server. To implement discovery through SLP, a Service Agent (SA) should be cohosted in the iSNS server, and a User Agent (UA) should be in each iSNS client. Each client multicasts a discovery message requesting the IP address of the iSNS server(s). The SA responds to this request. Optionally, the location of the iSNS can be stored in the SLP Directory Agent (DA).

Note that a complete description and specification of SLP can be found in [RFC2608], and is beyond the scope of this document. Additional details on use of SLP to discover iSNS can be found in [iSCSI-SLP].

3.5.2 Dynamic Host Configuration Protocol (DHCP)

The IP address of the iSNS server can be stored in a DHCP server to be downloaded by iSNS clients using a DHCP option. The DHCP option number to be used for distributing the iSNS server location is <<TBD>>.

3.5.3 iSNS Heartbeat Message

The iSNS heartbeat message is described in section 7.6.5.14. It allows iSNS clients within the broadcast or multicast domain of the iSNS server to discover the location of the active iSNS server and any backup servers.

3.6 iSNS and NAT

The existence of NAT will have an impact upon information retrieved from the iSNS. If the iSNS client exists in a different addressing domain than the iSNS server, then IP address information stored in

the iSNS server may not be correct when interpreted in the domain of the iSNS client.

There are several possible approaches to allow operation of iSNS within a NAT network. The first approach is to require use of the canonical TCP port number by both targets and initiators when addressing targets across a NAT boundary, and for the iSNS client to not query for nominal IP addresses. Rather, the iSNS client initiator queries for the DNS Fully Qualified Domain Name stored in the Entity Identifier field, when seeking addressing information. Once retrieved, the DNS name can be interpreted in each address

domain and mapped to the appropriate IP address by local DNS servers.

A second approach is to deploy a distributed network of iSNS servers. Local iSNS servers are deployed inside and outside NAT boundaries, with each local server storing relevant IP addresses for their respective NAT domains. Updates among the network of decentralized, local iSNS servers are handled using LDAP and using appropriate NAT translation rules implemented within the update mechanism in each server.

The final alternative is to simply disallow use of NAT in communication between the iSNS server and any iSNS client.

3.7 Transfer of iSNS Database Records between iSNS Servers

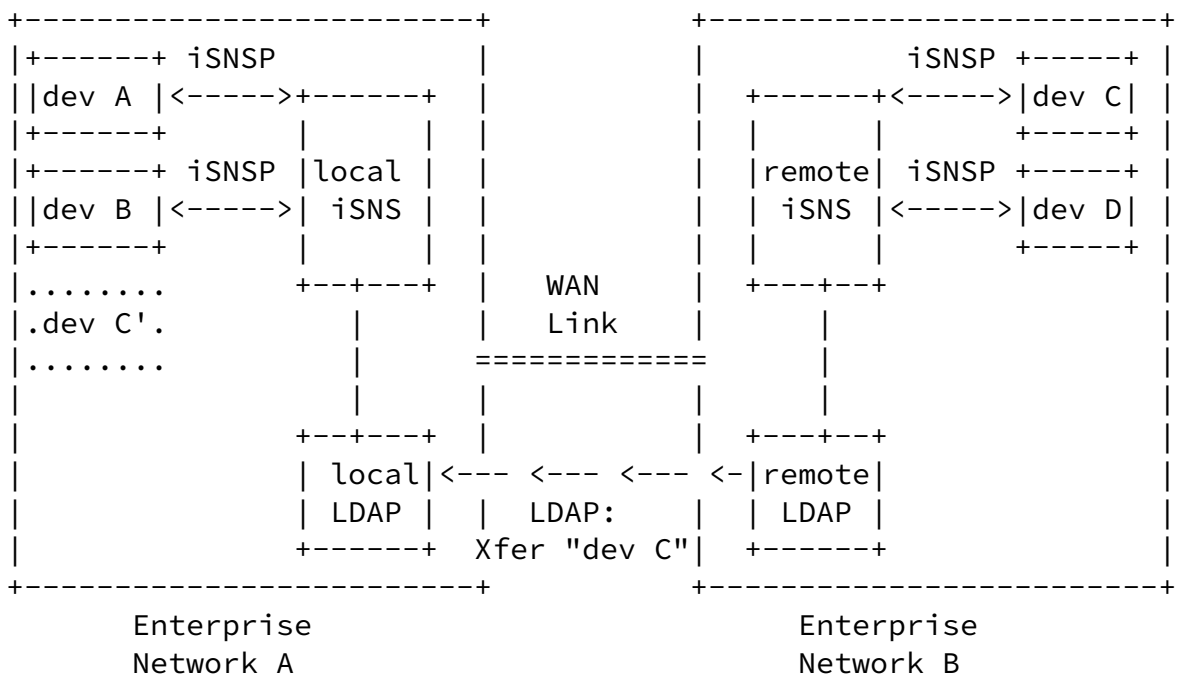
Transfer of iSNS database records between iSNS servers has important applications, including the following:

1) An independent organization needs to transfer storage information to a different organization. Each organization independently maintains its own iSNS infrastructure. To facilitate discovery of storage assets of the peer organization using IP, iSNS database records can be transfered between iSNS servers from each organization. This allows storage sessions to be established directly between devices residing in each organization's storage network infrastructure over a common IP network.

2) Multiple iSNS servers are desired for redundancy. Backup servers need to maintain copies of the primary server's dynamically changing database.

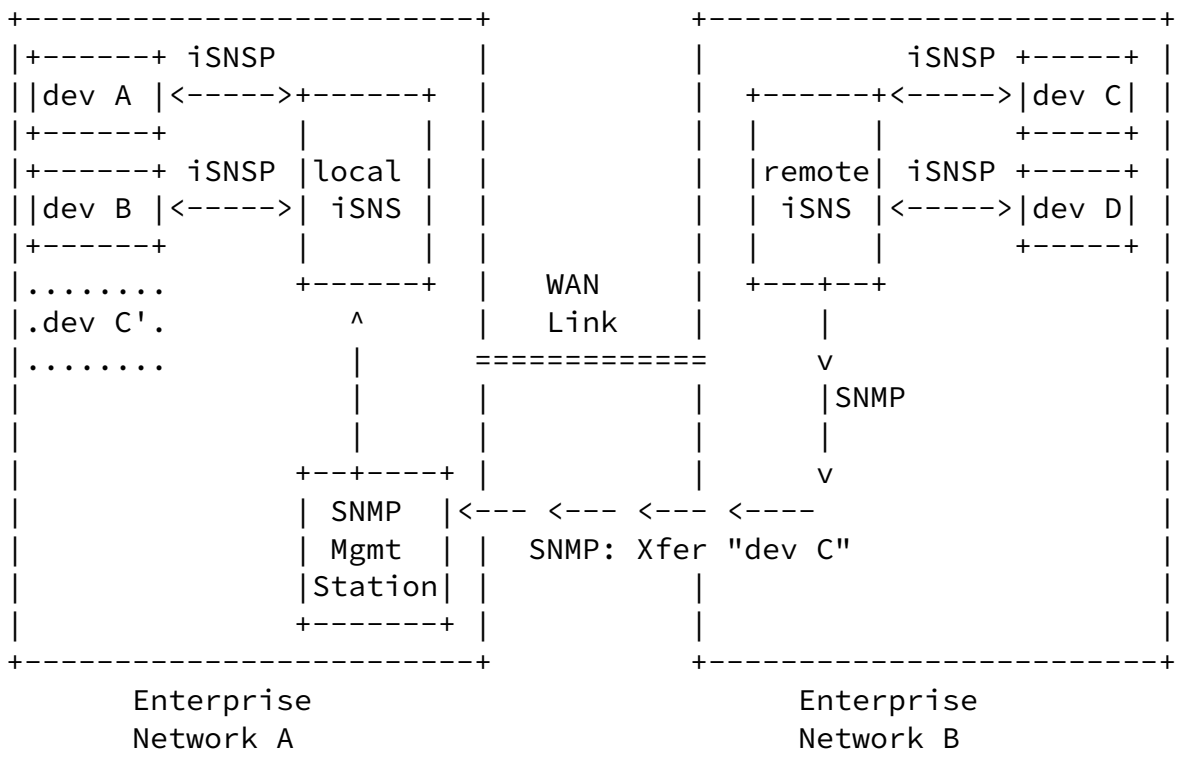
To support the above applications, information in an iSNS server can

be distributed to other iSNS servers either using the iSNS protocol, or through out-of-band mechanisms using non-iSNS protocols. The following examples illustrate possible methods to transfer data records between iSNS servers. In the first example, a back-end LDAP information base is used to support the iSNS server, and the data is transfered using the LDAP protcol. Once the record transfer of the remote device is completed, it becomes visible and accessible to local devices using the local iSNS server. This allows local devices to establish sessions with remote devices (provided firewall boundaries can be negotiated).



In the above diagram, two business partners wish to share storage "dev C". Using LDAP, the record for "dev C" can be transfered from Network B to Network A. Once accessible to the local iSNS in Network A, local devices A and B can now discover and connect to

"dev C".



The above diagram illustrates a second example of how iSNS records can be shared. This method uses an SNMP-based management station to manually download the desired record for "dev C", and then directly upload it to the local iSNS. Once the record is transfered to the local iSNS in Network A, "dev C" becomes visible and accessible (provided firewall boundaries can be negotiated) to other devices in Network A.

Other methods, including proprietary protocols, can be used to transfer device records between iSNS servers. Further discussion and explanation of these methodologies is beyond the scope of this document.

3.8 Backup iSNS Servers

Multiple iSNS servers can be used to provide redundancy in the event that the active iSNS server fails or is removed from the network. The methods described in section 3.7 above can be used to transfer name server records to backup iSNS servers. Each backup server maintains a redundant copy of the name server database found in the primary iSNS server, and can respond to iSNS protocol messages in

the same way as the active server. Each backup server SHOULD monitor the health and status of the active iSNS server, including checking to make sure its own database is synchronized with the active server's database. How each backup server accomplishes this is implementation-dependent, and may (or may not) include using the iSNS protocol. If the iSNS protocol is used, then the backup server MAY register itself in the active server's iSNS database as a control node, allowing it to receive state change notifications.

Generally, the administrator or some automated election process is responsible for initial and subsequent designation of the primary and each backup server.

A maximum of one backup iSNS server SHALL exist at any individual IP address.

In addition to proprietary vendor-specific ways of deploying multiple redundant iSNS servers, the iSNS heartbeat can also be used to coordinate designation and selection of primary and backup iSNS servers.

Each backup server should note its relative precedence in the active server's list of backup servers. If not already known, each backup server MAY learn its precedence from the iSNS heartbeat message, by noting the position of its IP address in the ordered list of backup server IP addresses. For example, if it is the first backup listed in the heartbeat message, then its backup precedence is 1. If it is the third backup server listed, then its backup precedence is 3.

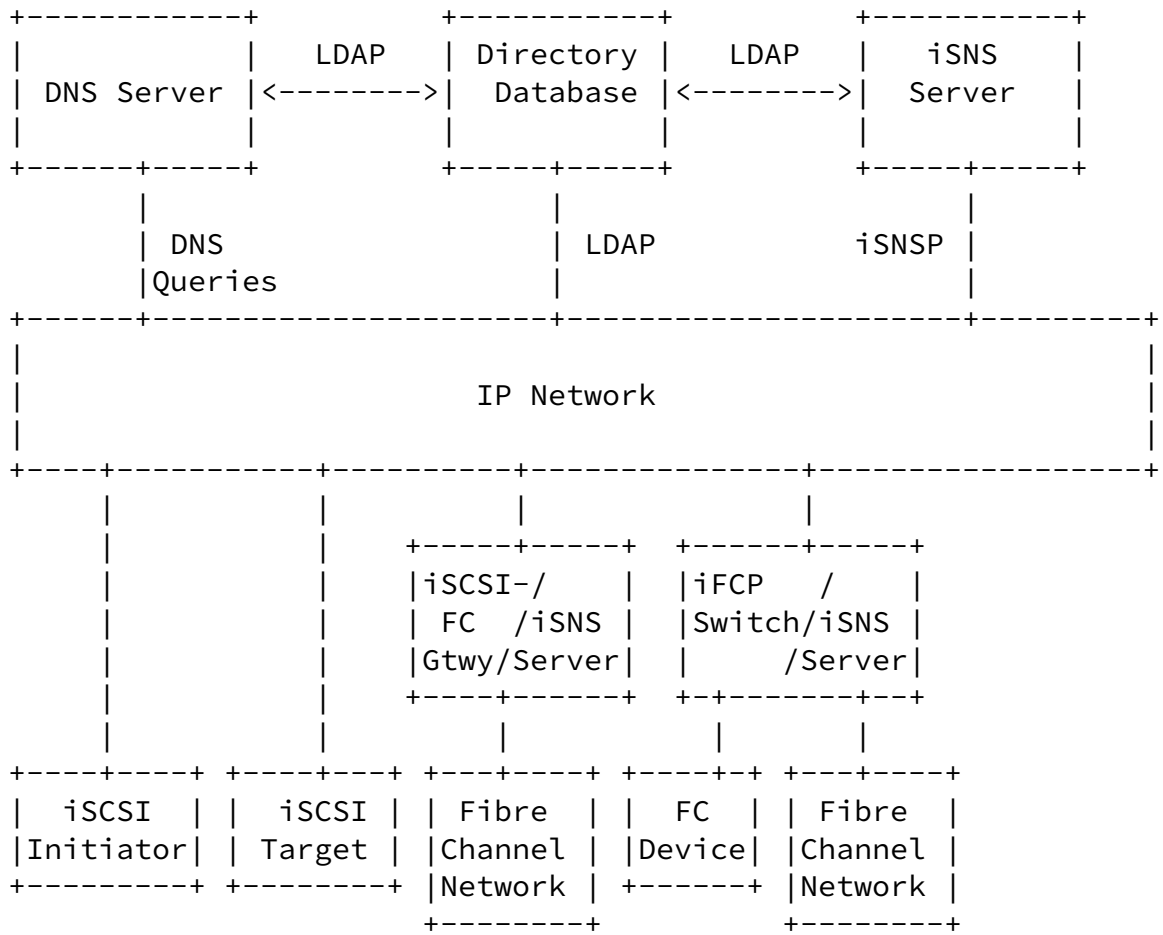
If a backup server establishes that it has lost connectivity to the active server and other backup servers of higher precedence, then it shall assume that it is the active server. The method of determining whether connectivity has been lost is implementation-specific. One possible approach is to assume that if the backup server does not receive iSNS hearbeat messages for a period of time, then connectivity to the active server has been lost. Alternately, the backup server may establish TCP connections to the active server and other backup servers, and loss of connectivity determined through non-response to periodic echo messages (using iSNSP, SNMP, or other protocols).

When a backup server becomes the active server, it shall assume all active server responsibilities, including (if used) transmission of the iSNS heartbeat message. If transmitting the iSNS heartbeat, the

backup server replaces the active Server IP Address and TCP/UDP Port entries with its own IP address and TCP/UDP Port, and begins incrementing the counter field from the last known value from the previously-active iSNS server. However, it MUST NOT change the original ordered list of backup server IP Address and TCP/UDP Port entries. If the primary backup server or other higher-precedence backup server returns, then the existing active server is responsible for updating the new active server's database before demoting itself to its original status as backup.

3.9 Deployment Architecture Diagram

The following diagram displays examples of where and how iSNS can be deployed, and of the various IP-based storage entities that it can support.



4. iSNS Object Model

iSNS provides the framework for the registration, discovery, and management of iSCSI devices and Fibre Channel-based devices (using iFCP). This architecture defines common objects that can be used to represent components referenced by each of these protocols.

This architecture framework provides elements needed to describe various storage device objects and attributes that may exist on an

IP storage network. Objects defined in this architecture framework include SAN, NETWORK ENTITY, PORTAL, STORAGE NODE, STORAGE DEVICE DISCOVERY DOMAIN, and DISCOVERY DOMAIN SET. Each of these objects are described in greater detail in the following sections.

4.1 NETWORK ENTITY Object

The NETWORK ENTITY object is a container of STORAGE NODE objects and PORTAL objects. It represents a logical device or gateway that is accessible from the IP network. All STORAGE NODEs and PORTALs contained within a single NETWORK ENTITY object operate in a coordinated manner.

Note that it is possible for a single physical device or gateway to be represented by more than one logical Network Entity in the iSNS database. For example, one of the storage nodes on a physical device may be accessible from only a subset of the network interfaces (i.e., portals) available on that device. In this case, a logical network entity (i.e., a "shadow entity") is created and used to contain the portals and storage nodes that can operate cooperatively. No object (portals, storage nodes, etc...) can be contained by more than one logical Network Entity.

4.2 PORTAL Object

The PORTAL object is an IP interface through which access to any STORAGE NODE within the NETWORK ENTITY can be obtained. A NETWORK ENTITY should have one or more PORTALs, each of which is usable by STORAGE NODEs contained in that NETWORK ENTITY to gain access to, or be accessible from, the IP network.

4.3 STORAGE NODE Object

The STORAGE NODE object is the logical endpoint of an iSCSI or iFCP session. In iFCP, the session endpoint is represented by the World Wide Port Name (WWPN). In iSCSI, the session endpoint is represented by the iSCSI Name of the device.

4.4 FC DEVICE Object (iFCP Only)

The FC DEVICE represents the Fibre Channel end node. Although mostly unused in support of the iFCP storage connection, this object contains information that may be useful in the management of the Fibre Channel device.

4.5 DISCOVERY DOMAIN Object

DISCOVERY DOMAINS (DD) are a security and management mechanism used to partition storage resources. Discovery Domains limit the discovery process to the administrator-configured subset of relevant storage devices, preventing initiators from inappropriately attempting login to devices that they shouldnÆt have access to. When queried, the iSNS server will provide information only for storage entities that share at least one common DD. Initiators will

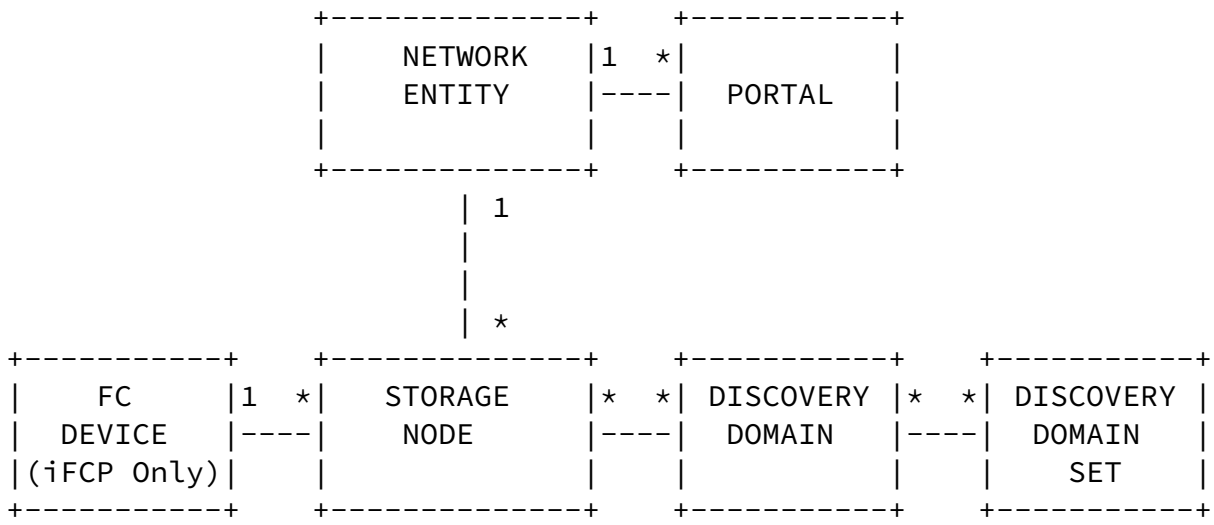
not be able to "see" devices with which they do not have at least one common DD.

4.6 DISCOVERY DOMAIN SET Object

The DISCOVERY DOMAIN SET (DDS) is a container object for DDÆs. DDSÆs may contain one or more DDÆs. Similarly, each DD can be a member of one or more DDSÆs. DDSÆs are a mechanism to store coordinated sets of DD mappings in the iSNS.

4.7 iSNS Database Model

The following shows the the various objects described above and their relationship to each other.



* represents 0 to many possible relationships

5. iSNS Implementation Requirements

iSNS can be implemented with features to support iSCSI and/or iFCP. Implementation of support for either or both of these protocols is OPTIONAL. IF iSNS is implemented to support a particular protocol, then a minimum set of attributes and iSNSP commands is REQUIRED for support of that protocol. This section details specific requirements for support of each of these IP storage protocols. Implementation requirements for security are described in section 1.1.

5.1 iSCSI Requirements

Use of iSNS in support of iSCSI is OPTIONAL. iSCSI devices MAY be manually configured with the iSCSI Name and IP address of peer devices, without the aid or intervention of iSNS. iSCSI devices also MAY use SLP [RFC 2608] to discover peer iSCSI devices. However, for scaling a storage network to a larger number of iSCSI devices, use of iSNS is RECOMMENDED.

5.1.1 Required Attributes for Support of iSCSI

The following attributes are available to support iSCSI. Attributes indicated in the REQUIRED TO IMPLEMENT column MUST be supported by an iSNS server used to support iSCSI. Attributes indicated in the REQUIRED TO USE column MUST be supported by an iSCSI device that elects to use the iSNS.

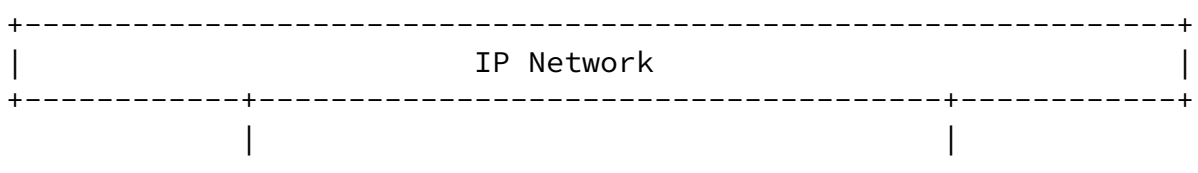
Object	Attribute	REQUIRED to Implement	REQUIRED to Use
NETWORK ENTITY	Entity Identifier	*	*
	Entity Protocol	*	*
	Management IP Address		
	Timestamp	*	
	Protocol Version Range	*	
	Registration Period	*	
	Entity Index	*	
	Entity IKE Phase-1 Proposal		
	Entity Certificate		
PORTAL	IP Address	*	*
	TCP/UDP Port	*	*
	Portal Symbolic Name	*	

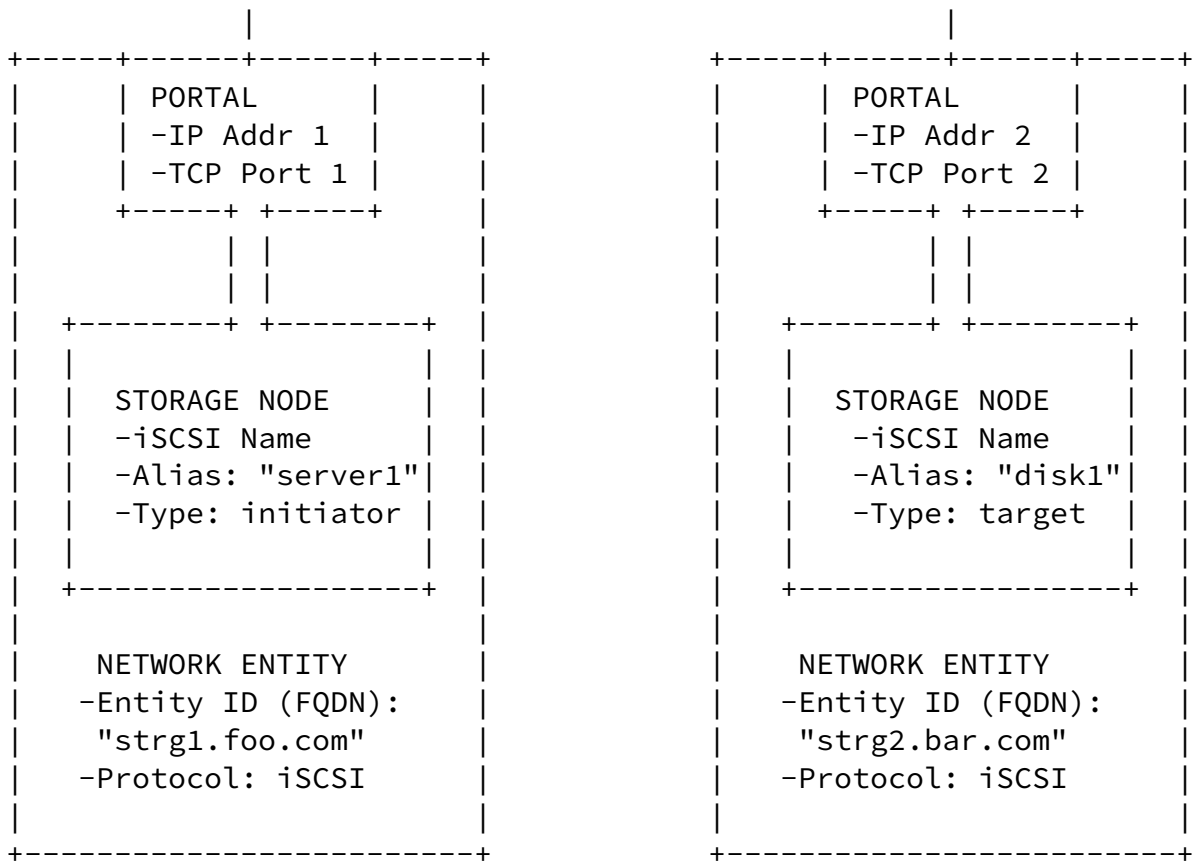
	ESI Interval	*	
	ESI Port	*	
	Portal Group	*	
	Portal Index	*	
	SCN Port	*	
	Portal Security Bitmap	*	
	Portal IKE Phase-1 Proposal		
	Portal IKE Phase-2 Proposal		
	Portal Certificate		
STORAGE NODE	iSCSI Name	*	*
	iSCSI Node Type	*	*
	Alias	*	
	iSCSI SCN Bitmap	*	
	iSCSI Node Index	*	
	EUI64 Token		
	iSCSI Node Certificate		
DISCOVERY DOMAIN	DD_ID	*	*
	DD_Symbolic Name	*	
	DD iSCSI Node Index	*	
	DD iSCSI Node Member	*	
	DD Features	*	
DISCOVERY DOMAIN SET	DDS Identifier	*	
	DDS Symbolic Name	*	
	Status	*	

All iSCSI user-specified and vendor-specified attributes are optional to implement and use.

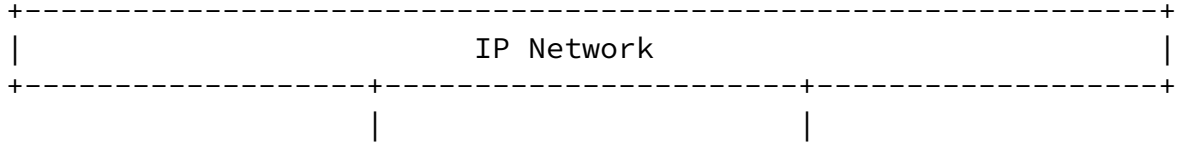
5.1.2 Example iSCSI Object Model Diagrams

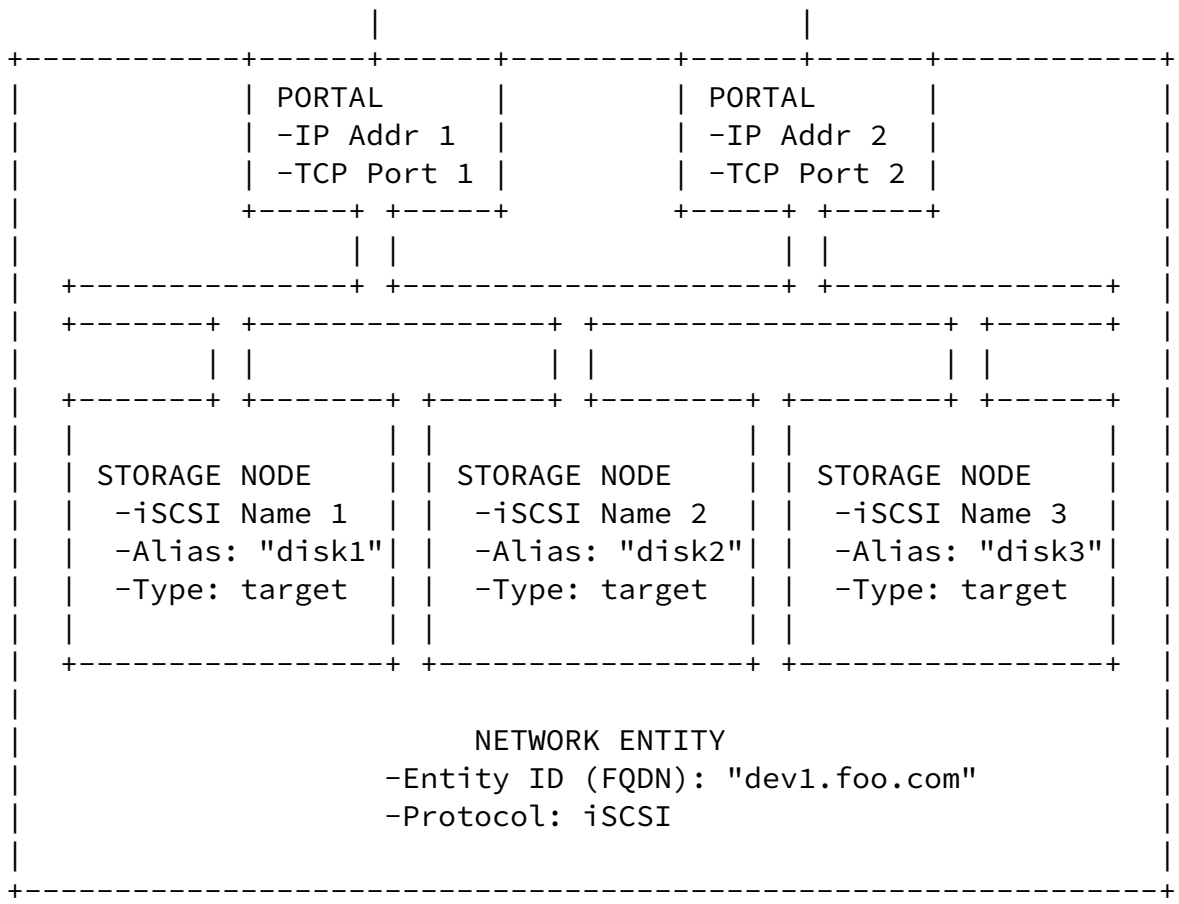
The following diagram models how a simple iSCSI-based initiator and target is represented using database objects stored in the iSNS. In this implementation, each target and initiator is attached to a single PORTAL.





The object model can be expanded to describe more complex devices, such as an iSCSI device with more than one storage controller, each controller accessible through any of multiple PORTAL interfaces. The storage controllers on this device which can be accessed through alternate PORTAL interfaces, if any original interface should fail. The following diagram describes such a device:





5.1.3 Required Commands and Response Messages for Support of iSCSI

The following are iSNSP messages and responses are available in support of iSCSI. Messages indicated in the REQUIRED TO IMPLEMENT column MUST be implemented in iSNS servers used for iSCSI devices. Messages indicated in the REQUIRED TO USE column must be implemented in iSCSI devices that elect to use the iSNS.

Message Description	Abbreviation	Func ID	REQUIRED TO: Implement	Use
Register Dev Attr Req	RegDevAttr	0x0001	*	*
Dev Attr Query Request	DevAttrQry	0x0002	*	*
Dev Get Next Request	DevGetNext	0x0003	*	
Deregister Dev Request	DeregDev	0x0004	*	*
SCN Register Request	SCNReg	0x0005	*	
SCN Deregister Request	SCNDereg	0x0006	*	
SCN Event	SCNEvent	0x0007	*	
State Change Notification	SCN	0x0008	*	
DD Register	DDReg	0x0009	*	*
DD Deregister	DDDereg	0x000A	*	*
DDS Register	DDSReg	0x000B	*	*
DDS Deregister	DDSDereg	0x000C	*	*
Entity Status Inquiry	ESI	0x000D	*	

Name Service Heartbeat	Heartbeat	0x000E
NOT USED		0x000F-0x0013
RESERVED		0x0014-0x00FF
Vendor Specific		0x0100-0x01FF
RESERVED		0x0200-0x8000

The following are iSNSP response messages used in support of iSCSI:

Response Message Desc	Abbreviation	Func_ID	REQUIRED TO: Implement	Use
Register Dev Attr Rsp	RegDevRsp	0x8001	*	*
Dev Attr Query Rsp	DevAttrQryRsp	0x8002	*	*
Dev Get Next Rsp	DevGetNextRsp	0x8003	*	
Deregister Dev Rsp	DeregDevRsp	0x8004	*	*
SCN Register Rsp	SCNRegRsp	0x8005	*	
SCN Deregister Rsp	SCNDeregRsp	0x8006	*	
SCN Event Rsp	SCNEventRsp	0x8007	*	
SCN Response	SCNRsp	0x8008	*	
DD Register Rsp	DDRegRsp	0x8009	*	*
DD Deregister Rsp	DDDeregRsp	0x800A	*	*
DDS Register Rsp	DDSRegRsp	0x800B	*	*
DDS Deregister Rsp	DDSDeregRsp	0x800C	*	*
Entity Stat Inquiry Rsp	ESIRsp	0x800D	*	
NOT USED		0x800E-0x8013		
RESERVED		0x8014-0x80FF		
Vendor Specific		0x8100-0x81FF		
RESERVED		0x8200-0xFFFF		

5.2 iFCP Requirements

In iFCP, use of iSNS is REQUIRED. No alternatives exist for support of iFCP Naming & Discovery functions. iSNS is integral to the operation of iFCP, in order to allow iFCP gateways to execute Fibre Channel S_ID and D_ID address mappings to remote gateways.

5.2.1 Required Attributes for Support of iFCP

The following table displays attributes that are used by iSNS to support iFCP. Attributes indicated in the REQUIRED TO IMPLEMENT column MUST be supported by the iSNS server that supports iFCP. Attributes indicated in the REQUIRED TO USE column MUST be supported

by iFCP gateways.

Object	Attribute	REQUIRED to Implement	REQUIRED to Use
NETWORK ENTITY	Entity Identifier	*	*
	Entity Protocol	*	*
	Management IP Address		
	Timestamp	*	
	Protocol Version Range	*	

	Registration period		
	Entity Index		
	Entity IKE Phase-1 Proposal		
	Entity Certificate		
PORTAL	IP Address	*	*
	TCP/UDP Port	*	*
	Symbolic Name	*	
	ESI Interval	*	
	ESI Port	*	
	Portal IKE Phase-1 Proposal		
	Portal IKE Phase-2 Proposal		
	Portal Certificate		
	Security Bitmap	*	
STORAGE NODE	Port Name (WWPN)	*	*
	Port_ID	*	*
	Port Type	*	*
	Port Symbolic Name	*	
	Fabric Port Name (FWWN)	*	
	Hard Address	*	
	Port IP Address	*	
	Class of Service	*	
	FC FC-4 Types	*	
	FC FC-4 Descriptors	*	
	FC FC-4 Features	*	
	SCN Bitmap	*	
	Port Certificate		
FC DEVICE	Node Name (WWNN)	*	*
	Node Symbolic Name	*	
	Node IP Address	*	
	Node IPA	*	
	Node Certificate		

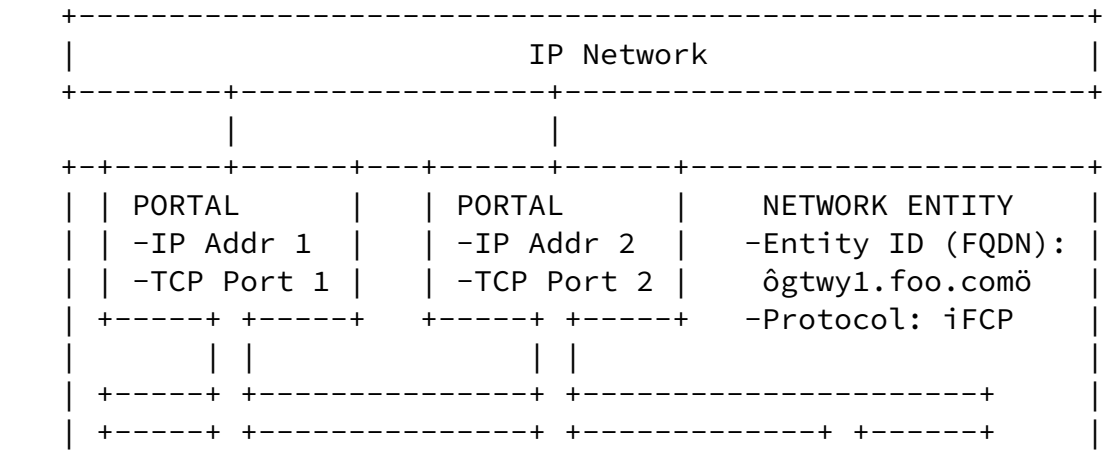
	Proxy iSCSI Name		
DISCOVERY DOMAIN	DD_ID	*	*
	DD_Symbolic Name	*	
	DD iFCP Member (WWPN)	*	
DISCOVERY DOMAIN SET	DDS Identifier	*	
	DDS Symbolic Name	*	
	DDS Status	*	
OTHER	Switch Name		
	Preferred_ID		
	Assigned_ID		
	Space Identifier		

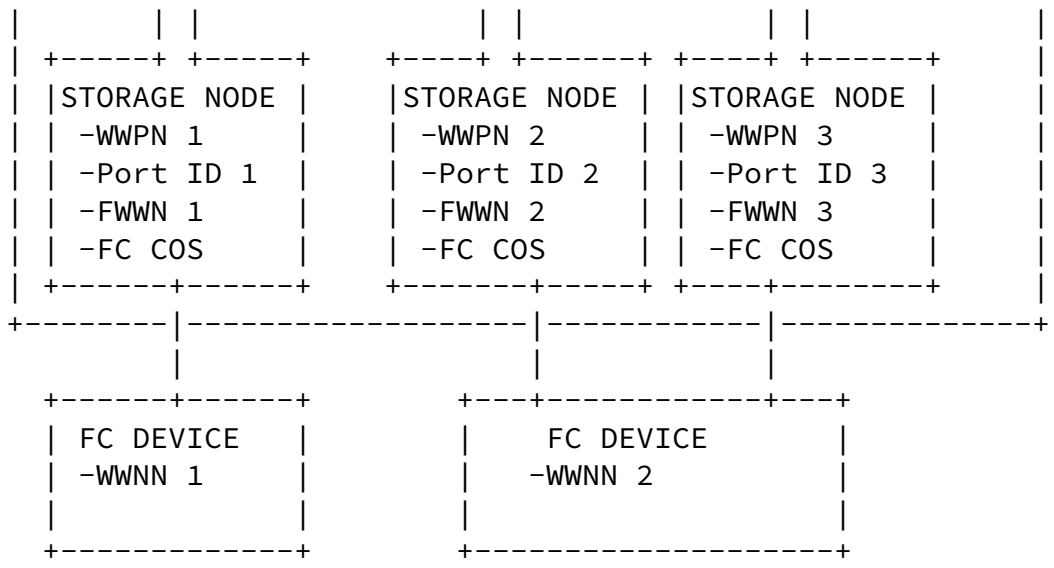
5.2.2 Example iFCP Object Model Diagram

The iFCP protocol allows native Fibre Channel devices, or Fibre Channel fabrics connected to an iFCP gateway, to be directly internetworked using IP.

When supporting iFCP, the iSNS stores Fibre Channel device attributes, iFCP gateway attributes, and Fibre Channel fabric switch attributes that might also be stored in an FC name server.

The following diagram shows a representation of a gateway supporting multiple Fibre Channel devices behind it. The two PORTAL objects represent IP interfaces on the iFCP gateway that can be used to access any of the three STORAGE NODE objects behind it. Note that the FC DEVICE object is not contained in the NETWORK ENTITY object. However, each FC DEVICE has a relationship to one or more STORAGE NODE objects.





5.2.3 Required Commands and Response Messages for Support of iFCP

The iSNSP messages and responses displayed in the following tables are available to support iFCP gateways. Messages indicated in the REQUIRED TO IMPLEMENT column MUST be supported by the iSNS server used by iFCP gateways. Messages indicated in the REQUIRED TO USE column MUST be supported by the iFCP gateways themselves.

Message Description	Abbreviation	Func ID	REQUIRED TO: Implement	Use
Register Dev Attr Req	RegDevAttr	0x0001	*	*
Dev Attr Query Request	DevAttrQry	0x0002	*	*
Dev Get Next Request	DevGetNext	0x0003	*	
Deregister Dev Request	DeregDev	0x0004	*	*
SCN Register Request	SCNReg	0x0005	*	
SCN Deregister Request	SCNDereg	0x0006	*	
SCN Event	SCNEvent	0x0007	*	
State Change Notification	SCN	0x0008	*	
DD Register	DDReg	0x0009	*	*
DD Deregister	DDDereg	0x000A	*	*
DDS Register	DDSReg	0x000B	*	*

DDS Deregister	DDSDereg	0x000C	*	*
Entity Status Inquiry	ESI	0x000D	*	
Name Service Heartbeat	Heartbeat	0x000E	*	
Reserved	Reserved	0x000F-0x0010		
Request Switch ID	RqstSwId	0x0011		
Release Switch ID	RlseSwId	0x0012		
Get Switch IDs	GetSwIds	0x0013		
RESERVED		0x0014-0x00FF		
Vendor Specific		0x0100-0x01FF		
RESERVED		0x0200-0x8000		

The following are iSNSP response messages in support of iFCP:

			REQUIRED TO:	
Response Message Desc	Abbreviation	Func_ID	Implement	Use
Register Dev Attr Rsp	RegDevRsp	0x8001	*	*

Dev Attr Query Resp	DevAttrQryRsp	0x8002	*	*
Dev Get Next Resp	DevGetNextRsp	0x8003	*	
Deregister Dev Resp	DeregDevRsp	0x8004	*	*
SCN Register Resp	SCNRegRsp	0x8005	*	
SCN Deregister Resp	SCNDeregRsp	0x8006	*	
SCN Event Resp	SCNEventRsp	0x8007	*	
SCN Response	SCNRsp	0x8008	*	
DD Register Rsp	DDRegRsp	0x8009	*	*
DD Deregister Rsp	DDDeregRsp	0x800A	*	*
DDS Register Rsp	DDSRegRsp	0x800B	*	*
DDS Deregister Rsp	DDSDeregRsp	0x800C	*	*
Entity Stat Inquiry Resp	ESIRsp	0x800D	*	
NOT USED		0x800E		
RESERVED		0x800F-0x8010		
Request Switch ID Resp	RqstSwIdRsp	0x8011		
Release Switch ID Resp	RlseSwIdRsp	0x8012		
Get Switch IDs	GetSwIdRsp	0x0013		
RESERVED		0x8014-0x80FF		
Vendor Specific		0x8100-0x81FF		
RESERVED		0x8200-0xFFFF		

5.3 Attribute Descriptions for Discovery Domain Registration

Discovery Domains are logical groupings of initiators and targets that are used to limit the login process to the appropriate subset of devices registered in the iSNS.

Support for Discovery Domains is required for all protocols. The iSNS attributes for Discovery Domain and Discovery Domain Set registration are shown in the following:

```
DISCOVERY DOMAIN SET
   |
   - DD Set_ID
   - DD Set_Symbolic Name
   - DD Set Enabled/Disabled

DD SET_MEMBER
   |
   - DD Set_ID
   - DD_ID

DISCOVERY DOMAIN
   |
   - DD_ID
   - DD_Symbolic Name
   - DD Features

DD_MEMBER
   |
   - DD_ID
   - iSCSI Node Index, iSCSI Name, or WWPN
```

Membership in a Discovery Domain is established by registering one of the following attributes in that DD:

- iSCSI Name or iSCSI Node Index : this places the iSCSI node in the Discovery Domain
- WWPN : this places the FC Port in the Discovery Domain

5.4 Use of TCP For iSNS Communication

TCP can be used for any or all iSNS communication. The iSNS server MUST accept TCP connections for client registrations. The well-known TCP port used by the iSNS server to receive TCP messages used is 3205. The iSNS client MAY use one or more TCP connections to register attributes and communicate with the server.

To receive ESI monitoring using TCP, the client registers the Portal ESI Interval using the TCP connection that will be used to receive ESI inquiry messages. TCP-based ESI monitoring requires that an open TCP connection be maintained by the iSNS server to every iSNS

client registered to receive monitoring. If the TCP connection supporting ESI monitoring is terminated, then the client must reregister for ESI messages on a new TCP connection in order to continue to receive ESI monitoring.

To receive SCN notifications using TCP, the client registers the iSCSI/iFCP SCN Bitmap by originating the SCNReg message from the TCP port that will be used to receive SCN notification messages. While the TCP connection does not necessarily need to be open, the client

MUST accept SCN messages using the same TCP port used to register for SCN notification.

It is possible for an iSNS client to use the same TCP connection for SCN, ESI, and iSNS queries. Alternately, separate connections may be used.

5.5 Use of UDP For iSNS Communication

The iSNS server MAY accept UDP messages for client registrations. The iSNS server MUST accept registrations from clients requesting UDP-based ESI and SCN messages. The well-known UDP port used to receive UDP messages is 3205.

To receive UDP-based ESI monitoring messages, the client registers the Portal ESI/SCN UDP Port to be used for communication of ESI messages from the server to the client.

To receive UDP-based SCN notifications messages, the client registers at least one Portal ESI/SCN UDP port to be used for communication of SCN messages from the server to the client. If an entity has multiple Portals with registered ESI/SCN UDP Ports, then ESI and SCN messages SHALL be delivered to each Portal registered to receive such messages.

6. iSNS Message Attributes

The following are attributes stored in the iSNS server, which can be retrieved using iSNS queries. Unless otherwise indicated, these attributes are supplied by iSNS clients using iSNS registration messages.

6.1 iSNS Attribute Summary

The following table lists all iSNSP message attributes for device registration and queries:

T	Entity Attributes	Length	Tag	Reg Key	Query Key
	Delimiter	0-256	0	N/A	N/A
^	Entity Identifier (EID)	0-256	1	1|@	@|1|2|16,17|32|64
&	Entity Protocol	4	2	1	@|1|2|16,17|32|64
	Mgt IP Address	16	3	1	@|1|2|16,17|32|64
=	Timestamp	8	4	1	@|1|2|16,17|32|64
	Protocol Version Range	4	5	1	@|1|2|16,17|32|64
	Registration Period	4	6	1	@|1|2|16,17|32|64
	Entity Index	4	7	1	@|1|2|16,17|32|64
	Entity ISAKMP Phase-1	var	11	1	@|1|2|16,17|32|64
*	Entity Certificate	var	12	1	@|1|2|16,17|32|64
#	Portal IP-Address	16	16	1	@|1|16,17|32|64
$	Portal TCP/UDP Port	4	17	1	@|1|16,17|32|64
	Portal Symbolic Name	0-256	18	16,17	@|1|16,17|32|64
	ESI Interval	4	19	16,17	@|1|16,17|32|64
	ESI Port	4	20	16,17	@|1|16,17|32|64
	Portal Group	4	21	16,17	@|1|16,17|32|64
	Portal Index	4	22	16,17	@|1|16,17|32|64
	SCN Port	4	23	16,17	@|1|16,17|32|64
	Portal Security Bitmap	4	27	16,17	@|1|16,17|32|64
*	Portal ISAKMP Phase-1	var	28	16,17	@|1|16,17|32|64
*	Portal ISAKMP Phase-2	var	29	16,17	@|1|16,17|32|64
*	Portal Certificate	var	31	16,17	@|1|16,17|32|64

```
  # iSCSI Name               4-256   32   1%      @|1|16,17|32|33
  & iSCSI Node Type          4       33   32      @|1|16,17|32
    Alias                    0-256   34   32      @|1|16,17|32
    iSCSI SCN Bitmap         4       35   32      @|1|16,17|32
    iSCSI Node Index         4       36   32      @|1|16,17|32
    EUI64 Token              8       37   32      @|1|16,17|32
  * iSCSI Node Certificate   var     43   32      @|1|16,17|32
  # Port Name WWPN           8       64   1%     @|1|16,17|64|66|96|128
    Port ID                  4       65   64      @|1|16,17|64
    Port Type                4       66   64      @|1|16,17|64
    Symbolic Port Name       0-256   67   64      @|1|16,17|64
    Fabric Port Name         8       68   64      @|1|16,17|64
    Hard Address             4       69   64      @|1|16,17|64
    Port IP-Address          16      70   64      @|1|16,17|64
    Class of Service         4       71   64      @|1|16,17|64
    FC-4 Types               32      72   64      @|1|16,17|64
    FC-4 Descriptor          0-256   73   64      @|1|16,17|64
    FC-4 Features            128     74   64      @|1|16,17|64
    iFCP SCN bitmap          4       75   64      @|1|16,17|64
    iFCP Port Type           4       76   64      @|1|16,17|64
  * Port Certificate         var     83   64      @|1|16,17|64
    FC-4 Type Code           4       95   Query Key only
  # Node Name WWNN           8       96   @       @|64|96
    Symbolic Node Name       0-256   97   96      @|64|96
    Node IP-Address          16      98   96      @|64|96
    Node IPA                 8       99   96      @|64|96
  * Node Certificate         var    100   96      @|64|96
    Proxy iSCSI Name         0-256  101   96      @|64|96
    Switch Name              8      128   128|@
    Preferred ID             4      129   128     @|128
```



```
    Assigned ID              4      130   128     @|128
    Space_Identifier         0-256  131   128     @|128
    RESERVED for iSNS server-specific    tags in range 132-255
    Company OUI              4      256
  * Vendor-Spec iSNS Srvr    var     -    tags in range 257-383
  * Vendor-Spec Entity       var     -    tags in range 384-511
  * Vendor-Spec Portal       var     -    tags in range 512-639
  * Vendor-Spec iSCSI Node   var     -    tags in range 640-767
  * Vendor-Spec Port Name    var     -    tags in range 768-895
  * Vendor-Spec Node Name    var     -    tags in range 896-1023
  * Vendor-Specific DD       var     -    tags in range 1024-1279
  * Other Vendor-Specific    var     -    tags in the range 1280-2047
    RESERVED                      2048-6000
    Standards-based Extensions    6001-12000
```

RESERVED	12001-65535
RESERVED	All Others

The following is a description of the columns used in the above table:

Attribute Type (T)

: Required key for object registration.
^ : Required key for object registration, unique value is assigned by the iSNS if value not provided during initial registration.
$: Required as part of the key, and the canonical value is used if one is not registered.
& : Attribute required during initial registration that is not a key.
* : Optional to implement in the iSNS.
= : Cannot be used as a query key or be explicitly registered. The value is assigned by the iSNS at registration / update.
@ : if no key is present then a new entry is created, or all entries of the operating attributes are returned.
| : used to separate the different sets of possible keys in the table.
% : If an iSCSI Name or Port Name WWPN is registered without an EID key, then an Entity will be created and an EID assigned. The assigned EID will be returned in the response as an Operating attribute.

Length - indicates the attribute length. Variable-length identifiers are NULL character terminated, which is included in the length.

Tag - the integer tag value used to identify the attribute. All undefined tag values are reserved.

Value û a description of the data.

Implementation Notes:

A well-formed registration contains the key of the object to register, or no key if it is to be generated by the iSNS server. If an attribute is present as a key, then it cannot also be an operating attribute.

The registration response will contain the key for each object registered, including any key values that were assigned by the iSNS as part of the registration. For example, if an entity, two portals and one Port Name was registered, then the response message key attributes section would contain the keys for each. The key attributes, returned in the response, may be in a different order they appeared in the registration.

iSNS attributes are defined below.

6.2 Entity Identifier-Keyed Attributes

The following attributes are stored in the iSNS using the Entity Identifier attribute as the key.

6.2.1 Entity Identifier (EID)

The Entity Identifier is a variable length identifier that uniquely identifies each network entity registered in the iSNS. The attribute length varies from 4 to 256 bytes, and is a unique value within the iSNS.

If the iSNS client does not provide an EID during registration the iSNS shall generate one that is unique within the iSNS. If an EID is to be generated, then the EID attribute value in the registration message shall be empty (0 length). The generated EID shall be returned in the registration response.

In environments where iSNS is integrated with a DNS infrastructure, the Entity Identifier may be used to store the Fully Qualified Domain Name (FQDN) of the iSCSI or iFCP device.

If FQDN's are not used, the iSNS server can be used to generate EIDs. By convention, EIDs generated by the iSNS server begin with the string ôiSNS:ö. iSNS clients SHOULD NOT generate and register EIDs beginning with the string "iSNS:".

6.2.2 Entity Protocol

Entity Protocol is a required 4-byte attribute that indicates the protocol of registered network entity. The valid types are defined as below:

Type Value	Entity Type
0	Protocol Neutral
1	iSCSI
2	iFCP
All Others	RESERVED

'Protocol neutral' is the standard registration for 'control nodes'.

6.2.3 Management IP Address

This field contains the IP Address used to manage the entity. The Management IP Address is a 16-byte field that may contain either a 32-bit IPv4 or 128-bit IPv6 address. When this field contains an IPv4 value, the most significant 12 bytes are set to 0x00. If the network entity is capable of being managed and this field is not set, then in-band management is assumed.

6.2.4 Entity Registration Timestamp

This field indicates the time the entity registration occurred, an associated object was updated, or the time of the most recent response from a message to the entity was received, whichever is later. The time format is, in seconds, the update period since the standard base time of 00:00:00 GMT on January 1, 1970. It cannot be used as a query key or be explicitly registered.

6.2.5 Protocol Version Range

This field contains the minimum and maximum version of the protocol supported by the entity. The most significant two bytes contain the maximum version supported, and the least significant two bytes contain the minimum version supported. If a range is not registered then the entity is assumed to support all versions of the protocol. If the entity is protocol neutral, then this field SHALL be set to 0.

6.2.6 Registration Period

This field indicates the maximum period, in seconds, that the entity registration will be maintained by the server without receipt of an iSNS message from the client. If the Registration Period is set to 0, then the Entity SHALL NOT be deregistered due to no contact with the iSNS client.

If ESI messages are not requested by an entity and the Registration Period is not set to 0, then the entity registration SHALL be removed if an iSNS Protocol message is not received from the iSNS client before the registration period has expired. Receipt of any iSNS Protocol message from the iSNS client automatically refreshes the Entity Registration Period and Entity Registration Timestamp. To prevent a registration from expiring, the iSNS client should send an iSNS Protocol message to the iSNS server at intervals shorter than the registration period. Such a message can be as simple as a query

for one of its own attributes, using its associated iSCSI Name or Port Name as the SOURCE attribute.

For a multi-node entity, receipt of an iSNS message from any node of that entity is sufficient to refresh the registration for all nodes of the entity.

Byte 0 and 1 represents the entity registration period, in seconds. Byte 2 and 3 are reserved.

If ESI support is requested as part of a portal registration, the ESI Response message received by the server SHALL act as an alternative to a refresh of the entity registration.

6.2.7 Entity Index

The Entity Index is a 4-byte integer value that uniquely identifies each network entity registered in the iSNS. The Entity Index is assigned by the iSNS server during the initial registration of an Entity. The value MAY BE assigned using a monotonically increasing process.

The Entity Index can be used to represent a registered entity in situations where the Entity Identifier is too long to be used. An example of this is when SNMP tables are used to access the contents of the iSNS server. In this case, the Entity Index may be used as the table index.

6.2.8 Entity ISAKMP Phase-1 Proposals

This field contains the IKE Phase-1 proposal listing in decreasing order of preference of the protection suites acceptable to protect all IKE Phase-2 messages sent and received by the Entity. This includes Phase-2 SA's from the iSNS client to the iSNS server as

well as to peer iFCP and/or iSCSI devices. This attribute contains the SA payload, proposal payload(s), and transform payload(s) in the ISAKMP format defined in [RFC2408].

This field should be used if the implementer wishes to define a single phase-1 SA security configuration used to protect all phase-2 IKE traffic. If the implementer desires to have a different phase-1 SA security configuration to protect each Portal interface, then the Portal Phase-1 Proposal (section 6.3.10) should be used.

6.2.9 Entity Certificate

This attribute contains one or more X.509 certificate that are bound to the NETWORK ENTITY of the iSNS client. This certificate is uploaded and registered to the iSNS by clients wishing to allow other clients to authenticate themselves and access the services offered by that NETWORK ENTITY.Portal-Keyed Attributes

6.3 Portal-Keyed Attributes

The following portal attributes are registered in the iSNS using the combined Portal IP-Address and Portal TCP/UDP Port as the key. Each portal is associated with one Entity Identifier object key.

6.3.1 Portal IP-Address

The IP address of the PORTAL through which a STORAGE NODE can transmit and receive storage data. When an IPv4 value is contained in this field, the most significant 12 bytes are set to 0x00. The Portal IP Address along with the Portal TCP/UDP Port number uniquely identifies a Portal.

6.3.2 Portal TCP/UDP Port

The TCP/UDP port of the PORTAL through which a STORAGE NODE can transmit and receive storage data. Bit 0 to 15 represents the TCP/UDP port number. Bit 16 represents the port type. If bit 16 is set, then the port type is UDP. Otherwise it is TCP. Bits 17 to 31 are reserved.

If the field value is 0, then the port number is the implied canonical port number and type of the protocol indicated by the

associated Entity Type.

The Portal IP-Address along with the Portal TCP/UDP Port number uniquely identifies a Portal.

6.3.3 Portal Symbolic Name

This is a variable-length text-based value from 0 to 256 bytes. The text field contains user-readable UTF-8 text, and is terminated with at least one NULL character. The Portal Symbolic Name is a user-readable description of the Portal entry in the iSNS.

6.3.4 Entity Status Inquiry Interval

This field indicates the requested time, in seconds, between Entity Status Inquiry (ESI) messages sent from the iSNS to this entity portal. ESI messages can be used to verify that a Portal registration continues to be valid. To request monitoring by the iSNS, an iSNS client registers a non-zero value for this portal attribute using a RegDevAttr message. The client must also register an ESI Port on at least one of its Portals to receive the ESI monitoring.

If the iSNS server does not receive an expected response to an ESI message, it shall attempt at least three re-transmissions of the ESI message. All re-transmissions MUST be sent before twice the ESI Interval period has passed since the last ESI response was received from the client. If no response is received from any of the ESI messages, then the Portal SHALL be deregistered. If TCP was used to transport the ESI messages, then that TCP connection SHALL be

closed. Note that only Portals that have registered a value in their ESI Port field can be deregistered in this way.

If all Portals associated with an entity that have registered for ESI messages are deregistered due to non-response, and no registrations have been received from the client for at least two ESI Interval periods, then the entity and all associated objects (including storage nodes) SHALL be deregistered.

If the iSNS server is unable to support ESI messages or the ESI Interval requested, it SHALL reject the ESI request by returning an "ESI Not Available" error code. The rejection might occur in situations where the resulting frequency of ESI messages being issued to clients would pass an implementation-specific threshold.

If at any time an iSNS client that is registered for ESI messages has not received an ESI message to any of its portals as expected, then the client MAY attempt to query the iSNS server using a DevAttrQry message using its Entity_ID as the key. If the query result is the error "no such entry", then the client SHALL close all remaining TCP connections to the iSNS server and assume that it is no longer registered in the iSNS database. Such a client MAY attempt re-registration.

6.3.5 ESI Port

This field contains the TCP or UDP port of the iSNS client used for ESI monitoring by the iSNS server. Bit 0 to 15 represents the port number. If bit 16 is set, then the port type is UDP. Otherwise, the port is TCP. Bits 17 to 31 are reserved.

The iSNS server SHALL return an error if an Entity is registered for ESI monitoring and none of the portals of that Entity has an entry for the ESI Port field. If multiple Portals have a registered ESI port, then the ESI message may be delivered to any of the indicated portals.

6.3.6 Portal Group

This field is used to group portals into aggregation groups. All entity portals that belong to the same Portal Group in a Network Entity can provide connections to the same STORAGE NODE. The value chosen for the Portal Group need only be unique within a given Network Entity. The same Portal Group value can represent unassociated Portal Groups in other Network Entities. This allows multiple sessions to be established to a node through multiple portals. The least significant two bytes contain the integer Portal Group value for the Portal. The most significant two bytes are reserved.

6.3.7 Portal Index

The Portal Index is a 4-byte integer value that uniquely identifies each portal registered in the iSNS. The Portal Index is assigned by

iSNS server during the initial registration of a portal. The value MAY BE assigned using a monotonically increasing process.

The Portal Index can be used to represent a registered portal in

situations where the Portal IP-Address and Portal TCP/UDP Port is unwieldy to use. An example of this is when SNMP tables are used to access the contents of the iSNS server. In this case, the Portal Index may be used as the Registered Portal table index.

6.3.8 SCN Port

This field contains the TCP or UDP port used by the iSNS client to receive SCN messages from the iSNS server. Bit 0 to 15 represents the port number. If bit 16 is set, then the port type is UDP. Otherwise, the port is TCP. If bit 16 is disabled, then the port number is for UDP. Bits 17 to 31 are reserved.

The iSNS server SHALL return an error if an SCN registration message is received and none of the portals of the iSNS client has an entry for the SCN Port. If multiple Portals have a registered SCN Port, then the SCN may be delivered to any of the indicated portals.

6.3.9 Security Bitmap

This field contains flags that indicate security attribute settings for the Portal. Bit 0 of this field must be 1 (enabled) in order for this field to contain significant information. If Bit 0 is enabled, this signifies the iSNS server can be used to store and distribute security policies and settings for iSNS clients (i.e., iSCSI devices).

Bit Field	Flag Description
0	1 = Bitmap VALID; 0 = INVALID
1	1 = IPSec Enabled; 0 = IPSec Disabled
2	1 = IKE Enabled; 0 = IKE Disabled
3	1 = Main Mode Enabled; 0 = MM Disabled
4	1 = Aggressive Mode Enabled; 0 = Disabled
5-9	RESERVED
10	1 = PFS Enabled; 0 = PFS Disabled
11	1 = Pre-shared Key Enabled; 0 = PSK Disabled
12	1 = Certificate Support Enabled; 0 = Disabled
All others reserved	

6.3.10 Portal ISAKMP Phase-1 Proposals

This field contains the IKE Phase-1 proposal listing in decreasing order of preference of the protection suites acceptable to protect all IKE Phase-2 messages sent and received by the Portal. This includes Phase-2 SA's from the iSNS client to the iSNS server as well as to peer iFCP and/or iSCSI devices. This attribute contains the SA payload, proposal payload(s), and transform payload(s) in the ISAKMP format defined in [RFC2408].

This field should be used if the implementer wishes to define phase-1 SA security configuration on a per-interface basis, as opposed to on a per-device basis. If the implementer desires to have a single phase-1 SA security configuration to protect all phase-2 traffic regardless of the interface used, then the Entity Phase-1 Proposal (section 6.2.8) should be used.

6.3.11 Portal ISAKMP Phase-2 Proposals

This field contains the IKE Phase-2 proposal, in ISAKMP format [RFC2408], listing in decreasing order of preference the security proposals acceptable to protect traffic sent and received by the Portal. This field is used only if bits 0, 1 and 2 of the Security Bitmap (see 6.3.9) are enabled. This attribute contains the SA payload, proposal payload(s), and associated transform payload(s) in the ISAKMP format defined in [RFC2408].

6.3.12 Portal Certificate

This attribute contains one or more X.509 certificates that is a credential of the PORTAL. This certificate is used to identify and authenticate communications to the IP address supported by the Portal.

6.4 iSCSI Node-Keyed Attributes

The following attributes are stored in the iSNS using the iSCSI Name attribute as the key. Each set of Node-Keyed attributes is associated with one Entity Identifier object key.

Although the iSCSI Name key is associated with one Entity Identifier, it is unique across the entire iSNS.

6.4.1 iSCSI Name

This identifier uniquely defines an iSCSI STORAGE NODE, and is a variable-length text-based value from 0 to 256 bytes. This field is required for iSCSI STORAGE NODEs, and is provided by the iSNS client.

If an iSCSI Name is registered without an EID key, then an Entity will be created and an EID assigned. The assigned EID will be returned in the registration response as an operating attribute.

6.4.2 iSCSI Node Type

This required 32-bit field is a bitmap indicating the type of iSCSI STORAGE NODE. The bit fields are defined below. An enabled bit indicates the node has the corresponding characteristics.

Bit Field	Node Type
0 (Lsb)	Target
1	Initiator
2	Control
All Others	RESERVED

If the 'Target' bit is set, then the node represents an iSCSI target. Setting of the 'Target' bit MAY be performed by iSNS clients using the iSNSP.

If the 'Initiator' bit is set, then the node represents an iSCSI initiator. Setting of the 'Initiator' bit MAY be performed by iSNS clients using the iSNSP.

If the control bit is set, then the node represents a gateway, management station, backup iSNS server, or other device which is not an initiator or target that requires the ability to send and receive iSNSP messages, including state change notifications. Setting of the control bit is an administrative task that MUST be performed on the iSNS server; iSNS clients SHALL NOT be allowed to change this bit using the iSNSP.

This field MAY be used by the iSNS server to distinguish among permissions by different iSCSI node types for accessing various iSNS functions. For example, an iSNS server implementation may be administratively configured to allow only targets to receive ESI's, or for only control nodes to have permission to add, modify, or delete discovery domains.

6.4.3 iSCSI Node Alias

This field is a variable-length text-based value from 0 to 256 bytes. The text field contains user-readable UTF-8 text, and is

terminated with at least one NULL character. The Alias is a user-readable description of the node entry in the iSNS.

6.4.4 iSCSI Node SCN Bitmap

This field indicates the events that the iSCSI Node is interested in. These events can cause a State Change Notification (SCN) to be generated. SCNs provide information about objects that are updated, added or removed from Discovery Domains that the source and destination are a member of. Detailed SCNs provide information about all changes to the network, and may be sent if requested and administratively allowed. Target and Self SCN's (bit 6) may be useful for iSCSI initiators. This SCN provides information only about changes to target devices, or if the iSCSI Node itself has undergone a change. Similarly, Initiator and Self SCN's (bit 7) may be useful for iSCSI targets, by providing information only about changes to initiator nodes, or the target itself.

Bit Field	Flag Description
0	MEMBER ADDED (DETAILED SCN ONLY)
1	MEMBER REMOVED (DETAILED SCN ONLY)
2	OBJECT UPDATED
3	OBJECT ADDED
4	OBJECT REMOVED
5	DETAILED SCN REQUESTED/SENT
6	TARGET AND SELF INFORMATION ONLY
7	INITIATOR AND SELF INFORMATION ONLY
All others	RESERVED

6.4.5 iSCSI Node Index

The iSCSI Node Index is a 4-byte integer value that uniquely identifies each iSCSI node registered in the iSNS. The iSCSI Node Index is assigned by the iSNS server during the initial registration of the iSCSI node. The value MAY BE assigned using a monotonically increasing process.

The iSCSI Node Index may be used to represent a registered node in situations where the iSCSI Name is too long to be used. An example of this is when SNMP tables are used to access the contents of the iSNS server. In this case, the iSCSI Node Index may be used as the

registered iSCSI Node table index.

6.4.6 EUI64 Token

This field contains a globally unique 64-bit integer value that can be used to represent the World Wide Node Name of the iSCSI device in a Fibre Channel fabric. It is a globally unique identifier is used during the device registration process, and uses a value conforming to IEEE EUI-64 [EUI-64].

The FC-iSCSI gateway uses the value found in this field to register the iSCSI device in the Fibre Channel name server. It is stored in the iSNS to prevent conflict when assigning "proxy" WWNN values to iSCSI initiators establishing storage sessions to devices in the FC fabric.

The iSNS server SHALL provide a value for this field upon registration of the iSCSI node, by applying a deterministic process to derive an EUI64 Token from the iSCSI name used to key the iSCSI node registration. The process by which the EUI64 Token is created by the iSNS server MUST include the following requirements:

1. The created EUI64 Token MUST be unique across the entire iSNS database. Collisions with existing EUI64 Token entries should be resolved through techniques such as modifying the fixed seed in the hash function used to create the EUI64 Token.

2. The created EUI64 Token MUST be derived using a repeatable, deterministic process. That is, successive re-registrations of the

same iSCSI node keyed by the same iSCSI Name results in the iSNS server creating the same EUI64 Token.

3. The created EUI64 Token MUST conform to the formatting requirements of [FC-FS] for World Wide Names (WWN's).

An iSNS client, such as an FC-iSCSI gateway or the iSCSI initiator, MAY overwrite the iSNS Server-supplied EUI64 value if it wishes to supply its own iSCSI-FC name mapping. If an iSNS client attempts to register a value for this field that is not unique in the iSNS database, then the registration SHALL be rejected with an error code of 3 (Invalid Registration).

6.4.7 iSCSI Node Certificate

This attribute contains one or more X.509 certificates that may be a credential used to authenticate the iSCSI node during iSCSI authentication.

6.5 FC Port-Keyed Attributes

The following attributes are registered in the iSNS using the FC Port World Wide Name (WWPN) attribute as the key. Each set of FC Port-Keyed attributes is associated with one Entity Identifier object key.

Although the FC Port World Wide Name is associated with one Entity Identifier, it is also globally unique.

6.5.1 Port Name (WWPN)

This 64-bit identifier uniquely defines the FC Port, and is the World Wide Port Name (WWPN) of the corresponding Fibre Channel device. This globally unique identifier is used during the device registration process, and uses a value conforming to IEEE EUI-64 [EUI-64].

6.5.2 Port ID

Along with the IP Address, this field uniquely identifies a native Fibre Channel device port in the network, and maps one-to-one to a specific Port Name (WWPN) entry.

6.5.3 Port Type

Indicates the type of FC port. Encoded values for this field are listed in the following table:

Type	Description
0x0000	Unidentified/Null Entry
0x0001	Fibre Channel N_Port

0x0002	Fibre Channel NL_Port
0x0003	Fibre Channel F/NL_Port
0x0004-0080	RESERVED
0x0081	Fibre Channel F_Port
0x0082	Fibre Channel FL_Port
0x0083	RESERVED
0x0084	Fibre Channel E_Port
0x0085-00FF	RESERVED
0xFF11	mFCP Port
0xFF12	iFCP Port
0xFF13-FFFF	RESERVED

6.5.4 Symbolic Port Name

A variable-length text-based description of up to 255 bytes, that is associated with the iSNS-registered Port Name in the network. The text field contains user-readable UTF-8 text and is terminated with at least one NULL character.

6.5.5 Fabric Port Name (FWWN)

This 64-bit identifier uniquely defines the fabric port. If the iSNS client is attached to a Fibre Channel fabric port with a registered Port Name, then that fabric Port Name shall be indicated in this field.

6.5.6 Hard Address

This field is the requested hard address 24-bit NL Port Identifier, included in the iSNSP for compatibility with Fibre Channel Arbitrated Loop devices and topologies.

6.5.7 Port IP Address

The Fibre Channel IP address associated with the FC Port. When an IPv4 value is contained in this field, the most significant 12 bytes are set to 0x00.

6.5.8 Class of Service (COS)

This 32-bit bit-map field indicates the Fibre Channel COS types that are supported by the registered port. The COS values are equivalent to Fibre Channel COS values. The valid COS types, and associated bit-map, are listed in the following table:

Class of Service	Description	Bit-Map
2	Delivery Confirmation Provided	bit 2 set
3	Delivery Confirmation Not Provided	bit 3 set
	RESERVED	other

6.5.9 FC-4 Types

This 32-byte field indicates the FC-4 protocol types supported by the associated port. This field can be used to support Fibre Channel devices and is consistent with FC-GS-4.

6.5.10 FC-4 Descriptor

A variable-length text-based description of up to 256 bytes, that is associated with the iSNS-registered device port in the network. This field can be used to support Fibre Channel devices and is consistent with FC-GS-4.

6.5.11 FC-4 Features

This is a 128-byte array, 4 bits per type, for the FC-4 protocol types supported by the associated port. This field can be used to support Fibre Channel devices and is consistent with FC-GS-4.

6.5.12 iFCP SCN Bitmap

This field indicates the events that the iSNS client is interested in. These events can cause SCN to be generated. SCNs provide information about objects that are updated, added or removed from Discovery Domains that the source and destination are a member of. Detailed SCNs provide information about all changes to the network, and may be sent if requested and administratively allowed

Bit Field	Flag Description
0	MEMBER ADDED (DETAILED SCN ONLY)
1	MEMBER REMOVED (DETAILED SCN ONLY)
2	OBJECT UPDATED
3	OBJECT ADDED
4	OBJECT REMOVED
5	DETAILED SCN REQUESTED/SENT
6	TARGET AND SELF INFORMATION ONLY
7	INITIATOR AND SELF INFORMATION ONLY

All others	RESERVED

6.5.13 iFCP Port Type

This required 32-bit field is a bitmap indicating the type of iFCP STORAGE NODE. The bit fields are defined below. An enabled bit indicates the node has the corresponding characteristics.

Bit Field	Node Type
0 (Lsb)	Target
1	Initiator
2	Control
All Others	RESERVED

If the 'Target' bit is set, then the port represents an FC target. Setting of the 'Target' bit MAY be performed by iSNS clients using the iSNSP.

If the 'Initiator' bit is set, then the port represents an FC initiator. Setting of the 'Initiator' bit MAY be performed by iSNS clients using the iSNSP.

If the 'control' bit is set, then the port represents a gateway, management station, iSNS backup server, or other device which is not an initiator or target that requires the ability to send and receive iSNSP messages, including state change notifications. Setting of the control bit is an administrative task that MUST be performed on the iSNS server; iSNS clients SHALL NOT be allowed to change this bit using the iSNSP.

This field MAY be used by the iSNS server to distinguish among permissions by different iSNS clients. For example, an iSNS server implementation may be administratively configured to allow only targets to receive ESI's, or for only control nodes to have permission to add, modify, or delete discovery domains.

6.5.14 Port Certificate

This attribute contains one or more X.509 certificates that is a credential of the iFCP STORAGE NODE.

6.6 Node-Keyed Attributes

The following attributes are registered in the iSNS using the FC Node Name (WWNN) attribute as the key. Each set of FC Node-Keyed attributes represents a single device, and can be associated with many FC Ports.

The FC Node Name is unique across the entire iSNS.

6.6.1 Node Name (WWNN)

The FC Node Name is a 64-bit identifier that is the World Wide Node Name (WWNN) of the corresponding Fibre Channel device. This globally unique identifier is used during the device registration process, and uses a value conforming to IEEE EUI-64 [EUI-64].

6.6.2 Symbolic Node Name

A variable-length text-based description of up to 256 bytes, that is associated with the iSNS-registered FC Device in the network. The

text field contains user-readable UTF-8 text and is terminated with at least one NULL character.

6.6.3 Node IP Address

This IP address is associated with the device node in the network. This field is included for compatibility with Fibre Channel. When an IPv4 value is contained in this field, the most significant 12 bytes are set to 0x00.

6.6.4 Node IPA

This field is the 8 byte Fibre Channel Initial Process Associator (IPA) associated with the device node in the network. The initial process associator can be used for communication between Fibre Channel devices.

6.6.5 Node Certificate

This attribute contains an X.509 certificate that is bound to the FC Node of the iSNS client.

6.6.6 Proxy iSCSI Name

This field contains the iSCSI Name used to represent the FC Node in the IP network. It is used as a pointer to the matching iSCSI Name entry in the iSNS server. Its value is usually registered by an FC-iSCSI gateway connecting the IP network to the fabric containing the FC device.

Note that if this field is used, there SHOULD be a matching entry in the iSNS database for the iSCSI device specified by the iSCSI name. The database entry should include the full range of iSCSI attributes needed for discovery and management of the "iSCSI proxy image" of the FC device.

6.7 Other Attributes

The following are not attributes of the previously-defined objects.

6.7.1 FC-4 Type Code

This is a 4-byte field, and is used to provide a FC-4 type during a FC-4 Type query. The FC-4 types are consistent with the FC-4 Types as defined in FC-PH. Byte 0 contains the FC-4 type. All other bytes are reserved.

6.7.2 iFCP Switch Name

The iFCP Switch Name is a 64-bit World Wide Name (WWN) identifier that uniquely identifies the iFCP switch in the network. This globally unique identifier is used during the switch registration switch ID assignment process, and uses a value conforming to IEEE EUI-64 [EUI-64]. The iSNS server SHALL track the state of all

Switch_ID values that have been allocated to each iFCP Switch Name. If a given iFCP Switch Name is deregistered from the iSNS database, then all Switch_ID values allocated to that iFCP Switch Name shall be returned to the unused pool of values.

6.7.3 Preferred ID

This is a 4-byte unsigned integer field, and is the requested value that the iSNS client wishes to use for the SWITCH_ID. The iSNS server SHALL grant the iSNS client the use of the requested value as the SWITCH_ID, if the requested value has not been already allocated. If the requested value is not available, the iSNS server SHALL return a different value that has not been allocated.

6.7.4 Assigned ID

This is a 4-byte unsigned integer field that is used to support iFCP Transparent Mode. When operating in iFCP Transparent Mode, the RqstSwId message SHALL be used by each iFCP gateway to reserve its own unique SWITCH_ID value from the range 1 to 239. When a Switch ID is no longer required, it SHALL be released by the iFCP gateway using the RlseSwId message. The iSNS MAY use the Entity Status Inquiry message to determine if an iFCP gateway is still present on the network.

6.7.5 Space_Identifier

This is a UTF-8 encoded string. The Space_Identifier string is used as a key attribute to identify a range of non-overlapping SWITCH_ID values to be allocated using RqstSwId. Each Space_Identifier string submitted by iSNS clients shall have its own range of non-overlapping SWITCH_ID values to be allocated to iSNS clients.

6.7.6 Server-Specific Attributes

Attributes with tags in the range 131 to 255 are server-specific and vendor-specific (see section 6.9). These attributes are unique for each logical iSNS server instance. Query and registration messages for these attributes SHALL NOT contain a key attribute.

6.8 Discovery Domain Registration Attributes

iSNS STORAGE NODE objects can be placed into Discovery Domains. Only objects that share the same enabled Discovery Domain can query for information about each other. Discovery Domains can overlap, so an iSCSI node may be a member of many DDÆs.

Enabled Discovery Domains are members of one or more enabled Discovery Domain Sets (DDS). Discovery Domains that are not members of at least one enabled DDS are disabled. Therefore, Discovery Domains are not directly enabled, but rather are enabled through their association with one or more enabled Discovery Domain Sets (DDS). Discovery Domain Sets are enabled by setting bit 0 in the DDS Status field.

6.8.1 iSNS Discovery Domain Attribute Summary

The following table lists the iSNSP DD attributes:

Attribute Name	Size(bytes)	ID	Reg Key	Query Key
DD_Set ID	4	101	@	1,32,64,101,104
DD_Set Sym Name	4-256	102	101	101
DD_Set Status	4	103	101	101
DD_ID	4	104	@|101*	1,32,64,101,104
DD_Symbolic Name	4-256	105	104	104
DD_iSCSI Node Index	4	106	104	104
DD_iSCSI Node Member	0-256	107	104	104
DD_iFCP Member (WWPN)	8	108	104	104
DD_Features	4	109	104	104

@ = no key required during registration
| = either key can be used during registration
* = When a DD ID is placed into a DD Set by using the DDS ID as a key

All undefined tag values are reserved.

6.8.2 DD Set ID Keyed Attributes

6.8.2.1 Discovery Domain Set ID (DDS ID)

The DDS ID is a unique unsigned integer identifier used in the iSNS directory database to indicate a Discovery Domain Set. A DDS is a collection of Discovery Domains that can be enabled or disabled by a management station. This value is used as a key for DDS attribute queries. When a Discovery Domain is registered it is initially not in any DDS.

If the iSNS client does not provide a DDS_ID in a DDS registration request message, the iSNS shall generate a DDS_ID value that is unique within the iSNS database for that new DDS. The created DDS ID shall be returned in the response message. The DDS ID value of 0 is reserved.

6.8.2.2 Discovery Domain Set Symbolic Name

The DDS_Symbolic Name is a UTF-8, variable-length, NULL-terminated string. This is an user-readable field used to assist a network administrator in tracking the DDS function. When registered by a client, the DDS symbolic name SHALL be verified unique by the iSNS. If the DDS symbolic name is not unique, then the DDS registration SHALL be rejected with an ôInvalid Registrationö error code. The invalid attribute(s), in this case the DDS symbolic name, SHALL be included in the response.

6.8.2.3 Discovery Domain Set Status

The DDS_Status field is a 32-bit bitmap indicating the status of the DDS. Bit 0 of the bitmap indicates whether the DDS is Enabled (1) or Disabled (0). The default value for the DDS Enabled flag is Disabled (0).

Bit Field	DDS Status
0 (Lsb)	DDS Enabled (1) / DDS Disabled (0)
All Others	RESERVED

6.8.2.4 Discovery Domain Set Member

The Discovery Domain Set Member is a DD ID for a previously registered Discovery Domain. The DD ID tag value is used to represents membership.

6.8.3 DD ID Keyed Attributes

6.8.3.1 Discovery Domain ID (DD ID)

The DD ID is a unique unsigned integer identifier used in the iSNS directory database to indicate the DD. This value is used as the key for any DD attribute query. If the iSNS client does not provide a DD_ID in a DD registration request message, the iSNS shall generate a DD_ID value that is unique within the iSNS database for that new DD (i.e., the iSNS client will be registered in a new DD). The created DD ID shall be returned in the response message. The DD ID value of 0 is reserved.

6.8.3.2 Discovery Domain Symbolic Name

The DD_Symbolic Name is a UTF-8 encoded, variable-length, NULL-terminated string. When registered by a client, the DD symbolic name SHALL be verified unique by the iSNS. If the DD symbolic name is not unique, then the DD registration SHALL be rejected with an ôInvalid Registrationö error code. The invalid attribute(s), in this case the DD symbolic name, SHALL be included in the response.

6.8.3.3 Discovery Domain iSCSI Node Index

This is the iSCSI Node Index of an iSNS client that is a member of

the DD. The DD may have a list of 0 to n members. The iSCSI Node Index is one alternate representation of membership in a Discovery Domain, the other alternative being the iSCSI Node Name. The Discovery Domain iSCSI Node Index is a 4-byte integer value.

The iSCSI Node Index can be used to represent a DD member in situations where the iSCSI Name is too long to be used. An example of this is when SNMP tables are used to access the contents of the iSNS server.

The iSCSI Node Index and iSCSI Node Name registered as a member in a DD SHALL be consistent with the iSCSI Node Index and iSCSI Node Name used for the registered node in the iSNS.

Both the iSCSI Name and iSCSI Node Index of a member are registered in the DD in order to maintain the unique 1:1 mapping between the two attributes for the member over multiple registration / deregistrations of the same member in the iSNS.

6.8.3.4 Discovery Domain Member--iSCSI Name

The iSCSI Name of an iSNS client that is a member of the DD. The DD may have a list of 0 to n members. The iSCSI Name of the iSNS client represents membership.

6.8.3.5 Discovery Domain Member--Port Name

The Port Name of an iSNS client that is a member of the DD. The DD may have a list of 0 to n members. Membership is represented by the Port Name (WWPN) of the iSNS client being listed.

6.8.3.6 Discovery Domain Features

The Discovery Domain Features is a bitmap indicating the features of this DD. The bit fields are defined below. An enabled bit indicates the DD has the corresponding characteristics.

Bit Field	DD Feature
0 (Lsb)	Boot List
All Others	RESERVED

Boot List: this feature indicates that the targets in this DD provide boot capabilities for the member initiators.

6.9 Vendor-Specific Attributes

Specific iSNS implementations MAY define vendor-specific attributes for private use. The tag values reserved for vendor-specific and user-specific use are defined in section 6.1. To avoid misinterpreting proprietary attributes, it is RECOMMENDED that the vendor's own OUI (Organizationally Unique Identifier) be placed in the upper three bytes of the attribute field itself. If the OUI is not used, then some other unique marker recognizable by the vendor SHOULD be used. The OUI is defined in IEEE Std 802-1990, and is the same constant used to generate 48 bit Universal LAN MAC addresses. A vendor's own iSNS implementation will then be able to recognize the OUI in the vendor-specific or user-specific attribute field, and be able to execute vendor-specific handling of the attribute.

6.10 Company OUI

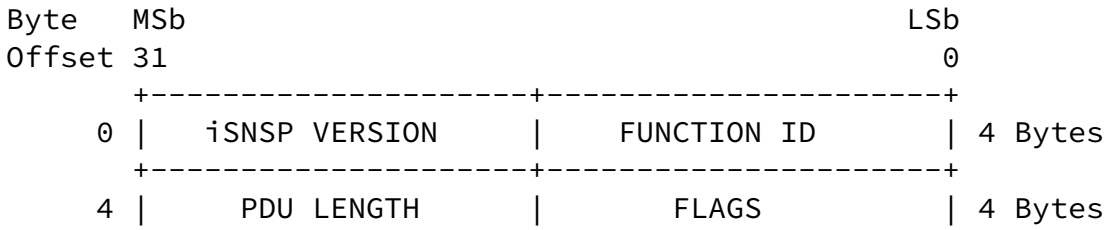
This attribute is the OUI (Organizationally Unique Identifier) identifying the specific vendor implementing the iSNS. It is used to identify the original creator of a vendor-specific iSNSP message.

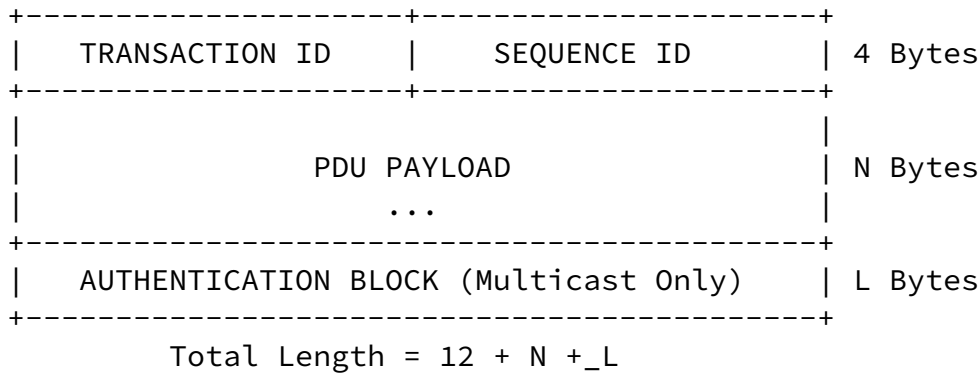
6.11 Standards-Based Extensions

These attributes are reserved for future work by other standards bodies.

7. iSNSP Message Format

The iSNSP message format is similar to the format of other common protocols such as DHCP, DNS and BOOTP. An iSNSP message may be sent in one or more iSNS Protocol Data Units (PDU). Each PDU is 4 byte aligned. The following describes the format of the iSNSP PDU:





7.1 iSNSP PDU Header

The iSNSP header contains the iSNSP VERSION, FUNCTION ID, PDU LENGTH, FLAGS, TRANSACTIONID, and SEQUENCE ID fields as defined below.

7.1.1 iSNSP Version

The iSNSP version is currently 0x0001.

7.1.2 iSNSP Function ID

The FUNCTION ID defines the type of iSNS message and the function the message is supporting. See section 5 under the appropriate protocol (i.e., iSCSI or iFCP) for a mapping of the FUNCTION_ID value to the iSNSP Command or Response message. All PDU's comprising an iSNSP message must have the same FUNCTION_ID and TRANSACTION ID value.

7.1.3 iSNSP PDU Length

The iSNS PDU LENGTH specifies the length of the PDU PAYLOAD field in bytes. The payload contains the data/attribute values for the operation.

7.1.4 iSNSP Flags

The FLAGS field indicates additional information about the message and the type of iSNS entity that generated the message. The following table displays the valid flags:

Bit Field	Enabled Means:

0-9	RESERVED
10	First PDU of the iSNS message
11	Last PDU of the iSNS message
12	Replace Flag (valid only for RegDevAttr)
13	RESERVED
14	Sender is the iSNS server
15	Sender is the iSNS client

7.1.5 iSNSP Transaction ID

The TRANSACTION ID is set to a unique random value for each request message. Replies MUST use the same TRANSACTION ID value as the associated iSNS request message. If a message is retransmitted, the same TRANSACTION ID value MUST be used.

7.1.6 iSNSP Sequence ID

The SEQUENCE ID is set to a unique value for each PDU within a single transaction. Each SEQUENCE_ID value in each PDU SHALL be numbered sequentially in the order that the PDU's are transmitted. If a message is retransmitted, then the same SEQUENCE_ID value MUST be used for all PDU's in the message.

7.2 iSNSP Message Segmentation and Reassembly

iSNS messages may be carried in one or more iSNS PDU's. If only one iSNS PDU is used to carry the iSNS message, then bit 10 (First PDU) and bit 11 in the FLAGS field (Last PDU) SHALL both be enabled. If multiple PDUs are used to carry the iSNS message, then bit 10 SHALL be enabled in the first PDU of the message, and bit 11 SHALL be enabled in the last PDU.

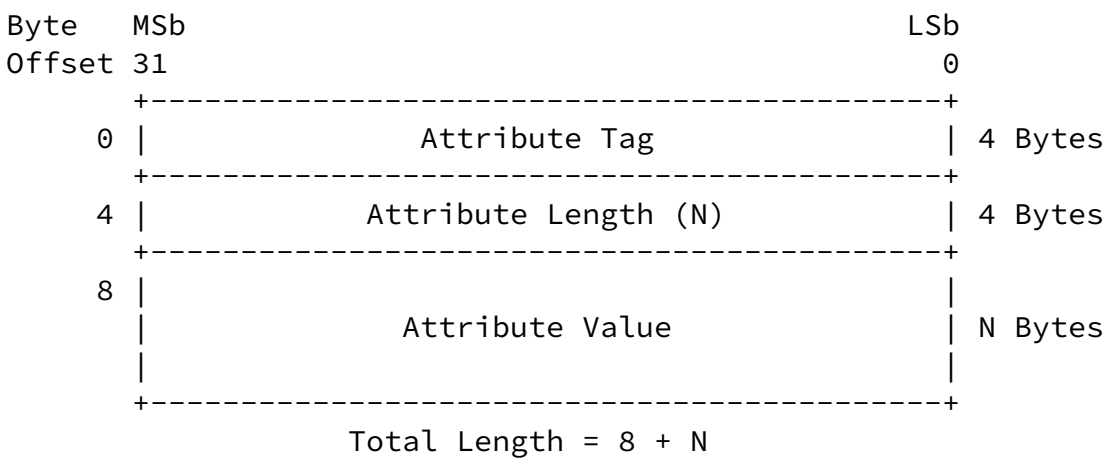
All PDU's comprising the same iSNSP message SHALL have the same FUNCTION_ID and TRANSACTION_ID values. Each PDU comprising an iSNSP message SHALL have a unique SEQUENCE_ID value.

The authentication operation described in section 7.5 SHALL be performed on a per-PDU basis.

7.3 iSNSP Message Payload

The MESSAGE PAYLOAD is variable length and contains attributes used for registration and query operations. The attribute data items use

a format similar to other protocols, such as DHCP (RFC 2131) options. Each iSNS attribute is specified in the iSNSP message payload using Tag-Length-Value (TLV) data format, as shown below:



Attribute Tag - a 4-byte tag field that identifies the attribute as defined in section 6.1. This field contains the ID of the indicated attribute.

Attribute Length - a 4-byte field that indicates the length, in bytes, of the attribute value to follow.

Attribute Value - a variable-length field containing the attribute value.

The above format is used to identify each attribute in the iSNS message payload. Each iSNSP request message contains several attributes in the above format to identify the requesting iSNS client and register or query for attribute values in the iSNS server.

7.3.1 Attribute Value 4-Byte Alignment

All attribute values are aligned at 4 byte boundaries. For variable length attributes, the value length is increased to the next 4-byte boundary and the value is NULL padded.

7.4 iSNSP Response Error Codes

All iSNSP response messages contain a 4-byte ERROR CODE field as the first field in the iSNSP PAYLOAD. If the original iSNSP request message was processed normally by the iSNS server, or the iSNS client for ESI and SCN messages, the field SHALL contain 0x00000000 (NO ERROR).

Error Code	Error Description
0	No Error
1	Unknown Error
2	Message Format Error
3	Invalid Registration
4	Requested ESI Period Too Short
5	Invalid Query
6	Authentication Unknown
7	Authentication Absent
8	Authentication Failed
9	No Such Entry
10	Version Not Supported
11	Internal Bus Error
12	Busy Now
13	Option Not Understood
14	Invalid Update
15	Message Not Supported
16	SCN Event Rejected
17	SCN Registration Rejected
18	Attribute not Implemented
19	SWITCH_ID not available
20	SWITCH_ID not allocated
21	ESI Not Available
22 And Above	RESERVED

All undefined Error Code values are RESERVED.

7.5 iSNS Multicast Message Authentication

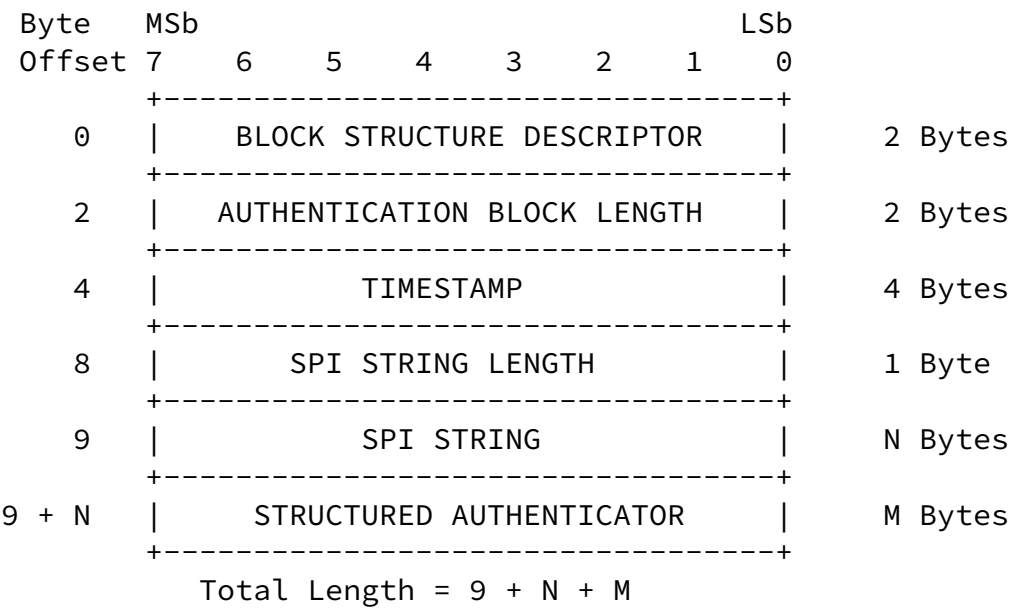
For iSNS multicast messages, the iSNSP provides authentication capability. The following section details the iSNS Authentication Block, which is identical in format to the SLP authentication block [RFC2608]. iSNS unicast messages SHOULD NOT include the authentication block, but rather should rely upon IPSec security mechanisms.

If a PKI is available with an X.509 certificate authority, then public key authentication of the iSNS server is possible. The authentication block leverages the DSA with SHA-1 algorithm, which can easily integrate into a public key infrastructure.

The authentication block contains a digital signature for the multicast message. The digital signature is calculated on a per-PDU basis. The authentication block contains the following information:

1. A time stamp, to prevent replay attacks
2. A structured authenticator containing a signature calculated over the time stamp and the message being secured
3. An indicator of the cryptographic algorithm that was used to calculate the signature.
4. An indicator of the keying material and algorithm parameters, used to calculate the signature.

The authentication block is described in the following figure:



BLOCK STRUCTURE DESCRIPTOR (BSD) - Defines the structure and algorithm to use for the STRUCTURED AUTHENTICATOR. Currently, the only defined value for BSD is 0x0002, which represents DSA with SHA-1. Details on DSA can be found in [DSS]. BSD values from 0x0000 to 0x7FFF are assigned by IANA, while 0x8000 to 0x8FFF are for private use. The BSD value 0x0002 is compatible with the X.509 PKI specification, allowing easy integration of the STRUCTURED AUTHENTICATOR format with an existing PKI infrastructure.

AUTHENTICATION BLOCK LENGTH - Defines the length of the authentication block, beginning with the BSD field and running through the last byte of the STRUCTURED AUTHENTICATOR.

TIMESTAMP - This is a 4-byte unsigned, fixed-point integer giving the number of seconds since 00:00:00 GMT on January 1, 1970.

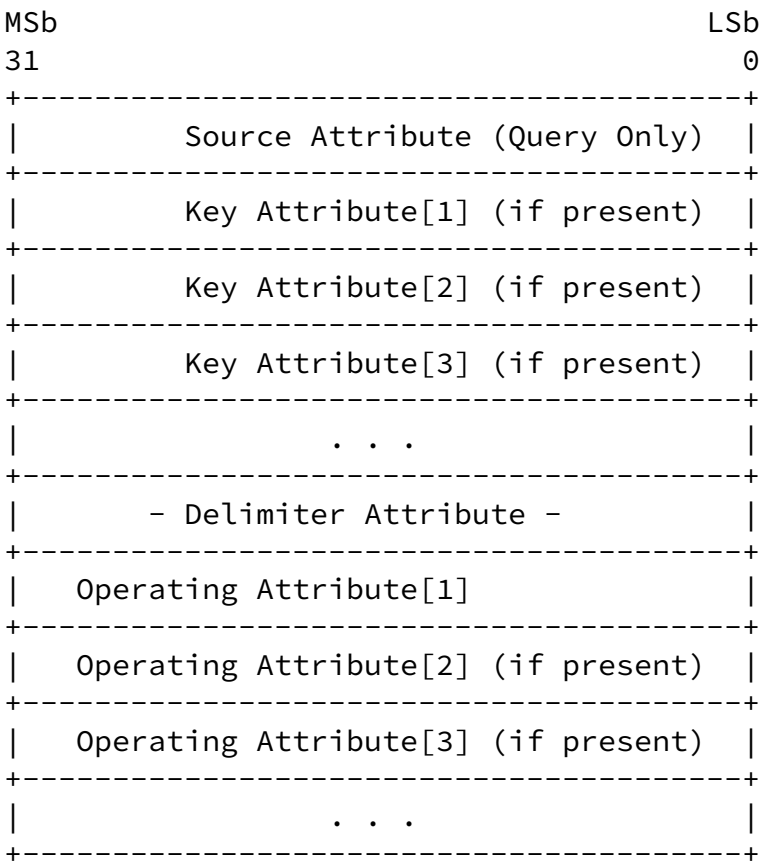
SPI STRING LENGTH - The length of the SPI STRING field.

SPI STRING (Security Parameters Index) - Index to the key and algorithm used by the message recipient to decode the STRUCTURED AUTHENTICATOR field.

STRUCTURED AUTHENTICATOR - Contains the digital signature. For the default BSD value of 0x0002, this field contains the binary ASN.1 encoding of output values from the DSA with SHA-1 signature calculation.

7.6 Registration and Query Messages

The iSNSP registration and query message payloads contain a list of attributes, and have the following format:



iSNS Registration and Query messages, sent by iSNS Clients, are sent to the iSNS IP-Address and TCP/UDP Port. The iSNS Responses will be sent to the iSNS Client IP-Address and the originating TCP/UDP Port used for the associated registration and query message.

7.6.1 Source Attribute

The source attribute is used to identify the iSNS client to the iSNS server for queries and other messages that require source identification. The source attribute uniquely identifies the source of the message. Valid source attribute types are shown below.

Valid Source Attributes
iSCSI Name
FC Port Name WWPN

For a query operation, the source attribute is used to limit the scope of the specified operation to the Discovery Domains of which the source is a member. Special control nodes, identified by the SOURCE attribute, may be administratively configured to perform the specified operation on all objects in the iSNS database without scoping to Discovery Domains.

7.6.2 Key Attributes

Key attributes are used to identify the object (or objects) in the iSNS server that the registration or query operation will be performed on. The number of Key Attributes depends on the specific iSNSP request or query operation being performed.

7.6.3 Delimiter Attribute

The Delimiter Attribute separates the key attributes from the operating attributes in a message payload. The Delimiter Attribute has a tag value of 0 and a length value of 0. The Delimiter Attribute is effectively 8 Bytes long, a 4 Byte tag containing 0x00000000, and a 4 Byte length field containing 0x00000000.

7.6.4 Operating Attributes

The Operating Attributes are a list of one or more attributes related to the actual iSNS registration or query operation being performed. In a registration, operating attributes represent values to be registered by the iSNS client performing the registration. In

a query, operating attributes represent values being requested by the iSNS client.

The number of possible Operating Attributes depends on the specific iSNSP request or query. For example, the Operating Attributes in a Device Attribute Query message are the set of attributes to be returned in the associated Device Attribute Query Response message that match the Key Attributes of the query.

Some iSNSP messages do not require any Operating Attributes.

7.6.4.1 Operating Attributes for Query and Get Next Requests

In Query and Get Next request messages, TLV attributes with length value of 0 are used to indicate what operating attributes are to be returned in the corresponding response. Operating Attribute values that match the TLV attributes in the original message are returned in the response message.

7.6.5 Registration and Query Message Types

The following describes each query and message type.

7.6.5.1 Register Device Attribute Request (RegDevAttr)

The RegDevAttr message type is 0x0001. The RegDevAttr message provides an iSNS client with the means to register network entities. The iSNS client formulates a RegDevAttr by specifying Key Attribute(s) and list of Operating Attributes to register. All values are in Tag Length Value (TLV) format.

Attributes following the Delimiter Attribute are Operating Attributes. Depending on the setting of the Replace bit in the FLAGS field, the Operating attribute values in the RegDevAttr message will either replace existing attributes(s), or be added to existing attributes(s). See section 7.6.5.1.1 below for a complete description of the Replace Flag.

The operating attributes are the elements that will be registered. Multiple attributes can be registered in one RegDevAttr. The

ordering of the operating attributes indicates the associations to be created in the iSNS. For example, Portal attributes following Entity attributes SHALL create a link between the registered entity and portal. Similarly, node attributes following entity attributes

will create an association.

A RegDevAttr message with no key attribute results in creation of a new entity (EID). If the EID attribute (with non-zero length) is included among the operating attributes in the RegDevAttr message, then the new entity SHALL be assigned the value contained in that EID attribute. Otherwise, if the EID attribute is not contained among the operating attributes of the RegDevAttr message, or if the EID is an operating attribute with TLV length of 0, then the iSNS SHALL assign the EID value that is returned in the RegDevAttr Response message.

One RegDevAttr message can contain attributes for Entity, Portal, and Node objects if each of these attributes are contained in the same Entity. When the registration contains attributes for the Entity, Portal, and Node objects together in the same message, then the appropriate Portal, and Node key attributes must be registered as part of the operating attributes.

Ordering of the attributes is important in multi-object registrations. For example, Node Attributes follow a valid Node key.

7.6.5.1.1 Replace Flag

The Replace Flag, contained in the message header FLAGS field, indicates whether the registration is a replacement of, or update to, an existing entry. If the Replace bit in the FLAGS field is enabled then a new object entry SHALL be created, replacing the existing object if one exists.

If the key attributes of the registration do not match an existing object then the Replace flag has no effect.

If the key attributes match an existing object in the iSNS, and the Replace flag is enabled, then the registration will replace the existing entry in the iSNS. The existing object(s) specified in the RegDevAttr message shall be de-registered. A new registration shall be created with the new attribute value(s) in the registration request. Existing associations between objects will be updated to reflect the new information. For example, an existing Node object may be de-registered and reregistered with a different Entity object as part of a registration.

If the key attributes match an existing object in the iSNS, and the Replace flag is not enabled, then the new attribute value(s) in the registration request SHALL update existing values and may add new, additional attributes for the key entry. Only non-key attributes can be updated. Existing associations between objects will be maintained. If a registration update of the existing object would

cause a change in associations, then the error ôInvalid Updateô SHALL be returned. For example, if a RegDevAttr message with an Entity Identifier key for one entity contains a Node attribute associated with another entity, then an error shall be returned.

7.6.5.2 Device Attribute Query Request (DevAttrQry)

The DevAttrQry message type is 0x0002. The DevAttrQry message provides an iSNS client with the means to query the iSNS server for network entity attributes. The source is used to scope the query to the Discovery Domains that the source attribute is a member of.

The Key Attribute(s) follow the source attribute in the message payload. The attributes returned by the query will be from objects WHERE the Key Attribute(s) match the object. The Key Attributes map to a type of object.

The DevAttrQry message shall support the following minimum set of Key Attributes:

Valid Key Attributes for Queries
Entity Identifier
Entity Protocol
Portal IP-Address
Portal IP-Address, Portal TCP/UDP Port
iSCSI Node Type
iSCSI Identifier
FC Port Name WWPN
FC Port Type
FC-4 Type
Switch Name (FC Device WWNN--for space identifier queries)

If the network entities matching key attributes are not in the same Discovery Domain as the Source Attribute, then the results for the network entity will not be included in the response message.

The Operating Attributes are the attributes whose values are being queried.

7.6.5.3 Device Get Next Request (DevGetNext)

The DevGetNext message type is 0x0003. This message provides the iSNS client with the means to sequentially retrieve Entity, Portal,

iSCSI Node, Port Name, or Node Name attributes from DD's to which the client has access. The source is used to scope the Get Next process to the Discovery Domains that the source attribute is a member of.

The Key Attribute follows the source attribute in the message payload. The Key Attribute may be an Entity Identifier, iSCSI Name, Portal IP Address and TCP/UDP Port, FC Node Name WWNN, or FC Port Name WWPN. If the key TLV length value entered is zero, signifying an empty key value field, then the first accessible Entity

Identifier, iSCSI Name, Portal IP Address and TCP/UDP Port, FC Node name, or FC Port Name instance shall be returned to the client. DevGetNext SHALL return the object that is stored sequentially after the object matching the key provided. If the key provided matches the last object instance, then the Error Code of "No Such Entry" SHALL be returned in the response.

The values of the matching Operating Attributes listed in the original DevGetNext message SHALL be returned in the DevGetNext response.

7.6.5.4 Deregister Device Request (DeregDev)

The DeregDev message type is 0x0004. An iSNS client port or device is removed from the iSNS directory database by using DeregDev. Upon receiving the DeregDev, the iSNS server removes all object registrations associated with the Key Attribute in the payload.

The DeregDev request message payload contains a Source Attribute and Key Attribute(s). Valid Key Attributes are shown below:

Valid Key Attributes for DeregDev
Entity Identifier
Portal IP-Address
Portal IP-Address, Portal TCP/UDP Port
iSCSI Name
FC Port Name WWPN
FC Node Name WWNN

The removal of the object will initiate an SCN message to registered iSNS clients that are in the same DD as the removed device or port. After removing the port or device, the iSNS server sends back an acknowledgement to the iSNS client.

If all nodes associated with an entity are deregistered from that entity, then the entity SHALL also be removed UNLESS the entity (through one or more Portals) is responding to ESI's.

If all Portals associated with an entity are deregistered from that entity, then that entity and all associated nodes SHALL be removed from the iSNS database.

7.6.5.5 SCN Register Request (SCNReg)

The SCNReg message type is 0x0005. The State Change Notification Registration Request (SCNReg) message allows an iSNS client to register a STORAGE NODE to receive State Change Notification (SCN) messages. SCN messages are sent to the indicated UDP or TCP Port specified in the SCN Port field (tag 23), notifying the iSNS client of changes within the DD or network (if administratively allowed).

The SCNReg request message payload contains a Source Attribute, a Key Attribute(s), and an Operating Attribute. Valid Key Attributes for an SCNReg are shown below:

Valid Key Attributes for SCNReg
iSCSI Name
FC Port Name WWPN

The iSCSI nodes or Node Names matching the Key Attributes are registered to receive SCNs.

The SCN Bitmap is the only operating attribute of this message, and it always overwrites the previous contents of this field in the iSNS database. The bitmap indicates those INTERESTED EVENT TYPES the node is registering for.

7.6.5.6 SCN Deregister Request (SCNDereg)

The SCNDereg message type is 0x0006. The SCNDereg message allows an iSNS client to disable State Change Notification (SCN) messages.

The SCNDereg request message payload contains a Source Attribute and Key Attribute(s). Valid Key Attributes for an SCNDereg are shown

below:

Valid Key Attributes for SCNDereg
iSCSI Name
FC Port Name WWPN

The iSCSI or iFCP nodes matching the Key Attributes are deregistered for SCNs.

There are no Delimiter or Operating Attributes in the SCNDereg message.

7.6.5.7 SCN Event (SCNEvent)

The SCNEvent message type is 0x0007. The SCNEvent is a message generated by an iSNS client. The SCNEvent allows the client to request generation of a State Change Notification (SCN) message by the iSNS server. The SCN, sent by the iSNS server, then notifies iFCP, iSCSI, and control nodes within the affected DD of the change indicated in the SCNEvent.

Most SCNs are automatically generated by the iSNS when nodes are registered or deregistered from the directory database. SCNs are also be generated when a network management application makes changes to the DD membership in the iSNS. However, an iSNS client can trigger an SCN by using SCNEvent.

The SCNEvent message payload contains a Source Attribute, Key Attribute, and Operating Attribute. Valid Key Attributes for an SCNEvent are shown below:

Valid Key Attributes for SCNEvent
iSCSI Name
FC Port Name WWPN

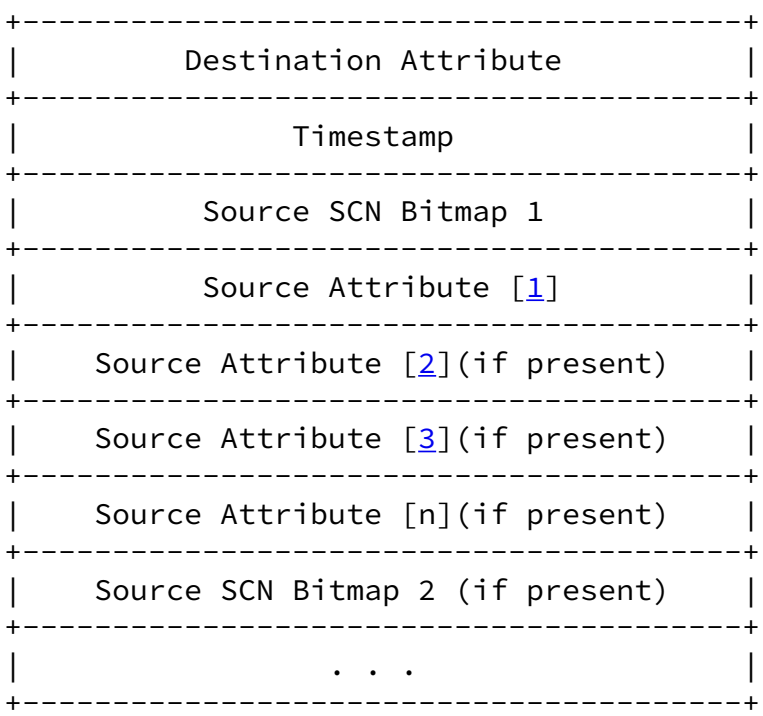
The Operating Attributes section SHALL contain the SCN Event Bitmap attribute. The bitmap indicates the event that caused the SCNEvent to be generated.

7.6.5.8 State Change Notification (SCN)

The SCN message type is 0x0008. The SCN is a message generated by the iSNS server which notifies a registered iFCP, iSCSI, or control node of changes within its DD. The SCN message is sent to each Portal of the affected node that has a registered TCP or UDP Port in the SCN Port field.

The types of events that a node can be notified about are based on the value of the SCN Event Bitmap for the node.

The format of the SCN payload is shown below:



All payload attributes are in TLV format.

The Destination Attribute is the node identifier that is receiving the SCN. The Destination Attribute can be an iSCSI Name, or FC Port Name.

The Timestamp field, using the Timestamp TLV format, indicates the time the SCN was generated.

The Source Attributes describe the object that caused the SCN to be generated. The Source Attributes can be an iSCSI Name, DD ID, DDS ID, or FC Port Name, and possibly include other attributes to describe the change that occurred. The additional attributes are

included to provide additional information about the source to minimize the possibility that the destination object needs to query the server for additional information.

The Source SCN Bitmap field indicates the type of event that caused the SCN to be generated. This field is also used as a delimiter between information about multiple objects, if the SCN message is providing multiple notifications.

7.6.5.9 DD Register (DDReg)

The DDReg message type is 0x0009. This message is used to create a new Discovery Domain (DD), update an existing DD Symbolic Name, and/or add DD members.

DDs are uniquely defined using DD_IDs. DD registration attributes are described in section 6.8.

The DDReg message payload contains the Source Attribute, and optionally Key and Operating Attributes.

A DDReg message with no key attribute results in creation of a new Discovery Domain (DD). If the DD_ID attribute (with non-zero length) is included among the operating attributes in the DDReg message, then the new Discovery Domain SHALL be assigned the value contained in that DD_ID attribute. Otherwise, if the DD_ID attribute is not contained among the operating attributes of the DDReg message, or if the DD_ID is an operating attribute with TLV length of 0, then the iSNS SHALL assign the DD_ID value that is returned in the DDReg Response message.

The Operating Attributes can contain the iSCSI Node Identifier or iFCP WWPN of iSNS clients to be added to the DD. It may also contain the DD_Symbolic_Name of the DD.

This message shall add any DD members listed as operating attributes to the Discovery Domain specified by the DD_ID. In addition, if the DD_Symbolic_Name is an operating attribute, then it will be stored in the iSNS as the DD_Symbolic_Name for the specified Discovery Domain.

7.6.5.10 DD Deregister (DDDereg)

The DDDereg message type is 0x000A. This message allows an iSNS client to deregister an existing Discovery Domain (DD) or remove members from an existing DD.

DDs are uniquely defined using DD_IDs. DD registration attributes are described in section 6.8.3.

The DDDereg message payload contains a Source Attribute, Key Attribute, and Operating Attributes.

The Key Attribute for a DDDereg message is the DD ID for the domain being removed, or having members removed. If the DD ID matches an existing DD, and there are no operating attributes, then the DD will be removed and a success error code returned. If the key attribute does not match an existing DD then the error code ôNo Such Entryö will be returned.

If the DD ID matches an existing DD, and there are operating attributes matching DD members, then the DD members identified by the operating attributes SHALL be removed from the DD and a success error code returned. If any of the operating attributes do not match existing DD members, then the error code ôNo Such Entryö will be returned, and no DD members shall be removed.

7.6.5.11 DDS Register (DDSReg)

The DDSReg message type is 0x000B. This message allows an iSNS client to create a new Discovery Domain Set (DDS), update an existing DDS Symbolic Name, or add DDS members.

DDSÆs are uniquely defined using DDS_IDÆs. DDS registration attributes are described in section 6.8.2.

The DDSReg message payload contains the Source Attribute, and optionally Key and Operating Attributes.

A DDSReg message with no key attribute results in creation of a new Discovery Domain Set (DDS). If the DDS_ID attribute (with non-zero length) is included among the operating attributes in the DDSReg message, then the new Discovery Domain Set SHALL be assigned the value contained in that DDS_ID attribute. Otherwise, if the DDS_ID attribute is not contained among the operating attributes of the DDSReg message, or if the DDS_ID is an operating attribute with TLV length of 0, then the iSNS SHALL assign the DDS_ID value that is returned in the DDSReg Response message.

The Operating Attributes can contain the DDS_Symbolic_Name and the DD_IDÆs of Discovery Domains to be added to the DDS.

This message shall add any DDS members listed as operating attributes to the Discovery Domain Set specified by the DDS_ID key

attribute. In addition, if the DDS_Symbolic_Name is an operating attribute, then it will be stored in the iSNS as the DDS_Symbolic_Name for the specified Discovery Domain Set.

7.6.5.12 DDS Deregister (DDSDereg)

The DDSDereg message type is 0x000C. This message allows an iSNS client to deregister an existing Discovery Domain Set (DDS) or remove some DDÆs from an existing DDS.

The DDSDereg message payload contains a Source Attribute, Key Attribute, and Operating Attributes.

The Key Attribute for a DDSDereg message is the DDS ID for the set being removed, or having members removed. If the DDS ID matches an existing DDS, and there are no operating attributes, then the DDS will be removed and a success error code returned. If the key attribute does not match an existing DDS then the error code ôNo Such Entryö will be returned.

If the DDS ID matches an existing DDS, and there are operating attributes matching DDS members, then the DDS members will be removed from the DDS and a success error code returned. If any of the operating attributes do not match existing DDS members, then the error code ôNo Such Entryö will be returned and no DDS members shall be removed.

7.6.5.13 Entity Status Inquiry (ESI)

The ESI message type is 0x000D. This message is sent by the iSNS server, and is used to verify that an iSNS client portal is reachable and available. The ESI message is sent to the ESI UDP port provided during registration, or the TCP connection used for ESI registration, depending on which communication type that is being used.

The ESI message payload contains several attributes in TLV format, including the current iSNS timestamp, the EID, the Portal IP Address, and Portal TCP/UDP Port.

The ESI response message payload contains the Attributes from the original ESI message.

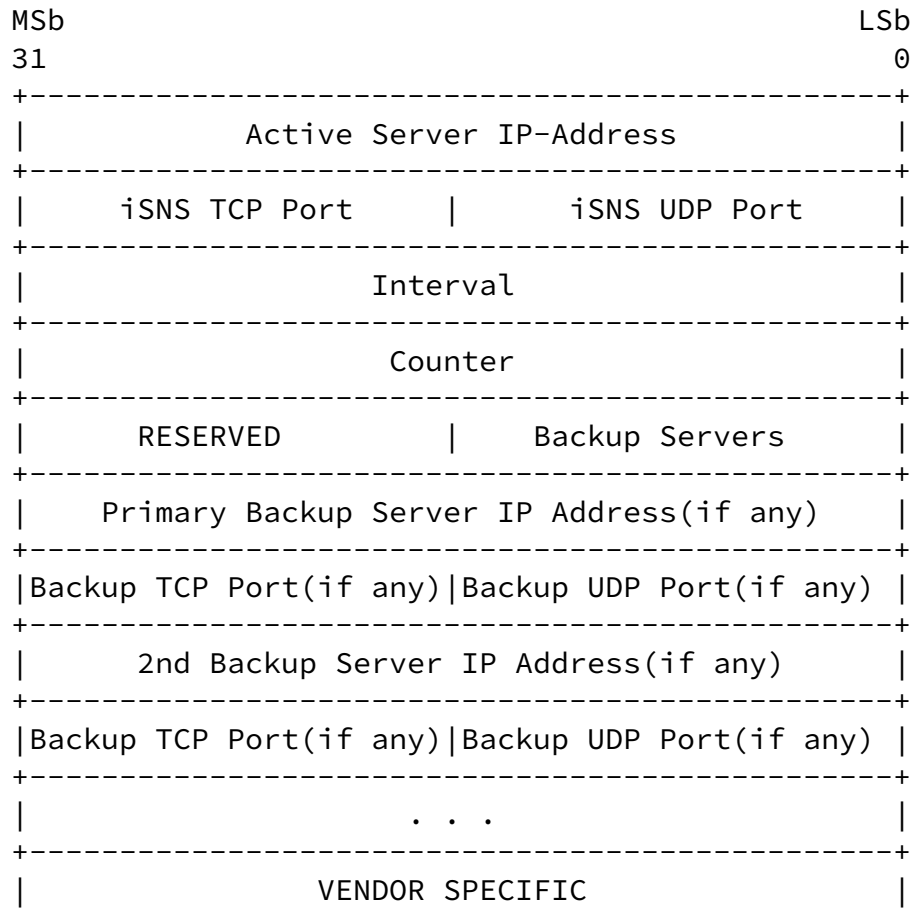
If the iSNS client portal fails to respond to three consecutive ESI

messages, then the iSNS SHALL remove that client portal from the iSNS database. If there are no other remaining ESI monitored portals for the associated entity, then the entity SHALL also be removed. The appropriate State Change Notifications, if any, SHALL be triggered.

7.6.5.14 Name Service Heartbeat (Heartbeat)

This message SHOULD only be sent by the active iSNS server. It allows iSNS clients and backup servers listening to the broadcast or multicast address to discover the IP address of the primary and backup iSNS servers. It also all concerned parties to monitor the health and status of the primary iSNS server.

This message is NOT in TLV format. There is no response message to the Name Service Heartbeat.



The heartbeat payload contains:

Active Server IP-Address: the IP_Address of the active iSNS server in IPv6 format.

Active TCP Port: the TCP Port of the server currently in use

Active UDP Port: the UDP Port of the server currently in use, otherwise 0

Interval: the interval, in seconds, of the heartbeat

Counter: a monotonically incrementing count of heartbeats sent

Backup Servers: the number of iSNS backup servers. The IP address, TCP Port, and UDP Port of each iSNS backup server follow this field. Note that if backup servers are used, then the active iSNS server SHOULD list be among the list of backup servers.

The content of the remainder of this message after the list of backup servers is vendor-specific. Vendors may use additional fields to coordinate between multiple iSNS servers, and/or to identify vendor specific features.

7.6.5.15 Request Switch ID (RqstSwId)

The RqstSwId message type is 0x0011. This message is used for iFCP Transparent Mode to allocate non-overlapping SWITCH_ID values

between 1 and 239. The iSNS server becomes the address assignment authority for the entire iFCP fabric. To obtain multiple SWITCH_ID values, this request must be repeated multiple times to the iSNS server.

The RqstSwId payload contains three TLV attributes in the following order: the requesting Switch Name (WWN) as the source attribute, the Space Identifier as the key attribute, and Preferred ID as the operating attribute. The Space Identifier is a string identifying the domain space for which the iSNS server shall allocate non-overlapping integer SWITCH_ID values between 1 and 239. The Preferred_ID is the nominal SWITCH_ID value requested by the iSNS client. If the Preferred_ID value is available and has not been already allocated for the Space_Identifier specified in the message,

the iSNS server shall return the requested Preferred_ID value as the Assigned_ID to the requesting client.

The RqstSwId response contains an Error Code, and the TLV attribute Assigned ID, which contains the integer value in the space requested. If no further unallocated values are available from this space, the iSNS server SHALL respond with the error code 18 "SWITCH_ID not available".

Once a SWITCH_ID value has been allocated to an iSNS client by the iSNS server for a given Space_Identifier, that SWITCH_ID value shall not be reused until it has been deallocated, or the ESI message detects that the iSNS client no longer exists on the network.

The iSNS server and client SHALL use TCP to transmit and receive RqstSwId, RqstSwIdRsp, RlseSwId, and RlseSwIdRsp messages.

7.6.5.16 Release Switch ID (RlseSwId)

The RlseSwId message type is 0x0012. This message may be used by iFCP Transparent Mode to release integer identifier values used to assign 3-byte Fibre Channel PORT_ID values.

The RlseSwId message contains three TLV attributes in the following order: the requesting entity EID as the source attribute, the Space_Identifier as the key attribute, and Assigned_ID as the operating attribute. Upon receiving the RlseSwId message, the iSNS server shall deallocate the SWITCH_ID value contained in the Assigned_ID attribute for the Space_Identifier attribute specified. Upon deallocation, that SWITCH_ID value can now be requested by, and assigned to, a different iSNS client.

The iSNS server and client SHALL use TCP to transmit and receive RqstSwId, RqstSwIdRsp, RlseSwId, and RlseSwIdRsp messages.

7.6.5.17 Get Switch IDs (GetSwIds)

The GetSwIds message type is 0x0013. This message is used to learn the currently-allocated SWITCH_ID values for a given Space_Identifier.

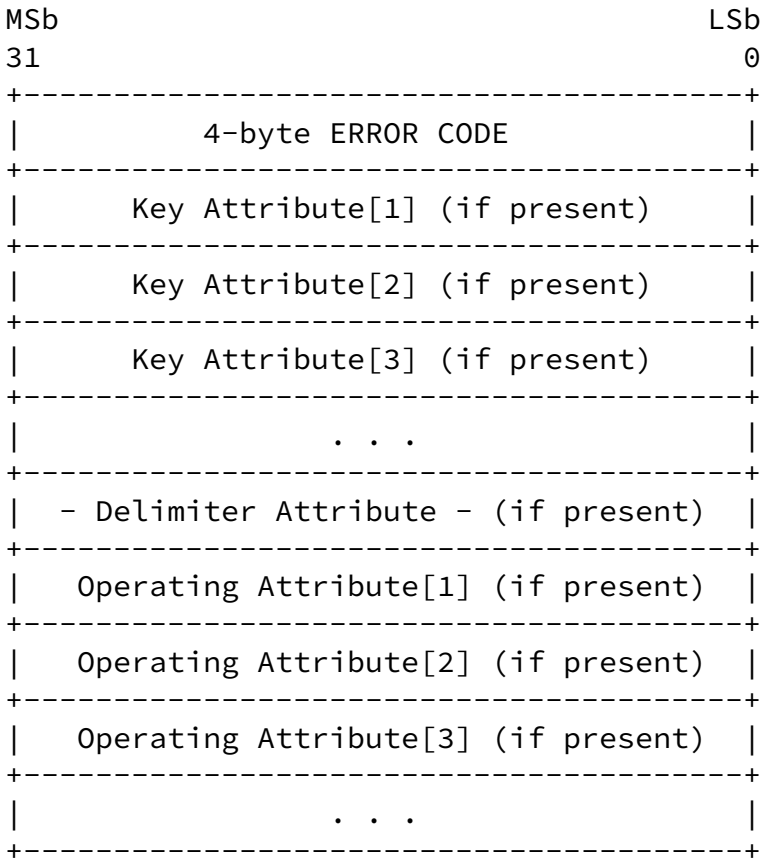
The GetSwIds message payload contains a Source Attribute and Key Attribute.

The Key Attribute for the GetSwIds message is the Space_Identifier.

The response to this message returns all of the SWITCH_ID values that have been allocated for the Space_Identifier specified.

7.7 Response Messages

The iSNSP response message payloads contain an Error Code, followed by a list of attributes, and have the following format:



The iSNS Response messages will be sent to the iSNS Client IP Address and the originating TCP/UDP Port that was used for the associated registration and query message.

7.7.1 Error Code

The first field in an iSNSP response message payload is the Error Code for the operation that was performed. The Error Code format is defined in section 7.4.

7.7.2 Key Attributes in Response

Depending on the specific iSNSP request, the response message will contain Key Attributes. For example, a Register Device Attribute Response message will contain the Key Attributes used in the Device Attribute Registration with the assigned values, if they were assigned by the iSNS.

7.7.3 Delimiter Attribute in Response

The Delimiter Attribute separates the key and operating attributes in a response message, if they exist. The Delimiter Attribute has a tag value of 0 and a length value of 0. The Delimiter Attribute is effectively 8 Bytes long, a 4 Byte tag containing 0x00000000, and a 4 Byte length field containing 0x00000000.

7.7.4 Operating Attributes in Response

The Operating Attributes in a response are the results related to the iSNS registration or query operation being performed.

The number of Operating Attributes in the response depends on the specific iSNSP request or query response. For example, the Operating Attributes in a Device Attribute Query Response message are the set of Operating Attributes from network entity entries that matched the Key Attributes in the associated Device Attribute Query message.

7.7.5 Registration and Query Message Types

The following describes each query and message type.

7.7.5.1 Register Device Attribute Response (RegDevRsp)

The RegDevRsp message type is 0x8001. The RegDevRsp message contains the results for the RegDevAttr message with the same TRANSACTION ID.

The Error Code contains the operation results. If the registration completed successfully the code of ôNo Errorö is returned. If an error occurred then the appropriate code will be returned.

The Key Attributes contain the set of keys for the objects registered by the Register Device Attribute message. If the iSNS assigned a unique Entity Identifier for a network entity, then the key attribute field shall contain the assigned Entity Identifier.

There are no Operating Attributes in the RegDevRsp message.

7.7.5.2 Device Attribute Query Response (DevAttrQryRsp)

The DevAttrQryRsp message type is 0x8002. The DevAttrQryRsp message

contains the results for the DevAttrQry message with the same TRANSACTION ID.

The Error Code contains the operation results. If the query completed successfully the code of ôNo Errorö is returned. If an error occurred then the appropriate code will be returned.

For a successful query result, the DevAttrQryRsp Operating Attributes will contain the results of the original DevAttrQry message.

7.7.5.3 Device Get Next Response (DevGetNextRsp)

The DevGetNextRsp message type is 0x8003. The DevGetNextRsp message contains the results for the DevGetNext message with the same TRANSACTION ID.

The Error Code contains the operation results. If the operation completed successfully the code of ôNo Errorö is returned. If an error occurred then the appropriate code will be returned.

The Key Attribute field contains the next key, in sequential order, after the Key Attribute used in the DevGetNext message.

The Operating Attribute field contains the same attributes as in the DevGetNext message. The values of the Operating Attributes are the attribute values associated with the key returned.

7.7.5.4 Deregister Device Response (DeregDevRsp)

The DeregDevRsp message type is 0x8004. If the DeregDe operation completed successfully then the code of ôNo Errorö is returned. If an error occurred then the appropriate code will be returned.

The DeregDevRsp message does not contain any key or operating attributes.

7.7.5.5 SCN Register Response (SCNRegRsp)

The SCNRegRsp message type is 0x8005. If the SCReg operation completed successfully then the code of ôNo Errorö is returned. If an error occurred then the appropriate code will be returned.

The SCNRegRsp message does not contain any key or operating attributes.

7.7.5.6 SCN Deregister Response (SCNDeregRsp)

The SCNDeregRsp message type is 0x8006. If the SCNDereg operation completed successfully then the code of ôNo Errorö is returned. If an error occurred then the appropriate code will be returned.

The SCNDeregRsp message does not contain any key or operating attributes.

7.7.5.7 SCN Event Response (SCNEventRsp)

The SCNEventRsp message type is 0x8007. If the SCNEvent operation completed successfully then the Error Code of ôNo Errorö is returned. If an error occurred then the appropriate code will be returned.

The SCNEventRsp message does not contain any key or operating attributes.

7.7.5.8 SCN Response (SCNRsp)

The SCNRsp message type is 0x8008. This message is sent by an iSNS client, and provides confirmation that the SCN message was received and processed.

If the SCN operation completed successfully, then the Error Code of ôNo Errorö is returned by the iSNS client. If an error occurred then the appropriate code will be returned.

The SCNRsp response message payload also contains the SCN Destination Attribute representing the node or entity identifier that received the SCN.

7.7.5.9 DD Register Response (DDRegRsp)

The DDRegRsp message type is 0x8009. If the DDReg operation completed successfully then the code of ôNo Errorö is returned. If an error occurred then the appropriate code will be returned.

If successful, the DD ID of the DD created or updated during the DDReg operation will be returned as an operating attribute of the message.

7.7.5.10 DD Deregister Response (DDDeregRsp)

The DDDeregRsp message type is 0x800A. If the DDDereg operation completed successfully then the code of ôNo Errorö is returned. If an error occurred then the appropriate code will be returned.

The DDDeregRsp message does not contain any key or operating attributes.

7.7.5.11 DDS Register Response (DDSRegRsp)

The DDSRegRsp message type is 0x800B. If the DDSRegRsp operation completed successfully then the code of ôNo Errorö is returned. If an error occurred then the appropriate code will be returned.

If successful, the DDS ID of the DDS created or updated during the DDSReg operation will be returned as an operating attribute of the message.

7.7.5.12 DDS Deregister Response (DDSDeregRsp)

The DDSDeregRsp message type is 0x800C. If the DDSDeregRsp operation completed successfully then the code of ôNo Errorö is returned. If an error occurred then the appropriate code will be returned.

The DDSDeregRsp message does not contain any key or operating attributes.

7.7.5.13 Entity Status Inquiry Response (ESIRsp)

The ESIRsp message type is 0x800D. This message is sent by an iSNS client, and provides confirmation that the ESI message was received and processed.

The ESIRsp response message payload contains the attributes from the original ESI message. These attributes represent the iSNS client portal that is responding to the ESI. The ESIRsp Attributes are in the order they were provided in the original ESI message. An error code of "No Error" is returned.

Upon receiving the ESIRsp from the iSNS client, the iSNS server SHALL update the timestamp attribute for that client entity and

portal.

7.7.5.14 Request Switch ID Response (RqstSwIdRsp)

The RqstSwIdRsp message type is 0x8011. This message provides the response for RqstSwId.

The RqstSwId response contains an Error Code and the TLV attribute Assigned ID, which contains the integer value in the space requested. If no further unallocated values are available from this space, the iSNS server SHALL respond with the error code 19 "SWITCH_ID not available".

Once a SWITCH_ID value is allocated by the iSNS server, it shall not be reused until it has been deallocated by the iSNS client to which the value was assigned, or the ESI message detects that the iSNS client no longer exists on the network.

The iSNS server and client SHALL use TCP to transmit and receive RqstSwId, RqstSwIdRsp, RlseSwId, and RlseSwIdRsp messages.

7.7.5.15 Release Switch ID Response (RlseSwIdRsp)

The RlseSwIdRsp message type is 0x8012. This message provides the response for RlseSwId. The response contains an Error indicating if the request was successful or not. If the Assigned_ID value in the original RlseSwId message is not allocated, then the iSNS server SHALL respond with this message using the error code 20 ôSWITCH_ID not allocatedö.

The iSNS server and client SHALL use TCP to transmit and receive RqstSwId, RqstSwIdRsp, RlseSwId, and RlseSwIdRsp messages.

7.7.5.16 Get Switch IDs Response (GetSwIdRsp)

The GetSwIdsResp message type is 0x8013. This message is used determine which SWITCH_ID values have been allocated for the Space_Identifier specified in the original GetSwId request message.

The GetSwIds response message payload contains an error code indicating if the request was successful, and a list of the Assigned IDs from the space requested. The Assigned_ID attributes are listed in TLV format.

7.8 Vendor Specific Messages

Vendor-specific iSNSP messages have a functional ID of between 0x0100 and 0x01FF, while vendor-specific responses have a functional ID of between 0x8100 and 0x81FF. The first key attribute in a vendor-specific message SHALL be the company OUI (tag=256) identifying original creator of the proprietary iSNSP message. The contents of the remainder of the message are vendor-specific.

8. Security Considerations

8.1 iSNS Security Threat Analysis

When the iSNS protocol is deployed, the interaction between iSNS server and iSNS clients are subject to the following security threats:

[1] An attacker could alter iSNS protocol messages, such as to direct iSCSI and iFCP devices to establish connections with rogue peer devices, or to weaken/eliminate IPSec protection for iSCSI or iFCP traffic.

[2] An attacker could masquerade as the real iSNS server using false iSNS heartbeat messages. This could cause iSCSI and iFCP devices to use rogue iSNS servers.

[3] An attacker could gain knowledge about iSCSI and iFCP devices by snooping iSNS protocol messages. Such information could aid an attacker in mounting a direct attack on iSCSI and iFCP devices, such as a denial-of-service attack or outright physical theft.

To address these threats, iSNS protocol messages need to have authentication support.

8.2 iSNS Security Implementation Requirements

Since iSNS is used to distribute authorizations for communications between iFCP and iSCSI peer devices, and can be used to distribute security policy for iFCP and iSCSI, IPsec ESP with null transform MUST be implemented. Where confidentiality is desired, IPsec ESP with non-null transform MAY be used.

In order to protect against an attacker masquerading as an iSNS server, client devices MUST be able to authenticate broadcast or multicast messages such as the iSNS heartbeat. The iSNS authentication block (which is identical in format to the SLP authentication block) MAY be used for this purpose. Note that the authentication block is used only for iSNS broadcast or multicast messages, and SHOULD NOT be used in unicast iSNS messages.

There is no requirement that the communicating identities in iSNS protocol messages be kept confidential. Specifically, the identity and location of the iSNS server shall not be considered confidential. However, in order to protect against an attacker masquerading as the real iSNS server when multicast or broadcast messages are used, the iSNS server MAY have the capability to allow client devices to authenticate broadcast or multicast messages. The iSNS authentication block (which is identical in format to the SLP authentication block) may be used for this purpose. Note that the authentication block is used only for iSNS broadcast or multicast messages, and SHOULD NOT be used in unicast iSNS messages.

For protecting unicast iSNS protocol messages, iSNS servers supporting security MUST implement ESP in tunnel mode and MUST conform to [RFC2401] requirements for support of ESP in transport mode. Section 4.1 of [RFC2401] states:

a) A host MUST support both transport and tunnel mode.

b) A security gateway is required to support only tunnel mode. If it supports transport mode, that should be used only when the security gateway is acting as a host, e.g., for network management.

All iSNS implementations supporting security MUST support the replay protection mechanisms of IPsec.

Conformant iSNS security implementations MUST support IKE for peer authentication, negotiation of security associations, and key management, using the IPSec DOI [RFC2407]. Manual keying SHOULD NOT be used since it does not provide the necessary rekeying support. Conformant iSNS security implementations MUST support peer authentication using a pre-shared key, and MAY support certificate-based peer authentication using digital signatures. Peer authentication using the public key encryption methods outlined in IKE's sections 5.2 and 5.3 [RFC2409] SHOULD NOT be supported.

When pre-shared keys are used for authentication, IKE Aggressive Mode SHOULD be used, and the IKE Main Mode SHOULD NOT be used. When digital signatures are used for authentication, either IKE Main Mode or IKE Aggressive Mode MAY be used. In all cases, access to locally stored secret information (pre-shared key or private key for digital signing) MUST be suitably restricted, since compromise of the secret information nullifies the security properties of the IKE/IPSec

protocols.

When digital signatures are used to achieve authentiation, an IKE negotiator SHOULD use IKE Certificate Request Payload(s) to specify the certificate authority (or authorities) that are trusted in accordance with its local policy. IKE negotiators SHOULD check the pertinent Certificate Revocation List (CRL) before accepting a PKI certificate for use in IKE's authentication procedures.

8.3 Using iSNS to Discover Security Requirements of Peer Devices

The iSNS protocol is used to transfer naming, discovery, and management information between iSCSI devices, iFCP gateways, management stations, and the iSNS server. Once communication between iSNS clients and the iSNS server have been secured through use of IPSec, the iSNS client devices have the capability to discover the security settings that they need to use for their peer-to-peer communications using the iSCSI and/or iFCP protocols. This provides a potential scaling advantage over device-by-device configuration of individual security policies for each iSCSI and iFCP device.

The iSNS server stores security settings for each iSCSI and iFCP device interface. These security settings, which can be retrieved by authorized hosts, include use or non-use of IPSec, IKE, Main Mode, Aggressive Mode, PFS, and certificates. For example, IKE may not be enabled for a particular interface of a peer device. If a peer device can learn of this in advance by consulting the iSNS server, it will not need to waste time and resources attempting to initiate an IKE phase 1 session with that peer device interface.

If iSNS is used for this purpose, then the minimum information that should be learned from the iSNS server is the use or non-use of IKE and IPSec by each iFCP or iSCSI peer device interface. This information is encoded in the Security Bitmap field of each Portal of the peer device, and is applicable on a per-interface basis for the peer device. iSNS queries to acquire security configuration data about peer devices MUST be protected by IPSec/ESP authentication.

8.4 Using iSNS to Configure Security Policies of Client Devices

The iSNS server can store policies that are used for ISAKMP phase 1

and phase 2 negotiations between client devices. The ISAKMP payload format includes a series of one or more proposals that the iSCSI or iFCP device will use when negotiating the appropriate IPSec policy to use to protect iSCSI or iFCP traffic. For further details on how to store and retrieve ISAKMP policy proposals in the iSNS server, see [58].

8.5 Resource Issues

The iSNS protocol is lightweight, and will not generate a significant amount of traffic. iSNS traffic is characterized by occassional registration, notification, and update messages that do not consume measurable amounts of bandwidth. Even software-based IPSec implementations should not have a problem handling the traffic loads generated by iSNS.

To fulfill iSNS security requirements, the only additional resources needed beyond what is already required for iSCSI and iFCP involves the iSNS server. Since iSCSI and iFCP end nodes are already

required to implement IKE and IPSec, these existing requirements can also be used to fulfill IKE and IPSec requirements for iSNS clients.

8.6 iSNS Interaction with IKE and IPSec

When IPSec security is enabled, each iSNS client that is registered in the iSNS database SHALL maintain at least one phase-1 and one phase-2 security association with the iSNS server. All iSNS protocol messages between iSNS clients and the iSNS server SHALL be protected by a phase-2 security association.

When an iSNS client is removed from the iSNS database, the iSNS server shall send a phase-1 delete message to the associated IKE peer, and tear down all phase-1 and phase-2 SA's associated with that iSNS client.

9. Normative References

[iSCSI] Satran, J., et al., "iSCSI", Internet draft (work in progress), draft-ietf-ips-iSCSI-09.txt, November 2001

[iFCP] Monia, C., et al., "iFCP - A Protocol for Internet Fibre Channel Storage Networking", Internet draft (work in progress), draft-ietf-ips-ifcp-07.txt, November 2001

[RFC2608] Guttman, E., Perkins, C., Veizades, J., Day, M., "Service Location Protocol, Version 2", RFC 2608, June 1999

[iSCSIName] Bakke, M., et al., "iSCSI naming and Discovery", draft-

ietf-ips-iscsi-name-disc-03.txt, November 2001

[iSCSI-SLP] Bakke, M., "Finding iSCSI Targets and Name Servers Using SLP", Internet draft (work in progress), draft-ietf-ips-iscsi-slp-01.txt, July 2001

[RFC2119] Bradner, S., "Key Words for Use in RFCs to Indicate Requirement Levels, BCP 14, RFC 2119, March 1997

[SEC-IPS] Aboba, B., et al., "Securing IP Block Storage Protocols", draft-ietf-ips-security-07.txt, December 2001

[RFC2401] Atkinson, R., Kent, S., "Security Architecture for the Internet Protocol", RFC 2401, November 1998

[RFC2406] Kent, S., Atkinson, R., "IP Encapsulating Security Payload (ESP)", RFC 2406, November 1998

[RFC2407] Piper, D., "The Internet IP Security Domain of Interpretation of ISAKMP", RFC 2407, November 1998

[RFC2408] Maughan, D., Schertler, M., Schneider, M., Turner, J., "Internet Security Association and Key Management Protocol (ISAKMP), RFC 2408, November 1998

[RFC2409] Harkins, D., Carrel, D., "The Internet Key Exchange (IKE)", RFC 2409, November 1998

[RFC2412] Orman, H., "The OAKLEY Key Determination Protocol", RFC 2412, November 1998

[RFC793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, September 1981

[DSS] FIPS PUB 186-2, National Institute of Standards and Technology, Digital Signature Standard (DSS), Technical Report

[EUI-64] Guidelines for 64-bit Global Identifier (EUI-64) Registration Authority, May 2001, IEEE, http://standards.ieee.org/regauth/oui/tutorials/EUI64.html

[802-1990] IEEE Standards for Local and Metropolitan Area Networks: Overview and Architecture, Technical Committee on Computer Communications of the IEEE Computer Society, May 31, 1990

[FC-FS] Fibre Channel Framing and Signaling Interface, NCITS Working Draft Project 1331-D

10. Informative References

[RFC1035] Mockapetris, P., "Domain Names - Implementation and Specification, RFC 1035, November 1987

[RFC1305] Mills, D., Network Time Protocol (Version 3), RFC 1305, March 1992

[FC-GS] Fibre Channel Generic Services, ANSI X3.288:1996

[FC-GS-2] Fibre Channel Generic Services-2, ANSI NCITS 288

[FC-GS-3] Fibre Channel Generic Services-3, NCITS 348-2000

[FC-GS-4] Fibre Channel Generic Services-4, NCITS Working Draft Project 1505-D

[RFC2026] Bradner, S., "The Internet Standards Process -- Revision 3", BCP 9, RFC 2026, October 1996.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997

11. Author's Addresses

Josh Tseng
Kevin Gibbons
Charles Monia
Nishan Systems
3850 North First Street
San Jose, CA 95134-1702
Phone: (408) 519-3749
Email: jtseng@nishansystems.com

Franco Travostino
Nortel Networks
3 Federal Street
Billerica, MA 01821
Phone: 978-288-7708
Email: travos@nortelnetworks.com

Tom McSweeney
Curt Du Laney
John Dowdy
IBM
4205 South Miami Blvd
Research Triangle Park, NC 27709
Email: jdowdy@us.ibm.com
Phone: (919) 254-5632

Chad Gregory
505 E. Huntland Drive, Suite 550
Austin, TX 78752
Email: chad.gregory@intel.com
Phone: (512) 407-2137

Full Copyright Statement

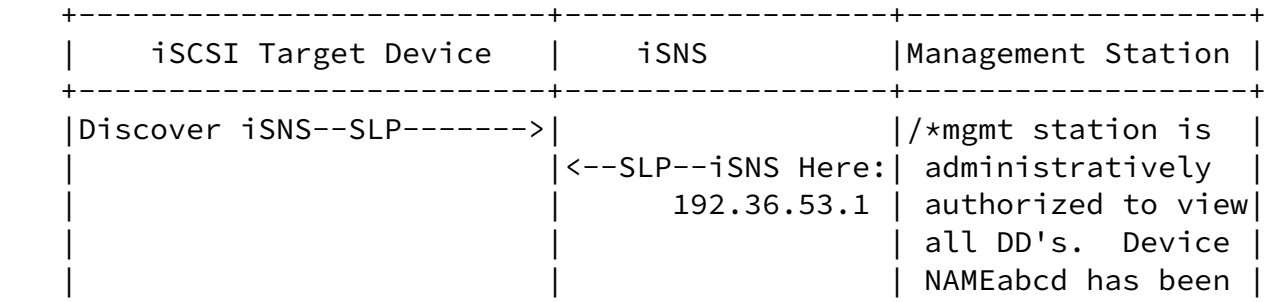
Appendix A -- iSNS Examples

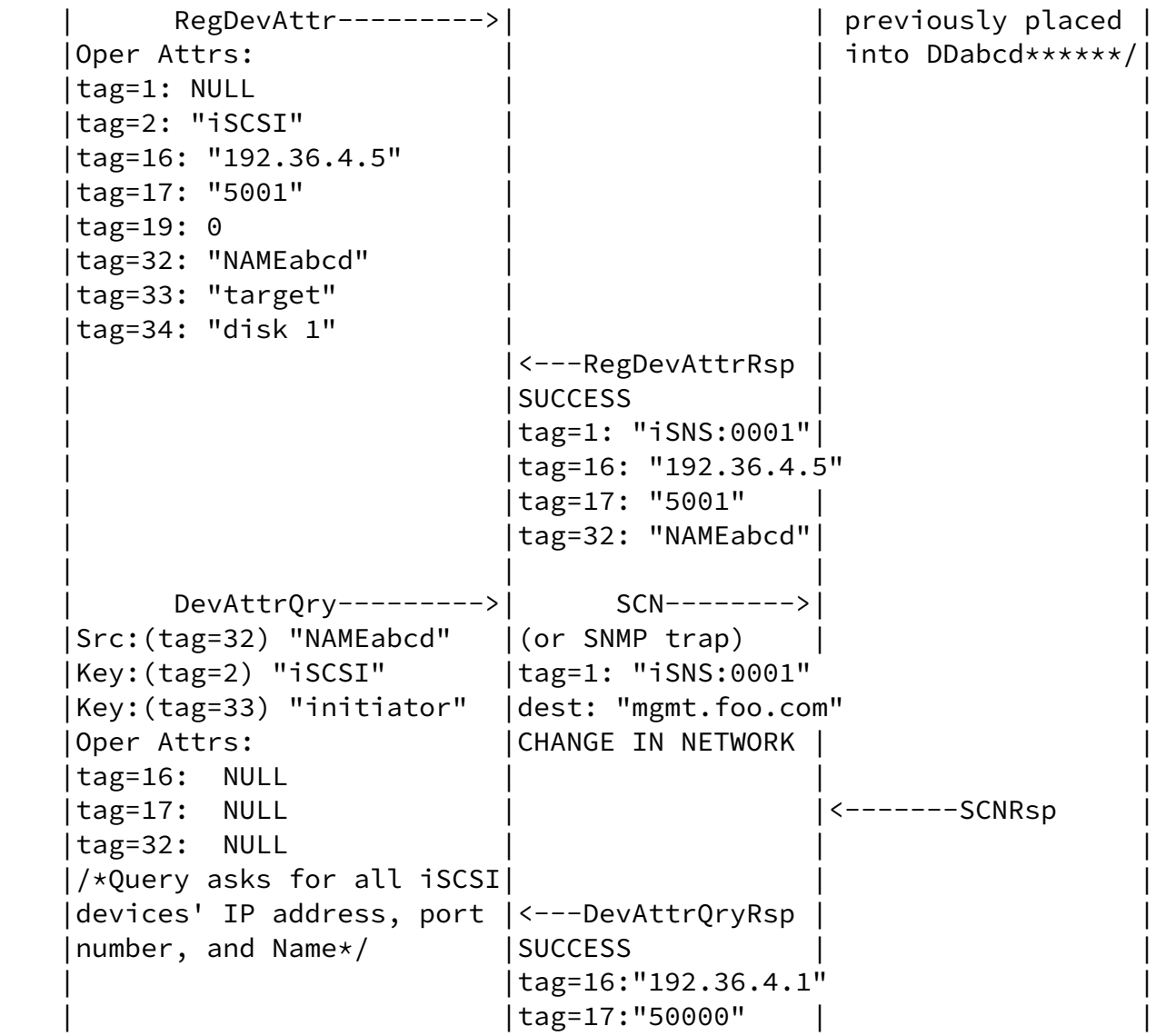
A.1 iSCSI Initialization Example

This example assumes an SLP Service Agent (SA) has been implemented on the iSNS host, and an SLP User Agent (UA) has been implemented on the iSNS initiator. See [RFC2608] for further details on SA's and UA's. This example also assumes the target is configured to use the iSNS, and have its access control policy subordinated to the iSNS.

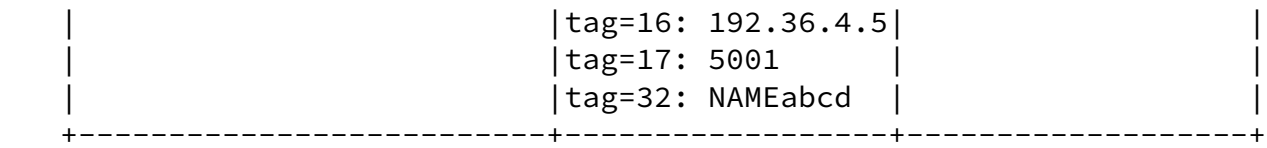
A.1.1 Simple iSCSI Target Registration

In this example, a simple target with a single iSCSI name registers with the iSNS. The target has not been assigned a Fully Qualified Domain Name (FQDN) by the administrator.



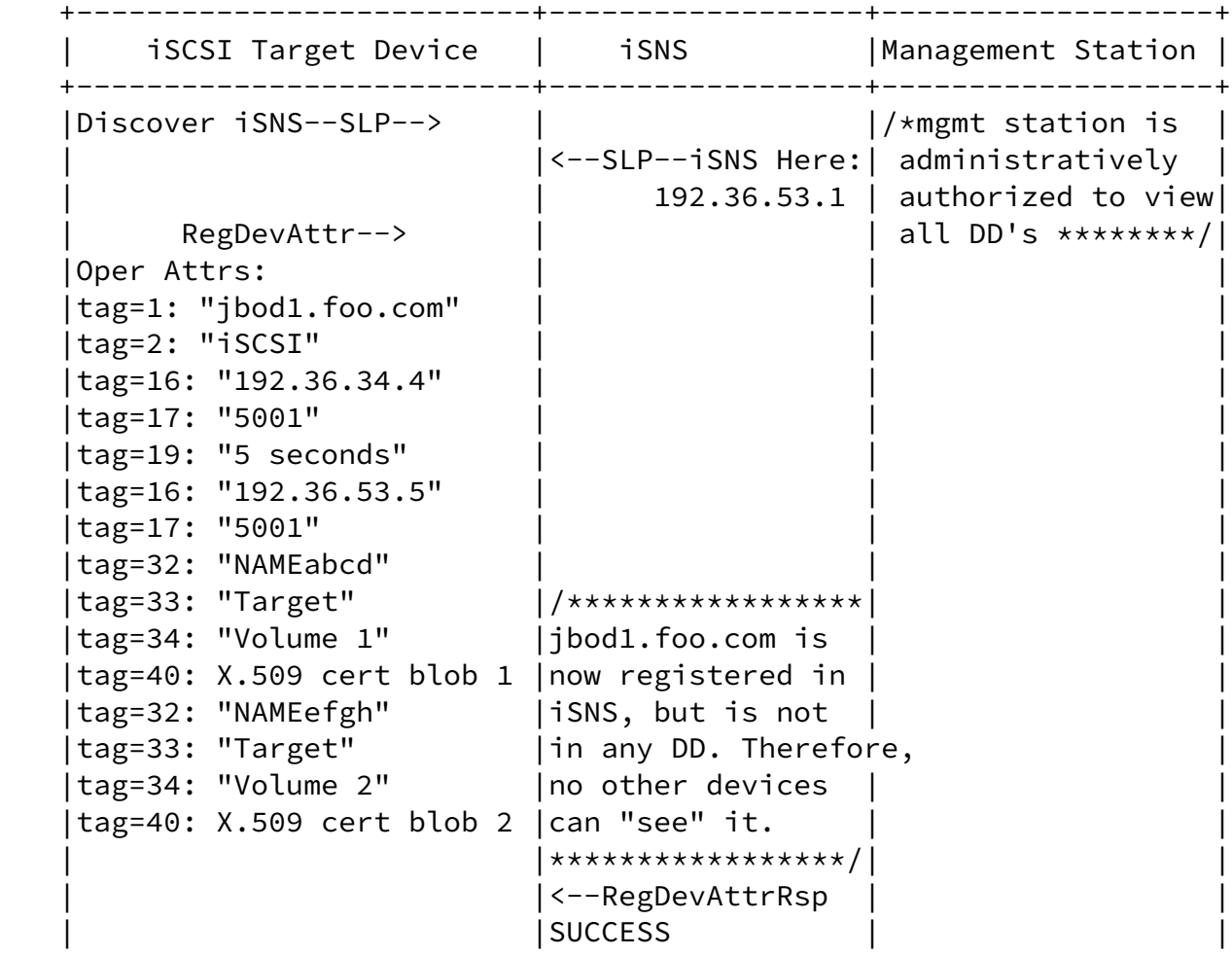

RegDevAttr---------->
Oper Attrs:
tag=1: NULL
tag=2: "iSCSI"
tag=16: "192.36.4.5"
tag=17: "5001"
tag=19: 0
tag=32: "NAMEabcd"
tag=33: "target"
tag=34: "disk 1"
previously placed into DDabcd******/
<---RegDevAttrRsp
SUCCESS
tag=1: "iSNS:0001"
tag=16: "192.36.4.5"
tag=17: "5001"
tag=32: "NAMEabcd"
DevAttrQry---------->
Src:(tag=32) "NAMEabcd"
Key:(tag=2) "iSCSI"
Key:(tag=33) "initiator"
Oper Attrs:
tag=16: NULL
tag=17: NULL
tag=32: NULL
/*Query asks for all iSCSI devices' IP address, port number, and Name*/
SCN-------->
(or SNMP trap)
tag=1: "iSNS:0001"
dest: "mgmt.foo.com"
CHANGE IN NETWORK
<-------SCNRsp
<---DevAttrQryRsp
SUCCESS
tag=16:"192.36.4.1"
tag=17:"50000"

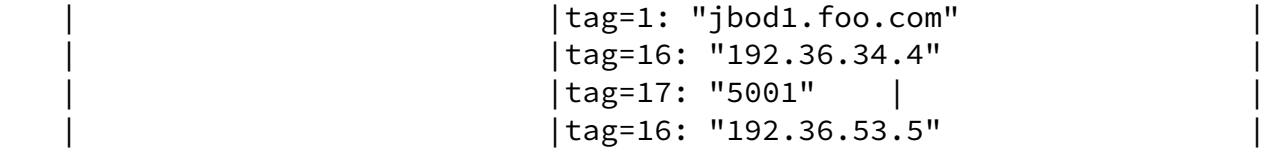

tag=32:"devpdq"
tag=16:"192.1.3.2"
tag=17:"50000"
tag=32:"devrst"
<-----DevAttrQry
src: ôMGMTname1ö
key:(tag=1)iSNS:0001
Op Attrs:
tag=16: NULL
tag=17: NULL
tag=32: NULL
/************************
Our target "iSNS:0001" discovers two initiators in the same DD. It will accept iSCSI logins from these two identified initiators presented by iSNS*********************/
DevAttrQryRsp--->
SUCCESS

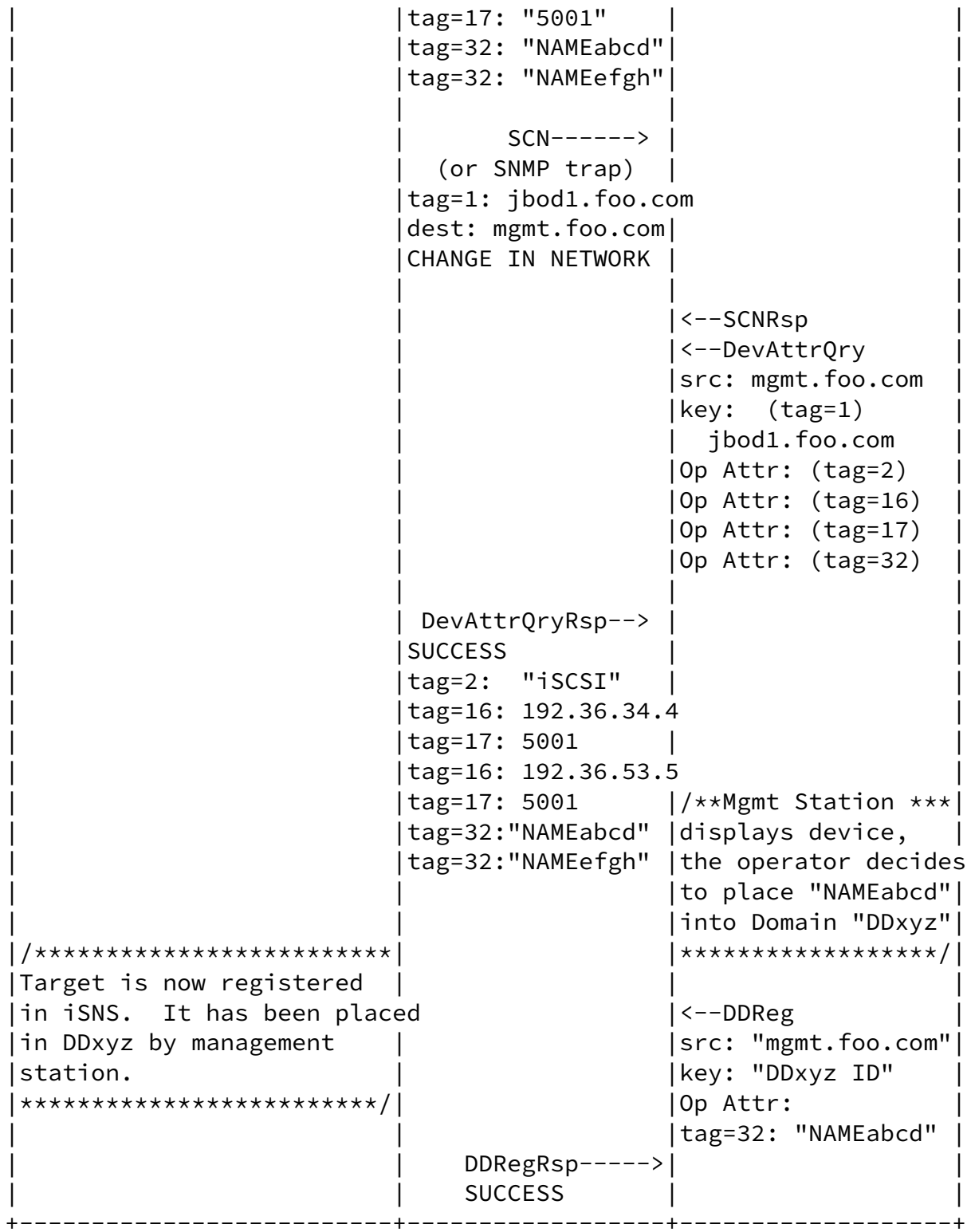


A.1.2 Target Registration and DD Configuration

In this example, a more complex target registers with the iSNS. This target has been configured with a Fully Qualified Domain Name (FQDN) in the DNS servers, and the user wishes to use this identifier for the device. Also, the user wishes to use public key certificates in the iSCSI login authentication.

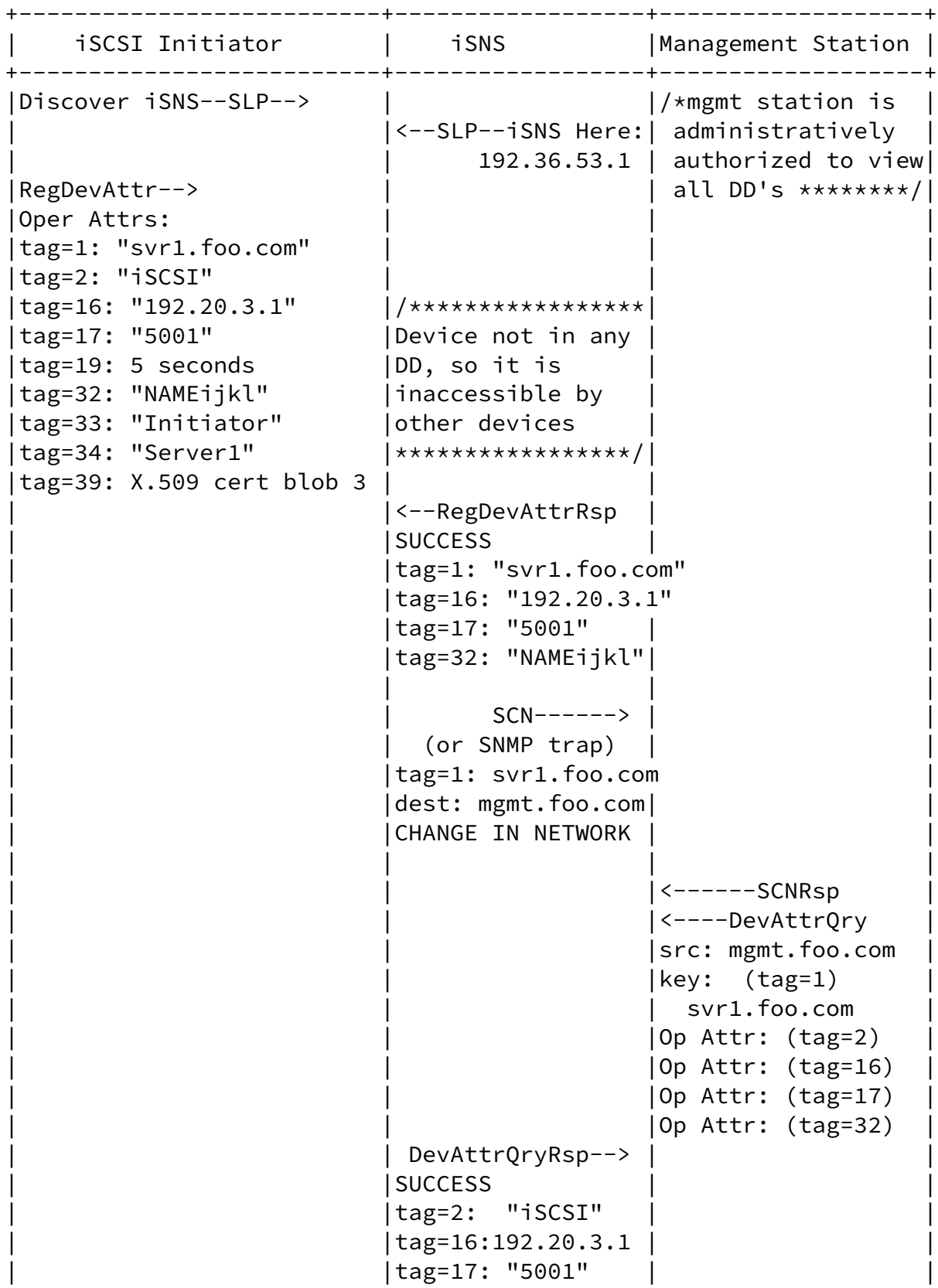


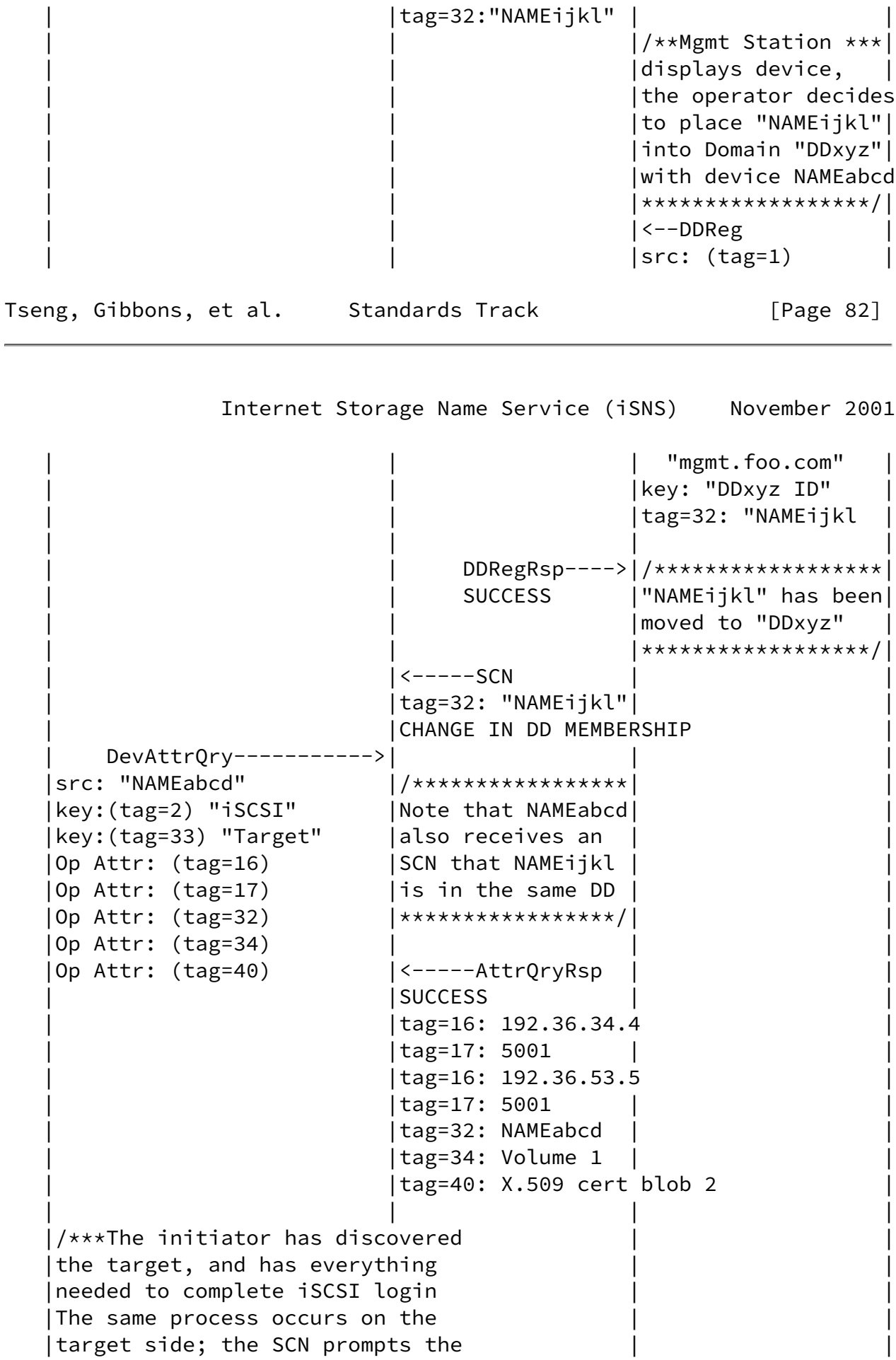


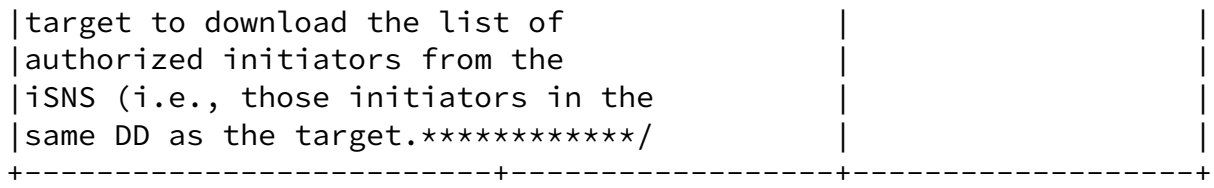


A.1.3 Initiator Registration and Target Discovery

The following example illustrates a new initiator registering with the iSNS, and discovering the target NAMEabcd from the example in A.1.2.


iSCSI Initiator
iSNS
Management Station
Discover iSNS--SLP-->
<--SLP--iSNS Here: 192.36.53.1
/*mgmt station is administratively authorized to view all DD's ********/
RegDevAttr-->
Oper Attrs:
tag=1: "svr1.foo.com"
tag=2: "iSCSI"
tag=16: "192.20.3.1"
tag=17: "5001"
tag=19: 5 seconds
tag=32: "NAMEijkl"
tag=33: "Initiator"
tag=34: "Server1"
tag=39: X.509 cert blob 3
/*****************
Device not in any DD, so it is inaccessible by other devices
*****************/
<--RegDevAttrRsp
SUCCESS
tag=1: "svr1.foo.com"
tag=16: "192.20.3.1"
tag=17: "5001"
tag=32: "NAMEijkl"
SCN------>
(or SNMP trap)
tag=1: svr1.foo.com
dest: mgmt.foo.com
CHANGE IN NETWORK
<------SCNRsp
<----DevAttrQry
src: mgmt.foo.com
key: (tag=1) svr1.foo.com
Op Attr: (tag=2)
Op Attr: (tag=16)
Op Attr: (tag=17)
Op Attr: (tag=32)
DevAttrQryRsp-->
SUCCESS
tag=2: "iSCSI"
tag=16:192.20.3.1
tag=17: "5001"


tag=32:"NAMEijkl"
/**Mgmt Station ***
displays device,
the operator decides
to place "NAMEijkl"
into Domain "DDxyz"
with device NAMEabcd
**********/
<--DDReg
src: (tag=1)
"mgmt.foo.com"
key: "DDxyz ID"
tag=32: "NAMEijkl
DDRegRsp---->
SUCCESS
/**********
"NAMEijkl" has been
moved to "DDxyz"
**********/
<-----SCN
tag=32: "NAMEijkl"
CHANGE IN DD MEMBERSHIP
DevAttrQry---------->
src: "NAMEabcd"
key:(tag=2) "iSCSI"
key:(tag=33) "Target"
Op Attr: (tag=16)
Op Attr: (tag=17)
Op Attr: (tag=32)
Op Attr: (tag=34)
Op Attr: (tag=40)
/**********
Note that NAMEabcd
also receives an
SCN that NAMEijkl
is in the same DD
**********/
<-----AttrQryRsp
SUCCESS
tag=16: 192.36.34.4
tag=17: 5001
tag=16: 192.36.53.5
tag=17: 5001
tag=32: NAMEabcd
tag=34: Volume 1
tag=40: X.509 cert blob 2
/***The initiator has discovered
the target, and has everything
needed to complete iSCSI login
The same process occurs on the
target side; the SCN prompts the


|target to download the list of
|authorized initiators from the
|iSNS (i.e., those initiators in the
|same DD as the target.***********/