iSCSI (Internet SCSI) 基礎と実際

藤田 智成

fujita.tomonori@lab.ntt.co.jp

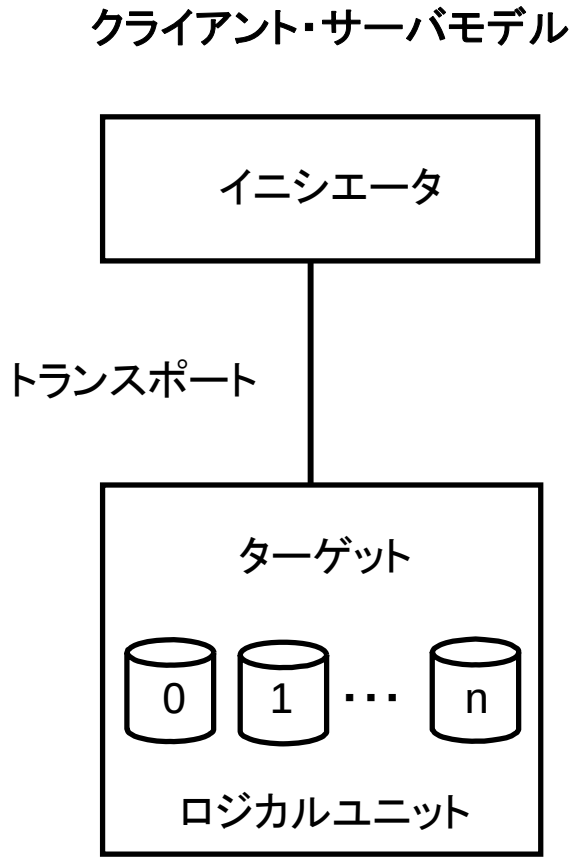
NTTサイバーソリューション研究所

SACSIS2006

The fundamentals: SCSI

SCSI

- イニシエータ
  - SCSIコマンドを送る
  - 通常の場合計算機
- トランスポート
  - イニシエータとターゲットを接続するための媒体
- ターゲット
  - SCSIコマンドを実行
  - ディスクアレイ、テープライブラリ等
- ロジカルユニット
  - ディスクドライブ、テープドライブ等



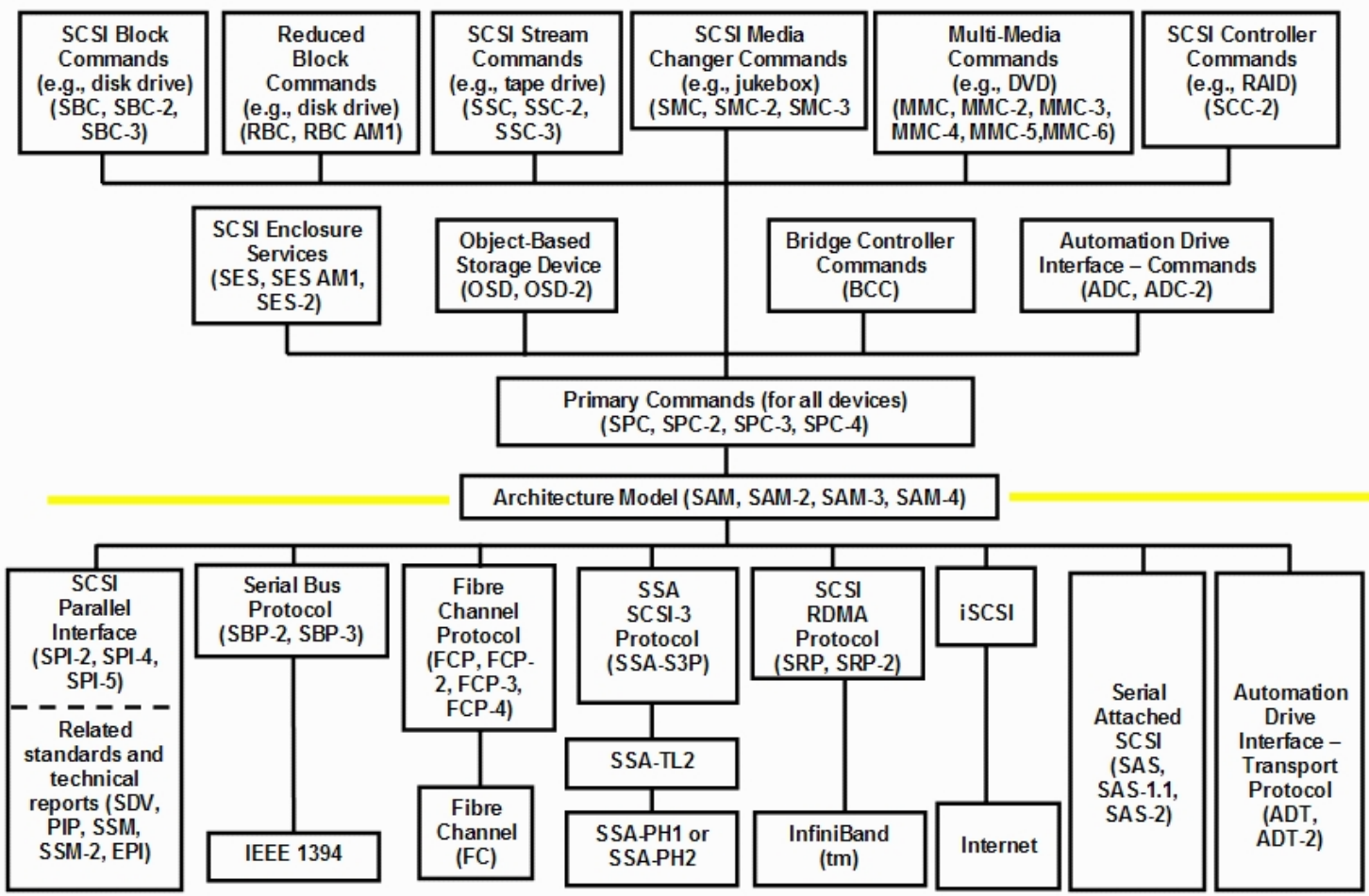
SCSI-1/SCSI-2

- トランスポートにパラレルインターフェイスだけを想定していた
  - 長さに制限あり
  - 高速化が難しい

SCSI-3

- SCSI規格が分割された
  - デバイス依存コマンド（ディスク、テープ等）
  - 共通コマンド（デバイスの種類に非依存）
  - トランスポート
- パラレルインターフェイスからの解放
  - 先進的なインターフェイス技術の採用

SCSI-3 Architecture



http://www.t10.org/scsi-3.htm

Storage meets Network

Storage Area Network (SAN)

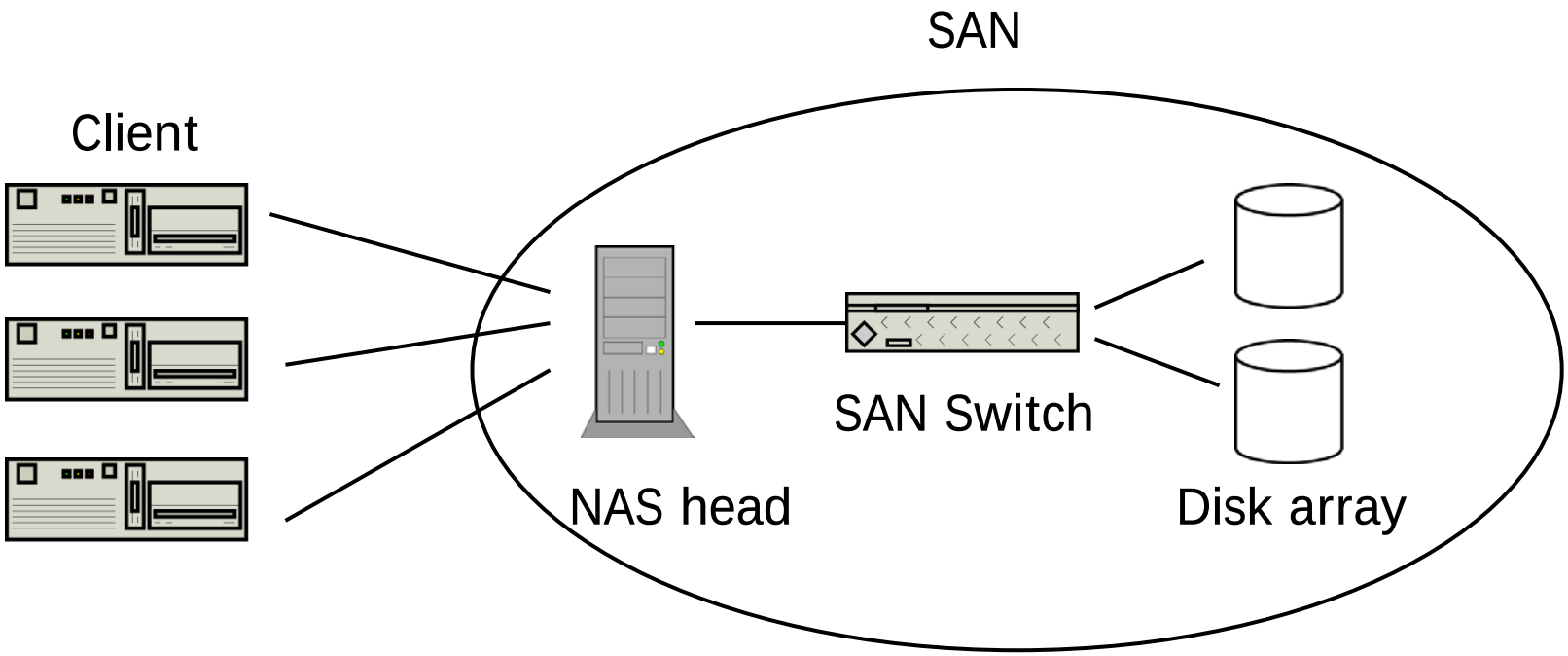
- パラレルケーブルの代わりに高速なインターコネクト技術を使う
- 代表的なインターフェイス
  - Fibre Channel Protocol (FCP)
    - Fibre Channel
  - Internet SCSI (iSCSI)
    - Ethernet, iSCSI HBA
  - SCSI RDMA Protocol (SRP)
    - RDMA-capable interface (Infiniband, RDMA NIC)

Network Attached Stoarge: NFS, CIFS, etc

- NAS
  - ファイルレベルのプロトコル
    - ファイル作成、ディレクトリ作成、ファイル読み取り、etc
  - クライアントは特定のファイルシステムを使う
- SAN
  - ブロックレベルのプロトコル
    - セクタ番号0番から16セクタ読む、etc
  - クライアントは、従来のファイルシステム、データベースをそのまま使う

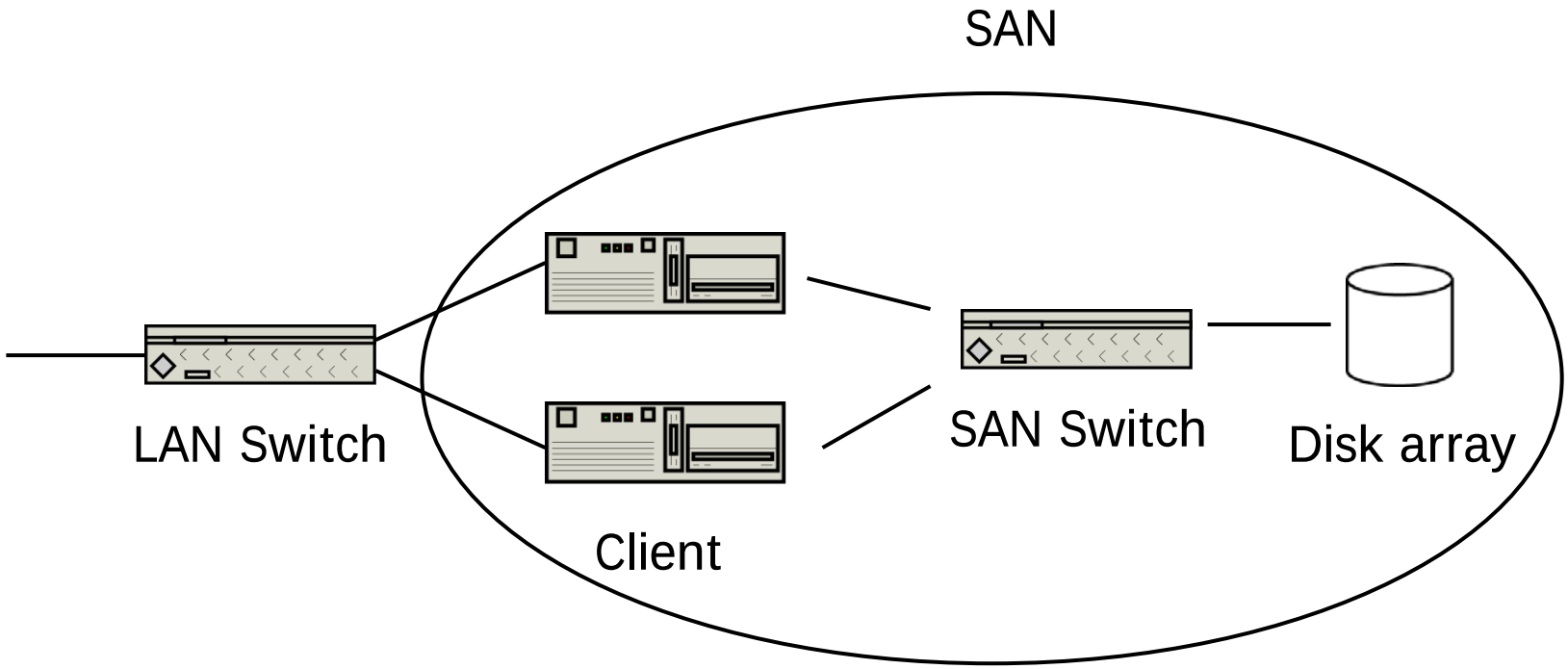
SAN利用例1: NAS head

- NASサーバのディスクとしてSANを使う



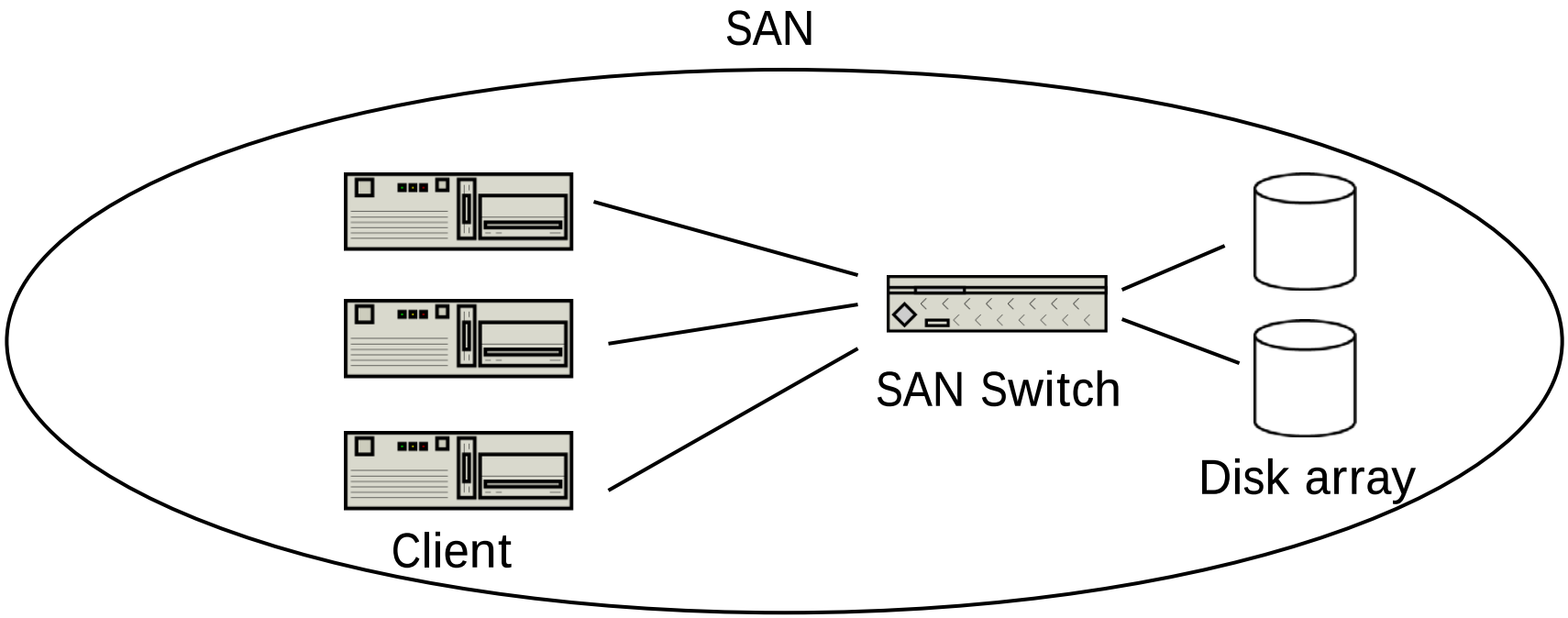
SAN利用例2: High Availability

- 複数のホストで一つのDisk arrayを共有
  - 稼動系･待機系



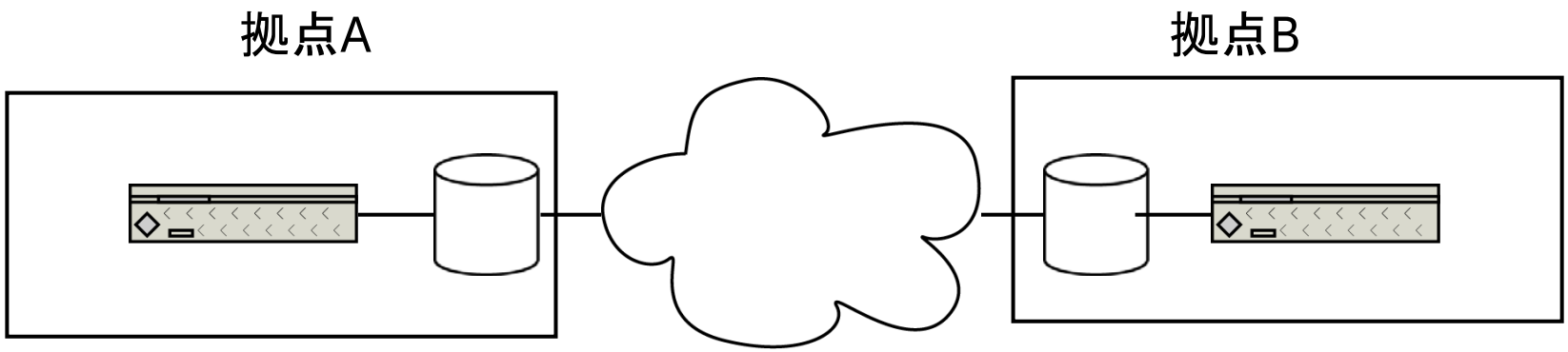
SAN利用例3: Storage Pool

- クライアントは直接SANのディスクを使う



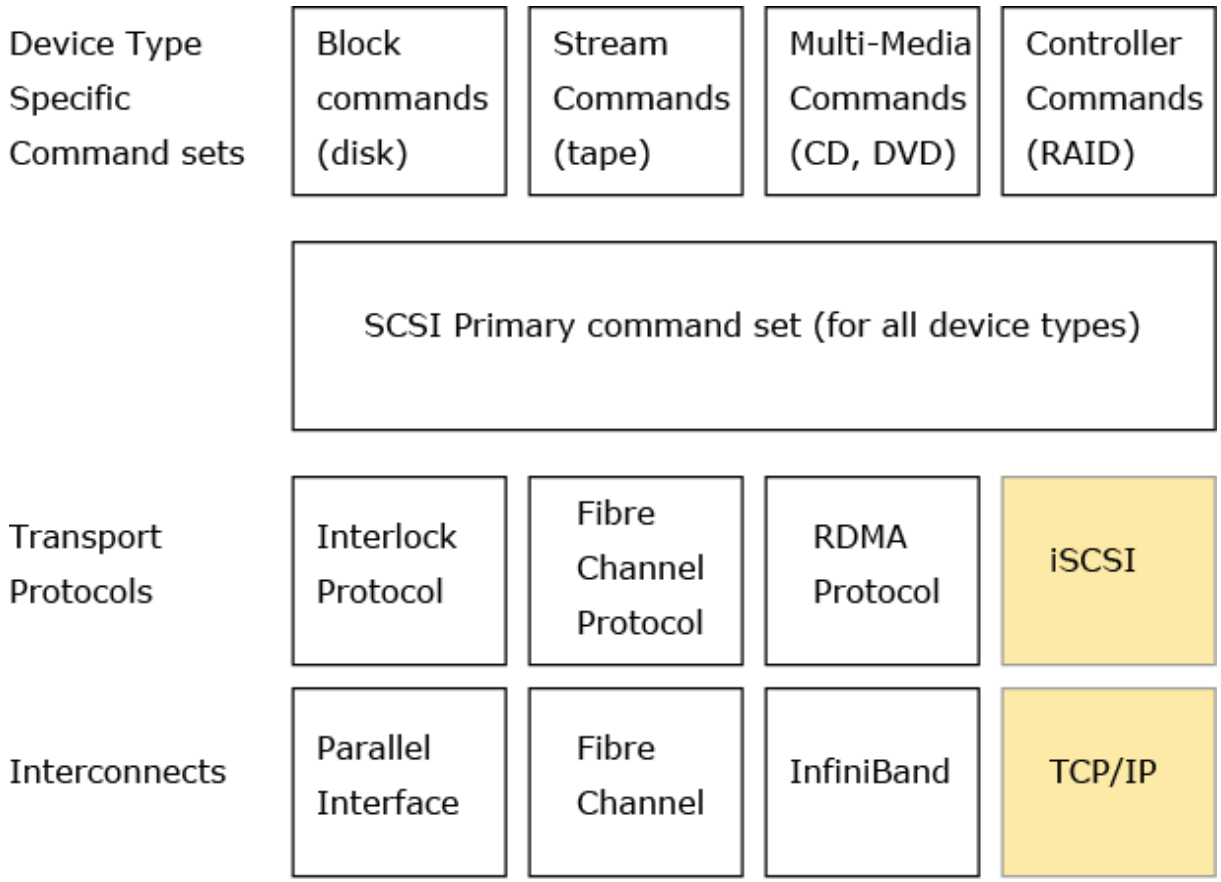
SAN利用例4: ディザスタリカバリ

- SANのディスク to ディスクのリモートミラーリング



What’s iSCSI?

iSCSIはトランスポートにTCP/IPを使いSCSIコマンドを運ぶ



なぜ、iSCSIが必要だったのか？

- Fibre Channel (FC)がSANの主流技術
  - ハードウェア（FC HBA、スイッチ、ストレージ）が高価
  - FCの技術者が少ないため高コスト

安価なIP技術を使うSAN - iSCSI

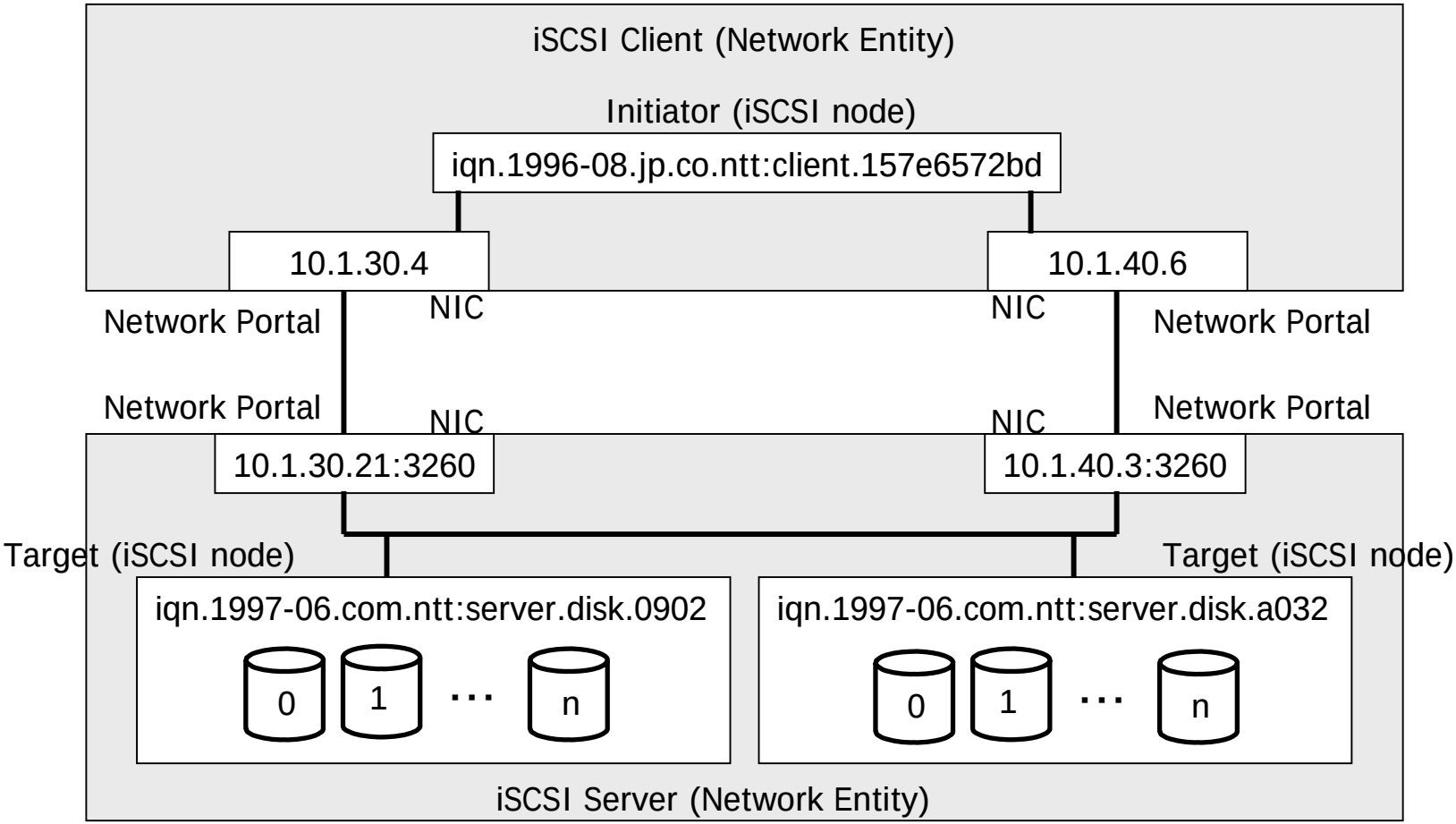
- 特徴
  - Ethernet用ハードウェアは安い
  - TCP/IPやEthernetの技術者は多い
  - 距離の制限がない
  - 他の様々なIP技術との連動が容易
- 普及状況
  - 2005年iSCSI市場は130%拡大（米IDC）

The basics of iSCSI

Sessions and Connections

- セッションは、SCSIケーブルに相当
  - 一つ以上のコネクションを保持
  - コマンドの配送順番は守られる

iSCSIアーキテクチャ例

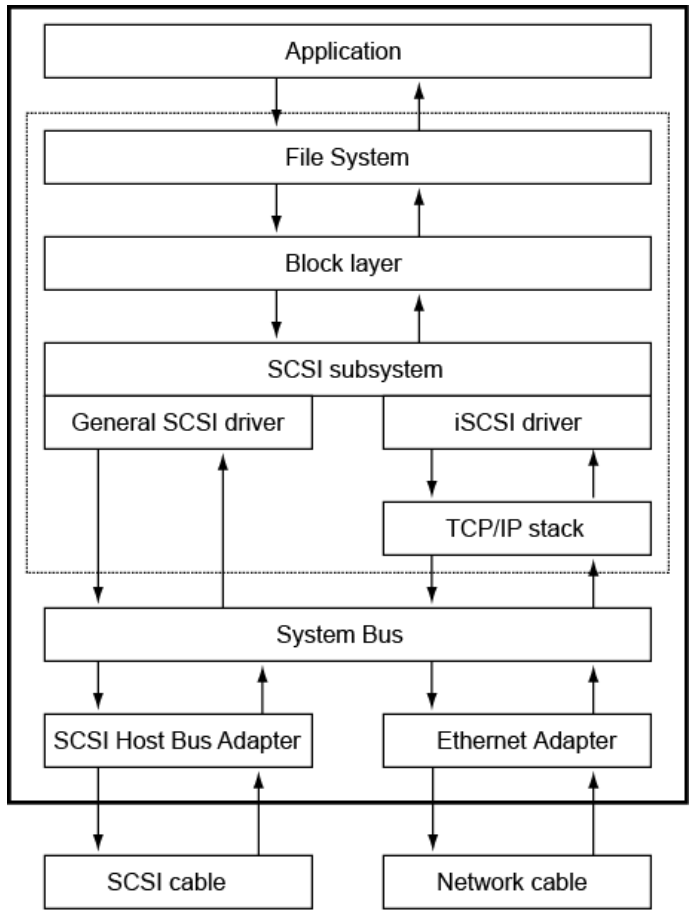


iSCSIアーキテクチャ

- iSCSI Entity (client, server)
  - Network Portal (IP network address)を一つ以上持つ
  - iSCSI Node (initiator, target)を一つ以上持つ
    - Globally uniqueな名前で識別される

iSCSIイニシエータドライバ

- 通常のSCSI HBAのドライバの一種
- HBAではなくTCP/IPスタックにアクセス
- 必要なハードウェアはNICだけ
- ファイルシステムやアプリケーションからは、違いは分からない



iSCSIターゲットドライバ

- 必要なハードウェア
  - NIC
  - ディスクドライブ
- 処理
  - イニシエータからiSCSIパケットを受け取る
  - パケットからSCSIコマンドを取り出す
  - コマンドを実行する
    - ディスクの入出力
  - コマンドの結果をiSCSIパケットに入れる
  - イニシエータにパケットを送信する

iSCSI messages

iSCSI Protocol Data Unit (PDU)

- イニシエータ・ターゲット間のデータフォーマット
  - Control messages
  - SCSI commands
  - Parameters
  - Data

PDU format

- Basic Header
  - 常に48バイト
- Additional Header
- Header digest
  - CRC32
- Data Segment
- Data Digest
  - CRC32

Basic Header	48バイト
Additional Header （オプション）	
Header digest （オプション）	4バイト
Data Segment （オプション）	
Data Digest （オプション）	4バイト

Initiator to target messages

00	NOP-out	Ping or answer to NOP-in
01	SCSI command	Send SCSI CDB
02	Task Mgmt request	Abort Tasks, reset target or LU
03	Login request	Create a connection
04	Text request	Used for Extensible commands
05	SCSI DataOut	Contain Data for SCSI Write
06	Logout request	End a connection/session
10	SNACK request	Request retransmission

Target to initiator messages

20	NOP-In	Ping or response to a NOP-Out
21	SCSI response	Send SCSI command status and sense
22	Task Mgmt response	
23	Login response	
24	Text Response	
25	SCSI DataIn	Contain data for SCSI READ
26	Logout response	
31	Ready To Transfer	target is ready to receive write data
32	Asynchronous Msg	Sends event notifications
3F	Reject	Reject initiator messages

iSCSI PDU例 - SCSI Read

```
⊞ Frame 74 (114 bytes on wire, 114 bytes captured)
⊞ Ethernet II, Src: 00:04:2e:01:cd:04, Dst: 00:04:2e:01:bd:6e
⊞ Internet Protocol, Src Addr: 192.168.0.128 (192.168.0.128), Dst Addr: 192.168.0.2 (192.168.0.2
⊞ Transmission Control Protocol, Src Port: 3438 (3438), Dst Port: 3260 (3260), Seq: 5773, Ack: 1
⊟ iSCSI (SCSI Command)
    Opcode: SCSI Command (0x01)
    .0.. .... = I: Queued delivery
  ⊞ Flags: 0xc1
    TotalAHSLength: 0x00
    DataSegmentLength: 0x00000000
    LUN: 0000000000000000
    InitiatorTaskTag: 0x18000000
    ExpectedDataTransferLength: 0x00001000
    CmdSN: 0x00000018
    ExpStatSN: 0x00000019
    Data In in: 75
    Response in: 75
⊟ SCSI CDB
    [LUN: 0x0000]
    Opcode: Read(10) (0x28)
    DPO = 0, FUA = 0, RelAddr = 0
    Logical Block Address (LBA): 4158
    Transfer Length: 8
    Vendor Unique = 0, NACA = 0, Link = 0
```

```
0000  00 04 2e 01 bd 6e 00 04  2e 01 cd 04 08 00 45 00   .....n.. ......E.
0010  00 64 44 7f 40 00 40 06  74 42 c0 a8 00 80 c0 a8   .dD.@.@. tB......
0020  00 02 0d 6e 0c bc 01 6e  ac bb 02 76 c6 b3 80 18   ...n...n ...v....
0030  17 b8 e4 b2 00 00 01 01  08 0a 00 18 fa 2b 00 19   ........ .....+..
0040  02 3d 01 c1 00 00 00 00  00 00 00 00 00 00 00 00   .=...... ........
0050  00 00 18 00 00 00 00 00  10 00 00 00 00 18 00 00   ........ ........
0060  00 19 28 00 00 00 10 3e  00 00 08 00 00 00 00 00   ..(....> ........
0070  00 00                                             ..
```

48バイトのiSCSI PDUが、TCPのデータとして含まれている

SCSI CDB (Command Descriptor Block)

iSCSI operations

ログイン

- コネクションを作成する
  - 最初のコネクションであればセッションも作る
- 三段階
  - Security Negotiation
    - 認証
  - Operational Negotiation
    - パラメータ
  - Full Feature Phase
    - SCSIコマンドが実行できる

Authentication

- No authentication
- CHAP
  - TargetによるInitiatorの認証
  - InitiatorによるTargetの認証
  - RADIUSも使えます
- Kerberos
- SPKM (Simple Public-key Mechanism)
- SRP (Secure Remote Password)

パラメータネゴシエーション例

Login request

```
Key/Value Pairs
  KeyValue: InitiatorName=iqn.1987-05.com.cisco:01.4438aca09387
  KeyValue: InitiatorAlias=lilac
  KeyValue: TargetName=iqn.2001-04.com.example:storage.disk2.amiens.sys1.xyz
  KeyValue: SessionType=Normal
  KeyValue: HeaderDigest=None,CRC32C
  KeyValue: DataDigest=None
  KeyValue: DefaultTime2Wait=0
  KeyValue: DefaultTime2Retain=0
  KeyValue: IFMarker=No
  KeyValue: OFMarker=No
  KeyValue: ErrorRecoveryLevel=0
  KeyValue: InitialR2T=No
  KeyValue: ImmediateData=Yes
  KeyValue: MaxBurstLength=16776192
  KeyValue: FirstBurstLength=262144
  KeyValue: MaxOutstandingR2T=1
  KeyValue: MaxConnections=1
  KeyValue: DataPDUInOrder=Yes
  KeyValue: DataSequenceInOrder=Yes
  KeyValue: MaxRecvDataSegmentLength=131072
Padding: 00
```

Login response

```
Key/Value Pairs
  KeyValue: TargetPortalGroupTag=1
  KeyValue: HeaderDigest=None
  KeyValue: DataDigest=None
  KeyValue: DefaultTime2Wait=2
  KeyValue: DefaultTime2Retain=0
  KeyValue: IFMarker=No
  KeyValue: OFMarker=No
  KeyValue: ErrorRecoveryLevel=0
  KeyValue: InitialR2T=Yes
  KeyValue: ImmediateData=Yes
  KeyValue: MaxBurstLength=262144
  KeyValue: FirstBurstLength=65536
  KeyValue: MaxOutstandingR2T=1
  KeyValue: MaxConnections=1
  KeyValue: DataPDUInOrder=Yes
  KeyValue: DataSequenceInOrder=Yes
  KeyValue: MaxRecvDataSegmentLength=16384
Padding: 00
```

パラメータ

- List / value
  - List型 - 選択可能な値をリストとして提示（応答側は一つを選択する）
  - Value型 – 特定の値を指定
- Negotiation / Declare
  - Negotiation型 – イニシエータとターゲットの双方がacceptできる値に設定される
    - 例えば、小さい方の値など（パラメータによって決定方法は異なる）
    - イニシエータとターゲットが1つの値を共有
  - Declare型 – 一方的に宣言して決定
    - イニシエータとターゲットが別の値を持つパラメータもある

SCSI Command Message

- Initiator Task Tag
  - このコマンドを識別するためのユニークな値
- Expected Data Transfer Length
  - このコマンドで発生するデータ転送の長さ
- Command sequence Number
  - コマンドの配送順序を守るために使う

OP code & R/W flags	4 bytes
AHS length Data Segment Length	4 bytes
LUN	8 bytes
Initiator Task Tag	4 bytes
Expected Data Transfer Length	4 bytes
Command SN	4 bytes
Expected Status SN	4 bytes
SCSI CDB	16 bytes

SCSI DataIn Message

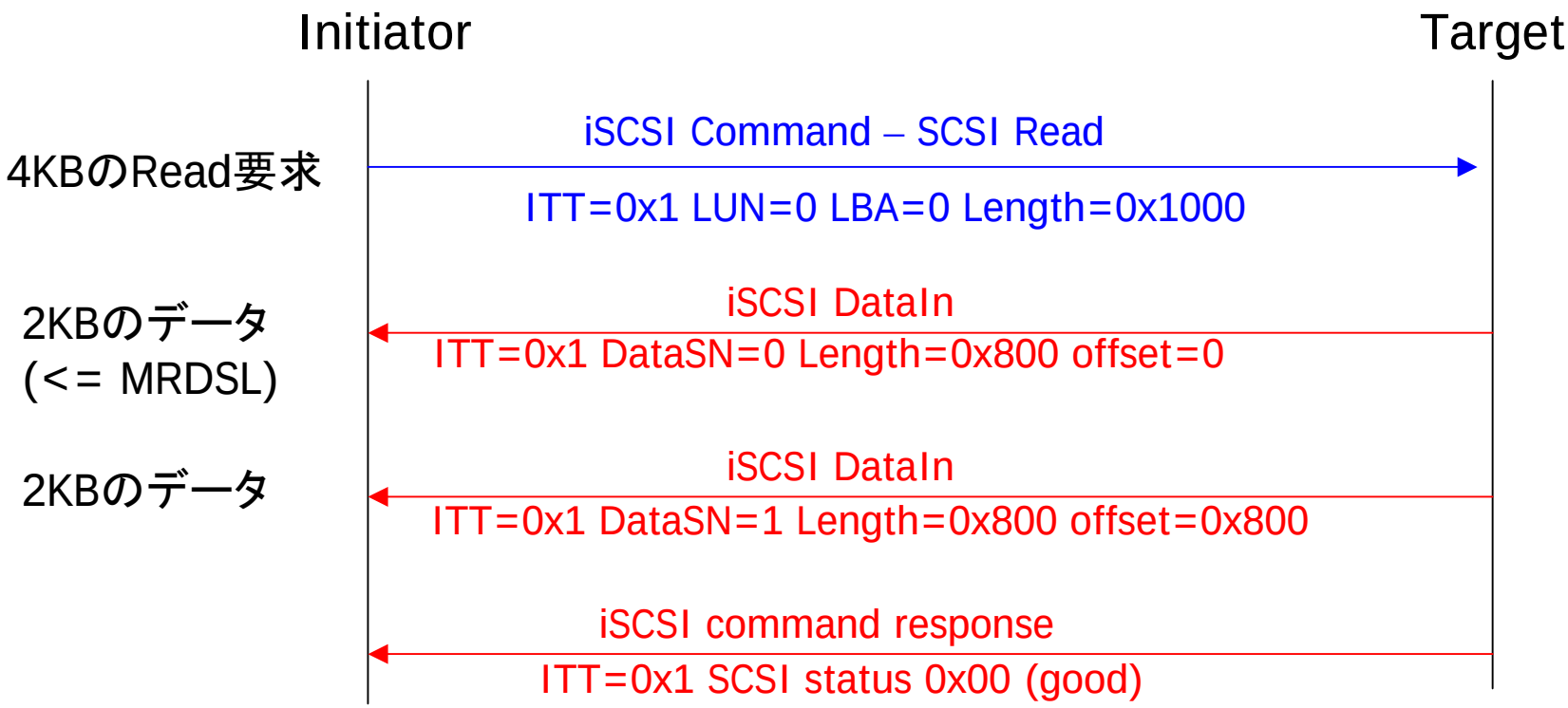
- Initiator Task Tag
  - Read commandと一致
- Data Segment Len
- Target Transfer Tag
  - Error recoveryに使用
- Max Command SN
  - Flow control
- Data Sequence Number
  - 0から始まりDataIn messege毎に増やす
- Buffer Offset
  - データのオフセット

OP code & flags	4 bytes
AHS & Data Segment Len	4 bytes
LUN	8 bytes
Initiator Task Tag	4 bytes
Target Transfer Tag	4 bytes
Status SN	4 bytes
Expected Command SN	4 bytes
Max Command SN	4 bytes
Data SN	4 bytes
Buffer offset	4 bytes
Residual Count	4 bytes

Readに関係するパラメータ

- イニシエータの宣言したMaxRecvDataSegmentLength(MRDSL)
  - イニシエータが受け取るPDUの最大のデータセグメント長
  - 宣言型のパラメータ
    - イニシエータのMRDSL値とターゲットのMRDSL値は異なる

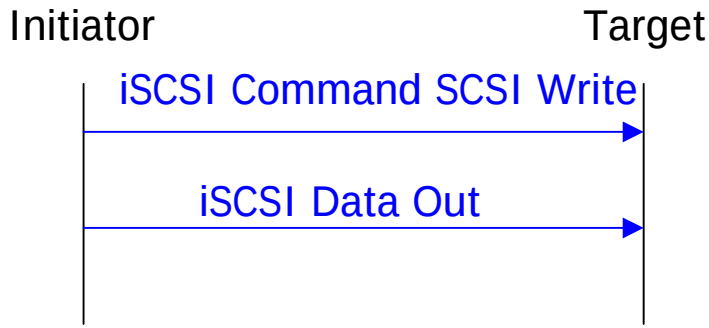
SCSI Read sequence

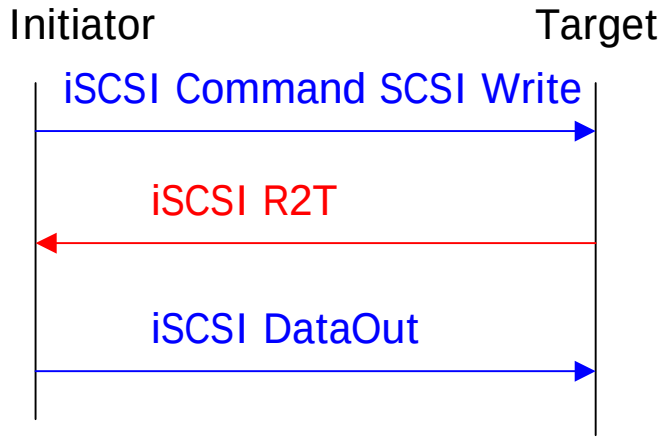


InitiatorのMRDSL = 2048　　4KのREADを2個のDataInで受け取る

SCSI Write Unsolicited/Solicited

- Unsolicited
  - InitiatorはTargetの許可を受けずにWrite dataを送信する
- Solicited
  - InitiatorはTargetの許可（R2T command）を得て、Write Dataを送信する





SCSI Write- Immediate data

- Unsolicitedの一種
  - iSCSI Command SCSI Writeと一緒にデータを送信する（DataSegmentとして）

SCSI R2T Message

- Initiator Task Tag
  - Write commandと一致
- Target Transfer Tag
  - このR2T messageを識別するユニークな値
- R2T sequence number
  - 同じITTを持つR2T messageの順序を保護
- Buffer Offset
  - 要求データのオフセット
- Desired Length
  - 要求するデータ長

OP code & flags	4 bytes
AHS & Data Segment Len	4 bytes
LUN	8 bytes
Initiator Task Tag	4 bytes
Target Transfer Tag	4 bytes
Status SN	4 bytes
Expected Command SN	4 bytes
Max Command SN	4 bytes
R2TSN	4 bytes
Buffer offset	4 bytes
Desired Data Xfer Length	4 bytes

SCSI DataOut Message

- Initiator Task Tag
  - Write commandと一致
- Target Transfer Tag
  - 0xffffffff (for Unsolicited)
  - R2Tと一致 (for solicited)
- DataSN
  - 1つのシーケンス内で0から始まりDataOut messege毎に増やす
- Buffer Offset
  - このメッセージが運ぶデータのオフセット

OP code & flags	4 bytes
AHS & Data Segment Len	4 bytes
LUN	8 bytes
Initiator Task Tag	4 bytes
Target Transfer Tag	4 bytes
Reserved	4 bytes
Expected Stat SN	4 bytes
Reserved	4 bytes
Data SN	4 bytes
Buffer offset	4 bytes
Reserved	4 bytes

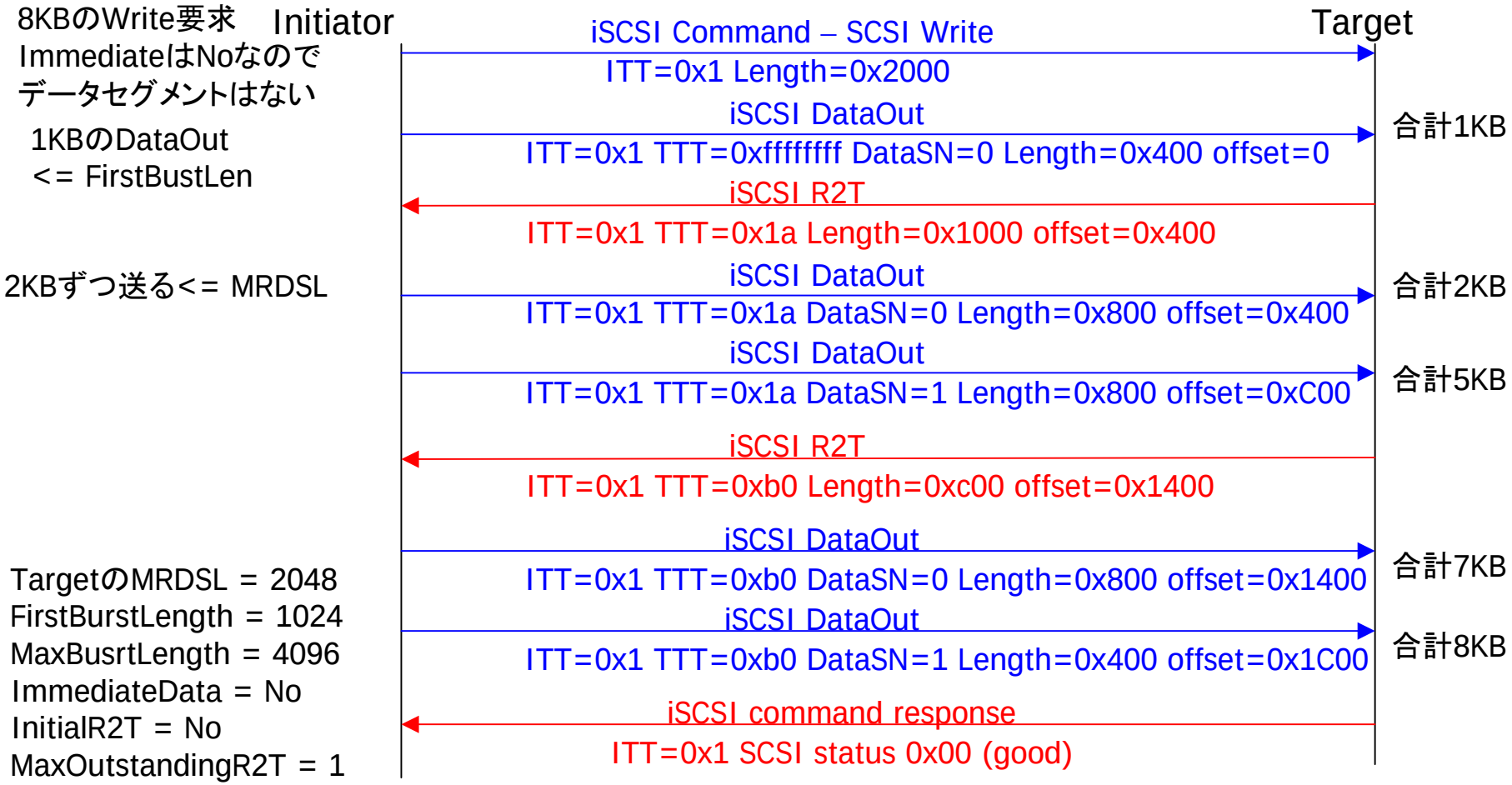
Writeに関係するパラメータ(1)

- ターゲットの宣言したMaxRecvDataSegmentLength
  - ターゲットが受け取るPDUの最大データセグメント長
- FirstBurstLength
  - UnsolicitedなWriteの最大長
- MaxBurstLength
  - R2Tで要求できるWriteの最大長
- InitialR2T
  - UnsolicitedなWriteを使えるか
  - Yes -> UnsolicitedなWriteを使えない

Writeに関係するパラメータ(2)

- MaxOutstandingR2T
  - 未処理のR2Tコマンドの最大数
- ImmediateData
  - Immediate dataを使えるか
  - Yes -> Immediate dataを使える

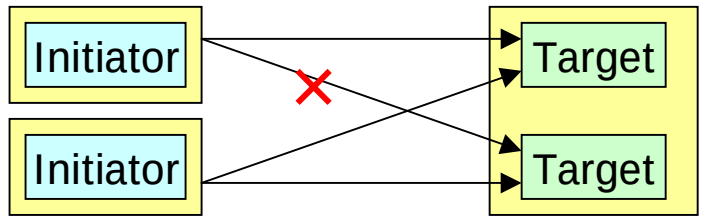
SCSI Write sequence

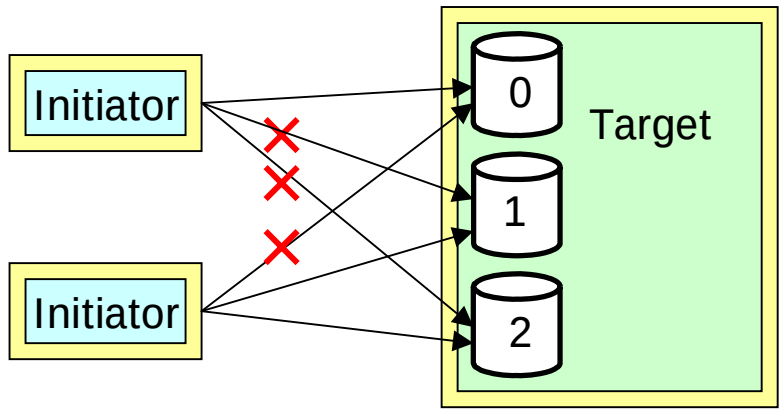


Try iSCSI

ターゲットによるイニシエータのアクセス制御

- ターゲットベース
  - ターゲット毎にアクセスできるイニシエータをIP等で制限
  - 一つのiSCSIサーバの中に複数のターゲットを起動
- LUNベース
  - ロジカルユニット毎にアクセスできるイニシエータをIP等で制限
  - LUN maskingと呼ばれる





イニシエータドライバのサポート状況

- Linux
  - 標準カーネルに含まれている
- Windows
  - Microsoftが配布

Linuxイニシエータドライバ

- sfnet (linux-iscsi 4.x)
  - 多くの商用ストレージがサポート
  - 2.6.10カーネルまでをサポート
  - RHEL等のディストリビューションに含まれている
- open-iscsi (linux-iscsi 5.x)
  - 標準カーネルに取り込まれている（2.6.15から）
  - 他の実装と比較すると完成度がまだ低い
  - 最新のカーネルでしかコンパイルできない
- core-iscsi

ターゲットドライバのサポート状況

- Linux
  - 標準カーネルのサポートに向けて開発中
- Windows
  - Windows Storage Server 2003に含まれる予定 (WinTargetと呼ばれていた他社製品を買収)

Linuxターゲットドライバ

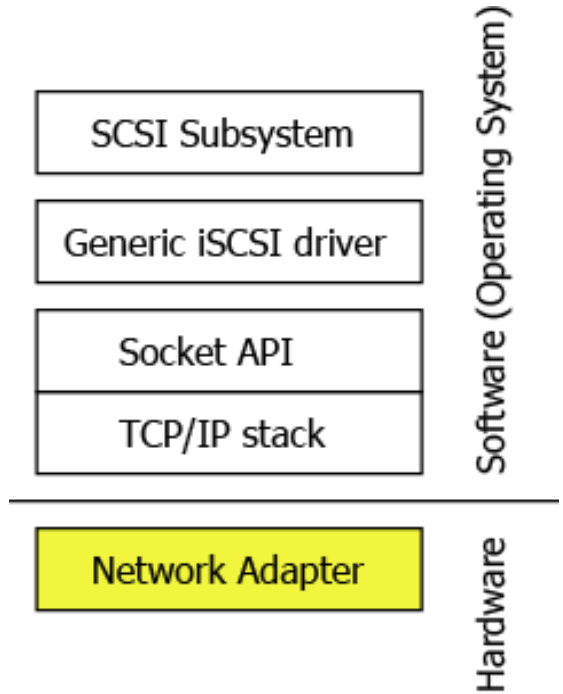
- iSCSI Enterprise Target Software (IET)
  - 最も広く使われているLinux用iSCSIターゲットドライバ
  - ファイルやブロックデバイス（LVM等を含む）をイニシエータにサービスできる
  - 標準のカーネルに含まれる実装とは異なる

More Performance?

TCP/IP is bottleneck

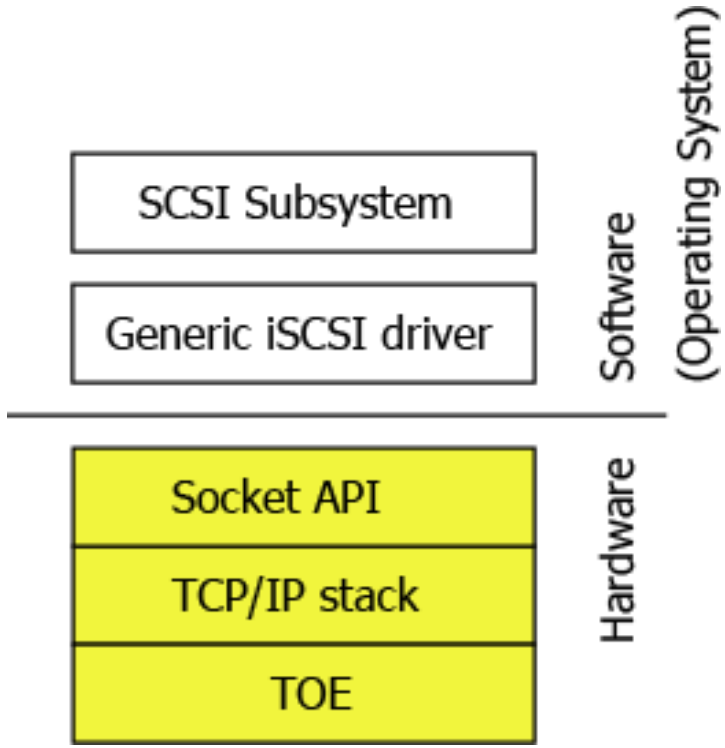
- 10GbEでは問題
  - プロトコル処理（割り込み）
  - メモリコピー
- OS機能のハードウェア化
  - TCP/IPスタック
  - iSCSI処理

通常のNICのモデル



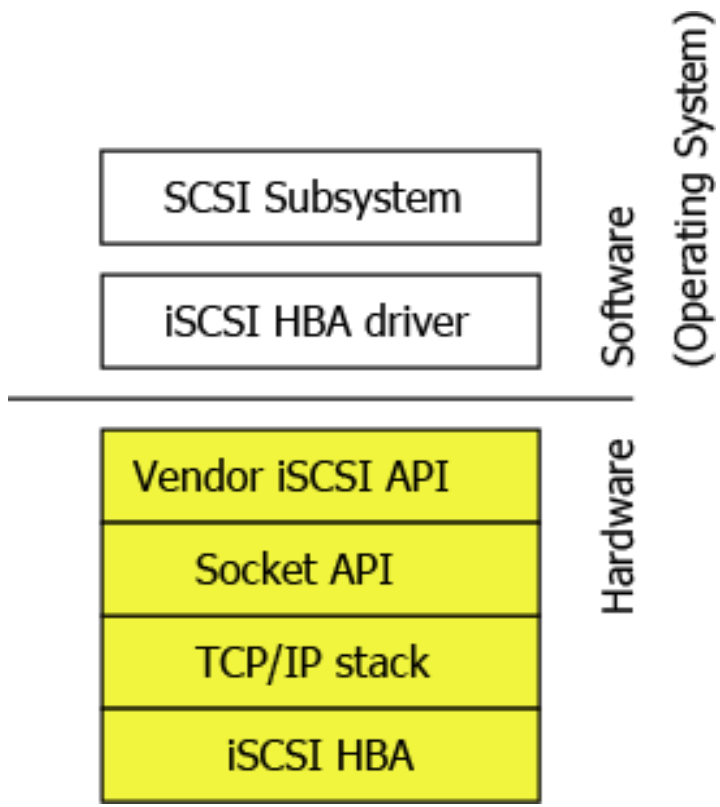
TCP Offload Engine

- TCP/IPスタックをNIC上のハードウェア化
- Socket APIを保持
  - 通常のNIC用iSCSIドライバがほぼそのまま使える
- メモリコピーは発生



iSCSI Host Bus Adapter (HBA)

- TCP/IPスタックとiSCSIプロトコル処理機能をNIC上のハードウェア化
- ベンダ独自のインターフェイスを利用
  - 各HBA毎にドライバが必要
  - 通常NIC用iSCSIドライバと協調するのが困難



iSCSI Extensions for RDMA (iSER)

- iSCSIがRDMAを利用してデータ転送するための拡張
- RDMA capable interface
  - InfiniBand (IB)
  - RDMA NIC (RNIC)
    - Internet Wide Area RDMA Protocol (iWARP)
      - RDMA Protocol Interfaceを提供
    - EthernetでRDMAを実現