

NDMP v5: Working Group Backgrounder:

Strategy, Architecture, and Other Key Components:

Author: Harald Skardal, Network Appliance, Inc.

Version 0.5: 2000/02/29.

Introduction:

This document summarizes our thoughts on how we are approaching the development of NDMP version 5. It is based on collected comments and discussions in the meetings so far. It is not intended as a complete list of issues for NDMP to solve. Rather it is a set of suggested perspectives for how to develop NDMP v5 such that it has the ability to solve the problems at hand, and have the ability to grow further in succeeding revisions.

The document is considered a working document intended to collect our current thoughts on NDMP v5. As such it will evolve over a period of time. It is not a definitive reference and may contain redundancies or ambiguities. We also expect that there will be new issues added to this document as the working group forms and grows.

This document is also intended as an orientation for newcomers to the NDMP standard's effort: people who wants to participate in the NDMP working group and help develop the NDMP standard, collateral, layered standards, etc., or for people who want to monitor the NDMP development process.

Definition of Terms:

The following defines terms and key concepts that are central to this document. An attempt is made to use directly or extend cleanly language that is already common in Internet RFC's:

Session (*):

A session is a set of data service senders and receivers and the data streams flowing from senders to receivers. A backup operation is an example of a data management session.

Note that sessions can be persistent. In a network where backup is being run continuously from multiple primary storage systems onto a tape library, a session may never end.

Data Management Application (DMA):

The application which uses the NDMP protocol and NDMP compliant storage product to create and run an NDMP session for the purpose of data management: performing a backup or restore of a data volume, replicating a file system, etc.

In previous versions of the NDMP specification the term used most commonly for a Data Management Application has been the NDMP client.

Data Service Provider (DSP):

An NDMP compliant data producer, consumer, or producer and consumer. The DSP reads a storage source into a stream, writes a stream onto a storage target, or translates one or more input streams into one or more output streams.

There are multiple types of DSP's: Data service, Tape service, SCSI Pass-through Service and the new service introduced in v5, the translate service. Some example DSP's: Data service providers include servers with internal storage, NAS devices such as filers, NDMP accessible direct or SAN attached storage subsystems. Tape services include general servers with shared tape drives, "juke boxes", write-able CD-ROM libraries, etc. Translate services include N to M stream multiplexers, data compression tools, etc.

In previous versions of the NDMP specification the term used most commonly for a Data Service Provider has been the NDMP server.

Network:

In the following “network” means any combination of Enterprise network, Intranet, Internet or storage area network.

* - This use of the word “session” is taken from IETF RFC’s such as RFC 2327, Session Definition Protocol.

Brief NDMP History:

NDMP v1 was developed to solve the specific issue of how backup management software supports storage appliances. Dave Hitz and Roger Stager, founders of Network Appliance and PDC respectively focused on a solution that would address the business needs of both companies and at the same time be generic enough for other ISVs and system vendors to utilize.

Both rsh & NFS mounted solutions were in use and still are today, but managing errors and optimizing these solutions for performance was viewed to be challenging at best.

A new protocol was needed, one that took the "appliance approach" and let the parties involved focus on their strengths. The DMA providers present a GUI to the system administrator, for scheduling backups and keeping track of what files are in a backup so that the data can later be restored with single file granularity. The DSP vendors understand the layout of and provide access to the data/storage so that the backup and restore methods can be optimized, and one can easily track file system or hardware changes.

Version 1 of NDMP concentrated on backup/restore using only locally attached tape devices in order to allow for rapid implementations and short time to market. Interfaces were defined to allow the storage appliance to *notify* the backup management software of files being backed up (file history), log messages and required tape changes, etc. This allowed the backup management software to take a "hands off" approach once the backup or restore was started.

Version 2 of the NDMP spec added support for both 3-way backup (tape device on another storage appliance) and remote (tape device on the [general purpose backup] DMA server) backup & restore. V2 also allowed customers to use a centralized tape library versus a tape device/library per storage appliance.

During the implementation of V2 the need for Data Service to Data Service and Tape Service to Tape Service data streams were identified and added to the version 3 NDMP spec. Other new functionality was improved configuration information and some auto discovery capabilities, allowing a DMA to query a DSP for the DSP’s capabilities.

NDMP Objectives:

The main objective with NDMP is to evolve an open protocol for facilitating interoperability between data management applications and data service providers on a network. In particular, NDMP focuses on a clean functional separation of data management application and data service providers, thus improving time to market for complete data storage and management solutions. Specifically we want to provide an evolutionaryreusability

Data management applications provide administrative control of the movement of data that is stored in storage devices on an Intranet or Internet. Data movement in the context of NDMP means data movement for administrative purposes (data administration) rather than data access.

The data management application is responsible for policy, user interface and for managing the data service providers.

NDMP is the protocol used by the data management application to set up, configure and control a data management session, in particular to manage the flow of data between the data service providers. NDMP is also used by the DSP's to notify the DMA when events occur and the DSP needs the DMA's attention.

The predominant use of NDMP is backup, restore and data replication (mirroring), although other applications of the protocol are also possible.

The data management application manages the media. I.e. it is responsible for taking the appropriate action when a tape cartridge is full or needs to be changed.

An objective with v5 is to "reuse" technologies that already exist in the IETF space: SNMP, LDAP, etc.; for needs that exist in the area of data management. This should allow NDMP to focus solely on the task of managing the movement of data on the network.

The result should be a "blindingly simple and explicit" NDMP protocol and architectural model which makes it simpler to implement NDMP compliant DMA's or data service providers.

V3 Issues

The two key limitations with v3 are the limitations to single stream backup-restore sessions, and interoperability issues due to ambiguities in the v2 and v3 specifications.

The single stream limitation prevents optimal use of the resources or prevents a session from being completed faster. Assume that a data producer can write data much faster than a consumer can read the data. A session topology which allows for de-multiplexing one data producer stream into N streams for N consumers will allow a backup of a single storage devices to finish in 1/N-th the time it would take to finish a backup with one consumer. Likewise, if a shared tape resource can consume data much faster than a single data producer, multiplexing M streams from M producers into one stream for one consumer will allow the backup of M data producers to finish in the same time as for a single producer.

There are several areas in the v2 and v3 specifications where there are significant ambiguities. This causes two problems. Either NDMP compliant DMAs and DSPs do not fully inter-operate. There are configurations where certain DMA-DSP combinations will fail or remain unstable. Secondly, and to overcome this problem, considerable efforts are still needed to make DMA's and DSPs fully compliant, leading to significant amounts of vendor specific code.

New Requirements:

Because of or in addition to the known limitations discussed above, a set of new functional requirements has emerged as key to NDMP's value going forward. The ones known at this point are:

Multiplexing:

The ability to backup one data producer such as a filer to two or more data consumers such as shared tape drives in order to reduce the backup time. This case is called "fan out". The ability to backup multiple data producers to one data consumer. This case is called "fan in". In the general case, such as in a complex storage environment, there are many data producers delivering data to many data consumers.

Snapshots:

Snapshots are becoming an important tool in the management of data. Many companies have efficient implementations of snapshots. NDMP must be able to provide DMA's with programmatic control over the overall snapshot management process.

Growth of Data:

As data capacity increase and data sets grow, the performance of the common data management operations becomes important. In order to provide higher performance solutions NDMP needs to support new technologies such as fiber channel, SAN, VI, and other important storage or network oriented technologies.

NDMP v5 Scope:

1: To create a new version of NDMP that is an evolutionary next step from v3. This will allow for significant reuse-ability by building upon most of v3, while providing the necessary improvements and future extendibility.

2: With NDMP v5 we want to improve the interoperability between storage products from different vendors and backup/data management applications:

- Create a stronger, cleaner, and better documented NDMP architectural model.
- Identify & fix ambiguities & inaccuracies that exist in v1-3,
- Improve the completeness and precision of NDMP error reporting:
 - Clarify error definition and implementation,
 - Define must and should behavior for messages and actions,
 - Improve the identification of vendor, product and version information.

One result should be a reduction in the need for vendor specific messages.

- Standardize and improve precision of the built in lists of NDMP specific and vendor specific environment variables. (general vs. tar vs. cpio vs. dump vs. . .)
- Move authoritative parts of the current workflow document into spec (appendixes)

3: We are making NDMP v5 more extensible by adding the new translator (X-late) data service provider. This new service allows for the creation of new functionality such as:

- Fan-in/Fan-out Multiplexing
- En-/decryption of data in transit or for storage
- De/Compression
- Other custom processing support

4: Adding support for Snapshots. We will liaise with the SNIA snapshot working group.

5: We are adding SAN support: NDMP sessions will support SAN attached storage service providers as well as direct and network attached storage systems.

6: We want to improve NDMP session management:

- Recoverability of broken data streams for restart
- Improved check pointing of NDMP sessions.
- Add support for session management.

Deferred Features

The following features have been considered but are currently deferred until a later version of NDMP.

- NDMP Proxy: This is the case when an application, such as an Oracle database server, takes control over the DMA. In this case the DMA becomes a “networked service” which can be controlled from programs, including scripts.

Out of Scope/Related work section:

The following tasks are related to the NDMP effort, and should be considered as background material with the purpose of focusing the NDMP effort itself, or as suggested areas for NDMP related work within the SNIA working groups.

- Standard multiplexing/de-multiplexing algorithms that can handle nested applications. The result is storage formats that for instance allow data to be recovered by a different DMA from the DMA that performed the backup.
- Self contained tape set with XML encoded session database on tape. This requires basically an XML DTD and grammar for how to put the complete state for an NDMP session on the tape such that another NDMP compliant DMA can recover the session state, and from that do a full or partial restore of the content.
- Directory services/LDAP: Over time we expect all NDMP capable DSPs, and each DSPs NDMP attributes, to be registered in an Enterprise level directory service. The coding of the NDMP attributes on a network and DSP level should be defined.
- Network Management, SNMP: The development of a MIB for NDMP services for the purpose of reporting network level status to a network management console.
- A standard format for data streams: There is a European standard for a tape stream format (SIDF). Standards in these two areas would make the content of a tape set recoverable all the way down to a potentially proprietary data stream.

Standardization Strategy:

The final goal with NDMPv5 is to become an IETF accepted RFC. The IETF has stated that it looks for SNIA to become a “pre standard” generating organization. This implies that when a proposal for a standard has been accepted by SNIA, it can be submitted to the IETF as an Internet draft to be reviewed for Internet standardization.

Since SNIA is becoming one of several pre-stage forums for developing Internet standards, the best way of working towards IETF standardization of NDMP v5 is to use SNIA to form a working group of people from the storage industry. This working group will assure that the result, NDMP v5, becomes a mechanism that is both necessary and sufficient for building data management solutions that address current and future needs.

Schedule and Deliverables:

We plan to provide the IETF with a draft proposal by end of this calendar year.

We announced the formation of an NDMP working group during the SNIA working group meetings in San Jose, February 8-11, 2000.

We will have the first meetings at March 7th during Connectathon.

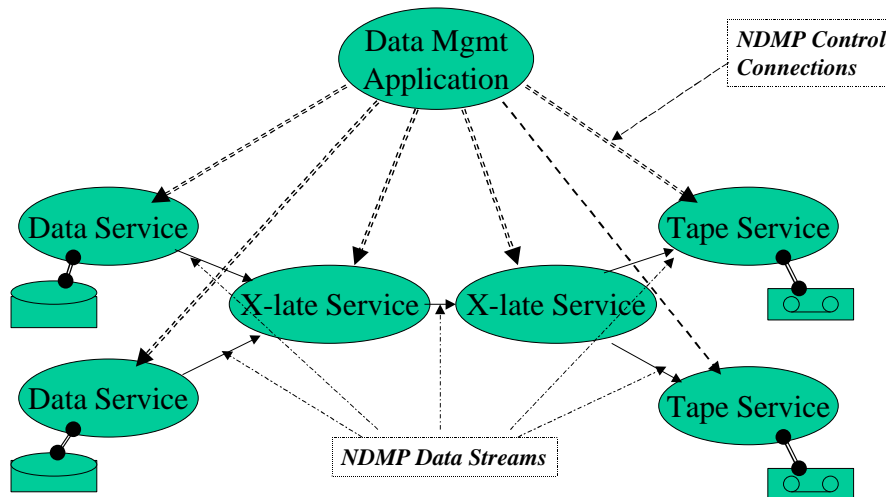
The SNIA NDMP working group has the following deliverables:

- NDMP Version 5 specification
- A Workflow document:
 - Educational, non-authoritative
 - Describes the use of the protocol in common scenarios
 - Helps developer developing compliant implementations
- Reference implementation and SDK
- Compliance test suite

- NDMP v5 FAQ

The Emerging Model:

One of the requirements to NDMP v5 is to have a clearer and better explained architectural model. The above requirements lead to the following abstract model for an NDMP session. See figure #:



The model is focused around the creation and management of control connections and data streams. Data from a volume or file system is turned into a data stream by a data service, a tape service takes a stream, converts it into a tape format and writes it to tape, or vice versa. In the middle, between the data producers and consumers is the place where more complex functionality can be added through one or more translator services. Streams from multiple sources can be multiplexed into one stream. A stream can be duplicated for parallel generation of identical tapes. A stream can be compressed or decompressed for more efficient storage or data transport. A stream can be encrypted or decrypted for secure network transport or storage.

This leads to a four component architecture.

- We call the module that interfaces to the volume or file system to be operated on as the “Data Service” (D).
- We call the module that interfaces to a tape drive the “Tape Service” (T).
- We call the modules that sit in between one or more Data Services and one or more Tape Services the “Translate Service” or X-late (X).
- We call the application that uses NDMP to set up and control the entire NDMP event the “data management application”.

The role of the NDMP protocol is to allow the DMA to set up, configure and control an NDMP session of NDMP controllable entities. A couple of points deserve mentioning:

In order to simplify the view of a session from the DMA, we allow for one and only one control connection between the DMA and each DSP. This allows the DMA to remain agnostic to general or optimized implementations or session configurations, such as running multiple services within one system, or using high-speed SAN connections between services.

Control Streams

Between the DMA and every DSP is one and only one control connection. The DMA uses control connections to manage each DSP. These control connections are implemented as TCP/IP sockets. Messages flow in both directions on a control connection. The DMA sends messages to the DSP for the purpose of managing the operations of the DSP. The DSP sends notifications to the DMA when the DSP requires the DMA's attention.

Note that this requirement: one and only one connection between the DMA and each DSP is a change from previous versions.

Data Streams

In the native or general case the transport for NDMP data streams is TCP/IP over any IP supported network media. In addition to handling the general case, where NDMP control connections and data streams are TCP/IP based, NDMPv5 needs to handle data streams for direct and SAN attached storage and tape devices, and to utilize SAN and other high bandwidth network fabrics optimized applications. The key mechanism that provides this is a special connection type, "stream by reference". When a data streams are needed between two storage services which are both connected to the same server or a shared SAN, the system can utilize high performance IPC or SAN for these streams. Notice that each server still must be connected to the LAN for TCP/IP based NDMP control connections.

DMA

In v5 we put great emphasis on making the DMA responsible for capturing and managing all state needed to provide the desired DMA capabilities such as recover/restart-ability of a halted session, or for enabling partial restores of data. In addition the DMA is responsible for media management.

The DSP only keeps local running state and will not retain state from one connection to another. The DMA will interact with the DSP in order to define potential restart points if a session fails, or to record sufficient information to enable optimized access to subsets of the backed up data on the tapes.

There are several benefits to an architecture that minimizes distributed state information:

- The architecture is simpler
- The protocol commands and event notifications are clearer and simpler.
- The state diagrams are simpler.
- The code becomes simpler and more supportable

The net result is more robust products, and therefore a more robust data management environment.

DSP:

There are two major categories of DSPs. One class of DSPs are the NDMP interfaces to the storage devices. Data services interface to primary storage devices such as servers with storage subsystems or filers, tape services interface to secondary storage devices such as tape drives or writeable CDRoms.

The other class of DSPs are the Translate service. It is the new component in NDMP v5. It provides NDMP with two major capabilities. First, with the ability to have multiple input and output data streams it creates an extensible session topology. Secondly, it provides a framework for an extensible translate service: any form of content inspection and manipulation can be implemented.

The DSPs are controlled by the DMA through a set of service parameters. There are two types of service parameters, service parameters which impact the NDMP protocol or state, and “NDMP opaque” service parameters that only impact vendor specific state in a DSP.

An example of opaque service parameters are the parameters controlling the operation of a translator service. A wide variety of vendor specific translator services will be created, for many different operations and many different stream formats. New services will be added over time. It is therefore important for NDMP to have a method for setting and reading parameter sets that are opaque to the protocol and the specification.

A new DSP is created by the DMA making a connection request to the ‘well known’ NDMP port (port 10000).

This creates a new, “vanilla” DSP that connects back to the DMA over a new port number. The DSP is transformed into a data, tape or translate service by the following commands that the DMA sends it.

Data Service

A data service provides the NDMP interface to a primary storage device such as a filer, a compute server with direct or SAN attached storage, or a read-only CDROM library. The data service allows a DMA to read or write the all or a subset of a volume or a file system, for the purpose of a backup or a restore. The data service produces or consumes one single data stream, for backup or replication, or restore respectively.

Tape Service

A tape service provides the NDMP interface to a secondary storage device, such as a tape drive, tape library or jukebox, or a writeable CDROM. The tape service is a single stream consumer or producer, for backup or restore respectively.

Note that a tape service only handles the reading or writing of one “cartridge”, a single tape or CD. The tape service when notify the DMA when a cartridge is read or written, the DMA will provide the necessary media management.

SCSI Pass Through Service

The SCSI pass through service allows a DMA to issue SCSI commands to a device such as a tape robot.

The Translator Service:

The new architectural component in NDMP is the translator service. In essence it is a data stream processor. It takes input from one or more data streams and outputs data on one or more data streams. Some sample translate services are data stream multiplexing and data stream compression.

The translator behavior is opaque to the NDMP protocol. Therefore the only relevant aspect to the NDMP protocol of the operation of a translator service are the control and state of the interfaces to the overall NDMP session, and the overall state of the translator as it relates to the protocol. : is it running, or does it need attention from the DMA.

As a consequence of the above, we see that the internal state of a translator service is opaque to the NDMP protocol. The NDMP protocol needs to be able to read and write a translator service’s internal state, but it should not have to understand this state, this state does not have parameters that are known to NDMP.

NDMP Design Considerations:

The following sections explain some of the design considerations for NDMP version 5. In the initial working group meetings significant amounts of time was spent on discussing these issues in order to develop the right architectural and behavioral model for NDMP.

NDMP Control Connections:

In v1-v3 we focused on providing the optimization of the most common case within the protocol. In v5 we propose simplification of the protocol and leaving optimization to the implementers. The protocol should enable optimization without specifically reflecting it.

V5 imposes the restriction of a single control connection between the DMA and each DSP. This is the only point where v5 is significantly different from v3, however, initial discussions suggest that ISVs providing DMA's see this as a simplification.

This change improves usability and coexistence of multiple versions of services and applications. For instance, this change implies that a DMA does not need to operate differently depending upon whether all, some or only one service run on the one machine.

Additionally, if the DMA implements a distributed control scheme for a single DSP, the DMA application has to solve this problem internally, and provide only one connection to each of the participating services.

When all services run in the same machine, the system can be optimized if the implementers puts all servers in the same process space, if the servers use shared memory for inter process data transfer etc. It is not the responsibility of NDMP to provide this optimization, it is considered an implementation issue. It is important that NDMP should not prevent such optimizations from taking place, nor that these optimizations should be reflected in the topology of the session.

The Interaction between DMA and DSPs:

In an active NDMP session the DMA is the master, and the DSPs are the slaves. This is done in order to simplify the protocol, and also to simplify the use of the protocol. The following bullets describe specific aspects of this:

- All activity is initiated and controlled by the DMA. A DSP will not perform any activity unless it is instructed to do so by the DMA.
- When an event happens which according to explicit or implicit service parameters makes the DSP stop, the DSP notifies the DMA that it has paused or halted. Only the DMA can resolve the situation, take the correct actions and command the DSP to continue.
- There is only one copy of DSP state in each service. In order to enable rapid restore of subsets of a backup, or to enable the restart of a backup which failed before it finished, it is the DMA's responsibility to record the necessary total session state needed for this purpose.

Increasing DMA control

The volume of data that needs to be managed in an enterprise keeps on growing exponentially. This increases the requirements to performance and flexibility of a data management session. First this means that it becomes important that data management resources are only allocated for the time they strictly required. Secondly it increases the requirement to performance by the DSP's. Third, it increases the need for the ability to accurately establish time/data points during a session that allows for restart of a session that has failed, or for the ability to quickly access and retrieve small subsets of session's the total data set.

In order to accomplish this the DMA may want to stop and "flush" the data paths in the DSPs in order to regularly record clean sets of synchronization points.

V5: Evolution vs. Revolution

A major goal towards NDMP simplicity and robustness are the efforts of i) moving state out of DSPs into the DMA, and ii) of providing a more extensible NDMP DSP architecture. One part of this is to simplify the DSP state machine and use the same state machine for all DSPs. This would create a simpler overall architecture, but also require significant changes to existing DSPs, data and tape services.

The major pros and con's to such a change are:

Allowing a change would make it easy to change the data service behavior as new DSPs are allowed to start up and do “local” work, such as preparing for a backup by creating a snapshot. This particular operation can take hours. Currently it is required that a data service is fully configured, with data streams to neighbor DSPs, before any local processing can start. This implies that a session needs to be fully set up, with all resources allocated, while the data service spends a long time preparing for the session. No data is flowing during this time.

A compromise is to use as much as possible of the old data service, but modify it sufficiently to allow for local processing before data streams are configured. It is believed that this will require minimal changes to existing DSPs and existing DMAs.

Scope of Participation:

We want to invite people from several industries including but not limited to Internet Service Providers, Application Service Providers, Storage Service providers, Computer Systems Manufacturers, Storage Subsystem Manufacturers, Storage Appliance Manufacturers, and Independent Software Providers.

Acknowledgements:

This document is a product of a collaborative effort. In addition to the author, the following contributors have provided valuable input:

Tim Gardner, Chewcoba Systems, Inc., Peggy Chang, Legato Software Inc., Roger Stager, Legato Software Inc., Grant Melvin, Network Appliance Corp., Stephen Manley, Network Appliance Corp., Herman Lee, Network Appliance Corp., Rajiv Malik, Network Appliance Corp., Greg Linn, Network Appliance, Inc.