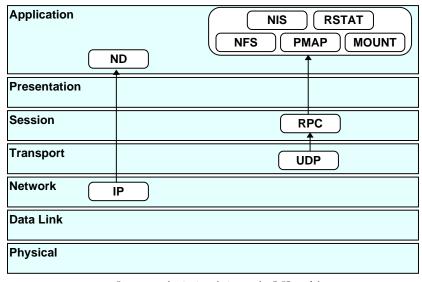
27

Sun Protocols

The Sun protocols are UNIX-based applications that use TCP/IP for transport. These protocols are based on the client-server model of networking, where client workstations make procedure calls to network servers where applications reside. The Sun suite includes the following protocols:

- MOUNT.
- NFS: Network File System.
- PMAP: Port Mapper.
- RPC: Remote Procedure Call.
- YP (NIS): Yellow Pages (Network Information Services).

The following diagram illustrates the Sun protocol suite in relation to the OSI model:



Sun protocol suite in relation to the OSI model



The MOUNT protocol is used to initiate client access to a server supporting the Network File System (NFS) application. It handles local operating system specifics such as path name format and user authentication. Clients desiring access to the NFS program first call the MOUNT program to obtain a file handle suitable for use with NFS.

Frames

MOUNT frames may be one of the following types:

Performs no operation. [no operation]

[add mount entry] Requests file handle for NFS.

Requests list of mounts for the client. [get all mounts] Requests deletion of mount entry. [del mount entry] Requests removal of mounts for client. [del all mounts] [get export list] Requests group permissions for mounts. [mount added] Supplies file handle for use with NFS. Supplies list of mounts for the client. [give all mounts] [mount deleted] Confirms deletion of mount entry. [mounts deleted] Confirms deletion of all client mounts. [give export list] Lists group permissions for mounts.

Frame Parameters

MOUNT frames can have the following parameters:

Path name

The server path name of the mounted directory displayed in double quotes.

File handle

The 32-byte file handle used to access the mounted directory.

Failure to mount on a UNIX system can lead to the display of the error message UNIX Error xxxx, where xxxx is a standard UNIX operating system error.

NFS

The Sun Network File System (NFS) protocol is the file sharing application for the Sun protocol suite. NFS allows distributed file resources to appear as a single-file system to clients. UNIX (Sun OS) is the standard platform for NFS, but Sun designed the file access procedures in NFS to be sufficiently generic to allow porting to a wide variety of operating systems and protocols.

Frames

NFS frames may be one of the following types:

[no operation] Performs no operation. [get file attrib] Request for file attributes. [set file attrib] Attempts to set file attributes. [get filsys root] Request for the root file handle.

[search for file] Requests a search for the specified file. [read from link] Requests a read from a symbolic link.

[read from file] Requests a read from a file. [write to cache] Requests a write to a cache. [write to file] Requests a write to a file.

[create file] Requests creation of the specified file.

[delete file] Requests deletion of a file. [rename file] Requests renaming of a file. [link to file] Requests creation of a file link. [make symb link] Requests creation of a symbolic link. [make directory] Requests creation of a directory. [remove directry] Requests deletion of a directory. [read directory] Requests a directory listing.

Requests information on the file system. [get filsys attr]

[give file attrib] Returns attributes for a file. Confirms setting of file attributes. [file attrib set] Returns the root file handle. [filesystem root] Returns result of file search. [file search done]

Returns result of a symbolic link read. [link read done]

[file read done] Returns result of a file read. [cache write done] Returns result of a cache write. [file write done] Returns result of a file write. [create file rep] Returns file creation status.

[delete file rep]	Returns file deletion status.
[rename file rep]	Returns file renaming status.
[link file reply]	Returns file linking status.
[symb link made]	Returns symbolic link creation status.
[make dir reply]	Returns directory creation status.
[remove dir reply]	Returns directory deletion status.
[dir read done]	Returns directory listing.
[filsystem attrib]	Returns file system attributes.

Frame Parameters

Sun NFS frames can contain the following parameters:

File handle

32-byte file handle used to access a file.

Number

Number of bytes read or written in file operations.

Offset

Beginning file offset for read and write operations.

File type

Includes the following file types:

NON Non-file.

REG Regular file.

DIR Directory.

BLK Block device.

CHR Character device.

LNK Symbolic link.

Mode

File access mode is represented in four fields as Mode/Owner/Group/Others and can include the following:

Field	Parameter	Explanation
Mode	\mathbf{U}	Executable by specifying user ID.
	G	Executable by specifying group ID.
	S	Save as executable after use.
Owner	R	Read permission.
	W	Write/delete permission.
	X	Execute/search permission.

Field	Parameter	Explanation
Group	r	Read permission.
-	W	Write/delete permission.
	X	Execute/search permission.
Others	r	Read permission.
	W	Write/delete permission.
	X	Execute/search permission.

Links

Number of hard links or filenames for the file.

Size

File size in bytes.

Block

Block size used for file storage.

File system ID

Identifying code of the file system.

File ID

Protocol ID for the file.

Cookie pointer

Pointer to first directory listing entry requested.

Maximum

Maximum number of entries to return for this request.

Read device

For block and character device, reads the number of the read device.

In addition to the above parameters, text strings such as file names are displayed in double quotes.

Response Frames

NFS response frames contain a completion status as indicated below:

 $\{OK\}$ Completed with no errors.

{Ownership required} Ownership required for operation.

{File/dir not found} File or directory not found. {Device error} System or hardware error.

{Device/addr not found} {Insuff access rights} {File/dir already exists} {Device not found} {Not a directory} {Invalid dir operation} {File too large} {Out of disk space} {Write protect violation} {Filename too long} {Directory not empty} {Disk quota exceeded} {Invalid file handle} {Write cache was flushed} {Unknown error} Device or I/O address error.
Insufficient access rights.
Duplicate file or directory.
Device access error.
Operation only valid on directory.
Operation not valid on directory.
File size too large.
File device out of space.
File system is read-only.
File name too long.
Directory not empty.
User disk quota exceeded.
File handle invalid.
Write cache flushed to disk.
Unknown error type.

PMAP

The Port Mapper (PMAP) protocol manages the allocation of transport layer ports to network server applications. Server applications obtain a port by requesting a port assignment. Clients wanting to access an application first call the PMAP program (on a well-known port) to obtain the transport port registered to the application. The client then calls the application directly using the registered port. Use of PMAP eliminates the need to reserve permanently a port number for each application. Only the PMAP application itself requires a reserved port.

Frames

PMAP frames may be one of the following types:

[no operation] Performs no operation.

[set port number] Attempt to register an application. [unset port numb] Attempt to unregister an application. [get port number] Requests registered port number. [get all ports] Requests all registered ports.

[call program] Direct call to a registered application. [port assigned] Registers an application to a port.

Unregisters an application. [port unassigned]

[give port number] Informs client of registered port. [give all ports] Informs client of all registered ports. [program called] Returns information from called program.

Frame Parameters

PMAP frames can contain the following parameters:

Transport port

Transport layer port assigned to the specified application.

Program number

Application program number.

Program version

Version number of the program.

Transport protocol

Transport layer protocol the program is registered to use.

Procedure

Procedure number within the called program.

If the program number is unregistered, the message {Program is unregistered} is displayed.

RPC

RFC 1057 http://www.cis.ohio-state.edu/htbin/rfc/rfc1057.html

The Remote Procedure Call (RPC) protocol is a session layer protocol that uses the XDR language in order to establish a remote procedure call. The RPC call message includes the procedures parameters and the RPC reply message includes the results. Once the reply message is received, the results of the procedure are extracted and the callers execution is resumed.

There are 2 message types; RPC call and RPC reply. The formats of these messages are shown in the following illustrations:

32	64	Octets
Transaction ID	Message type	8
RPC version	Program	16
Version	Procedure	24
RPC call message body		

RPC call message structure

32	64	Octets
Transaction ID	Message type	8
Status		16
RPC reply message body		

RPC reply message structure

Transaction ID

The transaction number.

Message type

The message type.

RPC version

The version of RPC being used.

Program

The name of the remote program. Programs are identified by number and, where well-known, by name. The well-known program numbers and their respective names are listed below:

Number	Program	Explanation
100000	Port_Mapper	Manages use of transport ports.
100001	RemoteStats	Remote statistics.
100002	RemoteUser	Remote users.
100003	NFS	Network file system.
100004	YellowPages	Yellow pages directory service.
100005	MountDaemon	Mount protocol.
100006	Remote_DBX	Remote DBX.
100007	YP_Binder	Yellow pages binding service.
100008	Shutdown	Shutdown messages.
100009	YP_Password	Yellow pages password server.
100010	Enet_Status	Ethernet status.
100011	Disk_Quotas	Disk quota manager.
100012	SprayPacket	Packet generator.
100013	3270_Mapper	3270 mapping service.
100014	RJE Mapper	Remote job entry mapping service.
100015	Select_Srvc	Selection service.
100016	Remote_DB	Remote database access.
100017	Remote_Exec	Remote execution.
100018	Alice_O/A	Alice office automation program.
100019	Scheduler	Scheduling service.
100020	Local_Lock	Local lock manager.
100021	NetworkLock	Network lock manager.
100022	X.25_INR	X.25 INR protocol.
100023	StatusMon_1	Status monitor 1.
100024	StatusMon_2	Status monitor 2.
100025	Select_Lib	Selection library.
100026	BootService	Boot parameters service.
100027	Mazewars	Mazewars game.
100028	YP_Updates	Yellow pages update service.
100029	Key_Server	Key server.
100030	SecureLogin	Secure login service.
100031	NFS_FwdInit	NFS network forwarding service.
100032	NFS_FwdTrns	NFS forwarding transmitter.
100033	SunLink_MAP	Sunlink MAP.
100034	Net_Monitor	Network monitor.

Number	Program	Explanation
100035	DataBase	Lightweight database.
100036	Passwd_Auth	Password authorization.
100037	TFS_Service	Translucent file service.
100038	NSE_Service	NSE server.
100039	NSE_Daemon	NSE activate process.
150001	PCPaswdAuth	PC password authorization.
200000	PyramidLock	Pyramid lock service.
200001	PyramidSys5	Pyramid Sys5 service.
200002	CADDS_Image	CV CADDS images.
300001	ADTFileLock	ADT file locking service.

Version

The version number of the remote program.

Procedure

The remote procedure being used.

Status

This field makes the identification of error messages possible.

Message body

{call successful}Call completed with no errors.{program unknown}Program number was not found.{bad program ver}Program version was not found.{proced unknown}Program procedure was not found.{bad parameters}Invalid call parameters were found.{bad RPC version}RPC Version not supported.

RPC reply frames can contain the following messages:

{bad credentials} Invalid call credentials supplied. {restart session} Request to begin a new session. {bad verifier} Invalid verifier was supplied. {verify rejected} Verifier has expired or was re-used. {failed security} Caller has insufficient privileges.

YP (NIS)

The Sun Yellow Pages (YP) protocol, now known as Network Information Service (NIS), is a directory service used for name look-up and general table enumeration. Each YP database consists of key-value pairs, maps, and domains. The key is the index, e.g., name to which YP maps values such as a phone number. Because keys and values mapped to them can be any arbitrary string of bytes, a YP server can contain a wide variety of database types.

YP defines a set of key-value pairs as a map. Each map belongs to a domain that is a category of maps. This hierarchy of key-value pairs, maps, and domains provides a generic structure for modeling a database of information.

An optional component to a YP server database implementation is the YP binder (YPbind) server. YP uses YP binder servers to provide addressing information about YP database servers to potential clients. A client desiring access to a YP database server can call the YP binder service with a domain name and receive the IP address and transport port of the YP server for that domain. If clients can more easily obtain this information elsewhere, the YP binder service is not needed.

Frames

YP and YPbind frames may be one of the following commands:

[no operation] Performs no operation.

Asks whether or not the specified domain is [domain serve query1]

served.

[domain serve query2] Asks only for servers that serve the

specified domain.

Asks for value associated with the specified [get key value]

[get first key pair] Requests the first key-value pair in map. Requests the next key-value pair in map. [get next key pair] [transfer map] Requests a new copy of the map to be

transferred.

[reset YP server] Requests the YP server to reset its internal

state.

r 11 1 · 1	D 11.1 1 · · · · · · · · · · · · · · · ·
loot all keys in man	Requests all key value bare in specified
[get all keys in map]	Requests all key-value pairs in specified

map.

Requests the name of master YP database get map master name

[get map number] Requests the creation time of the specified

map.

[get all maps] Requests all maps in the specified domain.

[domain serve reply1] Response to domain serve query1. [give key value] Returns value for specified key.

[give first key pair] Returns the first key-value pair in map. [give next key pair] Returns the next key-value pair in map.

[map transferred] Reports map transfer status. [YP server reset] Reports server reset status. [give all keys in map] Returns listing of all keys in map. Returns name of master YP database [give map master name]

[give map number] Returns creation date of map.

[give all maps] Returns listing of all maps in domain.

[no operation] Performs no operation.

Requests YP addressing information for the [get current binding]

specified domain.

[set domain binding] Installs YP addressing information for the

specified domain.

Returns YP addressing information for a [give current binding]

domain.

[domain binding set] Returns status of YP addressing installation.

Frame Parameters

YP and YPbind frames can contain the following parameters:

Bind address

IP address of the YP binder server.

Bind port

Transport port used by the YP binder server.

Created

Creation time of the map.

Domain

Domain name in use.

Key

Key index used to search for a value.

Map

Map name in use.

Master

Name of the master YP database server.

Peer

Server name of a peer YP server.

Transfer ID

Transfer ID used to reference map transfers.

Program

RPC program number used for map transfer.

Port

Transport layer port number used for map transfer.

Value

Value associated with a key.

Status

Map transfer status.

Version

YPbind protocol version.

Response Frames

YP response frames can contain the following status messages:

{OK}Request completed successfully.{Bad request arguments}Request parameters invalid.

{Domain not supported} Domain not supported by this YP server.

{General failure} Unspecified failure. {Invalid operation} Request invalid. {OK}

{No more entries in map}
 {No more key-value pairs in map.
 {No such map in domain}
 {No such key in map}
 {Server database is bad}
 {YP server error}
 No more key-value pairs in map.
 Specified map not in domain.
 Specified key not in map.
 Server database corrupt.
 Internal YP server error.

{YP version mismatch} YP server versions do not match.

YPbind response frames can contain the following messages:

{Internal error} Local YP binder error.

{No bound server for domain} No YP database servers known for the

domain.

Request completed successfully.

{Can't alloc system resource} YP binder resource error.

Transfer Status Frames

YPbind transfer status frames can contain the following messages:

{Transfer successful}
{Bad request arguments}
{Can't clear YP server}
{Can't find server f/map}
{Can't get master addr.}
{Domain not supported}

Transfer completed successfully.
Request parameters invalid.
Cannot clear the local YP server.
Cannot find YP server for map.
Cannot get YP master server address.
Domain not supported by this YP server.

{Local database failure} Local YP server database failure. {Local file I/O error} Local YP server file I/O error. {Map version mismatch} Map versions skewed in transfer. {Master dbase not newer} Master database is not newer. {Must override defaults} Must override default settings. {Resource alloc failure} Resource allocation failure. {RPC to server failed} No RPC response from server. {Server/map dbase error} YP server or map database error. YP server refused to transfer database. {Server refused transfer}

{YP transfer error} Error occurred during database transfer.