EMC²
TECHNICAL NOTES

EMC® NetWorker®

Configuring TCP Networks and Network Firewalls for EMC NetWorker

P/N 300-005-739
REV A08
April, 2011

This technical note explains best practices on TCP network configuration, how to identify and configure the required ports for NetWorker hosts that need to communicate across a packet filtering or statefull inspection firewall, and how to troubleshoot communication issues between NetWorker hosts.

The following topics are covered:

Terminology

NetWorker host — A NetWorker server, storage node, or client.

Service port — A port on which a server process listens for requests to provide a service. Service ports are also known as *listen* ports, *SYN* ports, or destination ports. For example, an HTTP server typically uses port 80 as a service port and an SSH server typically uses port 22 as a service port. If a firewall is not configured to allow clients to make connections to a service port, the service will be blocked.

> **Service port = target port = destination port = listen port = inbound port**

Connection port — A port used by a process to make requests. Connection ports are also known as communication ports, source ports, or outbound ports. Typically, firewalls do not block source ports because they only affect performance, not security. For example, an SSH server will allow an SSH client to initiate a session on service port 22. All firewall administrators know that port 22 is the service port for SSH. However, firewall administrators will not know which port will be used as the connection or source port by the SSH client for network traffic after a session has been initiated. Most programs, such as SSH, do not allow configuration of connection ports, but instead take connection ports directly from the operating system. However, NetWorker allows configuration of connection port ranges and this sometimes leads to confusion.

> **Connection port = source port = outbound port**

Direct connect communication model

NetWorker has a different architectural approach to network communication compared to many other backup applications in the market. NetWorker has been architected to move data efficiently across the network, with minimal overhead, resulting in efficient use of equipment.

To this end, NetWorker uses a direct connect communication. When NetWorker needs to move significant amounts of data, it opens a direct socket connection to the required service. For example, during a backup, the save operation reads data from the client and sends the data to the media-controlling `nsrmmd` service. In this case, NetWorker opens a direct socket connection to the required service. Thus, save will connect directly to the `nsrmmd` service, to the client file index service to store the indexes, and so on. In fact, with the advent of "immediate save," NetWorker has optimizations to bypass the network entirely if save and nsrmmd are on the same system—the two will communicate over shared memory. NetWorker PowerSnap™ can proxy backups to offload the application host entirely, and so on. NetWorker has many performance optimizations.

This is in marked contrast to some of NetWorker's competitors that use an indirect connect or multihop architecture. In NetWorker terms, such an architecture places a traffic aggregator/expander process on the client and storage node, so that communication flows from save to aggregator, then across the network to the expander, and then from the expander to nsrmmd.

This multihop architecture, which introduces two new processes into all communication channels to force every message to make three hops instead of one, is more firewall-friendly. By creating a single tunnel, such a solution reduces the firewall requirements to a single port, whereas the more efficient NetWorker direct connect approach uses ports more freely to deliver optimum performance.

NetWorker connection flow

NetWorker connection flow

Default NetWorker configuration results in the following connection flow for scheduled backups:

```
server:conn->client:7938 (nsrrpc)
server:conn->client:rpc/390113 (nsrexecd/7937)
server:conn->client:svc (save)
client:conn->server:7938 (nsrrpc)
client:conn->server:rpc/390119 (nsrexecd/7937)
client:conn->server:rpc/390436 (nsrauth/svc)
client:conn->server:rpc/390103 (nsrd/svc)
client:conn->server:rpc/390104 (nsrmmd/svc)
client:conn->server:rpc/390105 (nsrindexd/svc)
```

Note: This shows a file-system backup with target device residing directly on the backup server. When the backup stream is directed to storage node, the above example is expanded to include negotiation and communication toward the storage node as well as negotiation and communication from the storage node to the backup server.

Calculating and configuring port ranges

Reserved ports

The NetWorker host performs backups and recoveries by using a number of TCP ports (service and connection ports). Two of the TCP ports, 7937 and 7938, are reserved by the NetWorker host and are used as follows:

- Port 7937 as a service port for the `nsrexecd` daemon.
- Port 7938 as a service port for the EMC® NetWorker portmapper.

The EMC NetWorker portmapper (also referred to as `lgtomapper`) runs as a thread of the `nsrexecd` daemon.

In addition, port 514 is used as a fallback connection if communication with nsrexecd cannot be established. To avoid potentially slow performance with the connection, ensure that port 514 is not blocked.

Setting port ranges in NetWorker

To modify the NetWorker service and connection port ranges, use NetWorker Management Console or the nsrports command.

To use the nsrports command to configure the connection port and service port ranges, type:

```
# nsrports -s server [-S|-C] range
```

The following table contains the options for the **nsrports** command.

Option	Description
-s server	Specifies the system (a client or server) to contact.
-S	Sets the system's service ports range to the specified range.
-C	Sets the system's connection ports range to the specified range.

Calculating connection port ranges

In NetWorker 7.2 and earlier, the default value for this range is: 10001-30000. In NetWorker 7.3 and higher, the default value is 0-0. The 0-0 value has a special meaning: NetWorker allows the OS to select the port for TCP clients. Entering 0-0 is only allowed for NetWorker 7.3 and later.

From a NetWorker perspective, one connection port is required for any type of communication between the client, storage node and server. However, calculating the minimum required connection port range does not rely only on NetWorker operations because such ports are reserved for short-term re-use by the operating system. So depending on the specific operating system and operating system configuration, the number of required connection ports is always higher than highest number of parallel connections.

It is best to keep the connection port range as wide as possible as there is no security concern. However, if the range is too narrow, then one may see performance problems, or random malfunctions of the NetWorker product.

Calculating service port ranges

When NetWorker services start, they attempt to listen only in the service port range that is specified for that host. NetWorker processes attempt to connect to a service by using connection (or source) ports from the connection port range:

- Service port ranges correspond to TCP listen ports
- Connection port ranges correspond to TCP source ports

The NetWorker services and processes running on NetWorker servers, clients, and storage nodes listen and connect only on the specified port ranges. The minimum number of ports depends on the NetWorker configuration.

NetWorker client

A NetWorker 7.3 or later client uses `nsrexecd` that requires four service ports: the reserved ports 7937 and 7938 and two user-configurable ports from the service port range.

If the client uses NetWorker add-on products, additional ports may be required. The NetWorker add-on product documentation provides more information about those modules.

For example, the client connection wizard speeds the process by which a NetWorker client is configured for backup. Prior to NetWorker release 7.3, the client configuration wizard was shipped as a separate add-on product. With NetWorker 7.3 it became a part of the main NetWorker product, like the NetWorker Management Console. The NetWorker Client Configuration wizard uses one port for each open user interface on the NetWorker client that is being configured.

This is a dynamic port that is closed when the wizard is closed. These ports are selected from the port range configured using nsrports.

As a result, a client requires a minimum of four service ports.

NetWorker storage node

A NetWorker storage node (SN) is also a NetWorker client, and so it uses all of the ports for a client.

In addition to the four ports for a client, a storage node requires ports for nsrmmd and nsrlcpd daemons.

There is one nsrmmd per tape or file device on the machine to handle backup and recover data. An advanced file type device counts as two devices as it creates a read-only device for simultaneous restores, and thus has two nsrmmd. When spanning from one device to another, a helper nsrmmd is launched to mount the new tape. Helper nsrmmd also require a port. There can be up to two mmd per device on a system.

There is one nsrlcpd per robot in an autochanger.

As a result, a storage node requires a minimum of:
4 + (2 * #devices) + (#jukeboxes) service ports.

NetWorker server

A NetWorker server is also a NetWorker storage node, and so it uses all of the ports for a storage node.

In addition to the ports for a storage node, a server requires ports for nsrd, nsmmdbd, nsrindexd, nsrmmgd, and nsrjobd daemons. Each of these requires a TCP/IP port.

The nsrd and nsrmmgd daemons also require a UDP port. These ports do not need to be accessed from outside the firewall, but they come from the same service port range as the other ports.

As a result, a NetWorker 7.3.x server requires a minimum of:
11 + (2 * #devices) + (#jukeboxes) service ports.

NetWorker 7.4 introduces a new daemon, the client push daemon, which also consumes a TCP service port. As a result, a NetWorker 7.4 server requires a minimum of:
12 + (2 * #devices) + (#jukeboxes) service ports

The number of devices and jukeboxes in the formula are devices and jukeboxes on the NetWorker server. *Do not* count devices and jukeboxes on other storage nodes in the server's data zone.

Note: Different rules apply for older versions of the NetWorker software, or sites that have mixed NetWorker versions. The *EMC NetWorker Administration Guide* provides more details.

NetWorker Management Console

The Console server component of NMC uses 3 ports:

- One port (9000 by default) is used for the web server, which provides a way to download the java application code that acts as the Console front end. This port is selected during the installation process.
- The second port (9001 by default) is used for RPC calls from the Console Java client to the Console server. This port is selected during the installation process.
- The last port (2638 by default) is used for database queries.

NetWorker Management Console ports are *not* taken from the range configured using `nsrports` and are used only for communication between client workstations (systems running the Java GUI) and the NMC server.

The Console server communicates to the NetWorker server using service ports from the standard NetWorker range (as defined by `nsrports`).

Example A: Calculating service ports on a bidirectional firewall

This example shows how to apply the basic rules for a sample network with NetWorker clients A, B, C ..., NetWorker storage nodes X and Y, and a NetWorker server Z, with a single firewall that blocks both ways. The firewall in this example sits between the NetWorker server on the one side, and the clients and storage nodes on the other. Each storage node and the NetWorker server have a tape library and six drives, and there are no pre-NW 7.3 clients. There are no modules, and no requirement to use NMC or the client configuration wizard across the firewall. The hosts table looks as follows:

```
192.167.10.101       client_A
192.167.10.102       client_B
192.167.10.103       client_C
192.167.10.104       client_D
192.167.10.105       client_E
192.167.10.106       client_F
192.167.10.107       client_G
192.167.10.108       client_H
# ...
196.167.10.124       storage_node_X
192.167.10.125       storage_node_Y
192.167.10.126       NW_server_Z
```

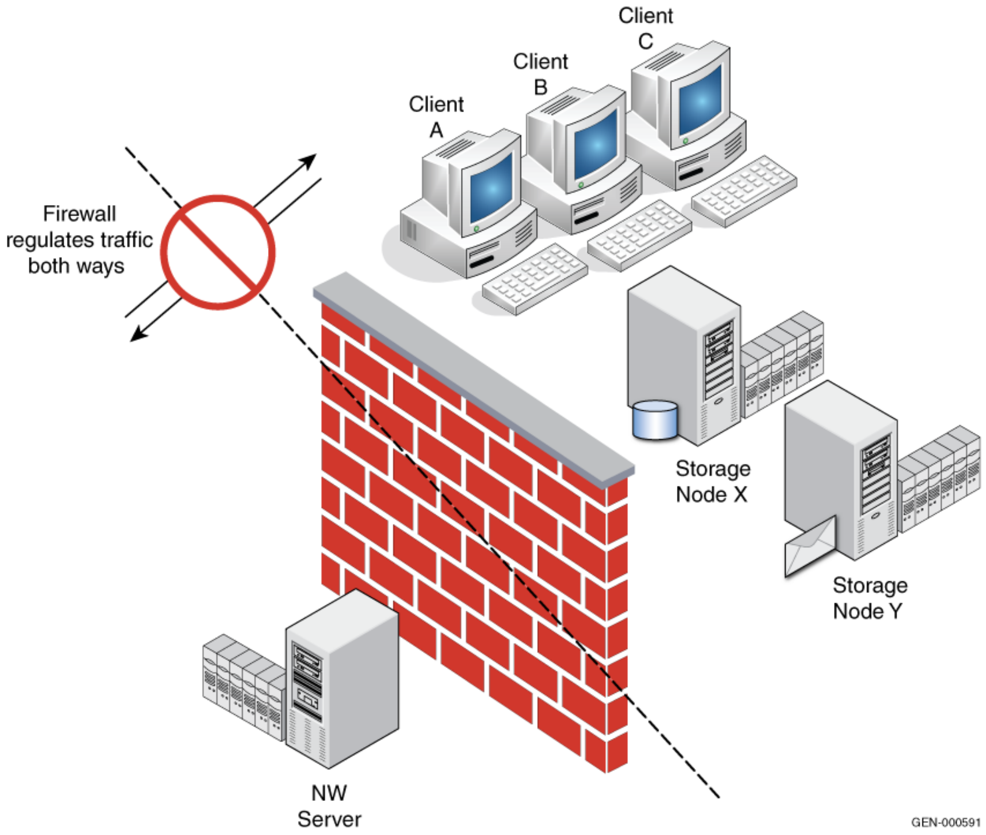

Figure 1 Service port ranges on a bi-directional firewall

The firewall in Figure 1 is bidirectional, blocking traffic from right to left as well as from left to right.

There is a NetWorker 7.3.x server on the left of the firewall. It has six devices in one library; hence, it needs **11 + 2 * (num devices) + (num libraries) = 24** service ports. Two ports must be 7937 and 7938, for example, select ports 7937–7960. A NetWorker 7.4 server would require one additional port to accommodate the client push daemon.

The NetWorker server must be configured to use 24 service ports, 7937–7960, and the firewall must allow traffic leftward (to the NetWorker server's IP address) on all the service ports configured. In pseudo syntax, the firewall rule for the service ports would be:

```
TCP, Service, src 192.167.10.*, dest 192.167.10.126, ports 7937-7960, action accept
```

There are NetWorker storage nodes on the right of the firewall. Storage node X has six devices and one library. So it needs **4 + 2 * (num devices) + (num libraries) = 17** service ports. Two ports must be 7937 and 7938, so, for example, select ports 7937–7953. Storage node Y is identical, and needs the same number of ports. It can use the same port range as well, 7937–7953.

Thus, each NetWorker SN must be configured to use 17 service ports, 7939–7953, and the firewall must allow traffic rightward (to each SN's IP address) on all the service ports configured. Because each SN can use the same port numbers, the firewall only needs to allow 17 ports for both storage node IP addresses. Configure all storage nodes to use the same 17 service port numbers to keep things simpler. These port numbers can be a subset of the NetWorker server's port numbers, as in this example. In pseudo syntax, the firewall rule for the service ports would be:

```
TCP, Service, src 192.167.10.*, dest 192.167.10.124, ports 7937-7953, action accept
TCP, Service, src 192.167.10.*, dest 192.167.10.125, ports 7937-7953, action accept
```

There are also NetWorker clients on the right of the firewall. Client A needs four service ports. Two ports must be 7937 and 7938, so, for example, select ports 7937–7940. Clients B and C have the same port requirements.

Configure each client to use at least four service ports, 7937–7940, and configure the firewall to allow traffic rightward (to each client's IP address) on all of the service ports configured. Configure all clients to use the same four service port numbers to keep things simpler. If these port numbers are a subset of the server's port numbers, as in this example, so much the better.   In pseudo syntax, the firewall rule for the service ports would be:

```
TCP, Service, src 192.167.10.*, dest 192.167.10.101, ports 7937-7940, action accept
TCP, Service, src 192.167.10.*, dest 192.167.10.102, ports 7937-7940, action accept
TCP, Service, src 192.167.10.*, dest 192.167.10.103, ports 7937-7940, action accept
...
```

Note: This example ignores connection (source) ports. They are not a concern, just as a firewall administrator assigned to configure an SSH server would configure port 22, the service port, but not the unknown connection ports the SSH clients use for SSH sessions. Connection ports affect performance (multiplexing into fewer ports reduces performance), but do not cause security concerns.

In the previous pseudo syntax, the firewall is configured to allow incoming service connections to the NetWorker server’s IP address on ports 7937–7960, from the IP addresses of each of the storage nodes or client machines (as well as any other machines on that subnet). The firewall is also configured to allow connections to the IP addresses for each storage node on ports 7937–7953, and to each client IP address on ports 7937–7940. Each NetWorker host must be configured with the appropriate port range for that machine, and the NetWorker services must be restarted on each machine after a change to the port range is made. This is the “tightest” configuration possible, but it requires work-specific configuration to each machine.

A simpler configuration to administer these machines would be to assign a range of 24 ports, 7937–7960, to all machines, and configure the firewall to allow traffic to these ports on any host, from any host. In pseudo syntax, this is:

```
TCP, Service, src 192.167.10.*, dest 192.167.10.*, ports 7937-7960, action accept
```

If the single firewall in the previous example were replaced with two firewalls, each containing a storage node and some clients, each machine would be configured exactly as before, and each firewall would be configured identically.

Example B: Calculating service ports on a unidirectional firewall

In this example, the NetWorker hosts in Figure 2 use the same IP addresses as in Figure 1. The difference in this example is that one NetWorker storage node is on either side of the firewall. NetWorker clients on the left side of the firewall back up data to the storage node on the left, and clients on the right side back up data to the storage node on the right. The clients on the right side of the firewall are in a demilitarized zone (DMZ). Everything to the left of the firewall is protected and trusted. Everything to the right of the firewall is not protected and cannot be trusted. Therefore, the firewall must block network traffic from right to left.

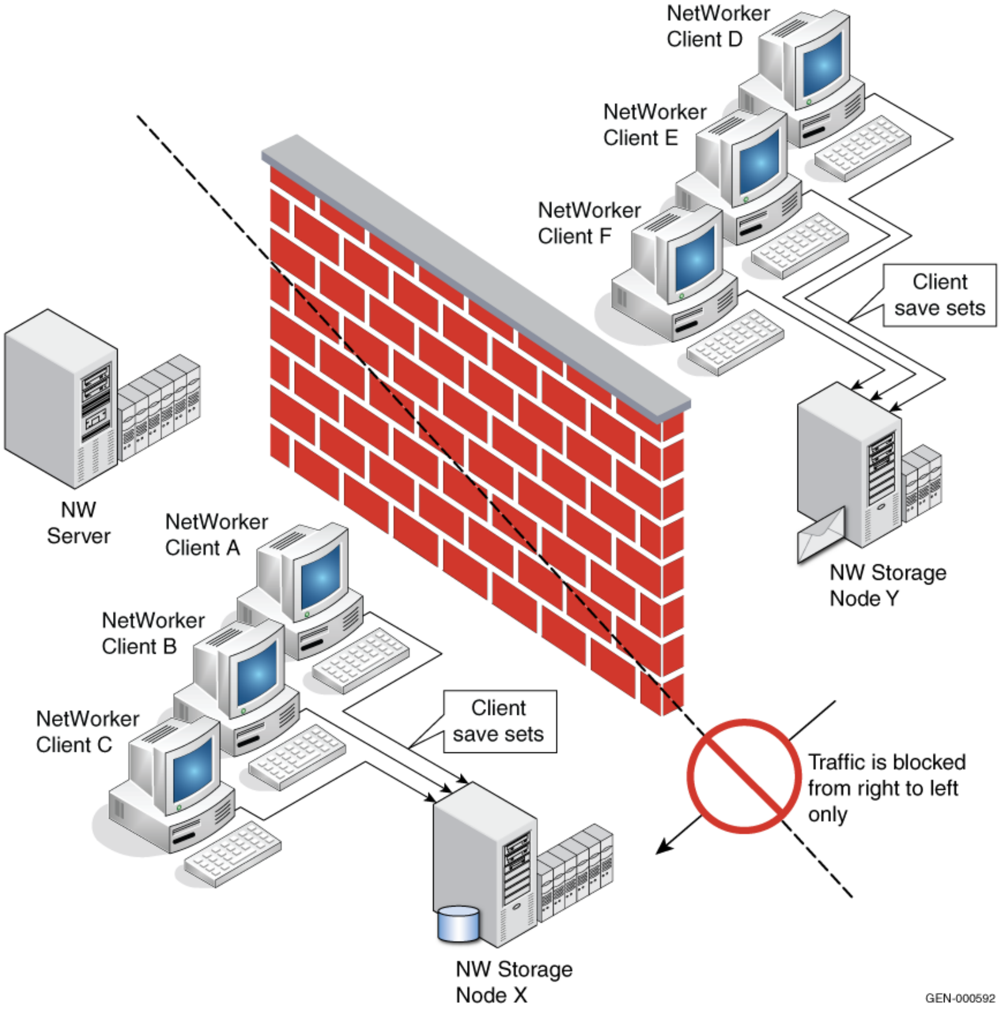

Figure 2 Service port ranges on a unidirectional firewall

Open the same ports on the NetWorker hosts as described in example A. However, the only ports that must be opened on the firewall are the service ports for the NetWorker server, so that traffic can flow leftward. In pseudo syntax, the firewall rule for the service ports on the NetWorker server would be:

```
TCP, Service, src 192.167.10.104, dest 192.167.10.126, ports 7937-7960, action accept
TCP, Service, src 192.167.10.105, dest 192.167.10.126, ports 7937-7960, action accept
TCP, Service, src 192.167.10.106, dest 192.167.10.126, ports 7937-7960, action accept
TCP, Service, src 192.167.10.125, dest 192.167.10.126, ports 7937-7960, action accept
```

Wildcards can also be used to specify all hosts in the DMZ, for example:

```
TCP, Service, src 192.167.10.*, dest 192.167.10.126, ports 7937-7960, action accept
```

Example C: Calculating service ports in a multihomed environment

Figure 3 in this example depicts an environment where each NetWorker client uses two NetWork Interface Cards (NICs). One card carries regular production network traffic, while the other card carries backup data (control data and savestream data). This is known as a multihomed solution. A multihomed solution can increase backup speed by providing a dedicated network for backup data.

In this example, a storage node is set up on each side of the firewall so that client savestreams do not have to pass through the firewall. The only data that must pass through the firewall is control data between the NetWorker server, clients, and storage node. In this way, the firewall is relieved from the burden of processing data-rich savestreams.

On the left side of the firewall, clients A, B, and C send their save sets to storage node X. On the right side, clients D, E, and F send their save sets to storage node Y. The NetWorker server and storage node X each have a tape library and six drives. Storage node Y has a tape library with only three drives. The hosts table looks like this:

```
192.167.10.101      client_A1 # NIC 1 for production network
192.167.12.101      client_A2 # NIC 2 for backup network
192.167.10.102      client_B1 # NIC 1 prod. network
192.167.12.102      client_B2 # NIC 2 backup network
192.167.10.103      client_C1 # NIC 1 prod. network
192.167.12.103      client_C2 # NIC 2 backup network
192.167.10.104      client_D1 # NIC 1 prod. network
192.167.12.104      client_D2 # NIC 2 backup network
192.167.10.105      client_E1 # NIC 1 prod. network
192.167.12.105      client_E2 # NIC 2 backup network
192.167.10.106      client_F1 # NIC 1 prod. network
192.167.12.106      client_F2 # NIC 2 backup network

196.167.10.124      storage_node_X1  # NIC 1 prod. network
196.167.12.124      storage_node_X2  # NIC 2 backup network
192.167.10.125      storage_node_Y1  # NIC 1 prod. network
196.167.12.125      storage_node_Y2  # NIC 2 backup network

192.167.10.126      NW_server_Z
```

As in example A, the firewall is bidirectional, blocking traffic from right to left as well as left to right. Additionally, there are no pre-NW 7.3 clients, no modules, and no requirement to use NMC or the client configuration wizard across the firewall.

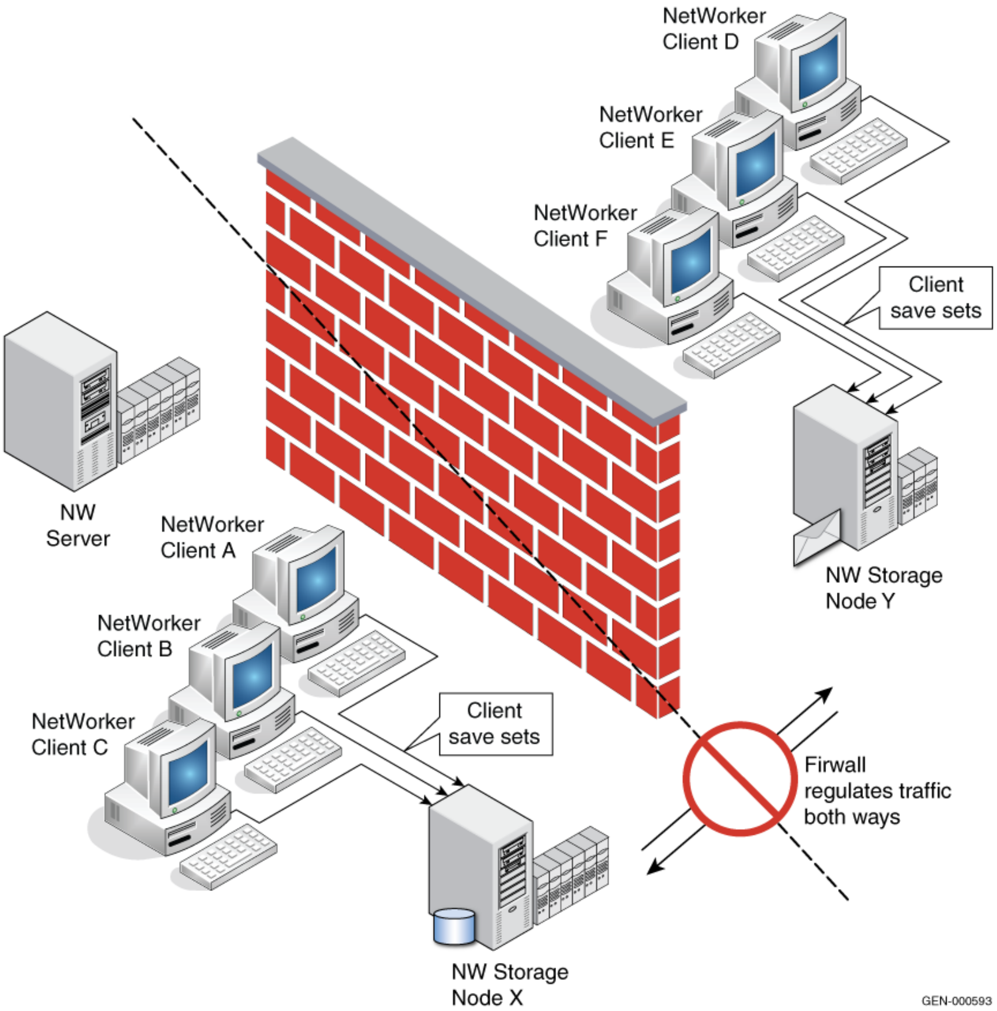

Figure 3 Service port ranges in a multihomed environment

As with example A, the NetWorker 7.3 server has 6 devices in one library; hence, it needs **11 + 2 * (num devices) + (num libraries) = 24** service ports. Two ports must be 7937 and 7938, for example, select ports 7937–7960. A NetWorker 7.4 server would require one additional port to accommodate the client push daemon

NetWorker requires that all hostnames (both short and fully qualified) for all NICS on a host are entered in an alias list for that host. The alias list for a host is set up in the NetWorker client resource. If the storage node itself is not an active client, you must still create a NetWorker client resource and type aliases for all hostnames used by the storage node.

To direct savestream data over a second NIC, type the hostname that resolves to the NIC in the **Storage Nodes** attribute that is located in the NetWorker client resource for each client using this storage node.

Note: All clients with the same name will use same Storage Node attribute.

Although the actual storage node can be configured using a different hostname (normally from the primary NIC), the NetWorker server will be able to correctly match entries because hostname aliases were entered. If the Storage Node Preference attribute is not specified, NetWorker will use the first NIC that properly routes packets to the primary name of the storage node.

To direct NetWorker control data over a specific NIC, enter the NetWorker server's hostname that resolves to that NIC in the **Server Network Interface** attribute for each client that has a route defined to the NetWorker server specific NIC. The Server Network Interface attribute is located in the client resource. You can also specify a specific NIC for a device. If this value is not specified, NetWorker will use the first NIC that properly routes packets to the primary name of the NetWorker server.

When it is not possible to resolve all client hostnames by the backup server (a typical case when clients use NAT—Network Address Translation), create entries in the NetWorker servers' hosts table so that all hostnames point to the one that is valid. This is required because NetWorker attempts to authenticate all hostnames for a specific client and if some routes are not resolvable, backup will fail.

The firewall must allow backup network traffic leftward (to the NetWorker server's IP address) on all 24 service ports for clients D, E, F, and storage node Y. In pseudo syntax, the firewall rule for the service ports would be:

```
TCP, Service, src 192.167.12.104, dest 192.167.10.126, ports 7937-7960, action accept
TCP, Service, src 192.167.12.105, dest 192.167.10.126, ports 7937-7960, action accept
TCP, Service, src 192.167.12.106, dest 192.167.10.126, ports 7937-7960, action accept
TCP, Service, src 192.167.12.125, dest 192.167.10.126, ports 7937-7960, action accept
```

Storage node Y is on the right of the firewall and it has one library with three devices. So it needs **4 + 2 * (num devices) + (num libraries) = 11** service ports. Two ports must be 7937 and 7938, so, for example, select ports 7937–7947.

Configure the storage node to use at least 11 service ports, 7937–7947, and configure the firewall to allow traffic rightward (from the NetWorker server to the storage node's "NIC 2" IP address). In pseudo syntax, the firewall rule for the service ports would be:

```
TCP, Service, src 192.167.10.126, dest 192.167.12.125, ports 7937-7947, action accept
```

There are also NetWorker clients on the right of the firewall. Clients D, E, and F each require 4 service ports. Two ports must be 7937 and 7938, so, for example, select ports 7937–7940.

Configure each client to use at least four service ports, 7937–7940, and configure the firewall to allow traffic rightward (to each client's NIC 1 IP address) on all of the service ports configured. In pseudo syntax, the firewall rule for the service ports would be:

```
TCP, Service, src 192.167.10.126, dest 192.167.12.104, ports 7937-7940, action accept
TCP, Service, src 192.167.10.126, dest 192.167.12.105, ports 7937-7940, action accept
TCP, Service, src 192.167.10.126, dest 192.167.12.106, ports 7937-7940, action accept
```

Configuring RPC

NetWorker requires a fully functional RPC portmapper service (otherwise known as rpcbind) to discover available program services and their current connection points. NetWorker can utilize either the default operating system SunRPC portmapper on port 111 (if present) or internal NsrRPC portmapper available inside the nsrexecd process (by default on port 7938).

If not explicitly specified, the order of initial RPC connections (SunRPC or NsrRPC) is decided by operating system.

Note that SunRPC portmapper is not required for NetWorker operations as full functionality is provided by NsrRPC, but if SunRPC is actively blocked by a firewall rule, it can cause delays on client/server connectivity as NetWorker has to wait for operating system timeout before attempting connection to NsrRPC.

If the NsrRPC portmapper cannot be reached on expected port, NetWorker will attempt fallback to SunRPC portmapper.

If RPC portmapper program is restarted, all applications registered to use RPC services must be restarted as well, because the registration information will be lost from the portmapper database. For example, if SunRPC service is present on the system and restarted, NetWorker must be restarted as well.

For dedicated NetWorker server hosts, it is recommended to disable the system SunRPC portmapper service (common process name is portmap or rpcbind).

Starting with NetWorker 7.5, a custom configuration on the operating system side can be used to ensure order of RPC services used. Custom configuration can be achieved by making sure that operating system is configured to lookup the services file for configuration and then by modifying that file as follows:

Services file entry	Action taken
sunrpc and nsrrpc	Value from nsrrpc is used for NetWorker portmapper. Default sunrpc portmapper is not used.
nsrrpc and no sunrpc	Value from nsrrpc is used for NetWorker portmapper. Default sunrpc portmapper is not used.
sunrpc and no nsrrpc	NetWorker portmapper used on default port (7938). Value from system portmapper sunrpc used as fallback. **This is default configuration Operating System configuration**
no sunrpc and no nsrrpc	NetWorker portmapper used on default port. System portmapper is not used.

Default location of services file is:

- On Unix/Linux: /etc/services
- On Windows: %SYSTEMROOT%\System32\Drivers\etc\services

Example services file:

```
sunrpc    111/tcp  rpcbind portmap #Sun RPC
sunrpc    111/udp  rpcbind portmap #Sun RPC
nsrrpc   7938/tcp  lgtomapper      #EMC NetWorker RPC
nsrrpc   7938/udp  lgtomapper      #EMC NetWorker RPC
```

Special use cases

Multihomed systems

In cases where the NetWorker server, storage node or client have more than one IP address, NetWorker advanced configuration can be used to specify the exact TCP/IP network path for data backup.

Note that a multihomed system is any system that has:

- More than one NIC, each having separate IP address.
- A single NIC but with multiple IP addresses.
- Multiple NICs in a single bond and then having multiple IP addresses.

Requirements for multihomed setup

- Each IP on any host must always resolve to a unique primary hostname. Per TCP RFC, it is not allowed to use same primary hostname for multiple NICs.
- Each IP bound to a separate physical NIC must reside in a separate subnet. Per TCP RFC, it is not allowed to have multiple IPs belonging to the same subnet and bound to the same NIC as it can break TCP session integrity due to undetermined outgoing route.
- Each hostname that belongs to any host (NetWorker server, storage node or client) should be entered in the Aliases attribute under client definitions in NetWorker.
- If NetWorker's servers file is utilized on NetWorker clients, all hostnames belonging to the NetWorker server should be entered.

Controlling data paths

- To connect from a NetWorker server to a NetWorker client over a specific NIC, configure the client using the name that is only reachable over the desired NIC.
- To direct client backup data to a NetWorker storage node over a specific NIC, set the Storage Nodes attribute in the Client resource to a desired hostname to that is only reachable over the desired NIC.

  Note that this also applies if the storage node is also a NetWorker server.
- To direct metadata from the client to the backup server over a specific NIC, set the Server NetWorker Interface attribute in the Client resource to a server name that is only reachable over the desired NIC.

Note that all instances of a target client resource must have the same value for the Server NetWorker Interface attribute. This metadata includes saveset control session information as well as all Index database operations.

The amount of data transferred over the network is proportionate to number of files being saved, and is much lower than the actual backup data.

This field is optional and has to be set correctly only if the client cannot communicate with backup server over the primary NIC.

- To direct metadata from devices on a storage node to a backup server over a specific NIC, set the Server NetWorker Interface attribute for the Device resource to a server name that is only reachable over the desired NIC. The amount of data transferred over network is proportionate to the size of savesets divided by target block size plus some overhead, but on average is very low.

  Note that all devices from the same storage node must have the same value for the Server NetWorker Interface attribute.

  This metadata includes device control session information as well as all Media database operations (connecting back to nsrmmdb).

  This field is optional and has to be set correctly only if the storage node cannot communicate with the backup server over the primary NIC.

- To direct metadata from a library that is controlled from the storage node to the backup server over a specific NIC, set the Server NetWorker Interface attribute for the Library resource to a server name that is only reachable over the desired NIC.

  This metadata includes SCSI commands for actual tape movements as well as library inventory operations (connecting back to nsrmmgd).

  The amount of data transferred over the network includes SCSI commands for library control as well as library inventory and status updates. On average this is very low.

  This field is optional and has to be set correctly only if the storage node cannot communicate with backup server over primary NIC.

Link aggregation

Link aggregation is a computer networking term which describes the use of multiple network interfaces in parallel to increase the link speed beyond the limits of any one single cable or port, and to increase redundancy for higher availability.

Other terms for link aggregation include "Ethernet trunk," "NIC teaming," "port channel," "port teaming," "port trunking", "link bundling," "EtherChannel," "Multi-Link Trunking (MLT)," "NIC bonding," and others.

Link aggregation on the TCP level, regardless of the protocol or algorithm used, has no effect on a single TCP session. Thus, combining multiple links into a single link will not increase the backup performance of a single session. Depending on the algorithm used, you will have a positive effect when executing parallel backup jobs with multiple NICs, which will act in load-balancing manner. To properly accomplish this, use a link aggregation algorithm which is TCP-session based and not host based: Best-practice would be the use of IEEE 803.3ad/802.1ax Link Aggregation Control Protocol (LACP).

The use of trunked interfaces is transparent from a NetWorker point of view and their configuration inside NetWorker does not differ from the configuration of standalone interfaces. Note that usage of TCP trunking can still be combined with multihoming as some NICs on the system can be trunked and some left working on separate subnets.

DHCP Clients

NetWorker relies on hostname and IP resolution, both forward and reverse, for communication to clients. In cases when the IP address changes due to DHCP allocation, NetWorker is not able to properly reverse-resolve the client's current IP address back to a valid hostname.

For backup of DHCP clients, possible solutions are:

1. Configure clients and DNS server to allow for Dynamic DNS Registration. The result of this is that each time a client is given a new IP address; it registers it with central DNS server so hostname resolving is fully functional. This is supported by all modern DNS servers (for example, Microsoft added support for dynamic DNS registration in Windows 2000 Server).
2. Configure the DHCP server to always issue the same IP address to clients. This can be achieved by using MAC address binding on the DHCP server. In addition, this IP address must be registered either in DNS server or both the client's and server's hosts file.

Performance tips

Default TCP parameters used on operating systems are tuned for maximum compatibility with legacy network infrastructures, but not for maximum performance.

Thus common throughput values are in the following ranges:

- 100Mbit link = 6-8 Mbytes/sec
- 1Gbit link = 45-65Mbytes/sec
- 10Gbit link = 200-350Mbytes/sec

But with optimized values, throughput for high-speed links can be increased to

- 100MBit link = 12 Mbytes/sec
- 1Gbit link = 90Mbytes/sec
- 10Gbit link = 700MBytses/sec

Common optimizations

Common rules for optimizing operating system TCP stack:

- Enable jumbo frames where possible.
- Disable software flow control
- Increase TCP buffer sizes

Network latency

Increased network TCP latency will have a negative impact on overall throughput, regardless of available link bandwidth. As a rule, a longer distance or a higher number of hops between network hosts results in lower overall throughput.

Ethernet Duplexing

NetWorker traffic flow is vulnerable to performance drops due to network links performing in half-duplex mode. This can lead to performance drop of several factors. For example, a 100MBit half-duplex link will result in backup performance of less than 1MByte/sec.

The common configuration settings on most operating systems for the duplexing parameter is Autonegotiated, as recommended by IEEE802.3. However, autonegotiation relies on proper cabling requirements as well as a compatible NIC adapter and switch. It is not uncommon that having autonegotiation enabled results in a link being negotiated as half duplex.

To avoid issues with the autonegotiation setting, force full-duplex settings on the NIC. In such cases, the forced full-duplex setting should be applied on both sides of the link, as having it forced on one side will result autonegotiation to fail on the other side.

Dropped TCP packets

Note that if the rate of inbound TCP packets is higher than the system can process, the operating system will drop some of the packets. This can lead to an undetermined NetWorker state and unreliable backup operations.

For NetWorker server or storage node systems equipped with high-speed interfaces, it is critical to monitor the system TCP statistics for dropped TCP packets, commonly done by using the netstat -s command.

To avoid dropped TCP packets, increase the TCP buffer size. Depending on the operating system, this parameter is referred as buffer size, queue size, hash size, backlog or connection depth.

Some examples:

Solaris

```
tcp_conn_req_max_q 8192
tcp_conn_req_max_q0 8192
tcp_max_buf 10485760
tcp_cwnd_max 10485760
tcp_recv_hiwat 65536
tcp_xmit_hiwat 65536
```

Linux

To modify the TCP buffer settings on Linux, add the following parameters to the /etc/sysctl.conf file and then run the **/sbin/sysctl -p** command:

```
net.core.rmem_default = 262144
net.core.wmem_default = 262144
net.core.rmem_max = 16777216
net.core.wmem_max = 16777216
net.ipv4.tcp_rmem = 8192 524288 16777216
net.ipv4.tcp_wmem = 8192 524288 16777216
```

Tru64

Modify /etc/sysconfigtab with following values:

```
socket:
  somaxconn = 65535
  sominconn = 65535
inet:
  tcbhashnum = 16
  tcbhashsize = 8192
```

Diagnostic tips

Before configuring NetWorker port ranges, consider the following:

- Allocate some extra service ports to accommodate growth. If a new drive is added to a storage node, will the people adding the drive remember to increase the port count by two in nsrports and the firewall?
- The `nsrexecd` daemon manages the NSR ports ranges resource. This daemon must be the first NetWorker daemon to start, as it does during system initialization. If the NetWorker software is manually started, be sure that the `nsrexecd` daemon is the first one started. If the `nsrexecd` daemon is not started first, ports may be assigned randomly.
- After changing the service or connection port ranges, restart the NetWorker software, including `nsrexecd`, and make any corresponding modifications to the firewall rules.
- Do not change the *TIME_WAIT* and *CLOSE_WAIT* intervals to reduce the port ranges demand. A change in the *TIME_WAIT* and *CLOSE_WAIT* intervals has minimal effect on the connection port ranges. *TIME_WAIT* and *CLOSE_WAIT* intervals that are set too low may corrupt new processes by packets that are re-sent to processes that have since exited.
- Use the **netstat -a** command to determine port allocation.
- The **`rpcinfo -p`** or **`ping`** commands may not always work across firewalls. RPC info requires connectivity using SunRPC on port 111, which is not required by NetWorker, while ping requires ICMP packets which may be blocked separately from TCP packets used by NetWorker.
- Use the **nsradmin** command to carry out limited testing of the client/server connectivity through firewall:
- To test the NetWorker server connection to the nsrexecd daemon running on the client, run the following command from the NetWorker server:
  ```
  nsradmin –s <client_name> -p 390113
  ```
- To test the NetWorker client connections to the `nsrd` and `nsrexecd` daemons on the backup server, run the following command from the NetWorker client:
  ```
  nsradmin –s <server_name>, nsradmin –s
  <server_name> -p 390113
  ```

- Maintain the connection port range for a NetWorker server, client, or storage node at the default range. In older versions of NetWorker, the default range was 10,001–30,000. NetWorker 7.3 or later can use a special range of 0–0 that lets the operating system pick the ports. These ports are used as connection ports only, and never as service ports.
- Define port ranges with the `nsrports` program, or some other technique from the *EMC NetWorker Multiplatform Version Administration Guide*. *Do not* modify the `nsr/res/nsrla.res` file directly.
- Do not assign ports from the reserved service port range (ports below 1024) in order to avoid conflict with other daemons or services on the host. Additionally, always place the starting point of the connection port range (if manual configuration is necessary) so that it starts after the range used by service ports for NetWorker or any other application.

Other applications

Although most EMC NetWorker application modules use the same port ranges as the NetWorker client, other EMC NetWorker family products may have their own reserved ports. This section contains a short summary, however consult the documentation for each product in your environment.

EMC Avamar

The EMC Avamar™ integration with NetWorker uses port 27000 (or 29000 if secure sockets layer – ssl is used). This port is used by the `nsravtar` command to perform backup operations between the NetWorker client and the Avamar DeDuplication storage node.

EMC AlphaStor

EMC AlphaStor™ uses ports 44444, 41025, 41114, 44460, and 44455.

NDMP

NDMP uses port 10000 as a target on a NAS filer, but does not require any standard NetWorker service ports on the client.

EMC SnapImage

The SnapImage data server uses two ports. The first port is port 10000 by default. The value for this port is configured in `/etc/services` (or equivalent). The port value can be changed at install time or by editing `/etc/services`.

Where the second port is configured and used, depends on whether you are using the DSA workflow. In both cases, this port is transient.

If the DSA workflow is used, then a port is needed on the NetWorker storage node. This port is selected from the port range configured using `nsrports`.

If the DSA workflow is not used, then a reserved port (from the range 0-1024) will be opened by the NDMP tape server.

Note: that the NDMP tape server is not a NetWorker or NetWorker add-on product. Refer to the NDMP tape server manual to determine which port to open. The NDMP tape server may or may not be installed on a system that is also running NetWorker. This port is not selected from the port range configured using `nsrports`.

EMC PowerSnap

EMC PowerSnap™ deploys a service (`nsrpsd`) on the NetWorker client host. This service operates in a similar manner as `nsrexecd`, that is, two instances of this service are active with one controlling the BRC protocol and the other performing the snapshot consistency check. Furthermore, PowerSnap creates a listening socket connection (service port) on the NetWorker client, to which the backup agent from the Proxy host connects. There are as many socket connections open on a NetWorker client as dictated by the value in the Application Information variable, NSR_PB_SAVE_PARALLELISM.

Sun StorageTek ACSLS

NetWorker communicates to ACSLS silos through a separate process that normally runs on the same host as the NetWorker server. This applies to physical ACSLS systems as well as VTLs that emulate ACLSL behavior.

- On Unix and Linux, an ssi process is delivered with NetWorker.
- On Windows, a separate package named “Library Attach” must be obtained from StorageTek

Communication between NetWorker and a particular instance of ssi can be configured using the following range (only one port is needed per ssi instance):

- 50004 (default for the first instance)
- 50011 to 50019
- 50021 to 50099

Communication between an instance of ssi and the ACSLS server is configurable as of ACSLS v7.1. For details, refer to ACSLS documentation.

Earlier versions of NetWorker

This section describes how to calculate service port ranges for earlier version of NetWorker.

Service Ports Range for NetWorker 7.1.x and Lower

The following section applies to NetWorker 7.1 and lower.

NetWorker Server

For NetWorker 7.1.x and lower, the following daemons are run on the server and only the server: `nsrd`, `nsrindexd`, `nsrmmdbd`. Each of these daemons uses one port. The daemon `nsrexecd` also runs on the NetWorker server and it uses two ports. The daemon `nsrmmd` will run on the machine that is responsible for managing the device, either the NetWorker server or storage node. One `nsrmmd` is started per device (except for an advanced file type device which has two `nsrmmd` daemons started). During a spanning operation, an additional helper `nsrmmd` is started. The helper `nsrmmd` uses one dynamic port. Since spanning can potentially take place on all devices at the same time, we say that the `nsrmmd` uses 2 ports per device configured on the host in question.

The `nsrexec` process uses one dynamic port per `nsrexec` running. The `nsrexec` is used to start processes, such as `savefs`, or `recover`, on the NetWorker client machine. Thus, there needs to be a number of dynamic ports for use by `nsrexec`; one port per remote process that is running. When a `savegrp` is started, `save` and `savefs` are spawned on the NetWorker clients being backed up. There could be as many `savefs` processes as there are NetWorker clients and there could be as many `save` processes as the server parallelism. So we say that the number of ports required by `nsrexec` is the server parallelism + the number of clients.

Unfortunately, the `nsrd` process uses one port for a UDP service. This port does not really need to be accessed from outside the firewall, but the port is taken from the service ports range configured using `nsrports`, so it needs to be added to the formula to calculate the number of ports needed in the service ports range.

The NetWorker server port usage formula is: **5 + 2 * #devices + P + C**

Where `P` is the server parallelism and `C` is the number of NetWorker clients.

Note that `#devices` includes only the devices configured on the NetWorker server.

NetWorker Storage Node

The `nsrexecd` and `nsrmmd` daemons are the only daemons that run on the storage node. The `nsrexecd` daemon (like the NetWorker server) uses two ports. The `nsrmmd` daemon uses 2 per device configured on the host in question.

The NetWorker 7.1.x port usage formula for a storage node is: **2 + 2 * #devices**

Note that `#devices` includes only the devices configured on the NetWorker storage node in question.

NetWorker Client

The `nsrexecd` daemon is the only daemon running on the NetWorker client and it still uses 2 ports. Thus the formula for port usage on the NetWorker client is: **2**

Service Ports Range for NetWorker 7.2.x

In 7.2.x, the dynamic ports used by `nsrexec` were removed, unless they are backing up an older client (NetWorker 7.1.x or lower).

NetWorker Server

The following daemons are run on the server and only the server: `nsrd`, `nsrindexd`, `nsrmmdbd`. Each of these daemons uses one port. The `nsrexecd` daemon also runs on the NetWorker server and it uses two ports. The `nsrmmd` daemon runs on the machine responsible for managing the device, either the NetWorker server or storage node. One `nsrmmd` is started per device (except for an advanced file type device which has two `nsrmmd`s started). During a spanning operation, an additional helper `nsrmmd` is started. This helper `nsrmmd` uses one dynamic port. Since spanning can potentially take place on all devices at the same time, we say that the `nsrmmd`s use 2 ports per device configured on the host in question.

Unfortunately, the `nsrd` process uses one port for a UDP service. This port does not really need to be accessed from outside the firewall, but the port is taken from the service ports range configured using `nsrports`, so it needs to be added to the formula to calculate the number of ports needed in the service ports range.

Thus the formula for how many ports the NetWorker server requires is: **6 + 2 * #devices**

Note that #devices includes only the devices configured on the NetWorker server.

NetWorker Storage Node

The nsrexecd and nsrmmd daemons are the only daemons that run on the storage node. The nsrexecd daemon (like on the server) uses two ports. The nsrmmd daemon uses 2 per device configured on the host in question. Thus the 7.1.x port usage formula is: **2 + 2 * #devices**

Note that #devices includes only the devices configured on the NetWorker storage node in question.

NetWorker Client

The nsrexecd daemon is the only daemon running on the NetWorker client and it still uses 2 ports. Thus the formula for port usage on the NetWorker client is: **2**

NetWorker Server in Mixed Environments

If your environment contains some current NetWorker clients and some earlier NetWorker client (7.1.x or earlier), you need the following additional ports: **C + P**

where P is the server parallelism or the number of 7.1.x clients (whichever is smaller), and C is the number of 7.1.x clients. These ports are dynamic ports used by nsrexec. One port per save, savefs, or directed recover running on a 7.1.x or earlier client will be used.

Troubleshooting

This section contains solutions to some common problems.

Backups appear to stop responding or slow down dramatically

If firewalls are configured to drop packets outside an allowed range, and the firewall configuration does not allow for proper NetWorker connectivity, NetWorker will not get proper notification that a connection is not possible. Another result of this situation is that socket connections may not close properly (they remain in TCP FIN_WAIT state) and NetWorker will require more ports for client connectivity.

Solution

The preferred solution is to configure the firewall to reject packets outside the allowed range. In this case, NetWorker will get an immediate notification of any connection failure and the remaining operations will continue.

If that is not possible, reduce the values for TCP timeouts on the NetWorker server's operating system to reduce the impact of the problem.

Fallback to RPC portmapper service on port 111

If NetWorker cannot connect to the default nsrexecd daemon on the target client, it will attempt to connect to the standard RPC portmapper service on port 111.

For connectivity towards the NetWorker storage node, this is default behavior as the NetWorker server checks if the appropriate services are already started and correctly registered.

Solution

Ensure that the nsrexecd daemon is connected and that it was the first NetWorker daemon to start.

If a fallback to system portmapper is causing delays in communication, it should be disabled. For details on how to disable fallback refer to the section Configuring RPC on page 15.

Cannot bind socket to connection port range

The message "Cannot bind socket to connection port range on system <*system_name*>" appears in the savegroup messages or in stdout during manual operations.

Solution

To resolve this issue:

1. Increase the connection (source) port range on the system.
2. Make a corresponding change in the firewall rules.

Services are attempting to use ports outside configured range

The message “Service using port outside of configured ranges” appears in the NetWorker Administrator window after the NetWorker server services start.

Note that communication between NetWorker processes on the same host does not follow defined rules. For example, NetWorker server daemons communicate internally outside of the defined port range so the firewall should never limit the range for TCP traffic inside a single system.

Solution

To resolve this issue:

1. Increase the service port range on the NetWorker Server.
2. Make a corresponding change in the firewall rules.

Timeout Issues

NetWorker uses persistent connections between daemons to transfer information as efficiently as possible. Connections are opened at the start of communication, and closed when communication is finished. For example, a running backup may have connections open to a nsrmmd with the backup data, nsrindexd with client file index information, and to nsrjobd for control and status information.

However, if a firewall arbitrarily cuts connections that seem idle, there will be a network error (transport layer problem) to NetWorker, exactly as if a network cable had been unplugged.

Solution

To prevent these problems, configure the firewall to not close idle connections, or have an operating system timeout long enough to accommodate the backup window.

The status connection to nsrjobd is frequently idle for most of the backup: if there are no error messages to report, the connection will not have traffic until the success message when the backup is done. As a result, this connection is the most common connection to be cut by an aggressive firewall.

If it is impossible to eliminate a firewall timeout, change the operating system's TCP keepalive interval on each machine if connections to other daemons time out.

All NetWorker services must be restarted once the setting is changed. Your operating system's documentation provides more information about setting the TCP keepalive value for your OS.

Operating System TCP Keep Alive

TCP/IP KeepAlive is defined in RFC 1122 and is an optional TCP feature present in all modern operating systems.

KeepAlive Parameters by Operating System

Operating System	Parameter wait time before probing the connection	Parameter interval between retry probes	Parameter maximum retry probes	Unit of measure
AIX	tcp_keepidle	tcp_keepintvl	n/a	half-seconds
HP-UX 11i	tcp_time_wait_interval	tcp_keepalive_interval	tcp_keepalives_kill (1)	milliseconds
Linux	tcp_keepalive_time	tcp_keepalive_intvl	tcp_keepalive_probes	Seconds
Solaris	tcp_time_wait_interval	tcp_keepalive_interval	n/a	milliseconds
Windows	KeepAliveTime	KeepAliveInterval	TcpMaxDataRetransmission	milliseconds

Recommended value for the Wait Before Probing and Interval Between Probes parameters is 57 minutes (the exact number depends on units of measure used on the operating system). These should be further decreased if network-enforced timeouts are shorter than the common one hour value.

There are no side-effects of enabling TCP KeepAlive or even setting it to very low values as the network overhead is minimal. Common values are:

```
57 min = 3420 seconds = 6840 half seconds = 3420000
milliseconds
```

Changing KeepAlive Parameters

AIX

To set the value temporarily until the computer is restarted:

```
# no -o <tcp_parameter>= <tcp _value>
```

To make the change permanent, add this commands to the TCP startup script, commonly/etc/rc.net.

HP-UX 11i

To set the value temporarily until the computer is restarted:

```
# ndd -set /dev/tcp <tcp_parameter> <tcp_value>
```

To make the change permanent, see the example in the file /etc/rc.config.d/nddconf.

Linux

To set the value temporarily until the computer is restarted:

```
# sysctl -w net.ipv4.<tcp_parameter> = <tcp_value>
```

To make the change permanent, update /etc/sysctl.conf with net.ipv4.<tcp_parameter> = <tcp_value> and issue one of the following commands:

- RHEL: chkconfig sysctl on
- SLES: chkconfig boot.sysctl on

Solaris

To set the value temporarily until the computer is restarted:

```
# ndd -set /dev/tcp <tcp_parameter> <tcp_value>
```

To make the change permanent, add this command to the TCP startup script, commonly /etc/rc2.d/S69inet.

Windows

Change the registry key:

```
HKLM\System\CurrentControlSet\Services\Tcpip\Paramete
rs\<tcp_parameter>:DWORD=<tcp_value>
```

After changing the registry key, you must restart the computer for the change to take effect.

Alternative approach to firewall configuration

*This example is provided as is and is not supported***.** In case of problems using this method, customers will be asked to use the standard firewall configuration procedures described previously in this document.

For firewalls that can perform service connection tracking, it is possible to use dynamic rules without opening any service port range on the firewall. This allows for maximum security since only valid RPC connections with a valid service ID will be allowed, regardless of port range.

Note: This example does not apply to connection ports.

Because each NetWorker daemon registers with the portmapper, service ID connection tracing is done based on the RPC program number.

Example

The following example is based on IP tables rules with L7 extensions to allow for service monitoring:

1. Enable RPC service monitoring on NetWorker portmapper:

Define NetWorker base communication

```
/etc/modprobe.conf
options ip_conntrack_rsh range=16383 ports=7937
options ip_conntrack_rpc_tcp nsrexec=7937 ports=7938
options ip_conntrack_rpc_udp ports=7938
options ipt_rpc ports=7938
```

And load modules

```
# modprobe ip_conntrack_rsh
# modprobe ip_conntrack_rpc_tcp
# modprobe ip_conntrack_rpc_udp
# modprobe ipt_rpc
```

2. Add initial NetWorker portmapper and client connectivity:

```
# iptables -A INPUT -j ACCEPT -p tcp -m state --state NEW -m tcp
--dport 7937
# iptables -A INPUT -j ACCEPT -p tcp -m state --state NEW -m tcp
--dport 7938
# iptables -A INPUT -j ACCEPT -p udp -m state --state NEW -m udp
--dport 7938
```

3. Allow all communication on valid sessions only:

```
# iptables -A INPUT -m state --state ESTABLISHED -j ACCEPT
# iptables -A INPUT -m state --state RELATED -j ACCEPT
```

4. Allow NetWorker daemons:

The following rule is sufficient for the standard backup server:

```
# iptables -A INPUT -m rpc --rpcs 390103, 390104, 390105,
390107, 390109, 390110, 390113 -j ACCEPT
# iptables -A INPUT -m rpc --rpcs 390115, 390120, 390402,
390433, 390435, 390436, 390109 -j ACCEPT
```

If there are additional services required for a backup running on the client or server, identify them using

```
# rpcinfo -p localhost
```

and looking at the program number for each nsr service.

For example, on the NetWorker client it would be just `nsrexecd`, so no additional rules are needed:

```
# rpcinfo -p <client_name>
program vers proto   port
390113    1   tcp   7937  nsrexecd
```

however, on a backup server:

```
# rpcinfo -p <server_name>
program vers proto   port
390103    2   tcp   8192  nsrd
390104  205   tcp   9847  nsrmmd
390105    5   tcp   9318  nsrindexd
390107    5   tcp   9882  nsrmmdbd
390109    2   tcp   8192  nsrstat
390110    1   tcp   8192  nsrjbd
390113    1   tcp   7937  nsrexecd
390115    1   tcp   9705  lgtolmd
390120    1   tcp   8192  nsrexecd
390402    1   tcp   9001  gstd
390433    1   tcp   9349  nsrjobd
390435    1   tcp   8070  nsrexecd
390436    1   tcp   8152  nsrd
390109    2   udp   9168  nsrstat
```

EMC believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

THE INFORMATION IN THIS PUBLICATION IS PROVIDED "AS IS." EMC CORPORATION MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WITH RESPECT TO THE INFORMATION IN THIS PUBLICATION, AND SPECIFICALLY DISCLAIMS IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.