

Routing Novell IPX Over Slow Serial Lines and SAP Management

Document ID: 10604

Introduction

Prerequisites

- Requirements

- Components Used

- Conventions

SAP Description

- SAP Interval

- Configurations

- NetWare SAP Restrictor

- SAP Filters

- Sample SAP Filters Configurations

- SAP Interpacket Gap

- I/O Queues

Related Information

Introduction

This document will help you through some of the difficulties in routing Novell IPX over slow serial lines. We'll give you background information about Service Advertising Protocol (SAP) as well as tricks for getting around some of the problems it raises.

Prerequisites

Requirements

There are no specific requirements for this document.

Components Used

This document is not restricted to specific software and hardware versions.

Conventions

Refer to Cisco Technical Tips Conventions for more information on document conventions.

SAP Description

The IPX protocol broadcasts all of its known services every one (1) minute. These services are contained in SAP frames. While the broadcast of these frames every one minute is fine in a Local Area Network (LAN), it is the downfall of this protocol in a Wide Area Network (WAN) environment.

SAP allows service-providing nodes (such as file servers, print servers, and gateway servers) to advertise their services and addresses, and it makes the process of adding and removing services on an internetwork dynamic. As servers are booted up, they advertise their services using SAP; when they are brought down,

other servers use SAP to indicate that those services are no longer available.

Through SAP, routers create and maintain a database of internetwork service information. This allows clients on the network to determine what services are available on the network and obtain the internetwork addresses of the nodes (servers) where they can access those services. This is important since a workstation cannot initiate a session with a file server without that server's address.

Because these broadcast frames are essential to the operation of Novell networking, the routers must forward them even though it is against common router theory to forward broadcasts. But because the router often routes over more than two interfaces, we must also compile the SAP information that we receive, and send the appropriate SAP updates out of each interface (just like Novell Servers).

Note: To see the number of Novell Services that your router is sending updates for, issue the **show ipx server EXEC** command at user mode (>) or enable mode (#). The **sh ipx serv** command displays servers in a different order than the actual internal order. To see the internal order, use the command **show ipx server unsort**. Servers are ordered by type, by route metric, and then by hop count as a tie breaker. A Get Nearest Server (GNS) request for a file server will choose the server at the top of its respective type portion of the table if all servers have the same metric and hop count. Whenever a file server is taken out of service and then put back in service sometime later, it will be put back in the SAP table at the top of the list for its respective type/metric.

There is a feature called "static SAP." With this feature you can add extra hops for the servers in order to enter the servers into the server table as an entry other than the first. This way you can put the servers which shouldn't respond to GNS requests further down in the server table. Routers respond to GNS requests with the first entry in the server table. So, the servers in the table with the extra hops won't be included in the GNS response from Cisco. A static SAP will not override the position of the service in the service table. The SAP table is ordered by routing metric (ticks) and then by the server hops field. The service table is by default displayed in a sorted order by type, metric, and name. To see which server is actually at the top of the list for a particular type, use the command **show ipx server unsort**.

Severe network performance problems can occur as the number of IPX services increases and the speed of the remote links to other networks decreases. Each SAP frame contains up to seven service entries, and each update is 64 bytes. This data, when combined with the IPX header information included in each frame, brings the size of each SAP frame to 480 bytes.

$7 * 64 = 448 + 32 \text{ (IPX header)} = 480 \text{ bytes per frame.}$

Therefore, if there were 500 services on an internetwork, then there are approximately seventy-two 480 byte frames per the one minute update period ($500 / 7 = 71.42$).

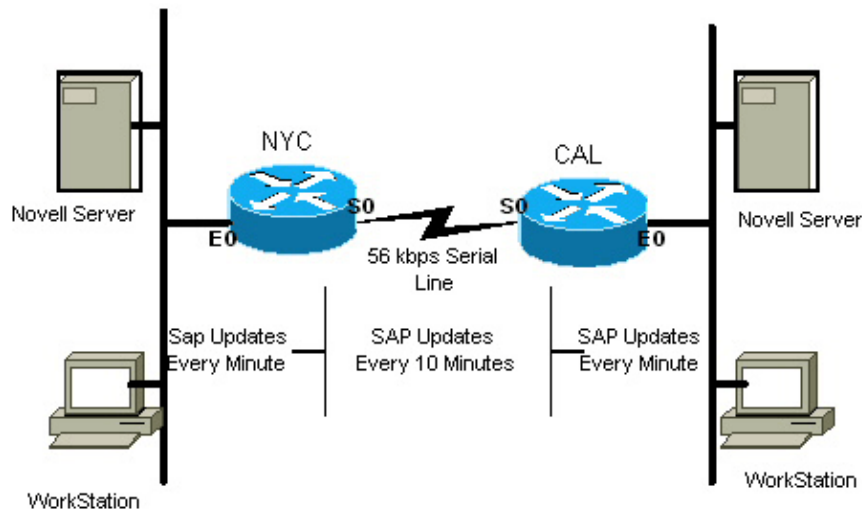
Seventy-two 480-byte frames is not a lot of data in some environments. Ethernet, for example, would be able to process all of these frames in less than .0037 of a second (adding in the 16 bit preamble and assuming no contention). That works out to a negligible 0.62% of bandwidth per minute.

But, this is a lot of data for a 56Kb serial line. 56Kb is approximately 1/120th of an Ethernet's bandwidth. This same scenario running across 56Kb lines would take 4.1 seconds to complete – or about 7% of the serial line's bandwidth every one minute. This is significant, because for 4.1 seconds the serial line is doing nothing but forwarding SAP update frames. Other applications may time-out, and other protocols will have to resort to "higher-layer" recovery schemes, generating even more traffic across the already full serial connection.

SAP Interval

Cisco has provided some solutions for loading slower serial lines. Through the use of the **ipx sap-interval** command, the user can set the SAP update period on serial interfaces to values greater than the one minute

default. Although Novell workstations and file servers on Ethernet and/or Token Ring networks require SAP updates every one minute, serial interfaces have only the two "nodes" on each end of the link. The **ipx sap-interval** command can be applied to both of these "nodes" to reduce SAP traffic while maintaining accurate service information to the networks that support servers and workstations. This command will be especially useful to you in increasing the efficiency of slower serial connections. Below is an example for how it works:



Configurations

NYC
<pre> NYC# interface serial 0 novell sap-interval 10 </pre>

CAL
<pre> CAL# interface serial 0 novell sap-interval 10 </pre>

The **novell sap-interval** command, used with a value of 10, reduces SAP traffic on the serial link by 90%. Note that the status of services on the other side of the network may not be received for a period of $(\text{sap-interval} + 1)$ – in this case 11 minutes. Although the value of the interval can be set to any number of minutes, keep it to a low value so that both sides of the internetwork are synchronized in a reasonable amount of time. If servers go away gracefully using the **down** command, the value of **ipx sap-interval** can be much higher, like 60 minutes.

Note: Never use the **novell sap-interval** command on Ethernet, Token Ring, or FDDI interfaces.

NetWare SAP Restrictor

The NetWare Service Advertising Restrictor enables you to modify how a file server deals with SAP information. It requires a NetWare v3.11 file server. Basically, it lets you determine what servers will be visible at any given point in an internetwork.

There are three main reasons you would want to do this.

- The first is to reduce the amount of network traffic generated by SAP. On a large (hundreds of servers) internetwork, the amount of traffic generated by SAP can become noticeable, especially on slower links.
- Second, (also in large environments) there tends to be quite a bit of SAP information coming to a filer server from remote portions of the network that its users never (or rarely) access. Restricting the servers seen can simplify matters greatly by eliminating the inappropriate ones.
- The third reason is one of security. While users may not want to see some servers, you have some servers you don't want to be seen except in specific cases.

The restrictor accomplishes its task by allowing you two points of control over SAP information. Any particular server can be restricted either as it enters in to the router, or as it is going out. Cisco has the capability to restrict SAP information using SAP filters.

SAP Filters

Cisco routers let you control SAP messages in several ways. Access lists and SAP filters combine to allow you to control how SAP messages from network segments or specific servers are routed among Novell networks.

To define an access list for filtering SAP requests, use this variation of the **access-list** command:

```
access-list number {permit|deny} network.[address] [service-type]
```

The argument **number** is the SAP access list, which is a decimal number in the range 1000 to 1099.

Enter the keyword **permit** or **deny** to establish the type of access desired.

Permit or deny access is based on the data provided.

The argument **network** is a hexadecimal Novell network number. 0 defines the local network, -1 defines all networks.

The optional **address** argument is a Novell node address.

The **service-type** argument defines the service type to filter. Service types are entered in hexadecimal. 0 is all services. Examples of other service types that can be entered are listed in the following table:

DESCRIPTION	SERVICE TYPE
-----	-----
User Group	2
Print Queue	3
File Server	4
BRetrieve VAP 4.11	50
Gateway	6
Print Server	7
Archive Queue	8
Archive Server	9
Job Queue	A
Novell SAA Gateway	304
Remote Bridge Server	24

Sample SAP Filters Configurations

If the following access-list (1001) was applied to an interface, it would block all access to SAP type 4 (File Servers) on the network. The second statement allows access to all of the other services (SAP types) on the network.

```
access-list 1001 deny -1 4
access-list 1001 permit -1
```

One of the quickest ways to reduce the amount of SAP traffic on a serial link is to filter out SAP type 7 – the Print Server Advertisement. Novell workstations rarely need to spool to a Print Server at a remote location. By taking a close look at the types of applications that run on local segments, it should be easy to eliminate a significant number of SAP frames that really don't need to cross the serial link. For example, does every remote location need to access the payroll Database Server (SAP type 4B)? Does everyone in your organization need access to the IBM host systems through the Novell SAA Gateway (SAP type 304)?

There are four SAP filters that take IPX access-list numbers as input. We won't go into detail about the application of these filters, but for your reference, they are:

```
ipx input-sap-filter
ipx output-sap-filter
ipx router-sap-filter
ipx output-gns-filter
```

These commands take a SAP IPX access list number as their input. The range for SAP lists is 1000 to 1099, but **ipx router-sap-filter** can take an 800 or 900 list. Follow these guidelines for SAP filtering:

- When the **ipx input-sap-filter** list is enabled, use the list to determine the services that will be accepted to be stored in the router's table and rebroadcasted.
- When the **ipx output-sap-filter** list is enabled, use the list to determine the services that will be included in SAP updates from Cisco routers.
- When the **ipx router-sap-filter** list is enabled, use the list to determine which source routers and SAP service types will be accepted.
- Use the **ipx output-gns-filter** to decide which servers should be in replies to GNS requests.

SAP Interpacket Gap

On slower speed serial links, and sometimes even on LAN interfaces, you should configure an **ipx output-sap-delay** command, which specifies the amount of time to wait before sending another SAP packet. This will avoid filling the output queue with nothing but SAP packets, especially on slower speed links. If a SAP delay is approximately the time it takes to transmit one 480 byte frame across the link, then there will be at most two interface buffers used by the SAP. Without a delay, and depending on the service table size, the router will queue the SAP packet to the output interface buffers as fast as the system can process through the service table. Once the output buffers are full, the SAP packet will be dropped.

If a router on one side of a WAN link has the entire service table but the remote side is missing some services, especially services near the end of the unsorted service table, try setting an **ipx output-sap-delay** command on both sides of the WAN link.

Use **ipx output-sap-delay** in Cisco IOS ® versions earlier than 9.21 with caution, since there may be problems with the interface input queue. If you must use **ipx output-sap-delay** in a Cisco IOS 9.1-based environment, use as small a delay as possible. Monitor the system input queue for signs of drops, ignores, or no input buffers increasing.

In the future, a 55ms output sap delay may become standard for all Cisco interfaces. A 55 ms gap is called for by Novell's IPX Router Specification. The current default is 0 ms.

I/O Queues

When a large number of SAPs are being sent from an interface on the router, we may need to adjust the input and output queue sizes to accommodate the extra traffic. SAP updates are read into the router and then sent

out at the proper interval (usually one minute) by the router. If the number of SAP frames to be sent exceeds the number of available output buffers on the interface, the interface will drop frames. Let's look at an Ethernet interface (**show interface ethernet 0**) set with default queue sizes:

```
Ethernet0 is up, line protocol is up
  Hardware is MCI Ethernet, address is 0000.0c01.b43e (bia 0000.0c01.b43e)
  Internet address is 7.0.0.3, subnet mask is 255.0.0.0
  MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec, rely 255/255, load 1/255
  Encapsulation ARPA, loopback not set, keepalive set (10 sec)
  ARP type: ARPA, ARP Timeout 4:00:00
  Last input 0:00:01, output 0:00:03, output hang never
  Last clearing of "show interface" counters never
  * Output queue 0/40, 0 drops; input queue 0/75, 0 drops
  Five minute input rate 1000 bits/sec, 0 packets/sec
  Five minute output rate 0 bits/sec, 0 packets/sec
    3154 packets input, 636983 bytes, 0 no buffer
    Received 2898 broadcasts, 0 runts, 0 giants
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
    0 input packets with dribble condition detected
    487 packets output, 53570 bytes, 0 underruns
    0 output errors, 0 collisions, 3 interface resets, 0 restarts
```

Note the line, **Output queue 0/40, 0 drops; input queue 0/75, 0 drops**. If we use the values from our previous example, 72 SAP frames per update, we see that the interface will not be able to accommodate the number of SAP frames that need to be sent.

While setting the size of these queues is not an exact science, there is a simple calculation that will work in most scenarios. Keep the values as low as possible, and remember that the input queue should always be larger than the output queue. Here's the formula:

Number of Services / 7 * 1.3 = size of Output Queue

Number of services / 7 * 1.6 = size of Input Queue

Applying this to our earlier example, we get the values:

500 SAPs / 7 (services per frame) = 71.43 (or 72)

72 * 1.3 = 93.6 or let's say 97 for the Output Queue

72 * 1.6 = 115.2 or let's say 115 for the Input Queue

Use the following configuration commands to implement these changes:

```
int e 0
hold-queue 97 out
hold-queue 115 in
```

Related Information

- **Technical Support & Documentation – Cisco Systems**

[Contacts & Feedback](#) | [Help](#) | [Site Map](#)

© 2009 – 2010 Cisco Systems, Inc. All rights reserved. [Terms & Conditions](#) | [Privacy Statement](#) | [Cookie Policy](#) | [Trademarks of Cisco Systems, Inc.](#)
