

# Understanding and Troubleshooting Local Source–Route Bridging

Document ID: 5202

---

## **Introduction**

### **Before You Begin**

Conventions

Prerequisites

Components Used

### **Routing Control Field**

### **Routing Designator Field**

### **Cisco Router Basic Configuration**

### **Spanning Explorers**

### **Source Bridging of Routed Protocols**

### **show Commands**

Source–Route Bridge Portion of show source Command Output

Explorer Traffic Portion of show source Command Output

More show Commands

### **Troubleshooting**

Hints

### **Debugging**

### **Related Information**

---

## **Introduction**

Source–Route Bridging (SRB) is the concept by which a station in a Token Ring environment can establish a route through a multiple ring network to its destination. This document discusses the components of SRB, and provides basic configuration and troubleshooting information.

## **Before You Begin**

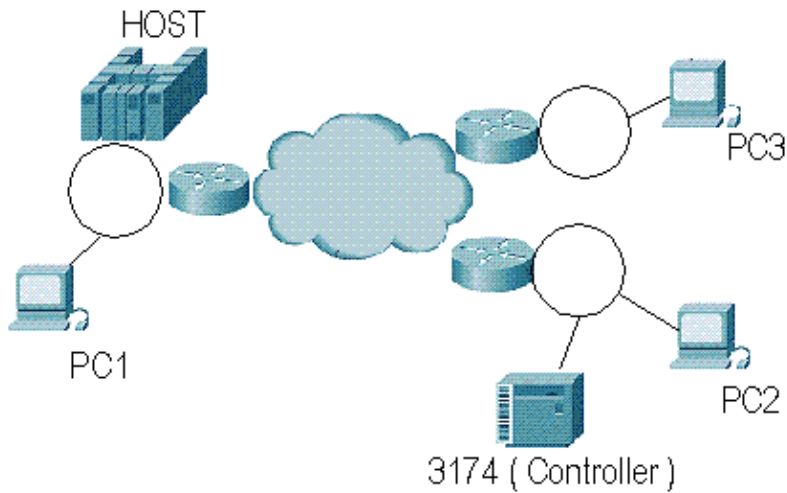
### **Conventions**

For more information on document conventions, see the Cisco Technical Tips Conventions.

### **Prerequisites**

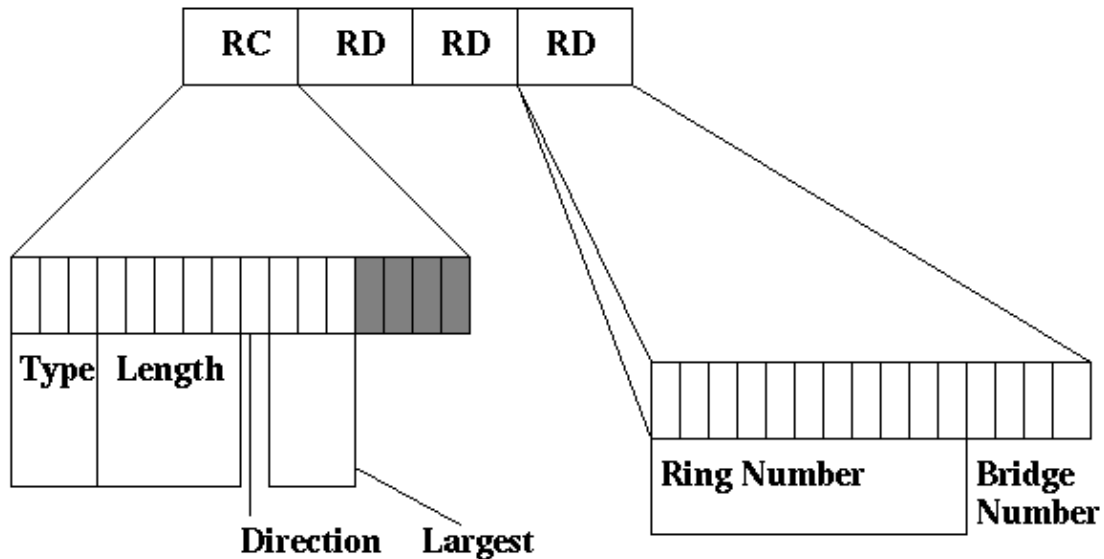
This document assumes that the reader is knowledgeable of basic concepts of source–route bridging as explained below:

The first step for a station to reach another is to create a packet called an explorer. This packet is copied by all bridges in the network. They each add information as to where the packet has traversed. As this is constructed through the network, the end station will start receiving these packets. The end station then decides which route to use to return the originator, or it will send another explorer back so that the originating station may determine the route.



In SRB, the Routing Information Field (RIF) is the part of the explorer that contains the information of where the explorer has traversed. Within the RIF, the route descriptor is where information is stored about the path to the network. The route control contains information about the RIF itself. The following diagram shows the RIF broken into these sections:

### *Routing Information Field*



### Components Used

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

The information presented in this document was created from devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If you are working in a live network, ensure that you understand the potential impact of any command before using it.

# Routing Control Field

The Routing Control (RC) field starts at byte 14 of the MAC Token Ring frame. This is the first part of the RIF field on the Token Ring frame.

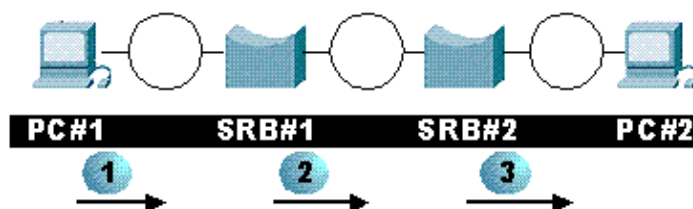
- The type field is 3 bits long. This table below lists the broadcast indicators.

Value	Broadcast type
B`0XX`	Directed frame
B`10X`	All route explorer
B`11X`	Single route explorer

- ◆ A **directed frame** indicates that the frame contains the defined path across the network and, by definition, no change is needed on the RIF.
- ◆ **All route explorers** go through the whole network. All SRB must copy the frame to every port except the one that has a destination ring that is already in the RIF.
- ◆ **Single route explorers** are explorers that pass through a predetermined path constructed by a Spanning-Tree Algorithm (STA) in the bridges. A station should receive only one single route explorer from the network.
- ◆ The explorer has a very important limit on how many rings it can hold in the routing information field. By definition of a Token Ring, the RIF can hold a total of 14 rings. IBM limited this to seven for the RIFs on the bridges in the network however; Cisco has also adopted this limitation. So, an explorer that has traversed 7 rings will be dropped by a Cisco router. There are parameters that can be set in the Cisco router to decrease this so that packets that reached x number of rings will get dropped. This is an effective way of controlling traffic in the network.

Also, the router checks only the RIF length on an explorer packet, but does not pay any attention if the frame is directed. If the sending station generates a packet with a static RIF, the router checks the RIF for forwarding purposes only and could have a 14 limit hop count.

- ◆ The third bit in this field is reserved (it is not currently used and is ignored by end stations).
- The **Length** field is 5 bits long and contains the length of the RIF in bytes.
- The **Direction** bit determines how the RIF should be read by SRB in the network to follow the path to reach the end station.
  - ◆ If the bit is set to **B'0'**, the RIF should be read from left to right.
  - ◆ If it is set to **B'1'**, the RIF should be read from right to left.
- The largest frame bits (3 bits) determine the largest frame that can traverse the network, as illustrated in the figure below.



- ◆ The following happens to the largest frame field:

◇ PC#1 constructs the RIF on this frame and in the largest frame bits puts B`111`. This interprets in sniffers as 49K.

- ◇ SRB#1 has a MTU of 4K on both interfaces. The source–route bridge adds information to the RIF regarding the ring numbers and modifies the length field and the largest frame. In this case, the value is changed to B`011`.
- ◇ SRB#2 has a MTU of 2K for both interfaces. The source–route bridge changes the largest frame to B`010`.

◆ The chart below lists the possible values.

Value	Largest frame size
000	516 bytes
001	1500 bytes
010	2052 bytes
011	4472 bytes
100	8191 bytes
101	114076 bytes
110	17800 bytes
111	Used in all route explorers

## Routing Designator Field

The Route Designator (RD) field contains information on the route that the packet should take to reach the destination station. Each ring in a Token Ring network must be unique, or the packet can finish in the wrong place. This is especially important in a RSRB environment because the router caches information about the remote ring. Each entry in the route designator field contains the ring number and the bridge number. The ring portion is 12 bits long and the bridge portion is 4 bits long. This makes it possible for the ring to have a value from 1 to 4095 and the bridge a value from 1 to 16. Cisco routers store these values in decimal value, but the RIF shows the values in hexadecimal.

RCF	RING	Bridge	RING	Bridge	RING	Bridge
C820	001	1	002	1	003	0
1100100000100000	00000000000001	0001	0000000000010	0001	0000000000011	0000

The table above contains the RIF in hexadecimal as it is displayed in the **show rif** command output. It then shows the same in binary to decode it. The decoded version is shown in the table below.

Bit position	Value	Description
1–3	110	Single route explorer
4–8	01000	RD length of 8 bytes
9	0	Read RIF in forward direction
10–12	010	Largest frame 2052
13–16	0000	Reserved

## Cisco Router Basic Configuration

This section discusses how to configure a Cisco router for SRB. One important detail of this configuration is the concept of the virtual ring. The virtual ring is an imaginary ring that is constructed logically inside the

router. It ties in all the interfaces of the router, which is important because an interface can only point to one destination ring, not multiple rings. An example configuration of an interface is shown below.

```
source-bridge ring-group 200
...
Interface tokenring 0/0
 ip address 10.1.1.1 255.255.255.0
 ring-speed 16
 source-bridge 100 1 200
```

The configuration above sets up a virtual ring group of 200 with the command **source-bridge ring-group 200**. The configuration of the interface points correctly from ring 100 to ring 200, which is the virtual interface.

You could also have a configuration in which you point to interfaces together without a virtual ring group. An example of this is shown below.

```
Interface tokenring 0/0
 ip address 10.1.1.1 255.255.255.0
 ring-speed 4
 source-bridge 100 1 300
Interface tokenring 0/1
 ip address 10.1.2.1 255.255.255.0
 ring-speed 16
 source-bridge 300 1 100
```

The configuration above connects the previous two interfaces for SRB. Now, these two interfaces can exchange SRB frames, but they cannot communicate with any other source-route bridge interface on this router.

The virtual ring plays a necessary role in Remote Source-Route Bridging (RSRB) and Data-Link Switching (DLSw) because it's necessary to configure for these features.

## Spanning Explorers

The **source-bridge spanning** command plays an important role. When we discussed earlier the different type of explorers, we mentioned all route explorers and single route explorers. The **source-bridge spanning** command permits us to forward single route explorer frames. Without this the router will simply drop the frame at the interface. No drop counters will ever increment to indicate this. So in network with NetBIOS stations you must make sure that you have enabled spanning. Also if you have configured DLSw you need to configure the **source-bridge spanning** command since DLSw is going to use single route explorer frames to locate stations. In the following configuration, the router is configured to forward single route explorer frames:

```
source-bridge ring-group 200

Interface tokenring 0/0
 ip address 10.1.1.1 255.255.255.0
 ring-speed 4
 source-bridge 100 1 200
 source-bridge spanning
Interface tokenring 0/1
 ip address 10.1.2.1 255.255.255.0
 ring-speed 16
 source-bridge 300 1 200
 source-bridge spanning
```

An expanded version of this configuration is shown below.

```

source-bridge ring-group 200
Interface tokenring 0/0
 ip address 10.1.1.1 255.255.255.0
 ring-speed 4
 source-bridge 100 1 200
 source-bridge spanning 1
Interface tokenring 0/1
 ip address 10.1.2.1 255.255.255.0
 ring-speed 16
 source-bridge 300 1 200
 source-bridge spanning 1
 bridge 1 protocol ibm

```

The IBM Spanning-Tree Protocol (STP) is used to create a spanning tree so that **single route explorer frames** are forwarded through a single path by blocking ports on the bridged environment. This is similar to regular IEEE spanning tree only that it's used for single route explorers only. If you have this configuration, you probably need to also monitor the **show spann** command output on the router to determine the state of the ports, since they could go in blocking state depending on the topology. This router is now configured to participate in the IBM spanning tree protocol.

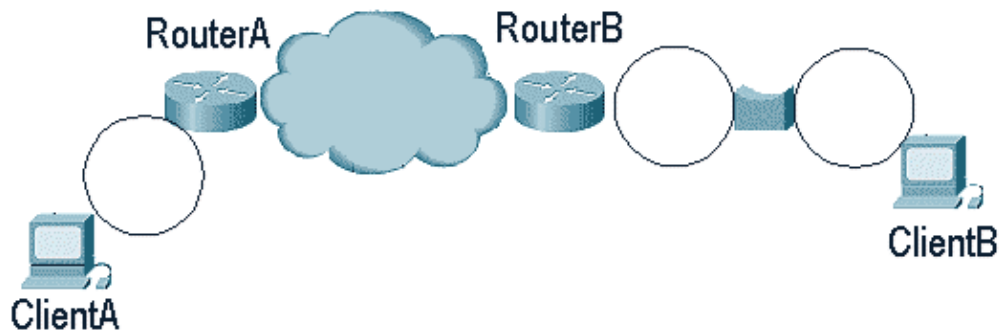
```

source-bridge ring-group 200
Interface tokenring 0/0
 ip address 10.1.1.1 255.255.255.0
 ring-speed 4
 source-bridge 100 1 200
 source-bridge spanning 1
Interface tokenring 0/1
 ip address 10.1.2.1 255.255.255.0
 ring-speed 16
 source-bridge 300 1 200
 source-bridge spanning 1
 bridge 1 protocol ibm

```

## Source Bridging of Routed Protocols

An important part of SRB in routers is the ability to pass a routed protocol across a source-route bridged network. The router always removes the LLC information from the routed frame and reconstructs the LLC layer for the destination media. This is illustrated in the diagram below:



If client A wants to reach client B, routerA must destroy all LLC information and below from the frame, create the LLC frame for the WAN, and dispatch the frame to routerB. RouterB now receives the frame, destroys the WAN LLC information from the frame, and has an IP frame ready to reach client B.

The router needs source-routed information to reach clientB because it is a ring away across a SRB. RouterB then acts as a source-route bridge network end station where it must find the path to reach client B. RouterB must send an explorer to determine the location of clientB. When clientB responds to routerB, it stores the routing information field (RIF) and uses it to send more packets to clientB.

This is what happens behind the scenes in routerB when multiring is configured on the interface. It isn't required if clientB is on the same ring as routerB because the router would send a broadcast locally and get a response back from clientB. The configuration for this is shown below:

```
Interface tokenring 0/1
ip address 10.1.2.1 255.255.255.0
ring-speed 16
multiring ip
```

Multiring can be configured for specific multiple protocols, or with **multiring all**, which specifies all routed protocols. This goes into effect only for protocols that are actually routed by the router. If the protocol is bridged, **multiring all** this does not apply.

The **show rif** command is important when multiring is configured. Because the router has to cache the RIF for future packets destined for clientB, it needs to store the RIF to avoid having to send an explorer for every packet that needs to reach clientB.

```
s4a#sh rif
Codes: * interface, - static, + remote

Dst HW Addr   Src HW Addr   How      Idle (min)  Routing Information Field
0000.30b0.3b69 N/A           To3/2    *           C820.0A01.0B02.0C00
s4a#
```

For IP networks in which you need to source-route IP packets, use the **show arp** command to display the MAC address for the station that you are trying to reach. Once you have the MAC address, you can use **theshow rif** command to determine the path that the router is using to reach that station in the source-routed network.

```
s4a#sh arp
Protocol  Address           Age (min)   Hardware Addr  Type   Interface
Internet  10.17.1.39        -           4000.0000.0039 SNAP   TokenRing3/0
Internet  171.68.120.39    -           4000.0000.0039 SNAP   TokenRing3/0
s4a#
```

## show Commands

The **show** commands are helpful when troubleshooting source-route bridge problems. Output from the **show interface** command is shown below.

```
TokenRing3/2 is up, line protocol is up
Hardware is cxBus Token Ring, address is 0000.30b0.3b69 (bia 0000.30b0.3b69)
MTU 4464 bytes, BW 16000 Kbit, DLY 630 usec, rely 255/255, load 1/255
Encapsulation SNAP, loopback not set, keepalive set (10 sec)
ARP type: SNAP, ARP Timeout 4:00:00
Ring speed: 16 Mbps
Single ring node, Source Route Transparent Bridge capable
Source bridging enabled, srn 25 bn 4 trn 31 (ring group)
proxy explorers disabled, spanning explorer disabled, NetBIOS cache disabled
Group Address: 0x00000000, Functional Address: 0x0800011A
Ethernet Transit OUI: 0x0000F8
Last Ring Status 0:21:03 <Soft Error> (0x2000)
Last input 0:00:02, output 0:00:02, output hang never
Last clearing of "show interface" counters never
Output queue 0/40, 0 drops; input queue 0/75, 0 drops
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
  41361 packets input, 2149212 bytes, 0 no buffer
  Received 3423 broadcasts, 0 runts, 0 giants
  3 input errors, 0 CRC, 0 frame, 0 overrun, 0 abort
```

```

40216 packets output, 2164005 bytes, 0 underruns
8 output errors, 0 collisions, 4 interface resets, 0 restarts
0 output buffer failures, 0 output buffers swapped out
4 transitions
s4a#

```

In the **show interface** command output, pay special attention to the following parts:

- The `ring speed` tells you the speed at which this ring is currently operational.
- When SRB is enabled, you can also check the information that is configured for ring and bridge numbers. For example SRN is the source ring number, BN is the bridge number, and TRN is the target ring number, that the virtual ring has selected for that router.
- Last `ring status` supplies the last ring status for the ring. For example, 0x2000 indicates a software error. A list of possible status values is shown below.

```

#define RNG_SIGNAL_LOSS  FIXSWAP(0x8000)
#define RNG_HARD_ERROR   FIXSWAP(0x4000)
#define RNG_SOFT_ERROR   FIXSWAP(0x2000)
#define RNG_BEACON       FIXSWAP(0x1000)
#define RNG_WIRE_FAULT   FIXSWAP(0x0800)
#define RNG_HW_REMOVAL   FIXSWAP(0x0400)
#define RNG_RMT_REMOVAL  FIXSWAP(0x0100)
#define RNG_CNT_OVRFLW   FIXSWAP(0x0080)
#define RNG_SINGLE       FIXSWAP(0x0040)
#define RNG_RECOVERY     FIXSWAP(0x0020)
#define RNG_UNDEFINED    FIXSWAP(0x021F)

#define RNG_FATAL        FIXSWAP(0x0d00)
#define RNG_AUTOFIX      FIXSWAP(0x0c00)
#define RNG_UNUSEABLE    FIXSWAP(0xdd00) /* may still be open */

```

- The `drops` counter helps to determine how many drops there have been in the outbound queue for process level traffic and for input memory buffers. This helps determine the quantity of throttles.
- The `output rate` and `input rate` give an overall idea of how busy the router is forwarding/receiving frames on the interface.
- `Runts` and `giants` are frames below and above the SPEC of Token Ring. You rarely encounter these in Token Ring, but they are very useful in Ethernet.
- `Input errors` are crucial. There should be none if the ring is healthy. If there are problems in the ring (such as lots of noise), CRCs will fail and the frames will be dropped. If the `ignore count` is incrementing, it means that the input buffers are filling up and the router is discarding packets destined for our interface.
- `Interface resets` can be either administrative (issue the **clear int tok x** command), or internal when an error occurs at the interface level.
- The `transitions` counter represents the number of times the interface went from up to down.

The **show source** command is the source of all the most important information for troubleshooting source–route bridging problems. Sample output from this command is shown below.

```

s4a#show source

Local Interfaces:
      srn bn  trn r p s n  max hops  receive      transmit      drops
      cnt:bytes cnt:bytes
Ch0/2  402 1  200 *  f   7 7 7      0:0          0:0           0
Ch0/2  111 1  200 *  f   7 7 7      0:0          0:0           0
Ch1/2   44 2   31 *  f   7 7 7  17787:798947  18138:661048  0
To3/0 1024 10 200 *  f   7 7 7      0:0          0:0           0
To3/1  222 1  200 *  b   7 7 7      0:0          0:0           0
To3/2   25 4   31 *  b   7 7 7  18722:638790  17787:692225  0

Global RSRB Parameters:
TCP Queue Length maximum: 100

```

```

Ring Group 401:
  No TCP peername set, TCP transport disabled
  Maximum output TCP queue length, per peer: 100
  Rings:

Ring Group 200:
  No TCP peername set, TCP transport disabled
  Maximum output TCP queue length, per peer: 100
  Rings:
  bn: 1  rn: 402  local  ma: 4000.30b0.3b29  Channel0/2          fwd: 0
  bn: 1  rn: 111  local  ma: 4000.30b0.3b29  Channel0/2          fwd: 0
  bn: 10 rn: 1024 local  ma: 4000.30b0.3b29  TokenRing3/0       fwd: 0
  bn: 1  rn: 222  local  ma: 4000.30b0.3ba9  TokenRing3/1       fwd: 0

Ring Group 31:
  No TCP peername set, TCP transport disabled
  Maximum output TCP queue length, per peer: 100
  Rings:
  bn: 4  rn: 25   local  ma: 4000.30b0.3b69  TokenRing3/2       fwd: 17787
  bn: 2  rn: 44   local  ma: 4000.30b0.3b29  Channell1/2       fwd: 17919

Explorers: ----- input -----          ----- output -----
                spanning  all-rings    total          spanning  all-rings    total
Ch0/2             0           0             0              0           0             0
Ch0/2             0           0             0              0           0             0
Ch1/2             0           0             0              0          219            219
To3/0             0           0             0              0           0             0
To3/1             0           0             0              0           0             0
To3/2             0          762            762            0           0             0

Local: fastswitched 762          flushed 0          max Bps 38400

                rings      inputs      bursts      throttles      output drops
Ch0/2             0           0           0             0             0
Ch0/2             0           0           0             0             0
Ch1/2             0           0           0             0             0
To3/0             0           0           0             0             0
To3/1             0           0           0             0             0
To3/2             0          762            0             0             0

```

The **show source** command is divided into several sections: the interface level SRB information, the RSRB portion, and the explorer portion. The explorer and SRB portions are explained below. The RSRB portion is covered in Configuring Remote Source–Route Bridging.

## Source–Route Bridge Portion of show source Command Output

The source–route bridge portion contains the following information:

```

Local Interfaces:
                srn bn  trn r p s n  max hops  receive      transmit      drops
                cnt:bytes  cnt:bytes
Ch0/2  402  1  200 *  f  7 7 7  0:0  0:0  0
Ch0/2  111  1  200 *  f  7 7 7  0:0  0:0  0
Ch1/2  44  2  31 *  f  7 7 7  17787:798947  18138:661048  0
To3/0  1024 10 200 *  f  7 7 7  0:0  0:0  0
To3/1  222  1  200 *  b  7 7 7  0:0  0:0  0
To3/2  25  4  31 *  b  7 7 7  18722:638790  17787:692225  0

```

- For each interface, you should see SRN, BN, and TRN. This tells you where the source–routed information has been forwarded from the interface.
- r: Ring group has been assigned to this interface.
- p: Interface has proxy explorers are configured.
- s: Spanning tree explorers are configured.

- n: NetBIOS name caching is configured.
- The `receive` and `transmit` counts show the quantity/bytes of SRB traffic that has been handled by this interface.
- drops: The quantity of source-routed frames dropped by the interface of the router. The possible reasons for these drops are listed below.
  - ◆ An SRB packet was received when there is no path (poorly configured **source-bridge** statement.)
  - ◆ The received RIF is too long.
  - ◆ A filter drops the frame.
  - ◆ The ring group specified in a **source-bridge** statement for an interface was not found.
  - ◆ A RIF was received that is too short.
  - ◆ A destination ring immediately beyond the ring group is specified, but the router does not have it in the remote ring list from any remote peer.
  - ◆ A RIF says to output a frame on the same interface from which it was input.
  - ◆ A poorly formed explorer was received (No RII, for example).
  - ◆ An explorer was sent with the D bit set or with an odd byte length RIF field.
  - ◆ A spanning explorer was received on an interface for which spanning is not specified.
  - ◆ An explorer frame tried to go out to a ring that it had entered.
  - ◆ The maximum RIF length would be exceeded if the router attempted to forward the frame.
  - ◆ A multicast frame not destined to the router does not have a RIF, so the router cannot forward it.

## Explorer Traffic Portion of show source Command Output

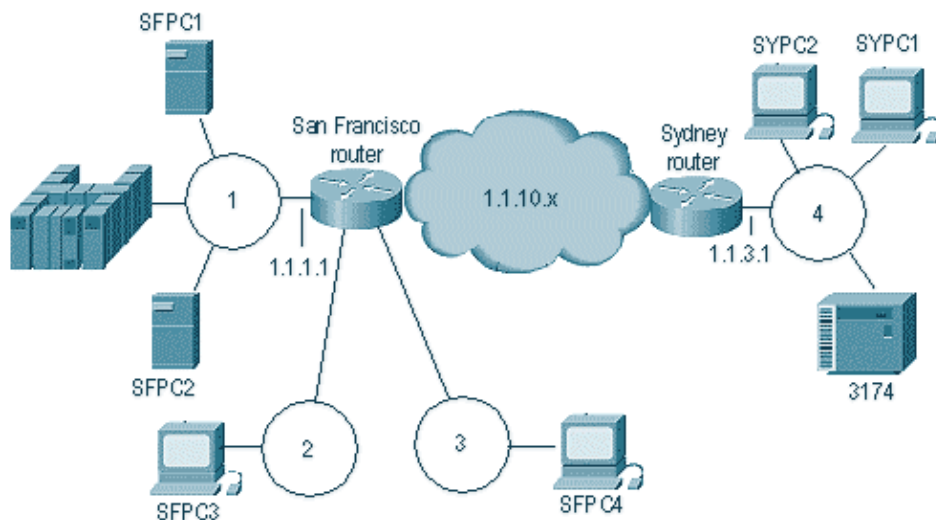
Cisco IOS separates the explorer traffic from regular source route traffic. This provides us with a beneficial troubleshooting tool. One of the worst problems with any broadcast medium is the large number of broadcasts. In an Ethernet environment, too many broadcasts can account for too many computers under the same Ethernet. In a Token Ring network, broadcasts are better known to as explorers, because they traverse from ring to ring exploring for a station on the ring. These explorers are limited to traversing seven rings only. In a meshed ring environment, however, one explorer can finish being copied by many bridges, which can cause too many explorers.

Because you can differentiate between explorers and real data, you can manipulate them to our advantage. The commands listed in the table below are used in the router for explorer manipulation.

Task	Command
Set the maximum explorer queue depth.	<b>source-bridge</b> <b>explorerq-depth</b> <i>depth</i>
Prevent explorer storms in redundant network topologies by filtering explorers that have already been forwarded once.	<b>source-bridge</b> <b>explorer-dup-ARE-filter</b>
Set the maximum byte rate of explorers per ring.	<b>source-bridge</b> <b>explorer-maxrate</b> <i>maxrate</i>
Turn off fast-switching of explorers.	<b>no source-bridge</b> <b>explorer-fastswitch</b>

In the diagram below, there are two different types of connections: those going from ring to ring in the router, and those going across the WAN. As of Cisco IOS 10.3, you can fast-switch explorers, which is about five times faster than process-switching them. You can use the **explorer-maxrate** or **explorer-qdepth** command

to do this.



In the diagram above, station SFPC4 sends an explorer to reach SFPC1. The router will fast-switch the explorer to rings 1 and 2. But the router will also send the explorer to the explorer queue for RSRB processing to send the frame to the remote site (this is assuming that **netbios enable name cache** and **proxy explorer** commands are turned off).

If this was an enormous NetBIOS shop, for example, the quantity of explorer traffic would be very high. To control this, you can use the **explorer-maxrate** and **explorer-qdepth** parameters. These both behave at different levels of operation. Explorer maxrate operates at the interface level with the fast-switch code and **explorer-qdepth** operates at process level. When used in combination, these parameters provide the best control of explorers. The default value for explorer-maxrate is 38400 for smaller boxes and 64000 for high-end boxes. The **explorer-qdepth** defaults to 30 for all platforms.

Below is the explorer portion of the **show source** command output.

```

Explorers:  ----- input -----
              spanning  all-rings      total          ----- output -----
              spanning  all-rings      total          spanning  all-rings      total
Ch0/2          0          0          0              0          0          0
Ch0/2          0          0          0              0          0          0
Ch1/2          0          0          0              0          219         219
To3/0          0          0          0              0          0          0
To3/1          0          0          0              0          0          0
To3/2          0          762         762            0          0          0

Local: fastswitched 762      flushed 0          max Bps 38400

rings         inputs      bursts          throttles      output drops
Ch0/2          0          0              0              0              0
Ch0/2          0          0              0              0              0
Ch1/2          0          0              0              0              0
To3/0          0          0              0              0              0
To3/1          0          0              0              0              0
To3/2          762         0              0              0              0

```

To determine the rate of explorers, refer to the parameters listed below.

- **fastswitched** shows the number of explorers that were fast-switched.
- **flushed** displays how many explorers were thrown away by the router because the maxrate value was exceeded at the interface level.
- **max Bps** indicates the quantity of explorer bytes per second that the router is accepting inbound per

interface.

- **bursts** shows the number of times that the router reached the maximum quantity of explorers in the explorer queue.
- **throttles** shows the number of times that the router cleaned the input buffers of an interface because the router wasn't able to service those buffers quickly enough. This causes all outstanding packets waiting in the input buffers to be dropped.
- **output drops** is the number of explorers that were dropped outbound on this interface.

For example, look at the San Francisco router in the previous diagram. It is currently configured to run at 38,400 Bps, and has a total of three local interfaces. Each can run at 38,400 Bps. This is checked every 10th of a second, so that means that for every 10th of a second the router can absorb 3,840 Bps of explorer traffic. If you divide 3,840 by 64 (which is the average NetBIOS explorer packet), it equals about 60 explorers per 10th of a second (600 explorers per second).

This is important because it can tell you how many explorers the router could hit outbound an interface. If the traffic was headed for ring 1 from both ring 2 and 3, there could be a forwarding rate outbound on ring 1 of 1200 explorers per second. This could easily create a problem in the network.

The **explorer-queue** is a different mechanism and is five times slower than maxrate. All explorers in the **explorer-queue** are process-switched by definition. This is usually what leads up to RSRB, but varies depending on the set-up, because you could easily tell the router to run all traffic in process-switch mode by turning off **explorer-fastswitch** (For more information on RSRB, please see Configuring Remote Source-Route Bridging ). The main measure for **explorer-queue** processing is the burst value in the **show source** output. This is the number of times that the router reached the maximum **explorer-queue** depth. If the queue is always maxed-out, the router will increment burst only once: the first time that the maximum is reached.

## More show Commands

The **show source interface** command provides a shorter version of the output from the **show source**. This is helpful if you have a large router and want a brief look at how it is configured. You can also use it to determine the MAC addresses of the interface of the router. Sample output from this command is shown below:

```
s4a#show source interface
```

Line	Pr	Status	MAC Address	v p s n r								Packets	
				srn	bn	trn	r	x	p	b	c	IP Address	In
Ch0/0	down	dn										0	0
Ch0/1	admin	dn								10.1.1.2		0	0
Ch0/2	up	up										0	0
Ch1/0	admin	dn										0	0
Ch1/1	up	up								10.17.32.1		31201	45481
Ch1/2	up	up								10.18.1.39		17787	18137
To3/0	admin	dn	4000.0000.00391024	10	200	*	f	F		10.17.1.39		0	0
To3/1	admin	dn	0000.30b0.3ba9	222	1	200	*	b	F			0	0
To3/2	up	up	0000.30b0.3b69	25	4	31	*	b	F			41598	40421
To3/3	admin	dn	0000.30b0.3be9									0	0
Lo0	up	up								11.100.100.1		0	28899

Another useful command is **show ip interface brief**. It summarizes the IP address per port and lets you know if the interface is up/up. Several other helpful **show** commands are listed in the table below.

Task	Command
	<b>show interfaces</b>

Provide high-level statistics about the state of source bridging for a particular interface.	
Show the current state of any current local acknowledgment for both LLC2 and SDLLC connections.	<del>show local-ack</del>
Display the contents of the NetBIOS cache.	<del>show netbios-cache</del>
Display the contents of the RIF cache.	<del>show rif</del>
Display the current source bridge configuration and miscellaneous statistics.	<del>show</del>
Display the spanning-tree topology for the router.	<del>source-bridge</del> <del>show span</del>
Display a summary of Silicon Switch Processor (SSP) statistics.	<del>show sse</del> <del>summary</del>

## Troubleshooting

When troubleshooting any networking problem, start from the bottom layer up. Do not immediately think that there is a bug in the code. First, start by issuing the **show interface** command on the routers in question. You will see the following output:

```
TokenRing3/2 is up, line protocol is up
Hardware is cxBus Token Ring, address is 0000.30b0.3b69 (bia 0000.30b0.3b69)
MTU 4464 bytes, BW 16000 Kbit, DLY 630 usec, rely 255/255, load 1/255
Encapsulation SNAP, loopback not set, keepalive set (10 sec)
ARP type: SNAP, ARP Timeout 4:00:00
Ring speed: 16 Mbps
Single ring node, Source Route Transparent Bridge capable
Source bridging enabled, srn 25 bn 4 trn 31 (ring group)
proxy explorers disabled, spanning explorer disabled, NetBIOS cache disabled
Group Address: 0x00000000, Functional Address: 0x0800011A
Ethernet Transit OUI: 0x0000F8
Last Ring Status 0:21:03 <Soft Error> (0x2000)
Last input 0:00:02, output 0:00:02, output hang never
Last clearing of "show interface" counters never
Output queue 0/40, 0 drops; input queue 0/75, 0 drops
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
  41361 packets input, 2149212 bytes, 0 no buffer
  Received 3423 broadcasts, 0 runts, 0 giants
3 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  40216 packets output, 2164005 bytes, 0 underruns
8 output errors, 0 collisions, 4 interface resets, 0 restarts
  0 output buffer failures, 0 output buffers swapped out
4 transitions
s4a#
```

From this output, ask yourself these questions:

- Is the interface UP/UP?
- How many packets/sec are entering or leaving the interface?
- Are there any input errors (such as CRCs, frame, overruns, and so on)?

Of course, if you are seeing 4000 input errors out of 4 billion input packets, that wouldn't be considered a problem. But, 4000 out of 8000 transmitted is very bad.

If you see an interface that is transmitting and receiving packets, the next command to issue is **show interface token x accounting**. This command gives you an idea of what type of packets are going through a interface. All routed traffic will show independent of the bridge traffic. If there is only SRB on the interface, that is all you will see. Sample output from this command is shown below.

```
s4a#sh int tok 3/2 acc
TokenRing3/2
          Protocol    Pkts In   Chars In   Pkts Out   Chars Out
          SR Bridge   10674     448030     5583       187995
          LAN Manager  119       4264       4          144
          CDP         6871     2039316   5326       1549866
s4a#
```

In this output, you can see an interface that is performing only SRB, Cisco Discovery Protocol (CDP), and LAN network manager. Use this information to determine if the router is receiving source-routed packets on the interface.

Once that you have ruled out that the interface is forwarding and receiving source-routed frames, look at the router's configuration to verify the source-route bridge configuration, as shown below.

```
!
interface TokenRing3/2
 ip address 10.17.30.1 255.255.255.0
 ring-speed 16
 source-bridge 25 4 31
 source-bridge spanning
!
```

From this configuration, you can determine that the router is configured to source-route from ring 25 through bridge 4 to ring 31. Verifying the configuration of the router shows us that ring 31 is a configured virtual ring. It is also configured for **source-bridge spanning**, which means that the router will forward single route explorer frames. Some configuration questions that you need to consider are listed below.

- Who else is pointing to ring 31?
- Does the other interface that is pointing to virtual ring 31 show packets inbound and outbound (source-routed)?
- If the interface is pointing to a virtual ring that has source-bridge remote-peers, refer to Configuring Remote Source-Route Bridging to diagnose from there.

The above steps will generally rule out configuration problems or no packets being received from a station. If you are using any type of filtering, NetBIOS name caching, or proxy explorers and cannot get connected through the router, start with the basics. Always try to move the interface to its most simple configuration. Either remove the entries or double-verify them. An incorrectly constructed access list on the interface could also be a cause of problems. An example is shown below:

```
!
interface TokenRing3/2
 ip address 10.17.30.1 255.255.255.0
 no keepalive
 ring-speed 16
 source-bridge 25 4 31
 source-bridge spanning
 source-bridge input-address-list 700
!
access-list 700 deny 4000.3745.0001 8000.0000.0000
access-list 700 permit 0000.0000.0000 ffff.ffff.ffff
```

This will make the router drop all packets whose source address is 4000.3745.0001. To verify access lists in the entire box, use the **show access-list** command. This command output tells you all the access lists in the

router.

Another cause of problems could be proxy explorers. If you have proxy explorers configured, look at the **show rif** command output, as shown below.

```
s4a#show rif
Codes: * interface, - static, + remote

Dst HW Addr      Src HW Addr      How      Idle (min)  Routing Information Field
0000.30b0.3b69  N/A              To3/2    *           -
s4a#
```

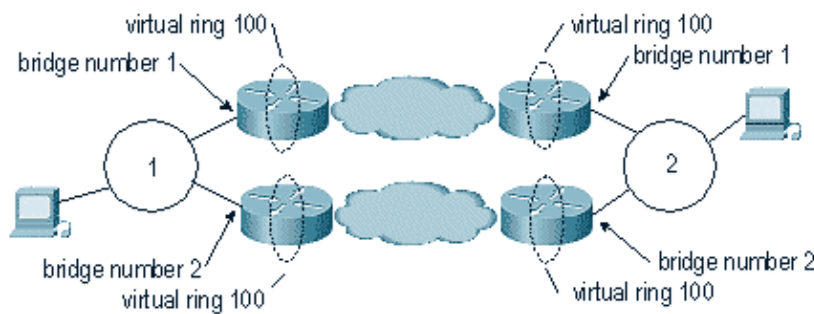
Browse the access list and look for the MAC address of the station/host you are trying to reach across the router. Proxy explorers may have cached incorrect information and is sending the frame in the incorrect direction. Try removing proxy explorers from the interfaces of the router in question and do a clear rif. If you are running local acknowledgement for RSRB, the router needs the RIF to locally acknowledge the frames. In a busy router this may be a little risky.

NetBIOS name caching is another possible cause of problems. To verify the NetBIOS name cache table, use the **show netbios** command. It provides helpful information about the number of frames that did not get sent across the router because of the caching functionality. This also relates to the **show rif** command; if the router is saving the packet from being copied to all ports, it must store information on how to reach the true destination.

To clear some of the caches discussed above, use the commands listed in the table below.

Task	Command
Clear the entries of all dynamically learned NetBIOS names.	<b>clear</b>
Clear the entire RIF cache.	<b>netbios-cache</b> <b>clear rif-cache</b>
Clear the SRB statistical counters.	<b>clear</b> <b>source-bridge</b>
Reinitialize the SSP on the Cisco 7000 series.	<b>clear sse</b>

Another common scenario is when there are multiple bridges on the same ring, as illustrated in the diagram below.



When there are multiple paths to the same ring coming from another ring, each bridge must have a different bridge number. The scenario shown in the diagram above is most common in environments with DLSw+ and RSRB.

## Hints

- Do not use **netbios name-caching** with DLSw. DLSw has a similar functionality built in. Using both will only create more problems.
- If you have a dual-TIC environment (where there are two FEPs with the same MAC address), do not run **proxy explorers** because the router will catch the RIF for both of the ticks' MAC addresses, but will only use the first in the table.
- Beware of the **clear rif** command in RSRB environments where local-acknowledgement is running.

## Debugging

Debugging SRB can be very complex. The **debug** commands that you'll use most often are **debug source error** and **debug source events**. These commands are most useful in RSRB environments.

You should try to avoid the **debug source bridge debug token ring** commands, even though they are the best to really determine if frames are actually getting through the router. These commands send large amounts of output to the screen while debugging, which can cause a router to hang. If you are telnetted to the router the effect isn't as severe, but the router CPU will be very high, and high traffic will make the effects even worse.

There is a feature in Cisco IOS 10.3 and later that allows you to apply an access list to debug output. This means that you can debug even in the busiest routers. Use this feature with caution.

To use this feature, first construct an 1100 type access list on the router, as shown below.

```
access-list 1100 permit 4000.3745.1234 8000.0000.0000 0800.1234.5678 8000.0000.0000
access-list 1100 permit 0800.1234.5678 8000.0000.0000 4000.3745.1234 8000.0000.0000
```

This access list permits traffic to/from the above two MAC addresses, allowing traffic in both directions. The 8000.0000.0000 bit mask tells the router to ignore the first bit of the MAC address. This is to avoid problems with frames that are source-routed and have the high order bit set. You can change the mask to ignore whatever you want on the MAC address. This is useful for applying the access list to all types of vendor-specific MACs.

After the access list is constructed, you can apply it to the debugging that you want to apply, as shown below.

```
s4a#debug list 1100
s4a#debug token ring
Token Ring Interface debugging is on
      for access list: 1100

s4a#
```

- *list*: (optional) An access list number in the range of 0—1199.
- *interface*: (optional) Interface type. Allowed values include:

- ◆ **channel** – IBM Channel interface
- ◆ **ethernet** – IEEE 802.3
- ◆ **fdi** – ANSI X3T9.5
- ◆ **null** – Null interface
- ◆ **serial** – Serial
- ◆ **tokenring** – IEEE 802.5
- ◆ **tunnel** – Tunnel interface

Additional **debug** commands are listed below.

- **debug llc2 errors**
- **debug llc2 packets**
- **debug llc2 state**
- **debug rif**
- **debug sdlc**
- **debug token ring**

This feature lets you debug the Token Ring interface (all packets in/out of the interface) with that access list, which is very useful in determining what is happening to the packet in the router. If you are doing RSRB, you need to issue the **debug source bridge** common under that access list to determine if that code saw the packet.

---

## Related Information

- **Local Source–Route Bridging Examples**
  - **Technical Support – Cisco Systems**
- 

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

© 2008 – 2009 Cisco Systems, Inc. All rights reserved. [Terms & Conditions](#) | [Privacy Statement](#) | [Cookie Policy](#) | [Trademarks of Cisco Systems, Inc.](#)

---

Updated: Sep 09, 2005

Document ID: 5202

---