

Устранение неполадок и общие сведения о соединении "локальный источник-маршрут"

Содержание

[Введение](#)

[Перед началом работы](#)

[Условные обозначения](#)

[Предварительные условия](#)

[Используемые компоненты](#)

[Контрольное поле маршрутизации](#)

[Поле Routing Designator](#)

[Базовая конфигурация маршрутизатора Cisco](#)

[Обозреватели мостов](#)

[Перенесение протоколов маршрутизации команды "show"](#)

[SRB-раздел выхода команды show source](#)

[Часть выходных данных команды show source, касающаяся трафика браузера](#)

[Еще команды show](#)

[Устранение неисправностей](#)

[Подсказки](#)

[Отладка](#)

[Дополнительные сведения](#)

Введение

Мостовое соединение с маршрутизацией от источника (SRB) является понятием, которым станция в Среде Token Ring может установить маршрут через многокольцевую сеть ее назначению. Этот документ обсуждает компоненты SRB и предоставляет базовую конфигурацию и сведения об устранении проблем.

Перед началом работы

Условные обозначения

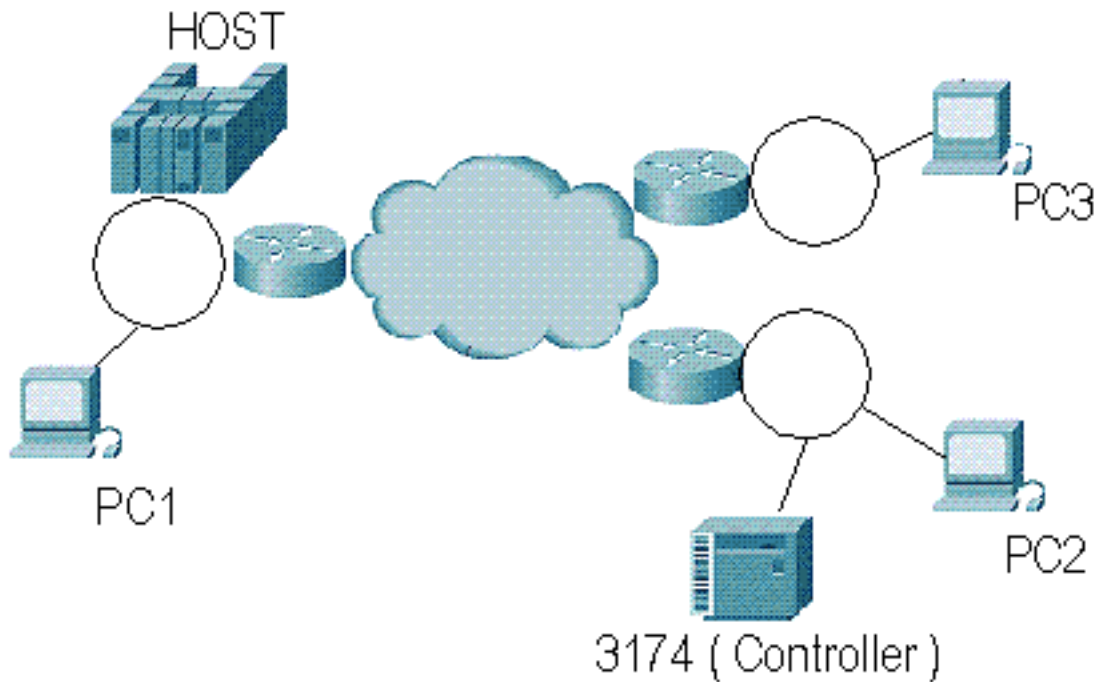
[Дополнительные сведения об условных обозначениях см. в документе Технические рекомендации Cisco. Условные обозначения.](#)

Предварительные условия

Этот документ предполагает, что читатель хорошо осведомлен относительно базовых

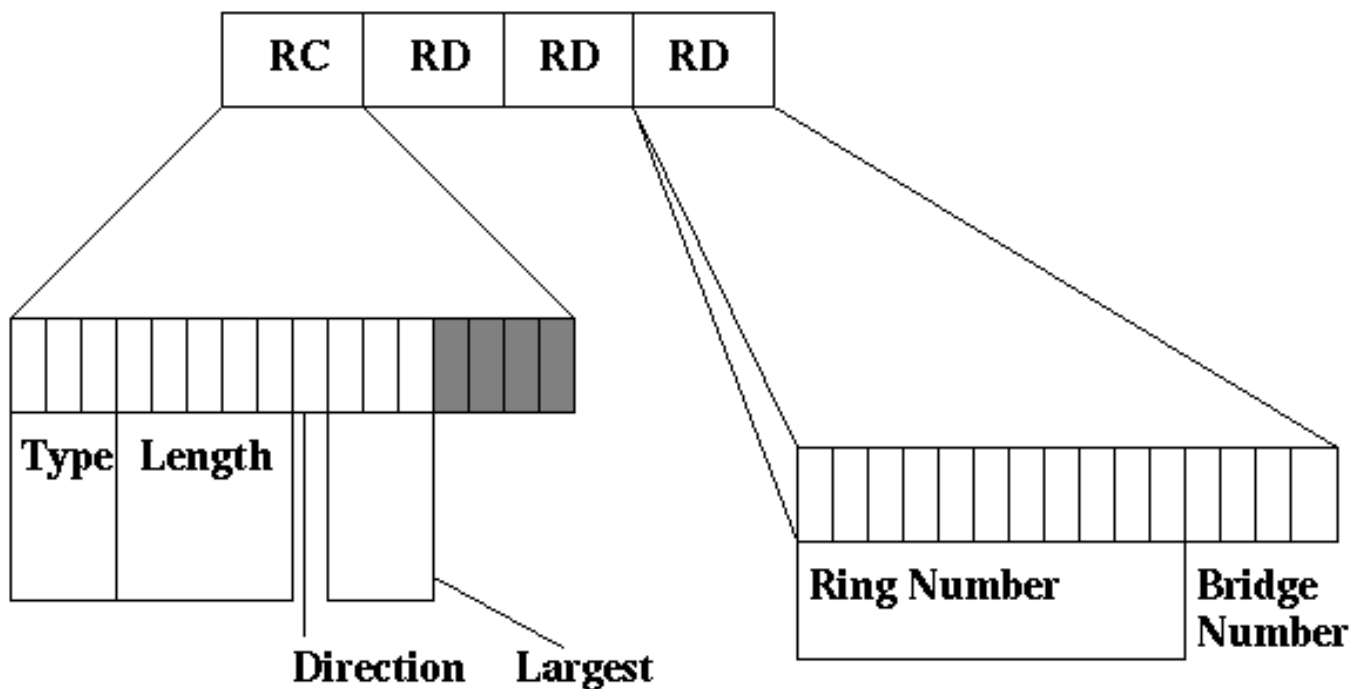
понятий мостового соединения исходного маршрута, как объяснено ниже:

Первый шаг для станции для достижения другого должен создать пакет, названный проводником. Этот пакет скопирован всеми мостами в сети. Каждый из них добавляет информацию как, туда, где пакет пересек. Поскольку это создано через сеть, конечная станция начнет получать эти пакеты. Конечная станция тогда решает, какой маршрут использовать для возврата инициатора, или она передаст другой проводник обратно так, чтобы исходящая станция могла определить маршрут.



В SRB Поле маршрутной информации (RIF) является частью проводника, который содержит информацию того, где проводник пересек. В RIF дескриптор маршрута, была информация, сохранен о пути к сети. Управление маршрутами содержит информацию о самом RIF. Следующая схема показывает RIF, в который входят эти разделы:

Routing Information Field



Используемые компоненты

Настоящий документ не имеет жесткой привязки к каким-либо конкретным версиям программного обеспечения и оборудования.

Сведения, содержащиеся в данном документе, были получены с устройств в специальной лабораторной среде. Все устройства, описанные в данном документе, были запущены с конфигурацией по умолчанию. При работе с реальной сетью необходимо полностью осознавать возможные результаты использования всех команд.

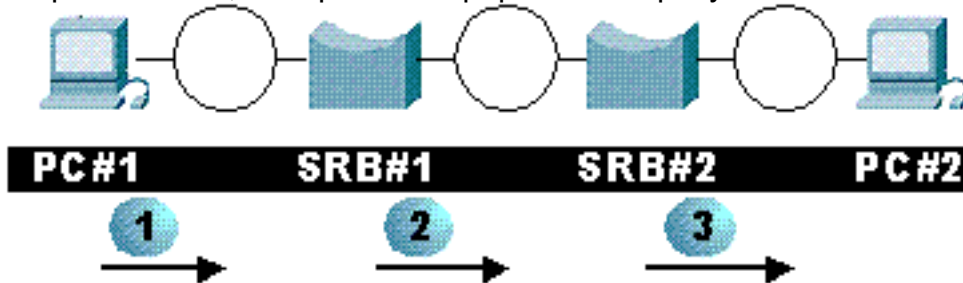
Контрольное поле маршрутизации

Поле Поля Routing Control (Управление маршрутизацией) (RC) запускает в байте 14 из Фрейма Token Ring MAC. Это - первая часть поля RIF на Фрейме Token Ring.

- Поле типа 3 бита длиной. Эта таблица ниже приводит широковещательные индикаторы. **Directed frame** указывает, что кадр содержит определенный тракт по сети и, по определению, никакое изменение не необходимо на RIF. **Все проводники маршрута** проходят всю сеть. Весь SRB должен скопировать кадр к каждому порту кроме того, который имеет вызов абонента, который уже находится в RIF. **Анализаторы одного маршрута** являются анализаторами, которые проходят через предварительно определенный путь, созданный Алгоритмом связующего дерева (STA) в мостах. Станция должна получить только один анализатор одного маршрута от сети. Проводник имеет очень важный предел на том, сколько вызовов он может держаться в поле сведений о маршрутизации. По определению Token Ring, RIF может держать в общей сложности 14 вызовов. IBM ограничил это семь для RIF на мостах в сети, однако; Cisco

также приняла это ограничение. Так, проводник, который пересек 7 вызовов, будет отброшен маршрутизатором Cisco. Существуют параметры, которые могут быть установлены в маршрутизаторе Cisco для уменьшения этого так, чтобы были отброшены пакеты, которые достигли x количества вызовов. Это - эффективный способ управлять трафиком в сети. Кроме того, маршрутизатор проверяет только длину RIF на пакете анализатора, но не обращает внимания, если направлен кадр. Если посылающая станция генерирует пакет со статическим RIF, маршрутизатор проверяет RIF для передачи целей только и мог иметь 14 предельных счетчиков переходов. Третий бит в этом поле зарезервирован (это в настоящее время не используется и проигнорировано конечными станциями).

- **Длина поля** 5 битов длиной и содержит длину RIF в байтах.
- **Направление** укусило, определяет, как RIF должен быть считан SRB в сети для придерживать пути для достижения конечной станции. Если бит установлен в **В '0'**, RIF должен быть считан слева направо. Если это установлено в **В '1'**, RIF должен быть считан справа налево.
- Самые большие кадровые биты (3 бита) определяют самый большой кадр, который может пересечь сеть, как проиллюстрировано на рисунке



ниже. Придерживающ
еся происходит с самым большим полем кадра: PC#1 создает RIF на этом кадре, и в самых больших кадровых битах помещает В '111'. Это интерпретирует в анализаторах как 49К. SRB#1 имеет MTU 4К на обоих интерфейсах. Маршрутизация от источника добавляет информацию к RIF относительно номеров кольца и модифицирует длину поля и самый большой кадр. В этом случае значение изменено на В '011'. SRB#2 имеет MTU 2К для обоих интерфейсов. Маршрутизация от источника изменяет самый большой кадр на В '010'. Диаграмма ниже перечисляет возможные значения.

Поле Routing Designator

Поле Route Designator (RD) содержит информацию о маршруте, которым пакет должен следовать для достижения станции назначения. Каждый звонит в Сети Token Ring, должно быть уникальным, или пакет может закончиться в неправильном месте. Это особенно важно в среде RSRB потому что сведения о кэшах маршрутизатора об удаленном вызове. Каждая запись в поле указателя маршрута содержит номер кольца и номер моста. Кольцевая часть 12 битов длиной, и фрагмент моста 4 бита длиной. Это позволяет вызову иметь значение от 1 до 4095 и мост значение от 1 до 16. Маршрутизаторы Cisco хранят эти значения в десятичном значении, но RIF показывает значения в шестнадцатеричном.

| RCF | ВЫЗОВ | М ос т | ВЫЗОВ | М ос т | ВЫЗОВ | М ос т |
|------------|---------|--------------|---------|--------------|---------|--------------|
| C820 | 001 | 1 | 002 | 1 | 003 | 0 |
| 1100100000 | 0000000 | 00 | 0000000 | 00 | 0000000 | 00 |

| | | | | | | |
|--------|-------|----|-------|----|-------|----|
| 100000 | 00001 | 01 | 00010 | 01 | 00011 | 00 |
|--------|-------|----|-------|----|-------|----|

Таблица выше содержит RIF в шестнадцатеричном, поскольку это отображено в выходных данных команды **show rif**. Это тогда показывает то же в двоичных файлах для декодирования его. Декодируемую версию показывают в таблице ниже.

| Позиция разряда | Значение | Описание |
|-----------------|----------|-----------------------------------|
| 1-3 | 110 | Анализатор одного маршрута |
| 4-8 | 01000 | Длина RD 8 байтов |
| 9 | 0 | Считайте RIF в прямом направлении |
| 10-12 | 010 | Самый большой кадр 2052 |
| 13-16 | 0000 | Зарезервированный |

[Базовая конфигурация маршрутизатора Cisco](#)

В этом разделе рассматриваются, как настроить маршрутизатор Cisco для SRB. Одна важная подробность этой конфигурации является понятием виртуального кольца. Виртуальное кольцо является воображаемым вызовом, который создан логически в маршрутизаторе. Это связывает все интерфейсы маршрутизатора, который важен, потому что интерфейс может только указать к одному вызову абонента, не множественным вызовам. Пример конфигурации интерфейса показывают ниже.

```
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
```

Конфигурация выше устанавливает виртуальную группу одновременного вызова 200 с **source-bridge ring-group 200** команды. Конфигурация интерфейса указывает правильно от вызова 100 для вызова 200, который является виртуальным интерфейсом.

У вас могла также быть конфигурация, в которой вы указываете к интерфейсам вместе без виртуальной группы одновременного вызова. Пример этой ситуации приведен ниже.

```
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
```

Конфигурация выше подключает предыдущие два интерфейса для SRB. Теперь, эти два интерфейса могут обмениваться кадрами SRB, но они не могут связаться ни с каким другим интерфейсом маршрутизации от источника на этом маршрутизаторе.

Виртуальное кольцо играет необходимую роль в [Протоколе RSRB](#) и [Коммутации соединения передачи данных \(DLSw\)](#), потому что необходимо настроить для этих функций.

Обозреватели мостов

Команда **source-bridge spanning** играет важную роль. Когда мы обсудили ранее разный тип анализаторов, мы упомянули все проводники маршрута и анализаторы одного маршрута. Команда **source-bridge spanning** разрешает нам передавать **single route explorer frames**. Без этого маршрутизатор просто отбросит кадр в интерфейсе. Никакие счетчики сбросов никогда не будут инкрементно увеличиваться для указания на это. Таким образом в сети со Станциями NETBIOS необходимо удостовериться, что вы позволили охватить. Также, если у вас есть **configured DLSw**, необходимо настроить команду **source-bridge spanning**, так как DLSw переходит к **single route explorer frames** использования для определения местоположения станций. В следующей конфигурации маршрутизатор настроен для передачи **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
```

Расширенную версию этой конфигурации показывают ниже.

```
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
```

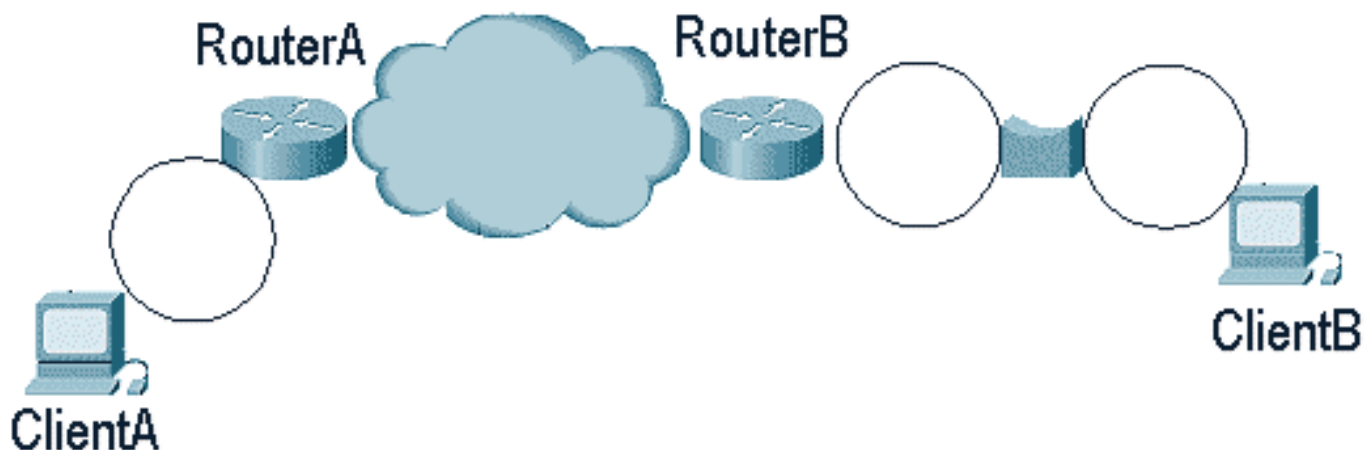
IBM Spanning Tree Protocol (STP) используется для создания связующего дерева так, чтобы **single route explorer frames** был передан через одиночный сервер блокирующими портами на среде мостовых соединений. Это подобно обычному связующему дереву IEEE только, что оно используется для анализаторов одного маршрута только. Если у вас есть эта конфигурация, вероятно, необходимо также контролировать выходные данные команды **show spann** на маршрутизаторе для определения состояния портов, так как они могли войти в состояние блокировки в зависимости от топологии. Этот маршрутизатор теперь настроен для участия в протоколе STP IBM.

```
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
```

Перенесение протоколов маршрутизации

Важная часть SRB в маршрутизаторах является способностью передать маршрутизируемый протокол через сеть с мостовыми соединениями, маршрутизируемым источником. Маршрутизатор всегда удаляет данные LLC из маршрутизируемого кадра и восстанавливает уровень LLC для интерфейса назначения. Это проиллюстрировано в приведенном ниже рисунке:



Если клиент А хочет достигнуть клиента Б, routerA должен уничтожить все данные LLC и ниже от кадра, создать кадр LLC для глобальной сети (WAN) и отправить кадр routerB. RouterB теперь принимает кадр, уничтожает LLC - информацию WAN от кадра и имеет кадр IP, готовый достигнуть клиента Б.

Маршрутизатору нужна информация с внутренней маршрутизацией для достижения clientB, потому что это - вызов далеко через SRB. RouterB тогда действует как станция сетевой концевой разделки маршрутизации от источника, где он должен найти, что путь достигает клиента Б. RouterB должен передать проводник для определения местоположения clientB. Когда clientB отвечает на routerB, он хранит поле маршрутной информации (RIF) и использует его для передачи большего количества пакетов к clientB.

Это - то, что происходит негласно в routerB, когда многокольцевой настроен на интерфейсе. Не требуется, находится ли clientB на том же вызове как routerB, потому что маршрутизатор передал бы широковещание локально и вернул бы ответ от clientB. Конфигурацию для этого показывают ниже:

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

Многокольцевой может быть настроен для определенных множественных протоколов, или с **multiring all**, который задает все маршрутизируемые протоколы. Это вступает в силу только для протоколов, которые фактически маршрутизируются маршрутизатором. Если протокол соединен, **multiring all**, это не применяется.

Команда show rif важна, когда многокольцевой настроен. Поскольку маршрутизатор должен кэшировать RIF для последующих пакетов, предназначенных для clientB, это должно

сохранить RIF, чтобы избежать иметь необходимость передать проводник за каждым пакетом, который должен достигнуть 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#
```

Для IP - сетей, в которых вам нужно к пакетам IP исходного маршрута, используйте команду **show arp** для отображения MAC-адреса для станции, которой вы пытаетесь достигнуть. Как только у вас есть MAC-адрес, можно использовать команду **show rif** для определения пути, который маршрутизатор использует для достижения той станции в с маршрутизацией источника сеть.

```
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#
```

команды "show"

Команды показа полезны при устренении проблем маршрутизации от источника. Выходные данные от команды **show interface** показывают ниже.

```
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#
```

В выходных данных команды **show interface** обратите особое внимание на следующие части:

- **ring speed** говорит вам скорость, на которой этот вызов в настоящее время в рабочем состоянии.
- Когда SRB включен, можно также проверить информацию, которая настроена для вызова и номеров моста. Например, **SRN** является номером исходного кольца, **BN** является номером моста, и **TRN** является номером конечного вызова, что виртуальное кольцо выбрало для того маршрутизатора.
- **Last ring status** предоставляет последний кольцевой статус для вызова. Например, **0x2000** указывает на программную ошибку. Список возможных значений состояния

```
показывают ниже.#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 */
```

- `drops counter` помогает определять сколько отбрасываний, там были в исходящей очереди для трафика уровня процесса и для входных буферов памяти. Это помогает определять количество дросселей.
- `output rate` и `input rate` дают общую идею того, как занятый маршрутизатор передает/получает кадры на интерфейсе.
- `Runts` и `giants` являются кадрами ниже и выше SPEC Token Ring. Вы редко встречаетесь с ними в Token Ring, но они очень полезны в Ethernet.
- `Input errors` крайне важен. Если вызов здоров, не должно быть ни одного. Если будут проблемы в вызове (такие как много шума), то CRC откажут, и кадры будут отброшены. Если `ignore count` инкрементно увеличивается, это означает, что входные буфера заполняются, и маршрутизатор сбрасывает от пакетов, предназначенных для нашего интерфейса.
- `Interface resets` может быть любой административным (выполните команду `clear int tok x`), или внутренний, когда ошибка происходит в уровне интерфейса.
- Счетчик `transitions` представляет число раз, от которого пошел интерфейс до вниз.

Команда `show source` является источником всей наиболее важной информации для устренения проблем мостового соединения исходного маршрута. Пример выходных данных от этой команды показывают ниже.

```
s4a#show source Local Interfaces: receive transmit srn bn trn r p s n max hops cnt:bytes
cnt:bytes drops 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 Ch0/2 0 0 0 0 Ch1/2 0 0 0 0 To3/0 0 0 0 0 To3/1 0 0 0 0 To3/2 762 0 0 0
```

Команда `show source` разделена на несколько разделов: информация SRB уровня интерфейса, часть RSRB и часть проводника. Проводник и части SRB объяснены ниже. Часть RSRB покрыта [Настройкой Удаленное Мостовое соединение исходного маршрута](#).

[SRB-раздел выхода команды show source](#)

Раздел source-route bridge содержит следующую информацию:

| Local Interfaces: | receive | transmit | drops |
|-----------------------------|--------------|--------------|-------|
| srn bn trn r p s n max hops | 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 |

- Для каждого интерфейса необходимо видеть SRN, BN и TRN. Это говорит вам, где информация с внутренней маршрутизацией была передана от интерфейса.

- r: Группа одновременного вызова была назначена на этот интерфейс.
- p : Интерфейс имеет использования прокси-сервера в обозревателе, настроены.
- s: Разведчики связующего дерева настроены.
- n: Кэширование имени NETBIOS настроено.
- receive и количество transmit показывают количество/байты трафика SRB, который был обработан этим интерфейсом.
- отбрасывания: количество фреймов с внутренней маршрутизацией понизилось интерфейсом маршрутизатора. Возможные причины для этих отбрасываний упомянуты ниже. Пакет SRB был получен, когда нет никакого пути (плохо настроенная **инструкция исходного моста**.) Полученный RIF является слишком длинным. Фильтр отбрасывает кадр. Группа одновременного вызова, заданная в **инструкции исходного моста** для интерфейса, не была найдена. RIF был получен, который слишком короток. Вызов абонента сразу вне группы одновременного вызова задан, но маршрутизатор не имеет его в удаленном списке рингтонов ни от какого удаленного узла. RIF говорит для вывода кадра на том же интерфейсе, от которого это был ввод. Плохо сформированный проводник был получен (Никакой RII, например). Проводник передавался с установленным битом D или с полем RIF нечетной длины в байтах. Анализатор связующего дерева был получен на интерфейсе, для которого не задан охват. Проверочный фрейм пытался выйти в вызов, который он ввел. Если бы маршрутизатор попытался передать кадр, максимальная длина RIF была бы превышена. Многоадресный кадр, не предназначенный к маршрутизатору, не имеет RIF, таким образом, маршрутизатор не может передать его.

Часть выходных данных команды show source, касающаяся трафика браузера

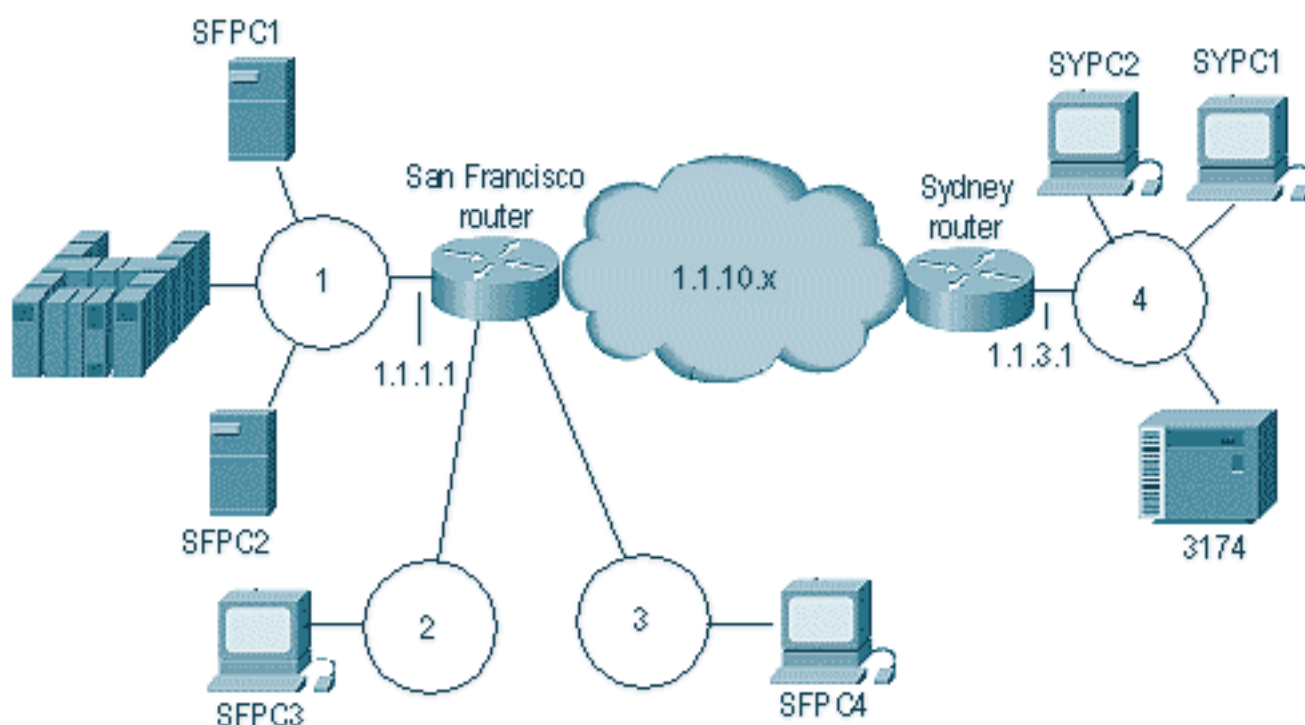
Cisco IOS разделяет трафик проводника от обычного трафика исходного маршрута. Это предоставляет нам выгодное средство устранения проблем. Одной из худших проблем с любым средством широковещания является большое число широковещательных сообщений. В Среде Ethernet слишком много широковещательных сообщений могут составлять слишком много компьютеров под той же Ethernet. В Сети Token Ring широковещательные сообщения более известны как анализаторы, потому что они пересекают от вызова до кольцевого исследования для станции на вызове. Эти анализаторы ограничены пересечением семи вызовов только. В решетчатой кольцевой среде, однако, один проводник может закончить быть скопированным многими мостами, которые могут вызвать слишком много анализаторов.

Поскольку можно дифференцироваться между анализаторами и реальными данными, можно манипулировать ими в наших интересах. Команды, перечисленные в таблице ниже, используются в маршрутизаторе для действий обозревателя.

| Задача | Команда |
|--|--|
| Установите максимальную емкость очереди анализатора. | <i>глубина</i> source- bridge explorerq- depth |
| Предотвратите зацикливания проводника в топологии избыточной сети путем фильтрации анализаторов, | source- bridge explorer- |

| | |
|---|---|
| которые были уже переданы однажды. | <code>dup-ARE-filter</code> |
| Установите максимальную скорость передачи байтов анализаторов на вызов. | <code>source-bridge</code> <code>explorer-maxrate</code> <code>maxrate</code> |
| Выключите быструю коммутацию анализаторов. | <code>no source-bridge</code> <code>explorer-fastswitch</code> |

В приведенном ниже рисунке существует два различных типа соединений: те, которые собираются от вызова звонить в маршрутизаторе и тех, которые идут через глобальную сеть (WAN). С Cisco IOS 10.3 можно выполнить быструю коммутацию анализаторы, который приблизительно в пять раз быстрее, чем процессная коммутация их. Можно использовать `explorer-maxrate` или команду `explorer-qdepth`, чтобы сделать это.



В схеме выше, SFPC4 станции передает проводник для достижения SFPC1. Маршрутизатор выполнит быструю коммутацию проводник к вызовам 1 и 2. Но маршрутизатор также передаст проводник к очереди проводника для Обработки RSRB для передачи кадра к удаленному узлу (это предполагает, что `netbios enable name cache` и команды использования прокси-сервера в обозревателе выключены).

Если бы это было огромным магазином NetBIOS, например, то количество трафика проводника было бы очень высоко. Для управления этим можно использовать параметры `explorer-qdepth` и `explorer-maxrate`. Они оба ведут себя на разных уровнях операции. Explorer maxrate работает в уровне интерфейса с выполнять быструю коммутацию кодом, и `explorer-qdepth` работает в уровне процесса. Когда используется в комбинации, эти параметры предоставляют лучший контроль анализаторов. Значение по умолчанию для `explorer-maxrate` 38400 для меньших коробок и 64000 для высокопроизводительных коробок. Настройки по умолчанию `explorer-qdepth` к 30 для всех платформ.

Ниже часть проводника выходных данных команды `show source`.

```
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
```

Для определения скорости анализаторов обратитесь к упомянутым ниже параметрам.

- `fastswitched` показывает количество анализаторов, которые были выполнены быструю коммутацию.
- `flushed` отображается, сколько анализаторов было выброшено маршрутизатором, потому что значение `maxrate` было превышено в уровне интерфейса.
- `max Bps` указывает на количество передаваемых байт в секунду, которое маршрутизатор принимает входящий через интерфейс.
- `bursts` показывает число раз, что маршрутизатор достиг максимального количества анализаторов в очереди проводника.
- `throttles` показывает число раз, что маршрутизатор убрал входные буфера интерфейса, потому что маршрутизатор не смог обслужить те буферы достаточно быстро. Это заставляет все сбойные пакеты, ждущие во входных буферах быть отброшенными.
- `output drops` является количеством анализаторов, которые были отброшены исходящие на этом интерфейсе.

Например, посмотрите на маршрутизатор Сан-Франциско в предыдущей схеме. Это в настоящее время настраивается для достижения 38,400 байт/с и имеет в общей сложности три локальных интерфейса. Каждый может достигнуть 38,400 байт/с. Это проверено каждая 10-я из секунды, так, чтобы средства, что в течение каждой 10-й из секунды маршрутизатор может поглотить 3,840 байт/с трафика проводника. Если вы делите 3,840 на 64 (который является средним пакетом анализатора NetBIOS), он равняется приблизительно 60 анализаторам в 10-ю из секунды (600 анализаторов в секунду).

Это важно, потому что это может сказать вам, сколько анализаторов маршрутизатор мог совершить нападки исходящий интерфейс. Если бы трафик возглавлялся для вызова 1 и от вызова 2 и от 3, то могла быть скорость переадресации, исходящая на вызове 1 из 1200 анализаторов в секунду. Это могло легко создать проблему в сети.

Емкость очереди анализатора является другим механизмом и в пять раз медленнее, чем `maxrate`. Все анализаторы в **емкости очереди анализатора** являются процессной коммутацией по определению. Это обычно, что ведет до RSRB, но варьируется в зависимости от настройки, потому что вы могли легко сказать маршрутизатору выполнять весь трафик в режиме обычной коммутации путем выключения **проводника-fastswitch** (Для

получения дополнительной информации о RSRB, посмотрите [Настройку Удаленное Мостовое соединение исходного маршрута](#)). Основной мерой для **обработки очереди анализатора** является пиковое значение в **выходных данных show source**. Это - число раз, что маршрутизатор достиг максимальной глубины **емкости очереди анализатора**. Если очередь будет всегда истрачена, то маршрутизатор инкрементно увеличит пакет только однажды: первоначально, что достигнут максимум.

Еще команды show

Команда **show source interface** предоставляет более короткую версию выходных данных из **источника показа**. Это полезно, если вы имеете большой маршрутизатор и хотите беглый взгляд в том, как это настроено. Можно также использовать его для определения MAC-адресов интерфейса маршрутизатора. Пример выходных данных от этой команды показывают ниже:

```
s4a#show source interface Status v p s n r Packets Line Pr MAC Address srn bn trn r x p b c IP
Address In Out 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
```

Другая полезная команда является **кратким описанием show ip interface**. Это суммирует IP-адрес на порт и сообщает, является ли интерфейс up/up. Несколько других полезных **команд показа** перечислены в таблице ниже.

| Задача | Команда |
|---|---------------------------|
| Предоставьте высокоуровневую статистику о состоянии моста с маршрутизацией от источника для определенного интерфейса. | show interfaces |
| Покажите текущее состояние любого текущего локального подтверждения и для LLC2 и для соединений SDLLC. | show localack |
| Отобразите содержание Кэша NetBIOS. | show netbios-cache |
| Отобразите содержание кэша RIF. | show rif |
| Отобразите конфигурацию моста текущего источника и разную статистику. | show source-bridge |
| Отобразите топологию связующего дерева для маршрутизатора. | show span |
| Отобразите сводку статистики Кремниевого процессора коммутатора (SSP). | show sse summary |

Устранение неисправностей

При устранении проблем любой проблемы сети запустите с нижнего уровня. Сразу не

думайте, что существует дефект в коде. Во-первых, запустите путем запуска команды **show interface** на рассматриваемых маршрутизаторах. Отобразится следующая информация:

```
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#
```

От этих выходных данных задайте себе эти вопросы:

- Интерфейсный UP/UP?
- Сколько пакетов/сек. вводит или оставляет интерфейсом?
- Есть ли какие-либо ошибки ввода (такие как CRC, кадр, переполнения, и так далее)?

Конечно, если бы вы видите 4000 ошибок ввода из 4 миллиардов входящих пакетов, которые не считали бы проблемой. Но, 4000 из 8000 переданных очень плохо.

Если вы видите интерфейс, который передает и получает пакеты, следующая команда, которая выведет, является *show interface token x accounting*. Эта команда дает вам общее представление о том, какие пакеты проходят интерфейс. Весь маршрутизированный трафик покажет независимый от трафика моста. Если существует только SRB на интерфейсе, который является всем, что вы будете видеть. Пример выходных данных от этой команды показывают ниже.

```
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#
```

В этих выходных данных вы видите интерфейс, который выполняет только SRB, протокол CDP и менеджера сети LAN. Используйте эту информацию, чтобы определить, является ли маршрутизатор ресевинг пакетами с маршрутизацией от источника на интерфейсе.

Однажды то, что вы исключили это, интерфейс передает и получает фреймы с внутренней маршрутизацией, посмотрите на конфигурацию маршрутизатора для проверки конфигурации маршрутизации от источника, как показано ниже.

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

От этой конфигурации можно решить, что маршрутизатор настроен к исходному маршруту от вызова 25 через мост 4 для вызова 31. Проверка конфигурации маршрутизатора показывает нам, что вызов 31 является настроенным виртуальным кольцом. Это также настроено для **охвата source-bridge**, что означает, что маршрутизатор передаст single route

explorer frames. Некоторые конфигурационные вопросы, которые необходимо рассмотреть, упомянуты ниже.

- Кто еще указывает для вызова 31?
- Другой интерфейс, который указывает на виртуальное кольцо 31, показывают пакеты, входящие и исходящие (с маршрутизацией источника)?
- Если интерфейс указывает на виртуальное кольцо, которое имеет source-bridge remote-peer, обратитесь к [Настройке Удаленное Мостовое соединение исходного маршрута](#) диагностировать оттуда.

Вышеупомянутые шаги будут обычно исключать проблемы конфигурации или никакие пакеты, получаемые от станции. При использовании какой-либо тип фильтрации, кэширования Имени NETBIOS или использований прокси-сервера в обозревателе и не можете быть связаны через маршрутизатор, запустите с основ. Всегда пытайтесь переместить интерфейс в его большую часть простой конфигурации. Или удалите записи или дважды - проверяют их. Неправильно созданный список доступа на интерфейсе мог также быть причиной проблем. Ниже приводится пример:

```
!  
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
```

Это заставит маршрутизатор отбросить все пакеты, адрес источника которых 4000.3745.0001. Для проверки списков доступа во всей коробке используйте команду **show access-list**. Эти выходные данные команды говорят вам все списки доступа в маршрутизаторе.

Другой причиной проблем могли быть использования прокси-сервера в обозревателе. Если вам настроили использования прокси-сервера в обозревателе, посмотрели на выходные данные команды **show rif**, как показано ниже.

```
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#
```

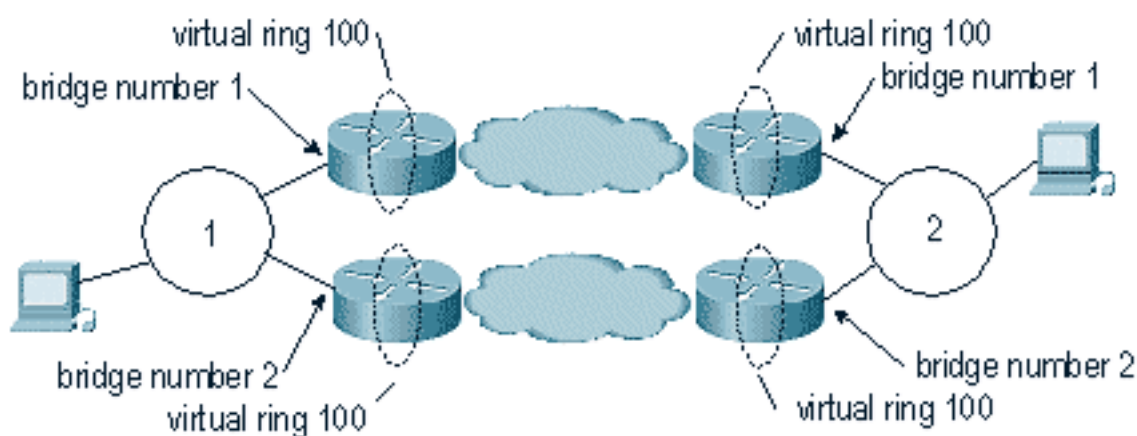
Просмотрите список доступа и ищите MAC-адрес станции/хоста, которой вы пытаетесь достигнуть через маршрутизатор. Использования прокси-сервера в обозревателе, возможно, кэшировались, неверная информация передает кадр в неверном направлении. Попытайтесь удалить использования прокси-сервера в обозревателе из интерфейсов рассматриваемого маршрутизатора и сделайте ясное сокращение штатов. При выполнении локального подтверждения для RSRB маршрутизатору нужен RIF для локального подтверждения кадров. В маршрутизаторе в состоянии занято это может быть немного опасно.

Кэширование имени NETBIOS является другой возможной причиной проблем. Для проверки таблицы NetBIOS name cache используйте команду **show netbios**. Это предоставляет полезные сведения о количестве кадров, которые не стали передаваемыми через маршрутизатор из-за кэширующейся функциональности. Это также касается команды **show rif**; если маршрутизатор сохраняет пакет от того, чтобы быть скопированным до всех портов, это должно хранить информацию о том, как достигнуть истинного назначения.

Для очистки некоторых кэшей, обсужденных выше, используйте команды, перечисленные в таблице ниже.

| Задача | Команда |
|--|----------------------------------|
| Очистите записи всех динамично изученных Имен NETBIOS. | <code>clear netbios-cache</code> |
| Очистите весь кэш RIF. | <code>clear rif-cache</code> |
| Очистите статистические счетчики SRB. | <code>clear source-bridge</code> |
| Повторно инициализируйте SSP на серии Cisco 7000. | <code>clear sse</code> |

Когда существуют множественные мосты на том же вызове, как проиллюстрировано в приведенном ниже рисунке, другой общий сценарий.



Когда существуют разнообразные пути к тому же вызову, прибывающему из другого вызова, каждый мост должен иметь другой номер моста. Сценарий, показанный в схеме выше, наиболее распространен в средах с [DLSw+](#) и [RSRB](#).

Подсказки

- Не используйте **netbios name-caching** с DLSw. DLSw Встроили подобный functionality. Использование обоих только создаст большие проблемы.
- Если у вас есть среда двойного TIC (где существует два FEP с тем же MAC-адресом), не выполняйте **использования прокси-сервера в обозревателе**, потому что маршрутизатор поймает RIF для обоих из MAC-адресов галочек, но будет только использовать первое в таблице.
- Остерегайтесь **команды clear rif** в средах RSRB, куда работает локальное подтверждение.

Отладка

Отладка SRB может быть очень сложной. **Команды отладки**, которые вы будете использовать чаще всего, являются **исходной ошибкой отладки** и **отлаживают исходные события**. Эти команды являются самыми полезными в средах RSRB.

Необходимо попытаться избежать команд **debug source bridge debug token ring**, даже при том, что они являются лучшими, чтобы действительно определить, проходят ли кадры фактически через маршрутизатор. Эти команды передают большие количества выходных данных на экран при отладке, который может заставить маршрутизатор "зависать". Если вы с установленным сеансом Telnet к маршрутизатору, эффект не так серьезен, но процессор маршрутизатора будет очень высок, и большой объем трафика сделает эффекты еще хуже.

Существует функция в Cisco IOS 10.3 и позже который позволяет вам применять список доступа к выходным данным отладки. Это означает, что можно отладить даже в самых занятых маршрутизаторах. Используйте эту функцию с осторожностью.

Для использования этой функции сначала создайте список доступа типа 1100 года на маршрутизаторе, как показано ниже.

```
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
```

Этот список доступа разрешает к/оту трафика вышеупомянутые два MAC-адреса, позволяя трафик в обоих направлениях. 8000.0000.0000 битных маски говорят маршрутизатору игнорировать первый бит MAC-адреса. Это должно избежать проблем с кадрами, которые являются с маршрутизацией источника и установили старший бит. Можно изменить маску для игнорирования независимо от того, что вы хотите на MAC-адресе. Это полезно для применения списка доступа ко всем типам определяемого поставщиком MACS.

После того, как список доступа создан, можно применить его к отладке, которую вы хотите применить, как показано ниже.

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

- *[список]*: (дополнительный) номер списка доступа в диапазоне 0 - 1199.
- *interface*: (дополнительный) Тип интерфейса. Разрешенные значения включают: канал - Канальный интерфейс IBMethernet (IEEE 802.3)fdi - X3T9.5 ANSInустрой указатель - Интерфейс NULLпоследовательный - Последовательныйtoken Ring (IEEE 802.5)tunnel - Туннельный интерфейс

Дополнительные команды отладки упомянуты ниже.

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

Эта функция позволяет вам отладить Интерфейс Token Ring (все пакеты, входящие/исходящие из интерфейса) с тем списком доступа, который очень полезен в определении, что происходит с пакетом в маршрутизаторе. При выполнении RSRB необходимо выполнить **debug source bridge**, распространенный под тем списком доступа, чтобы определить, видел ли тот код пакет.

[Дополнительные сведения](#)

- [Техническая поддержка - Cisco Systems](#)