

ATM 라우터 인터페이스의 디버그 atm 이벤트 출력 이해

목차

[소개](#)

[사전 요구 사항](#)

[요구 사항](#)

[사용되는 구성 요소](#)

[표기 규칙](#)

[소프트웨어 기능 블록 이해](#)

[사서함이란?](#)

[ATM 코어-플랫폼 드라이버 및 PCI 호스트 드라이버](#)

[PA 펌웨어에 대한 PCI 호스트 드라이버](#)

[IMA Network Module의 소프트웨어 아키텍처](#)

[관련 정보](#)

소개

전용 시스템 프로세서 모듈과 인터페이스 하드웨어에 로컬로 상주하는 여러 프로세서가 함께 작동하여 ATM VC(virtual circuit)를 통해 패킷을 성공적으로 전송하고 수신합니다. 이러한 프로세서는 메시지를 게시하여 VC 설정 및 분해, 물리적 레이어 통계 수집, 경고 생성 등의 기능을 수행함으로써 서로 통신합니다. 사랑의 편지 혹은 사랑 메시지라고 불리는 이 메시지는 한 프로세서가 메모리 블록에 씁니다. 그러면 수신 프로세서가 메시지를 읽습니다. `debug atm events` 명령의 출력은 PA-A3의 다음 출력과 같은 이 메시징 메커니즘에 창을 제공합니다.

```
Jun 17 12:48:50.631 BST: atmdx_mailbox_proc(ATM5/0/0): received report type 2
```

```
Jun 17 12:48:50.631 BST: atmdx_process_love_letter(ATM5/0/0): 2 VCs core statistics
```

```
Jun 17 12:48:55.631 BST: atmdx_mailbox_proc(ATM5/0/0): received report type 3
```

```
Jun 17 12:48:55.631 BST: atmdx_process_love_letter(ATM5/0/0): 1 VCs aux statistics
```

이 문서의 목적은 정보 메시지와 운영 문제를 가리키는 메시지를 구분하는 데 도움이 되는 샘플 `debug atm` 이벤트 출력을 보여 줍니다. 이 문서에서는 표준 ATM 인터페이스 소프트웨어 아키텍처도 검토합니다.

주의: `debug` 명령을 실행하기 전에 디버그 명령에 [대한 중요 정보를 참조하십시오](#). `debug atm events` 명령은 통계를 보고해야 하는 VC의 수와 VC 관련 이벤트의 양에 따라 프로덕션 라우터에서 대량의 중단 디버그 출력을 인쇄할 수 있습니다.

[사전 요구 사항](#)

[요구 사항](#)

이 문서에 대한 특정 요건이 없습니다.

사용되는 구성 요소

이 문서는 특정 소프트웨어 및 하드웨어 버전으로 한정되지 않습니다.

표기 규칙

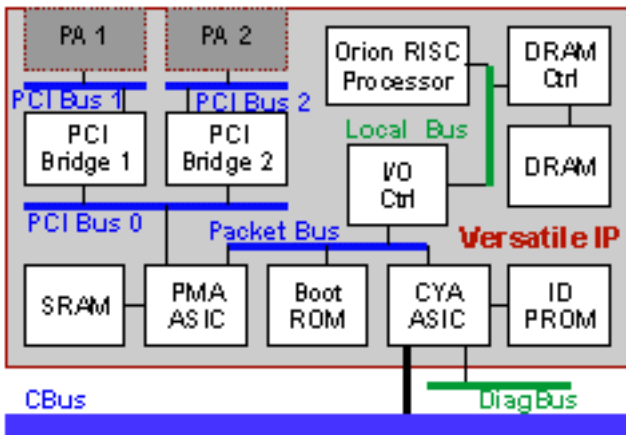
문서 규칙에 대한 자세한 내용은 [Cisco 기술 팁 표기 규칙](#)을 참조하십시오.

소프트웨어 기능 블록 이해

모든 ATM 인터페이스는 여러 블록으로 구성된 소프트웨어 아키텍처를 사용합니다. 이러한 소프트웨어 블록을 살펴보기 전에 먼저 라우터 내의 Cisco IOS® 소프트웨어 드라이버 및 PCI 버스 아키텍처를 이해해야 합니다.

드라이버를 사용하면 소프트웨어 엔지니어가 하드웨어 추상화라는 것을 구현할 수 있습니다. 엔지니어가 어떤 플랫폼에서도 실행되는 기본적인 소프트웨어 블록 집합을 만든 다음 드라이버를 사용하여 7200 Series 또는 3600 Series와 같은 특정 플랫폼에 이 플랫폼 독립 코드를 적용할 수 있습니다.

PA-A3는 세그멘테이션 및 SAR(Reassembly's) 프로세서가 7200/7400 시리즈의 길이를 실행하는 PCI(Peripheral Component Interconnect) 버스 및 RSP 플랫폼의 VIP(Versatile Interface Processor)와 인터페이스할 수 있는 PCI 호스트 드라이버를 지원합니다. PCI 버스는 VIP 또는 NPE(Network Processing Engine)/NSE(Network Services Engine)의 포트 어댑터와 호스트 메모리 간의 데이터 경로 역할을 합니다. 다음 다이어그램은 VIP2의 아키텍처와 PCI 버스의 위치를 보여줍니다.



다음 표에는 PA-A3의 소프트웨어 블록이 나열되어 있습니다.

소프트웨어 블록	합수
ATM 코어	모든 ATM 인터페이스에서 사용하는 플랫폼 또는 PA 독립 소프트웨어 기능 예를 들어 ATM 코어는 OAM 및 ILMI 관리를 처리합니다.
플랫폼	플랫폼 종속 소프트웨어 기능은 PCI 호스트 드라이

플랫폼 드라이버	버 소프트웨어를 사용하여 일반 ATM 코어 소프트웨어를 "연결"합니다. ATM 코어 및 PCI 호스트 드라이버 교환 명령, 상태 업데이트 및 브리지를 통한 통계. 플랫폼 ATM 드라이버는 show controller atm 디스플레이에 표시된 대로 수신 패킷 포워딩, 플랫폼별 초기화 함수 및 물리적 레이어 통계도 처리합니다.
PCI 호스트 드라이버	PA-A3의 SAR 칩에 대한 PCI 호스트 인터페이스를 제공합니다. 몇 가지 주요 기능을 수행합니다. <ul style="list-style-type: none"> • SAR에 펌웨어 다운로드 • 패킷 전송 • 통계 수집 • 프레임 경보 모니터링
호스트 인터페이스	각 SAR의 하드웨어 기능 블록의 일부분입니다. 다음과 같은 몇 가지 주요 작업을 수행합니다. <ul style="list-style-type: none"> • 부팅 코드를 다운로드하여 SAR을 구성하고 제어 데이터를 PCI 호스트 드라이버와 교환할 수 있습니다. • SAR에서 수신 경로의 메모리에 셀을 쓰고 전송 경로의 셀을 예약해야 하는 경우 인터럽트를 생성합니다. • 빈 버퍼를 PCI 호스트 드라이버로 반환합니다. • PCI 호스트 드라이버에서 전송된 명령을 처리하고 로컬로 수집된 통계를 PCI 호스트 드라이버로 릴레이합니다.
펌웨어	시작 또는 부팅 코드 및 수신 및 전송 SAR의 ATM 프로세서 장치(APU)에 최적화된 런타임 이미지 PCI 호스트 드라이버에서 다운로드됩니다.

RSP/VIP 플랫폼에서 플랫폼 드라이버는 RSP 시스템 이미지 및 VIP 시스템 이미지에, PCI 호스트 드라이버는 VIP 시스템 이미지의 일부입니다. 7200 플랫폼에서는 두 드라이버가 모두 시스템 이미지의 일부입니다.

PA-A3 관련 소프트웨어는 VIP 소프트웨어 또는 다른 지원 플랫폼용 시스템 소프트웨어와 함께 번들로 제공됩니다.

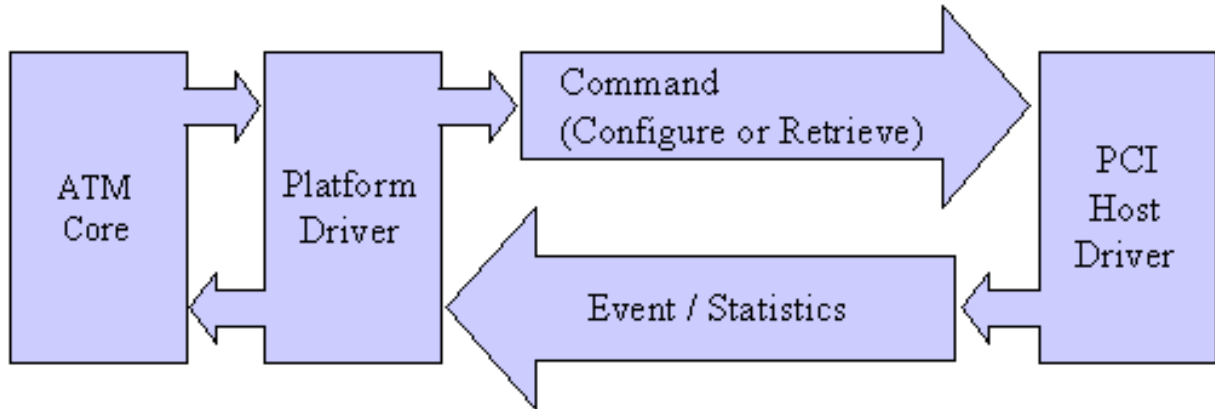
사서함이란?

위에서 언급한 것처럼, 사서함은 Cisco IOS가 두 CPU 간에 메시지를 전송하는 데 사용하는 메시징 모델의 일부입니다. 이 프로세스의 일반적인 작동 방식은 다음과 같습니다.

1. 드라이버는 메시지 버퍼를 할당합니다.
2. 사랑 메모나 문자가 메시지 버퍼를 채웁니다.
3. 수신 프로세서가 메시지 버퍼를 읽습니다.
4. 명령 버퍼 읽기를 마치면 프로세서에서 "message done" 인터럽트를 생성합니다.
5. 메시지 버퍼가 사용 가능한 버퍼 풀로 반환됩니다.

이제 이 문서에서는 위 표에 설명된 Cisco IOS Software 구성 요소를 실행하는 프로세서 간에 교환되는 두 메시지 세트를 살펴봅니다.

ATM 코어-플랫폼 드라이버 및 PCI 호스트 드라이버



PCI 호스트 드라이버는 각 패킷에 대한 VC별 통계를 수집합니다. VIP 플랫폼 드라이버는 이 통계를 매 초마다 RSP 플랫폼 드라이버에 자동으로 릴레이합니다. `show atm vc` 명령은 현재 VC 데이터를 표시합니다. VIP 플랫폼 드라이버는 프레임 통계를 10초마다 RSP에 릴레이합니다. 시스템이 초기화될 때 시스템 중단을 최소화하기 위해 인터럽트 레벨이 아닌 예약된 프로세스로 VIP의 자동 통계를 처리하는 특수 백그라운드 프로세스가 생성됩니다.

`debug atm events` 명령은 설정 및 해체 같은 VC 관련 이벤트에 출력을 인쇄합니다.

함수	설명
setupvc	VC를 설정합니다. 플랫폼 종속 드라이버는 PCI 호스트 드라이버에 요청을 전달합니다.
분해	기존 VC를 해제합니다. 플랫폼 종속 드라이버는 PCI 호스트 드라이버에 요청을 릴레이합니다.
getvc_stats	온디맨드 방식으로 VC 통계를 검색합니다. 예서는 단일 VC 요청만 지원합니다.
qos_params_verify	VC를 설정하기 전에 QoS 매개변수를 확인합니다.

PA 펌웨어에 대한 PCI 호스트 드라이버

SAR는 내부적으로 하드웨어 기능 블록으로 구성됩니다. 그러한 블록 중 하나는 ATM 특정 확장을 위한 맞춤형 로직을 갖춘 miniRISC인 APU(ATM Processing Unit)입니다. ATM 펌웨어를 실행하는 PCI 호스트 드라이버와 APU는 메시지 사서함을 통해 통신합니다. 언제든지 각 APU에 대해 해결되지 않은 명령 하나를 사용하여 PA 펌웨어가 VC 설정과 같은 특정 작업을 수행하도록 지시합니다. 펌웨어는 데이터가 변경될 경우 10초마다 VC당 및 PA당 통계를 PCI 호스트 드라이버로 릴레이합니다.

`debug atm 이벤트`에서 생성된 다음 출력은 PCI 호스트 드라이버가 펌웨어로 보낸 명령을 보여줍니다. 펌웨어는 명령의 성공을 나타내는 확인 응답만 반환합니다. 이러한 확인은 디버그 출력에 표시되지 않습니다.

```

7200-1.3(config)# int atm 6/0
7200-1.3(config-if)# pvc 1/100
  
```

```

7200-1.3(config-if-atm-vc)# vbr-nrt 45000 45000
7200-1.3#
17:07:43: atmdx_setup_vc(ATM6/0): vc:14 vpi:1 vci:100 state:2 config_status:0
17:07:43: atmdx_pas_vc_setup(ATM6/0): vcd 14, atm_hdr 0x00100640, mtu 4482
17:07:43: VBR: pcr 96000, scr 96000, mbs 94
17:07:43: vc tx_limit=1600, rx_limit=480
17:07:43: Created 64-bit VC counterss

7200-1.3(config)# int atm 6/0
7200-1.3(config-if)# no pvc 1/100
7200-1.3(config-if)#
17:08:48: atmdx_tearardown_vc(ATM6/0): idb state 4 vcd 14 state 4
17:08:48: atmdx_pas_tearardown_vc(ATM6/0): vcd 14

```

IMA Network Module의 소프트웨어 아키텍처

이제 이 문서는 2600 및 3600 라우터 시리즈에 대한 IMA(Inverse Multiplexing over ATM) 네트워크 모듈(NM)의 소프트웨어 아키텍처를 통해 이전 정보를 적용합니다.

IMA NM에는 프로세서 모듈의 기능이나 메모리를 나타내는 "호스트" 측면이 있고 네트워크 모듈 자체의 기능이나 메모리를 나타내는 "로컬" 측면이 있습니다. 호스트 측에서 플랫폼에 구애받지 않고 플랫폼에 종속된 드라이버를 실행합니다. 로컬 측에서 호스트 드라이버가 다운로드한 펌웨어를 NM의 온보드 CPU에 실행합니다. 이 이미지는 프레임 ASIC의 제어, 물리적 레이어 통계 수집, 루프백 및 경보 생성 등 물리적 레이어 기능을 처리합니다. Cisco IOS 드라이버 및 NM 펌웨어는 메일 메시지를 통해 통신합니다.

로컬 측면에서 NM IMA는 또한 메시지 사서함을 사용하여 로컬 CPU와 통신하는 IMA 드라이버를 실행합니다.

호스트 측면에서 로컬 방향으로 메시지가 주로 컨피그레이션을 위해 설계되었습니다. 이러한 메시지는 다음과 같습니다.

- 물리적 레이어 E1/T1 컨피그레이션 데이터
- IMA 그룹 구성
- 루프백 컨피그레이션
- 디버그 구성
- IMA 그룹/링크 상태 쿼리
- RFC 1406 MIB(Management Information Base) 데이터 쿼리
- IMA MIB 데이터 쿼리

로컬 측면에서 호스트 쪽으로 전송되는 메시지는 다음과 같은 라인 상태 변경 및 성능 통계를 전달하는 데 사용됩니다.

- 물리적 레이어 E1/T1 상태 변경
- IMA 그룹 상태 변경
- IMA 링크 상태 변경
- 루프백 상태 변경
- 디버그 메시지
- RFC 1406 MIB 데이터의 응답
- IMA MIB 데이터의 응답

다음 샘플 출력은 VC를 설정 및 해제하는 데 사용되는 러브 노트를 보여줍니다. 물리적 인터페이스를 달았다가 종료하지 않아 강제로 해체. "rs8234"는 NM의 SAR를 나타냅니다.

```

3640-1.1(config)# int atm2/ima2
3640-1.1(config-if)# pvc 1/1
3640-1.1(config-if-atm-vc)# shut
3640-1.1(config-if)#
*Mar 1 00:17:20.323: Reserved bw for 1/1 Available bw = 6000
*Mar 1 00:17:20.323: rs8234_setup_vc(ATM2/IMA2): vc:4 vpi:1 vci:1
*Mar 1 00:17:20.323: rs8234_setup_vc_common() VCD=260 vp/vc=17/1 etype=0
*Mar 1 00:17:20.323: rs8234_setup_cos(ATM2/IMA2): vc:4 wred_name:- max_q:0
*Mar 1 00:17:20.327: Created 64-bit VC counters
*Mar 1 00:17:20.327: rs8234_teardown_vc(ATM2/IMA2): vc:260 vpi:1 vci:1
*Mar 1 00:17:20.327: rs8234_teardown_vc proceeds (ATM2/IMA2): vc:260 vpi:1
vci:1
*Mar 1 00:17:20.327: Status and ptr is 400 Status Q is 1
*Mar 1 00:17:20.331: Resetting ATM2/IMA2
*Mar 1 00:17:20.331: rs8234_teardown_vc(ATM2/IMA2): vc:260 vpi:1 vci:1
*Mar 1 00:17:20.331: rs8234_teardown_vc proceeds (ATM2/IMA2): vc:260 vpi:1 vci:1
*Mar 1 00:17:20.331: Remove link with ports 8,links 4,channel 1
*Mar 1 00:17:22.327: %LINK-5-CHANGED: Interface ATM2/IMA2, changed state to administratively
down
3640-1.1(config-if)# no shut
3640-1.1(config-if)#
*Mar 1 00:17:31.287: Resetting ATM2/IMA2
*Mar 1 00:17:31.287: IMA config_interface ATM2/IMA2
*Mar 1 00:17:31.287: IMA config_restart ATM2/IMA2
*Mar 1 00:17:31.287: IMA restarting 0 VCs
*Mar 1 00:17:31.287: rs8234_setup_vc(ATM2/IMA2): vc:4 vpi:1 vci:1
*Mar 1 00:17:31.287: rs8234_setup_vc_common() VCD=260 vp/vc=17/1 etype=0
*Mar 1 00:17:31.287: rs8234_setup_cos(ATM2/IMA2): vc:4 wred_name:- max_q:0

```

관련 정보

- [Cisco ATM 포트 어댑터](#)
- [ATM 기술 지원](#)
- [Technical Support - Cisco Systems](#)