

Catalyst 9000 스위치의 Mac 주소 테이블 관리자 문제 해결

목차

[소개](#)

[사전 요구 사항](#)

[요구 사항](#)

[사용되는 구성 요소](#)

[배경 정보](#)

[용어](#)

[구성](#)

[다음을 확인합니다.](#)

[FED MATM의 동적 Mac 주소 학습](#)

[IOS MATM의 동적 Mac 주소 학습](#)

[IOS MATM에서 정적 Mac 주소 학습](#)

[FED MATM의 정적 Mac 주소 학습](#)

[MATM의 EVPN Mac 주소 학습](#)

[문제 해결](#)

[네트워크 연결 문제](#)

[대상 장치의 IP ARP 및 Mac 주소 확인](#)

[목적지 디바이스에 연결된 포트에서 이그레스 및 인그레스 트래픽 확인](#)

[반환 트래픽의 자세한 출력 확인](#)

[SVI 100의 Mac 주소 확인](#)

[가능한 연결 문제 원인](#)

[MATM 하드웨어 리소스 확인](#)

[MATM Syslog 오류](#)

[잠재적 픽스](#)

[관련 정보](#)

소개

이 문서에서는 Catalyst 9000 Series 스위치의 Mac Address Table Manager를 이해하고 문제를 해결하는 방법을 설명합니다.

사전 요구 사항

요구 사항

이 문서에 대한 특정 요건이 없습니다.

사용되는 구성 요소

이 문서의 정보는 다음 소프트웨어 및 하드웨어 버전을 기반으로 합니다.

- Cisco IOS® XE 16.x 및 17.x 소프트웨어의 Cisco Catalyst 9500 고성능, 9600 Series 스위치
- Cisco IOS® XE 16.x 및 17.x Software의 Cisco Catalyst 9200, 9300, 9400, 9500 Non-High Performance Series 스위치

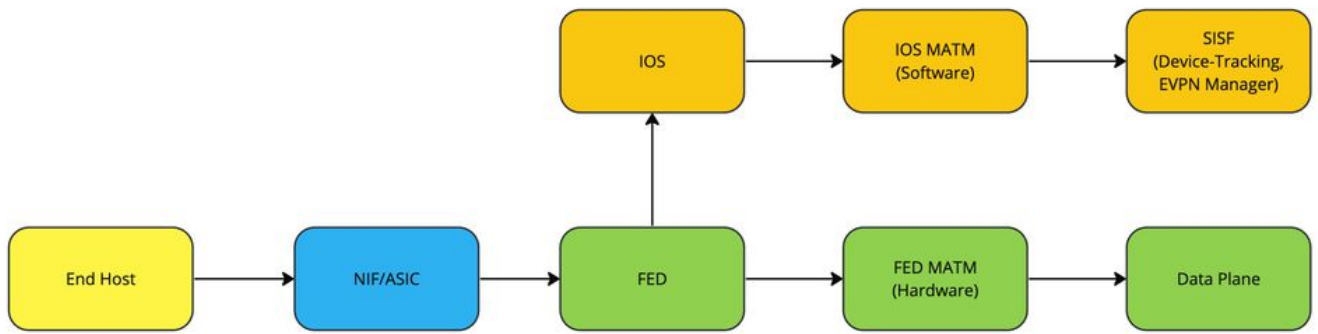
이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다. 이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다. 현재 네트워크가 작동 중인 경우 모든 명령의 잠재적인 영향을 미리 숙지하시기 바랍니다.

배경 정보

MATM(Mac Address Table Manager)은 학습한 Mac 주소가 기록되어 저장되는 데이터베이스입니다. 이 문서에 설명된 두 가지 유형의 MATM은 다음과 같습니다.

- IOS MATM(소프트웨어)
- FED MATM(하드웨어)

엔드 호스트가 패킷을 스위치에 처음 전송하면 NIF/ASIC를 거쳐 FED로 보내집니다. 이 시점에서 FED는 이 새로운 최종 호스트 정보를 IOS for IOS MTM에 입력하여 해당 정보를 데이터베이스에 기록하면서 동시에 다이어그램에서 볼 수 있는 것처럼 동일한 정보를 FED MTM에 기록합니다.



각 MATM의 중요성은 전달되는 트래픽 유형에 따라 달라집니다.

- 로컬 스위치 트래픽은 FED MATM을 사용하여 하드웨어(데이터 플레인)에서 패킷을 전달합니다.
- IOS MATM에서 구축한 디바이스 추적 데이터베이스를 사용하는 LISP(SDA Fabric) 또는 EVPN과 같은 컨트롤 플레인 기능



참고: 스위치에 SVI가 생성되면 먼저 IOS MATM에서 생성되어 작성된 다음 학습을 위해 FED MATM으로 푸시다운됩니다.

용어

매트	Mac 주소 테이블 관리자
MAC 주소	네트워크의 디바이스에 대한 12자리 고유 하드웨어 식별자
디핸들	대상 인덱스 핸들
pmap_intf	포트 맵 인터페이스
니프	네트워크 인터페이스
FED	포워드 엔진 드라이버
IOS	인터넷워크 운영 체제

데이터 플레인	하드웨어에서 전달된 트래픽
SISF	스위치 통합 보안 기능
TCAM	터너리 내용 주소화 메모리
SVI	스위치 가상 인터페이스



참고: 플랫폼별로 CLI에 스위치라는 용어가 포함될 수 있습니다(단, 항상은 아님). (show platform soft fed switch <number|active|standby> matm macTable 대 show platform soft fed active matm macTable).

구성

컨피그레이션 요구 사항은 없습니다.

다음을 확인합니다.

FED MATM의 동적 Mac 주소 학습

<#root>

Switch#

```
show platform software fed switch active matm macTable vlan 100 mac 9c54.1631.8bd1
```

VLAN	MAC	Type	Seq#	EC_Bi	Flags	machandle	siHandle	riHandle

100								

9c54.1631.8bd1

0x1

248 0 0

0x7f7490c93bd8

0x7f7490c73d98

0x0

0x7f7490a4e108

300 8

TenGigabitEthernet2/1/1

Yes

=====platform hardware details =====

Asic: 0

htm-handle = 0x7f7490c80ce8 MVID = 5 gpn = 1
SI = 0xb6 RI = 0x1a DI = 0x537d
DI = 0x537d pmap = 0x00000000 0x00000000

Asic: 1

SI = 0xb6 RI = 0x1a DI = 0x537d

DI = 0x537d pmap = 0x00000000 0x10000000 pmap_intf : [TenGigabitEthernet2/1/1]

This is a snippet from the bottom of the output of

show platform software fed switch active matm macTable

to showcase the classification of

Type

to help indicate how the

Mac Address

is being learned on the

Switch:

Type:

MAT_DYNAMIC_ADDR		0x1					
MAT_STATIC_ADDR		0x2		MAT_CPU_ADDR		0x4	
MAT_ALL_VLANS		0x10		MAT_NO_FORWARD		0x20	
MAT_DO_NOT_AGE		0x100		MAT_SECURE_ADDR		0x200	
MAT_DUP_ADDR		0x1000		MAT_NULL_DESTINATION		0x2000	
MAT_WIRELESS_ADDR		0x10000		MAT_SECURE_CFG_ADDR		0x20000	
MAT_DLR_ADDR		0x100000		MAT_MRP_ADDR		0x200000	
MAT_LISP_REMOTE_ADDR		0x1000000		MAT_VPLS_ADDR		0x2000000	
				MAT_DISCARD_ADDR		0x8	
				MAT_IPMULT_ADDR		0x40	
				MAT_NO_PORT		0x400	
				MAT_DOT1X_ADDR		0x4000	
				MAT_OPQ_DATA_PRESENT		0x40000	
				MAT_MSRP_ADDR		0x400000	
				MAT_LISP_GW_ADDR		0x4000000	



참고: 트러블슈팅은 일반적으로 IOS MATM을 확인하는 것으로 시작되지만, FED에서 먼저 학습합니다(이 경우).

IOS MATM의 동적 Mac 주소 학습

<#root>

Switch#

show mac address-table address 9c54.1631.8bd1 <--- What IOS Matm sees

Mac Address Table			

Vlan	Mac Address	Type	Ports
----	-----	-----	-----
100	9c54.1631.8bd1	DYNAMIC	Te2/1/1

<--- Showcases which vlan, how its learned

Total Mac Addresses for this criterion: 1

하드웨어 프로그래밍과 IOS 프로그래밍의 불일치를 확인합니다.

<#root>

Switch#

show platform software fed switch active matm macTable vlan 100 mac 9c54.1631.8bd1 detail

VLAN	MAC	Type	Seq#	EC_Bi	Flags	machandle	siHandle	riHandle

100								
	9c54.1631.8bd1							
	0x1							
		248	0	0				
	0x7f7490c93bd8							
	0x7f7490c73d98							
	0x0							
	0x7f7490a4e108							
		300	5					Yes

Detailed Resource Information (ASIC_INSTANCE# 0)

Number of HTM Entries:

1

Entry 0: (handle 0x7f7490c80ce8)

Absolute Index: 6442

Time Stamp: 5

KEY - vlan:5 mac:0x9c5416318bd1 l3_if:0 gpn:125 epoch:0 static:0 flood_en:0 vlan_lead_wless_flood_en: 0
MASK - vlan:0 mac:0x0 l3_if:0 gpn:0 epoch:0 static:0 flood_en:0 vlan_lead_wless_flood_en: 0 client_home:
SRC_AD - need_to_learn:0 lrn_v:0 catchall:0 static_mac:0 chain_ptr_v:0 chain_ptr: 0 static_entry_v:0 au:
DST_AD - si:0xb6 bridge:0 replicate:0 blk_fwd_o:0 v4_rmac:0 v6_rmac:0 catchall:0 ign_src_lrn:0 port_mas

=====

Detailed Resource Information (ASIC_INSTANCE# 0)

Station Index (SI) [0xb6]
RI = 0x1a
DI = 0x537d
stationTableGenericLabel = 0
stationFdConstructionLabel = 0x7
lookupSkipIdIndex = 0
rcpServiceId = 0
dejaVuPreCheckEn = 0x1
Replication Bitmap: CD

Detailed Resource Information (ASIC_INSTANCE# 1)

Station Index (SI) [0xb6]
RI = 0x1a
DI = 0x537d
stationTableGenericLabel = 0
stationFdConstructionLabel = 0x7
lookupSkipIdIndex = 0
rcpServiceId = 0
dejaVuPreCheckEn = 0x1
Replication Bitmap: LD

=====

Detailed Resource Information (ASIC_INSTANCE# 0)

Destination index = 0x537d
pmap = 0x00000000 0x00000000
cmi = 0x0
rcp_pmap = 0x0
al_rsc_cmi
CPU Map Index (CMI) [0]
ctiLo0 = 0
ctiLo1 = 0
ctiLo2 = 0
cpuQNum0 = 0
cpuQNum1 = 0
cpuQNum2 = 0
npuiIndex = 0
stripSeg = 0
copySeg = 0

Detailed Resource Information (ASIC_INSTANCE# 1)

Destination index = 0x537d
pmap = 0x00000000 0x10000000
pmap_intf : [TenGigabitEthernet2/1/1]

<--- Note the ASIC Instance # as it is based

<--- Port map interface is learned correctly

cmi = 0x0
rcp_pmap = 0x0
al_rsc_cmi
CPU Map Index (CMI) [0]

```

ctiLo0          = 0
ctiLo1          = 0
ctiLo2          = 0
cpuQNum0        = 0
cpuQNum1        = 0
cpuQNum2        = 0
npuIndex        = 0
stripSeg        = 0
copySeg         = 0
=====

```



주의: active 구문이 있는 인터페이스에 대해 detailed 명령을 실행하지만 인터페이스가 다른 스위치에 있는 경우 포트 맵 인터페이스 출력을 수신하지 않습니다.

IOS MATM에서 정적 Mac 주소 학습

이 예에서는 Switch SVI Mac Address를 사용하여 적절한 프로그래밍을 보여줍니다.

<#root>

Switch#

show run interface vlan 100

<--- Verify SVI co

Building configuration...

Current configuration : 82 bytes

!

interface Vlan100

ip address 192.168.1.2 255.255.255.0

end

Switchk#

show interface vlan 100

Vlan100 is up

, line protocol is up , Autostate Enabled

Hardware is Ethernet SVI, address is 706b.b929.f751

(

bia 706b.b929.f751

)

<--- Mac Address assigned to SVI 100 by the Switch

Internet address is 192.168.1.2/24
<snippet>

Switch#

show mac address-table address 706b.b929.f751

<--- Verify macTa

Mac Address Table			

Vlan	Mac Address	Type	Ports
----	-----	-----	-----
100	706b.b929.f751	STATIC	Vl100

Total Mac Addresses for this criterion:

1

FED MATM의 정적 Mac 주소 학습

<#root>

Switch#

show platform software fed switch active matm macTable vlan 100

<--- Verify macTa

VLAN	MAC	Type	Seq#	EC_Bi	Flags	machandle	siHandle	riHandle

100	706b.b929.f751	0x8002						
	0	0	64	0x7fc210e57908	0x7fc210cb7d78	0x0	0x0	
0	0							

Vlan100

Yes

100	0027.90be.20d1	0x101	192	0	64	0x7fc210cdc058	0x7fc210cd6da8	0x0
-----	----------------	-------	-----	---	----	----------------	----------------	-----

Total Mac number of addresses:: 2

Summary:

Total number of secure addresses:: 0

Total number of drop addresses:: 0

Total number of lisp local addresses:: 0

Total number of lisp remote addresses:: 0

*a_time=aging_time(secs) *e_time=total_elapsed_time(secs)

Type:

MAT_DYNAMIC_ADDR 0x1

MAT_STATIC_ADDR 0x2

MAT_CPU_ADDR	0x4	MAT_DISCARD_ADDR	0x8		
MAT_ALL_VLANS	0x10	MAT_NO_FORWARD	0x20	MAT_IPMULT_ADDR	0x40
MAT_DO_NOT_AGE	0x100	MAT_SECURE_ADDR	0x200	MAT_NO_PORT	0x400

<--- Note 0x8000 + 0x2 == 0x8002 ---> Routed Address that is Statically assigned on the Switch (SVI)

MAT_DUP_ADDR	0x1000	MAT_NULL_DESTINATION	0x2000	MAT_DOT1X_ADDR	0x4000
--------------	--------	----------------------	--------	----------------	--------

MAT_ROUTER_ADDR 0x8000

MAT_WIRELESS_ADDR	0x10000	MAT_SECURE_CFG_ADDR	0x20000	MAT_OPQ_DATA_PRESENT	0x40000	MAT_WIR
MAT_DLR_ADDR	0x100000	MAT_MRP_ADDR	0x200000	MAT_MSRRP_ADDR	0x400000	MAT_LIS
MAT_LISP_REMOTE_ADDR	0x1000000	MAT_VPLS_ADDR	0x2000000	MAT_LISP_GW_ADDR	0x4000000	



참고: 스위치에서 생성된 SVI는 라우티드 주소이므로 diHandle이 없습니다.

MATM의 EVPN Mac 주소 학습

Mac 주소가 학습할 Vlan을 확인하고 MATM을 확인합니다.



참고: EVPN에 대한 자세한 내용은 BGP EVPN [VXLAN 컨피그레이션 가이드를 참조하십시오](#)

<#root>

Switch#

show platform software fed switch active matm macTable vlan 201

VLAN	MAC	Type	Seq#	EC_Bi	Flags	machandle	siHandle	riHandle
201	0006.f601.cd42	0x1	32436	0	0	0x71e058dc3368	0x71e058655018	0x0

201	0006.f601.cd01	0x1	32437	0	0	0x71e058dae308	0x71e058655018	0x0
-----	----------------	-----	-------	---	---	----------------	----------------	-----

201

0006.f617.ee81 0x1000001 0 0 64

0x71e059191ee8 0x71e058e11468

0x71e058ef0d18

0x0 0

5335175 VTEP 172.16.255.4 adj_id 1376

No

Total Mac number of addresses:: 4

Summary:

Total number of secure addresses:: 0

Total number of drop addresses:: 0

Total number of lisp local addresses:: 0

Total number of lisp remote addresses:: 2

<--- Remotely learned addresses from EV

*a_time=aging_time(secs) *e_time=total_elapsed_time(secs)

Type:

MAT_DYNAMIC_ADDR 0x1

MAT_STATIC_ADDR	0x2	MAT_CPU_ADDR	0x4	MAT_DISCARD_ADDR	0x8
MAT_ALL_VLANS	0x10	MAT_NO_FORWARD	0x20	MAT_IPMULT_ADDR	0x40
MAT_DO_NOT_AGE	0x100	MAT_SECURE_ADDR	0x200	MAT_NO_PORT	0x400
MAT_DUP_ADDR	0x1000	MAT_NULL_DESTINATION	0x2000	MAT_DOT1X_ADDR	0x4000

<--- Note 0x1000000 + 0x1 == 0x1000001 ---> Mac Address remotely learned Dynamically via EVPN

MAT_WIRELESS_ADDR	0x10000	MAT_SECURE_CFG_ADDR	0x20000	MAT_OPQ_DATA_PRESENT	0x40000	MAT_WIR
MAT_DLR_ADDR	0x100000	MAT_MRP_ADDR	0x200000	MAT_MSRP_ADDR	0x400000	MAT_LIS

MAT_LISP_REMOTE_ADDR 0x1000000

MAT_VPLS_ADDR	0x2000000	MAT_LISP_GW_ADDR	0x4000000
---------------	-----------	------------------	-----------



참고: EVPN Type 플래그는 LISP Mac Learning과 동일한 MAT_LISP_REMOTE_ADDR 표기법을 사용합니다.

문제 해결

네트워크 연결 문제

이 예에서는 2 C9300-48UN의 스위치 스택을 사용합니다. 여기서 SVI 100은 네트워크의 L3 게이트웨이이며 다음과 같이 자체 Mac 주소가 올바르게 프로그래밍되지 않습니다.

- 대상 장치가 스위치 2의 포트에 연결됨
- 소스 디바이스가 스위치 1의 포트에 연결됨
- SVI 100은 게이트웨이임
- 소스 디바이스에서 대상 디바이스에 대한 연결 없음(ICMP는 테스트에 사용됨)
- 대상 장치가 스위치 1에 연결되면 연결이 복원됩니다

대상 장치의 IP ARP 및 Mac 주소 확인

```
Switch#show ip arp 192.168.1.3 Protocol Address Age (min) Hardware Addr Type Interface
Internet 192.168.1.3 6 9c54.1631.8bd1 ARPA Vlan100 <— Vlan 100에서 ARP가 올바르게 해결됨
Switch#show mac add address 9c54.1631.8bd1 Mac Address Table -----
---- Vlan Mac Address Type Ports ----
----- 100 9c54.1631.8bd1 DYNAMIC Te2/1/1
<— IOS Programmed Destination Mac Address correctly Total Mac Addresses for this criterion: 1
```

목적지 디바이스에 연결된 포트에서 이그레스 및 인그레스 트래픽 확인

주요 목표는 인그레스 트래픽이 보이도록 하는 것입니다. 이 작업은 EPC를 통해 수행할 수 있으며

이 예에서는 ICMP 트래픽을 사용합니다.

Switch#monitor capture tac 인터페이스 Te2/1/1 모두 모든 시작과 일치
<대기 시간>

스위치#모니터 캡처 tac 중지

Switch#show monitor capture tac buffer brief | i ICMP

908 4.969635 192.168.1.2 -> 192.168.1.3 ICMP 114 Echo (ping) request id=0x0008, seq=0/0, ttl=255 909 4.970165 192.168.1.2 ICMP 118 Echo (ping) reply id=0x0008, seq=0/0, ttl=254 (request in 908). <— EPC 쇼케이싱 트래픽에서 인그레스(ingress)가 표시되므로 대상 디바이스에 올바르게 응답합니다. 910 4.970425 192.168.1.2 -> 192.168.1.3 ICMP 114 에코(ping) 요청 id=0x0008, seq=1/256, ttl=255 911 4.970724 192.168.1.3 -> 192.168.1.2 ICMP 118 에코(ping) 응답 id=0x0008, seq=1/256, ttl=254(910의 요청) 912.970889 192.168.1.2 -> 192.168.1.3 ICMP 114 에코(ping) 요청 id=0x0008, seq=2/512, ttl=255 913 4.971211 192.168.1.3 -> 192.168.1.2 ICMP 118 에코(ping) 응답 id=0x0008, seq=2/512, ttl=254(912의 요청) 914.971436 168.1.2 - 1> ICMP 114 에코(ping) 요청 id=0x0008, seq=3/768, ttl=255 915 4.971558 192.168.1.3 -> 192.168.1.2 ICMP 118 에코(ping) 응답 id=0x0008, seq=3/768, ttl=254(914의 요청)

반환 트래픽의 자세한 출력 확인

Switch#show monitor capture tac buffer 상세 정보 | 프레임 909 시작 패킷 표시 시작 Ctrl + Shift + 6을 눌러 프레임 1을 종료합니다. 118 bytes on wire(944비트), /tmp/epc_ws/wif_to_ts_pipe 인터페이스에서 118 바이트 캡처(944비트), id 0 인터페이스 id: 0 (/tmp/epc_ws/wif_to_ts_pipe) 인터페이스 이름: /tmp/epc_ws/wif_to_ts_pipe 캡슐화 유형: 이더넷 (1) 도착 시간: 2024년 4월 19일 19:14:13.044770000 UTC [이 패킷의 시간 변경: 0.000000000초] 에포크 시간: 1713554053.044770000초[이전 캡처된 프레임의 시간 델타: 0.000000000초] [이전에 표시된 프레임의 시간 델타: 0.000000000초] [참조 또는 첫 번째 프레임 이후 시간: 0.000000000초] 프레임 번호: 1 프레임 길이: 118바이트(944비트) 캡처 길이: 118바이트(944비트) [프레임이 표시됨: False] [프레임이 무시됩니다. False] [프레임의 프로토콜: eth:ethertype:vlan:ethertype:ip:icmp:data] 이더넷 II, 소스: 9c:54:16:31:8b:d1(9c:54:16:31:8b:d1), 일광: 70:6b:b9:29:f7:51 (70:6b:b9:29:f7:51) <— 이 소스 mac이 192.168.1.3 대상의 arp 항목 / macTable에서 위의 출력과 일치하는지 확인합니다. 70:6b:b9:29:f7:51 (70:6b:b9:29:f7:51) <— 이 DMAC(대상 Mac 주소)가 게이트웨이(SVI 100) 주소와 일치하는지 확인합니다. 70:6b:b9:29:f7:51 (70:6b:b9:29:f7:51) 0. = LG 비트: 전역 고유 주소(공장 기본값).... 0.... = IG 비트: 개별 주소(유니캐스트) 소스: 9c:54:16:31:8b:d1 (9c:54:16:31:8b:d1) 주소: 9c:54:16:31:8b:d1 (9c:54:16:31:8b:d1) 0. = LG 비트: 전역 고유 주소(공장 기본값).... 0.... = IG 비트: 개별 주소(유니캐스트)

Switch#show interface vlan 100 Vlan100이 작동하고, 라인 프로토콜이 작동하고, Autostate Enabled 하드웨어가 이더넷 SVI이고, 주소가 706b.b929.f751(비아 706b.b929.f751)입니다. <— ICMP 회신 인터넷 주소의 DMAC와 일치합니다. 192.168.1.4/24

SVI 100의 Mac 주소 확인

Switch#show mac address-table address 706b.b929.f751 Mac Address Table -----
----- Vlan Mac Address Type Ports ---- <— IOS에 SVI 100 Mac 주소가
전혀 프로그래밍되지 않음(예기치 않음)

Switch#show platform software fed switch active matm macTable vlan 100 mac 706b.b929.f751
VLAN MAC 유형 Seq# EC_Bi Flags machandle siHandle riHandle diHandle *a_time *e_time ports
Con -----
----- 100 706b.b929.f751
0x80 64 0x7fc210e57908 0x7fc210cb7d78 0x0 0x0 0 0 Vlan1010 Yes=====platform details
===== Asic: 0 <— 스위치 1(액티브)의 Matm에 SVI 100 MAC이 올바르게 프로그래밍되었습니다.
htm-handle = 0x7fc210cb9e68 MVID = 5 gpn = 1 SI = 0x2d RI = 0x1 DI = 0x5234 Asic: 1 SI =
0x2d RI = 0x1 DI = 0x5234



참고: 대상 장치가 스위치 1에 연결될 때 연결이 복원되는 이유는 MATM이 스위치 2에 비해
여전히 제대로 프로그래밍되었기 때문입니다.

Switch#show platform software fed switch 2 matm macTable vlan 100 mac 706b.b929.f751 총
Mac 주소 수: 0 <— 스위치 2의 Matm에 SVI 100 MAC 프로그램 요약이 없습니다. 총 보안 주소 수:
0 총 삭제 주소 수: 0 총 lisp 로컬 주소 수: 0 총 lisp 원격 주소 수: 0

가능한 연결 문제 원인

MATM 하드웨어 리소스 확인

Mac 주소 프로그래밍을 담당하는 스위치 하드웨어 리소스가 소진되면 더 이상 주소를 학습할 수 없
습니다

Switch#show platform hardware fed switch active fwd-asic resource tcam utilization Code: EM -
Exact_Match, I - Input, O - Output, IO - Input & Output, NA - ASIC에 대해 적용할 수 없는 CAM 사
용률 [0] 테이블 하위 유형 Dir 최대 사용됨 %Used V4 V6 MPLS 기타 -----
----- Mac 주소 테이블 EM I 32768 31788 97.01% 0 0 0
0 0 0 0 31788 <— 95% 이상의 값 찾기 Mac 주소 테이블 TCAM I 1024 199.51% 0 0 0 0 1019 L3
멀티캐스트 EM I 8192 0.00 I3 멀티캐스트 TCAM I 512 9 1.76% 3 6 0 0 0 L2 멀티캐스트 EM I
8192 0 0.00% 0 0 0 0 L2 멀티캐스트 TCAM I 512 11 2.15% 3 8 0 0 IP 경로 테이블 EM I 24576 3
0.01% 2 0 1 0 IP 경로 테이블 TCAM I 8192 19 0.23% 6 10 2 1 QOS ACL TCAM IO 5120 85.66%
28 38 0 19 TCAM I 45 0.88% 15 20 0 10 TCAM O 40 0.78% 13 18 0 9 보안 ACL TCAM IO 5120
131 2.56% 26 60 0 45 TCAM I 88 1.72% 12 36 0 40 TCAM O 43 0.84% 14 24 0 5 Netflow ACL
TCAM I 256 2.34% 2 2 0 2 PBR ACL TCAM I 1024 36.52% 30 0 0 Netflow ACL TCAM O 768
0.78% 2 2 0 2 Flow SPAN ACL TCAM IO 1024 13 1.27% 3 6 0 4 TCAM I 5 0.49% 1 2 0 2 TCAM
O 8 0.78% 2 4 0 2 컨트롤 플레인 TCAM I 512 2 290 56.64% 138 106 0 46 터널 종료 TCAM I 5120

3.91% 8 12 0 0 0 Lisp Inst 매핑 TCAM I 2048 1 0.05% 0 0 0 1 보안 연결 TCAM I 254 1.56% 2 2 0 0 0 CTS Cell Matrix/VPN Label EM O 8192 0.00% 0 0 0 0 0 CTS Cell Matrix/VPN Label TCAM O 512 1 0.20 0 0 0 1 클라이언트 테이블 EM I 4096 0.00% 0 0 0 0 0 클라이언트 테이블 TCAM I 256 0.00% 0 0 0 0 0 입력 그룹 LE TCAM I 1024 0.00% 0 0 0 출력 그룹 LE TCAM O 1024 0.00% 0 0 0 0 0 Macsec tcam I 256 2 0.78% 0 0 0 2



참고: 하드웨어 리소스에 대한 자세한 내용은 [Catalyst 9000](#) 스위치의 [하드웨어 리소스 이해 설명서](#)를 참조하십시오.

MATM Syslog 오류

MATM 로그 메시지	정의	복구 작업
MATM-3-MAX_ENTRIES: 스위치 1 F0/0: fed: 최대 MAC 주소 수에 도달했습니다.32768	Mac 주소용으로 예약된 하드웨어의 공간이 부족합니다.	스위치에서 학습하는 Mac 주소의 스케일 수 감소

잠재적 픽스

옵션 #1:

스위치에서 학습한 Mac 주소의 양을 줄입니다.

- 네트워크 루프가 발생할 수 있으며, 문제가 해결되면 하드웨어 리소스가 감소하고 정상적인 Mac 학습이 계속됩니다
- 네트워크 확장에 영향을 미치며 더 큰 하드웨어 용량을 가진 스위치를 사용합니다. (예: C9300에는 최대 32768개의 Mac 주소가 있지만 C9500H에는 최대 82,000개가 있음)

옵션 #2:

합법적인 오프로그래밍이 발생하고 있습니다.

1. 모든 관련 데이터 수집:

- 기술 지원 표시
- 추적 로그 아카이브

- 디버그 MATM

2. 장치 전원 껐다 켜기

- 문제가 계속되면 Cisco TAC에서 케이스를 여십시오.

관련 정보

- [Catalyst 9000 스위치의 하드웨어 리소스 이해](#)
- [BGP EVPN VXLAN 컨피그레이션 가이드](#)

이 번역에 관하여

Cisco는 전 세계 사용자에게 다양한 언어로 지원 콘텐츠를 제공하기 위해 기계 번역 기술과 수작업 번역을 병행하여 이 문서를 번역했습니다. 아무리 품질이 높은 기계 번역이라도 전문 번역가의 번역 결과물만큼 정확하지는 않습니다. Cisco Systems, Inc.는 이 같은 번역에 대해 어떠한 책임도 지지 않으며 항상 원본 영문 문서(링크 제공됨)를 참조할 것을 권장합니다.