

# SNMP 프로세스로 인한 높은 CPU 사용률 문제 해결

## 목차

---

[소개](#)

[사용되는 구성 요소](#)

[로그 수집](#)

[로그 분석](#)

[SNMP 보기 컨피그레이션](#)

[문제 해결 팁](#)

[결론](#)

[관련 정보](#)

---

## 소개

이 문서에서는 9800 Wireless Lan Controller에서 SNMP 프로세스에 대한 높은 CPU 사용률을 해결하고 문제를 해결하는 구조화된 접근 방식을 설명합니다.

## 사용되는 구성 요소

이 문서의 정보는 다음 소프트웨어 및 하드웨어 버전을 기반으로 합니다.

- 무선 컨트롤러: 17.09.03을 실행하는 C9800-80-K9

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.

## 로그 수집

CPU 사용률 패턴 식별 SNMP 프로세스와 연결된 높은 CPU 사용률 보고서를 받으면 첫 번째 작업 과정은 지정된 기간 동안 자세한 로그를 수집하는 것입니다. 이는 SNMP 프로세스가 가장 활발하고 리소스가 많이 사용되는 시간을 정확히 파악하는 데 필수적인 CPU 사용량의 패턴이나 경향을 설정하는 데 도움이 될 것입니다.

로그 수집을 시작하기 전에 문제 해결 프로세스를 지원하는 데 사용되는 특정 정보를 수집해야 합니다. 먼저 문제에 대한 몇 가지 정보를 수집합니다.

- 시스템에 스파이크가 발생하거나 지속적으로 사용량이 많습니까?
- 두 경우 모두 사용률이 어떻게 됩니까?
- CPU 사용률이 높은 빈도는 얼마입니까?
- 각 SNMP 서버가 WLC를 폴링하는 빈도는 얼마나 됩니까?

- 누가 제일 많이 떠들어 대나요?

10분 간격으로 2분 간격으로 9800 WLC의 명령 출력을 수집합니다. 이 데이터를 사용하여 SNMP 프로세스와 관련된 CPU 사용률 문제를 분석할 수 있습니다.

```
#terminal length 0
#show clock
#show process cpu sorted | exclude 0.0
#show process cpu history
#show processes cpu platform sorted | exclude 0.0
#show snmp stats oid
#show snmp stats hosts
```

## 로그 분석

이러한 로그를 수집한 후 분석을 통해 그 영향을 파악해야 합니다.

샘플 CPU 사용률 로그를 살펴보고 CPU를 가장 많이 사용하는 SNMP 프로세스를 알아보겠습니다.

<#root>

```
WLC#show process cpu sorted | exclude 0.0
CPU utilization for five seconds: 96%/7%; one minute: 76%; five minutes: 61%
  PID Runtime(ms)   Invoked    uSecs   5Sec   1Min   5Min TTY Process
   250   621290375     58215467   10672
58.34% 39.84% 34.11%    0 SNMP LA Cache pr  <-- High utilization

   93   167960640     401289855    418 14.50% 11.88%  9.23%  0 IOSD ipc task
  739   141604259     102242639   1384  8.57%  6.95%  7.21%  0 SNMP ENGINE
   763         7752         34896     222  4.00%  3.41%  1.83%  5 SSH Process
   648   6216707      181047548    34  0.72%  0.37%  0.31%  0 IP SNMP
   376   3439332      51690423    66  0.40%  0.36%  0.25%  0 SNMP Timers
   143   3855538      107654825    35  0.40%  0.35%  0.23%  0 IOSXE-RP Punt Se
   108   6139618      17345934    353  0.40%  0.30%  0.34%  0 DBAL EVENTS
```

정렬된 show process cpu의 출력 | exclude 0.0 명령은 SNMP 프로세스가 CPU 리소스를 과도하게 사용하고 있음을 나타냅니다. 특히 SNMP LA 캐시 사전 프로세스는 CPU를 가장 많이 사용하는 프로세스이며, 그 다음은 기타 SNMP 관련 프로세스입니다.

다음 명령 집합은 SNMP 활용률이 높은 프로세스를 드릴다운하는 데 도움이 됩니다.

<#root>

```
WLC#show snmp stats oid

time-stamp           #of times requested           OID
11:02:33 Austral Jun 8 2023
```

11:02:23	Austral	Jun 8 2023	1	sysUpTime
11:02:23	Austral	Jun 8 2023	17	cLSiD11SpectrumIntelligenceEnable
11:02:23	Austral	Jun 8 2023	1	cLSiD11SpectrumIntelligenceEnable
11:02:23	Austral	Jun 8 2023	6	cLSiD11Band
11:02:23	Austral	Jun 8 2023	1	cLSiD11Band
11:02:23	Austral	Jun 8 2023	1	cLSiD11Band
11:02:23	Austral	Jun 8 2023	1	cLSiD11Band
11:02:19	Austral	Jun 8 2023	24	cLcCdpApCacheApName
11:02:19	Austral	Jun 8 2023	1	cLcCdpApCacheDeviceIndex
11:02:19	Austral	Jun 8 2023	9	cLApCpuAverageUsage
11:02:19	Austral	Jun 8 2023	1315	cLApCpuCurrentUsage
11:02:19	Austral	Jun 8 2023	2550	bsnAPIfDBNoisePower

show snmp stats oid 명령의 출력은 다양한 OID가 폴링되는 빈도를 나타냅니다. 특정 OID인 bsnAPIfDBNoisePower는 요청 수가 매우 많아 두드러집니다. 이는 이 OID의 적극적인 폴링이 WLC에서 관찰된 높은 CPU 사용률에 영향을 미칠 가능성이 있음을 시사합니다.

OID bsnAPIfDBNoisePower가 수행하는 작업과 데이터 저장 시간을 파악해 보겠습니다.

SNMP Object [Navigator\(SNMP 개체 탐색기\)](#)로 이동하여 OID "bsnAPIfDBNoisePower"를 검색합니다.

Translate OID into object name or object name into OID to receive object details

Enter OID or object name:  examples -  
 OID: 1.3.6.1.4.1.9.9.27  
 Object Name: ifIndex

#### Object Information

##### Specific Object Information

Object	bsnAPIfDBNoisePower
OID	1.3.6.1.4.1.14179.2.2.15.1.21
Type	<a href="#">Integer32</a>
Permission	read-only
Status	current
MIB	<a href="#">AIRESpace-WIRELESS-MIB</a> ; - <a href="#">View Supporting Images</a>
Description	"This is the average noise power in dBm on each channel that is available to Airespace AP"

OID 검색 결과

이제 bsnAPIfDBNoisePower 객체가 각 AP에서 보고한 대로 각 채널의 노이즈 파워를 보고한다는 것을 이해하게 되었습니다. WLC에서 관리하는 채널 및 AP의 수가 많으므로 이 OID에서 생성되는 SNMP 데이터는 상당할 수 있습니다. WLC가 많은 수의 AP에 서비스를 제공하는 경우 이 OID를 폴

링하여 생성되는 데이터의 양이 엄청날 수 있습니다. 이는 WLC가 이러한 광범위한 SNMP 요청을 처리할 때 높은 CPU 사용률로 이어질 수 있습니다.

마찬가지로, 공격적으로 폴링되는 특정 OID의 동작을 이해해야 합니다.

다음 명령은 WLC를 폴링하는 SNMP 서버를 파악하는 데 도움이 됩니다.

```
WLC#show snmp stats hosts
Request Count    Last Timestamp  Address
77888844        00:00:00 ago    10.10.10.120
330242          00:00:08 ago    10.10.10.150
27930314        00:00:09 ago    10.10.10.130
839999          00:00:36 ago    10.10.10.170
6754377         19:45:34 ago    10.10.10.157
722            22:00:20 ago    10.10.10.11
```

이 명령은 SNMP 서버 목록을 요청 수 및 폴링 활동의 마지막 타임스탬프와 함께 제공합니다.

9800 WLC를 폴링하는 서로 다른 서버가 여러 개 있는 것을 확인할 수 있습니다. 지난 10분 동안 수집된 전체 로그 데이터를 보면 폴링 빈도도 측정할 수 있습니다.

이제 각 서버로 이동하여 위반 OID가 폴링되는 빈도를 확인할 수 있습니다. 이 예에서 OID는 30초마다 폴링되며, 이는 필요한 경우보다 훨씬 더 빈번합니다. WLC는 180초마다 RF/RRM 데이터를 수신하므로, 30초마다 OID를 폴링하면 불필요한 처리가 발생하고 CPU 사용률이 높아집니다.

문제가 되는 OID와 서버가 파악되면 WLC의 로드를 줄이기 위해 여러 가지 다른 솔루션을 시도할 수 있습니다.

1. SNMP 서버의 폴링 빈도를 줄입니다.
2. 작업 사용에 OID가 필요하지 않은 경우 SNMP 서버에서 해당 OID의 폴링을 비활성화합니다.
3. SNMP 서버를 제어할 수 없는 경우 SNMP 보기를 사용하여 위반 OID를 차단할 수 있습니다.

## SNMP 보기 컨피그레이션

차단할 OID를 제외하는 새 뷰를 정의합니다. 예를 들어, OID 1.3.6.1.4.1.14179.2.2.15.1.21을 차단하고 새 뷰를 생성한 다음 OID를 뷰에 연결합니다.

```
<#root>
```

```
snmp-server view blockOIDView
```

```
1.3.6.1.4.1.14179.2.2.15.1.21
```

```
excluded
```

```
<-- This is the OID of bsnAPIfDBNoisePower
```

```
snmp-server community TAC view blockOIDView RO
```

```
<-- This command assigns the blockOIDView to the community myCommunity with read-only (RO) access.
```

```
snmp-server group TAC v3 priv read blockOIDView
```

```
<-- This command assigns the blockOIDView to the group myGroup with the priv security level for SNMPv3.
```

## 문제 해결 팁

- 기존 CPU 사용량: SNMP 프로세스로 인해 사용량이 많지 않을 때 일반 CPU 사용률 레벨을 문서화합니다.
- SNMP 컨피그레이션: 커뮤니티 문자열, 버전(v2c 또는 v3) 및 액세스 목록을 비롯한 현재 SNMP 컨피그레이션 설정을 검토합니다.
- SNMP 모범 사례: 9800 WLC 모범 사례 문서를 사용하고 SNMP에 대해 권장되는 컨피그레이션을 최대한 일치시킵니다.

```
C9800(config)#snmp-server subagent cache
C9800(config)#snmp-server subagent cache timeout ?
<1-100> cache timeout interval (default 60 seconds)
```

- SNMP 폴링 빈도: 빈도가 높으면 CPU 로드가 증가할 수 있으므로 SNMP 쿼리로 WLC를 폴링하는 빈도를 결정합니다.
- 네트워크 토폴로지 및 SNMP 관리자: 네트워크 설정을 이해하고 WLC와 상호 작용하는 모든 SNMP 관리자를 식별합니다.
- 시스템 업타임: 마지막 재부팅 이후 경과한 시간을 확인하여 업타임과 CPU 사용률 간에 상관관계가 있는지 확인합니다.
- 최근 변경 사항: WLC 컨피그레이션 또는 네트워크에 대한 최근 변경 사항이 높은 CPU 사용률의 시작과 일치할 수 있습니다.
- 9800 WLC에서는 텔레메트리를 중점적으로 살펴보았습니다. 텔레메트리는 WLC가 쿼리할 필요 없이 관련 정보를 서버로 전송하는 "푸시" 모델에서 작동합니다. SNMP 쿼리가 WLC CPU 사이클을 사용하고 운영 문제를 일으키는 경우 원격 분석으로 이동하는 것이 좋습니다.

## 결론

CPU 사용률 데이터를 체계적으로 분석하고 이를 SNMP 폴링 활동과 상관 관계를 분석함으로써 Cisco 9800 WLC의 SNMP 프로세스로 인해 발생하는 높은 CPU 사용률 문제를 해결하고 해결할 수 있습니다. 구현 후 모니터링은 트러블슈팅의 성공을 확인하고 최적의 네트워크 성능을 유지하기 위해 필수적입니다.

## 관련 정보

- [OID를 사용하여 SNMP를 통해 Catalyst 9800 WLC 모니터링](#)
- [SNMP V2, V3 및 NetCONF를 사용하여 Prime Infrastructure로 Catalyst 9800 Wireless Controller Series 관리](#)
- [Simple Network Management Protocol 보안](#)

- [Cisco Catalyst 9800 Series 구성 모범 사례](#)

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.