

# BGP RR 확장 고려 사항 및 KPI 모니터링

## 목차

---

### [소개](#)

### [HW/SW 플랫폼 선택](#)

### [확장 및 성능 고려 사항](#)

#### [BGP 피어 수](#)

#### [주소군](#)

#### [업데이트 그룹 수](#)

#### [RPL의 복잡성\(경로 정책\)](#)

#### [업데이트 빈도](#)

#### [TCP MSS 및 인터페이스/경로 MTU](#)

#### [Dual-RP 라우터의 NSR](#)

#### [느린 피어](#)

#### [Nexthop trigger-delay](#)

### [검증된 다차원 BGP RR 스케일의 예](#)

### [설계 고려 사항](#)

### [BGP KPI\(핵심 성과 지표\) 모니터링](#)

#### [데이터 경로 전달자 모니터링](#)

##### [XRv9000 데이터 플레인 에이전트 모니터링\(DPA\)](#)

##### [ASR9000NP\(Network Processor\) 모니터링](#)

#### [LPTS 모니터링](#)

#### [모니터 SPP](#)

#### [NetIO 모니터링](#)

#### [XIPC 대기열 모니터링](#)

#### [BGP 입력 및 출력 대기열 모니터링](#)

#### [BGP 메시지 속도 모니터링](#)

#### [CPU 사용률 모니터링](#)

#### [TCP 통계 모니터링](#)

#### [메모리 사용률 모니터링](#)

#### [BGP 프로세스 성능 모니터링](#)

#### [BGP 통합 모니터링](#)

---

## 소개

이 문서에서는 BGP RR(Border Gateway Protocol) Route-Reflectors가 BGP RR 성능 모니터링을 달성하고 안내하는 최대 규모의 주요 기여자에 대해 설명합니다.

## HW/SW 플랫폼 선택

대규모 BGP RR은 일반적으로 인터넷 서비스 공급자가 제공하는 서비스를 전달하는 패킷의 전달

경로에 없습니다. 따라서 데이터 경로에서 패킷을 전달하는 것이 대부분인 BGP RR 및 라우터에 대한 하드웨어 요구 사항은 다릅니다. 표준 라우터는 강력한 데이터 경로 전달 요소와 비교적 보통 수준의 제어 경로 요소로 구축됩니다. BGP RR은 제어 계획에서 모든 작업을 수행합니다.

Cisco IOS® XR 제품군 내에서 BGP RR 역할을 위한 3가지 유형의 HW/SW 플랫폼을 선택할 수 있습니다.

물리적 Cisco IOS XR 라우터	Cisco IOS XRv 9000 어플라이언스	Cisco IOS XRv 9000 라우터 (XRv9k라고도 함)
<ul style="list-style-type: none"> <li>• 보통의 컨트롤 플레인 용량(일반적으로 RP XR VM에 할당된 CPU 코어 2개 ~ 6개)</li> <li>• 사용되지 않는 데이터 경로 용량</li> </ul>	<ul style="list-style-type: none"> <li>• 높은 컨트롤 플레인 용량 (Cisco UCS M5 기반 어플라이언스에서 36개의 CPU 코어가 RP XR VM 전용)</li> <li>• 데이터 경로 용량과 제어 경로 용량을 균등하게 분리</li> <li>• XRv9k 이미지는 최대 성능을 위해 베어 본에서 실행됩니다.</li> </ul>	<ul style="list-style-type: none"> <li>• 사용자 지정 가능한 컨트롤 플레인 용량</li> <li>• BGP RR 이미지를 사용할 경우 데이터 경로와 제어 경로 간의 균등한 분할.</li> <li>• 가상화의 추가 레이어는 성능에 영향을 줍니다.</li> </ul>

이 문서에서는 XRv9k Appliance가 최고의 제어 평면 용량과 최대 성능을 제공하므로 BGP RR에 최적의 플랫폼을 선택할 수 있습니다.

## 확장 및 성능 고려 사항

지원되는 데이터 플레인 엔티티의 스케일은 규모에 따라 달라지는 경우가 거의 없기 때문에 비교적 쉽게 표현할 수 있습니다. 예를 들어, TCAM 조치는 활성 TCAM 엔트리 수에 관계없이 동일한 시간이 소요됩니다.

제어 평면 엔티티의 지원되는 스케일은 규모와 성능이 상호 연결되기 때문에 훨씬 더 복잡합니다. 1M 경로가 있는 BGP RR을 고려하십시오. 이 BGP 테이블을 유지 관리하기 위해 BGP 프로세스가 수행해야 하는 작업은 다음에 따라 달라집니다.

1. 얼마나 많은 BGP 피어가 활성 상태입니까?
2. 어떤 주소지가 활성 상태입니까?
3. 업데이트 그룹으로 어떻게 배포됩니까?
4. RPL(경로 정책)의 복잡성
5. 업데이트 빈도(수신 업데이트 및 발신 업데이트 - 광고 간격)
6. TCP MSS, 인터페이스/경로 MTU - 이를 조정하면 성능 향상
7. Dual-RP인 경우 NSR이 활성화됨
8. 별도의 업데이트 그룹에 없는 알려진 저속 피어
9. Nexthop trigger-delay 값

## BGP 피어 수

BGP 피어의 수는 일반적으로 첫 번째이며, BGP 규모를 고려할 때 가장 먼저 떠오르는 경우가 많습니다. 지원되는 BGP 스케일은 BGP 피어 수를 언급하지 않고는 표현할 수 없지만 가장 중요한 요소는 아닙니다. 다른 많은 측면들도 똑같이 관련이 있다.

## 주소군

AF(Type of Address Family)는 일반적인 구축에서는 단일 경로의 크기에 영향을 주기 때문에 BGP 성능 고려 사항에서 중요한 요소입니다. 단일 TCP 세그먼트로 압축할 수 있는 IPv4 경로 수가 VPNv4 경로 수보다 훨씬 많습니다. 따라서 동일한 규모의 BGP 테이블 변경 사항에 대해 IPv4 BGP RR은 VPNv4 BGP RR에 비해 더 적은 작업이 가능합니다. 모든 경로에 많은 수의 커뮤니티가 추가된 구축에서는 AF 간의 차이가 줄어들지만 단일 경로의 크기가 더 커지므로 고려해야 합니다.

## 업데이트 그룹 수

BGP 프로세스는 동일한 업데이트 그룹의 모든 멤버에 대해 단일 업데이트를 준비합니다. 그런 다음 TCP 프로세스는 업데이트 그룹의 각 구성원을 향해 업데이트 데이터를 TCP MSS에 따라 필요한 수의 TCP 세그먼트로 분할합니다. 명령을 사용하여 활성 업데이트 그룹 및 해당 멤버를 볼 수 `show bgp update-group` 있습니다. 동일한 업데이트 그룹에 속할 피어 그룹에 대한 공통 아웃바운드 정책을 생성하면 어떤 피어와 얼마나 많은 피어가 업데이트 그룹의 멤버인지 영향을 줄 수 있습니다. BGP RR에서 다수의 BGP RR 클라이언트로 전송되는 단일 업데이트는 Cisco IOS XR 라우터의 LPTS(Local Packet Transport Service) 구성 요소에서 삭제할 수 있는 TCP ACK의 버스트를 트리거할 수 있습니다.

### RPL의 복잡성(경로 정책)

BGP에서 사용하는 경로 정책의 복잡성은 BGP 프로세스 성능에 영향을 미칩니다. 수신된 또는 전송된 모든 경로는 구성된 경로 정책과 비교하여 평가해야 합니다. 정책이 너무 길면 이 작업에 많은 CPU 사이클을 사용해야 합니다. 정규식을 포함하는 경로 정책은 처리 시 특히 중요합니다. 정규식을 사용하면 경로 정책을 적은 수의 행으로 표현할 수 있지만, 정규식을 사용하지 않는 해당 경로 정책보다 처리하는 동안 더 많은 CPU 주기가 필요합니다.

### 업데이트 빈도

업데이트 빈도는 BGP 규모에 중요한 영향을 미칩니다. 업데이트 횟수는 예측하기 어려운 경우가 많습니다. BGP(라우팅) 업데이트 전송 사이의 최소 간격을 설정하는 "`advertisement-interval`" 명령을 사용하여 업데이트 빈도에 영향을 줄 수 있습니다. iBGP 피어에 대한 기본값은 0초이고 eBGP 피어에 대한 기본값은 30초입니다.

### TCP MSS 및 인터페이스/경로 MTU

업데이트를 여러 TCP 세그먼트로 분할하면 대규모 및 업데이트 빈도가 높은 환경에서 TCP 프로세스 리소스에 큰 부담을 줄 수 있습니다. 경로 MTU가 크고 TCP MSS가 클수록 BGP 및 TCP 성능에 더 적합합니다.

### Dual-RP 라우터의 NSR

NSR은 이중화를 위한 훌륭한 기능이지만 BGP 성능에 영향을 미칩니다. Cisco IOS XR 라우터에서 두 RP는 인그레스 라인 카드의 NPU에서 직접 모든 BGP 업데이트를 동시에 수신합니다. 따라서 활성 RP가 대기 RP에 업데이트를 복제하는 데 시간을 낭비할 필요가 없습니다. 그러나 Active RP에 의해 생성된 모든 업데이트는 Standby RP로 그리고 Standby RP에서 BGP 피어로 전송되어야 합니다.

그러면 대기 RP가 시퀀스 및 승인 번호에서 항상 최신 상태로 유지되지만 전체 BGP 성능에 영향을 미칩니다. 따라서 BGP RR은 단일 RP 라우터로 사용하는 것이 좋습니다.

### 느린 피어

느린 피어는 모든 피어가 인식할 때까지 BGP 프로세스가 업데이트를 메모리에 유지해야 하므로 업데이트 그룹의 모든 멤버에 대한 업데이트 속도를 늦출 수 있습니다. 일부 피어(예: 네트워크의 레거시 부분에 있는 라우터)가 훨씬 느리다는 것을 알고 있는 경우 이를 업데이트 그룹으로 미리 구분합니다. 기본적으로 Cisco IOS XR은 syslog 메시지를 통해 느린 피어를 보고합니다. 다른 사람과 업데이트 그룹을 공유하지 않는 고정 저속 피어를 만들거나 글로벌 또는 인접 디바이스별 컨피그레이션 모드에서 BGPslow-peer 컨피그레이션 명령을 사용하여 동적 저속 피어 동작을 세부적으로 조정할 수 있습니다. 이에 대한 자세한 내용은 Cisco xrdocs.io 포털의 [IOS XR에서 Troubleshoot Slow BGP Convergence Due to Suboptimal Route Policies](#)(Suboptimal 경로 정책으로 인한 느린 BGP 컨버전스 문제 해결)를 참조하십시오.

### Nexthop trigger-delay

짧은 시간 간격으로 여러 BGP next-hop이 변경되고 경로 수가 많은 AF(주소군)에 중요한 nexthop 트리거 지연 값 0이 구성된 경우 모든 next-hop 변경 이벤트에서 AF의 전체 워크가 실행되어야 합니다. 이러한 AF를 반복하면 임계 nexthop 트리거 지연 값이 낮은 주소군의 컨버전스 시간이 증가합니다. "show bgp all nexthops" 명령을 실행하여 next-hop trigger-delay 값을 볼 수 있습니다.

### 검증된 다차원 BGP RR 스케일의 예

다차원 스케일 결과는, 특히 제어 평면 피처에 있어서, 특정 테스트 환경에 크게 의존한다. 일부 매개변수가 변경되면 테스트 결과가 크게 달라질 수 있습니다.

매개변수	가치	가치
플랫폼	XRv9k 어플라이언스 (UCS M5 기반)	ASR9902
IOS XR 릴리스	7.5.2 + Cisco 버그 ID CSCwf09600용 <a href="#">umbrella SMU</a>  (이 umbrella SMU의 구성 요소는 Cisco IOS XR 릴리스 7.9.2 이상에 통합되어 있음)	7.11.2
피어	VPNv4 eBGP: 2500	VPNv4 iBGP: 2000개

	VPNv4 iBGP: 1700	
BGP 경로	세션당: 200 합계: 40만 경로당 경로: 1	세션당: 750 VPNv4: 1.36M VPNv6: 150k IPv4: 950k IPv6: 200k 합계: ~260만 경로당 경로: 1
IGP 경로	10k(ISIS)	10k(ISIS)
BGP 업데이트 그룹	1	1
BGP 타이머	기본값	기본값
LPTS BGP 알려진 폴리서 속도	50,000	25,000
tcp num-thread 구성	16 16	16 16
BGP 송신 버퍼 크기	기본값	기본값
KPI(핵심 성과 지표) 요약	<ul style="list-style-type: none"> <li>• 입력 및 출력 패킷 속도가 가장 높은 테스트 사례:             <ul style="list-style-type: none"> <li>◦ 입력: 49.4kpps</li> <li>◦ 출력: 95kpps</li> <li>◦ ==&gt; LPTS 삭제 (50kpps의 폴리서)</li> <li>◦ ==&gt; NetIO 클라이언트의 삭제 없음</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• 입력 패킷 속도가 가장 높은 테스트 사례:             <ul style="list-style-type: none"> <li>◦ 입력: 16030 pkts/s</li> <li>◦ 출력: 31pkt/s</li> <li>◦ ==&gt; LPTS 및 NetIO 클라이언트에 드롭이 없습니다.</li> <li>◦ ==&gt; 최대 XIPC 큐 크</li> </ul> </li> </ul>

	<ul style="list-style-type: none"> <li>◦ ==&gt; Max XIPC Queue size (BGP)(최대 XIPC 큐 크기(BGP)): 1362</li> <li>◦ ==&gt; 최대 XIPC 큐 크기(TCP): 1248</li> </ul>	<ul style="list-style-type: none"> <li>기(BGP): 378</li> <li>◦ ==&gt; 최대 XIPC 큐 크기(TCP): 1021</li> <li>• 출력 패킷 속도가 가장 높은 테스트 사례: <ul style="list-style-type: none"> <li>◦ 입력: 12172 pkts/s</li> <li>◦ 출력: 23465 pkts/s</li> <li>◦ ==&gt; LPTS 및 NetIO 클라이언트에 드롭이 없습니다.</li> <li>◦ ==&gt; 최대 XIPC 큐 크기(BGP): 109</li> <li>◦ ==&gt; 최대 XIPC 큐 크기(TCP): 1518</li> </ul> </li> </ul>
--	---	---

설계 고려 사항

네트워크에서 BGP RR 배치에 대한 2가지 접근 방식이 있습니다.

- 중앙/플랫 BGP RR 설계.
- 분산/계층적 BGP RR 설계.

중앙 집중식/플랫 설계에서 네트워크의 모든 BGP RR 클라이언트는 동일한 정보를 보유하는 BGP RR 디바이스 집합(일반적으로 쌍)과 BGP 피어링을 설정합니다. 이 접근 방식은 간단하게 구현할 수 있으며 소규모에서 중간 규모의 네트워크에서도 원활하게 작동합니다. BGP 테이블의 모든 변경 사항은 모든 BGP RR 클라이언트에 신속하게 전파됩니다. BGP RR 클라이언트의 수가 증가하면 BGP RR 디바이스의 TCP 연결 수가 성능에 영향을 미칠 정도로 증가하면 설계에서 스케일 제한에 도달할 수 있습니다.

분산형/계층적 설계에서 네트워크는 여러 지역으로 분할됩니다. 영역의 모든 라우터는 동일한 정보를 보유하는 BGP RR 디바이스 집합(일반적으로 쌍)과 BGP 피어링을 설정합니다. 이러한 BGP RR 디바이스는 다른 BGP RR 디바이스 세트(일반적으로 쌍)에 대한 BGP RR 클라이언트 역할을 합니다. 이 설계 방식을 사용하면 모든 단일 BGP RR의 TCP 연결 수를 특정 제한 이하로 유지하면서 손실

게 네트워크를 확장할 수 있습니다.

또 다른 설계 고려 사항은 BGP 업데이트의 수신자 범위를 조정하는 것입니다. BGP RR 클라이언트 간의 VRF 분포에 따라 RT Constrained Route Distribution을 고려할 필요가 있습니다. 모든 BGP RR 클라이언트에 동일한 VRF의 인터페이스가 있는 경우 RT Constrained Route Distribution은 많은 이점을 제공하지 않습니다. 그러나 VRF가 모든 BGP RR 클라이언트에 드문드문 배포되는 경우 RT Constrained Route Distribution을 사용하면 BGP RR의 bgp 프로세스에 대한 로드가 크게 줄어듭니다.

### BGP KPI(핵심 성과 지표) 모니터링

BGP RR의 KPI(핵심 성능 지표)를 모니터링하는 것은 올바른 네트워크 작동을 보장하는 데 중요합니다.

네트워크 토폴로지의 중대한 변경(예: 주요 DWDM 링크 플랩)은 BGP RR로 및/또는 BGP RR로부터 과도한 트래픽을 생성하는 라우팅 업데이트를 트리거할 수 있습니다. BGP RR에 도달하는 중요 트래픽은 일반적으로 다음과 같습니다.

- BGP 피어로부터의 업데이트.
- BGP 피어에서 생성된 TCP ACK, BGP RR에서 전송한 업데이트에 대한 응답 및 그 반대

이 섹션에서는 일반적인 BGP RR에서 모니터링해야 하는 KPI와 두 가지 주요 BGP 트래픽 유형 중 어떤 유형이 높은 컨트롤 플레인 트래픽 속도를 유발하는지 확인하는 방법에 대해 설명합니다.

라우터 내의 BGP 패킷 경로는 다음과 같이 표시할 수 있습니다.

<b>편트</b>
이더넷 컨트롤러 -(패킷)-> 데이터 경로 전달자 -(패킷)-> LPTS -(패킷)-> SPP -(패킷) -> NetIO -(패킷)-> TCP -(메시지)-> BGP
<b>삽입</b>
BGP -(메시지)-> TCP -(패킷)-> NetIO -(패킷)-> SPP -(패킷) -> Datapath 전달자 -(패킷)-> 이더넷 컨트롤러

KPI는 다음과 같이 나눌 수 있습니다.

필수 사항:

- 데이터 경로 전달자
- LPTS(하드웨어 편트 폴리스터 설정, 카운터 수락 및 카운터 삭제)
- SPP
- NetIo

- IPC 큐(NetIO <==> TCP <==> BGP)
- BGP InQ/OutQ 크기

선택 사항:

- CPU 사용률
- 메모리 사용률
- TCP 통계
- BGP 프로세스 성능
- BGP 통합

#### 데이터 경로 전달자 모니터링

XRv9000에서 데이터 경로 전달자는 DPA(Data Plane Agent)인 반면, ASR9000 플랫폼에서는 NP(Network Processor)입니다.

#### XRv9000 데이터 플레인 에이전트 모니터링(DPA)

DPA의 로드 및 통계를 확인할 수 있는 유용한 명령은 다음과 같습니다.

```
show controllers dpa statistics global
```

이 명령은 모든 0이 아닌 카운터를 보여주며, 이는 네트워크 인터페이스에서 RP CPU로 보내진 패킷의 유형 및 수, RP CPU에서 네트워크 인터페이스로 주입된 패킷 및 삭제된 패킷 수에 대한 통찰력을 제공합니다.

<#root>

```
RP/0/RP0/CPU0:xrv9k-01#
```

```
show controllers dpa statistics global
```

```
Index Debug Count ----- 350 TBP
```

#### ASR9000 NP(Network Processor) 모니터링

시스템에서 각 NP의 로드 및 통계를 확인하는 데 유용한 명령은 다음과 같습니다.

```
show controllers np load all
```

```
show controllers np counters all
```

ASR9000의 NP에는 처리 및 삭제된 패킷의 수, 속도 및 유형을 보여주는 다양한 카운터 집합이 있습니다.

<#root>

```
RP/0/RSP0/CPU0:ASR9k-B#
```

```
show controllers np load all
```

```
Node: 0/0/CPU0: ----- Load Packet Rate NP0:
```

<#root>

```
RP/0/RSP0/CPU0:ASR9k-B#
```

```
show controllers np counters all
```

```
Node: 0/0/CPU0: ----- Show global stats cou
```

### LPTS 모니터링

표준 BGP RR이 전달 경로에 없으므로 네트워크 인터페이스에서 수신된 모든 패킷이 제어 평면으로 보내집니다. BGP RR의 data-path 요소는 패킷이 컨트롤 플레인으로 전송되기 전에 간단한 몇 가지 작업을 수행합니다. data-path 요소가 정체 지점이 될 가능성이 낮기 때문에 라인 카드에서 모니터링이 필요한 유일한 요소는 LPTS 통계입니다.

XRv9k의 경우 하드웨어 통계가 vPP에 매핑됩니다

명령을 사용합니다:

```
show lpts pifib hardware police location <location> | inc "Node|flow_type|BGP"
```

예:

```
RP/0/RP0/CPU0:rxv9k-01#sh lpts pifib hardware police location 0/0/CPU0 | i "Node|flow_type|BGP" Node 0/0/CPU0: flow_type priority sw_police_id hv
```

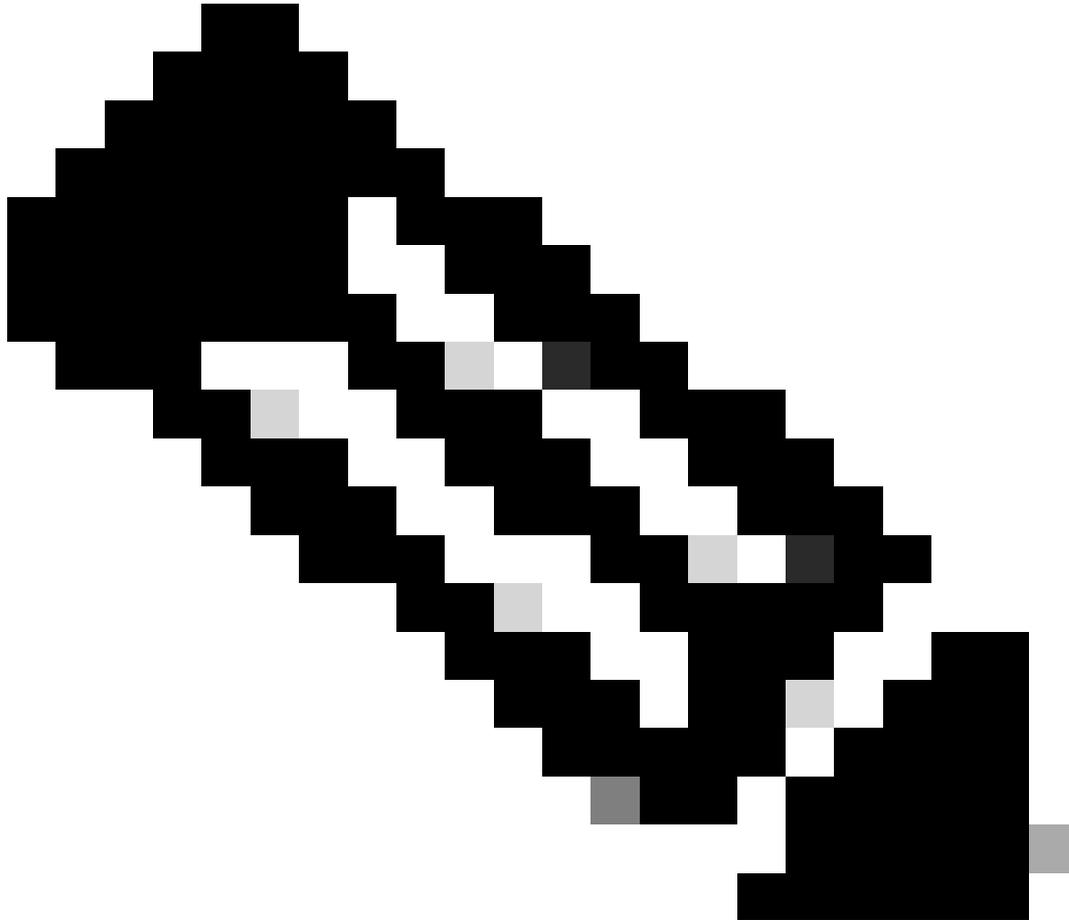
### 살펴볼 내용:

BGP로 알려진 플로우 유형에 대한 AggDrops가 크게 증가한 것이 관찰되면, 이와 같은 대규모 제어 평면 변경을 트리거한 네트워크 토폴로지 변경 사항을 찾기 시작합니다.

텔레메트리 데이터 경로:

Cisco-IOS-XR-lpts-pre-ifib-oper:lpts-pifib

---



참고: LPTS 통계 카운터를 지울 수 있습니다. 모니터링 시스템에서 이러한 가능성을 고려해야 합니다.

---

모니터 SPP

SPP는 내부 패브릭을 통해 NP 또는 DPA에서 펀트된 패킷을 수신하는 경로 프로세서 또는 라인 카드 CPU의 첫 번째 엔티티이며, NP

또는 DPA에 주입하기 위해 패브릭으로 전달되기 전에 소프트웨어 패킷 처리의 마지막 포인트입니다.

SPP 모니터링에 대한 관련 명령:

```
show spp node-counters
```

```
show spp client
```

이 **show spp node-counters** 명령은 펀트된/삽입된 패킷의 비율을 보여주며 읽기 쉽고 이해하기 쉽습니다. BGP 세션의 경우 관련 카운터가 활성 RP **client/punt** 아래 및 **client/inject** 위에 있습니다.

의 출력 **show spp client** 은 더욱 풍부하며, 하이 워터마크는 물론 클라이언트에 대해 큐에 넣거나 삭제된 패킷의 수에 대한 더 자세한 정보를 제공합니다.

<#root>

```
RP/0/RP0/CPU0:xrv9k-01#
```

```
show spp node-counters
```

```
0/RP0/CPU0:
```

```
socket/rx Punted packets: 595305 Punt bulk reads: 6 Punt non-bulk reads: 595293 Management packets: 74
client/inject Injected from client: 140534413 Non-bulk injects: 140534413 -----
----- 0/0/CPU0: <. . .>
```

<#root>

```
RP/0/RP0/CPU0:xrv9k-01#
```

```
show spp client
```

```
Sat Apr 20 17:11:40.725 UTC 0/RP0/CPU0: Clients ===== <. . .> netio, JID 254 (pid 4591) -----
```

NetIO 모니터링

LPTS 폴리서는 해당 폴리서가 수락하거나 삭제한 패킷의 수만 표시하지만, NetIO 레벨에서 RP CPU로 펀트된 패킷의 비율을 확인할 수 있습니다. 일반적인 BGP RR에서는 수신된 패킷의 대부분이 BGP 패킷이므로 전체 NetIO 속도는 수신된 BGP 패킷의 비율을 매우 근접하게 나타냅니다.

<#root>

**Command:**

show netio rates

**예:**

<#root>

RP/0/RP0/CPU0:xrv9k-01#

show netio rates

Netio packet rate for node 0/RP0/CPU0 ----- Current rate (updated 0 seconds)

**살펴볼 내용:**

- NetIO 속도가 크게 빨라질 것으로 예상될 경우, 이와 같은 대규모 컨트롤 플레인 변경을 유발한 네트워크 토폴로지 변경 사항을 찾기 시작합니다.

**텔레메트리 데이터 경로:**

- 텔레메트리에서는 속도가 아니라 카운터 값을 스트리밍해야 하므로 적용할 수 없습니다. BGP-known LPTS policer accept 카운터를 텔레메트리 컬렉터에서 사용하여 알려진 피어에서 수신한 BGP 패킷의 평균 속도를 근사화할 수 있습니다.

**XIPC 대기열 모니터링**

NetIO가 LPTS에서 수신한 punt 경로 패킷은 TCP 및 BGP로 전달됩니다. 이러한 대기열을 모니터링하는 것이 중요합니다.

1. NetIO가 패킷을 TCP로 전달하는 TCP 높은 우선순위 큐
2. BGP 제어 큐
3. BGP 데이터 큐

삽입 경로에서 패킷은 TCP에 의해 생성되고 NetIO로 전달됩니다. 이러한 대기열을 모니터링하는 것이 중요합니다.

- OutputL XIPC 큐

**명령:**

show netio clients show processes bgp | i "Job Id" show xipcq jid <bgp\_job\_id> show xipcq jid <bgp\_job\_id> queue-id <n>

예:

NetIO에서 TCP로, NetIO의 관점에서 보기:

```
RP/0/RP0/CPU0:xrv9k-01#show netio clients < . . > Input Punt XIPC InputQ XIPC PuntQ ClientID Drop/Total Drop/Total Cur/High/Max Cur/High/Max
```

TCP에서 NetIO로, NetIO의 관점에서 보기:

```
RP/0/RP0/CPU0:xrv9k-01#show netio clients < . . > XIPC queues Dropped/Queued Cur/High/Max ----- Outp
```

NetIO에서 TCP로, TCP 프로세스 관점에서 보기:

<#root>

```
RP/0/RP0/CPU0:xrv9k-01#
```

```
show processes tcp
```

```
| i "Job Id"
Job Id: 430
```

```
RP/0/RP0/CPU0:xrv9k-01#
```

```
show xipcq jid
```

```
430 Mon Apr 17 16:16:11.315 CEST Id Name Size Cur Size Produced Dropped HWM -----
```

TCP-BGP:

<#root>

```
RP/0/RP0/CPU0:xrv9k-01#
```

```
show processes bgp
```

```
| i "Job Id" Job Id: 1078 RP/0/RP0/CPU0:xrv9k-01#
```

```
show xipcq jid
```

```
1078 Mon Apr 17 16:09:33.046 CEST Id Name Size Cur Size Produced Dropped HWM -----
```

BGP 데이터 큐:

<#root>

```
RP/0/RP0/CPU0:xrv9k-01#
```

```
show xipcq jid
```

```
1078
```

```
queue-id 1
```

```
XIPC_xipcq_12_0_9854_6506_inst_1_data_toapp
```

```
:
```

```
Magic: 12344321 Version: 0 SHM Size: 192392 Owner PID: 9854 Owner JID: 1078 Queue ID: 1 Owner MQ handl
```

BGP 제어 큐:

```
<#root>
```

```
RP/0/RP0/CPU0:xrv9k-01#
```

```
show xipcq jid
```

```
1078
```

```
queue-id
```

```
2 XIPC_xipcq_12_0_9854_6506_inst_1_ctrl_toapp: Magic: 12344321 Version: 0 SHM Size: 480392 Owner PID: 9854
```

살펴볼 내용:

- 관련 대기열에 삭제되지 않아야 합니다.
- XIPC 대기열 통계 HWM(high watermark)이 대기열 크기의 50%를 초과해서는 안 됩니다.

하이 워터마크 값 진화를 더 잘 추적하려면 읽을 때마다 하이 워터마크 값을 지워야 합니다. 이 경우 HWM 카운터만 지워지는 것이 아니라 모든 대기열 통계도 지워집니다. XIPC 대기열 통계를 지우기 위한 명령의 형식은 다음과 같습니다. `clear xipcq statistics queue-name <queue_name>`

대기열 이름에 종종 프로세스 ID(PID)가 포함되므로 프로세스 재시작 후 대기열 이름이 변경됩니다.

관련 대기열 통계를 지우기 위한 명령의 몇 가지 예는 다음과 같습니다.

```
clear xipcq statistics queue-name XIPC_tcp_i0
clear xipcq statistics queue-name XIPC_tcp_i1
clear xipcq statistics queue-name XIPC_xipcq_12_0_9854_6506_inst_1_data_toapp
clear xipcq statistics queue-name XIPC_xipcq_12_0_9854_6506_inst_1_ctrl_toapp
```

텔레메트리 경로:

- XIPC에 대한 텔레메트리 센서 경로가 없습니다.

#### BGP 입력 및 출력 대기열 모니터링

BGP는 모든 BGP 피어에 대한 입력 및 출력 대기열을 유지 관리합니다. 데이터는 TCP가 BGP에 전달했지만 BGP가 아직 처리하지 않은 InQ에 배치됩니다. BGP가 데이터를 패킷으로 분할하고 전송하기 위해 TCP에서 대기하는 동안 데이터는 OutQ에 배치됩니다.

BGP InQ/OutQ의 즉각적인 크기는 BGP 프로세스의 사용 빈도를 나타냅니다.

명령을 사용합니다:

```
show bgp <AFI> <SAFI> summary
```

예:

```
RP/0/RP0/CPU0:rxv9k-01#show bgp all all summary Address Family: VPNv4 Unicast ----- BGP router identifier 192.168.0.1, local A
```

살펴볼 내용:

- 네트워크가 안정적이면 InQ/OutQ의 크기는 0이어야 합니다. 업데이트가 교환되면 빠르게 변경됩니다.
- InQ/OutQ 크기는 시간이 지남에 따라 단조롭게 증가해서는 안 됩니다.

텔레메트리 경로:

- Cisco-IOS-XR-ipv4-bgp-oper:bgp

#### BGP 메시지 속도 모니터링

네트워크 토폴로지가 불안정한 경우 일부 BGP 인접 디바이스에서 지속적으로 업데이트 또는 철회를 전송할 수 있습니다. 그런 다음 BGP RR은 이러한 라우팅 테이블 변경을 모든 RR 클라이언트에 수천 번 복제해야 합니다. 따라서 인접 디바이스로부터 수신된 메시지 속도를 모니터링하여 불안정의 원인을 추적하는 것이 중요합니다.

명령을 사용합니다:

```
show bgp <AFI> <SAFI> summary
```

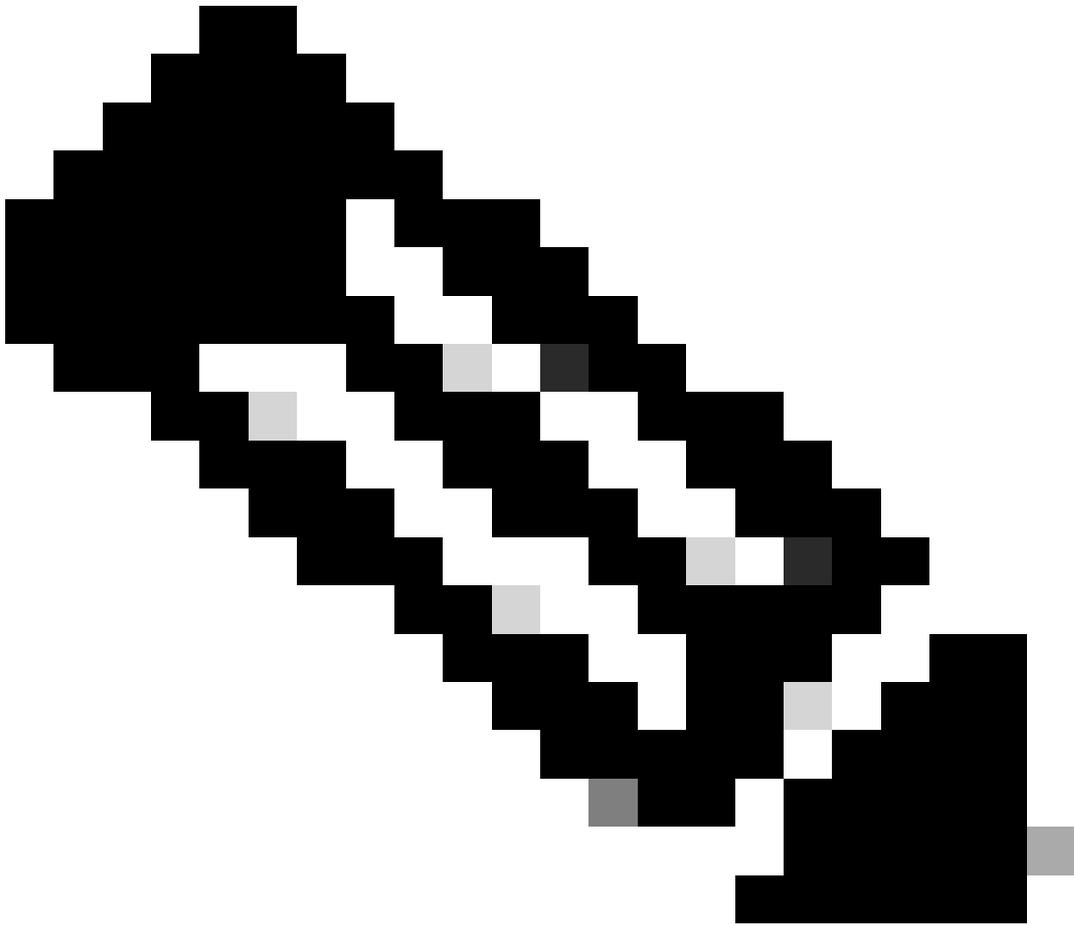
예:

```
RP/0/RP0/CPU0:xrv9k-01#show bgp all all summary Address Family: VPNv4 Unicast ----- BGP router identifier 192.168.0.1, local A
```

RR 클라이언트 대기열의 MsgSent 양은 거의 같지만 일부 네이버의 MsgRcvd 수가 다른 네이버보다 높을 수 있습니다. 메시지 속도를 평가하려면 이 명령의 여러 스냅샷을 캡처해야 합니다.

문제가 되는 피어를 식별한 후에는 **show bgp neighbor <neighbor> detail** 및 **show bgp neighbor <neighbor> performance-statistics** 또는 와 같은 다른 명령 **show bgp recent-prefixes** 을 통해 어떤 접두사가 플래핑되는지 그리고 항상 같은 접두사인지 아니면 다른 접두사인지 이해할 수 있습니다.

---



---

**참고:** MsgRcvd 및 MsgSent 카운터는 네이버별로 구성되지만 주소군별로 구성되지는 않습니다. 이와 같은 명령을 실행할 때 show bgp all all summary 는 여러 주소 패밀리에 대한 섹션에서 네이버당 동일한 카운터를 볼 수 있습니다. 해당 주소 패밀리에 대해 해당 네이버에서 주고받은/해당 네이버로 보낸 메시지 수를 나타내는 것이 아니라 주소 패밀리에 걸쳐 보냅니다.

---

## CPU 사용률 모니터링

모든 라우터에서 CPU 사용률을 모니터링해야 하지만, 컨트롤 플레인 전용 CPU 코어 수가 많은 라우터에서는 일부 판독이 직관적이지 않을 수 있습니다. XRv9k Appliance와 같이 RP(Routing Processor) 전용 CPU 코어 수가 많은 BGP RR에서는 활성 스레드가 서로 다른 CPU 코어에서 실행되지만, 유휴 상태의 CPU 코어는 많습니다. 그 결과 일부 CPU 코어는 매우 사용량이 많을 수 있지만 모든 CPU 코어에서 계산된 전체 CPU 사용률은 보통 수준을 유지합니다.

따라서 CLI를 통해 CPU 코어 사용률을 적절하게 모니터링하려면 이 명령을 **show processes cpu thread** 사용합니다.

## TCP 통계 모니터링

Cisco IOS®는 모든 TCP 세션에 대한 자세한 통계를 유지 관리합니다. CLI 명령은 모든 기존 TCP 세션의 목록을 **show tcp brief** 표시합니다. 이 요약 출력에서는 모든 TCP 세션에 대해 다음 정보를 볼 수 있습니다.

- **PCB:** 고유한 TCP 세션 식별자.
- **VRF-ID:** 세션이 있는 VRF의 ID입니다.
  - 해당 VRF 이름을 보려면 다음 명령을 실행합니다.
  - `show cef vrf all summary | utility egrep "^VRF:|Vrfid" | utility egrep -B1 <VRF-ID>`
- **Recv-Q:** 수신 Q의 즉각적인 크기입니다. 수신 대기열은 NetIO에서 받은 패킷을 보관합니다. tcp 프로세스는 패킷에서 데이터를 추출하여 해당 애플리케이션에 전송합니다.
- **Send-Q:** 전송 큐의 즉각적인 크기입니다. 전송 큐는 응용 프로그램에서 받은 데이터를 보관합니다. tcp 프로세스는 데이터를 TCP 세그먼트(협상된 최대 세그먼트 크기 - TCP MSS에 의해 지정됨)로 분할하고, 모든 세그먼트를 해당 주소군(IPv4 또는 IPv6)의 레이어 3 헤더로 캡슐화한 다음 패킷을 NetIO로 전송합니다.
- **Local Address(로컬 주소):** TCP 소켓과 연결된 로컬 IPv4 또는 IPv6 주소입니다. LISTEN 상태의 TCP 세션은 일반적으로 "any" IP 주소에 바인딩됩니다. 이는 각각 IPv4 또는 IPv6의 경우 "0.0.0.0" 또는 "::"으로 표시됩니다.
- **외부 주소:** TCP 소켓과 연결된 원격 IPv4 또는 IPv6 주소입니다. LISTEN 상태의 TCP 세션은 일반적으로 "any" IP 주소에 바인딩됩니다. 이는 각각 IPv4 또는 IPv6의 경우 "0.0.0.0" 또는 "::"으로 표시됩니다.

- **상태:** TCP 세션 상태입니다. 가능한 TCP 세션 상태는 LISTEN, SYNSENT, SYNRCVD, ESTAB, LASTACK, CLOSING, CLOSEWAIT, FINWAIT1, FINWAIT2, TIMEWAIT, CLOSED입니다.

잘 알려진 BGP 포트 번호가 179이므로 표시되는 TCP 세션을 BGP 애플리케이션과 연결된 세션으로 제한할 수 있습니다.

예:

```
RP/0/RSP0/CPU0:ASR9k-B#show tcp brief | include "PCB|:179 " PCB VRF-ID Recv-Q Send-Q Local Address Foreign Address State 0x00007ff7d403bd
```

표시된 PCB 값을 사용하여 특정 TCP 세션에 대한 통계를 확인할 수 있습니다. TCP 프로세스 통계에 대한 정보를 제공하는 CLI 명령:

글로벌:

```
show tcp statistics clients location <active_RP>
```

```
show tcp statistics summary location <active_RP>
```

PCB당:

```
show tcp brief | i ":179"
```

```
show tcp detail pcb <pcb> location 0/RP0/CPU0
```

```
show tcp statistics pcb <pcb> location <active_RP>
```

전역 TCP statistics 명령은 TCP 세션의 전체 상태를 표시합니다. 데이터 패킷 통계(in/out)와는 별도로 체크섬 오류가 있는 패킷, 형식이 잘못된 패킷, 인증 오류로 인해 삭제된 패킷, 순서가 잘못된 패킷, 윈도우 이후 데이터가 있는 패킷 등 TCP 피어의 동작을 나타내는 패킷이 있는지 여부를 확인할 수 있습니다.

PCB당 명령에서 MSS, 최대 왕복 시간 등과 같은 TCP 세션의 중요한 매개변수를 볼 수 있습니다.

명령 출력의 관련 카운터는 show tcp detail pcb 다음과 같습니다.

- **Retrans Timer Starts:** 재전송 타이머가 시작된 횟수를 나타냅니다.
- **Retrans Timer Wakeups:** 재전송 타이머가 몇 번 초과되어 TCP 세그먼트의 재전송이 발생했는지 나타냅니다.
- **현재 전송 큐 크기(바이트):** 피어에서 승인되지 않은 바이트.
- **현재 수신 큐 크기(바이트/패킷):** BGP에서 아직 읽을 바이트/패킷입니다.

- 순서가 잘못된 바이트: TCP 수신 창의 구멍으로 인해 저장 대기열에서 대기된 바이트

<#root>

RP/0/RSP0/CPU0:ASR9k-B#

show tcp detail pcb 0x4a4400e4

===== Connection state is ESTAB, I/O status: 0

Current send queue size in bytes: 0 (max 16384)

Current receive queue size in bytes: 0 (max 65535)

mis-ordered: 0 bytes

Current receive queue size in packets: 0 (max 60)

Timer Starts Wakeups Next(msec)

Retrans 2795 0 0

SendWnd 1341 0 0 TimeWait 0 0 0 AckHold 274 2 0 KeepAlive 333 1 299983 PmtuAger 0 0 0 GiveUp 0 0 0 Thro  
SRTT: 162 ms, RTTO: 415 ms, RTV: 253 ms, KRTT: 0 ms  
minRTT: 0 ms, maxRTT: 247 ms ACK hold time: 200 ms, Keepalive time: 300 sec, SYN waittime: 30 sec Giveu

메모리 사용률 모니터링

BGP 경로 테이블은 BGP 프로세스 힙 메모리에 저장됩니다. 라우팅 테이블은 RIB 프로세스 힙 메모리에 저장됩니다.

힙 메모리 모니터링에 유용한 명령:

show memory summary

show memory summary detail

show memory-top-consumers

show memory heap summary all

텔레메트리 센서 경로:

FIB는 전달 항목을 공유 메모리 공간에 저장합니다.

공유 메모리 모니터링에 유용한 명령:

```
show memory summary
```

```
show memory summary detail
```

```
show shmwin summary
```

BGP 프로세스 성능 모니터링

BGP 프로세스 성능에 대한 내부 데이터를 제공하는 유용한 명령:

```
show bgp process performance-statistics
```

```
show bgp process performance-statistics detail
```

BGP 통합 모니터링

또 다른 유용한 명령은 BGP 통합의 전체 상태를 보여 주는 명령입니다. `show bgp convergence`

네트워크가 안정되면 다음과 같은 상황이 발생합니다.

```
RP/0/RP0/CPU0:ASR9k-B#show bgp convergence Mon Dec 18 13:55:47.976 UTC Converged. All received routes in RIB, all neighbors updated. All nei
```

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.