

IOS-XR에서 최적화되지 않은 경로 정책으로 인한 느린 BGP 통합 문제 해결

목차

[소개](#)

[배경](#)

[문제](#)

[솔루션](#)

[확인](#)

[결론](#)

소개

이 문서에서는 최적화되지 않은 경로 정책으로 인해 발생하는 Cisco IOS® XR 라우터의 느린 BGP 통합 문제를 진단하는 방법에 대해 설명합니다.

배경

BGP 컨버전스 시간은 여러 요인으로 구성됩니다. 그 중 하나는 구성된 경로 정책에 의해 인그레스 또는 이그레스 BGP 업데이트를 처리하는 시간입니다. 특정 작업을 수행하기 위해 경로 정책을 작성하는 방법에는 여러 가지가 있습니다. 최적의 방법은 BGP 통합을 개선하고 잠재적인 트래픽 삭제를 최소화하며 임시 라우팅 루프를 방지하는 데 도움이 됩니다. Cisco IOS® XR에는 처리 시간을 예측하기 위해 특정 경로 정책에서 소요된 시간을 측정하는 프로파일링 툴이 포함되어 있습니다.

문제

느린 BGP 컨버전스는 최적 상태가 아닌 방식으로 작성된 경로 정책의 결과이기도 합니다.

솔루션

특정 연결 지점에서 경로 정책의 각 명령문에서 소요되는 시간을 측정하기 위해 성능에 영향을 주지 않고 사용할 수 있는 경로 정책에 대한 정책 프로파일링 툴이 있습니다. 이 특정 연결 지점에서 경로 정책의 실행 시간을 확인할 수 있습니다. 기본적으로 프로파일링은 종합 경로 정책 통계에 대해서만 활성화됩니다.

```
router bgp 65000
neighbor 10.0.54.6
remote-as 65000
update-source Loopback0
address-family ipv4 unicast
route-policy INGRESS-ROUTE-POLICY in
```

```
!  
neighbor 10.0.54.11  
remote-as 65001  
ebgp-multihop 255  
update-source Loopback0  
address-family ipv4 unicast  
route-policy EGRESS-ROUTE-POLICY out
```

<#root>

```
RP/0/RSP1/CPU0:XR1#
```

```
show pc1 protocol bgp speaker-0 neighbor-in-dflt default-IPv4-Uni-10.0.54.6 policy profile
```

```
Policy profiling data  
Policy : INGRESS-ROUTE-POLICY  
Pass : 1440233  
Drop : 0  
# of executions : 1440233  
Total execution time : 57095msec <=====
```

```
RP/0/RSP1/CPU0:XR1#
```

```
show bgp ipv4 unicast neighbors 10.0.54.11 | i Update group
```

```
Update group: 0.3 Filter-group: 0.5 No Refresh request being processed  
RP/0/RSP1/CPU0:XR1#
```

```
RP/0/RSP1/CPU0:XR1#
```

```
show pc1 protocol bgp speaker-0 neighbor-out-dflt default-IPv4-Uni-UpdGrp-0.3-Out policy profile
```

```
Policy profiling data  
Policy : EGRESS-ROUTE-POLICY  
Pass : 726751  
Drop : 0  
# of executions : 726751  
Total execution time : 108099msec <=====
```

```
RP/0/RSP1/CPU0:XR1#
```

INGRESS-ROUTE-POLICY 및 EGRESS-ROUTE-POLICY를 처리하는 데 소요된 누적 시간을 확인 할 수 있습니다.

프로파일링은 모든 연결 지점에 대한 인그레스 또는 이그레스 경로 정책에 적용할 수 있습니다.

<#root>

```
RP/0/RSP1/CPU0:XR1#
```

```
show pc1 protocol bgp speaker-0 ?
```

debug-policy	Attachpoint name
permetnet	Attachpoint name

import	Attachpoint name
export	Attachpoint name
interafi-import	Attachpoint name
source-rt	Attachpoint name
interafi-export	Attachpoint name
retain-rt	Attachpoint name
addpath	Attachpoint name
neighbor-in-dflt	Attachpoint name
neighbor-in-vrf	Attachpoint name
neighbor-out-dflt	Attachpoint name
neighbor-out-vrf	Attachpoint name
orf-dflt	Attachpoint name
orf-vrf	Attachpoint name
dampening-dflt	Attachpoint name
dampening-vrf	Attachpoint name
default-originate-dflt	Attachpoint name
default-originate-vrf	Attachpoint name
clear-policy	Attachpoint name
show-policy-node0_RSP1_CPU0	Attachpoint name
aggregation-dflt	Attachpoint name
aggregation-vrf	Attachpoint name
nexthop	Attachpoint name
allocate-label	Attachpoint name
label-mode	Attachpoint name
l2vpn-import	Attachpoint name
l2vpn-export	Attachpoint name
redistribution-dflt	Attachpoint name
redistribution-vrf	Attachpoint name
rib-install-dflt	Attachpoint name
rib-install-vrf	Attachpoint name
network-dflt	Attachpoint name
network-vrf	Attachpoint name
redistribution-dflt	Attachpoint name
redistribution-vrf	Attachpoint name
rib-install-dflt	Attachpoint name
rib-install-vrf	Attachpoint name
network-dflt	Attachpoint name
network-vrf	Attachpoint name
l2vpn-export-mp2mp	Attachpoint name
l2vpn-export-vfi	Attachpoint name
l2vpn-export-evi	Attachpoint name
l2vpn-export-mspw	Attachpoint name
l2vpn-export-instance	Attachpoint name
WORD	Attachpoint name

RP/0/RSP1/CPU0:XR1#

필요에 따라 통계를 지을 수 있습니다.

<#root>

RP/0/RSP1/CPU0:XR1#

clear pc1 protocol bgp speaker-0 neighbor-in-dflt default-IPv4-Uni-10.0.54.6 policy profile

RP/0/RSP1/CPU0:XR1#

clear pc1 protocol bgp speaker-0 neighbor-out-dflt default-IPv4-Uni-UpdGrp-0.3-Out policy profile

debug pcl profile detail을 활성화하면 경로 정책 항목당 자세한 통계를 확인할 수 있습니다.

<#root>

RP/0/RSP1/CPU0:XR1#

debug pcl profile detail

이러한 출력은 전체 인터넷 BGP 테이블 스케일이 수신되어 더 전파된 후에 수집되었습니다.

<#root>

RP/0/RSP1/CPU0:XR1#

show pcl protocol bgp speaker-0 neighbor-in-dflt default-IPv4-Uni-10.0.54.6 policy profile

Policy profiling data

Policy : INGRESS-ROUTE-POLICY

Pass : 720100

Drop : 0

of executions : 720100

Total execution time : 222788msec <===== about 3.7 minutes to process ingress updates

Node Id	Num visited	Exec time	Policy engine operation
---------	-------------	-----------	-------------------------

PXL_0_1	720100	221796msec	if as-path aspath-match ... then <=====
---------	--------	------------	---

PXL_0_3	3525	3msec	set local-preference 150
	3525	0msec	

PXL_0_2	716575	225msec	set local-preference 50
	716575	82msec	

RP/0/RSP1/CPU0:XR1#

show pc1 protocol bgp speaker-0 neighbor-out-dflt default-IPv4-Uni-UpdGrp-0.3-Out policy profile

Policy profiling data

Policy : EGRESS-ROUTE-POLICY

Pass : 720105

Drop : 0

of executions : 720105

Total execution time : 221975msec <===== about 3.7 minutes to process egress updates

Node Id	Num visited	Exec time	Policy engine operation
---------	-------------	-----------	-------------------------

PXL_0_1	720105	3005msec	if as-path aspath-match ... then
---------	--------	----------	----------------------------------

PXL_0_5	0	0msec	set med 70
	0	0msec	

PXL_0_2	720105	218008msec	if as-path aspath-match ... then	<=====
---------	--------	------------	----------------------------------	--------

PXL_0_3	25	0msec	set med 80
	25	0msec	

PXL_0_4	720080	145msec	set med 90
	720080	76msec	

RP/0/RSP1/CPU0:XR1#

보다시피 INGRESS-ROUTE-POLICY의 경우 PXL_0_1 줄을, EGRESS-ROUTE-POLICY의 경우 PXL_0_2 줄을 사용하면 특히 시간이 많이 소요되며 컨버전스가 느려집니다.

경로 정책과 상호 연결할 경우 INGRESS-ROUTE-POLICY의 AS-PATH-SET-11 및 EGRESS-ROUTE-POLICY의 AS-PATH-SET-22가 문제를 일으킨다는 것을 알 수 있습니다. 각 AS 경로 집합은 100개의 정규식 라인으로 구성되어 있는데, 이는 가능한 최적화나 regex의 기능을 활용하지 않기 때문에 정책을 작성하는 매우 비효율적인 방법입니다.

```
route-policy INGRESS-ROUTE-POLICY
  if as-path in AS-PATH-SET-11 then
    set local-preference 150
  else
    set local-preference 50
  endif
end-policy
```

```
route-policy EGRESS-ROUTE-POLICY
  if as-path in AS-PATH-SET-21 then
    set med 70
  elseif as-path in AS-PATH-SET-22 then
    set med 80
  else
    set med 90
  endif
end-policy
```

```
as-path-set AS-PATH-SET-11
  ios-regex '^65101 65201_',
  ios-regex '^65102 65202_',
  ios-regex '^65103 65203_',
  ios-regex '^65104 65204_',
  ios-regex '^65105 65205_',
  --- removed 90 similar lines ---
  ios-regex '^65195 65295_',
  ios-regex '^65196 65296_',
  ios-regex '^65197 65297_',
  ios-regex '^65198 65298_',
```

```

ios-regex '^65199 65299_'
end-set

as-path-set AS-PATH-SET-21
ios-regex '^$'
end-set

as-path-set AS-PATH-SET-22
ios-regex '^65169(_65169)*$',
ios-regex '^65392(_65392)*$',
ios-regex '^65133(_65133)*$',
ios-regex '^65231(_65231)*$',
ios-regex '^65161(_65161)*$',
--- removed 90 similar lines ---
ios-regex '^65281(_65281)*$',
ios-regex '^65336(_65336)*$',
ios-regex '^65238(_65238)*$',
ios-regex '^65381(_65381)*$',
ios-regex '^65103(_65103)*$'
end-set

```

정책 성능을 개선하기 위해 정규식 대신 네이티브 as-path 일치 작업을 사용하여 AS 경로 세트의 컨피그레이션을 평가할 수 있습니다. 또는 경로 정책 내에서 정규식을 축소된 방식으로 사용할 수 있으므로 사용되는 정규식 행의 수를 줄일 수 있습니다.

이 표에는 RPL(Route Policy Language)에서 제공하는 AS 경로 일치 기준이 나열되어 있습니다. 네이티브 일치 함수는 정규식 일치 엔진에 비해 더 나은 성능을 제공하는 이진 일치 알고리즘을 사용합니다. 대부분의 일반적인 ios-regex 일치 시나리오는 해당 시나리오(또는 해당 조합)와 함께 작성할 수 있습니다.

명령을 사용합니다	설명
로컬임	라우터(또는 이 자울 시스템 또는 연합 내의 다른 라우터)가 경로를 시작했는지 확인합니다
길이	AS 경로 길이를 기반으로 조건부 검사를 수행합니다
네이버	하나 이상의 정수 계열 값 또는 매개 변수 시퀀스를 기준으로 AS 경로의 헤드에 있는 자동 시스템 번호를 테스트합니다.
출처:	시작부터 AS 시퀀스에 대해 AS 경로를 테스트합니다경로를 시작한 AS 번호로 설정합니다.
통과	지정된 정수 또는 매개 변수가 AS 경로의 아무 곳이나 나타나는지 또는 정수 및 매

	개 변수의 시퀀스가 나타나는지 테스트합니다.
고유 길이	중복을 무시한 채 AS 경로 길이를 기준으로 특정 검사를 수행합니다.

확인

이는 네이티브 일치 기준의 도움으로 작성된 재정렬된 경로 정책입니다. 이는 실질적으로 감소된 처리 시간을 초래한다.

```
route-policy INGRESS-ROUTE-POLICY
  if as-path in AS-PATH-SET-11 then
    set local-preference 150
  else
    set local-preference 50
  endif
end-policy

route-policy EGRESS-ROUTE-POLICY
  if as-path is-local then
    set med 70
  elseif as-path in AS-PATH-SET-22 and as-path unique-length is 1 then
    set med 80
  else
    set med 90
  endif
end-policy

as-path-set AS-PATH-SET-11
  neighbor-is '65101 65201',
  neighbor-is '65102 65202',
  neighbor-is '65103 65203',
  neighbor-is '65104 65204',
  neighbor-is '65105 65205',
  --- removed 90 similar lines ---
  neighbor-is '65195 65295',
  neighbor-is '65196 65296',
  neighbor-is '65197 65297',
  neighbor-is '65198 65298',
  neighbor-is '65199 65299'
end-set

as-path-set AS-PATH-SET-22
  originates-from '65169',
  originates-from '65392',
  originates-from '65133',
  originates-from '65231',
  originates-from '65161',
  --- removed 90 similar lines ---
  originates-from '65281',
  originates-from '65336',
  originates-from '65238',
  originates-from '65381',
  originates-from '65103'
```

end-set

이러한 출력은 전체 인터넷 BGP 테이블 스케일을 수신하고 더 전파한 후에 수집됩니다.

<#root>

RP/0/RSP1/CPU0:XR1#

show pcl protocol bgp speaker-0 neighbor-in-dflt default-IPv4-Uni-10.0.54.6 policy profile

Policy profiling data

Policy : INGRESS-ROUTE-POLICY

Pass : 720100

Drop : 0

of executions : 720100

Total execution time : 9612msec <===== about 10 seconds to process ingress updates

Node Id	Num visited	Exec time	Policy engine operation
PXL_0_1	720100	8540msec	if as-path aspath-match ... then

PXL_0_3	7128	2msec	set local-preference 150
	7128	1msec	

PXL_0_2	712972	276msec	set local-preference 50
	712972	80msec	

RP/0/RSP1/CPU0:XR1#

show pcl protocol bgp speaker-0 neighbor-out-dflt default-IPv4-Uni-UpdGrp-0.3-Out policy profile

Policy profiling data

Policy : EGRESS-ROUTE-POLICY

Pass : 720126

Drop : 0

of executions : 720126

Total execution time : 12399msec <===== about 12 seconds to process egress updates

Node Id	Num visited	Exec time	Policy engine operation
PXL_0_1	720126	190msec	if as-path is-local then
PXL_0_7	0	0msec	set med 70
	0	0msec	
PXL_0_2	720126	11190msec	if as-path aspath-match ... then
PXL_0_4	262734	65msec	if as-path unique-length is 1 then
PXL_0_5	25	0msec	set med 80
	25	0msec	

PXL_0_6	720101	164msec	set med 90
	720101	57msec	

GOTO : PXL_0_6

RP/0/RSP1/CPU0:XR1#

또는 ios-regex 라인을 축소할 수 있습니다. 이는 또한 성능을 개선하는 데도 도움이 됩니다.

[illegible]

```

if as-path in (ios-regex '^65141_65241|65142_65242|65143_65243|65144_65244|65145_65245|65146_65246|6
    set local-preference 150
endif
if as-path in (ios-regex '^65151_65251|65152_65252|65153_65253|65154_65254|65155_65255|65156_65256|6
    set local-preference 150
endif
if as-path in (ios-regex '^65161_65261|65162_65262|65163_65263|65164_65264|65165_65265|65166_65266|6
    set local-preference 150
endif
if as-path in (ios-regex '^65171_65271|65172_65272|65173_65273|65174_65274|65175_65275|65176_65276|6
    set local-preference 150
endif
if as-path in (ios-regex '^65181_65281|65182_65282|65183_65283|65184_65284|65185_65285|65186_65286|6
    set local-preference 150
else
    set local-preference 50
endif
end-policy

route-policy EGRESS-ROUTE-POLICY
if as-path in (ios-regex '^$') then
    set med 70
endif
if as-path in (ios-regex '^65169(_65169)*$|^65392(_65392)*$|^65133(_65133)*$|^65231(_65231)*$|^65161(_
    set med 80
endif
if as-path in (ios-regex '^65354(_65354)*$|^65331(_65331)*$|^65342(_65342)*$|^65295(_65295)*$|^65208(_
    set med 80
endif
if as-path in (ios-regex '^65149(_65149)*$|^65350(_65350)*$|^65115(_65115)*$|^65300(_65300)*$|^65322(_
    set med 80
endif
if as-path in (ios-regex '^65102(_65102)*$|^65329(_65329)*$|^65237(_65237)*$|^65218(_65218)*$|^65153(_
    set med 80
endif
if as-path in (ios-regex '^65263(_65263)*$|^65116(_65116)*$|^65112(_65112)*$|^65114(_65114)*$|^65378(_
    set med 80
endif
if as-path in (ios-regex '^65105(_65105)*$|^65296(_65296)*$|^65211(_65211)*$|^65317(_65317)*$|^65115(_
    set med 80
endif
if as-path in (ios-regex '^65371(_65371)*$|^65214(_65214)*$|^65325(_65325)*$|^65354(_65354)*$|^65384(_
    set med 80
endif
if as-path in (ios-regex '^65220(_65220)*$|^65277(_65277)*$|^65219(_65219)*$|^65213(_65213)*$|^65336(_
    set med 80
endif
if as-path in (ios-regex '^65249(_65249)*$|^65112(_65112)*$|^65314(_65314)*$|^65385(_65385)*$|^65152(_
    set med 80
endif
if as-path in (ios-regex '^65196(_65196)*$|^65252(_65252)*$|^65162(_65162)*$|^65271(_65271)*$|^65357(_
    set med 80
endif
if as-path in (ios-regex '^65317(_65317)*$|^65360(_65360)*$|^65198(_65198)*$|^65256(_65256)*$|^65246(_
    set med 80
endif
if as-path in (ios-regex '^65356(_65356)*$|^65359(_65359)*$|^65302(_65302)*$|^65118(_65118)*$|^65346(_
    set med 80
endif
if as-path in (ios-regex '^65225(_65225)*$|^65307(_65307)*$|^65313(_65313)*$|^65189(_65189)*$|^65288(_
    set med 80
endif

```


--- removed lines ---
GOTO : PXL_0_3

RP/0/RSP1/CPU0:XR1#

show pci protocol bgp speaker-0 neighbor-out-dflt default-IPv4-Uni-UpdGrp-0.3-Out policy profile

Policy profiling data

Policy : EGRESS-ROUTE-POLICY

Pass : 720110

Drop : 0

of executions : 720110

Total execution time : 106566msec <===== about 1.8 minutes to process egress updates

Node Id	Num visited	Exec time	Policy engine operation
PXL_0_1	720110	2958msec	if as-path aspath-match ... then

PXL_0_2	0	0msec	set med 70
PXL_0_3	720110	5222msec	if as-path aspath-match ... then

PXL_0_4	3	0msec	set med 80
PXL_0_5	720110	4979msec	if as-path aspath-match ... then

PXL_0_6			set med 80
---------	--	--	------------

--- removed lines ---
GOTO : PXL_0_3

RP/0/RSP1/CPU0:XR1#

결론

네이티브 as-path 일치 또는 축소된 정규식 패턴을 사용하여 경로 정책의 성능을 향상시킬 수 있습니다.

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.