

ODR을 사용하여 대규모 Stub 네트워크 설계

목차

[소개](#)

[Stub 네트워크 vs. 트랜짓 네트워크](#)

[허브 앤 스포크 네트워크 및 ODR](#)

[단일 종료 지점이 있는 스포크](#)

[여러 종료 지점이 있는 스포크](#)

[단일 허브를 통한 로드 밸런싱 또는 백업](#)

[다중 허브를 통한 로드 밸런싱 또는 백업](#)

[ODR과 기타 라우팅 프로토콜 비교](#)

[ODR과 EIGRP 비교](#)

[ODR 대 OSPF](#)

[OSPF Point-to-Point Stub 네트워크](#)

[OSPF Point-to-Multipoint Stub 네트워크](#)

[ODR과 RIPv2 비교](#)

[RIPv2 Over Demand Circuit](#)

[ODR을 통한 대규모 네트워크 설계](#)

[허브에서 EIGRP가 실행되는 ODR](#)

[이중화 및 요약](#)

[허브에서 OSPF가 실행되는 ODR](#)

[이중화 및 요약](#)

[포인트 투 포인트 네트워크를 사용하는 ODR](#)

[Point-to-Multipoint 네트워크를 사용하는 ODR](#)

[ODR 및 여러 공급업체](#)

[향후 성장 문제](#)

[성능](#)

[더 빠른 컨버전스를 위한 타이머 조정](#)

[허브 라우터](#)

[스포크 라우터](#)

[ODR 경로 필터링 및 요약](#)

[Telco 타이머 조정](#)

[CPU 성능](#)

[개선 사항](#)

[관련 정보](#)

소개

ODR(On-Demand Routing)은 브로드캐스트 또는 비브로드캐스트 미디어에서 다른 Cisco 디바이스를 검색하는 데 사용되는 프로토콜인 Cisco CDP(Discovery Protocol)의 향상된 기능입니다. CDP의 도움을 받아 디바이스 유형, IP 주소, 인접 Cisco 디바이스에서 실행되는 Cisco IOS[®] 버전, 인접 디

바이스의 기능 등을 찾을 수 있습니다. Cisco IOS 소프트웨어 릴리스 11.2에서는 CDP를 통해 stub 라우터의 연결된 IP 접두사를 알리기 위해 ODR이 CDP에 추가되었습니다. 이 기능은 각 네트워크 또는 서브넷에 대해 추가 5바이트, IP 주소에 4바이트, IP와 함께 서브넷 마스크를 광고하려면 1바이트가 필요합니다. ODR은 VLSM(Variable Length Subnet Mask) 정보를 전달할 수 있습니다.

ODR은 라우팅 프로토콜 업데이트에 네트워크 대역폭을 사용하지 않으려는 엔터프라이즈 소매 고객을 위해 설계되었습니다. 예를 들어 X.25 환경에서는 해당 링크를 통해 라우팅 프로토콜을 실행하는 데 많은 비용이 듭니다. 고정 라우팅은 좋은 선택이지만, 고정 경로를 수동으로 유지 관리하기에는 오버헤드가 너무 많습니다. ODR은 CPU를 많이 사용하지 않으며 레이어 2를 통해 동적으로 IP 경로를 전파하는 데 사용됩니다.

ODR은 라우팅 프로토콜이 아니므로 구성할 때 이러한 방식으로 처리해서는 안 됩니다. ODR이 레이어 2에서 CDP를 사용하므로 다른 IP 라우팅 프로토콜에 대한 기존 컨피그레이션이 ODR에서 작동하지 않습니다. ODR을 구성하려면 허브 라우터에서 `router odr` 명령을 사용합니다. ODR과 다른 IP 라우팅 프로토콜의 설계, 구현 및 상호 작용은 어려울 수 있습니다.

LAN 에뮬레이션(LANE)을 제외하고 ODR은 Cisco 700 Series 라우터 또는 ATM 링크를 통해 실행되지 않습니다.

Stub 네트워크 vs. 트랜짓 네트워크

어떤 정보도 네트워크를 통과하지 못하면 stub 네트워크입니다. 허브 앤 스포크 토폴로지는 stub 네트워크의 좋은 예입니다. 많은 사이트가 데이터 센터에 연결된 대규모 조직에서는 이러한 유형의 토폴로지를 사용합니다.

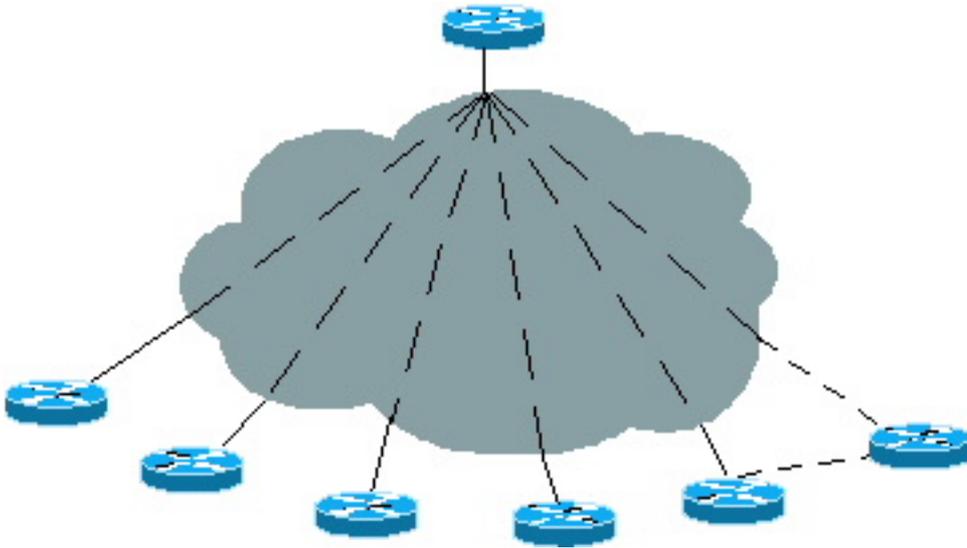
Cisco 2500, 1600 및 1000 Series 라우터와 같은 로우엔드 라우터는 스포크 측에서 사용됩니다. 정보가 스포크 라우터를 통해 다른 네트워크로 전달되면 stub 라우터는 트랜짓 라우터가 됩니다. 이 컨피그레이션은 스포크가 허브 라우터 이외의 다른 라우터에 연결될 때 발생합니다.

일반적인 문제는 스포크가 전송할 수 있는 ODR 업데이트의 크기가 얼마나 큰지입니다. 일반적으로 스포크는 허브에만 연결됩니다. 스포크가 다른 라우터에 연결되어 있으면 더 이상 토막글이 아니며 전송 네트워크가 됩니다. 로우엔드 상자에는 일반적으로 LAN 인터페이스가 하나 또는 두 개 있습니다. 예를 들어 Cisco 2500은 2개의 LAN 인터페이스를 지원할 수 있습니다. 일반적인 상황에서는 10바이트 패킷이 CDP의 일부로 전송됩니다(스포크 쪽에 2개의 LAN이 있는 경우). CDP는 기본적으로 활성화되어 있으므로 추가 오버헤드가 발생하지 않습니다. 대규모 ODR 업데이트가 있는 상황은 결코 없습니다. 진정한 허브 앤 스포크 환경에서는 ODR 업데이트 크기가 문제가 되지 않습니다.

허브 앤 스포크 네트워크 및 ODR

허브 앤 스포크 네트워크는 허브(하이엔드 라우터)가 많은 스포크(로우엔드 라우터)를 제공하는 일반적인 네트워크입니다. 특수한 경우 이중화를 위해 또는 별도의 허브를 통해 추가 스포크를 지원하기 위해 둘 이상의 허브가 있을 수 있습니다. 이 경우 두 허브에서 ODR을 활성화합니다. 또한 두 허브 간에 ODR 라우팅 정보를 교환하기 위한 라우팅 프로토콜이 있어야 합니다.

그림 1: 허브 앤 스포크 토폴로지



단일 종료 지점이 있는 스포크

위의 그림 1에서 스포크는 하나의 종료 지점으로 허브에 대한 모든 라우팅 정보를 수신하는 대신 기본 게이트웨이를 사용할 수 있도록 하나의 허브에 연결됩니다. 스포크에 모든 정보를 전달할 필요는 없습니다. 스포크가 지능적인 라우팅 결정을 내릴 필요가 없기 때문입니다. 스포크는 항상 트래픽을 허브로 전송하므로 스포크는 허브를 가리키는 기본 경로만 필요합니다.

스포크의 서브넷 정보를 허브로 보낼 수 있는 방법이 있어야 합니다. Cisco IOS 11.2 이전의 유일한 방법은 스포크에서 라우팅 프로토콜을 활성화하는 것이었습니다. 그러나 ODR을 사용하면 스포크 측에서 라우팅 프로토콜을 활성화할 필요가 없습니다. ODR을 사용하는 경우 스포크에는 Cisco IOS 11.2 및 허브를 가리키는 고정 기본 경로만 필요합니다.

여러 종료 지점이 있는 스포크

기본 링크에 오류가 발생할 경우 스포크는 이중화 또는 백업 목적으로 허브에 여러 연결을 가질 수 있습니다. 이러한 이중화에는 대개 별도의 허브가 필요합니다. 이 경우 스포크는 여러 개의 출구 지점을 가집니다. ODR도 이 네트워크에서 잘 작동합니다.

스포크는 point-to-point여야 합니다. 그렇지 않으면 부동 고정 기본 경로가 작동하지 않습니다. 멀티포인트 컨피그레이션에서는 브로드캐스트 미디어에서와 같이 다음 홉의 장애를 탐지할 방법이 없습니다.

단일 허브를 통한 로드 밸런싱 또는 백업

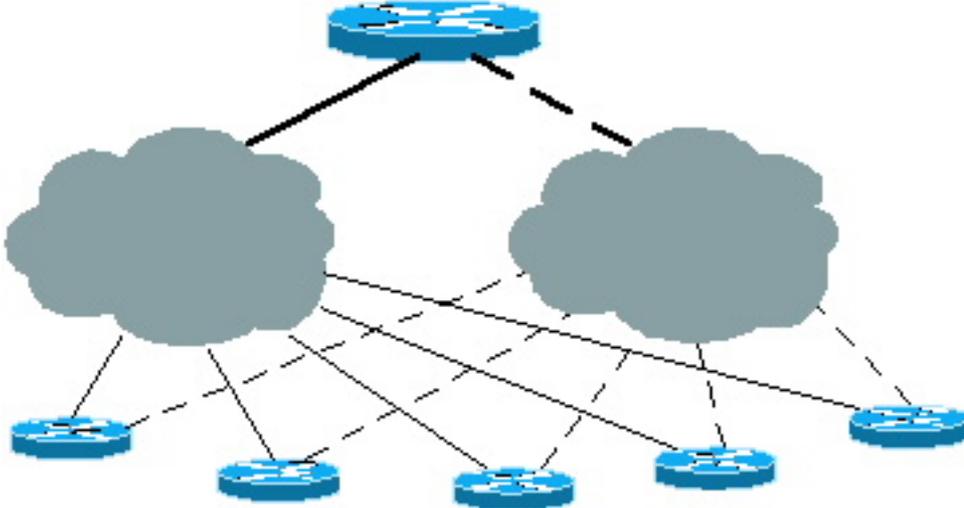
로드 밸런싱을 수행하려면 스포크에 동일한 거리의 두 고정 기본 경로를 정의하고 스포크는 두 경로 간의 로드 밸런싱을 수행합니다. 대상에 대한 두 개의 경로가 있는 경우 ODR은 라우팅 테이블에 두 경로를 모두 유지하고 허브에서 로드 밸런싱을 수행합니다.

백업의 경우 두 개의 고정 기본 경로를 다른 경로보다 더 먼 거리에서 정의합니다. 스포크는 기본 링크를 사용하며 기본 링크가 다운되면 부동 고정 경로가 작동합니다. 허브 라우터에서 각 CDP 인접 주소에 **distance** 명령을 사용하고 다른 주소보다 한 거리를 더 크게 만듭니다. 이 컨피그레이션에서는 한 링크를 통해 학습된 ODR 경로가 다른 링크보다 우선합니다. 이 구성은 빠른 기본 링크 및 느린(낮은 대역폭) 백업 링크가 있고 로드 밸런싱이 필요하지 않은 환경에서 유용합니다.

참고: 현재 스포크 측면에서는 위에 설명한 것을 제외하고 단일 허브 상황에서 한 링크를 다른 링크

보다 선호하는 다른 방법이 없습니다. IOS 12.0.5T 이상을 사용하는 경우 허브는 두 링크를 통해 기본 경로를 자동으로 전송하고 스포크는 두 경로를 구분할 수 없으며 라우팅 테이블에 둘 다 설치됩니다. 기본 경로를 다른 경로보다 선호하는 유일한 방법은 원하는 관리 거리가 낮은 경로가 있는 스포크에서 고정 기본 경로를 사용하는 것입니다. 이렇게 하면 ODR을 통해 스포크에 들어오는 기본 경로가 자동으로 재정의됩니다. 현재, 스피커를 다른 링크를 선호할 수 있는 노브를 제공하는 발상은 고려 중입니다.

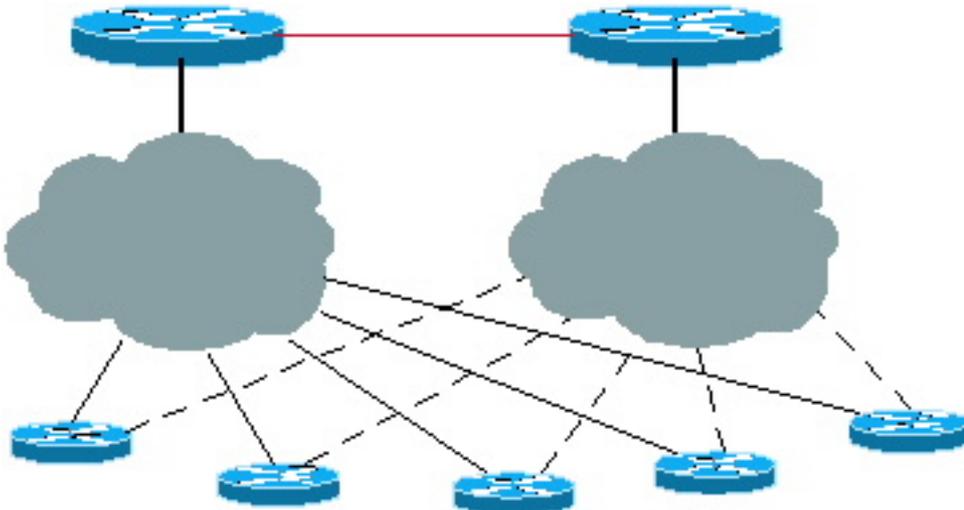
그림 2: 여러 출구 지점 및 단일 허브가 있는 스포크



다중 허브를 통한 로드 밸런싱 또는 백업

이러한 컨피그레이션은 여러 허브가 있는 경우 로드 밸런싱 또는 백업에 사용할 수도 있습니다. 모든 허브는 스포크의 링크 중 하나에 장애가 발생하더라도 두 번째 허브를 통해 대상에 도달할 수 있도록 완전히 메시징되어야 합니다. 자세한 내용은 이 문서의 [ODR 대 기타 라우팅 프로토콜](#) 섹션을 참조하십시오. 마찬가지로 다중 허브의 경우 IOS 12.0.5T 이상이 사용되는 경우 허브는 ODR 기본 경로를 스포크로 전송하고 두 경로를 라우팅 테이블에 모두 설치합니다. 향후 개선을 통해 스포크는 다른 허브에 비해 하나의 허브를 선호할 수 있습니다. 현재 이 작업은 스포크의 라우터에 정의된 고정 기본 경로를 통해 수행할 수 있으며, 정적 route 명령에서 admin 거리를 사용하여 다른 허브를 선호할 수 있습니다. 이는 로드 밸런싱 상황에 영향을 미치지 않습니다.

그림 3: 여러 출구 지점 및 다중 허브가 있는 스포크



ODR과 기타 라우팅 프로토콜 비교

IP 라우팅보다 ODR의 가장 큰 장점은 허브 라우터가 레이어 3에서 라우팅 프로토콜을 활성화하지

않고 IP 접두사를 학습한다는 것입니다. ODR 업데이트는 레이어 2에서 CDP의 일부입니다.

ODR과 EIGRP 비교

진정한 허브 앤 스포크 환경에서는 모든 스포크에 모든 라우팅 정보를 전달할 필요가 없습니다. 슬로우 링크 스포크는 라우팅 업데이트 및 인접 디바이스 관계 유지에 대역폭을 낭비합니다. 스포크에서 EIGRP(Enhanced Interior Gateway Routing Protocol)를 활성화하면 스포크에 라우팅 업데이트가 전송됩니다. 대규모 네트워크에서 이러한 업데이트는 규모가 커져서 CPU 대역폭이 낭비되고 스포크 라우터에 더 많은 메모리가 필요할 수 있습니다.

EIGRP의 더 나은 접근 방식은 허브에 필터를 적용하는 것입니다. 허브가 기본 경로를 스포크에만 동적으로 전송하도록 라우팅 정보가 제어됩니다. 이러한 필터는 스포크 측의 라우팅 테이블 크기를 줄이는 데 도움이 되지만, 허브에 인접 디바이스가 손실되면 다른 모든 네이버에 쿼리를 보냅니다. 허브는 네이버로부터 응답을 받지 않으므로 이러한 쿼리는 필요하지 않습니다.

가장 좋은 방법은 ODR을 사용하여 EIGRP 쿼리 및 네이버 유지 관리의 오버헤드를 제거하는 것입니다. ODR 타이머를 조정하여 통합 시간을 늘릴 수 있습니다.

오늘날 EIGRP의 새로운 기능은 허브 및 스포크 상황에서 EIGRP를 훨씬 더 효과적으로 확장합니다. EIGRP [stub 기능에](#) 대한 자세한 내용은 Enhanced IGRP Stub Routing을 참조하십시오.

ODR 대 OSPF

OSPF(Open Shortest Path First)는 허브 앤 스포크(hub and spoke) 환경을 위한 여러 옵션을 제공하며 stub **no-summary** 옵션은 오버헤드가 가장 낮습니다.

대규모 허브 앤 스포크 네트워크에서 OSPF를 실행할 때 문제가 발생할 수 있습니다. 이 섹션의 예는 가장 일반적인 허브 앤 스포크 토폴로지이므로 프레임 릴레이를 사용합니다.

OSPF Point-to-Point Stub 네트워크

이 예에서는 포인트-투-포인트 컨피그레이션으로 연결된 100개의 스포크에서 OSPF가 활성화됩니다. 첫째, /30 네트워크 마스크로 서브넷을 설정하더라도 IP 주소가 많이 낭비됩니다. 둘째, 한 영역에 이러한 100개의 스포크를 포함하고 한 스포크가 플랩되면 SPF(shortest path first) 알고리즘이 실행되어 CPU 집약적 상태가 될 수 있습니다. 링크가 계속 플래핑되는 경우 스포크 라우터에서는 특히 이 상황이 적용됩니다. 스포크 라우터에 관한 한 더 많은 네이버 플랩이 문제를 일으킬 수 있습니다.

OSPF에서 영역은 인터페이스가 아니라 stub입니다. stub 네트워크에 100개의 라우터가 있는 경우 큰 데이터베이스를 보유하려면 스포크에 더 많은 메모리가 필요합니다. 이 문제는 큰 stub 영역을 작은 영역으로 나누어서 해결할 수 있습니다. 그러나 한 stub 영역의 플랩은 여전히 SPF를 스포크에서 실행하도록 유도하므로 요약도 없고 외부도 없는 작은 stub 영역을 만들어 이 오버헤드를 해결할 수 없습니다.

또 다른 옵션은 각 링크를 한 영역에 포함하는 것입니다. 이 옵션을 사용하면 허브 라우터가 각 영역에 대해 별도의 SPF 알고리즘을 실행하고 해당 영역의 경로에 대한 요약 LSA(link-state advertisement)를 생성해야 합니다. 이 옵션은 허브 라우터의 성능을 해칠 수 있습니다.

더 나은 플랫폼으로 업그레이드하는 것은 영구적인 솔루션이 아닙니다. 그러나 ODR은 솔루션을 제공합니다. ODR을 통해 학습된 경로를 OSPF로 재배포하여 다른 허브 라우터에 이러한 경로에 대해 알릴 수 있습니다.

OSPF Point-to-Multipoint Stub 네트워크

Point-to-Multipoint 네트워크에서는 모든 스포크를 동일한 서브넷에 배치하여 IP 주소 공간이 저장됩니다. 또한 생성된 라우터 LSA 허브의 크기는 모든 지점 간 링크에 대해 stub 링크를 하나만 생성하므로 절반으로 줄어듭니다. Point-to-Multipoint 네트워크는 전체 서브넷을 한 영역에 포함시킵니다. 링크 플랩의 경우 스포크는 CPU가 많이 사용될 수 있는 SPF를 실행합니다.

헬로 스톱

OSPF hello 패킷은 작지만 인접 디바이스가 너무 많으면 크기가 커질 수 있습니다. hello는 멀티캐스트이므로 라우터는 패킷을 처리합니다. OSPF 허브는 20바이트의 IP 헤더, 24바이트의 OSPF 헤더, 20바이트의 hello 매개변수, 표시된 각 네이버에 4바이트로 구성된 hello 패킷을 보내고 받습니다. 인접 디바이스 100개가 있는 Point-to-Multipoint 네트워크에서 허브의 OSPF hello 패킷은 464바이트가 될 수 있으며 30초마다 모든 스포크에 플래딩됩니다.

표 1:100 인접 디바이스용 OSPF hello 패킷

20바이트 IP 헤더
24바이트 OSPF 헤더
20바이트 hello 매개변수
4바이트 각 인접 라우터 ID(RID)
...
...
...
...
...

허브에서 스포크로 추가 정보가 전송되지 않으므로 오버헤드는 ODR에서 해결됩니다. 스포크는 서브넷 당 5바이트의 IP 접두사를 허브 라우터로 전송합니다. Hello 패킷의 크기를 고려하여 30초 간격 동안 ODR의 5바이트(연결된 서브넷의 스포크 전송 정보)를 OSPF의 68바이트(스포크에서 허브로 전송된 IP 헤더를 포함한 가장 작은 hello 패킷 크기)와 68바이트(허브에서 스포크로 전송된 가장 작은 hello 패킷)를 비교합니다. 또한 OSPF 도움말은 레이어 3에서 발생하는 동안 레이어 2에서 ODR 업데이트가 발생합니다. ODR을 사용하면 훨씬 적은 정보가 전송되므로 링크 대역폭을 중요한 데이터에 사용할 수 있습니다.

ODR과 RIPv2 비교

RIPv2(Routing Information Protocol version 2)는 허브 앤 스포크(hub and spoke) 환경에도 적합합니다. RIPv2를 설계하려면 허브에서 스포크로 기본 경로를 전송합니다. 그러면 스포크는 RIP를 통해 연결된 인터페이스를 알립니다. RIPv2는 스포크에 광고해야 하는 보조 주소가 있거나 여러 공급업체 라우터를 사용하는 경우 또는 상황이 허브와 스포크가 아닌 경우에 사용할 수 있습니다.

RIPv2 Over Demand Circuit

버전 2에는 몇 가지 수정 사항이 있지만 프로토콜을 크게 변경하지 않습니다. 이 섹션에서는 수요 회로에 대한 RIP의 몇 가지 향상된 기능에 대해 설명합니다.

오늘날의 인터넷 네트워크는 다수의 원격 사이트에 대한 연결을 제공하기 위해 전화 접속 네트워크 또는 기본 사이트 백업 방향으로 이동하고 있습니다. 이러한 유형의 연결은 정상 작동 중에 데이터 트

래픽이 거의 또는 전혀 전달되지 않을 수 있습니다.

RIP의 주기적인 동작으로 인해 이러한 회로에 문제가 발생합니다. RIP는 낮은 대역폭, 포인트 투 포인트 인터페이스에 문제가 있습니다. 고대역폭을 사용하는 대규모 라우팅 테이블을 사용하여 30초마다 업데이트가 전송됩니다. 이 경우 Triggered RIP를 사용하는 것이 가장 좋습니다.

트리거된 RIP

트리거된 RIP는 모든 라우팅 정보를 인접 디바이스와 교환하는 라우터를 위해 설계되었습니다. 라우팅을 변경하면 변경 사항만 네이버로 전파됩니다. 수신 라우터가 변경 사항을 즉시 적용합니다.

다음과 같은 경우에만 트리거된 RIP 업데이트가 전송됩니다.

- 라우팅 업데이트에 대한 요청을 받았습니다.
- 새 정보가 수신됩니다.
- 대상이 회로에서 회로로 바뀌었습니다.
- 라우터가 먼저 켜져 있습니다.

다음은 트리거된 RIP의 컨피그레이션 예입니다.

```
Spoke# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Spoke(config)# int s0.1
Spoke(config-if)# ip rip triggered
Spoke(config)# int s0.2
Spoke(config-if)# ip rip triggered

interface serial 0
encapsulation frame-relay

interface serial 0.1 point                /* Primary PVC */
ip address 10.x.x.x 255.255.255.0
ip rip triggered
frame-relay interface-dlci XX

interface serial 0.2 point                /* Secondary PVC */
ip address 10.y.y.y 255.255.255.0
ip rip triggered
frame-relay interface-dlci XX

router rip
 network 10.0.0.0

Spoke# show ip protocol
Routing Protocol is "rip"
  Sending updates every 30 seconds, next due in 23 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Redistributing: rip
  Default version control: send version 1, receive any version
    Interface          Send  Recv   Triggered RIP   Key-chain
    Ethernet0          1      1 2
    Serial0.1          1      1 2                Yes
    Serial0.2          1      1 2                Yes
Routing for Networks:
  10.0.0.0
```

Routing Information Sources:

Gateway Distance Last Update

Distance: (default is 120)

스포크에 연결하는 허브 라우터의 인터페이스에서 ip rip triggered 명령을 구성해야 합니다.

RIPv2를 ODR과 비교할 때 RIPv2가 레이어 3에서 작동하고 레이어 2에서 ODR이 수행되므로 ODR이 더 나은 선택입니다. 허브가 RIPv2 업데이트를 1,000개가 넘는 스포크에 보낼 때 각 스포크에 대해 레이어 3의 패킷을 복제해야 합니다. ODR은 CPU 사용량이 전혀 없는 레이어 2에서 매분마다 일반적인 CDP 업데이트를 제외하고 허브에서 아무것도 전송하지 않습니다. 스포크에서 레이어 2에 연결된 서브넷 정보를 전송하는 것은 레이어 3에서 RIPv2를 전송하는 것보다 CPU 사용량이 훨씬 적습니다.

ODR을 통한 대규모 네트워크 설계

ODR은 다른 어떤 라우팅 프로토콜보다 대규모 네트워크에서 더 잘 작동합니다. ODR의 가장 큰 장점은 연결된 직렬 링크에서 라우팅 프로토콜을 활성화할 필요가 없다는 것입니다. 현재 연결된 인터페이스에서 라우팅 정보를 활성화하지 않으면 라우팅 정보를 전송할 수 있는 라우팅 프로토콜이 없습니다.

허브에서 EIGRP가 실행되는 ODR

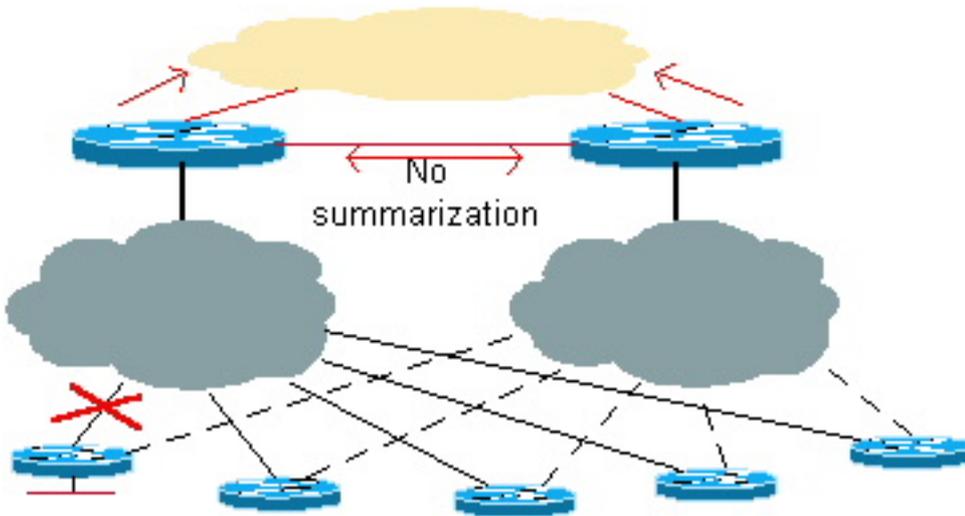
EIGRP를 실행할 때 허브 앤 스포크 네트워크에 패시브 인터페이스 연결을 설정하여 링크에 불필요한 EIGRP hello를 보내지 않도록 합니다. 가능한 경우, 연결이 끊기면 EIGRP에서 코어 네이버로 불필요한 쿼리를 보내지 않으므로 허브와 스포크 간에 네트워크에 대한 네트워크 문을 두지 않는 것이 좋습니다. 컨피그레이션에 네트워크 문을 배치하지 않기 때문에 이러한 링크가 EIGRP 도메인에 포함되지 않도록 항상 허브와 스포크 간에 가짜 네트워크를 선택합니다.

이중화 및 요약

단일 허브 상황에서는 추가 설정이 필요하지 않습니다. 스포크의 연결된 특정 서브넷을 요약하여 코어로 누수합니다. 그러나 쿼리의 오버헤드는 항상 존재합니다. 스포크 중 하나에서 특정 경로가 손실되면 코어 라우터의 모든 네이버에 쿼리를 보냅니다.

다중 허브의 경우 두 허브가 모두 연결되고 EIGRP가 허브 간에 실행되고 있어야 합니다. 가능한 경우 이 링크는 스포크로 이동하는 다른 링크를 방해하지 않도록 고유한 주 넷이어야 합니다. 이 컨피그레이션은 특정 인터페이스에서 EIGRP를 활성화할 수 없으므로 필요하므로 인터페이스를 패시브로 설정하더라도 EIGRP를 통해 계속 광고됩니다. 인터페이스가 요약된 경우 스포크 하나가 손실되어도 쿼리는 계속 전송됩니다. 두 허브 간의 링크가 스포크와 동일한 주 네트워크에 있지 않으면 컨피그레이션이 제대로 작동해야 합니다.

그림 4: 이중화 및 요약: 코어는 요약 경로를 수신하고 있습니다.



EIGRP의 장점은 인터페이스 레벨에서 요약할 수 있기 때문에 스포크 서브넷의 요약 경로가 코어로 전송되고 더 구체적인 경로를 다른 허브로 전송한다는 것입니다. 허브와 스포크 간 링크가 중단되면 두 번째 허브를 통해 대상에 도달할 수 있습니다.

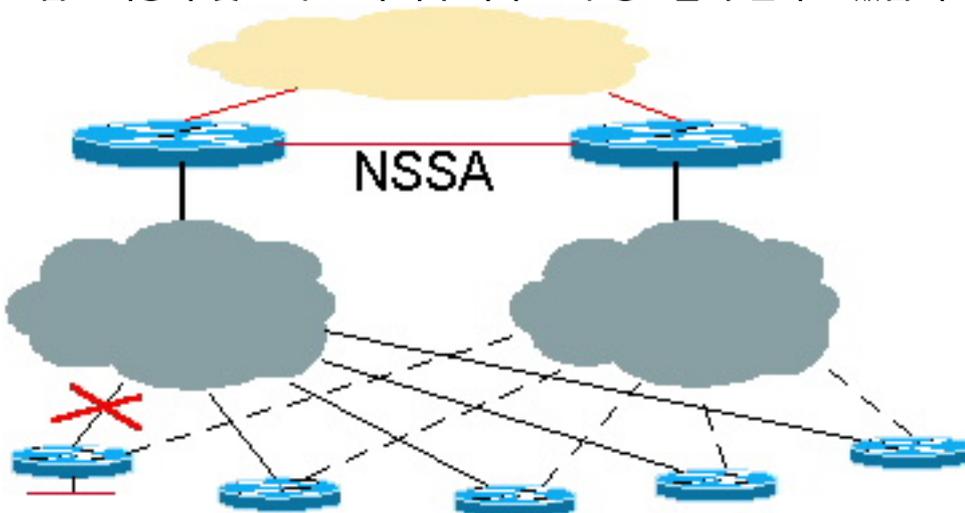
허브에서 OSPF가 실행되는 ODR

이 시나리오에서는 스포크를 연결하는 링크에서 OSPF를 활성화할 필요가 없습니다. 일반적인 시나리오에서 링크에서 OSPF가 활성화되어 있고 하나의 특정 링크가 지속적으로 플래핑되는 경우 SPF 실행, 라우터 LSA 재생성, 요약 LSA 재생성 등 여러 문제가 발생할 수 있습니다. ODR을 실행할 때 OSPF 도메인에 연결된 직렬 링크를 포함하지 마십시오. 주요 관심사는 스포크의 LAN 세그먼트 정보를 수신하는 것입니다. 이 정보는 ODR을 통해 얻을 수 있습니다. 링크 하나가 지속적으로 플래핑되면 허브 라우터의 라우팅 프로토콜에 지장을 주지 않습니다.

이중화 및 요약

스포크의 연결된 인터페이스 중 하나가 다운될 경우 경로 계산을 방지하기 위해 코어로 유출되기 전에 모든 특정 링크를 요약할 수 있습니다. 코어 라우터 정보를 요약하면 탐지할 수 없습니다.

그림 5: 이중화 및 요약: 코어 라우터가 요약 경로를 수신하고 있습니다.



이 예에서는 이중화를 위해 허브를 서로 연결하는 것이 매우 중요합니다. 이 연결은 OSPF 코어로 유출되기 전에 스포크 연결 서브넷을 요약합니다.

NSSA 및 향후 개선 사항

NSSA(Not-So-Stubby Areas) 기능은 코어로 요약할 뿐만 아니라 NSSA 링크를 통해 허브 전반에 걸쳐 더 구체적인 정보를 제공하는 OSPF NSSA(Not-So-Stubby Areas) 기능을 제공합니다. NSSA를 실행하면 요약된 경로를 코어로 전송할 수 있다는 이점이 있습니다. 그런 다음 코어는 트래픽을 두 허브 중 하나로 전송하여 스포크의 대상에 도달할 수 있습니다. 허브와 스포크 간 링크가 다운되면 두 허브에 있는 더 구체적인 Type 7 LSA가 다른 허브를 통해 목적지에 도달합니다.

다음은 NSSA를 사용하는 컨피그레이션 예입니다.

N2507: **Hub 1**

```
router odr
  timers basic 8 24 0 1
!
router ospf 1
  redistribute odr subnets
  network 1.0.0.0 0.255.255.255 area 1
  area 1 nssa
```

N2504: **Hub 2**

```
router odr
  timers basic 8 24 0 1
!
router ospf 1
  redistribute odr subnets
  network 1.0.0.0 0.255.255.255 area 1
  area 1 nssa
```

N2507# **show ip route**

```
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
       U - per-user static route, o - ODR
```

Gateway of last resort is not set

```
C    1.0.0.0/8 is directly connected, Serial0
C    2.0.0.0/8 is directly connected, Serial1
     3.0.0.0/24 is subnetted, 1 subnets
C      3.3.3.0 is directly connected, Ethernet0
o    150.0.0.0/16 [160/1] via 3.3.3.2, 00:00:23, Ethernet0
o    200.1.1.0/24 [160/1] via 3.3.3.2, 00:00:23, Ethernet0
o    200.1.2.0/24 [160/1] via 3.3.3.2, 00:00:23, Ethernet0
```

N2504# **show ip route**

```
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
       U - per-user static route, o - ODR
```

Gateway of last resort is not set

```
C    1.0.0.0/8 is directly connected, Serial0
C    2.0.0.0/8 is directly connected, Serial1
     3.0.0.0/24 is subnetted, 1 subnets
C      3.3.4.0 is directly connected, TokenRing0
```

```
C 5.0.0.0/8 is directly connected, Loopback0
C 6.0.0.0/8 is directly connected, Loopback1
O N2 150.0.0.0/16 [110/20] via 1.0.0.1, 00:12:06, Serial0
O N2 200.1.1.0/24 [110/20] via 1.0.0.1, 00:12:06, Serial0
O N2 200.1.2.0/24 [110/20] via 1.0.0.1, 00:12:06, Serial0
```

NSSA를 통한 요약 및 향후 개선

다음 예와 같이 서브넷을 OSPF 코어에 적절하게 요약할 수 있도록 스포크에 인접 서브넷 블록을 할당합니다. 서브넷이 요약되지 않고 연결된 서브넷 하나가 다운되면 전체 코어가 이를 탐지하고 경로를 재계산합니다. 연속 블록에 대한 요약 경로를 전송하여 스포크 서브넷이 플랩되면 코어는 이를 탐지하지 못합니다.

```
N2504# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
N2504(config)# router ospf 1
N2504(config-router)# summary-address 200.1.0.0 255.255.0.0
```

```
N2504# show ip ospf database external
```

```
OSPF Router with ID (6.0.0.1) (Process ID 1)
```

```
Type-5 AS External Link States
```

```
LS age: 1111
Options: (No TOS-capability, DC)
LS Type: AS External Link
Link State ID: 200.1.0.0 (External Network Number )
Advertising Router: 6.0.0.1
LS Seq Number: 80000001
Checksum: 0x2143
Length: 36
Network Mask: /16
Metric Type: 2 (Larger than any link state path)
TOS: 127
Metric: 16777215
Forward Address: 0.0.0.0
External Route Tag: 0
```

거리 문제

이 예에서는 두 허브에서 더 구체적인 정보를 수신합니다. OSPF 거리는 110이고 ODR 거리는 160이므로 동일한 서브넷에 대해 다른 허브에서 ODR을 수신하면 정보가 간섭됩니다. 다른 허브는 항상 스포크 대상에 도달하는 것을 선호하므로 최적의 라우팅이 되지 않습니다. 상황을 해결하려면 **distance** 명령을 사용하여 ODR 거리를 110 미만으로 줄여서 ODR 경로가 항상 OSPF 경로보다 우선합니다. ODR 경로가 실패하면 OSPF 외부 경로가 데이터베이스의 라우팅 테이블에 설치됩니다.

```
N2504(config)# router odr
N2504(config-router)# distance 100
N2504(config-router)# end
```

```
N2504# show ip route
```

```
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
```

U - per-user static route, o - ODR

Gateway of last resort is not set

```
C 1.0.0.0/8 is directly connected, Serial0
C 2.0.0.0/8 is directly connected, Serial1
  3.0.0.0/24 is subnetted, 1 subnets
C    3.3.4.0 is directly connected, TokenRing0
C 5.0.0.0/8 is directly connected, Loopback0
C 6.0.0.0/8 is directly connected, Loopback1
o 150.0.0.0/16 [100/1] via 3.3.4.1, 00:00:39, TokenRing0
o 200.1.1.0/24 [100/1] via 3.3.4.1, 00:00:39, TokenRing0
o 200.1.2.0/24 [100/1] via 3.3.4.1, 00:00:39, TokenRing0
O 200.1.0.0/16 is a summary, 00:04:38, Null0
```

N2 경로는 아직 데이터베이스에 있으며 스포크에 대한 주 허브 링크가 다운되면 활성화됩니다.

N2504# **show ip ospf database nssa**

OSPF Router with ID (6.0.0.1) (Process ID 1)

Type-7 AS External Link States (Area 1)

```
LS age: 7
Options: (No TOS-capability, Type 7/5 translation, DC)
LS Type: AS External Link
Link State ID: 150.0.0.0 (External Network Number )
Advertising Router: 6.0.0.1
LS Seq Number: 80000002
Checksum: 0x965E
Length: 36
Network Mask: /16
  Metric Type: 2 (Larger than any link state path)
  TOS: 0
  Metric: 20
  Forward Address: 1.0.0.2
  External Route Tag: 0
```

NSSA가 개선됨에 따라 Type 7 더 구체적인 LSA가 NSSA 데이터베이스에 포함될 것입니다. 요약된 경로 대신 NSSA 데이터베이스의 출력이 아래와 같이 표시됩니다.

```
LS age: 868
Options: (No TOS-capability, Type 7/5 translation, DC)
LS Type: AS External Link
Link State ID: 200.1.1.0 (External Network Number)
Advertising Router: 3.3.3.1
LS Seq Number: 80000001
Checksum: 0xDFE0
Length: 36
Network Mask: /24
  Metric Type: 2 (Larger than any link state path)
  TOS: 0
  Metric: 20
  Forward Address: 1.0.0.1
  External Route Tag: 0
```

```
LS age: 9
Options: (No TOS-capability, Type 7/5 translation, DC)
LS Type: AS External Link
Link State ID: 200.1.2.0 (External Network Number)
```

Advertising Router: 3.3.3.1

LS Seq Number: 80000002

Checksum: 0xFDC3

Length: 36

Network Mask: /24

Metric Type: 2 (Larger than any link state path)

TOS: 0

Metric: 20

Forward Address: 1.0.0.2

External Route Tag: 0

수요 회로

수요 회로는 허브 앤 스포크 네트워크에서도 사용할 수 있는 Cisco IOS 11.2 기능입니다. 이 기능은 일반적으로 다이얼 백업 시나리오와 X.25 또는 SVC(Frame Relay Switched Virtual Circuit) 환경에서 유용합니다. 다음은 수요 회로의 구성 예입니다.

```

router ospf 1
network 1.1.1.0 0.0.0.255 area 1
area 1 stub no-summary

interface Serial0 /* Link to the hub router */
ip address 1.1.1.1 255.255.255.0
ip ospf demand-circuit
clockrate 56000

```

```

Spoke#show ip o int s0
Serial0 is up, line protocol is up
Internet Address 1.1.1.1/24, Area 1
Process ID 1, Router ID 141.108.4.2, Network Type POINT_TO_POINT, Cost: 64
Configured as demand circuit.
Run as demand circuit.
DoNotAge LSA not allowed (Number of DCbitless LSA is 1).
Transmit Delay is 1 sec, State POINT_TO_POINT,
Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
Hello due in 00:00:06
Neighbor Count is 1, Adjacent neighbor count is 1
Adjacent with neighbor 130.2.4.2
Suppress hello for 0 neighbor(s)

```

허브 앤 스포크(hub and spoke) 네트워크에서 디맨드 회로(demand circuit) 기능을 사용하면 토폴로지에 변경 사항이 있을 경우 회로가 활성화되고 새로운 인접성이 형성됩니다. 예를 들어 스포크에 폴랩하는 서브넷이 있는 경우 수요 회로는 인접성을 표시하고 이 정보를 플러딩합니다. stubby-area 환경에서는 stub 영역 전체에 이 정보가 플러딩됩니다. ODR은 이 정보를 다른 스포크에 유출하지 않고 이 문제를 해결합니다. 자세한 내용은 [OSPF Demand Circuit 기능](#)을 참조하십시오.

포인트 투 포인트 네트워크를 사용하는 ODR

IDB(Interface Descriptor Block) 제한의 현재 Cisco IOS 12.0 상태는 다음과 같습니다.

라우터	제한
1000	300
2600	300
3600	800
4x00	300
5200	300

5300	700
5800	3000
7200	3000
RSP	1000

IOS 12.0 이전에는 IDB 제한으로 인해 허브가 지원할 수 있는 최대 스포크 수는 300개입니다. 네트워크에 스포크가 300개 이상 필요한 경우 포인트 투 포인트 컨피그레이션이 적합하지 않았습니다. 또한 각 링크에 대해 별도의 CDP 패킷이 생성되었습니다. 포인트 투 포인트 링크에서 CDP 업데이트를 전송하는 데 따르는 복잡성은 n^2 입니다. 위의 표를 보면 다른 플랫폼에 대한 IDB 제한을 알 수 있습니다. 각 플랫폼에서 지원되는 최대 스포크 수는 다르지만 각 링크에 대해 별도의 CDP 패킷을 생성하는 오버헤드는 여전히 문제입니다. 따라서 대규모 허브 및 스포크 상황에서 포인트-투-멀티포인트 인터페이스를 구성하는 것이 포인트-투-포인트 인터페이스보다 더 나은 솔루션입니다.

Point-to-Multipoint 네트워크를 사용하는 ODR

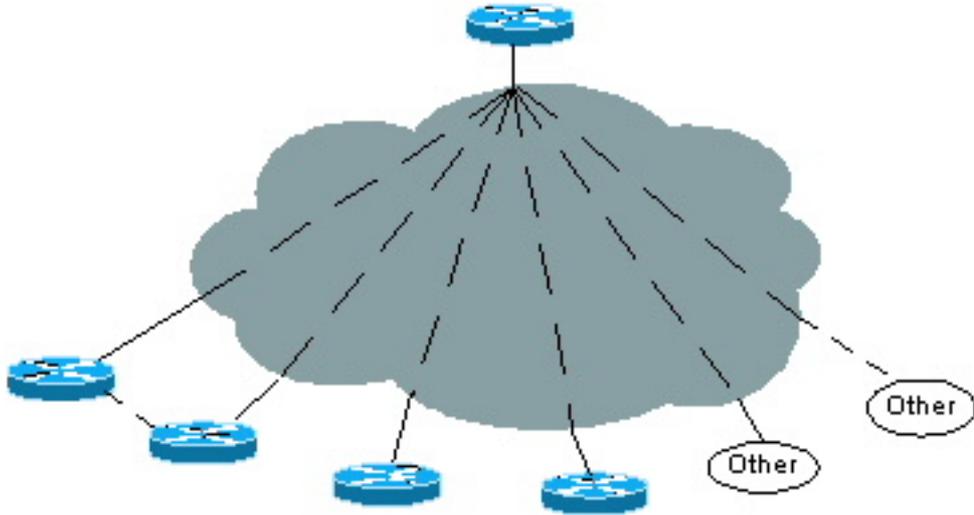
허브가 여러 스포크를 지원하는 Point-to-Multipoint 네트워크에는 다음과 같은 세 가지 주요 문제가 있습니다.

- 허브는 300개 이상의 스포크를 쉽게 지원할 수 있습니다. 예를 들어 10.10.0.0/22 네트워크는 멀티포인트 인터페이스를 사용하여 1024-2 스포크를 지원할 수 있습니다.
- 다중 지점 환경에서는 모든 네이버에 대해 하나의 CDP 패킷이 생성되고 레이어 2에 복제됩니다. CDP 업데이트의 시간 복잡성이 n 으로 감소합니다.
- 지점 간 컨피그레이션에서는 모든 스포크에 하나의 서브넷만 할당할 수 있습니다.

ODR 및 여러 공급업체

한 가지 일반적인 오해 중 하나는 여러 벤더를 사용할 경우 ODR이 작동하지 않을 것이라는 것입니다. 네트워크가 진정한 허브 앤 스포크 네트워크인 경우 ODR이 작동합니다. 예를 들어 100개의 스포크가 있고 두 스포크가 다른 벤더의 라우터인 경우, 다른 라우터에 연결하는 링크에서 라우팅 프로토콜을 활성화하고 나머지 98개의 Cisco 스포크에서 ODR을 실행할 수 있습니다.

그림 6: 여러 벤더의 ODR

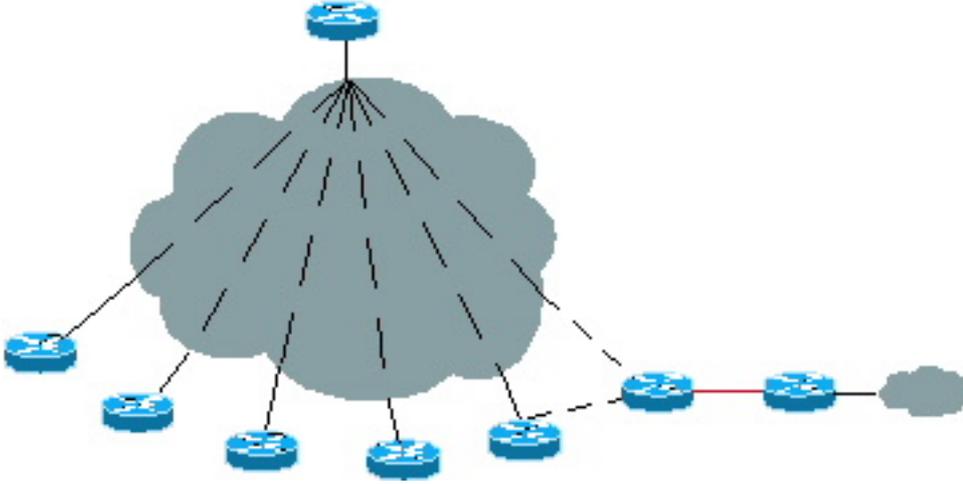


98 Cisco 라우터에 연결된 허브 라우터는 ODR을 통해 서브넷 업데이트를 수신하며 나머지 두 라우터에서 라우팅 프로토콜 업데이트를 수신합니다. 서로 다른 라우터에 연결하는 링크는 개별 포인트-투-포인트 또는 포인트-투-멀티포인트 서브넷에 있어야 합니다.

향후 성장 문제

조직에서 100개 스포크에서 ODR을 실행하는 경우 결국 허브 앤 스포크(hub and spoke) 네트워크에서 토폴로지를 변경하려고 할 수 있습니다. 예를 들어, 한 스포크를 더 큰 플랫폼으로 업그레이드하여 스포크를 20개의 다른 새 스포크에 대한 허브로 만들 수 있습니다.

그림 7:향후 성장



새 허브에서 라우팅 프로토콜을 실행하고 ODR 설계를 그대로 유지할 수 있습니다. 새 허브가 20개 이상의 새 스포크를 지원할 경우 ODR은 새 허브에서 실행될 수 있습니다. 새 허브는 ODR을 통해 새 스포크 서브넷에 대해 학습하고 다른 라우팅 프로토콜을 통해 이 정보를 원래 허브에 재배포할 수 있습니다.

이 상황은 ODR이 두 개의 허브로 시작할 때와 비슷합니다. 프로토콜 변경에 따른 오버헤드가 없습니다. 기본적으로 ODR은 라우터가 stub인 한 실행될 수 있습니다.

성능

ODR을 실행할 때 컨버전스 속도와 성능 향상을 위해 몇 가지 설정을 조정할 수 있습니다.

더 빠른 컨버전스를 위한 타이머 조정

대규모 ODR 환경에서는 ODR 타이머를 조정하여 통합 속도를 높이고, 허브에서 스포크로 CDP 업데이트 타이머를 늘려 허브의 CPU 성능을 최소화합니다.

허브 라우터

CDP 업데이트 타이머는 기본적으로 60초로 설정되어야 허브에서 스포크로의 트래픽 양을 줄일 수 있습니다. 대기 시간은 최대(255초)로 늘려야 합니다. 허브 라우터는 너무 많은 CDP 인접성 테이블을 유지 관리해야 하며 일부 인접 디바이스가 다운될 경우 255초 동안(허용되는 최대 대기 시간) 메모리에서 CDP 항목을 삭제하지 마십시오. 이 컨피그레이션은 허브 라우터에 유연성을 부여합니다. 네이버가 4분 내에 복구되면 CDP 인접성을 다시 생성할 필요가 없기 때문입니다. 이전 테이블 항목을 사용할 수 있으며 holddown 타이머를 업데이트할 수 있습니다.

다음은 중앙 라우터에 대한 IP 컨피그레이션 템플릿의 예입니다.

```
cdp holdtime 255
```

```
router odr
```

```
timers basic 8 24 0 1 /* odr timer's are update, invalid, hold down, flush
```

```
router eigrp 1
```

```
network 10.0.0.0
```

```
redistribute odr
```

```
default-metric 1 1 1 1 1
```

각 원격 사이트(참고, 지역 및 참고)에는 3개의 영구 PVC(Virtual Circuit)가 있습니다. PVC 중 2개는 별도의 중앙 라우터 2개로 이동합니다. 세 번째 PVC는 PayPoint 라우터로 이동합니다. PayPoint 네트워크로 향하는 트래픽에는 PayPoint 경로에 대한 PVC를 사용해야 합니다. 다른 두 PVC는 다른 모든 트래픽에 대해 기본 및 백업 기능을 제공합니다. 이러한 요구 사항에 따라 각 원격 라우터의 [구성 템플릿](#)을 참조하십시오.

ODR 타이머를 **유효하지 않음**, 대기, 플러시 등 더 빠른 컨버전스를 위해 조정하는 것이 매우 중요합니다. CDP는 라우터 odr이 구성되면 IP 접두사를 전송하지 않지만 컨버전스 타이머가 업데이트 타이머가 있는 경우에만 설정할 수 있으므로 ODR 업데이트 타이머가 인접 디바이스 CDP 업데이트 타이머와 일치해야 합니다. 이 타이머는 CDP 타이머와 다르며 더 빠른 컨버전스에만 사용할 수 있습니다.

[스포크 라우터](#)

스포크가 CDP 패킷에서 ODR 업데이트를 전송하기 때문에 CDP 업데이트 타이머를 매우 작게 유지하여 더 빠르게 통합해야 합니다. 실제 스포크 환경에서는 CDP 테이블에 **보관할 스포크 항목**이 몇 개뿐이므로 CDP 인접 디바이스의 대기 시간을 제한할 수 없습니다. 허브 PVC가 다운되어 4분 내에 다시 돌아오면 이전 테이블 항목을 사용할 수 있으므로 새 CDP 인접성이 필요하지 않도록 최대 대기 시간 255초를 사용하는 것이 좋습니다.

다음은 원격 사이트에 대한 IP 구성 템플릿의 예입니다.

```
cdp timer 8
```

```
cdp holdtime 255
```

```
interface serial 0
```

```
encapsulation frame-relay
```

```
cdp enable
```

```
interface serial 0.1 point
```

```
/* Primary PVC */
```

```
ip address 10.x.x.x 255.255.255.0
```

```
frame-relay interface-dlci XX
```

```
interface serial 0.2 point
```

```
/* Secondary PVC */
```

```
ip address 10.y.y.y 255.255.255.0
```

```
frame-relay interface-dlci XX
```

```
interface bri 0
```

```
interface BRI0
```

```
description Backup ISDN for frame-relay
```

```
ip address 10.c.d.e 255.255.255.0
```

```
encapsulation PPP
```

```
dialer idle-timeout 240
```

```
dialer wait-for-carrier-time 60
```

```
dialer map IP 10.x.x.x name ROUTER2 broadcast xxxxxxxxxx
ppp authentication chap
dialer-group 1
isdn spid1 xxxxxxxx
isdn spid2 xxxxxxxx

access-list 101 permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
dialer-list 1 LIST 101
```

/* following are the static routes that need to be configured on the remote routers

```
ip route 0.0.0.0 0.0.0.0 10.x.x.x
ip route 0.0.0.0 0.0.0.0 10.y.y.y
ip route 0.0.0.0 0.0.0.0 bri 0 100
ip classless
```

허브 라우터가 모든 스포크에 자동으로 기본 경로를 전송하므로 IOS 12.0.5T 이상을 사용하는 경우에는 고정 기본 경로가 필요하지 않습니다.

ODR 경로 필터링 및 요약

ODR 경로는 코어로 유출되기 전에 필터링할 수 있습니다. `distribute-list in` 명령을 사용합니다. 코어로 누출될 때 스포크의 연결된 모든 서브넷을 요약해야 합니다. 요약이 불가능한 경우 허브 라우터에서 불필요한 경로를 필터링할 수 있습니다. 여러 허브 네트워크에서 스포크는 다른 허브에 대한 링크인 연결된 인터페이스를 알릴 수 있습니다.

이 경우 허브가 라우팅 테이블에 경로를 넣지 않도록 `distribute-list` 명령을 적용해야 합니다. ODR이 허브에 재배포될 때 해당 정보는 코어로 유출되지 않습니다.

Telco 타이머 조정

스포크의 컨버전스 시간을 늘리려면 통신 타이머를 조정하는 것이 중요합니다. 허브 측의 PVC가 다 운되면 스포크는 이를 신속하게 탐지하여 두 번째 허브로 전환할 수 있어야 합니다.

CPU 성능

ODR 프로세스에는 CPU 사용률이 많지 않습니다. ODR은 CPU 사용률이 3~4%인 약 1,000개의 인접 디바이스에 대해 테스트되었습니다. 허브에서 ODR의 적극적인 타이머 설정은 더 빠른 컨버전스를 지원합니다. 기본 설정을 사용하는 경우 CPU 사용률은 0~1%로 유지됩니다.

적극적인 ODR 및 CDP 타이머에서도 아래 출력에서는 높은 CPU 사용률이 없었다는 것을 보여줍니다. 이 테스트는 Cisco 7206의 150MHz 프로세서로 수행되었습니다.

```
Hub# show proc cpu
CPU utilization for five seconds: 4%/0%; one minute: 3%; five minutes: 3%
PID Runtime(ms) Invoked uSecs 5Sec 1Min 5Min TTY Process
.
.
18 11588036 15783316 734 0.73% 1.74% 1.95% 0 CDP Protocol
.
.
48 3864 5736 673 0.00% 0.00% 0.00% 0 ODR Router
```

Hub# **show proc cpu**

CPU utilization for five seconds: 3%/0%; one minute: 3%; five minutes: 3%

PID	Runtime(ms)	Invoked	uSecs	5Sec	1Min	5Min	TTY	Process
.
18	11588484	15783850	734	2.21%	1.83%	1.96%	0	CDP Protocol
.
48	3864	5736	673	0.00%	0.00%	0.00%	0	ODR Router

Hub# **show proc cpu**

CPU utilization for five seconds: 2%/0%; one minute: 3%; five minutes: 3%

PID	Runtime(ms)	Invoked	uSecs	5Sec	1Min	5Min	TTY	Process
.
18	11588676	15784090	734	1.31%	1.79%	1.95%	0	CDP Protocol
.
48	3864	5736	673	0.00%	0.00%	0.00%	0	ODR Router

Hub# **show proc cpu**

CPU utilization for five seconds: 1%/0%; one minute: 3%; five minutes: 3%

PID	Runtime(ms)	Invoked	uSecs	5Sec	1Min	5Min	TTY	Process
.
18	11588824	15784283	734	0.65%	1.76%	1.94%	0	CDP Protocol
.
48	3864	5737	673	0.00%	0.00%	0.00%	0	ODR Router

Hub# **show proc cpu**

CPU utilization for five seconds: 3%/0%; one minute: 3%; five minutes: 3%

PID	Runtime(ms)	Invoked	uSecs	5Sec	1Min	5Min	TTY	Process
.
18	11589004	15784473	734	1.96%	1.85%	1.95%	0	CDP Protocol
.
48	3864	5737	673	0.00%	0.00%	0.00%	0	ODR Router

Hub# **show proc cpu**

CPU utilization for five seconds: 3%/0%; one minute: 3%; five minutes: 3%

PID	Runtime(ms)	Invoked	uSecs	5Sec	1Min	5Min	TTY	Process
.
18	11589188	15784661	734	1.63%	1.83%	1.94%	0	CDP Protocol
.
48	3864	5737	673	0.00%	0.00%	0.00%	0	ODR Router

개선 사항

Cisco IOS 12.0.5T 이전 버전의 ODR에는 몇 가지 제한이 있었습니다.다음은 Cisco IOS 12.0.5T 이상의 개선 사항 목록입니다.

- CSCdy48736 이전에는 보조 서브넷이 CDP 업데이트에서 /32로 광고됩니다.12.2.13T 이상에서 수정되었습니다.

- CDP 허브는 기본 경로를 스포크에 전파하므로 스포크에 고정 기본 경로를 추가할 필요가 없습니다. 컨버전스 시간이 크게 증가합니다. 다음 홉이 다운되면 스포크는 ODR을 통해 이를 빠르게 탐지하고 컨버전스를 수행합니다. 이 기능은 12.0.5T에서 버그 CSCdk91586을 통해 추가됩니다.
- 허브와 스포크 간 링크가 번호가 지정되지 않은 IP인 경우, 허브에서 보낸 기본 경로가 스포크에서 표시되지 않을 수 있습니다. 이 버그 CSCdx66917은 IOS 12.2.14, 12.2.14T 이상에서 고정되어 있습니다.
- 스포크의 ODR 거리를 늘리거나 줄여서 한 허브를 다른 허브에 비해 선호하게 하려면 CSCdr35460을 통해 추적되는 제안이 있습니다. 이 코드는 이미 테스트되었으며 곧 고객에게 제공될 예정입니다.

관련 정보

- [IP 라우팅 지원 페이지](#)
- [Technical Support - Cisco Systems](#)