

Designing Large-Scale Stub Networks with ODR

Document ID: 13710

Author: Faraz Shamim, Cisco Systems, Inc.

Introduction

Stub Networks vs. Transit Networks

Hub-and-Spoke Networks and ODR

Spokes with a Single Exit Point

Spokes with Multiple Exit Points

- Load Balancing or Backup with a Single Hub

- Load Balancing or Backup with Multiple Hubs

ODR vs. Other Routing Protocols

ODR vs. EIGRP

ODR vs. OSPF

- OSPF Point-to-Point Stub Networks

- OSPF Point-to-Multipoint Stub Networks

ODR vs. RIPv2

- RIPv2 Over Demand Circuit

Large-Scale Network Design with ODR

ODR with EIGRP Running on Hubs

- Redundancy and Summarization

ODR with OSPF Running on Hubs

- Redundancy and Summarization

ODR with Point-to-Point Networks

ODR with Point-to-Multipoint Networks

ODR and Multiple Vendors

Future Growth Issues

Performance

Timers Adjustment for Faster Convergence

- Hub Router

- Spoke Router

Filter and Summarization of ODR Routes

Telco Timer Adjustment

CPU Performance

Enhancements

Related Information

Introduction

On-Demand Routing (ODR) is an enhancement to Cisco Discovery Protocol (CDP), a protocol used to discover other Cisco devices on either broadcast or non-broadcast media. With the help of CDP, it is possible to find the device type, the IP address, the Cisco IOS[®] version running on the neighbor Cisco device, the capabilities of the neighbor device, and so on. In Cisco IOS software release 11.2, ODR was added to CDP to advertise the connected IP prefix of a stub router via CDP. This feature takes an extra five bytes for each network or subnet, four bytes for the IP address, and one byte to advertise the subnet mask along with the IP. ODR is able to carry Variable Length Subnet Mask (VLSM) information.

ODR was designed for enterprise retail customers who do not want to use their network bandwidth for routing protocol updates. In an X.25 environment, for example, it is often very costly to run a routing protocol over that link. Static routing is a good choice, but there is too much overhead to manually maintain the static

routes. ODR is not CPU-intensive and it is used to propagate IP routes dynamically over Layer 2.

ODR is not a routing protocol and should not be treated as such when configuring it. Traditional configurations for different IP routing protocols will not work in ODR, as ODR uses CDP on Layer 2. To configure ODR, use the **router odr** command on the hub router. The design, implementation, and interaction of ODR with other IP routing protocols can be difficult.

ODR will not run on Cisco 700 series routers or over ATM links with the exception of LAN emulation (LANE).

Stub Networks vs. Transit Networks

When no information is passing through the network, it is a stub network. Hub-and-spoke topology is a good example of a stub network. Large organizations with many sites connected to a data center use this type of topology.

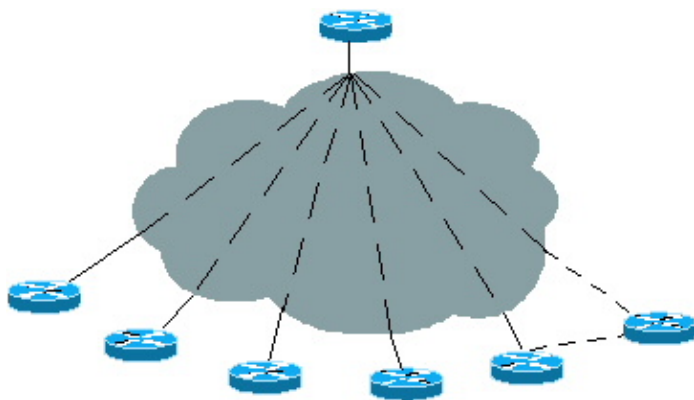
Low-end routers such as Cisco 2500, 1600, and 1000 series routers are used on the spoke side. If information passes through spoke routers to get to some other network, that stub router becomes a transit router. This configuration occurs when a spoke is connected to another router besides the hub router.

A common concern is how big of an ODR update a spoke can send. Normally, spokes are connected only to a hub. If spokes are connected to other routers, they are no longer stubs and become a transit network. Low-end boxes usually have one or two LAN interfaces. For example, the Cisco 2500 can support two LAN interfaces. In normal situations, a 10-byte packet is sent (in case there are two LANs on the spoke side) as a part of CDP. CDP is enabled by default, so there is no issue of extra overhead. There will never be a situation where there is a large ODR update. The size of ODR updates will not be a problem in a true hub-and-spoke environment.

Hub-and-Spoke Networks and ODR

A hub-and-spoke network is a typical network where a hub (high-end router) serves many spokes (low-end routers). In special cases, there may be more than one hub, either for redundancy purposes or to support additional spokes via a separate hub. In this situation, enable ODR on both hubs. It is also necessary to have a routing protocol to exchange ODR routing information between the two hubs.

Figure 1: Hub-and-spoke topology



Spokes with a Single Exit Point

In Figure 1 above, the spokes are connected to one hub so that they can rely on the default gateway instead of receiving all the routing information for the hub with one exit point. It is not necessary to pass all the

information to the spokes, since a spoke will not have to make an intelligent routing decision. A spoke will always send the traffic to the hub, so the spokes need only a default route pointing toward a hub.

There must be a way for the spoke's subnet information to be sent to the hub. Before Cisco IOS 11.2, the only way to achieve this was to enable a routing protocol at the spoke. Using ODR, however, routing protocols do not need to be enabled on the spoke side. With ODR, only Cisco IOS 11.2 and a static default route pointing toward a hub are needed on the spoke.

Spokes with Multiple Exit Points

A spoke may have multiple connections to the hub for redundancy or backup purposes in case the primary link fails. A separate hub is often required for this redundancy. In this situation, the spokes have multiple exit points. ODR also works well in this network.

Spokes must be point-to-point, otherwise the floating static default route will not work. In a multi-point configuration, there is no way to detect a failure of the next hop, just like in a broadcast media.

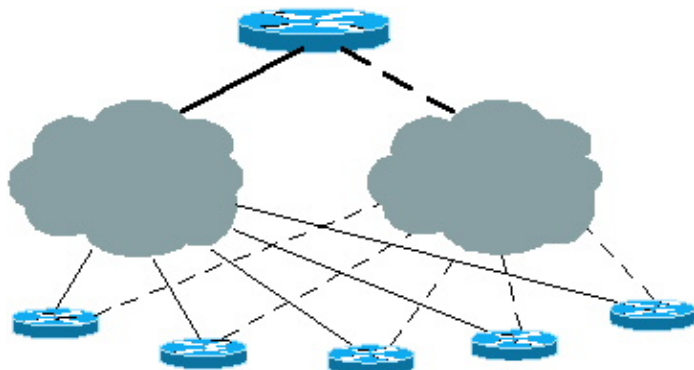
Load Balancing or Backup with a Single Hub

To achieve load balancing, define two static default routes on spokes with the same distance and the spoke will do load balancing between those two paths. If there are two paths to the destination, ODR will keep both routes in the routing table and will do load balancing on the hub.

For backups, define two static default routes with a distance of one better than the other. The spoke will use the primary link and, when the primary link goes down, the floating static route will work. In the hub router, use the **distance** command for each CDP neighbor address and make one distance better than the other. With this configuration, the ODR routes learned through one link will be preferred over the other. This configuration is useful in an environment where there are fast primary links and slow (low bandwidth) backup links and where load balancing is not desired.

Note: Today, there is no other method on the spoke side to prefer one link over the other in a single hub situation, except as described above. If you are using IOS 12.0.5T or later, the hub automatically sends the default route via both links and the spoke can not distinguish between the two paths and will install both in its routing table. The only way to prefer one default route over the other is to use a static default route on the spoke that has a path with a lower admin distance for which you want to prefer. This automatically overrides the default routes that are coming on the spokes via ODR. Currently, the idea to provide the spoke a knob, where it can prefer one link over the other, is under consideration.

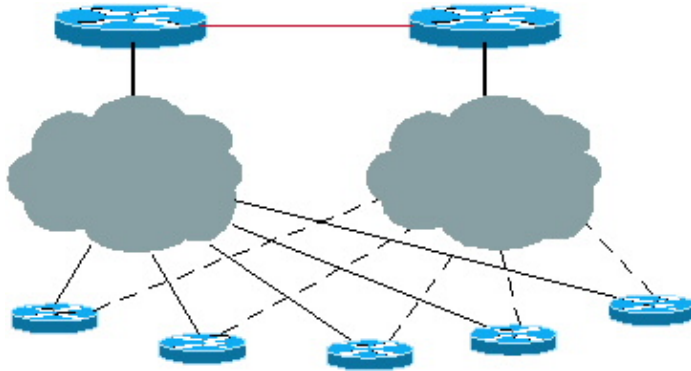
Figure 2: Spokes with multiple exit points and a single hub



Load Balancing or Backup with Multiple Hubs

These configurations can also be used for load balancing or backups when there are multiple hubs. All the hubs must be fully meshed so that if one of the links from the spokes fails, the destination can still be reached through a second hub. See the ODR vs. Other Routing Protocols section of this document for a more detailed explanation. Similarly, in the case of multiple hubs, if IOS 12.0.5T or later is in used, the hubs send the ODR default routes to spokes and spokes install both in the routing table. A future enhancement will allow a spoke to prefer one hub over the other. Currently, this can be done through a static default route defined on the spoke's router and the use of admin distance in the static route command to prefer one hub over the other. This does not affect load balancing situations.

Figure 3: Spokes with multiple exit points and multiple hubs



ODR vs. Other Routing Protocols

The biggest advantage of ODR over IP routing is that the hub router will learn IP prefixes without enabling routing protocols on Layer 3. The ODR updates are part of CDP on Layer 2.

ODR vs. EIGRP

In a true hub-and-spoke environment, it is unnecessary to pass all routing information to all spokes. Slow-link spokes waste bandwidth in routing updates and maintaining neighbor relationships. By enabling Enhanced Interior Gateway Routing Protocol (EIGRP) on the spokes, routing updates are sent to the spokes. In large networks, these updates become huge, waste CPU bandwidth, and may require more memory on spoke routers.

A better approach with EIGRP is to apply filters at the hub. The routing information is controlled so that hubs only send a default route dynamically to the spokes. These filters help reduce the size of the routing table on the spoke side, but if the hub loses a neighbor, it will send out queries to all the other neighbors. These queries are unnecessary because the hub will never get a reply from a neighbor.

The best approach is to eliminate the overhead of EIGRP queries and neighbor maintenance using ODR. By adjusting the ODR timers, the convergence time can be increased.

Today, we have a new feature in EIGRP which scales EIGRP much better in a hub and spoke situation. Refer to Enhanced IGRP Stub Routing for more information on the EIGRP stub feature.

ODR vs. OSPF

Open Shortest Path First (OSPF) offers several options for hub-and-spoke environments, and the **stub no-summary** option has the least overhead.

You may encounter problems when running OSPF on large-scale hub-and-spoke networks. The examples in this section use Frame Relay because it is the most common hub-and-spoke topology.

OSPF Point-to-Point Stub Networks

In this example, OSPF is enabled on 100 spokes connected by a point-to-point configuration. First, there are a lot of wasted IP addresses, even if we subnet with a /30 network mask. Second, if we include those 100 spokes in one area and one spoke is flapped, the shortest path first (SPF) algorithm will run and can become CPU-intensive. This situation is especially true for spoke routers if the link is flapping constantly. More neighbor flaps can cause problems as far as spoke routers are concerned.

In OSPF, the area is stub and not the interface. If there are 100 routers in a stub network, more memory is necessary on the spokes to hold the large database. This problem can be resolved by dividing a big stub area into a small areas. However, a flap in one stub area will still trigger SPF to run on the spokes, so this overhead cannot be cured by making a small stub area with no summary and no externals.

Another option is to include each link into one area. With this option, the hub router will have to run a separate SPF algorithm for each area and create a summary link-state advertisement (LSA) for routes in the area. This option can hurt the performance of the hub router.

Upgrading to a better platform is not a permanent solution; however, ODR provides a solution. Routes learned via ODR can be redistributed into OSPF to inform other hub routers about these routes.

OSPF Point-to-Multipoint Stub Networks

In point-to-multipoint networks, IP address space is saved by putting every spoke on the same subnet. Also, the size of the router LSA hub that is generated will be halved since it will generate only one stub link for all point-to-point links. A point-to-multipoint network will force the whole subnet to be included into one area. In case of a link flap, the spoke will run SPF, which can be CPU-intensive.

The Hello Storm

OSPF hello packets are small, but if there are too many neighbors, their size can become large. Since hellos are multicast, the router processes the packets. The OSPF hub sends and receives hello packets comprised of 20 bytes of IP header, 24 bytes of OSPF header, 20 bytes of hello parameters, and 4 bytes for each neighbor seen. An OSPF hello packet from a hub in a point-to-multipoint network with 100 neighbors can become 464 bytes long and will be flooded to all spokes every 30 seconds.

20 bytes IP header

24 bytes OSPF header

20 bytes hello parameters

4 bytes each neighbor router-ID (RID)

...

...

...

...

...

The overhead is resolved in ODR because no extra information is sent from the hub to the spokes. The spokes send the IP prefix of 5 bytes per subnet to the hub router. Considering the size of the hello packet, compare the 5 bytes in ODR (the spoke sending information of one connected subnet) to the 68 bytes of OSPF (the smallest hello packet size including an IP header sent from the spoke to the hub) plus 68 bytes (the smallest hello packet sent from the hub to the spoke) during a 30-second interval. Also, the OSPF hellos occur on Layer 3 while the ODR updates occur on Layer 2. With ODR, far less information is sent out, so the link bandwidth can be used for important data.

ODR vs. RIPv2

Routing Information Protocol version 2 (RIPv2) is also a good choice for hub-and-spoke environments. To design RIPv2, send the default route from the hub to the spokes. The spokes then advertise their connected interface via RIP. RIPv2 can be used when there are secondary addresses on the spokes that need to be advertised or if several vendor routers are used or if the situation is not truly a hub and spoke.

RIPv2 Over Demand Circuit

Version 2 has a few modifications, but it does not change the protocol drastically. This section discusses a few enhancements to RIP for demand circuits.

Today's internetworks are moving in the direction of dialup networks or backups of primary sites to provide connections to a large number of remote sites. Such types of connections may pass either very little or no data traffic during normal operation.

The periodic behavior of RIP causes problems on these circuits. RIP has problems with low bandwidth, point-to-point interfaces. Updates are sent every 30 seconds with large routing tables that use high bandwidth. In this situation, it is best to use Triggered RIP.

Triggered RIP

Triggered RIP is designed for routers that exchange all routing information with their neighbor. If changes in the routing take place, only the changes are propagated to the neighbor. The receiving router applies the changes immediately.

Triggered RIP updates are sent only when:

- A request for a routing update is received.
- New information is received.
- The destination has changed from circuit down to circuit up.
- The router is first turned on.

Following is a configuration example for Triggered RIP:

```
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)

```

The **ip rip triggered** command must be configured on the hub router's interface connecting to the spokes.

When comparing RIPv2 to ODR, ODR is a better choice because RIPv2 works on Layer 3 and ODR occurs on Layer 2. When the hub sends RIPv2 updates to more than 1000 spokes, it has to replicate the packet on Layer 3 for each spoke. ODR sends nothing from the hub except the usual CDP update every minute on Layer 2, which is not CPU intensive at all. Sending connected subnet information in Layer 2 from the spoke is far less CPU intensive than sending RIPv2 on Layer 3.

Large-Scale Network Design with ODR

ODR works better in a large-scale network than any other routing protocol. The biggest advantage of ODR is that routing protocols do not have to be enabled on the connected serial links. Currently, there are no routing protocols able to send routing information without enabling them on the connected interface.

ODR with EIGRP Running on Hubs

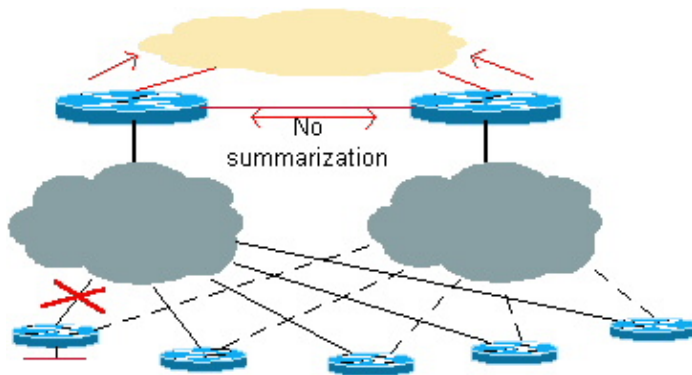
When running EIGRP, make a passive interface connection to the hub-and-spoke network so that it won't send the unnecessary EIGRP hellos on the link. If possible, it is better not to put network statements for networks between the hub and spokes because, if the link goes down, EIGRP won't send out unnecessary queries to the core neighbors. Always choose a bogus network between the hub and spokes so those links won't be included in the EIGRP domain because you won't put network statements in the configurations.

Redundancy and Summarization

In a single hub situation, no extra settings are required. Summarize the specific, connected subnets of the spokes and leak them into the core. The overheads of queries, however, will always be there. If specific routes are lost from one of the spokes, send out the queries to all the neighbors in the core routers.

In the case of multiple hubs, it is very important that both hubs are connected and that EIGRP is running between the hubs. If possible, this link should be a unique major net so that it will not interfere with other links going to the spokes. This configuration is necessary because EIGRP cannot be enabled on a specific interface, so even if we make the interface passive, it will still be advertised via EIGRP. If the interface is summarized, queries will still be sent out if one spoke is lost. As long as the link between the two hubs is not in the same major net as the spokes, the configuration should work properly.

Figure 4: Redundancy and Summarization: The core is receiving summarized routes



An advantage of EIGRP is that it can summarize on the interface level, so the summarized route of spoke subnets will be sent into the core and it will send a more specific route to the other hub. If the link between a hub and spoke goes down, it is possible to reach the destination via the second hub.

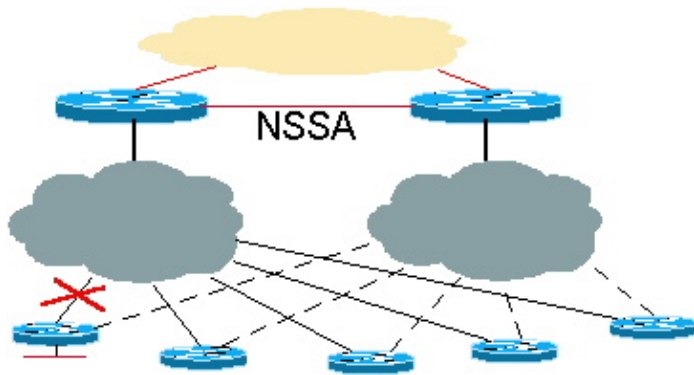
ODR with OSPF Running on Hubs

In this scenario, OSPF does not have to be enabled on the link connecting the spokes. In a normal scenario, if OSPF is enabled on the link, and one specific link is constantly flapping, it can cause several problems, including SPF execution, router LSA regeneration, summary LSA regeneration, and so on. When running ODR, do not include the connected serial link in the OSPF domain. The main concern is to receive the LAN segment information of the spokes. This information can be obtained through ODR. If one link is constantly flapping, it will not interfere with the routing protocol in the hub router.

Redundancy and Summarization

All of the specific links can be summarized before leaking into the core to avoid route calculation if one of the connected interfaces of a spoke goes down. It cannot be detected if the core router information is summarized.

Figure 5: Redundancy and Summarization: The core router is receiving summarized routes



In this example, it is very important for the hubs to be connected to each other for redundancy purposes. This connection will also summarize the spoke-connected subnets before leaking into the OSPF core.

NSSA with Future Enhancement

There will eventually be an OSPF Not-So-Stubby Areas (NSSA) feature that will allow not only summarizing into the core, but also more specific information across the hub through the NSSA link. The advantage of running NSSA is that the summarized routes can be sent into the core. Then, the core can send the traffic to either hub to reach the spoke's destination. If the link between a hub and a spoke goes down, there will be a more specific Type 7 LSA in both hubs to reach the destination through other hub.

Following is a configuration example using NSSA:

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
```

Summarization and Future Enhancement with NSSA

Assign a contiguous block of subnets to the spokes so that those subnets can be summarized properly into the OSPF core, as shown in the following example. If the subnets are not summarized and one connected subnet goes down, then the whole core would detect it and would recalculate the routes. By sending the summary route for a contiguous block, if the spoke subnet flaps, the core will not detect it.

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
```

Distance Problem

In this example, more specific information is received from both hubs. Since the OSPF distance is 110 and the ODR distance is 160, the information will interfere with ODR when it is received from the other hub about the same subnet. The other hub will always be preferred to get to the spoke destination, which will cause

suboptimal routing. To remedy the situation, decrease the ODR distance to less than 110 with the **distance** command, so that the ODR route will always be preferred over the OSPF route. If the ODR route fails, the OSPF external route will be installed into the routing table from the database.

```
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
```

The N2 routes are still in the database and will become active if the main hub link to the spoke goes down.

```
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
```

With the enhancement to NSSA, the Type 7 more specific LSA will be in the NSSA database. Instead of a summarized route, the output of the NSSA database will appear as shown below:

```
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

```

Demand Circuit

The demand circuit is a Cisco IOS 11.2 feature that can also be used in hub-and-spoke networks. This feature is typically useful in dial backup scenarios and in X.25 or Frame Relay switched virtual circuit (SVC) environments. Following is a configuration example of a demand circuit:

```

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)

```

Using the demand circuit feature in a hub-and-spoke network will bring the circuit up and form new adjacency if there is any change in the topology. For example, if there is a subnet in a spoke that flaps, the demand circuit will bring up the adjacency and flood this information. In a stubby-area environment, this information will be flooded throughout the stub area. ODR solves this problem by not leaking this information to the other spokes. Refer to OSPF Demand Circuit Feature for more information.

ODR with Point-to-Point Networks

The current Cisco IOS 12.0 status on Interface Descriptor Block (IDB) limits is as follows:

| Router | Limit |
|--------|-------|
| 1000 | 300 |
| 2600 | 300 |
| 3600 | 800 |
| 4x00 | 300 |
| 5200 | 300 |
| 5300 | 700 |
| 5800 | 3000 |
| 7200 | 3000 |
| RSP | 1000 |

Prior to IOS 12.0, the maximum number of spokes a hub could support was 300 due to IDB limits. If a network required more than 300 spokes, then point-to-point configuration wasn't a good choice. Also, a separate CDP packet was generated for each link. The time complexity for sending CDP updates on point-to-point links is n^2 . The table above gives us the IDB limits for different platforms. The maximum number of spokes supported on each platform varies, but the overhead of creating a separate CDP packet for each link is still an issue. Therefore, in a large hub and spoke situation, configuring a point-to-multipoint interface is a better solution than a point-to-point interface.

ODR with Point-to-Multipoint Networks

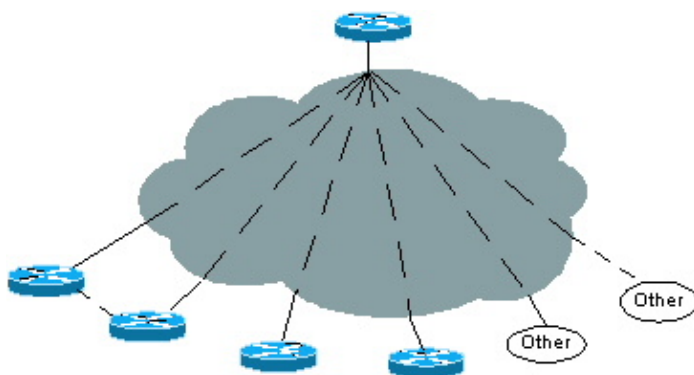
In a point-to-multipoint network where a hub supports multiple spokes, there are three major issues:

- The hub can easily support more than 300 spokes. For example, a 10.10.0.0/22 network would be able to support 1024-2 spokes with a multipoint interface.
- In a multipoint environment, one CDP packet is generated for all neighbors and is replicated on Layer 2. The time complexity of the CDP update is reduced to n .
- In a point-to-multipoint configuration, you can assign only one subnet to all spokes.

ODR and Multiple Vendors

One common misconception is that ODR will not work if multiple vendors are used. ODR will work as long as the network is a true hub-and-spoke network. For example, if there are 100 spokes and two of the spokes are routers from a different vendor, then it is possible to enable a routing protocol on those links connecting to the different routers and still run ODR on the remaining 98 Cisco spokes.

Figure 6: ODR with multiple vendors

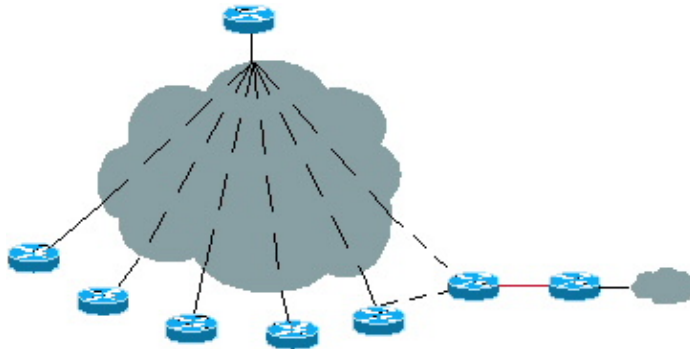


The hub router connected to the 98 Cisco routers will receive subnet updates through ODR and will receive routing protocol updates from the remaining two different routers. The links connecting to the different routers must be on separate point-to-point or point-to-multipoint subnets.

Future Growth Issues

If an organization is running ODR on 100 spokes, they eventually may want to change their topology from a hub-and-spoke network. For example, they may decide to upgrade one spoke to a bigger platform, making that spoke a hub for 20 other new spokes.

Figure 7: Future growth



It is possible to run a routing protocol on the new hub and still keep the ODR design as it is. If the new hub supports 20 or more new spokes, ODR can run on the new hub. The new hub can learn about those new spoke subnets through ODR and redistribute this information to the original hub through another routing protocol.

This situation is similar when ODR begins with two hubs. There is no overhead of changing protocols. Basically, ODR can run as long as the router is a stub.

Performance

Several settings can be adjusted for faster convergence and better performance when running ODR.

Timers Adjustment for Faster Convergence

In a large ODR environment, adjust the ODR timers for faster convergence and increase the timers of the CDP update from the hub to the spoke to minimize the CPU performance of the hub.

Hub Router

The CDP update timer should default to 60 seconds to decrease the amount of traffic from the hub to the spokes. The holdtime should be increased to the maximum (255 seconds). Since the hub router has to maintain too many CDP adjacency tables and in case a few neighbors go down, do not delete the CDP entries from the memory for 255 seconds (the maximum holddown time allowed). This configuration will give flexibility to the hub router because if the neighbor comes back up within four minutes, CDP adjacency will not have to be recreated. The old table entry can be used and the holddown timer can be updated.

Following is an example of an IP configuration template for a central router:

```
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

```

There are three permanent virtual circuits (PVCs) from each remote site (warehouse, region, and depot). Two of the PVCs go to two separate central routers. The third PVC goes to a PayPoint router. It is required that the PVC to the PayPoint route be used for traffic destined for the PayPoint network. The other two PVCs serve primary and backup functions for all other traffic. Based on these requirements, see the configuration template below for each remote router.

It is very important to adjust the ODR timers such as **invalid**, **holddown**, and **flush** for faster convergence. Even though CDP does not send out an IP prefix once **router odr** is configured, the ODR update timer should match the neighbor CDP update timer because the convergence timer can only be set if there is an update timer. This timer is different than the CDP timer and can only be used for faster convergence.

Spoke Router

Since the spokes are sending ODR updates in CDP packets, the timer for CDP updates should be kept very small for faster convergence. In a true spoke environment, there is no restriction for **holddown** time for the CDP neighbor, since there are just a few entries for the spoke to keep in its CDP table. The maximum holddown time of 255 seconds is recommended so that if the hub PVC goes down and comes back within four minutes, no new CDP adjacency is needed because the old table entry can be used.

Following is an example of an IP configuration template for a remote site:

```

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

```

The static default routes are not required if you are using IOS 12.0.5T or later since the hub router sends the default route automatically toward all spokes.

Filter and Summarization of ODR Routes

ODR routes can be filtered before they are leaked into the core. Use the **distribute-list in** command. All connected subnets of the spokes should be summarized when leaking into the core. If summarization is not possible, then unnecessary routes can be filtered at the hub router. In multiple hub networks, the spokes may advertise the connected interface which is the link to another hub.

In this situation, the **distribute-list** command must be applied so that the hub does not put those routes into the routing table. When ODR is redistributed into the hub, that information is not leaked into the core.

Telco Timer Adjustment

It is important to adjust the telco timer to increase the convergence time for the spokes. If the PVC from the hub side goes down, the spokes should be able to detect it quickly to switch to the second hub.

CPU Performance

The ODR process does not take lot of CPU utilization. ODR has been tested for approximately 1000 neighbors with CPU utilization of three to four percent. The aggressive timer setting of the ODR on the hub helps with faster convergence. If the default settings are used, the CPU utilization remains at zero to one percent.

Even with aggressive ODR and CDP timers, the output below shows that there was not a high CPU utilization. This test was performed with a 150 MHz processor on a Cisco 7206.

```

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

```

Enhancements

The version of ODR prior to Cisco IOS 12.0.5T had a few limitations. Following is the list of enhancements in Cisco IOS 12.0.5T and higher:

- Prior to CSCdy48736, secondary subnets get advertised as /32 in a CDP update. This is fixed in 12.2.13T and later.
 - CDP hubs now propagate default routes to the spokes, so there is no need to add static default routes in the spokes. The convergence time increases significantly. When the next hop goes down, the spoke quickly detects it via ODR and converges. This feature is added in 12.0.5T through bug CSCdk91586.
 - If the link between the hub and spoke is IP unnumbered, the default route sent by the hub may not be seen at the spokes. This bug, CSCdx66917, is fixed in IOS 12.2.14, 12.2.14T, and later.
 - To increase/decrease the ODR distance on the spokes so they can prefer one hub over the other, a suggestion has been made which is being tracked via CSCdr35460. The code has already been tested and will be available soon for customers.
-

Related Information

- [IP Routing Support Page](#)
 - [Technical Support – Cisco Systems](#)
-

[Contacts & Feedback](#) | [Help](#) | [Site Map](#)

© 2007 – 2008 Cisco Systems, Inc. All rights reserved. [Terms & Conditions](#) | [Privacy Statement](#) | [Cookie Policy](#) | [Trademarks of Cisco Systems, Inc.](#)

Updated: Aug 10, 2005

Document ID: 13710
